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

XÁC SUẤT THỐNG KÊ

Bài tập lớn

Phân tích ảnh hưởng giữa thông số kỹ thuật trong chip đồ họa GPUs

Giáo viên hướng dẫn: Thầy Nguyễn Đình Huy
Cô Phan Thị Khánh Vân
Sinh viên thực hiện: **Phó Ngọc Song Khuê - 2252386**
Đỗ Mỹ Quyên - 2252697
Nguyễn Thị Thanh Bình - 2252083
Trần Tường Khang - 2252313

TP. HỒ CHÍ MINH, THÁNG 4/2024

Mục lục

1 Cơ sở lý thuyết:

1.1 Hồi quy

Hồi quy chính là một phương pháp thống kê để thiết lập mối quan hệ giữa một biến phụ thuộc và một nhóm tập hợp các biến độc lập. Mô hình với một biến phụ thuộc với hai hoặc nhiều biến độc lập được gọi là hồi quy bội (hay còn gọi là hồi quy đa biến).

Ví dụ: Tỷ lệ tử vong trẻ em của một quốc gia phụ thuộc vào thu nhập bình quân đầu người, trình độ giáo dục,...

1.1.1 Giới thiệu mô hình hồi quy tuyến tính bội:

Mô hình hồi quy tuyến tính bội có dạng tổng quát như sau:

$$\mathrm{Y} = \beta_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \mathrm{u}$$

Trong đó:

Y: biến phụ thuộc

X_1: biến độc lập

β_1: hệ số tự do (hệ số chặn)

β_i: hệ số hồi quy riêng. β_i đo lường tác động riêng phần của biến X_i lên Y với điều kiện các biến số khác trong mô hình không đổi. Cụ thể hơn, nếu các biến khác trong mô hình không đổi, giá trị kỳ vọng của Y sẽ tăng β_i đơn vị nếu X_i tăng 1 đơn vị u: sai số ngẫu nhiên.

Như vậy, *Hồi quy tuyến tính* là một phương pháp để dự đoán giá trị biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Thuật ngữ tuyến tính dùng để chỉ rằng bản chất của các thông số của tổng thể β_1 và β_i là tuyến tính (bậc nhất). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ: dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số người đã truy cập vào một website nào đó v.v... Bằng dữ liệu thu thập được, ta đi ước lượng hàm hồi quy của tổng thể, đó là ước lượng các tham số của tổng thể: β_1, β_2,..., β_k.

1.1.2 Ước lượng các tham số của mô hình hồi quy tuyến tính bội:

a. Hàm hồi quy tổng thể (PRF - Population Regression Function

Với Y là biến phụ thuộc, X_2, X_3,..., X_k là biến độc lập, Y là ngẫu nhiên và có một phân phối xác suất nào đó. Suy ra: Tồn tại E(Y|X_2, X_3,..., X_k) = giá trị xác định. Do vậy, F(X_2, X_3,..., X_k) = E(Y|X_2, X_3,..., X_k) là hàm hồi quy tổng thể của Y theo X_2, X_3,..., X_k.

Với một cá thể i, tồn tại ($X_{2,i}$, $X_{3,i}$,..., $X_{k,i}$, Y_i)

Ta có: $Y_i \neq F(X_2, X_3, ..., X_k => u_i = Y_i - F$

Do vậy: Y_i = E(Y|X_2, X_3,..., X_k) + u_i

Hồi quy tổng thể PRF:

- Y = E(Y|X) + U
- E(Y|X) = F(X)

b. Hàm hồi quy mẫu (SRF – Sample Regression Function):

Do không biết tổng thể, nên chúng ta không biết giá trị trung bình tổng thể của biến phụ thuộc là đúng ở mức độ nào. Do vậy chúng ta phải dựa vào dữ liệu mẫu để ước lượng.

Trên một mẫu có n cá thể, gọi $\hat{Y} = \hat{F}\ (X_2, X_3, \ldots, X_k)$ là hồi quy mẫu. Với một cá thể mẫu $Y_i \neq \hat{F}(X_{2,i}, X_{3,i}, ..., X_k)$; e_i gọi là phần dư SRF.

Ta có hàm hồi quy mẫu tổng quát được viết dưới dạng như sau:

$$\hat{y} = \hat{\beta}_1 + \hat{\beta}_2 x_{2,i} + \hat{\beta}_3 x_{3,i} + ... + \hat{\beta}_k x_{k,i}$$

Phần dư sinh ra: $e_i = y_i - \hat{y}_i$

Ký hiệu: $\hat{\beta}_m$ là ước lượng của β_m. Chúng ta trông đợi $\hat{\beta}_m$ là ước lượng không chệch của β_m, hơn nữa phải là một ước lượng hiệu quả.

