

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC ỨNG DỤNG



**BÀI TẬP LỚN
XÁC SUẤT THỐNG KÊ – MT2013**

**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH
ĐỀ TÀI: LINH KIỆN MÁY TÍNH (CPU VÀ GPU)**

Giảng viên hướng dẫn: ThS. Nguyễn Kiều Dung

Lớp: L01 – L02 – L04 – L05

Nhóm: MT01

TP. Hồ Chí Minh, tháng 11/2023



Mô tả đóng góp

STT	Thành viên nhóm	MSSV	Lớp	Mô tả đóng góp	Điểm chia (±2)
1	Ngô Trường Bách	2210183	L01	Long description text goes here, which will automatically wrap to fit within the specified width of the column.	0
2	Dương Minh Hiếu	2210978	L04	Long description text goes here, which will automatically wrap to fit within the specified width of the column.	0
3	Nguyễn Huy Hoàng	2011230	L02	Long description text goes here, which will automatically wrap to fit within the specified width of the column.	0
4	Lê Đăng Huy	2211186	L02	Long description text goes here, which will automatically wrap to fit within the specified width of the column.	0
5	Lê Minh Thê	2213218	L05	Long description text goes here, which will automatically wrap to fit within the specified width of the column.	0



Mục lục

1	Tổng quan dữ liệu	8
2	Kiến thức nền	9
2.1	Lý thuyết mẫu	9
2.1.1	Một số khái niệm	9
2.1.2	Phương pháp chọn mẫu	9
2.1.3	Các đặc trưng của tổng thể – các đặc trưng của mẫu	9
2.2	Lý thuyết ước lượng	10
2.2.1	Dẫn nhập	10
2.2.2	Một số khái niệm	10
2.2.3	Các phương pháp ước lượng	11
2.2.4	Bài toán tìm khoảng tin cậy đối xứng cho trung bình tổng thể	12
2.3	Kiểm định giả thuyết thống kê	15
2.3.1	Một số khái niệm	16
2.3.2	Bài toán kiểm định trung bình	16
2.4	Phân tích phương sai	20
2.4.1	Khái niệm	20
2.4.2	Phân loại	21
2.4.3	Phân tích phương sai một yếu tố	21
2.5	Hồi quy tuyến tính	26
2.5.1	Hồi quy tuyến tính đơn	27
2.5.2	Hồi quy tuyến tính bội	32
3	Tiền xử lý số liệu	33
3.1	Đọc dữ liệu	33
3.2	Trích xuất các tiêu thức quan trọng của dữ liệu	34
3.3	Xử lý định dạng dữ liệu	35
3.4	Xử lý dữ liệu khuyết	38
4	Thống kê tóm tắt	41
4.1	Các giá trị đặc trưng của mẫu	41
4.2	Phân phối tần số	41
4.3	Phân phối chuẩn	43
4.4	Mối liên hệ giữa các biến	44
5	Thống kê suy diễn	47
5.1	Tìm khoảng tin cậy một mẫu	47
5.1.1	Mục tiêu	47



5.1.2	Bài toán	47
5.1.3	Nhận xét bài toán	47
5.1.4	Kiến thức R	47
5.1.5	Tiến hành	48
5.1.6	Kết quả	48
5.1.7	Nhận xét	48
5.1.8	Kết luận	48
5.2	Kiểm định hai mẫu	49
5.2.1	Mục tiêu	49
5.2.2	Bài toán	49
5.2.3	Nhận xét bài toán	49
5.2.4	Kiến thức R	49
5.2.5	Tiến hành	50
5.2.6	Kết quả	51
5.2.7	Nhận xét	51
5.2.8	Kết luận	51
5.3	Phân tích phương sai	51
5.3.1	Mục tiêu	51
5.3.2	Bài toán	52
5.3.3	Kiểm tra giả thiết áp dụng mô hình ANOVA	52
5.3.4	Kiến thức R	52
5.3.5	Tiến hành	52
5.3.6	Kết luận	56
5.4	Hồi quy tuyến tính	57
5.4.1	Mục tiêu	57
5.4.2	Bài toán	57
5.4.3	Kiến thức R	57
5.4.4	Tiến hành	57



1 Tổng quan dữ liệu

Tập tin **All_GPUs.csv** chứa 34 thông số của hơn 3400 bộ xử lý đồ họa GPU - Graphic Processing Unit (một bộ phận vi mạch của máy tính có chức năng chuyên dụng để thao tác xử lý hình ảnh). Dữ liệu được đưa ra ở đây chủ yếu thuộc về Intel, Game-Debate và các công ty liên quan đến việc sản xuất bộ phận này. Dữ liệu gốc được cung cấp tại: <http://www.futureelectronics.com/en/Microprocessors/embedded-processors.aspx>.

Tập tin chứa một vài thông số quan trọng của GPU có thể kể đến như:

- **Name:** Tên mẫu GPU.
- **Best Resolution:** Là một thuộc tính cho biết độ phân giải của máy tính để GPU có thể hoạt động hiệu quả nhất. Độ phân giải (resolution) là số lượng điểm ảnh (pixels) trên màn hình mà GPU có khả năng hiển thị. Độ phân giải được biểu thị bằng hai giá trị: chiều rộng (số pixel theo chiều ngang) và chiều cao (số pixel theo chiều dọc). Ví dụ, độ phân giải thông dụng là 1920x1080, tức là có 1920 pixel theo chiều ngang và 1080 pixel theo chiều dọc.
- **Manufacturer:** Công ty hoặc tổ chức sản xuất và phân phối GPU. Các nhà sản xuất GPU nổi tiếng như NVIDIA, AMD, Intel,... Mỗi nhà sản xuất có sản phẩm GPU riêng của họ với các đặc điểm và tính năng riêng biệt.
- **Core Speed [MHz]:** Là tốc độ hoạt động của các lõi xử lý (cores) trên GPU. Đây là một trong những thông số quan trọng để đánh giá hiệu năng của GPU. Thông số này được sử dụng để đo lường khả năng xử lý tính toán của GPU. Tốc độ lõi cao hơn thường cho phép GPU thực hiện nhiều phép tính trong một khoảng thời gian ngắn hơn, cải thiện hiệu năng tổng thể. Tốc độ này phụ thuộc vào nhiều yếu tố, trong đó có loại kiến trúc dùng để xây dựng nên GPU.
- **Memory [MB hoặc GB]:** Dung lượng bộ nhớ truy cập đồ họa của GPU. Đây là một trong những thuộc tính quan trọng của GPU, ảnh hưởng trực tiếp đến khả năng xử lý đồ họa của máy tính. Dung lượng này cho biết khả năng lưu trữ các dữ liệu và tài nguyên đồ họa mà GPU có thể sử dụng trong quá trình hoạt động.
- **Memory Bandwidth [GB/giây]:** Memory Bandwidth của GPU là một chỉ số quan trọng của GPU, do lường khả năng truy xuất dữ liệu từ bộ nhớ của GPU. Băng thông bộ nhớ càng cao, GPU có khả năng nhanh chóng truy cập và đọc/ghi dữ liệu từ bộ nhớ, tăng hiệu suất trong các tác vụ.
- **Release Date:** Ngày phát hành mẫu GPU.

Ngoài ra, tập tin còn chứa các thông số khác như **Architecture** (Kiểu kiến trúc xây dựng GPU), **Pixel Rate** (Tỉ lệ điểm ảnh), **PSU** (Mức điện năng sử dụng),...



2 Kiến thức nền

2.1 Lý thuyết mẫu

2.1.1 Một số khái niệm

Tổng thể thống kê là tập hợp các phần tử thuộc đối tượng nghiên cứu, cần được quan sát, thu thập và phân tích theo một hoặc một số đặc trưng nào đó.

Đơn vị tổng thể là các phần tử tạo thành tổng thể thống kê.

Mẫu là một số đơn vị được chọn ra từ tổng thể theo một phương pháp lấy mẫu nào đó. Các đặc trưng mẫu được sử dụng để suy rộng ra các đặc trưng của tổng thể nói chung.

Đặc điểm thống kê (tiêu thức) gồm các tính chất quan trọng liên quan trực tiếp đến nội dung nghiên cứu và khảo sát cần thu thập dữ liệu trên các đơn vị tổng thể. Tiêu thức được chia làm 2 loại: tiêu thức thuộc tính và tiêu thức số lượng.

2.1.2 Phương pháp chọn mẫu

Việc nghiên cứu toàn bộ tổng thể chỉ phù hợp khi kích thước tổng thể nhỏ, có được sự kết hợp với các khảo sát quy mô lớn, có sự hỗ trợ của công nghệ trong việc thu thập và xử lý số liệu lớn,... Trong những trường hợp phổ biến hơn, người ta áp dụng phương pháp nghiên cứu không toàn bộ, đặc biệt là phương pháp chọn mẫu.

Mẫu mà ta nghiên cứu được chọn theo một cách nào đó mang tính ngẫu nhiên, khách quan, gọi là mẫu ngẫu nhiên. Nếu mẫu được chọn ra một cách ngẫu nhiên và xử lý bằng các phương pháp xác suất thì thu được kết luận một cách nhanh chóng, đỡ tốn kém mà vẫn đảm bảo độ chính xác cần thiết.

Phân loại mẫu theo phương pháp chọn mẫu:

- *Mẫu không hoàn lại (mẫu không lặp)* là mẫu được chọn bằng cách phần tử đã lấy ra quan sát thì loại khỏi tổng thể rồi mới lấy phần tử tiếp theo. Trong mẫu không hoàn lại, mỗi phần tử của tổng thể chỉ được chọn một lần.
- *Mẫu hoàn lại (mẫu lặp)* là mẫu được chọn bằng cách phần tử đã lấy ra quan sát được bỏ trở lại tổng thể rồi mới lấy phần tử tiếp theo. Do đó, một phần tử của tổng thể có thể được chọn nhiều lần.

Về mặt lý thuyết, ta giả định rằng các phần tử được lấy vào mẫu theo phương thức có hoàn lại và mỗi phần tử của tổng thể đều được lấy vào mẫu với khả năng như nhau.

2.1.3 Các đặc trưng của tổng thể – các đặc trưng của mẫu

Kích thước tổng thể là số lượng các phần tử của tổng thể. Trong nhiều trường hợp, ta không biết được chính xác kích thước tổng thể.

Khi khảo sát tổng thể theo một dấu hiệu nghiên cứu nào đó, người ta mô hình hóa nó bởi một biến ngẫu nhiên X , gọi là **biến ngẫu nhiên gốc**.

Mẫu ngẫu nhiên 1 chiều kích thước n là tập hợp của n biến ngẫu nhiên độc lập X_1, X_2, \dots, X_n được thành lập từ biến ngẫu nhiên X của tổng thể nghiên cứu và có cùng quy luật phân phối xác suất với X .

Kí hiệu của mẫu tổng quát kích thước n là: $W = (X_1, X_2, \dots, X_n)$.

Việc thực hiện một phép thử đối với mẫu ngẫu nhiên W chính là thực hiện một phép thử đối với mỗi thành phần X_i . Ta gọi kết quả $w_n = (x_1, x_2, \dots, x_n)$ tạo thành là **mẫu cụ thể**.

Một mẫu cụ thể có thể được biểu diễn bằng **bảng phân phối tần số thực nghiệm**:

x_i	x_1	x_2	\dots	x_k
n_i	n_1	n_2	\dots	n_k

Các đặc trưng cơ bản của tổng thể và mẫu cụ thể:

Đặc trưng	Tổng thể	Mẫu
Trung bình	μ	$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
Phương sai	σ^2	$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$
Dộ lệch	σ	$s = \sqrt{s^2}$
Sai số chuẩn	$SE(X) = \frac{\sigma}{\sqrt{n}}$	$SE(\bar{x}) = \frac{s}{\sqrt{n}}$
Tỉ lệ phần tử mang dấu hiệu nghiên cứu	p	$f = \frac{m}{n}$

2.2 Lý thuyết ước lượng

2.2.1 Dẫn nhập

Lý thuyết ước lượng là một phần quan trọng của thống kê, xoay quanh việc rút ra kết luận từ dữ liệu và đưa ra ước tính về các thông số của một tổng thể dựa trên một mẫu con. Điều này đặc biệt hữu ích khi chúng ta không thể thu thập dữ liệu từ toàn bộ tổng thể và phải dựa vào mẫu nhỏ để đưa ra suy luận.

Trong lý thuyết ước lượng, ta tìm cách xây dựng các ước lượng chính xác và hiệu quả cho các thông số của tổng thể. Các phương pháp ước lượng thường dựa vào các phân phối xác suất và cơ sở lý thuyết để tính toán ước lượng và xác định sai số.

2.2.2 Một số khái niệm

Trong lý thuyết ước lượng, một số khái niệm quan trọng bao gồm:

- **Thông số ước lượng θ :** Đây là giá trị dự đoán của một thông số trong tổng thể dựa trên mẫu con. Thông số ước lượng có thể là trung bình μ , phương sai σ^2 , tỷ lệ p , hoặc bất kỳ thông số nào khác liên quan đến tổng thể.
- **Sai số ước lượng ε :** Là sự chênh lệch giữa giá trị thực tế của thông số trong tổng thể và giá trị ước lượng từ mẫu con. Sai số ước lượng thường không thể tránh khỏi, và lý thuyết ước lượng giúp đánh giá và kiểm soát sai số này.
- **Phân phối ước lượng:** Đây là phân phối xác suất của các giá trị ước lượng có thể có trong nhiều mẫu con khác nhau từ cùng một tổng thể. Phân phối ước lượng giúp ta hiểu rõ tính biến đổi của các ước lượng và xác định khoảng tin cậy cho chúng.

2.2.3 Các phương pháp ước lượng

- **Ước lượng điểm:** Là việc dùng một tham số thống kê mẫu đơn lẻ $\hat{\theta}$ để ước lượng giá trị tham số θ của tổng thể.

Ví dụ: Khảo sát ngẫu nhiên điểm thi môn Giải tích 1 của 500 sinh viên năm I tại một trường đại học, người ta tính được điểm trung bình của 500 sinh viên này là 5.2. Phương pháp ước lượng điểm cho phép ta đánh giá điểm thi trung bình môn Giải tích 1 của mỗi sinh viên năm I tại trường đại học này là 5.2.

Ước lượng không chêch: $\hat{\theta}$ gọi là ước lượng không chêch của θ nếu $E(\hat{\theta}) = \theta$. Định lí:

- Tỉ lệ mẫu con f là ước lượng không chêch tỉ lệ của tổng thể p .
- Trung bình mẫu con \bar{X} là ước lượng không chêch trung bình tổng thể μ .
- Phương sai mẫu con s^2 là ước lượng không chêch phương sai tổng thể σ^2 .

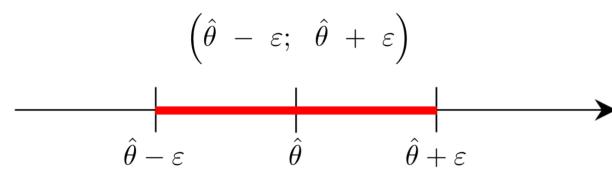
Nhược điểm: Một nhược điểm cơ bản của phương pháp ước lượng điểm là khi kích thước mẫu chưa thực sự lớn thì ước lượng điểm tìm được có thể sai lệch rất nhiều so với giá trị của tham số cần ước lượng. Hơn nữa, phương pháp ước lượng điểm không đánh giá được mức độ sai lệch là bao nhiêu.

- **Ước lượng khoảng:** Là việc tìm ra một khoảng $(G_1; G_2)$ sao cho xác suất thông số ước lượng θ thuộc vào khoảng đó là $1 - \alpha$ (độ tin cậy của ước lượng). Tức:

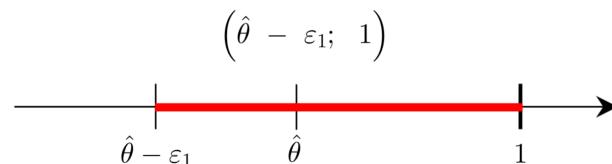
$$P(G_1 < \theta < G_2) = 1 - \alpha$$

Trong đó, α là khả năng mắc sai lần của phương pháp.

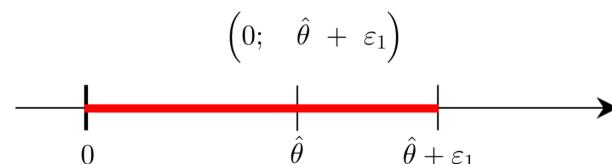
Phân loại khoảng tin cậy: Với $\hat{\theta}$ là ước lượng không chêch của θ , các khoảng tin cậy được phân loại thành khoảng tin cậy đối xứng, khoảng tin cậy bên trái và khoảng tin cậy bên phải.



Hình 2.1: Khoảng tin cậy đối xứng



Hình 2.2: Khoảng tin cậy bên trái



Hình 2.3: Khoảng tin cậy bên phải

Ưu điểm: Phương pháp ước lượng khoảng có ưu thế hơn phương pháp ước lượng điểm vì nó làm tăng độ chính xác và đánh giá được mức độ tin cậy của ước lượng.

2.2.4 Bài toán tìm khoảng tin cậy đối xứng cho trung bình tổng thể

Bài toán Cho mẫu có kích thước n , trung bình mẫu \bar{X} , phương sai mẫu s^2 (hoặc phương sai tổng thể σ^2). Tìm khoảng ước lượng đối xứng cho trung bình tổng thể μ của mẫu này với độ tin cậy $1 - \alpha$.

Phương pháp giải Như đã trình bày, khoảng tin cậy đối xứng cần tìm sẽ có dạng:

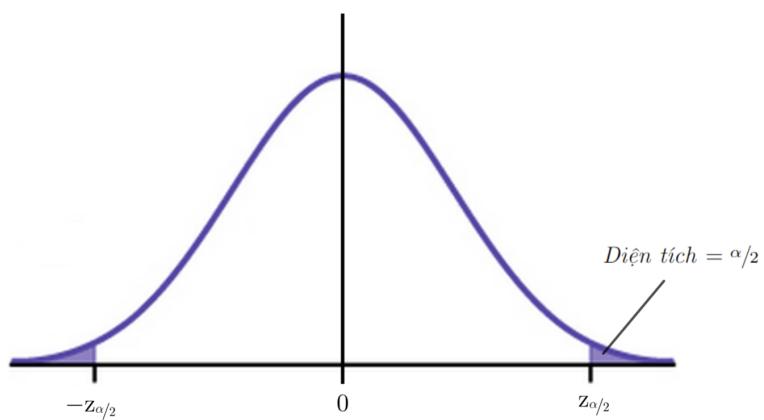
$$(\bar{X} - \varepsilon; \bar{X} + \varepsilon)$$

Với \bar{X} đã biết, ta cần tìm giá trị sai số ước lượng ε . Việc tính giá trị này sẽ rơi vào nhiều trường hợp, tùy theo dữ kiện đã biết, cụ thể:

Dạng		Công thức
X phân phối chuẩn	Biết σ^2	$\varepsilon = \frac{z_{\alpha/2} \times \sigma}{\sqrt{n}}$
	Chưa biết σ^2	$\varepsilon = \frac{t_{\alpha/2}(n-1) \times s}{\sqrt{n}}$
X phân phối tuỳ ý Mẫu lớn ($n \geq 30$)	Biết σ^2	$\varepsilon = \frac{z_{\alpha/2} \times \sigma}{\sqrt{n}}$
	Chưa biết σ^2	$\varepsilon = \frac{z_{\alpha/2} \times s}{\sqrt{n}}$

Trong đó:

- $z_{\alpha/2}$ là giá trị trong bảng phân phối chuẩn, xác định vị trí trên phân phối chuẩn để bao phủ một phần diện tích xác định.

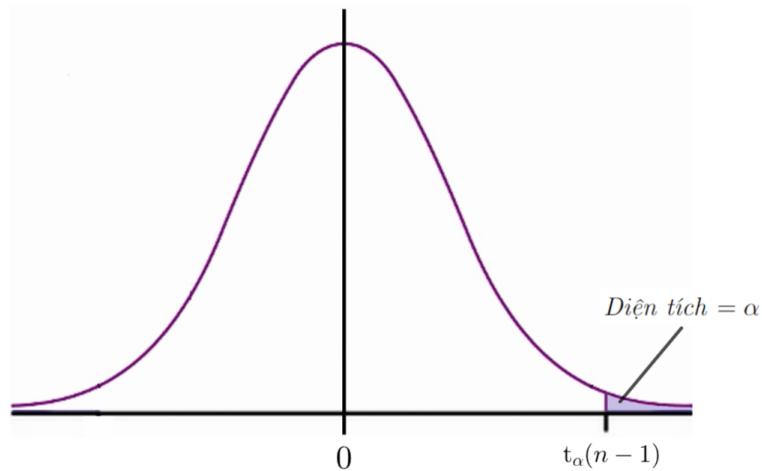


Hình 2.4: Minh họa giá trị $z_{\alpha/2}$

<i>z</i>	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.500000	0.503989	0.507978	0.511967	0.515953	0.519939	0.532922	0.527903	0.531881	0.535856
0.1	0.539828	0.543795	0.547758	0.551717	0.555760	0.559618	0.563559	0.567495	0.571424	0.575345
0.2	0.579260	0.583166	0.587064	0.590954	0.594835	0.598706	0.602568	0.606420	0.610261	0.614092
0.3	0.617911	0.621719	0.625516	0.629300	0.633072	0.636831	0.640576	0.644309	0.648027	0.651732
0.4	0.655422	0.659097	0.662757	0.666402	0.670031	0.673645	0.677242	0.680822	0.684386	0.687933
0.5	0.691462	0.694974	0.698468	0.701944	0.705401	0.708840	0.712260	0.715661	0.719043	0.722405
0.6	0.725747	0.729069	0.732371	0.735653	0.738914	0.742154	0.745373	0.748571	0.751748	0.754903
0.7	0.758036	0.761148	0.764238	0.767305	0.770350	0.773373	0.776373	0.779350	0.782305	0.785236
0.8	0.788145	0.791030	0.793892	0.796731	0.799546	0.802338	0.805106	0.807850	0.810570	0.813267
0.9	0.815940	0.818589	0.821214	0.823815	0.826391	0.828944	0.831472	0.833977	0.836457	0.838913
1.0	0.841345	0.843752	0.846136	0.848495	0.850830	0.853141	0.855428	0.857690	0.859929	0.862143
1.1	0.864334	0.866500	0.868643	0.870762	0.872857	0.874928	0.876976	0.878999	0.881000	0.882977
1.2	0.884930	0.886860	0.888767	0.890651	0.892512	0.894350	0.896165	0.897958	0.899727	0.901475
1.3	0.903199	0.904902	0.906582	0.908241	0.909877	0.911492	0.913085	0.914657	0.916207	0.917736
1.4	0.919243	0.920730	0.922196	0.923641	0.925066	0.926471	0.927855	0.929219	0.930563	0.931888
1.5	0.933193	0.934478	0.935744	0.936992	0.938220	0.939429	0.940620	0.941792	0.942947	0.944083
1.6	0.945201	0.946301	0.947384	0.948449	0.949497	0.950529	0.951543	0.952540	0.953521	0.954486
1.7	0.955435	0.956367	0.957284	0.958185	0.959071	0.959941	0.960796	0.961636	0.962462	0.963273
1.8	0.964070	0.964852	0.965621	0.966375	0.967116	0.967843	0.968557	0.969258	0.969946	0.970621
1.9	0.971283	0.971933	0.972571	0.973197	0.973810	0.974412	0.975002	0.975581	0.976148	0.976705
2.0	0.977250	0.977784	0.978308	0.978822	0.979325	0.979818	0.980301	0.980774	0.981237	0.981691
2.1	0.982136	0.982571	0.982997	0.983414	0.983823	0.984222	0.984614	0.984997	0.985371	0.985738
2.2	0.986097	0.986447	0.986791	0.987126	0.987455	0.987776	0.988089	0.988396	0.988696	0.988989
2.3	0.989276	0.989556	0.989830	0.990097	0.990358	0.990613	0.990863	0.991106	0.991344	0.991576
2.4	0.991802	0.992024	0.992240	0.992451	0.992656	0.992857	0.993053	0.993244	0.993431	0.993613
2.5	0.993790	0.993963	0.994132	0.994297	0.994457	0.994614	0.994766	0.994915	0.995060	0.995201
2.6	0.995339	0.995473	0.995604	0.995731	0.995855	0.995975	0.996093	0.996207	0.996319	0.996427
2.7	0.996533	0.996636	0.996736	0.996833	0.996928	0.997020	0.997110	0.997197	0.997282	0.997365
2.8	0.997445	0.997523	0.997599	0.997673	0.997744	0.997814	0.997882	0.997948	0.998012	0.998074
2.9	0.998134	0.998193	0.998250	0.998305	0.998359	0.998411	0.998462	0.998511	0.998559	0.998605
3.0	0.998650	0.998694	0.998736	0.998777	0.998817	0.998856	0.998893	0.998930	0.998965	0.998999
3.1	0.999032	0.999065	0.999096	0.999126	0.999155	0.999184	0.999211	0.999238	0.999264	0.999289
3.2	0.999313	0.999336	0.999359	0.999381	0.999402	0.999423	0.999443	0.999462	0.999481	0.999499
3.3	0.999517	0.999533	0.999550	0.999566	0.999581	0.999596	0.999610	0.999624	0.999638	0.999650
3.4	0.999663	0.999675	0.999687	0.999698	0.999709	0.999720	0.999730	0.999740	0.999758	
3.5	0.999767	0.999776	0.999784	0.999792	0.999800	0.999807	0.999815	0.999821	0.999828	0.999835
3.6	0.999841	0.999847	0.999853	0.999858	0.999864	0.999869	0.999874	0.999879	0.999883	0.999888
3.7	0.999892	0.999896	0.999900	0.999904	0.999908	0.999912	0.999915	0.999918	0.999922	0.999925
3.8	0.999928	0.999931	0.999933	0.999936	0.999938	0.999941	0.999943	0.999946	0.999948	0.999950
3.9	0.999952	0.999954	0.999956	0.999958	0.999959	0.999961	0.999963	0.999964	0.999966	0.999967

Hình 2.5: Bảng tra hàm phân phối chuẩn giúp tra ngược giá trị $z_{\alpha/2}$

- $t_{\alpha/2}(n-1)$ là giá trị trong bảng phân vị của hàm Student ứng với $\alpha/2$ và bậc tự do $n-1$.



Hình 2.6: Minh họa giá trị $t_{\alpha}(n-1)$

$\frac{\alpha}{v}$.40	.25	.10	.05	.025	.01	.005	.0025	.001	.0005
1	.325	1.000	3.078	6.314	12.706	31.821	63.657	127.32	318.31	636.62
2	.289	.816	1.886	2.920	4.303	6.965	9.925	14.089	23.326	31.598
3	.277	.765	1.638	2.353	3.182	4.541	5.841	7.453	10.213	12.924
4	.271	.741	1.533	2.132	2.776	3.747	4.604	5.598	7.173	8.610
5	.267	.727	1.476	2.015	2.571	3.365	4.032	4.773	5.893	6.869
6	.265	.718	1.440	1.943	2.447	3.143	3.707	4.317	5.208	5.959
7	.263	.711	1.415	1.895	2.365	2.998	3.499	4.029	4.785	5.408
8	.262	.706	1.397	1.860	2.306	2.896	3.355	3.833	4.501	5.041
9	.261	.703	1.383	1.833	2.262	2.821	3.250	3.690	4.297	4.781
10	.260	.700	1.372	1.812	2.228	2.764	3.169	3.581	4.144	4.587
11	.260	.697	1.363	1.796	2.201	2.718	3.106	3.497	4.025	4.437
12	.259	.695	1.356	1.782	2.179	2.681	3.055	3.428	3.930	4.318
13	.259	.694	1.350	1.771	2.160	2.650	3.012	3.372	3.852	4.221
14	.258	.692	1.345	1.761	2.145	2.624	2.977	3.326	3.787	4.140
15	.258	.691	1.341	1.753	2.131	2.602	2.947	3.286	3.733	4.073
16	.258	.690	1.337	1.746	2.120	2.583	2.921	3.252	3.686	4.015
17	.257	.689	1.333	1.740	2.110	2.567	2.898	3.222	3.646	3.965
18	.257	.688	1.330	1.734	2.101	2.552	2.878	3.197	3.610	3.922
19	.257	.688	1.328	1.729	2.093	2.539	2.861	3.174	3.579	3.883
20	.257	.687	1.325	1.725	2.086	2.528	2.845	3.153	3.552	3.850
21	.257	.686	1.323	1.721	2.080	2.518	2.831	3.135	3.527	3.819
22	.256	.686	1.321	1.717	2.074	2.508	2.819	3.119	3.505	3.792
23	.256	.685	1.319	1.714	2.069	2.500	2.807	3.104	3.485	3.767
24	.256	.685	1.318	1.711	2.064	2.492	2.797	3.091	3.467	3.745
25	.256	.684	1.316	1.708	2.060	2.485	2.787	3.078	3.450	3.725
26	.256	.684	1.315	1.706	2.056	2.479	2.779	3.067	3.435	3.707
27	.256	.684	1.314	1.703	2.052	2.473	2.771	3.057	3.421	3.690
28	.256	.683	1.313	1.701	2.048	2.467	2.763	3.047	3.408	3.674
29	.256	.683	1.311	1.699	2.045	2.462	2.756	3.038	3.396	3.659
30	.256	.683	1.310	1.697	2.042	2.457	2.750	3.030	3.385	3.646
40	.255	.681	1.303	1.684	2.021	2.423	2.704	2.971	3.307	3.551
60	.254	.679	1.296	1.671	2.000	2.390	2.660	2.915	3.232	3.460
120	.254	.677	1.289	1.658	1.980	2.358	2.617	2.860	3.160	3.373
∞	.253	.674	1.282	1.645	1.960	2.326	2.576	2.807	3.090	3.291

v = degrees of freedom.

Hình 2.7: Bảng phân vị của hàm Student

Khi đã tìm được sai số ước lượng ε , ta kết luận được khoảng tinh cận đối xứng cho trung bình tổng thể.

2.3 Kiểm định giả thuyết thống kê

Kiểm định giả thuyết thống kê là dùng các thống kê từ một mẫu để khẳng định hay bác bỏ một giả thuyết nào đó nói về tổng thể.

Giả sử cần kiểm định một giả thuyết H . Khi điểm định có thể xảy ra một trong hai loại sai lầm sau đây:

- Loại 1: bác bỏ H trong lúc H đúng.
- Loại 2: chấp nhận H trong lúc H sai.

Phương pháp chung để kiểm định là cho phép xác suất xảy ra sai lần loại 1 không quá α , số α gọi là mức ý nghĩa của kiểm định. Với mức ý nghĩa đã cho, ta chấp nhận H nếu xác suất xảy ra sai lầm lại 2 nhỏ nhất.

2.3.1 Một số khái niệm

- **Giả thuyết không H_0 (Null Hypothesis)**: là giả thuyết về yếu tố cần kiểm định của tổng thể ở trạng thái bình thường, không chịu tác động của các hiện tượng liên quan. Yếu tố trong H_0 phải được xác định cụ thể, ví dụ:
 - Tỉ lệ sinh viên của một trường đại học đạt điểm thi môn Giải tích 1 trên trung bình trong một học kì là 65%.
 - Tỉ lệ nảy mầm của một hạt giống là 89%
 - Chiều cao của trẻ em 5 tuổi tuân theo phân phối chuẩn với kỳ vọng là 110 cm và phương sai là 15 cm².
- **Giả thuyết đối H_1 (Alternative Hypothesis)**: là một mệnh đề mâu thuẫn với H_0 , thể hiện xu hướng cần kiểm định.
- **Tiêu chuẩn kiểm định**: là hàm thống kê $G = G(X_1, X_2, \dots, X_n, \theta_0)$ được xây dựng trên mẫu ngẫu nhiên $W = (X_1, X_2, \dots, X_n)$ và tham số θ_0 liên quan đến H_0 . Điều kiện đặt ra với thống kê G là nếu H_0 đúng thì quy luật phân phối xác suất của G phải hoàn toàn xác định.
- **Miền bác bỏ giả thuyết RR (Rejection region)**: là miền số thực thỏa xác xuất G thuộc vào đó với điều kiện H_0 đúng là α . Tức

$$P(G \in \text{RR} \mid H_0 \text{ đúng}) = \alpha$$

- **Quy tắc kiểm định**: Từ mẫu thực nghiệm, ta tính được một giá trị cụ thể của tiêu chuẩn kiểm định, gọi là **giá trị kiểm định thống kê**:

$$g_{qs} = G(x_1, x_2, \dots, x_n, \theta_0).$$

Theo nguyên lý xác suất bé, biến cố $G \in \text{RR}$ có xác xuất nhỏ nên với một mẫu thực nghiệm ngẫu nhiên, nó không thể xảy ra. Do đó:

- Nếu $g_{qs} \in \text{RR}$: bác bỏ giả thuyết H_0 , thừa nhận giả thuyết H_1 .
- Nếu $g_{qs} \notin \text{RR}$: chưa đủ dữ liệu khẳng định H_0 sai. Vì vậy ta chưa thể chứng minh được H_1 đúng.

2.3.2 Bài toán kiểm định trung bình

Với giới hạn bài tập lớn này, nhóm tập trung nghiên cứu và trình bày về bài toán kiểm định mà ở đó yếu tố cần kiểm định là trung bình của tổng thể.

2.3.2.1 Bài toán kiểm định trung bình một mẫu

Bài toán Giả sử giả thuyết cho rằng một tổng thể có trung bình μ_0 , biết (hoặc không biết) phương sai σ^2 . Một mẫu có kích thước n , trung bình mẫu \bar{X} , phương sai s^2 được chọn ra từ tổng thể. Trên mẫu



đã chọn, hãy kiểm định với mức ý nghĩa α , giả thuyết trung bình μ của tổng thể có đáng tin cậy hay không.

Các bước giải

- Đặt giả thuyết kiểm định

Giả thuyết không H_0 :

$$\mu = \mu_0$$

Giả thuyết đối H_1 : Tuỳ theo yếu tố cần kiểm định mà đặt H_1 có thể là

$$\mu \neq \mu_0 \quad \text{hoặc} \quad \mu < \mu_0 \quad \text{hoặc} \quad \mu > \mu_0$$

- Tìm miền bác bỏ RR

Miền bác bỏ H_0 cho bài toán kiểm định trung bình một mẫu phụ thuộc vào giả thuyết đối H_1 và dữ kiện đã biết. Cụ thể:

Dạng		H_1	Miền bác bỏ H_0
X phân phối chuẩn	Biết σ^2	$\mu \neq \mu_0$	$(-\infty; -z_{\alpha/2}) \cup (z_{\alpha/2}; +\infty)$
		$\mu < \mu_0$	$(-\infty; -z_\alpha)$
		$\mu > \mu_0$	$(z_\alpha; +\infty)$
	Chưa biết σ^2	$\mu \neq \mu_0$	$(-\infty; -t_{\alpha/2}(n-1)) \cup (t_{\alpha/2}(n-1); +\infty)$
		$\mu < \mu_0$	$(-\infty; -t_\alpha(n-1))$
		$\mu > \mu_0$	$(t_\alpha(n-1); +\infty)$
X phân phối tuỳ ý Mẫu lớn ($n \geq 30$)	$\mu \neq \mu_0$	$(-\infty; -z_{\alpha/2}) \cup (z_{\alpha/2}; +\infty)$	
	$\mu < \mu_0$	$(-\infty; -z_\alpha)$	
	$\mu > \mu_0$	$(z_\alpha; +\infty)$	

- Tính giá trị kiểm định thống kê

Hàm kiểm định thống kê cho bài toán kiểm định trung bình một mẫu rơi vào nhiều trường hợp, tuỳ theo dữ kiện đã biết, cụ thể:

Dạng		Tiêu chuẩn kiểm định
X phân phối chuẩn	Biết σ^2	$Z_{qs} = \frac{\bar{X} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$
	Chưa biết σ^2	$T_{qs} = \frac{\bar{X} - \mu_0}{\frac{s}{\sqrt{n}}}$
X có phân phối tuỳ ý Mẫu lớn ($n \geq 30$)	Biết σ^2	$Z_{qs} = \frac{\bar{X} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$
	Chưa biết σ^2	$T_{qs} = \frac{\bar{X} - \mu_0}{\frac{s}{\sqrt{n}}}$

- Kết luận

- Nếu Z_{qs} (hoặc T_{qs}) $\in RR$: bác bỏ giả thuyết H_0 , thừa nhận giả thuyết H_1 . Tức kết luận với mức ý nghĩa α , giả thuyết trung bình của tổng thể là μ là không đáng tin cậy.
- Nếu Z_{qs} (hoặc T_{qs}) $\notin RR$: chưa đủ dữ liệu khẳng định H_0 sai. Tức với mức ý nghĩa α , chưa thể nói giả thuyết trung bình của tổng thể là μ là không đáng tin cậy.

2.3.2.2 Bài toán kiểm định trung bình hai mẫu

Hai mẫu độc lập – hai mẫu phụ thuộc tương ứng theo cặp

- Hai mẫu được gọi là độc lập khi sự thay đổi trong một mẫu không ảnh hưởng đến sự thay đổi trong mẫu kia. Nói cách khác, sự kiện trong một mẫu không tác động lên sự kiện trong mẫu kia và ngược lại.

Ví dụ: Mẫu chiều cao của 50 sinh viên nam và chiều cao của 50 sinh viên nữ tại một trường đại học là hai mẫu độc lập.

- Hai mẫu được gọi là tương ứng phụ thuộc theo cặp khi kích thước của chúng bằng nhau và có một mối liên quan hay sự phụ thuộc giữa chúng. Điều này có nghĩa là sự kiện trong một mẫu có thể ảnh hưởng đến sự kiện trong mẫu kia và ngược lại.

Ví dụ: Khảo sát độ tuổi và chiều cao của 100 thanh thiếu niên tại một vùng. Khi đó, ống với mỗi phần tử trong mẫu, hai tiêu thức độ tuổi và chiều cao có xu hướng phụ thuộc lẫn nhau (độ tuổi tăng thì chiều cao cũng có xu hướng tăng). Hơn nữa, do khảo sát được thực hiện trên cùng 100 thanh thiếu niên nên kích thước của hai mẫu bằng nhau. Do đó, hai mẫu này là phụ thuộc tương ứng theo cặp.

Để kiểm tra tính độc lập (hay phụ thuộc) giữa hai mẫu, ta sử dụng kiểm định Chi bình phương (Chi-square – χ^2)

Trong giới hạn bài tập lớn này với bài toán kiểm định hai mẫu, nhóm chỉ xét trường hợp hai độc lập nhau.

Bài toán Giả sử giả thuyết cho rằng tổng thể I có trung bình μ_1 , tổng thể II có trung bình μ_2 . Biết (hoặc chưa biết) phương sai hai tổng thể σ_1^2, σ_2^2 . Từ hai tổng thể, có hai mẫu kích thước lần lượt n_1 và n_2 độc lập, có trung bình lần lượt là \bar{X}_1, \bar{X}_2 . Hãy kiểm định, với mức ý nghĩa α , giả thuyết trung bình hai tổng thể bằng nhau có đáng tin cậy không.

Các bước giải

- Đặt giả thuyết kiểm định

Gọi $\mu_D = \mu_1 - \mu_2$ là chênh lệch giữa trung bình hai tổng thể I và II.

Giả thuyết không H_0 : Trung bình hai tổng thể không có sự khác biệt.



$$\mu_D = 0$$

Giả thuyết đối H_1 : Tuỳ theo yêu tố cần kiểm định mà có thể đặt H_1 là

- $\mu_D \neq 0$: Trung bình hai tổng thể có sự khác biệt.
- $\mu_D < 0$: Trung bình tổng thể I nhỏ hơn trung bình tổng thể II.
- $\mu_D > 0$: Trung bình tổng thể I lớn hơn trung bình tổng thể II.

• Tìm miền bác bỏ RR

Miền bác bỏ H_0 cho bài toán kiểm định trung bình hai mẫu độc lập phụ thuộc vào giả thuyết đối H_1 và dữ kiện đã biết. Cụ thể:

Dạng		H_1	Tiêu chuẩn kiểm định	
X_1, X_2 phân phối chuẩn	Biết σ_1^2, σ_2^2	$\mu_1 \neq \mu_2$	$(-\infty; -t_{\alpha/2}(n-1)) \cup (t_{\alpha/2}(n-1); +\infty)$	
		$\mu_1 < \mu_2$	$(-\infty; -z_\alpha)$	
		$\mu_1 > \mu_2$	$(z_\alpha; +\infty)$	
	Chưa biết σ_1^2, σ_2^2	$\frac{s_1}{s_2} \in [0.5; 2]$ $\mu_1 \neq \mu_2$	$(-\infty; -t_{\alpha/2}(n_1 + n_2 - 2)) \cup (t_{\alpha/2}(n_1 + n_2 - 2); +\infty)$	
		$\mu_1 < \mu_2$	$(-\infty; -t_\alpha(n_1 + n_2 - 2))$	
		$\mu_1 > \mu_2$	$(t_\alpha(n_1 + n_2 - 2); +\infty)$	
		$\frac{s_1}{s_2} \notin [0.5; 2]$ $\mu_1 \neq \mu_2$	$(-\infty; -t_{\alpha/2}(v)) \cup (t_{\alpha/2}(v); +\infty)$	
		$\mu_1 < \mu_2$	$(-\infty; -t_\alpha(v))$	
		$\mu_1 > \mu_2$	$(t_\alpha(v); +\infty)$	
X_1, X_2 phân phối tuỳ ý Kích thước hai mẫu lớn ($n_1, n_2 \geq 30$)		$\mu_1 \neq \mu_2$	$(-\infty; -t_{\alpha/2}(n-1)) \cup (t_{\alpha/2}(n-1); +\infty)$	
		$\mu_1 < \mu_2$	$(-\infty; -z_\alpha)$	
		$\mu_1 > \mu_2$	$(z_\alpha; +\infty)$	

Trong đó, bậc tự do v được tính:

$$v = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2} \right)^2}{\frac{\left(\frac{s_1^2}{n_1} \right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2} \right)^2}{n_2 - 1}}$$

• Tính giá trị kiểm định thống kê

Hàm kiểm định thống kê cho bài toán kiểm định trung bình hai mẫu rơi vào nhiều trường hợp, tùy theo dữ kiện đã biết, cụ thể:

Dạng		Tiêu chuẩn kiểm định	
X_1, X_2 phân phối chuẩn		Biết σ_1^2, σ_2^2	$Z_{qs} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$
		Chưa biết σ_1^2, σ_2^2	$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$ $T_{qs} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_p^2}{n_1} + \frac{s_p^2}{n_2}}}$ $T_{qs} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$
X_1, X_2 phân phối tuỳ ý Kích thước hai mẫu lớn ($n_1, n_2 \geq 30$)		Biết σ_1^2, σ_2^2	$Z_{qs} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$
		Chưa biết σ_1^2, σ_2^2	$Z_{qs} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$

- Kết luận

- Nếu Z_{qs} (hoặc T_{qs}) $\in RR$: bác bỏ giả thuyết H_0 , thừa nhận giả thuyết H_1 . Tức kết luận với mức ý nghĩa α , giả thuyết trung bình hai tổng thể bằng nhau là không đáng tin cậy.
- Nếu Z_{qs} (hoặc T_{qs}) $\notin RR$: chưa đủ dữ liệu khẳng định H_0 sai. Tức với mức ý nghĩa α , chưa thể nói giả thuyết trung bình hai tổng thể bằng nhau là không đáng tin cậy.

2.4 Phân tích phương sai

2.4.1 Khái niệm

Phân tích phương sai (Analysis of Variance – ANOVA) là một mô hình dùng để xem xét sự biến động của một biến ngẫu nhiên định lượng X chịu tác động trực tiếp của một hay nhiều yếu tố nguyên nhân (định tính). Mục đích chính của ANOVA là để xác định xem những biến động trong dữ liệu có phản ánh sự khác biệt thực sự giữa các nhóm hay không, hay chúng chỉ là kết quả của sự ngẫu nhiên.



2.4.2 Phân loại

Có hai loại phân tích phương sai thông dụng nhất:

- *Phân tích phương sai một yếu tố (one-way ANOVA)*: trong một mẫu chỉ xem xét một yếu tố hoặc một biến độc lập.