Ước lượng SRF: chọn một phương pháp nào đó để ước lượng các tham số của F qua việc tìm các tham số của $\hat{F}$ và lấy giá trị quan sát của các tham số này làm giá trị xấp xỉ cho tham số của F.

c. Phương pháp bình phương nhỏ nhất (Ordinary Least Squares):

Phương pháp bình phương nhỏ nhất được đưa ra bởi nhà Toán học Đức Carl Friedrich Gauss. Tư tưởng của phương pháp này là cực tiểu tổng bình phương của các phần dư. Do đó có thể nói để có được hồi quy thích hợp nhất, chúng ta chọn các ước lượng có tung độ gốc và độ dốc sao cho phần dư là nhỏ.

- Các giả thiết của phương pháp bình phương nhỏ nhất cho mô hình hồi quy tuyến tính bội:

Phương pháp bình phương nhỏ nhất (OLS) là phương pháp rất đáng tin cậy trong việc ước lượng các tham số của mô hình, tuy nhiên mô hình ước lượng phải thỏa mãn 7 giả thiết. Khi thỏa mãn các giả thiết, ước lượng bình phương nhỏ nhất (OLS) là ước lượng tuyến tính không chệch có hiệu quả nhất trong các ước lượng. Vì thế phương pháp OLS đưa ra ước lượng không chệch tuyến tính tốt nhất (BLUE).

Kết quả này được gọi là Định lý Gauss – Markov, theo lý thuyết này ước lượng OLS là BLUE, nghĩa là trong tất cả các tổ hợp tuyến tính không chệch của Y, ước lượng OLS có phương sai bé nhất. Các giả thiết như sau:

- **Hàm hồi quy là tuyến tính theo các hệ số:**

Điều này có nghĩa là quá trình thực hành hồi quy trên thực tế được miêu tả bởi mối quan hệ dưới dạng:

$$\mathrm{y} = \beta_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + ... + \beta_k x_k + u$$

hoặc mối quan hệ thực tế có thể được viết lại ví dụ như dưới dạng lấy loga cả hai vế.

- **$\mathbf{E}(u_i) = 0$: Kỳ vọng của các yếu tố ngẫu nhiên u_i bằng 0.**

Trung bình tổng thể sai số là bằng 0. Điều này có nghĩa là có một số giá trị sai số mang dấu dương và một số sai số mang dấu âm. Do hàm xem như là đường trung bình nên có thể giả định rằng các sai số ngẫu nhiên trên sẽ bị loại trừ nhau, ở mức trung bình, trong tổng thể.

- **$\mathbf{Var}(u_i) = \sigma^2$: Phương sai bằng nhau và thuần nhất với mọi u_i.**

Tất cả giá trị u được phân phối giống nhau với cùng phương sai σ^2, sao cho:
Var (u_i) = E $(u_i^2) = \sigma^2$

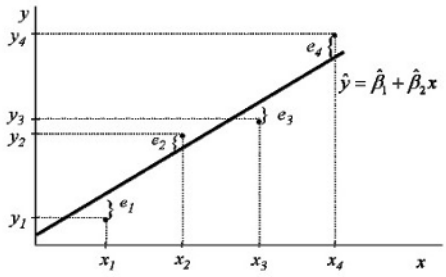
- u_i **phân phối chuẩn**.

Điều này rất quan trọng khi phát sinh khoảng tin cậy và thực hiện kiểm định giả thuyết trong những phạm vi mẫu là nhỏ. Nhưng phạm vi mẫu lớn hơn, điều này trở nên không mấy quan trọng.

- **Giữa các** u_i **thì độc lập với nhau**.

- Ước lượng:

Ta đặt: y_i ký hiệu giá trị thực của biến y tại quan sát i
$\hat{y}_i$ ký hiệu giá trị của hàm hồi quy mẫu



e_i ký hiệu phần dư $y_i - \hat{y}_i$
Do đó cực tiểu hoá $\Sigma(y_i - \hat{y}_i)^2$ sẽ tương đương với cực tiểu Σe_i^2 từ đó tìm ra $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k$
Ta có:

$$\Sigma e_i^2 = \Sigma(y_i - (\hat{\beta}_1 + \hat{\beta}_2 x_{2,i} + \hat{\beta}_3 x_{3,i} + \hat{\beta}_4 x_{4,i} + ... + \hat{\beta}_k x_{k,i}))^2$$

Chúng ta có thiết lập các điều kiện bậc nhất cho phép tính tối thiểu này như sau:

$$\frac{\partial \Sigma e_i^2}{\partial \hat{\beta}_1} = -2\Sigma\left(y_i - (\hat{\beta}_1 + \hat{\beta}_2 x_{2i} + \hat{\beta}_3 x_{3i} + ... + \hat{\beta}_k x_{ki})\right) x_i = 0$$

$$\frac{\partial \Sigma e_i^2}{\partial \hat{\beta}_2} = -2\Sigma\left(y_i - (\hat{\beta}_1 + \hat{\beta}_2 x_{2i} + \hat{\beta}_3 x_{3i} + ... + \hat{\beta}_k x_{ki})\right) x_{2i} = 0$$