Ví dụ: Phân tích xem kết quả bài kiểm tra có khác nhau hay không dựa trên mức độ lo lắng giữa các học sinh (chia học sinh thành ba nhóm độc lập: học sinh có mức độ căng thẳng thấp, trung bình và cao). Trong đó, mức độ lo lắng là một yếu tố độc lập được dùng cho mô hình phân tích.

- *Phân tích phương sai hai yếu tố (two-way ANOVA)*: là mở rộng của phân tích phương sai một yếu tố. Với two-way ANOVA, sẽ có hai yếu tố độc lập được dùng để phân tích.

Ví dụ: Phân tích xem kết quả bài kiểm tra có khác nhau hay không dựa vào giới tính và mức độ lo lắng giữa các học sinh. Trong đó, giới tính và mức độ lo lắng là hai yếu tố độc lập được dùng cho mô hình phân tích.

Trong khuôn khổ bài tập lớn này, nhóm áp dụng mô hình ANOVA một yếu tố với bộ dữ liệu được cho.

2.4.3 Phân tích phương sai một yếu tố

Trong mô hình phân tích phương sai 1 yếu tố, chúng ta kiểm định so sánh trung bình của biến ngẫu nhiên X ở những tổng thể (còn gọi là nhóm) khác nhau dựa vào các mẫu quan sát lấy từ những tổng thể này. Các tổng thể được phân biệt bởi các mức độ khác nhau của yếu tố đang xem xét.

2.4.3.1 giả thuyết bài toán

Để áp dụng được mô hình phân tích phương sai một yếu tố cho bài toán gồm k tổng thể, các tổng thể cần thoả các giả thuyết:

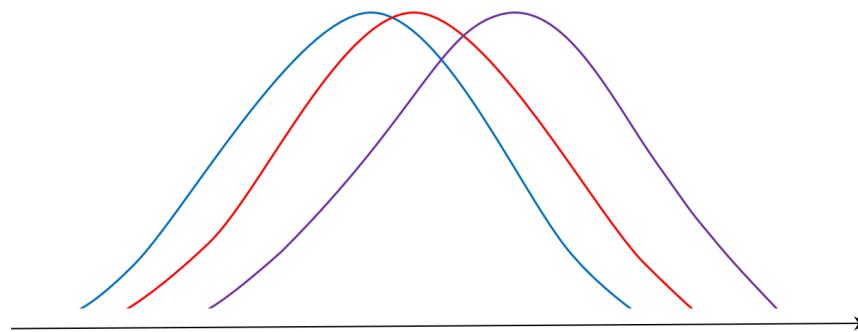
- Các tổng thể có phân phối chuẩn:

$$X_i \sim N(\mu_i; \sigma_i^2), \quad i = \overline{1..k}$$

- Phương sai các tổng thể bằng nhau:

$$\sigma_i^2 = \sigma_j^2, \quad \forall i, j = \overline{1..k}$$

- Các mẫu quan sát (từ các tổng thể) được lấy độc lập.



Hình 2.8: Các mẫu quan sát lấy ra từ các tổng thể tuân theo quy luật phân phối chuẩn với phương sai gần bằng nhau

Nếu các mẫu được chọn thoả mãn giả thuyết, ta có thể thống kê các giá trị trung bình cũng như kích thước của từng mẫu:

	Nhóm 1	Nhóm 2	...	Nhóm k
Các mẫu quan sát	x_{11}	x_{12}		x_{1k}
	x_{21}	x_{22}		x_{2k}

	$x_{n_1 1}$	$x_{n_2 2}$		$x_{n_k k}$
Kích thước từng mẫu	n_1	n_2	...	n_k
Trung bình từng mẫu	\bar{x}_1	\bar{x}_2	...	\bar{x}_k
Kích thước mẫu gộp	$N = n_1 + n_2 + \dots + n_k$			
Trung bình mẫu gộp	$\bar{x} = \frac{1}{N} \sum_{j=1}^k \sum_{i=1}^{n_j} x_{ij} = \frac{n_1 \bar{x}_1 + n_2 \bar{x}_2 + \dots + n_k \bar{x}_k}{N}$			

2.4.3.2 Các đại lượng đặc trưng

- **Sum of Squares Between (SSB):** Đo lường tổng phương sai giữa các nhóm, thể hiện sự khác biệt giữa trung bình của các nhóm dữ liệu.
- **Sum of Squares Within (SSW):** Đo lường tổng phương sai trong từng nhóm, thể hiện sự biến đổi ngẫu nhiên bên trong các nhóm.
- **Sum of Squares Total (SST):** Đo lường tổng phương sai trong tập dữ liệu, thể hiện toàn bộ sự biến đổi của tập dữ liệu.

Đại lượng	Công thức
SSB	$\sum_{j=1}^k n_j (\bar{x}_j - \bar{x})^2$
SSW	$\sum_{j=1}^k \sum_{i=1}^{n_j} (x_{ij} - \bar{x}_j)^2$
SST	$\sum_{j=1}^k \sum_{i=1}^{n_j} (x_{ij} - \bar{x})^2 = SSB + SSW$

Nhận xét:

- **SSB**: Phần biến thiên của giá trị X do các mức độ của yếu tố đang xem xét tạo ra.
- **SSW**: Phần biến thiên của giá trị X do các yếu tố nào đó không được đề cập đến tạo ra.
- **SST**: Tổng các biến thiên của X do tất cả các yếu tố tạo ra.

Từ đó, ta có **hệ số xác định**:

$$R^2 = \frac{SSB}{SST} \times 100\%$$

Hệ số xác định R^2 của mô hình Phân tích phương sai được sử dụng để đo mức độ ảnh hưởng của yếu tố được xem xét trong mô hình đối với sự biến động của các giá trị của biến ngẫu nhiên X quanh giá trị trung bình của nó. Giá trị R^2 càng lớn chứng tỏ mô hình càng thích hợp.

- **Mean Squares Between (MSB)**: Do lường sự biến đổi trung bình giữa các nhóm mẫu. MSB được tính bằng cách chia tổng biến đổi giữa các nhóm (SSB) cho bậc tự do của biến đổi giữa các nhóm (dfB).
- **Mean Squares Within (MSW)**: Do lường sự biến đổi trung bình bên trong mỗi nhóm mẫu. MSW được tính bằng cách chia tổng biến đổi trong từng nhóm cho bậc tự do của biến đổi trong từng nhóm (dfW).

Đại lượng	Bậc tự do df	Công thức
MSB	$k - 1$	$\frac{SSB}{k - 1}$
MSW	$N - k$	$\frac{SSW}{N - k}$

2.4.3.3 Các bước thực hiện mô hình phân tích phương sai một yếu tố

- Đặt giả thuyết kiểm định

Gọi $\mu_1, \mu_2, \dots, \mu_k$ lần lượt là trung bình của tổng thể 1, 4, ..., k .

Giả thuyết không H_0 :

$$\mu_1 = \mu_2 = \dots = \mu_k$$

Giả thuyết đối H_1 :

$$\exists \mu_i \neq \mu_j, \quad \text{với } i \neq j$$

- Tính giá trị kiểm định thống kê

Trong phân tích phương sai một yếu tố, giá trị kiểm định thống kê chính là giá trị F (F -statistic). Giá trị F được sử dụng để kiểm tra xem có sự khác biệt đáng kể giữa các nhóm trong biến ngẫu nhiên X đang nghiên cứu hay không.

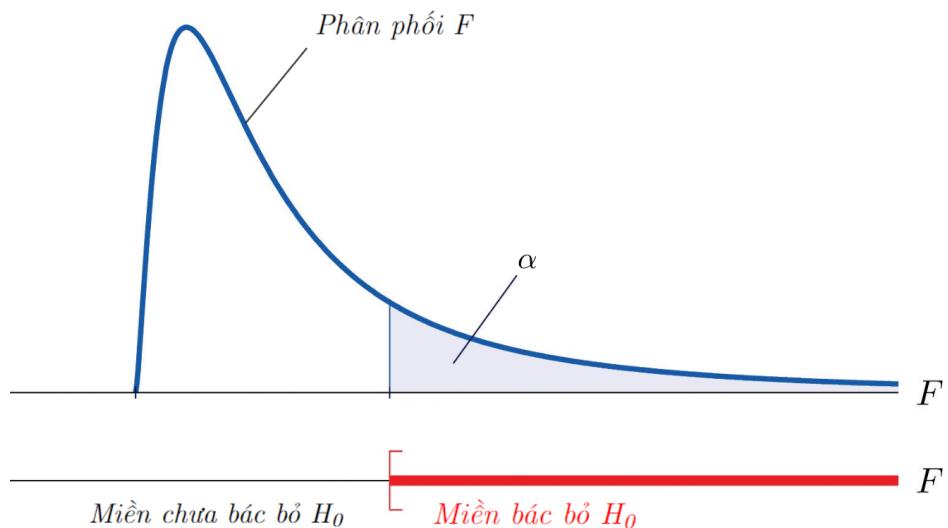
Công thức tính giá trị kiểm định thống kê F :

$$F = \frac{\text{MSB}}{\text{MSW}}$$

- Tìm miền bác bỏ

Miền bác bỏ trong phân tích phương sai một yếu tố nằm ở bên phải của phân phối F , được xác định dựa trên mức ý nghĩa thống kê α cùng với bậc tự do của biến đổi giữa các nhóm (dfB) và biến đổi bên trong từng nhóm (dfW). Tức:

$$\text{RR} = \left(f_{\alpha} (\text{dfB}; \text{dfW}) ; +\infty \right) = \left(f_{\alpha} (k-1; N-k) ; +\infty \right)$$



Hình 2.9: Phân phối F và miền bác bỏ của phân tích phương sai một yếu tố

		Degrees of freedom for the numerator (v_1)																			
		1	2	3	4	5	6	7	8	9	10	12	15	20	24	30	40	60	120	∞	
v_2	Degrees of freedom for the denominator (v_3)	1	4052	4999.5	5403	5625	5764	5859	5928	5982	6022	6056	6106	6157	6209	6235	6261	6287	6313	6339	6366
		2	98.50	99.00	99.17	99.25	99.30	99.33	99.36	99.37	99.39	99.40	99.42	99.43	99.45	99.46	99.47	99.47	99.48	99.49	99.50
3	34.12	30.82	29.46	28.71	28.24	27.91	27.67	27.49	27.35	27.23	27.05	26.87	26.69	26.00	26.50	26.41	26.32	26.22	26.13		
4	21.20	18.00	16.69	15.98	15.52	15.21	14.98	14.80	14.66	14.55	14.37	14.20	14.02	13.93	13.84	13.75	13.65	13.56	13.46		
5	16.26	13.27	12.06	11.39	10.97	10.67	10.46	10.29	10.16	10.05	9.89	9.72	9.55	9.47	9.38	9.29	9.20	9.11	9.02		
6	13.75	10.92	9.78	9.15	8.75	8.47	8.26	8.10	7.98	7.87	7.72	7.56	7.40	7.31	7.23	7.14	7.06	6.97	6.88		
7	12.25	9.55	8.45	7.85	7.46	7.19	6.99	6.84	6.72	6.62	6.47	6.31	6.16	6.07	5.99	5.91	5.82	5.74	5.65		
8	11.26	8.65	7.59	7.01	6.63	6.37	6.18	6.03	5.91	5.81	5.67	5.52	5.36	5.28	5.20	5.12	5.03	4.95	4.46		
9	10.56	8.02	6.99	6.42	6.06	5.80	5.61	5.47	5.35	5.26	5.11	4.96	4.81	4.73	4.65	4.57	4.48	4.40	4.31		
10	10.04	7.56	6.55	5.99	5.64	5.39	5.20	5.06	4.94	4.85	4.71	4.56	4.41	4.33	4.25	4.17	4.08	4.00	3.91		
11	9.65	7.21	6.22	5.67	5.32	5.07	4.89	4.74	4.63	4.54	4.40	4.25	4.10	4.02	3.94	3.86	3.78	3.69	3.60		
12	9.33	6.93	5.95	5.41	5.06	4.82	4.64	4.50	4.39	4.30	4.16	4.01	3.86	3.78	3.70	3.62	3.54	3.45	3.36		
13	9.07	6.70	5.74	5.21	4.86	4.62	4.44	4.30	4.19	4.10	3.96	3.82	3.66	3.59	3.51	3.43	3.34	3.25	3.17		
14	8.86	6.51	5.56	5.04	4.69	4.46	4.28	4.14	4.03	3.94	3.80	3.66	3.51	3.43	3.35	3.27	3.18	3.09	3.00		
15	8.68	6.36	5.42	4.89	4.36	4.32	4.14	4.00	3.89	3.80	3.67	3.52	3.37	3.29	3.21	3.13	3.05	2.96	2.87		
16	8.53	6.23	5.29	4.77	4.44	4.20	4.03	3.89	3.78	3.69	3.55	3.41	3.26	3.18	3.10	3.02	2.93	2.84	2.75		
17	8.40	6.11	5.18	4.67	4.34	4.10	3.93	3.79	3.68	3.59	3.46	3.31	3.16	3.08	3.00	2.92	2.83	2.75	2.65		
18	8.29	6.01	5.09	4.58	4.25	4.01	3.84	3.71	3.60	3.51	3.37	3.23	3.08	3.00	2.92	2.84	2.75	2.66	2.57		
19	8.18	5.93	5.01	4.50	4.17	3.94	3.77	3.63	3.52	3.43	3.30	3.15	3.00	2.92	2.84	2.76	2.67	2.58	2.59		
20	8.10	5.85	4.94	4.43	4.10	3.87	3.70	3.56	3.46	3.37	3.23	3.09	2.94	2.86	2.78	2.69	2.61	2.52	2.42		
21	8.02	5.78	4.87	4.37	4.04	3.81	3.64	3.51	3.40	3.31	3.17	3.03	2.88	2.80	2.72	2.64	2.55	2.46	2.36		
22	7.95	5.72	4.82	4.31	3.99	3.76	3.59	3.45	3.35	3.26	3.12	2.98	2.83	2.75	2.67	2.58	2.50	2.40	2.31		
23	7.88	5.66	4.76	4.26	3.94	3.71	3.54	3.41	3.30	3.21	3.07	2.93	2.78	2.70	2.62	2.54	2.45	2.35	2.26		
24	7.82	5.61	4.72	4.22	3.90	3.67	3.50	3.36	3.26	3.17	3.03	2.89	2.74	2.66	2.58	2.49	2.40	2.31	2.21		
25	7.77	5.57	4.68	4.18	3.85	3.63	3.46	3.32	3.22	3.13	2.99	2.85	2.70	2.62	2.54	2.45	2.36	2.27	2.17		
26	7.72	5.53	4.64	4.14	3.82	3.59	3.42	3.29	3.18	3.09	2.96	2.81	2.66	2.58	2.50	2.42	2.33	2.23	2.13		
27	7.68	5.49	4.60	4.11	3.78	3.56	3.39	3.26	3.15	3.06	2.93	2.78	2.63	2.55	2.47	2.38	2.29	2.20	2.10		
28	7.64	5.45	4.57	4.07	3.75	3.53	3.36	3.23	3.12	3.03	2.90	2.75	2.60	2.52	2.44	2.35	2.26	2.17	2.06		
29	7.60	5.42	4.54	4.04	3.73	3.50	3.33	3.20	3.09	3.00	2.87	2.73	2.57	2.49	2.41	2.33	2.23	2.14	2.03		
30	7.56	5.39	4.51	4.02	3.70	3.47	3.30	3.17	3.07	2.98	2.84	2.70	2.55	2.47	2.39	2.30	2.21	2.11	2.01		
40	7.31	5.18	4.31	3.83	3.51	3.29	3.12	2.99	2.89	2.80	2.66	2.52	2.37	2.29	2.20	2.11	2.02	1.92	1.80		
60	7.08	4.98	4.13	3.65	3.34	3.12	2.95	2.82	2.72	2.63	2.50	2.35	2.20	2.12	2.03	1.94	1.84	1.73	1.60		
120	6.85	4.79	3.95	3.48	3.17	2.96	2.79	2.66	2.56	2.47	2.34	2.19	2.03	1.95	1.86	1.76	1.66	1.53	1.38		
∞	6.63	4.61	3.78	3.32	3.02	2.80	2.64	2.51	2.41	2.32	2.18	2.04	1.88	1.79	1.70	1.59	1.47	1.32	1.00		

Hình 2.10: Bảng tra hàm phân phối F với $\alpha = 0.01$

- Kết luận

Nếu giá trị thống kê kiểm định F thuộc vào miền bác bỏ RR, ta bác bỏ được giả thuyết H_0 , kết luận trung bình của các tổng thể là khác nhau (sự khác biệt có ý nghĩa thống kê). Ngược lại, ta chưa bác bỏ được H_0 , hay chưa có bằng chứng về sự khác biệt giữa trung bình của các tổng thể.

2.4.3.4 Phân tích sâu

Khi kết luận cho mô hình ANOVA, có hai trường hợp xảy ra:

- Chưa bác bỏ được giả thuyết H_0 , hay là chưa có bằng chứng về sự khác biệt của các trung bình.
- Bác bỏ H_0 , chấp nhận H_1 , nghĩa là trung bình của các nhóm không bằng nhau (hay là sự khác biệt có ý nghĩa thống kê).

Khi đó, ta có nhu cầu phân tích thêm sự khác biệt đó đến từ một (hay những) nhóm nào, nhóm nào đó có trung bình lớn hơn, bằng, hay nhỏ hơn so với những nhóm khác,...

Có nhiều phương pháp đưa đến kết quả mong muốn, ta còn gọi đó là các phương pháp so sánh bội (Multiple comparison methods). Trong đó có phương pháp dùng kiểm định LSD (Least Significant Difference Test).

Các bước tiến hành kiểm định LSD:

- Lần lượt kiểm định tất cả C_k^2 cặp trung bình μ_i và μ_j của hai nhóm khác nhau. Đặt giả thuyết:

- Giả thuyết không H_0 :



$$\mu_i = \mu_j$$

– Giả thuyết đối H_1 :

$$\mu_i \neq \mu_j$$

- Tính giá trị kiểm định thống kê

$$LSD_{i;j} = t_{\alpha/2}(N - k) \sqrt{\text{MSW} \left(\frac{1}{n_i} + \frac{1}{n_j} \right)}$$

- Kết luận

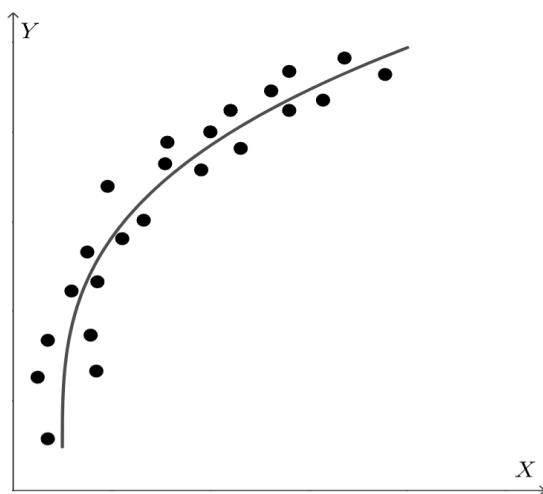
Nếu $|\bar{X}_i - \bar{X}_j| > LSD_{i;j}$, ta bác bỏ được giả thuyết H_0 , kết luận trung bình của hai nhóm i và j là khác nhau (sự khác biệt có ý nghĩa thống kê). Ngược lại, ta chưa bác bỏ được H_0 , hay chưa có bằng chứng về sự khác biệt giữa trung bình của hai nhóm.

2.5 Hồi quy tuyến tính

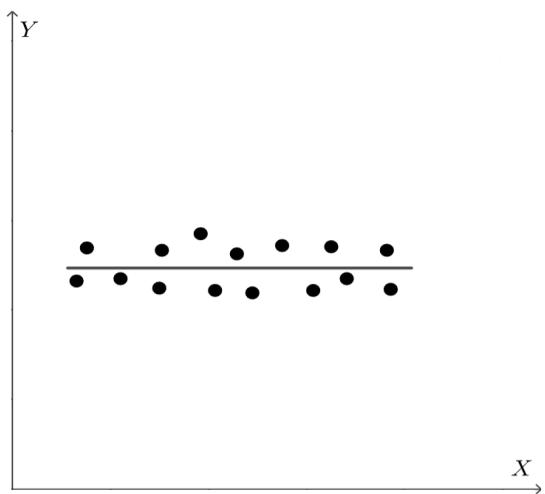
Hồi quy tuyến tính (Liner Regression) là một mô hình phân tích quan hệ giữa biến phụ thuộc Y với một hay nhiều biến độc lập X bằng một hàm tuyến tính bậc nhất. Đây là một kỹ thuật phân tích dữ liệu để dự đoán giá trị của dữ liệu không xác định. Các tham số của mô hình được tính toán từ những dữ liệu thực tế đã có. Khi tìm được hàm số tuyến tính xấp xỉ cho tập dữ liệu cho trước, ta có thể đưa ra những dự báo cho mô hình.

Ví dụ: Giả sử ta có tập dữ liệu về điểm thi của 100 sinh viên cùng các biến độc lập của sinh viên đó như thời gian tự học, số ngày đến lớp,... Kỹ thuật hồi quy tuyến tính phân tích bộ dữ liệu này và tìm ra một hàm tuyến tính cho điểm thi phụ thuộc vào các yếu tố độc lập. Khi đó, ta có thể đưa ra một ước lượng về điểm số của một sinh viên bất kì khi biến các yếu tố độc lập của sinh viên đó.

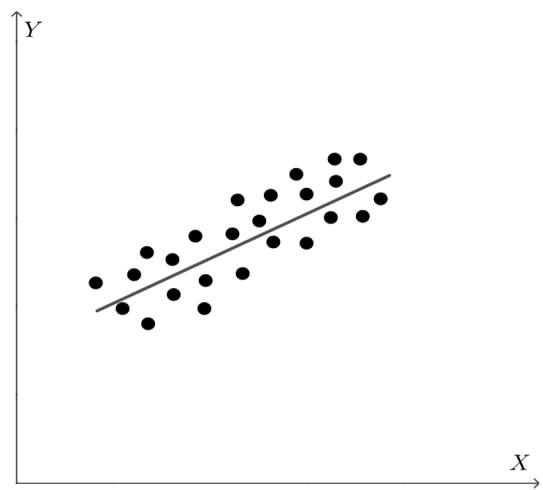
Mô hình hồi quy tuyến tính được sử dụng rộng rãi trong thực tế do tính hiệu quả, đơn giản, dễ thực hiện. Mô hình càng cho kết quả tốt khi tập dữ liệu ban đầu có xu hướng phân bố dọc theo một hướng thẳng. Trong trường hợp phân bố là ngẫu nhiên không theo quy luật, hoặc quy luật không phải tuyến tính (phi tuyến), mô hình này tỏ ra không mấy hiệu quả.



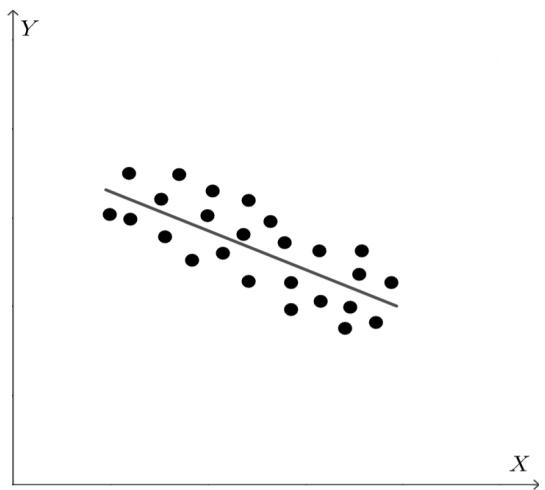
Hình 2.11: Liên hệ phi tuyến



Hình 2.12: Không có liên hệ



Hình 2.13: Liên hệ tuyến tính thuận



Hình 2.14: Liên hệ tuyến tính nghịch

Các hình trên biểu diễn mối liên hệ giữa hai biến số. Mỗi chấm là một sự kết hợp giữa X và Y cho ta một cặp giá trị cụ thể. Các đường liền nét trong hình là đường lý thuyết cho thấy xu hướng của tập dữ liệu.

2.5.1 Hồi quy tuyến tính đơn

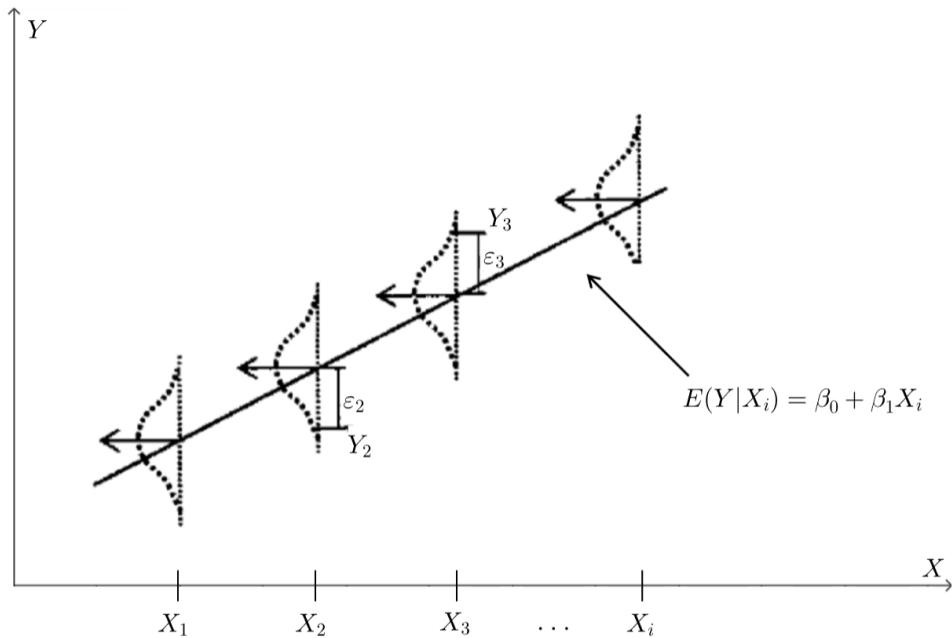
2.5.1.1 Định nghĩa

Hàm hồi quy của Y theo X chính là kì vọng có điều kiện của Y đối với X , tức là $E(Y|X)$.

Giả định của mô hình hồi quy tuyến tính đơn: Mô hình có các tham số β_0 , β_1 và σ^2 sao cho với mỗi giá trị x của biến độc lập, biến Y phụ thuộc vào x theo phương trình:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Trong đó, ε là sai số ngẫu nhiên có phân phối chuẩn $N(0; \sigma^2)$.



Hình 2.15: Đường thẳng hồi quy nối các giá trị trung bình của Y tại các giá trị khác nhau của biến độc lập X_i

Từ giả định của sai số ngẫu nhiên, suy ra $Y|X_i$ cũng tuân theo phân phối chuẩn với phương sai bằng với phương sai của ε :

$$Y|X_i \sim N(\beta_0 + \beta_1 X_i; \sigma^2)$$



2.5.1.2 Một số đặc trưng

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i; \quad \bar{x^2} = \frac{1}{n} \sum_{i=1}^n x_i^2$$

$$\widehat{s_x}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \equiv \bar{x^2} - \bar{x}^2; \quad s_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i; \quad \bar{y^2} = \frac{1}{n} \sum_{i=1}^n y_i^2$$

$$\widehat{s_y}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 \equiv \bar{y^2} - \bar{y}^2; \quad s_y^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$$

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

$$S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2$$

$$S_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \times \sum_{i=1}^n y_i$$

$$S_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n y_i^2 - \frac{1}{n} \left(\sum_{i=1}^n y_i \right)^2$$

2.5.1.3 Hiệp phương sai (Covariance) – Hệ số tương quan (Correlation coefficient)

Hiệp phương sai là đại lượng đo sự biến thiên cùng nhau của hai biến ngẫu nhiên. Hiệp phương sai giữa hai biến ngẫu nhiên X và Y được tính:

$$\text{cov}(X, Y) = E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y)$$

Với

$$E(XY) \stackrel{\text{RR}}{=} \sum_j \sum_i x_i y_j p_{ij}$$

Dễ thấy:

$$\text{cov}(X, X) = E[(X - E(X))^2] = E(X^2) - E^2(X) \equiv V(X)$$

$$\text{cov}(Y, Y) = E[(Y - E(Y))^2] = E(Y^2) - E^2(Y) \equiv V(Y)$$

Hệ số tương quan đo lường mức độ quan hệ tuyến tính giữa hai biến, không phân biệt biến này phụ thuộc vào biến kia. Hệ số tương quan giữa hai biến ngẫu nhiên X và Y được tính:

$$\rho_{XY} = \frac{\text{cov}(X, Y)}{\sqrt{V(X)} \sqrt{V(Y)}}$$



Ma trận tương quan (ma trận hiệp phương sai) của hai biến ngẫu nhiên X, Y là ma trận vuông $V(X, Y)_{2 \times 2}$, có dạng:

$$V(X, Y) = \begin{bmatrix} \text{cov}(X, X) & \text{cov}(X, Y) \\ \text{cov}(Y, X) & \text{cov}(Y, Y) \end{bmatrix}$$

Hiệp phương sai và hệ số tương quan dùng để đặc trưng cho mức độ chặt chẽ của mối liên hệ phụ thuộc giữa các biến ngẫu nhiên X và Y .

- Hệ số tương quan không có đơn vị đo và $\rho_{XY} \in [-1; 1]$.
- Nếu $\rho_{XY} = 0$, ta nói X và Y không tương quan, ngược lại, nếu $\rho_{XY} \neq 0$, ta nói X, Y có tương quan.
- Nếu X, Y độc lập thì $\text{cov}(X, Y) = \rho_{XY} = 0$.

Điều ngược lại không đúng, tức nếu $\text{cov}(X, Y) = 0$ thì hoặc X, Y độc lập, hoặc chúng phụ thuộc ở một dạng thức nào đó.

- Nếu $\rho_{XY} = \pm 1$ thì X, Y có tương quan tuyến tính (thuận/nghịch).

Khi $\rho_{XY} \approx \pm 1$ thì X, Y có tương quan "gần" tuyến tính.

Hiệp phương sai và hệ số tương quan của mẫu

$$\text{cov}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \frac{S_{xy}}{n} \equiv \bar{xy} - \bar{x}\bar{y}$$

$$r_{xy} = \frac{S_{xy}}{\sqrt{S_{xx}}\sqrt{S_{yy}}} \equiv \frac{\bar{xy} - \bar{x}\bar{y}}{\hat{s}_x\hat{s}_y}$$

r_{xy} là một ước lượng của hệ số tương quan ρ_{XY} . Mức độ mối quan hệ tuyến tính của X và Y phụ thuộc vào giá trị r_{xy} :

Nếu	Mức độ mối quan hệ tuyến tính của X và Y
$ r_{xy} \leq 0.3$	Không có hoặc rất yếu
$0.3 < r_{xy} \leq 0.5$	Rất yếu
$0.5 < r_{xy} \leq 0.8$	Trung bình
$0.8 < r_{xy} $	Mạnh

2.5.1.4 Đường hồi quy tuyến tính mẫu

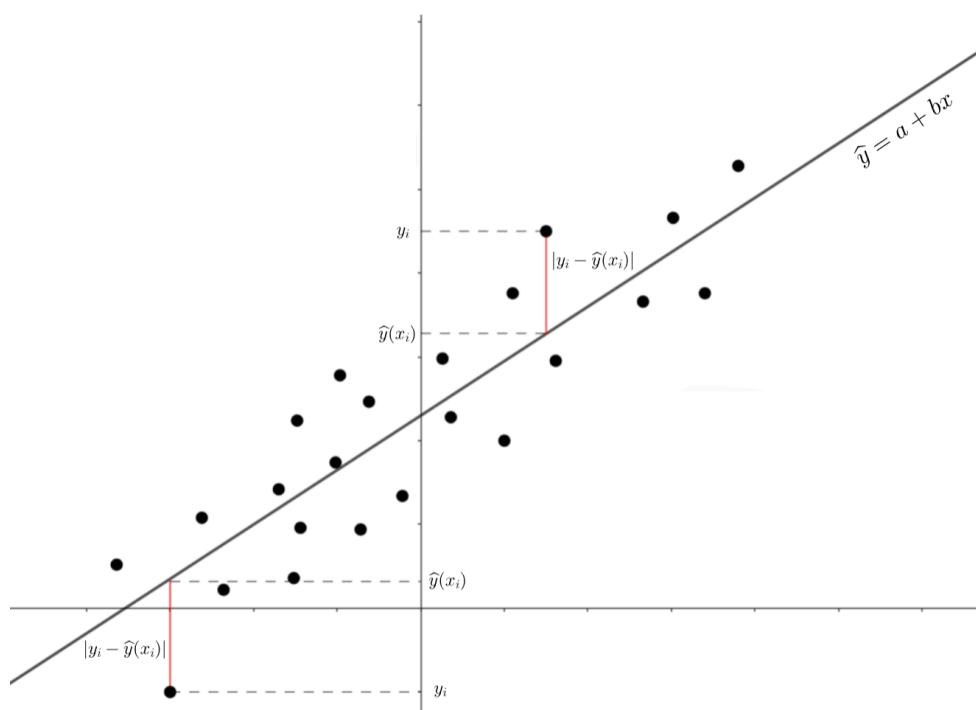
Giả sử ta có một mẫu cụ thể $\{(x_i; y_i)\}_{i=1..n}$. Theo phương pháp tổng bình phương nhỏ nhất (OLS), hàm

$$\hat{y} = a + bx$$

là đường hồi quy tuyến tính mẫu của Y theo X nếu hàm

$$Q = \sum_{i=1}^n [y_i - \hat{y}(x_i)]^2$$

đạt giá trị nhỏ nhất. Trong đó y_i là giá trị thực trong mẫu, còn $\hat{y}(x_i)$ là giá trị của đường hồi quy mẫu ứng với khi $x = x_i$.



Hình 2.16: $\hat{y} = a + bx$ là đường hồi quy tuyến tính của mẫu khi tổng bình phương các giá trị $|y_i - \hat{y}(x_i)|$ (độ dài các đoạn màu đỏ) đạt giá trị nhỏ nhất

Dựa vào toán học, người ta tìm được các hệ số a và b của đường hồi quy tuyến tính mẫu:

$$b = \frac{S_{xy}}{S_{xx}} = \frac{\bar{xy} - \bar{x}\bar{y}}{\hat{s}_x^2}$$

$$a = \bar{y} - b\bar{x}$$

Giá trị a, b lần lượt là một ước lượng cho các hệ số β_0, β_1 của đường hồi quy tổng thể.

Phương trình hồi quy tìm được có thể dùng để nội suy giá trị $E(Y|X = x_0)$:

$$\hat{y}_0 = a + bx_0$$

2.5.1.5 Các thông số khác – Hệ số xác định R^2

- **Sum of Squares in Regression (SSR):** Do lường sai số do khác biệt giữa đường hồi quy mẫu và trung bình của Y .
- **Sum of Squares for Error (SSE):** Do lường tổng bình phương sai số ước lượng do sự chênh lệch giữa từng giá trị quan sát với giá trị ước lượng.
- **Sum of Squares Total (SST):** Do lường tổng biến động các giá trị quan sát y_i xung quanh giá trị trung bình của mẫu.

Dại lượng	Công thức
SSR	$\sum_{i=1}^n (\hat{y}_i - \bar{y})^2$
SSE	$\sum_{i=1}^n (y_i - \hat{y}_i)^2$
SST	$\sum_{i=1}^n (y_i - \bar{y})^2 = \text{SSR} + \text{SSE}$

Nhận xét:

- **SSR:** Sự khác biệt này được giải thích bởi sự biến động của X . SSR đo sự phân tán của dữ liệu do mô hình hồi quy gây ra.
- **SSE:** Sai số do những yếu tố khác ngoài X hoặc do lấy mẫu ngẫu nhiên.

Từ đó, ta có **hệ số xác định**:

$$R^2 = \frac{\text{SSR}}{\text{SST}} \times 100\%$$

Hệ số R^2 thể hiện trong 100% sự biến động của Y so với trung bình của nó thì có bao nhiêu phần trăm là do biến X gây ra.

Trong mô hình hồi quy tuyến tính đơn

$$R^2 = r_{xy}^2 \quad (\text{bình phương hệ số tương quan mẫu}).$$

2.5.2 Hồi quy tuyến tính bội

2.5.2.1 Định nghĩa

Hồi quy tuyến tính bội là một phần mở rộng của hồi quy tuyến tính đơn. Nó được sử dụng để dự đoán giá trị của một biến phụ thuộc Y dựa trên giá trị của hai hay nhiều biến độc lập X_j khác. Mô hình hồi quy tuyến tính bội cũng cho phép ta xác định sự phù hợp tổng thể của mô hình và đóng góp tương đối của từng yếu tố độc lập.

Giả sử biến Y phụ thuộc vào k biến độc lập X_1, X_2, \dots, X_k . Mô hình hồi quy tuyến tính bội có dạng:



$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon$$

Trong đó:

- β_0 : hệ số chẵn (hệ số tự do), cho biết trung bình của Y khi X_1, X_2, \dots, X_k bằng 0.
- $\beta_j, j = \overline{1..k}$: các hệ số hồi quy riêng, thể hiện độ biến thiên của Y khi X_i thay đổi.
- ε : sai số ngẫu nhiên.

2.5.2.2 Giả thiết cho mô hình hồi quy tuyến tính bội

Để các ước lượng của mô hình hồi quy tuyến tính có ý nghĩa, mô hình cần phải thỏa mãn những giả thiết:

- Tồn tại mối quan hệ tuyến tính giữa Y và các X_i .
- Biến độc lập X không tồn tại mối quan hệ tương quan với sai số ε .
- Sai số ε có phân phối chuẩn $N(0, \sigma^2)$ với phương sai σ^2 không đổi đối với các lần quan sát khác nhau.
- Các biến X_i độc lập nhau.

2.5.2.3 Phương trình hồi quy tuyến tính mẫu

Gọi các hệ số a, b_1, b_2, \dots, b_k là ước lượng cho $\beta_0, \beta_1, \beta_2, \dots, \beta_k$ có được từ mẫu cụ thể $\{(x_{ji}; y_i)\}_{i=\overline{1..n}; j=\overline{1..k}}$. Khi đó,

$$\hat{y} = a + b_1 x_1 + b_2 x_2 + \dots + b_k x_k$$

là phương trình hồi quy tuyến tính mẫu của Y theo các X_i .

3 Tiềm xử lý số liệu

3.1 Đọc dữ liệu

Dùng lệnh `read.csv()` đọc dữ liệu trong tập tin `All_GPUs.csv`, lưu vào biến `df`, đồng thời dùng `head()` để hiển thị một số dòng đầu tiên cũng như lệnh `View()` để xem tổng quan bảng dữ liệu đã nhập:

```
1 df <- read.csv("All_GPUs.csv")
2 head(df)
3 View(df)
```

```
> head(df)
# Architecture Best_Resolution Boost_Clock Core_Speed DVI_Connection Dedicated Direct_X DisplayPort_Connection HDMI_Connection Integrated L2_Cache Manufacturer
1 Tesla G92b 738 MHz 2 Yes DX 10.0 NA 0 No 0KB Nvidia
2 R600 XT 1366 x 768 \n- 2 Yes DX 10 NA 0 No 0KB AMD
3 R600 PRO 1366 x 768 \n- 2 Yes DX 10 NA 0 No 0KB AMD
4 RV630 1024 x 768 \n- 2 Yes DX 10 NA 0 No 0KB AMD
5 RV630 1024 x 768 \n- 2 Yes DX 10 NA 0 No 0KB AMD
6 RV630 1024 x 768 \n- 2 Yes DX 10 NA 0 No 0KB AMD
Max_Power Memory Memory_Bandwidth Memory_Bus Memory_Speed Memory_Type Name Notebook_GPU Open_GL PSU Pixel_Rate
1 141 Watts 1024 MB 64GB/sec 256 Bit 1000 MHz GDDR3 GeForce GTS 150 No 3.3 450 Watt & 38 Amps 12 GPixel/s
2 215 Watts 512 MB 106GB/sec 512 Bit 828 MHz GDDR3 Radeon HD 2900 XT 512MB No 3.1 550 Watt & 35 Amps 12 GPixel/s
3 200 Watts 512 MB 51.2GB/sec 256 Bit 800 MHz GDDR3 Radeon HD 2900 Pro No 3.1 550 Watt & 35 Amps 10 GPixel/s
4 256 MB 36.8GB/sec 128 Bit 1150 MHz GDDR4 Radeon HD 2600 XT Diamond Edition No 3.3 400 Watt & 25 Amps 3 GPixel/s
5 45 Watts 256 MB 22.4GB/sec 128 Bit 700 MHz GDDR3 Radeon HD 2600 XT No 3.1 400 Watt & 25 Amps 3 GPixel/s
6 50 Watts 256 MB 35.2GB/sec 128 Bit 1100 MHz GDDR4 Radeon HD 2600 XT 256MB GDDR4 No 3.3 400 Watt & 26 Amps 3 GPixel/s
Power_Connector Process ROPs Release_Date Resolution_WKH SLI_Crossfire Shader TMUS Texture_Rate VGA_Connection
1 None 55nm 16 \n01-Mar-2009 2560x1600 Yes 4 64 47 GTexel/s 0
2 None 80nm 16 \n14-May-2007 2560x1600 Yes 4 16 12 GTexel/s 0
3 None 80nm 16 \n07-Dec-2007 2560x1600 Yes 4 16 10 GTexel/s 0
4 None 65nm 4 \n01-Jul-2007 2560x1600 Yes 4 8 7 GTexel/s 0
5 None 65nm 4 \n28-Jun-2007 2560x1600 Yes 4 8 6 GTexel/s 0
6 None 65nm 4 \n26-Jun-2007 2560x1600 Yes 4 8 6 GTexel/s 0
```

Hình 3.1: Một vài dòng đầu tiên của dữ liệu

Architecture	Best_Resolution	Boost_Clock	Core_Speed	DVI_Connection	Dedicated	Direct_X	DisplayPort_Connection	HDMI_Connection	Integrated	L2_Cache
1 Tesla G92b			738 MHz	2	Yes	DX 10.0	NA	0	No	0KB
2 R600 XT	1366 x 768		\n-	2	Yes	DX 10	NA	0	No	0KB
3 R600 PRO	1366 x 768		\n-	2	Yes	DX 10	NA	0	No	0KB
4 RV630	1024 x 768		\n-	2	Yes	DX 10	NA	0	No	0KB
5 RV630	1024 x 768		\n-	2	Yes	DX 10	NA	0	No	0KB
6 RV630	1024 x 768		\n-	2	Yes	DX 10	NA	0	No	0KB
7 R700 RV790 XT	1920 x 1080		870 MHz	1	Yes	DX 10.1	NA	1	No	0KB
8 R600 GT	1024 x 768		\n-	2	Yes	DX 10	NA	0	No	0KB
9 Pitcairn XT GL	1920 x 1080		\n-	0	Yes	DX 11.2	NA	0	No	0KB
10 RV100			\n-	NA	Yes	DX 7	NA	NA	No	0KB
11 NV28GL A2			\n-	2	Yes	DX 8.1	NA	0	No	0KB
12 Fermi GF110	1920 x 1080		650 MHz	2	Yes	DX 12.0	0	1	No	768KB
13 Kepler GK110			705 MHz	0	Yes	DX 12.0	NA	0	No	1536KB
14 Kepler GK110	2560 x 1600		706 MHz	0	Yes	DX 12.0	NA	0	No	1536KB
15 RV200			\n-	NA	Yes	DX 7	NA	NA	No	0KB
16 GCN 1.1 Oland XT + Kaveri	1600 x 900	1100 MHz	1050 MHz	1	Yes	DX 12.0	0	1	No	2304KB(x2)
17 Kepler GK104 x2			\n-	0	Yes	DX 11.1	NA	0	No	0KB
18 Kepler GK110			732 MHz	0	Yes	DX 12.0	NA	0	No	1536KB
19 Skylake GT3-E	1366 x 768	1000 MHz	300 MHz	NA	No	DX 12.1	NA	NA	Yes	512KB
20 R360	1024 x 768	\n-	\n-	1	Yes	DX 9	0	0	No	0KB
21 Fermi GF100			575 MHz	1	Yes	DX 12.0	NA	0	No	768KB
22 Fermi GF100			575 MHz	1	Yes	DX 12.0	NA	0	No	768KB
23 NV25 A2		\n-	\n-	2	Yes	DX 8.1	NA	0	No	0KB
24 NV28GL A2		\n-	\n-	2	Yes	DX 8.1	NA	0	No	0KB
25 Skylake GT3-E	1366 x 768	1000 MHz	300 MHz	NA	No	DX 12.1	NA	NA	Yes	512KB
26 NV28		\n-	\n-	2	Yes	DX 8.1	NA	0	No	0KB

Hình 3.2: Tổng quan dữ liệu đã nhập

3.2 Trích xuất các tiêu thức quan trọng của dữ liệu

Từ 34 tiêu thức có trong bảng, ta chọn ra 7 tiêu thức quan trọng của GPU để xử lý: **Name**, **Best_Resolution**, **Core_Speed**, **Memory**, **Memory_Bandwidth**, **Manufacturer**, **Release_Date**.

```
1 df <- df[, c("Name", "Best_Resolution", "Core_Speed", "Memory",
2           "Memory_Bandwidth", "Manufacturer", "Release_Date")]
3 View(df)
```

	Name	Best_Resolution	Core_Speed	Memory	Memory_Bandwidth	Manufacturer	Release_Date
1	GeForce GTS 150		738 MHz	1024 MB	64GB/sec	Nvidia	01-Mar-2009
2	Radeon HD 2900 XT 512MB	1366 x 768	-	512 MB	106GB/sec	AMD	14-May-2007
3	Radeon HD 2900 Pro	1366 x 768	-	512 MB	51.2GB/sec	AMD	07-Dec-2007
4	Radeon HD 2600 XT Diamond Edition	1024 x 768	-	256 MB	36.8GB/sec	AMD	01-Jul-2007
5	Radeon HD 2600 XT	1024 x 768	-	256 MB	22.4GB/sec	AMD	28-Jun-2007
6	Radeon HD 2600 XT 256MB GDDR4	1024 x 768	-	256 MB	35.2GB/sec	AMD	26-Jun-2007
7	Radeon HD 4890 Sapphire Vapor-X OC 2GB Edition	1920 x 1080	870 MHz	2048 MB	134.4GB/sec	AMD	13-Jul-2009
8	Radeon HD 2900 GT	1024 x 768	-	256 MB	51.2GB/sec	AMD	06-Nov-2007
9	FirePro D300	1920 x 1080	-	2048 MB	160GB/sec	AMD	18-Jan-2014
10	Radeon 7000 64mb		-	64 MB	2.9GB/sec	AMD	02-Jan-2001
11	Quadro4 980 XGL		-	128 MB	5.2GB/sec	Nvidia	01-Nov-2002
12	Tesla M2090	1920 x 1080	650 MHz	6144 MB	177.6GB/sec	Nvidia	25-Jul-2011
13	Tesla K20		705 MHz	5120 MB	168GB/sec	Nvidia	01-Nov-2012
14	Tesla K40c	2560 x 1600	706 MHz	12288 MB	288.4GB/sec	Nvidia	12-Nov-2013
15	All-in-Wonder Radeon 7500		-	64 MB	5.8GB/sec	AMD	22-Jan-2002
16	Radeon R7 250 v2 MSI OC 2GB + Radeon R7 7870K Dual	1600 x 900	1050 MHz	3072 MB	57.6GB/sec	AMD	28-May-2015
17	Tesla K10		-	8192 MB	320GB/sec	Nvidia	15-May-2012
18	Tesla K20X		732 MHz	6144 MB	249.6GB/sec	Nvidia	12-Nov-2012
19	Iris i3 6167U	1366 x 768	300 MHz		34.1GB/sec	Intel	01-Sep-2015
20	Radeon 9800 XT	1024 x 768	-	256 MB	23.4GB/sec	AMD	01-Oct-2003
21	Tesla C2070		575 MHz	6144 MB	144GB/sec	Nvidia	01-Sep-2010
22	Tesla C2075		575 MHz	6144 MB	144GB/sec	Nvidia	01-Jul-2011
23	Quadro4 750 XGL		-	128 MB	3.6GB/sec	Nvidia	19-Feb-2002
24	Quadro4 900 XGL		-	128 MB	5.2GB/sec	Nvidia	01-Nov-2002
25	Iris i5 6360U	1366 x 768	300 MHz		34.1GB/sec	Intel	01-Sep-2015
26	Quadro4 780 XGL		-	128 MB	4.4GB/sec	Nvidia	01-Nov-2002

Hình 3.3: Một vài phần tử đầu tiên của dữ liệu sau khi trích xuất

3.3 Xử lý định dạng dữ liệu

Sử dụng thư viện `tidyverse` có sẵn của R Studio để thực hiện việc tính toán sơ bộ, tách đơn vị của các tiêu thức.

```
1 # Khai báo thư viện tidyverse sử dụng cho việc định dạng dữ liệu
2 library(tidyverse)
```

Đối với tiêu thức **Best_Resolution**: Ta cần tính số điểm ảnh (pixels) mà GPU hoạt động tốt nhất bằng cách thực hiện phép nhân giữa số pixel theo nhiều ngang và chiều dọc. Đồng thời, đổi tên tiêu thức này thành **Number_of_Pixels**:

```
1 # Đổi tên tiêu thức Best_Resolution thành Number_of_Pixels
2 names(df)[names(df) == "Best_Resolution"] <- "Number_of_Pixels"
3 # Thực hiện phép nhân số pixel hai chiều ngang và dọc
4 df$Number_of_Pixels <- sapply(strsplit(df$Number_of_Pixels, " x "),
5                                 function(x) as.numeric(x[1]) * as.numeric(x[2]))
```

Đối với các tiêu thức **Core_Speed**, **Memory** và **Memory_Bandwidth**: Ta cần tách mỗi tiêu thức thành hai cột, phần **giá trị** và phần **đơn vị**:

```
1 # Do cột Core_Speed có các giá trị khuyết biếu diển bởi "-" nên cần loại
2 # bỏ trước khi xử lý
3 df$Core_Speed[grep("-", df$Core_Speed)] <- ""
4 # Tách cột Core_Speed thành hai cột Core_Speed_Value và Core_Speed_Unit
5 df <- separate(df, col = Core_Speed, into = c("Core_Speed_Value", "Core_
6 Speed_Unit"), sep = " ", fill = "right")
7 # Chuyển cột Core_Speed_Value về định dạng số
8 df$Core_Speed_Value <- as.numeric(df$Core_Speed_Value)
9
10 # Thực hiện tương tự với các tiêu thức còn lại
11 df <- separate(df, col = Memory, into = c("Memory_Value", "Memory_Unit"))
12 , sep = " ", fill = "right", extra = "drop")
13 df$Memory_Value <- as.numeric(df$Memory_Value)
14 table(df$Memory_Unit)
15
16 df <- separate(df, col = Memory_Bandwidth, into = c("Memory_Bandwidth_
17 Value", "Memory_Bandwidth_Unit"), sep = "(?<=\d)(?=([A-Za-z])",
18 fill = "right")
19 df$Memory_Bandwidth_Value <- as.numeric(df$Memory_Bandwidth_Value)
```

Kiểm tra đơn vị của các tiêu thức bằng lệnh `table()`:

```
1 table(df$Core_Speed_Unit)
2 table(df$Memory_Unit)
3 table(df$Memory_Bandwidth_Unit)

> table(df$core_Speed_Unit)
MHz
2470
> table(df$Memory_Unit)
MB
2986
> table(df$Memory_Bandwidth_Unit)
GB/sec MB/sec
3281 4
```

Hình 3.4: Kiểm tra đơn vị của các tiêu thức

Có thể thấy, hai tiêu thức **Core_Speed** và **Memory** chỉ có một loại đơn vị. Tuy nhiên, tiêu thức **Memory_Bandwidth** lại có hai loại đơn vị – [GB/sec] và [MB/sec]. Do đó, ta cần đưa tiêu thức này về cùng một đơn vị. Ở đây, chọn đưa về [GB/sec] ($1 \text{ [GB/sec]} = 1024 \text{ [MB/sec]}$).

```
1 # Chia giá trị của các phần tử có đơn vị [MB/sec] cho 1024 để đưa về đơn
2 # vị [GB/sec]
3 df$Memory_Bandwidth_Value <- ifelse(df$Memory_Bandwidth_Unit == "MB/sec"
4 , df$Memory_Bandwidth_Value / 1024, df$Memory_Bandwidth_Value)
5 # Thay đổi đơn vị của các phần tử đã chuyển giá trị thành [GB/sec]
6 df$Memory_Bandwidth_Unit <- ifelse(df$Memory_Bandwidth_Unit == "MB/sec",
7 "GB/sec", df$Memory_Bandwidth_Unit)
```



```
8 # Kiểm tra việc chuyển đổi
9 table(df$Memory_Bandwidth_Unit)

> table(df$Memory_Bandwidth_Unit)
GB/sec
3285
```

Hình 3.5: Đơn vị của tiêu thức *Memory_Bandwidth* sau khi chuyển đổi

Đối với tiêu thức **Release_Date**: Chuyển về định dạng ngày bằng lệnh `as.Date()`. Đồng thời, tạo thêm cột **Release_Value** theo công thức:

$$\text{Release_Value} = \text{năm} + \frac{\text{tháng}}{12}$$

```
1 # Chuyển về định dạng ngày tháng
2 df$Release_Date <- as.Date(sub("^\\s*\\n", "", df$Release_Date),
3                               format = "%d-%b-%Y")
4 # Tạo cột Release_Value theo công thức
5 df$Release_Value <- as.numeric(format(df$Release_Date, "%Y")) +
6                           as.numeric(format(df$Release_Date, "%m")) / 12
```

Dùng lệnh `str()` và `View()` để kiểm tra lại toàn bộ quá trình xử lý định dạng:

```
1 # Xem kiểu dữ liệu của các tiêu thức
2 str(df)
3 # Xem một vài phần tử đầu tiên của dữ liệu
4 View(df)
```

```
> str(df)
'data.frame': 3406 obs. of 11 variables:
 $ Name          : chr "GeForce GTS 150" "Radeon HD 2900 XT 512MB" "Radeon HD 2900 Pro" "Radeon HD 2600 XT Diamond Edition" ...
 $ Number_of_Pixels: num NA 1049088 1049088 786432 786432 ...
 $ Core_Speed_Value: num 738 NA NA NA NA NA 870 NA NA NA ...
 $ Core_Speed_Unit: chr "MHz" NA NA NA ...
 $ Memory_Value   : num 1024 512 512 256 256 ...
 $ Memory_Unit    : chr "MB" "MB" "MB" "MB" ...
 $ Memory_Bandwidth_Value: num 64 106 51.2 36.8 22.4 ...
 $ Memory_Bandwidth_Unit: chr "GB/sec" "GB/sec" "GB/sec" "GB/sec" ...
 $ Manufacturer   : chr "Nvidia" "AMD" "AMD" "AMD" ...
 $ Release_Date    : Date, format: "2009-03-01" "2007-05-14" "2007-12-07" "2007-07-01" ...
 $ Release_Value   : num 2009 2007 2008 2008 2008 ...
```

Hình 3.6: Kiểm tra kiểu dữ liệu của các tiêu thức

Name	Number_of_Pixels	Core_Speed_Value	Core_Speed_Unit	Memory_Value	Memory_Unit	Memory_Bandwidth_Value	Memory_Bandwidth_Unit	Manufacturer	Release_Date	Release_Value
1 GeForce GTS 150	NA	738	MHz	1024	MB	64.0	GB/sec	Nvidia	2009-03-01	2009.250
2 Radeon HD 2900 XT 512MB	1049088	NA	NA	512	MB	106.0	GB/sec	AMD	2007-05-14	2007.417
3 Radeon HD 2900 Pro	1049088	NA	NA	512	MB	51.2	GB/sec	AMD	2007-12-07	2008.000
4 Radeon HD 2600 XT Diamond Edition	786432	NA	NA	256	MB	36.8	GB/sec	AMD	2007-07-01	2007.583
5 Radeon HD 2600 XT	786432	NA	NA	256	MB	22.4	GB/sec	AMD	2007-06-28	2007.500
6 Radeon HD 2600 XT 256MB GDDR4	786432	NA	NA	256	MB	35.2	GB/sec	AMD	2007-06-26	2007.500
7 Radeon HD 4890 Sapphire Vapor-X OC 2GB Edition	2073600	870	MHz	2048	MB	134.4	GB/sec	AMD	2009-07-13	2009.583
8 Radeon HD 2900 GT	786432	NA	NA	256	MB	51.2	GB/sec	AMD	2007-11-06	2007.917
9 FirePro D300	2073600	NA	NA	2048	MB	160.0	GB/sec	AMD	2014-01-18	2014.083
10 Radeon 7000 64mb	NA	NA	NA	64	MB	2.9	GB/sec	AMD	2001-01-02	2001.083
11 Quadro4 960 XGL	NA	NA	NA	128	MB	5.2	GB/sec	Nvidia	2002-11-01	2002.917
12 Tesla M2090	2073600	650	MHz	6144	MB	177.6	GB/sec	Nvidia	2011-07-25	2011.583
13 Tesla K20	NA	705	MHz	5120	MB	168.0	GB/sec	Nvidia	2012-11-01	2012.917
14 Tesla K40c	4096000	706	MHz	12288	MB	288.4	GB/sec	Nvidia	2013-11-12	2013.917
15 All-in-Wonder Radeon 7500	NA	NA	NA	64	MB	5.8	GB/sec	AMD	2002-01-22	2002.083
16 Radeon R7 250 v2 MSI OC 2GB + Radeon R7 7870K Dual	1440000	1050	MHz	3072	MB	57.6	GB/sec	AMD	2015-05-28	2015.547
17 Tesla K10	NA	NA	NA	8192	MB	320.0	GB/sec	Nvidia	2012-05-15	2012.417
18 Tesla K20X	NA	732	MHz	6144	MB	249.6	GB/sec	Nvidia	2012-11-12	2012.917
19 Iris i3 6167U	1049088	300	MHz	NA	NA	34.1	GB/sec	Intel	2015-09-01	2015.570
20 Radeon 9800 XT	786432	NA	NA	256	MB	23.4	GB/sec	AMD	2003-10-01	2003.833
21 Tesla C2070	NA	575	MHz	6144	MB	144.0	GB/sec	Nvidia	2010-09-01	2010.750
22 Tesla C2075	NA	575	MHz	6144	MB	144.0	GB/sec	Nvidia	2011-07-01	2011.583
23 Quadro4 750 XGL	NA	NA	NA	128	MB	3.6	GB/sec	Nvidia	2002-02-19	2002.167
24 Quadro4 900 XGL	NA	NA	NA	128	MB	5.2	GB/sec	Nvidia	2002-11-01	2002.917
25 Iris i5 6360U	1049088	300	MHz	NA	NA	34.1	GB/sec	Intel	2015-09-01	2015.570
26 Quadro4 780 XGL	NA	NA	NA	128	MB	4.4	GB/sec	Nvidia	2002-11-01	2002.917

Hình 3.7: Một vài phần tử đầu tiên của dữ liệu

3.4 Xử lý dữ liệu khuyết

Với dữ liệu bảng, việc các trường thông tin bị khuyết (có giá trị NA) là thường xuyên xảy ra. Việc này đến từ quá trình thu thập dữ liệu. Người dùng có thể thực sự không có thông tin đó hoặc không muốn tiết lộ vì lý do cá nhân.

Có nhiều cách để xử lý dữ liệu khuyết dạng số:

- Xoá bỏ phần tử khuyết dữ liệu khỏi mẫu.
- Thay giá trị khuyết bằng giá trị trung bình các phần tử không khuyết của mẫu.
- Thay giá trị khuyết bằng giá trị trung vị các phần tử không khuyết của mẫu.

Trước tiên, ta cần kiểm tra số lượng và tỉ lệ khuyết dữ liệu của các mẫu tiêu thức bằng lệnh `apply()`:

```

1 # Đếm số lượng phần tử bị khuyết dữ liệu của các mẫu dữ liệu
2 apply(is.na(df[, c("Name", "Number_of_Pixels", "Core_Speed_Value",
3   "Memory_Value", "Memory_Bandwidth_Value", "Manufacturer",
4   "Release_Value")]), MARGIN = 2, FUN = sum)
5 # Tính tỉ lệ khuyết dữ liệu của các mẫu
6 apply(is.na(df[, c("Name", "Number_of_Pixels", "Core_Speed_Value",
7   "Memory_Value", "Memory_Bandwidth_Value", "Manufacturer",
8   "Release_Value")]), MARGIN = 2, FUN = mean)

```



```
> apply(is.na(df[, c("Name", "Number_of_Pixels", "Core_Speed_Value", "Memory_Value",
+                 "Memory_Bandwidth_Value", "Manufacturer", "Release_Value")]], MARGIN = 2, FUN = sum)
      Name       Number_of_Pixels     Core_Speed_Value     Memory_Value
      0                  642                  936                  420
Memory_Bandwidth_Value      Manufacturer      Release_Value
      121                      0                     30
> apply(is.na(df[, c("Name", "Number_of_Pixels", "Core_Speed_Value", "Memory_Value",
+                 "Memory_Bandwidth_Value", "Manufacturer", "Release_Value")]], MARGIN = 2, FUN = mean)
      Name       Number_of_Pixels     Core_Speed_Value     Memory_Value
      0.0000000000          0.188490898          0.274809160          0.123311803
Memory_Bandwidth_Value      Manufacturer      Release_Value
      0.035525543          0.000000000          0.008807986
```

Hình 3.8: Số lượng và tỉ lệ khuyết dữ liệu của mỗi mẫu

Nhận xét: Tỉ lệ khuyết dữ liệu (nếu có) của các mẫu thể hiện ở bảng dưới đây:

Mẫu tiêu thức	Tỉ lệ khuyết dữ liệu (%)
Number_of_Pixels	18.8%
Core_Speed	27.5%
Memory	12.3%
Memory_Bandwidth	3.6%
Release_Value	0.9%

Có thể thấy, tỉ lệ khuyết thông tin của dữ liệu khá cao. Ta sẽ xử lý bằng cách thay giá trị khuyết bằng giá trị trung bình:

```
1 # Hàm thay giá trị trung bình cho các ô trống của col_value
2 # đồng thời gán đơn vị cho các ô trống của col_unit
3 replaceMean <- function(col_value, col_unit = NULL, unit = "") {
4   col_value[is.na(col_value)] <- mean(col_value, na.rm = TRUE)
5   if (!is.null(col_unit))
6     col_unit[is.na(col_unit)] <- unit
7
8   return(list(col_value = col_value, col_unit = col_unit))
9 }
10
11 # Thay giá trị trung bình cho Number_of_Pixels
12 temp <- replaceMean(df$Number_of_Pixels)
13 df$Number_of_Pixels <- temp$col_value
14
15 # Thay giá trị trung bình cho Core_Speed_Value
16 # đồng thời gán đơn vị MHz cho Core_Speed_Unit
17 temp <- replaceMean(df$Core_Speed_Value, df$Core_Speed_Unit, "MHz")
18 df$Core_Speed_Value <- temp$col_value
19 df$Core_Speed_Unit <- temp$col_unit
20
21 # Tương tự cho các cột còn lại
22 temp <- replaceMean(df$Memory_Value, df$Memory_Unit, "MB")
23 df$Memory_Value <- temp$col_value
24 df$Memory_Unit <- temp$col_unit
25
```

```
26 temp <- replaceMean(df$Memory_Bandwidth_Value, df$Memory_Bandwidth_Unit,
27                         "GB/sec")
28 df$Memory_Bandwidth_Value <- temp$col_value
29 df$Memory_Bandwidth_Unit <- temp$col_unit
30
31 temp <- replaceMean(df$Release_Value)
32 df$Release_Value <- temp$col_value
33
34 # Kiểm tra lại việc cập nhật
35 apply(is.na(df[, c("Name", "Number_of_Pixels", "Core_Speed_Value",
36                 "Memory_Value", "Memory_Bandwidth_Value", "Manufacturer",
37                 "Release_Value")]), MARGIN = 2, FUN = mean)
38 # Xem lại dữ liệu sau khi cập nhật
39 View(df)
```

```
> apply(is.na(df[, c("Name", "Number_of_Pixels", "Core_Speed_Value", "Memory_Value",
+                 "Memory_Bandwidth_Value", "Manufacturer", "Release_Value")]), MARGIN = 2, FUN = mean)
      Name      Number_of_Pixels      Core_Speed_Value      Memory_Value
0          0                  0                      0
Memory_Bandwidth_Value      Manufacturer      Release_Value
0          0                  0                      0
```

Hình 3.9: Sau khi cập nhật, tỉ lệ dữ liệu khuyết của các tiêu thức bằng 0

Name	Number_of_Pixels	Core_Speed_Value	Core_Speed.Unit	Memory_Value	Memory.Unit	Memory_Bandwidth_Value	Memory_Bandwidth.Unit	Manufacturer	Release_Date	Release_Value
1 GeForce GTS 150	2185228	738.0000	MHz	1024.000	MB	64.0000	GB/sec	Nvidia	2009-03-01	2009.250
2 Radeon HD 2900 XT 512MB	1049088	946.8939	MHz	512.000	MB	106.0000	GB/sec	AMD	2007-05-14	2007.417
3 Radeon HD 2900 Pro	1049088	946.8939	MHz	512.000	MB	51.2000	GB/sec	AMD	2007-12-07	2008.000
4 Radeon HD 2600 XT Diamond Edition	786432	946.8939	MHz	256.000	MB	36.8000	GB/sec	AMD	2007-07-01	2007.583
5 Radeon HD 2600 XT	786432	946.8939	MHz	256.000	MB	22.4000	GB/sec	AMD	2007-06-28	2007.500
6 Radeon HD 2600 XT 256MB GDDR4	786432	946.8939	MHz	256.000	MB	35.2000	GB/sec	AMD	2007-06-26	2007.500
7 Radeon HD 4890 Sapphire Vapor-X OC 2GB Edition	2073600	870.0000	MHz	2048.000	MB	134.4000	GB/sec	AMD	2009-07-13	2009.583
8 Radeon HD 2900 GT	786432	946.8939	MHz	256.000	MB	51.2000	GB/sec	AMD	2007-11-06	2007.917
9 FirePro D300	2073600	946.8939	MHz	2048.000	MB	160.0000	GB/sec	AMD	2014-01-18	2014.4083
10 Radeon 7000 64mb	2185228	946.8939	MHz	64.000	MB	2.9000	GB/sec	AMD	2001-01-02	2001.083
11 Quadro4 980 XGL	2185228	946.8939	MHz	128.000	MB	5.2000	GB/sec	Nvidia	2002-11-01	2002.917
12 Tesla M2090	2073600	650.0000	MHz	6144.000	MB	177.6000	GB/sec	Nvidia	2011-07-25	2011.583
13 Tesla K20	2185228	705.0000	MHz	512.0000	MB	168.0000	GB/sec	Nvidia	2012-11-01	2012.917
14 Tesla K40c	4096000	706.0000	MHz	12288.000	MB	288.4000	GB/sec	Nvidia	2013-11-12	2013.917
15 All-in-Wonder Radeon 7500	2185228	946.8939	MHz	64.000	MB	5.8000	GB/sec	AMD	2002-01-22	2002.083
16 Radeon R7 250 v2 MSI OC 2GB + Radeon R7 7870K Dual	1440000	1050.0000	MHz	3072.000	MB	57.6000	GB/sec	AMD	2015-05-28	2015.417
17 Tesla K10	2185228	946.8939	MHz	8192.000	MB	320.0000	GB/sec	Nvidia	2012-05-15	2012.417
18 Tesla K20X	2185228	732.0000	MHz	6144.000	MB	249.6000	GB/sec	Nvidia	2012-11-12	2012.917
19 Iris iD 6167U	1049088	300.0000	MHz	2872.769	MB	34.1000	GB/sec	Intel	2015-09-01	2015.750
20 Radeon 9800 XT	786432	946.8939	MHz	256.000	MB	23.4000	GB/sec	AMD	2003-10-01	2003.833
21 Tesla C2070	2185228	575.0000	MHz	6144.000	MB	144.0000	GB/sec	Nvidia	2010-09-01	2010.750
22 Tesla C2075	2185228	575.0000	MHz	6144.000	MB	144.0000	GB/sec	Nvidia	2011-07-01	2011.583
23 Quadro4 750 XGL	2185228	946.8939	MHz	128.000	MB	3.6000	GB/sec	Nvidia	2002-02-19	2002.167
24 Quadro4 900 XGL	2185228	946.8939	MHz	128.000	MB	5.2000	GB/sec	Nvidia	2002-11-01	2002.917
25 Iris iS 6360U	1049088	300.0000	MHz	2872.769	MB	34.1000	GB/sec	Intel	2015-09-01	2015.750
26 Quadro4 780 XGL	2185228	946.8939	MHz	128.000	MB	4.4000	GB/sec	Nvidia	2002-11-01	2002.917

Hình 3.10: Xem qua một vài phần tử đầu tiên của dữ liệu sau cập nhật

4 Thống kê tóm tắt

4.1 Các giá trị đặc trưng của mẫu

Dùng các hàm `apply()`, `mean()`, `median()`, `sd()`, `min()`, `max()`,... để thống kê các đặc trưng cơ bản của các tiêu thức định lượng.

```
1 # Trích xuất các mẫu dữ liệu định lượng
2 quantitative_df <- df[, c("Number_of_Pixels", "Core_Speed_Value",
3                           "Memory_Value", "Memory_Bandwidth_Value", "Release_Value")]
4
5 # Tính trung bình của từng mẫu
6 mean <- apply(quantitative_df, 2, mean)
7 # Tính phương sai (hiệu chỉnh)
8 s2 <- apply(quantitative_df, 2, var)
9 # Tính các điểm tứ phân vị
10 Q1 <- apply(quantitative_df, 2, function(x) quantile(x, probs=0.25))
11 Q2 <- apply(quantitative_df, 2, function(x) quantile(x, probs=0.50))
12 Q3 <- apply(quantitative_df, 2, function(x) quantile(x, probs=0.75))
13 # Tính giá trị nhỏ nhất
14 min <- apply(quantitative_df, 2, min)
15 # Tính giá trị lớn nhất
16 max <- apply(quantitative_df, 2, max)
17 # In các đặc trưng theo dạng bảng, làm tròn 2 chữ số thập phân
18 round(data.frame(mean, sd, Q1, Q2, Q3, med, min, max), 2)
```

Kết quả khi thực thi đoạn chương trình trên:

	mean	s2	q1	q2	q3	min	max
Number_of_Pixels	2185227.9132	1.974064e+12	1440000.000	2073600.0000	2185227.91	307200.0000	9216000.00
Core_Speed_Value	946.8939	5.476335e+04	851.000	946.8939	1046.00	100.0000	1784.00
Memory_Value	2872.7689	6.831529e+06	1024.000	2048.0000	4096.00	16.0000	32000.00
Memory_Bandwidth_Value	137.1846	1.757990e+04	28.800	112.0000	192.30	0.7812	1280.00
Release_Value	2012.5224	1.091260e+01	2011.083	2012.8333	2014.75	1998.2500	2017.75

Bảng 4.1: Bảng các giá trị đặc trưng của các mẫu định lượng

4.2 Phân phối tần số

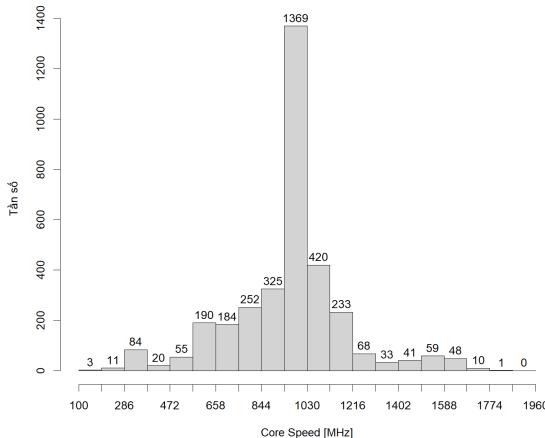
Dùng hàm `hist()` với các tham số phù hợp để vẽ biểu đồ tần số cho một số mẫu định lượng.

```
1 # Hàm tự định nghĩa để vẽ biểu đồ tần số của một mẫu sample
2 plotHist <- function(sample, name) {
3   # Thiết lại lai khoảng chia trên trục tung
4   x_axis = seq(floor(min(sample)), max(sample)*1.1,
5                 by = floor((max(sample)*1.1-min(sample))/20))
6
7   hist(sample, main="", xlab=name, xaxt="n", ylab="Tần số",
8         labels=TRUE, breaks=x_axis)
9   # Hiện trục tung với khoảng chia đã thiết lập
```

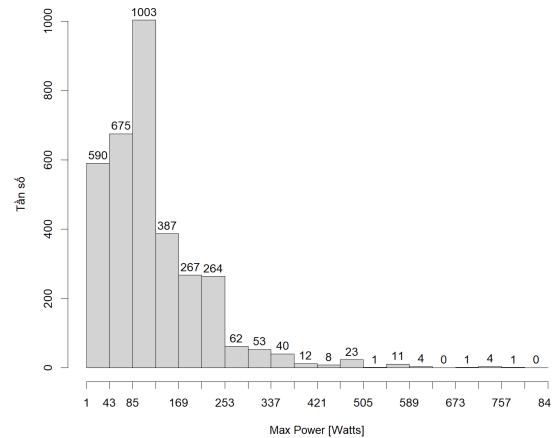
```

10     axis(1, at = x_axis)
11     return(NULL)
12 }
13
14 # Lần lượt dùng hàm đã định nghĩa để vẽ biểu đồ tần số của một vài mẫu
15 plotHist(df$Core_Speed_Value, name = "Core Speed [MHz]")
16 plotHist(df$Max_Power_Value, name = "Max Power [Watts]")
17 plotHist(df$Memory_Bandwidth_Value, name = "Memory Bandwidth [GB/sec]")
18 plotHist(df$Memory_Speed_Value, name = "Memory Speed [MB]")

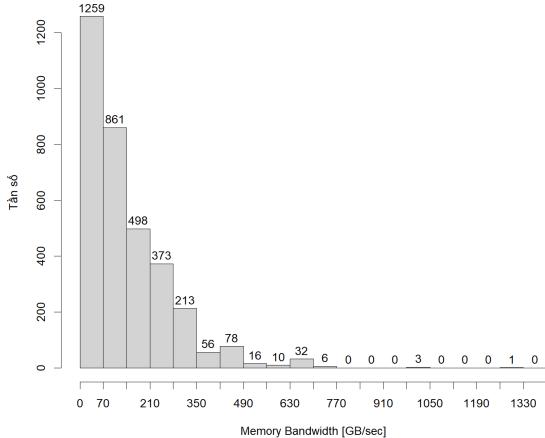
```



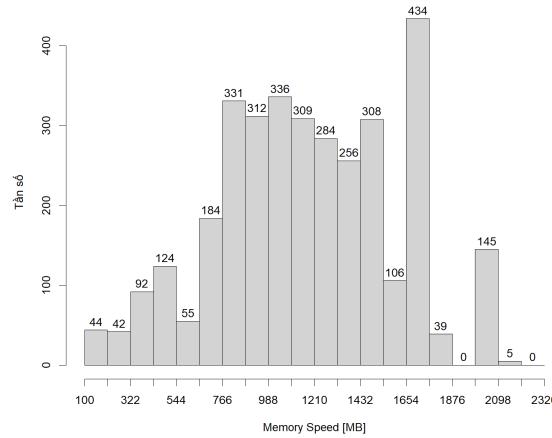
Hình 4.1: Biểu đồ phân phối tần số mẫu Core Speed



Hình 4.2: Biểu đồ phân phối tần số mẫu Max Power



Hình 4.3: Biểu đồ phân phối tần số mẫu Memory Bandwidth



Hình 4.4: Biểu đồ phân phối tần số mẫu Memory Speed

Nhận xét:

Các mẫu **Core Speed** và **Memory Speed** phân phối có dạng tương đối chuẩn, có một vài giá trị có tần số dột biến, vượt ngoài phân phối chuẩn. Nguyên do có thể đến từ việc thu thập dữ liệu chưa đủ ngẫu

nhiên, tỉ lệ khuyết dữ liệu khá cao, việc xử lý dữ liệu khuyết chưa đủ tốt,...

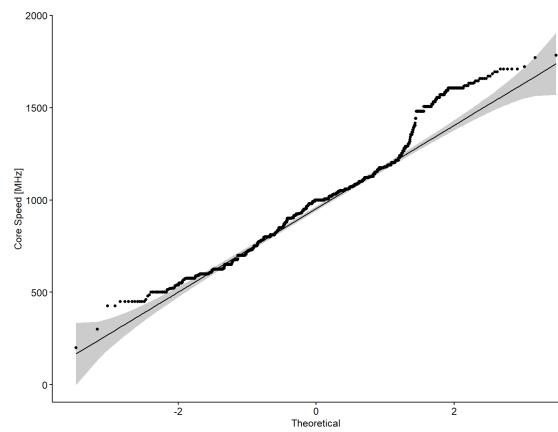
Các mẫu **Max Power** và **Memory Bandwidth** có phân phối lệch trái, phần lớn giá trị nằm trong khoảng từ < 170 [Watts] cho **Max Power** và < 200 [GB/sec] cho **Memory Bandwidth**. Phần còn lại chiếm tỉ lệ tương đối nhỏ.

4.3 Phân phối chuẩn

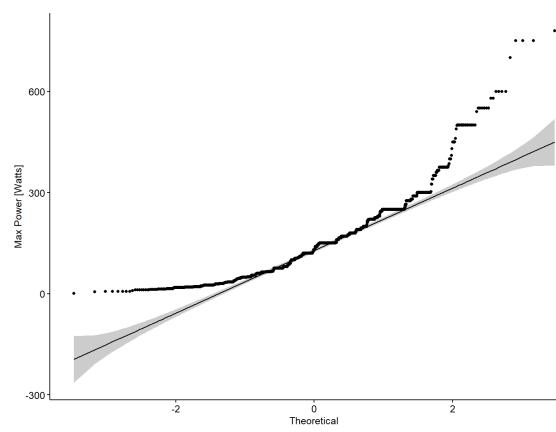
Phân phối chuẩn là giả thuyết rất quan trọng cho một số mô hình. Vì thế, ta cần kiểm tra xem các yếu tố đã trích xuất có (hoặc có gần với) phân phối chuẩn hay không.

Để làm điều đó, ta dùng hàm `ggqqplot()` trong thư viện `ggpubr` để vẽ biểu đồ QQ-plot (biểu đồ kiểm tra mẫu có phân phối chuẩn hay không) cho các mẫu tiêu thức định lượng.

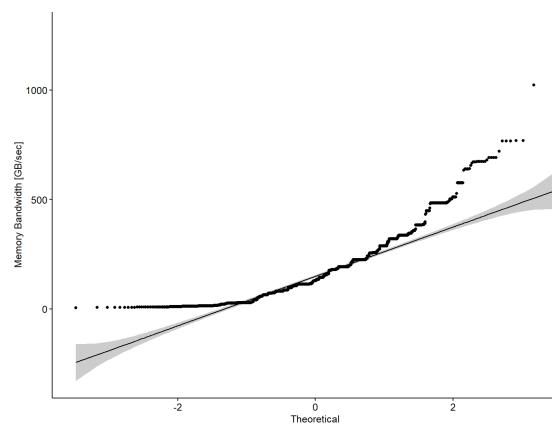
```
1 library(ggpubr)
2 ggqqplot(df$core_Speed_Value, ylab = "Core Speed [MHz]")
3 ggqqplot(df$Max_Power_Value, ylab = "Max Power [Watts]")
4 ggqqplot(df$Memory_Bandwidth_Value, ylab = "Memory Bandwidth [GB/sec]")
5 ggqqplot(df$Memory_Speed_Value, ylab = "Memory Speed [MHz]")
```



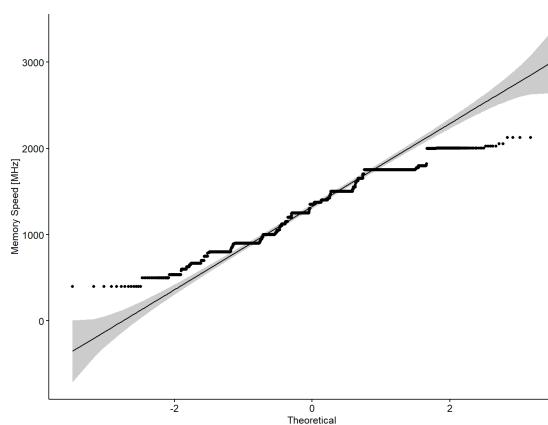
Hình 4.5: Biểu đồ QQ-plot cho mẫu **Core Speed**



Hình 4.6: Biểu đồ QQ-plot cho mẫu **Max Power**



Hình 4.7: Biểu đồ QQ-plot
cho mẫu **Memory Bandwidth**



Hình 4.8: Biểu đồ QQ-plot
cho mẫu **Memory Speed**

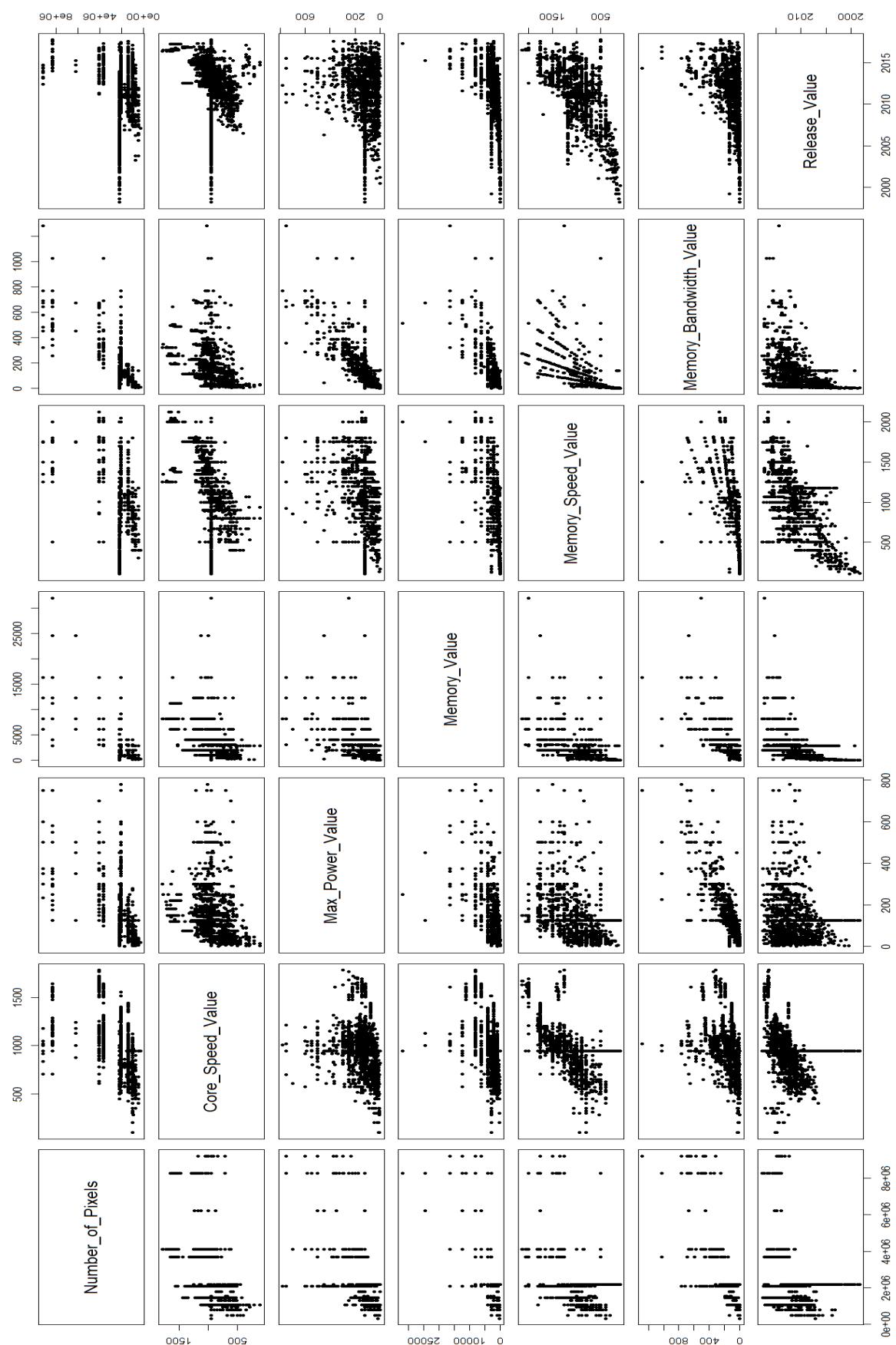
Nhận xét:

Có thể thấy những giá trị quan sát của mẫu **Core Speed** và **Memory Speed** phần nhiều tập trung gần đường thẳng kì vọng của phân phối chuẩn, do đó có thể xem hai mẫu này có phân phối tương đối chuẩn.

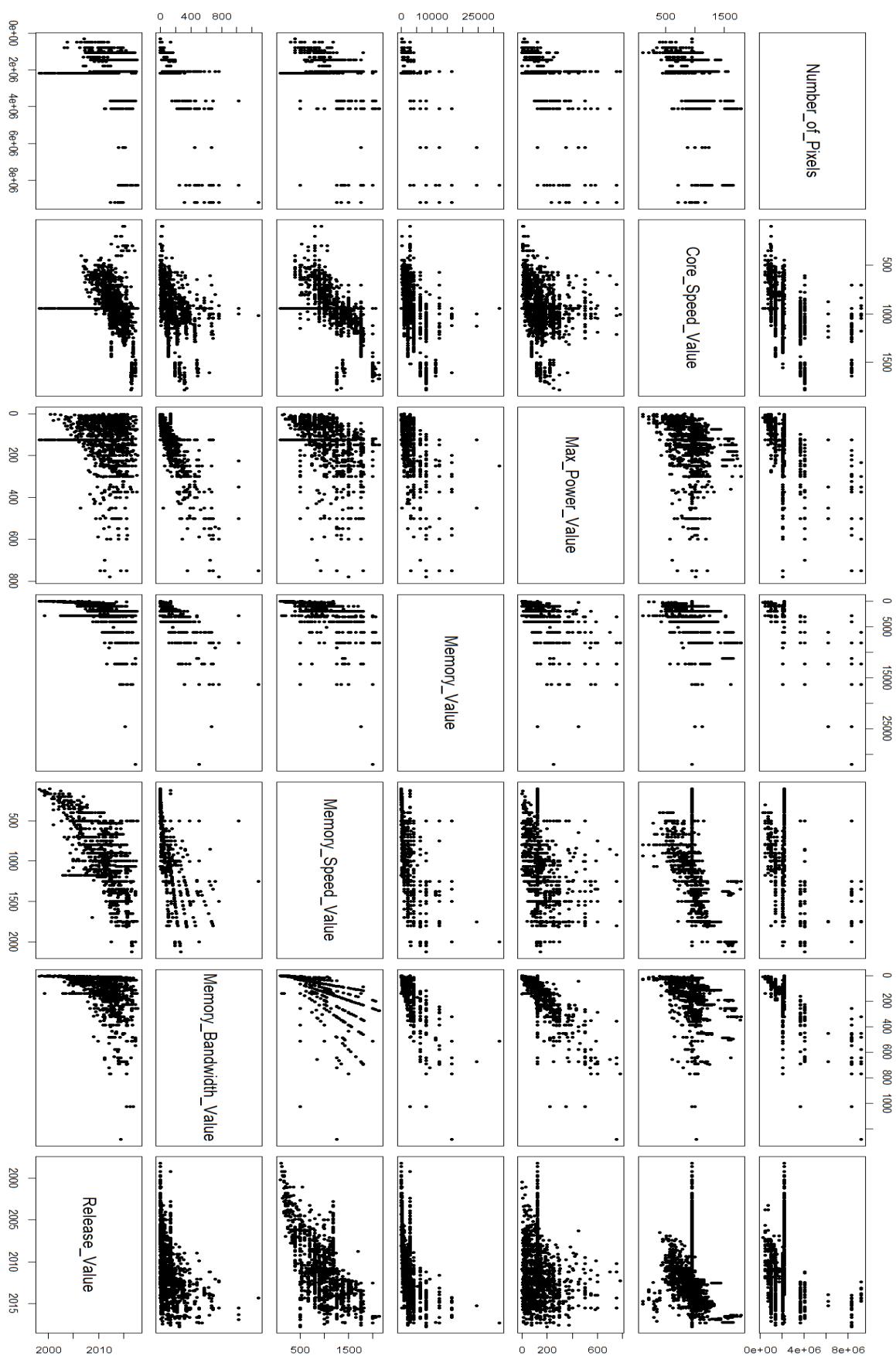
Ngược lại, các mẫu **Max Power** và **Memory Bandwidth** có khá nhiều lượng quan sát lệch xa đường thẳng kì vọng của phân phối chuẩn, do đó hai mẫu này có khả năng không tuân theo phân phối chuẩn.

4.4 Mối liên hệ giữa các biến

Dùng hàm `pairs()` với các tham số phù hợp để vẽ ma trận các biểu đồ tán xạ thể hiện mối liên hệ giữa các mẫu định lượng với nhau.



Hình x.x: Ma trận các biểu đồ tán xạ thể hiện mối liên hệ giữa các mẫu



Hình x.x: Ma trận các biểu đồ tán xạ thể hiện mối liên hệ giữa các mẫu

Nhận xét:

Mẫu **Number_of_Pixels** thường như độc lập với các mẫu còn lại (do các cặp dữ liệu giữa mẫu này với các mẫu khác phần lớn phân bổ song song với hai trục). Có thể kết luận sơ bộ rằng độ phân giải của màn hình để GPU hoạt động tốt nhất (Best Resolution) là độc lập với các thông số khác của GPU.

Có một số cặp mẫu có mối liên hệ mạnh với nhau, do phân biểu đồ phân tán giữa chúng có một quy luật nhất định. Chẳng hạn biểu đồ phân tán giữa **Memory_Speed** và **Release_Value** cho thấy theo thời gian thì tốc độ truy xuất bộ nhớ của các GPU được thiết kế ngày càng tốt.

5 Thông kê suy diễn

5.1 Tìm khoảng tin cậy một mẫu

5.1.1 Mục tiêu

- Nắm được các khái niệm liên quan đến lý thuyết ước lượng, bài toán ước lượng khoảng tin cậy thông số của một tổng thể.
- Biết cách giải một bài toán tìm khoảng tin cậy một mẫu.
- Tìm hiểu và hiện thực các lệnh của ngôn ngữ R để giải quyết bài toán.

5.1.2 Bài toán

Xét mẫu **Core_Speed_Value** trong dữ liệu đã trích xuất và tiền xử lý. Tiến hành ước lượng khoảng tin cậy trung bình tổng thể của mẫu này với độ tin cậy 95%.

5.1.3 Nhận xét bài toán

Đây là bài toán tìm khoảng tin cậy cho trung bình tổng thể – một mẫu. Không biết phương sai tổng thể. Ta dùng R kết hợp với các kí thuật đã học để giải quyết bài toán.

5.1.4 Kiến thức R

Trong R, các hàm tìm khoảng tin cậy cho trung bình (cũng là các hàm kiểm định cho trung bình) cùng các tham số trong từng hàm được xây dựng dựa trên các trường hợp đã phân ra trong phần lí thuyết. Cụ thể:

- Khi tổng thể tuân theo phân phối chuẩn, đã biết phương sai, ta dùng hàm **z.test()**.
- Khi tổng thể tuân theo phân phối chuẩn, không biết phương sai, ta dùng hàm **t.test()**.
- Khi tổng thể không tuân theo phân phối chuẩn nhưng cỡ mẫu lớn thì theo định lý giới hạn, thống kê được sử dụng có phân phối xấp xỉ phân phối chuẩn nên ta dùng hàm **z.test()**. Tuy nhiên, khi



cỡ mẫu lớn thì phân phối chuẩn và phân phối Student xấp xỉ nhau nên trong trường hợp này, ta cũng có thể dùng phân phối Student để thay thế, tức dùng hàm `t.test()`.

Lưu ý: Hàm `t.test()` có sẵn trong các gói cơ bản của R, còn hàm `z.test()` phải sử dụng thông qua gói thư viện `BSDA`.

5.1.5 Tiết hành

Ta đã kiểm tra mẫu **Core_Speed_Value** và thấy rằng kích thước mẫu lớn, tổng thể của mẫu này có dạng phân phối chuẩn, không biết phương sai tổng thể. Do đó, ta dùng hàm `t.test()` để tìm khoảng tin cậy trung bình cho tổng thể của mẫu này:

```
1 # Trích xuất mẫu Core_Speed_Value từ tập dữ liệu
2 sample <- df$Core_Speed_Value
3 # Độ tin cậy = 1 - alpha => alpha = 1 - Độ tin cậy
4 alpha <- 1 - 0.95
5 # Áp dụng t-test cho mẫu sample
6 t.test(sample, conf.level = 1 - alpha)
```

5.1.6 Kết quả

One Sample t-test

```
data: sample
t = 236.14, df = 3405, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 939.0321 954.7558
sample estimates:
mean of x
 946.8939
```

5.1.7 Nhận xét

Ta quan tâm đến dòng `95 percent confidence interval`. Kết quả cho ta biết với độ tin cậy 95%, khoảng ước lượng hai phía cho trung bình của tổng thể **Core Speed** là

$$(939.0321; 954.7558)$$

5.1.8 Kết luận

Trong bài toán này, nhóm đã thành công trong việc tìm hiểu các lệnh của R để thực hiện việc tính toán, tìm ra khoảng ước lượng cho trung bình tổng thể của một mẫu mêt kì trong tập dữ liệu, với độ tin cậy cho trước. Tuy nhiên, như đã nói, do tỉ lệ khuyết dữ liệu của mẫu khá lớn, các phần tử bị khuyết được thay bằng trung bình của các phần tử còn lại, nên nhìn chung kết quả mà nhóm tìm được có thể sai khác đôi chút với kết quả chính xác.



5.2 Kiểm định hai mẫu

5.2.1 Mục tiêu

- Nắm được các khái niệm liên quan đến kiểm định giả thuyết thống kê.
- Biết cách giải một bài toán kiểm định tỉ lệ một mẫu, kiểm định tỉ lệ hai mẫu.
- Tìm hiểu và hiện thực các lệnh của ngôn ngữ R để giải quyết bài toán.

5.2.2 Bài toán

Trích xuất ra hai mẫu con từ **Core_Speed_Value** dựa trên hai tần số cao nhất của tiêu thức **Manufacturer** (nhãn hàng sản xuất loại GPU) tương ứng. Từ hai mẫu con đã trích, hãy kiểm định xem với mức ý nghĩa 5%, có thể coi giá trị trung bình về **Core Speed** của tổng thể các GPU do hai nhãn hàng sản xuất là bằng nhau hay không.

5.2.3 Nhận xét bài toán

Dây là bài toán kiểm định trung bình hai mẫu độc lập (do mỗi GPU chỉ có thể đến từ một trong hai nhãn hàng, và hai nhãn hàng sản xuất độc lập nhau), không biết phương sai tổng thể. Ta dùng R kết hợp với các lí thuyết đã học để giải quyết bài toán.

5.2.4 Kiến thức R

Các hàm **z.test()** và **t.test()** trong R giúp ta kiểm định cho trung bình hai mẫu. Các hàm này cùng các tham số trong từng hàm được xây dựng dựa trên các trường hợp đã phân ra trong phần lí thuyết. Cụ thể:

- Khi các tổng thể tuân theo phân phối chuẩn, đã biết phương sai, ta dùng hàm **z.test()**.
- Khi các tổng thể tuân theo phân phối chuẩn, không biết phương sai, ta dùng hàm **t.test()**.
- Khi các tổng thể không tuân theo phân phối chuẩn nhưng cỡ mẫu lớn thì theo định lý giới hạn, thống kê được sử dụng có phân phối xấp xỉ phân phối chuẩn nên ta dùng hàm **z.test()**. Tuy nhiên, khi cỡ mẫu lớn thì phân phối chuẩn và phân phối Student xấp xỉ nhau nên trong trường hợp này, ta cũng có thể dùng phân phối Student để thay thế, tức dùng hàm **t.test()**.

Lưu ý: Hàm **t.test()** có sẵn trong các gói cơ bản của R, còn hàm **z.test()** phải sử dụng thông qua gói thư viện **BSDA**.

Để kết luận bài toán, ta dựa vào kết quả **p-value** mà hàm kiểm định trả về. Nếu:

- $p\text{-value} \leq \alpha$: ta bác bỏ giả thuyết trung bình về **Core Speed** của tổng thể các GPU do hai nhãn hàng sản xuất là bằng nhau, tức thừa nhận có sự khác biệt về trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất từ hai nhãn hàng.

- $p\text{-value} > \alpha$: ta chưa thể bác bỏ giả thuyết trung bình về **Core Speed** của tổng thể các GPU do hai nhãn hàng sản xuất là bằng nhau, tức chưa thể kết luận có sự khác biệt về trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất từ hai nhãn hàng.

Nếu có sự khác biệt về trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất từ hai nhãn hàng và ta muốn biết cụ thể liệu trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất từ nhãn hàng nào lớn hơn, ta quan tâm đến giá trị t mà hàm kiểm định trả về. Nếu:

- $t < 0$: trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất từ nhãn hàng I là nhỏ hơn nhãn hàng II.
- $t > 0$: trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất từ nhãn hàng I là lớn hơn nhãn hàng II.

5.2.5 Tiết hành

Tiết hành trích xuất hai mẫu con từ **Core_Speed_Value**:

- Thống kê số lượng GPU mỗi nhãn hàng sản xuất có trong bảng dữ liệu bằng hàm **table()**.

```
1 # Thống kê các hãng sản xuất GPU
2 table(df$Manufacturer)
```

Kết quả khi thực thi đoạn lệnh trên:

	AMD	ATI	Intel	Nvidia
	1317	92	254	1743

Từ thống kê, có thể thấy bảng số liệu chứa nhiều thông tin GPU đến từ **AMD** và **Nvidia** nhất.

Do đó, ta sẽ trích xuất hai mẫu con của **Core_Speed_Value** theo hai nhãn hàng này.

```
1 # Trích xuất hai mẫu con từ Core_Speed_Value theo Manufacturer là
2 # "AMD" và "Nvidia"
3 sample1 <- df$Core_Speed_Value[df$Manufacturer == "AMD"]
4 sample2 <- df$Core_Speed_Value[df$Manufacturer == "Nvidia"]
```

Tiết hành kiểm định trung bình hai mẫu con vừa trích xuất:

```
1 # Mức ý nghĩa của kiểm định, cho trước = 5%
2 alpha <- 0.05
3 # Dùng hàm t.test() để kiểm định trung bình cho hai mẫu
4 t.test(sample1, sample2, var.equal = (var(sample1) == var(sample2)),
5         conf.level = 1 - alpha, alternative = "two.sided")
```

5.2.6 Kết quả

welch Two Sample t-test

```
data: sample1 and sample2
t = -12.751, df = 2798.7, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-104.33954 -76.52665
sample estimates:
mean of x mean of y
914.2359 1004.6690
```

5.2.7 Nhận xét

Do phương sai của hai mẫu khác biệt nhau (phép so sánh `var(sample1) == var(sample2)` trả về FALSE) nên hàm `t.test()` sử dụng Welch Two Sample t-test để kiểm định cho bài toán này.

Ta quan tâm đến giá trị p-value. Do p-value $< 2.2 \times 10^{-16}$, bé hơn rất nhiều so với mức ý nghĩa $\alpha = 5\%$. Do đó, ta kết luận có sự khác biệt về trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất bởi hai nhãn hàng **AMD** và **Nvidia**.

Cụ thể, giá trị $t < 0$, do đó, kết luận trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất bởi nhãn hàng **AMD** là thấp hơn nhãn hàng **Nvidia**.

Ngoài ra, hàm kiểm định còn cung cấp khoảng tin cậy hai phía cho chênh lệch giữa trung bình giá trị **Core Speed** của tổng thể các GPU được sản xuất bởi nhãn hàng **AMD** so với nhãn hàng **Nvidia** với độ tin cậy $1 - \alpha = 95\%$:

$$(-104.33954; -76.52665)$$

5.2.8 Kết luận

Trong bài toán này, nhóm đã thành công trong việc tìm hiểu các lệnh của R để thực hiện việc tính toán, kiểm định được trung bình tổng thể của hai mẫu độc lập trong tập dữ liệu với mức ý nghĩa. Tuy nhiên, như đã nói, do tỉ lệ khuyết dữ liệu của mẫu khá lớn, các phần tử bị khuyết được thay bằng trung bình các phần tử còn lại, nên nhìn chung kết quả mà nhóm tìm được có thể có sai khác so với kết quả chính xác.

5.3 Phân tích phương sai

5.3.1 Mục tiêu

- Nắm được các khái niệm liên quan đến phân tích phương sai ANOVA.
- Biết cách giải một bài toán phân tích phương sai một yếu tố.
- Tìm hiểu và hiện thực các lệnh của ngôn ngữ R để giải quyết bài toán.

5.3.2 Bài toán

Dùng mô hình ANOVA kiểm định trung bình của yếu tố **Memory_Speed** giữa các nhóm có **Name** (tên) khác nhau với mức ý nghĩa $\alpha = 5\%$.

5.3.3 Kiểm tra giả thiết áp dụng mô hình ANOVA

Như đã trình bày ở phần Thống kê tách, trong các mẫu định lượng thì yếu tố **Memory_Speed** là một trong những mẫu có phân phối gần với dạng chuẩn, đây là giả thiết cần cho mô hình ANOVA.

Tiếp theo, ta cần kiểm tra sự tương đồng về phương sai mẫu giữa các nhóm quan sát. Để làm điều đó, ta có thể dùng kiểm định Barlett hoặc kiểm định Levene.

Tiếp theo, do ta phân loại các mẫu với nhau thông qua tên của chúng, nên có thể coi các mẫu quan sát được lấy một cách độc lập.

5.3.4 Kiến thức R

5.3.4.1 Kiểm định sự tương đồng về phương sai giữa các mẫu

Để kiểm tra sự tương đồng về phương sai giữa các nhóm quan sát, R có hàm **bartlett.test()** để thực hiện kiểm định Barlett. Với giả thuyết không H_0 là phương sai mẫu giữa các nhóm quan sát tương đồng nhau, ta cần kiểm tra **p-value** hàm trả về. Nếu:

- $p\text{-value} \leq \alpha$: ta bác bỏ giả thuyết H_0 , tức thừa nhận sự khác nhau về phương sai giữa các mẫu.
- $p\text{-value} > \alpha$: ta chưa thể bác bỏ giả thuyết H_0 , tức chưa thể kết luận có sự khác nhau về phương sai giữa các mẫu. Lúc này, ta có thể coi như phương sai giữa các mẫu tương đồng nhau.

5.3.4.2 Mô hình ANOVA một yếu tố

Trong R, hàm **aov()** dùng để thực hiện phân tích phương sai cho một mẫu liên tục được chia ra theo từng nhóm dựa trên mẫu phân loại.

Lưu ý: cần dùng thêm hàm **summary()** cho kết quả của hàm **aov()** để thu được thống kê chi tiết.

5.3.4.3 Phân tích sâu

Nếu giả thuyết đối H_0 được bác bỏ, ta cần kiểm tra sự khác nhau của từng cặp nhóm với nhau bằng kiểm định LSD. Để thực hiện điều đó, R cung cấp hàm **LSD.test()**. Ngoài ra, ta cũng có thể kiểm định giả thuyết này thông qua hàm **pairwise.t.test()**.

5.3.5 Tiến hành

5.3.5.1 Kiểm tra - xử lý mẫu phân loại

Dùng hàm **table()** để thống kê số lượng quan sát tương ứng với từng tên trong mẫu **Name**



```
1 # Lưu kết quả của hàm table() vào biến và in kết quả đó ra
2 print(name_table <- table(df$Name))
```

FirePro	Geforce	GeForce	Iris	Mobility	Quadro	Radeon	Tesla
3	1	1188	2	11	18	741	2

Bảng 5.1: Kết quả khi thực thi đoạn chương trình trên

Ta thấy, có một vài loại GPU có số lượng quan sát khá thấp. Do đó, để mô hình ANOVA chính xác hơn, ta sẽ lọc bỏ các quan sát có số lượng không lớn hơn 10.

```
1 # Trích xuất dữ liệu với số lượng mỗi quan sát theo tên nhiều hơn 10
2 df_filtered <- df[df$Name %in% names(name_table[name_table > 10]), ]
3 # Kiểm tra lại thống kê các tên
4 table(df_filtered$Name)
```

GeForce	Mobility	Quadro	Radeon
1188	11	18	741

Bảng 5.2: Kết quả khi thực thi đoạn chương trình trên

Sau khi thực thi đoạn chương trình trên, ta còn lại bốn nhóm GPU. Đây sẽ là bốn nhóm phân loại cho đầu vào của mô hình ANOVA.

5.3.5.2 Kiểm tra giả thiết áp dụng mô hình ANOVA

Như đã trình bày, ta đã có giả thiết về phân phối chuẩn cũng như cách chọn mẫu độc lập. Do đó, ta cần kiểm định thêm về sự tương đồng của phương sai các nhóm thông qua hàm bartlett.test().

```
1 bartlett.test(Memory_Speed_Value ~ Name, data = df_filtered)
```

Bartlett test of homogeneity of variances

```
data: Memory_Speed_Value by Name
Bartlett's K-squared = 44.865, df = 3, p-value = 9.885e-10
```

Bảng 5.3: Kết quả khi thực thi đoạn chương trình trên

Nhận xét: p-value của kiểm định rất nhỏ so với mức ý nghĩa $\alpha = 5\%$. Điều này có nghĩa ta bác bỏ giả thuyết về sự bằng nhau giữa phương sai các nhóm. Nói cách khác, có sự khác biệt về phương sai giữa các loại GPU với nhau. Trên lý thuyết, mẫu này không thoả giả thiết có thể áp dụng mô hình ANOVA, tuy nhiên với giới hạn về mặt dữ liệu, ta giả sử mẫu thoả giả thiết và tiếp tục thực hiện mô hình.

5.3.5.3 Tiến hành phân tích phương sai

Dùng hàm `aov()` kết hợp `summary()` với những tham số phù hợp để tiến hành áp dụng mô hình phân tích phương sai một yếu tố cho mẫu `Memory_Speed_Value`, phân loại theo `Name` đã lọc.

```
1 # Lưu kết quả của mô hình ANOVA vào biến (để sử dụng lại cho sau này)
2 aov_result <- aov(Memory_Speed_Value ~ Name, data = df_filtered)
3 # Hiển thị kết quả chi tiết của mô hình ANOVA
4 summary(aov_result)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Name	3	7150640	2383547	16.64	1.13e-10 ***
Residuals	1954	279946227	143268		

				Signif. codes:	0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Bảng 5.4: Kết quả khi thực thi đoạn chương trình trên

Từ kết quả thu được ở trên, ta thống kê các giá trị đặc trưng của mô hình ANOVA:

Đại lượng	Giá trị
Bậc tự do $dfB = k - 1$	3
Bậc tự do $dfW = N - k$	1954
SSB	7150640
SSW	279946227
MSB	2383547
MSW	143268
Giá trị kiểm định thống kê F	16.64
p-value	1.13×10^{-10}

Nhận xét:

Giá trị thống kê kiểm định F biểu thị tỷ lệ bình phương trung bình giữa các nhóm với bình phương trung bình trong mỗi nhóm. Giá trị F khá cao, cho thấy trung bình của nhóm không phải tất cả đều bằng nhau.

Hơn nữa, p-value thấp hơn rất nhiều so với mức ý nghĩa $\alpha = 5\%$, cho thấy rõ ràng sự khác biệt về trung bình giữa các nhóm được quan sát là có ý nghĩa thống kê.

Tóm lại, kết quả mô hình ANOVA cho bài toán này dẫn đến việc bác bỏ giả thuyết không H_0 – giả thuyết cho rằng giá trị **Memory Speed** trung bình của các nhóm đều bằng nhau. Nói cách khác, ta có thể kết luận giá trị trung bình **Memory Speed** của các nhóm GPU khác nhau không cùng bằng nhau.

5.3.5.4 Phân tích sâu

Do giả thuyết không H_0 đã được bác bỏ, ta có nhu cầu kiểm định giả thuyết về trung bình **Memory Speed** giữa từng nhóm.

Sử dụng hàm `LSD.test()` trong thư viện `agricolae` để xem kết quả về giả thuyết trung bình tổng thể giữa các nhóm.

```
1 library(agricolae)
2 # Dùng LSD.test() cho kết quả của mô hình ANOVA để kiểm định
3 # trung bình giữa các nhóm
4 print(LSD.test(anova_result, "Name", alpha = 0.05))
```

```
$statistics
  MSerror      Df      Mean       CV
  143268.3 1954 1308.349 28.93019

$parameters
  test p.adjusted name.t ntr alpha
  Fisher-LSD      none     Name   4  0.05

$means
  Memory_Speed_Value      std      r      se
  GeForce      1344.6027 402.72239 1188 10.98163
  Mobility      854.5455 93.41987   11 114.12445
  Quadro        981.2778 282.77632   18 89.21518
  Radeon        1264.9082 340.87649  741 13.90484
  LCL          UCL Min Max Q25 Q50 Q75
  GeForce    1323.0658 1366.140 400 2127 1000 1376 1752.0
  Mobility    630.7270 1078.364 800 1000 800 800 900.0
  Quadro      806.3109 1156.245 650 1502 750 901 1187.5
  Radeon      1237.6384 1292.178 400 2000 1000 1250 1500.0

$comparison
NULL

$groups
  Memory_Speed_Value groups
  GeForce            1344.6027    a
  Radeon            1264.9082    b
  Quadro             981.2778    c
  Mobility           854.5455    c
```

Bảng 5.5: Kết quả khi thực thi đoạn chương trình trên



Nhận xét: Hàm `LSD.test()` trả về rất nhiều thông tin như MSW, trung bình, sai số, giá trị lớn nhất, nhỏ nhất, khoảng từ phân vị,... của từng nhóm. Tuy nhiên, ta cần quan tâm đến trường `$groups`.

Có thể thấy bốn tên GPU được chia thành ba nhóm *a*, *b* và *c*. Trong đó, **Quadro** và **Mobility** thuộc cùng một nhóm, còn **GeForce** và **Radeon** thuộc hai nhóm riêng, độc lập. Điều này thể hiện rằng với mức ý nghĩa $\alpha = 5\%$, chỉ có trung bình **Memory Speed** của tổng thể hai nhóm **Quadro** và **Mobility** xem như bằng nhau, còn giữa các cặp nhóm còn lại đều có sự khác biệt.

Để chắc chắn hơn vào kết luận trên, ta có thể dùng hàm `pairwise.t.test()` để xem ma trận các p-value của kiểm định Student theo cặp với độ lệch chuẩn gộp giữa các nhóm.

```
1 pairwise.t.test(df_filtered$Memory_Speed_Value, df_filtered$Name,  
2 p.adjust.method = "none")
```

```
Pairwise comparisons using t tests with pooled SD  
  
data: df_filtered$Memory_Speed_Value and df_filtered$Name  
  
      GeForce Mobility Quadro  
Mobility 2.0e-05 - -  
Quadro   5.5e-05 0.38175 -  
Radeon   7.3e-06 0.00037 0.00171
```

Bảng 5.6: Kết quả khi thực thi đoạn chương trình trên

Nhận xét: Ma trận p-value giữa các nhóm từ hàm `pairwise.t.test()` cho thấy chỉ có p-value giữa **Quadro** và **Mobility** (bằng 0.38175) là lớn hơn mức ý nghĩa $\alpha = 5\%$. Điều này phù hợp với kết luận đã đưa ra bên trên.

5.3.6 Kết luận

Trong bài toán này, nhóm đã thành công trong việc, nghiên cứu các lý thuyết về phân tích phương sai ANOVA, tìm hiểu các lệnh của R để thực hiện việc tính toán, áp dụng mô hình ANOVA cho một yếu tố cụ thể, đồng thời kiểm tra sâu được giả thuyết về trung bình giữa các nhóm của yếu tố, đưa ra được kết luận cuối cùng cho bài toán.

Tuy nhiên, kết quả nhóm trình bày có thể chưa thật sự chính xác. Nguyên nhân có thể đến từ nhiều yếu tố:

- Ảnh hưởng của việc loại bỏ các quan sát khuyết trong quá trình tiền xử lý số liệu.
- Tổng thể chưa thoả giả thiết của mô hình: phân phối chưa thật sự chuẩn, phương sai của các nhóm không bằng nhau.
- Sự chênh lệch lớn về số lượng quan sát giữa các nhóm.



Chính vì những hạn chế trên, nhìn chung kết quả mà nhóm tìm được có thể có sai khác so với kết quả chính xác.

5.4 Hồi quy tuyến tính

5.4.1 Mục tiêu

- Nắm được các khái niệm liên quan đến mô hình hồi quy tuyến tính.
- Biết cách giải một bài toán hồi quy tuyến tính đơn, hồi quy tuyến tính bội.
- Tìm hiểu và hiện thực các lệnh của ngôn ngữ R để giải quyết bài toán, biết cách đọc, giải thích, vẽ các biểu đồ minh họa cho kết quả.

5.4.2 Bài toán

Từ các yếu tố đã trích xuất, xây dựng mô hình hồi quy tuyến tính đánh giá các yếu tố ảnh hưởng đến **Release _ Price** (giá phát hành) của GPU với mức ý nghĩa $\alpha = 5\%$.

5.4.3 Kiến thức R

5.4.4 Tiến hành

5.4.4.1 Xử lý dữ liệu

Trích ra các mẫu định lượng trong tập dữ liệu, đồng thời lọc lấy những quan sát có yếu tố **Release _ Price** trong tập dữ liệu (bỏ đi các quan sát không có **Release _ Price** – bước này đã tạm thời bỏ qua trong phần xử lí dữ liệu khuyết).

```
1 # Trích các mẫu định lượng
2 samples <- df[, c("Release_Price", "Number_of_Pixels",
3 #                   "Core_Speed_Value", "Max_Power_Value",
4 #                   "Memory_Value", "Memory_Bandwidth_Value",
5 #                   "Memory_Speed_Value", "Release_Value")]
6 # Giữ lại những dòng có giá trị Release_Price
7 samples <- samples[!is.na(samples$Release_Price), ]
8 # Xem dữ liệu sau khi trích xuất và xử lý
9 View(samples)
```



	Release_Price	Number_of_Pixels	Core_Speed_Value	Max_Power_Value	Memory_Value	Memory_Bandwidth_Value	Memory_Speed_Value	Release_Value
48	2999.00	9216000	705	375	12288	672.0	1750	2014.417
50	2999.00	8294400	705	375	12288	672.0	1750	2014.417
52	1099.00	8294400	1140	250	12288	336.6	1753	2015.250
55	1998.00	6220800	1000	450	24576	673.2	1753	2015.250
56	999.00	8294400	1000	250	12288	336.6	1753	2015.250
57	1029.99	8294400	1127	250	12288	336.6	1753	2015.250
60	999.00	8294400	1000	300	12288	384.0	500	2016.667
62	1299.00	8294400	1000	375	16384	512.0	500	2016.667
66	999.00	3686400	889	250	6144	336.0	1750	2014.167
69	999.00	3686400	837	250	6144	288.4	1502	2013.167
171	249.00	2073600	960	190	1024	134.4	1050	2009.417
173	249.00	2073600	850	190	1024	124.8	975	2009.333
184	130.00	1440000	575	95	512	57.6	900	2008.833
187	109.00	1049088	750	80	512	51.2	800	2009.333
203	67.00	1049088	750	59	512	25.6	800	2008.750
942	1499.00	8294400	1018	500	8192	640.0	1250	2014.333
945	649.00	8294400	1000	175	4096	512.0	500	2015.667
949	559.00	4096000	1000	375	4096	512.0	500	2015.583
950	649.00	4096000	1050	275	4096	512.0	500	2015.500
951	549.00	4096000	1000	275	4096	512.0	500	2015.583
959	419.00	4096000	1100	300	8192	390.4	1525	2015.500
962	858.00	8294400	1050	550	16384	768.0	1500	2015.500
966	429.00	4096000	1050	275	8192	384.0	1500	2015.500
968	429.00	4096000	1050	275	8192	384.0	1500	2015.500
969	329.00	3686400	1010	375	8192	384.0	1500	2015.500

Bảng 5.7: Kết quả khi thực thi đoạn chương trình trên

5.4.4.2 Tiến hành

*Xây dựng mô hình

Dùng hàm `lm()` để tiến hành áp dụng mô hình hồi quy tuyến tính cho bộ dữ liệu đã xử lý, với `Release_Price` là biến phụ thuộc, các biến còn lại là các biến độc lập.

```
1 # Dấu . thể hiện các biến còn lại trong samples
2 lm_result <- lm(Release_Price ~ ., data = samples)
3 summary(lm_result)
```

```
Call:  
lm(formula = Release_Price ~ ., data = samples)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-1744.0  -167.0   25.8  144.5 11253.2  
  
Coefficients:  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) -4.250e+04  4.853e+04 -0.876  0.38171  
Number_of_Pixels -3.544e-05  4.168e-05 -0.850  0.39568  
Core_Speed_Value -3.979e-02  1.798e-01 -0.221  0.82498  
Max_Power_Value  5.704e+00  6.951e-01  8.206 2.43e-15 ***  
Memory_Value      1.118e-01  1.922e-02  5.814 1.16e-08 ***  
Memory_Bandwidth_Value -2.979e+00  6.707e-01 -4.442 1.13e-05 ***  
Memory_Speed_Value -3.254e-01  1.153e-01 -2.822  0.00498 **  
Release_Value     2.120e+01  2.416e+01  0.877  0.38070  
---  
signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
Residual standard error: 626.7 on 449 degrees of freedom  
Multiple R-squared:  0.3242,    Adjusted R-squared:  0.3136  
F-statistic: 30.77 on 7 and 449 DF,  p-value: < 2.2e-16
```

Bảng 5.8: Kết quả khi thực thi đoạn chương trình trên

Với giả thiết không H_0 : Biến độc lập không có ý nghĩa cho mô hình, ta xem xét p-value và so sánh với mức ý nghĩa $\alpha = 5\%$ để kết luận về giả thiết H_0 cho từng biến:

Biến	p-value	Bắc bỏ H_0 – Thì thừa nhận biến có ý nghĩa cho mô hình
Number_of_Pixels	0.39568	
Core_Speed	0.82498	
Max_Power	2.43×10^{-15}	✓
Memory	0.16×10^{-8}	✓
Memory_Bandwidth	1.13×10^{-5}	✓
Memory_Speed	0.00498	
Release_Value	0.38070	

Nhận xét: Do các biến **Max_Power**, **Memory** và **Memory_Bandwidth** có ý nghĩa cho mô hình, ta sẽ giũa lại các biến này, đồng thời loại đi những biến còn lại.

Gọi lại hàm **lm()** để xây dựng mô hình hồi quy tuyến tính với các biến có nghĩa:

```
1 # Xây dựng mô hình hồi quy tuyến tính với cái biến độc lập có ý nghĩa  
2 lm_result <- lm(Release_Price ~ Max_Power_Value + Memory_Value)
```



```
3           + Memory_Bandwidth_Value, data = samples)
4 summary(lm_result)
```

Call:
`lm(formula = Release_Price ~ Max_Power_Value + Memory_Value +
Memory_Bandwidth_Value, data = samples)`

Residuals:

Min	1Q	Median	3Q	Max
-1624.9	-129.1	-7.6	132.5	11575.7

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-332.40698	60.82860	-5.465	7.67e-08 ***
Max_Power_Value	5.94176	0.56976	10.429	< 2e-16 ***
Memory_Value	0.08747	0.01510	5.791	1.31e-08 ***
Memory_Bandwidth_Value	-3.07001	0.52452	-5.853	9.28e-09 ***

Signif. codes:	0 ‘***’	0.001 ‘**’	0.01 ‘*’	0.05 ‘.’
	0.1 ‘ ’	1		

Residual standard error: 631.1 on 453 degrees of freedom
Multiple R-squared: 0.3085, Adjusted R-squared: 0.3039
F-statistic: 67.36 on 3 and 453 DF, p-value: < 2.2e-16

Bảng 5.9: Kết quả khi thực thi đoạn chương trình trên

*Kết luận sơ bộ

Ta đã xây dựng xong mô hình hồi quy tuyến tính cho biến **Release_Price** phụ thuộc vào các biến độc lập có ý nghĩa. Kết quả trên cho ta một số thông tin:

- Dựa vào cột **Estimate**, ta có được các hệ số a, b_1, b_2, \dots của mô hình, từ đó xác định được phương trình hồi quy tuyến tính mẫu:

$$\widehat{\text{Release_Price}} = -332.40698 + 5.94176 \text{ Max_Power} + 0.08747 \text{ Memory} \\ - 3.07001 \text{ Memory_Bandwidth}$$

- Hệ số xác định hiệu chỉnh $R^2 = 30.39\%$.

Nhận xét: Đây là một tỉ lệ tương đối thấp, cho thấy các biến độc lập đưa vào phân tích hồi quy giải thích được 30.39% sự biến thiên của biến phụ thuộc, phần còn lại có thể được giải thích bởi phần dư gồm các biến độc lập ngoài mô hình và sai số ngẫu nhiên.

*Kiểm định giả thiết mô hình

Từ kết quả có được về mô hình hồi quy tuyến tính đã xây dựng, ta cần kiểm định lại các giả thiết của mô hình:

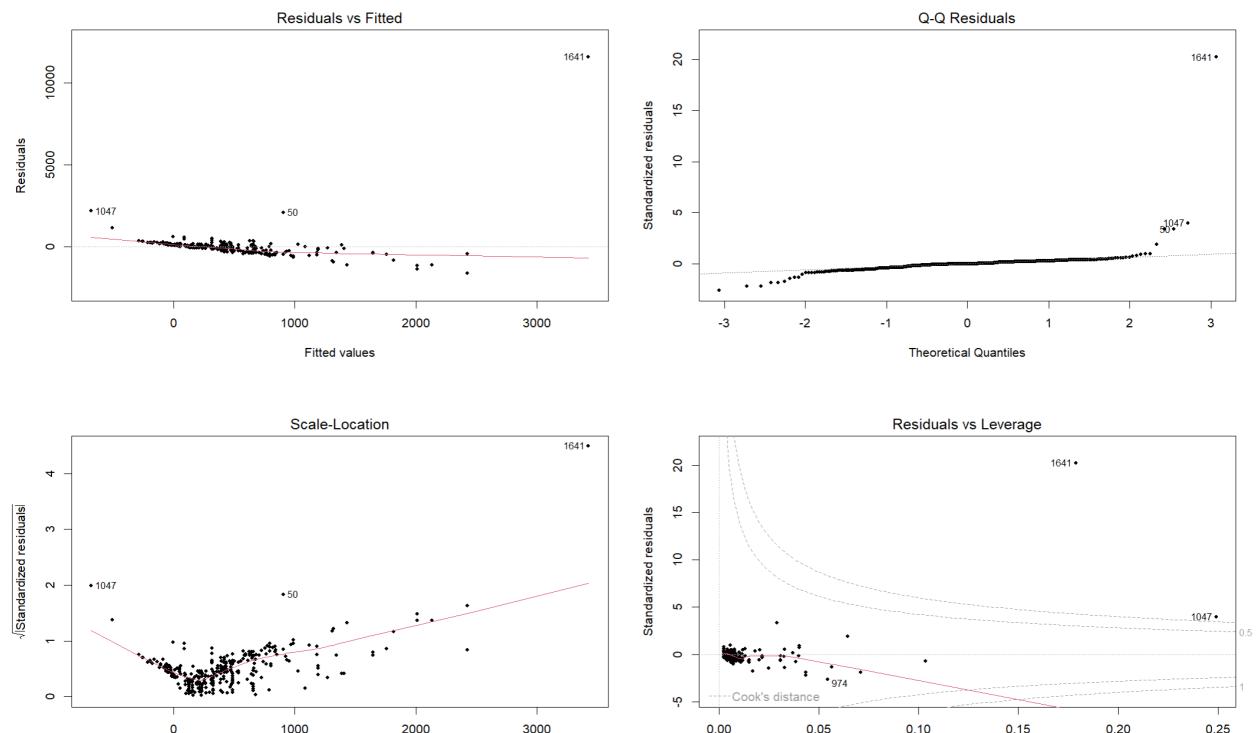
- (1) Tính tuyến tính của dữ liệu.
- (2) Phản dư có phân phối chuẩn.
- (3) Phản dư có trung bình bằng 0.
- (4) Phản dư có phương sai không đổi.
- (5) Sai số ngẫu nhiên có phân phối chuẩn.
- (6) Sai số ngẫu nhiên có kỳ vọng tại mỗi giá trị bằng 0.
- (7) Sai số ngẫu nhiên có phương sai không đổi.
- (8) Giữa các biến độc lập không có mối quan hệ đa cộng tuyến hoàn hảo.

Dùng hàm `plot()` vẽ biểu đồ phản dư để kiểm tra các giả thiết trên cho mô hình đã xây dựng:

```

1 # Thiết lập vị trí các biểu đồ trên cửa sổ
2 par(mfrow = c(2, 2))
3 # Dùng hàm plot() cho kết quả mô hình đã xây dựng
4 plot(lm_result, pch = 20)

```



Hình 5.1: Các biểu đồ nhận được khi thực thi đoạn chương trình trên

Ý nghĩa các biểu đồ:

- **Biểu đồ Residuals vs Fitted:** Dùng để kiểm tra giả thiết tuyến tính của dữ liệu và giả thiết phần dư có trung bình bằng 0. Trục tung biểu thị giá trị của phần dư, trục hoành biểu thị giá trị tiên lượng (\hat{y}_i) của biến phụ thuộc. Nếu đường màu đỏ trên biểu đồ càng có dạng một đường thẳng nằm ngang, điều đó càng chứng tỏ tính tuyến tính của dữ liệu càng cao. Mặc khác, giả thiết phần dư có trung bình bằng 0 thỏa mãn nếu đường màu đỏ gần với đường nét đứt nằm ngang (ứng với phần dư bằng 0) trên biểu đồ.
- **Biểu đồ Normal Q-Q:** Dùng để kiểm tra giả thiết phần dư có phân phối chuẩn. Nếu các điểm thăng dư có xu hướng phân bố trên một đường thẳng thì điều kiện về phân phối chuẩn của phần dư được thỏa.
- **Biểu đồ Scale - Location:** Dùng để kiểm định giả thiết phương sai của phần dư là không đổi. Trục tung là căn bậc hai của phần dư (đã được chuẩn hóa), trục hoành là giá trị tiên lượng (\hat{y}_i) của các biến phụ thuộc. Nếu đường màu đỏ trên đồ thị là đường thẳng nằm ngang và các điểm thăng dư phân tán đều xung quanh đường thẳng này thì giả thiết về phương sai của phần dư được thỏa.
- Ngoài ra, **Biểu đồ Residuals vs Leverage** mặc dù không dùng để kiểm định giả thiết cho mô hình, tuy nhiên biểu đồ này giúp xác định những điểm outliers (các phần tử bất thường có thể là nguyên nhân gây ra sự vi phạm các giả thiết hay làm sai lệch kết quả dự báo của mô hình). Những điểm outliers sẽ cách xa đường màu đỏ trên biểu đồ.

Nhận xét:

- **Biểu đồ Residuals vs Fitted** cho thấy giả thiết (1) về tính tuyến tính của dữ liệu hơi vi phạm. Tuy nhiên giả thiết (3) về trung bình của phần dư có thể coi là thỏa mãn.
- **Biểu đồ Normal Q-Q** cho thấy giả thiết (2) về phần dư có phân phối chuẩn được thỏa mãn.
- **Biểu đồ Scale - Location:** đường màu đỏ có độ dốc, không thẳng, các điểm thăng dư phân tán không đều xung quanh đường thẳng này. Do đó, giả thiết (4) về tính đồng nhất của phương sai đối với mô hình này bị vi phạm.
- **Biểu đồ Residuals vs Leverage** chỉ ra có các quan sát thứ 1047 và 1641 có thể là các điểm outliers, gây ảnh hưởng đến mô hình đã xây dựng.

Để kiểm tra các giả thiết còn lại về sai số ngẫu nhiên và quan hệ đa cộng tuyến giữa các biến độc lập, ta có thể sử dụng các kiểm định có trong R:

- Trích xuất phần dư của mô hình thông qua hàm `resid()`, sau đó dùng Shapiro-test (hàm `shapiro.test()` của R) để kiểm định giả thiết (5). Với giả thuyết không H_0 : sai số ngẫu nhiên có phân phối chuẩn.

¹ # Trích xuất các phần dư từ mô hình đã xây dựng
² `resid <- resid(lm_result)`



```
3 # Dùng Shapiro-test cho mẫu phần dư đã trích để kiểm định về sai số
   ngẫu nhiên
4 shapiro.test(resid)
```

Shapiro-Wilk normality test

```
data: resid
W = 0.32554, p-value < 2.2e-16
```

Bảng 5.10: Kết quả khi thực thi đoạn chương trình trên

Nhận xét: p-value rất nhỏ, chứng tỏ bác bỏ giả thuyết H_0 . Tức giả thiết (5) không thoả.

- Giả thiết (6) có thể được kiểm định thông qua hàm Student-test `t.test()`. Với giả thuyết không H_0 : Kỳ vọng của sai số ngẫu nhiên tại mỗi giá trị bằng 0.

```
1 alpha <- 0.05
2 t.test(resid, mu = 0, conf.level = 1 - alpha)
```

One Sample t-test

```
data: resid
t = 6.7124e-16, df = 456, p-value = 1
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
-57.82733 57.82733
sample estimates:
mean of x
1.975183e-14
```

Bảng 5.11: Kết quả khi thực thi đoạn chương trình trên

Nhận xét: p-value = 1, lớn hơn mức ý nghĩa $\alpha = 5\%$, chứng tỏ chưa bác bỏ giả thuyết H_0 . Tức có thể kết luận giả thiết (6) thoả mãn.

- Giả thiết (7) có thể được kiểm định thông qua hàm `ncvTest()` (Non-constant Variance Score Test) cho trực tiếp mô hình đã xây dựng. Với giả thuyết không H_0 : Phương sai sai số ngẫu nhiên không đổi.

```
1 ncvTest(lm_result)
```



Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 7306.092, Df = 1, p = < 2.22e-16

Bảng 5.12: Kết quả khi thực thi đoạn chương trình trên

Nhận xét: p-value rất nhỏ, chứng tỏ bác bỏ giả thuyết H_0 . Tức giả thiết (7) không thoả.

- Giả thiết (8) về tính đa cộng tuyến giữa các biến độc lập có thể được kiểm định thông qua việc đánh giá **hệ số lạm phát phương sai** (Variance Inflation Factor – VIF). Hệ số này đo lường mối tương quan và cường độ tương quan giữa các biến dự đoán trong mô hình hồi quy. Công thức hệ số lạm phát phương sai của biến ngẫu nhiên X_j :

$$VIF_j = \frac{1}{1 - R_j^2}$$

Trong đó R_j^2 là hệ số xác định từ mô hình hồi quy tuyến tính của X_j theo các biến độc lập còn lại trong mô hình.

Nguyên tắc:

- Nếu $VIF_j = 1$: Không có mối tương quan giữa X_j với các biến độc lập con lại trong mô hình. Tức giả thiết (8) thoả mãn.
- Nếu $1 < VIF_j \leq 5$: Biểu thị mối tương quan vừa phải giữa X_j với các biến độc lập còn lại trong mô hình nhưng vẫn có thể chấp nhận được. Tức có thể cho rằng giả thiết (8) thoả mãn.
- Nếu $5 < VIF_j$: Cho thấy mối tương quan mật thiết giữa X_j với các biến độc lập còn lại trong mô hình. Trong trường hợp này, mô hình không thoả mãn giả thiết (8).

Trong R, ta có thể tính các giá trị VIF_j thông qua hàm `vif()` cho trực tiếp mô hình đã xây dựng.

```
1 vif(lm_result)
```

Max_Power_Value	Memory_Value	Memory_Bandwidth_Value
3.345470	2.878253	6.103723

Bảng 5.13: Kết quả khi thực thi đoạn chương trình trên

Nhận xét: Tồn tại một giá trị VIF của biến **Memory_Bandwidth** là lớn hơn 5, vậy có thể kết luận có sự đa cộng tuyến giữa biến này với các biến độc lập còn lại trong mô hình. Tức giả thiết (8) không thoả.