...

$$\frac{\partial \Sigma e_i^2}{\partial \hat{\beta}_k} = -2\Sigma\left(y_i - (\hat{\beta}_1 + \hat{\beta}_2 x_{2i} + \hat{\beta}_3 x_{3i} + ... + \hat{\beta}_k x_{ki})\right) x_{ki} = 0$$

Hệ phương trình mà chúng ta có được gọi là hệ phương trình chuẩn của hồi quy mẫu. Chúng ta có thể giải k phương trình chuẩn này để tìm k hệ số chưa biết. $\hat{\beta}_1, \hat{\beta}_2 ..., \hat{\beta}_k$ được gọi là các ước lượng bình phương nhỏ nhất.

d. Độ phù hợp của mô hình

Để có thể biết mô hình giải thích được như thế nào hay bao nhiêu % biến động của biến phụ thuộc, người ta sử dụng R^2
Ta có:

$$\Sigma(y_i - \bar{y})^2 = \Sigma[(y_i - \hat{y}_i) + (\hat{y}_i - \bar{y})]^2 = \Sigma[e_i + (\hat{y}_i - \bar{y})]^2$$

$$= \Sigma e_i^2 + 2\Sigma e_i(\hat{y}_i - \bar{y}) + \Sigma(\hat{y}_i - \bar{y})^2$$

Đặt:

$\Sigma(y_i - \overline{y})^2$: TSS – Total Sum of Squares

$\Sigma(\hat{y}_i - \overline{y})]^2$: ESS – Explained Sum of Squares

Σe_i^2: RSS – Residual Sum of Squares

Do $\Sigma e_i(\hat{y}_i - \overline{y}) = 0$
Ta có thể viết: TSS = ESS + RSS
Ý nghĩa của các thành phần:

- TSS là tổng bình phương của tất cả các sai lệch giữa các giá trị quan sát Yi và giá trị trung bình.
- ESS là tổng bình phương của tất cả các sai lệch giữa các giá trị của biến phụ thuộc Y nhận được từ hàm hồi quy mẫu và giá trị trung bình của chúng. Phần này đo độ chính xác của hàm hồi quy.
- RSS là tổng bình phương của tất cả các sai lệch giữa các giá trị quan sát Y và các giá trị nhận được từ hàm hồi quy.
- TSS được chia thành 2 phần: một phần do ESS và một phần do RSS gây ra.

Từ TTS = ESS + RSS, ta chia cả hai vế cho TSS, ta có:

$$1 = \frac{ESS}{TSS} + \frac{RSS}{TSS} = \frac{\Sigma(\hat{y}_i - \bar{y})^2}{\Sigma(y_i - \bar{y})^2} + \frac{\Sigma e_i^2}{\Sigma(y_i - \bar{y})^2}$$

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS} = \frac{\left(\Sigma(y_i - \bar{y})(y_i - \bar{y})\right)^2}{\left(\Sigma(y_i - \bar{y})^2\right)\left(\Sigma(y_i - \bar{y})^2\right)}$$

Tỷ số giữa tổng biến thiên được giải thích bởi mô hình cho tổng bình phương cần được giải thích được gọi là hệ số xác định, hay là trị thống kê "good of fit". Từ định nghĩa R^2 chúng ta thấy R^2 đo tỷ lệ hay số % của toàn bộ sai lệch Y với giá trị trung bình được giải thích bằng mô hình. Khi đó người ta sử dụng R^2 để đo sự phù hợp của hàm hồi quy:

$$0 \leq R^2 \leq 1.$$

- R^2 cao nghĩa là mô hình ước lượng được giải thích được một mức độ cao biến động của biến phụ thuộc.

- Nếu $R^2 = 1$, nghĩa là đường hồi quy giải thích 100% thay đổi của y.
- Nếu $R^2 = 0$, nghĩa là mô hình không đưa ra thông tin nào về sự thay đổi của biến phụ thuộc y.

Trong mô hình hồi quy đa biến tỷ lệ của toàn bộ sự khác biệt của biến y do tất cả các biến x_2 và x_3 gây ra được gọi là hệ số xác định bội, ký hiệu là R^2

$$R^2 = \frac{\hat{\beta}_2 \Sigma(y_i - \bar{y})(x_{2i} - \bar{x}_2) + \hat{\beta}_3 \Sigma(y_i - \bar{y})(x_{3i} - \bar{x}_3)}{\Sigma(y_i - \bar{y})^2} = 1 - \frac{\Sigma e_i^2}{\Sigma(y_i - \bar{y})^2}$$

e. Khoảng tin cậy và kiểm định các hệ số hồi quy

- Ước lượng khoảng tin cậy đối với các hệ số hồi quy

Mục đích của phân tích hồi quy không phải chỉ suy đoán về β_1, β_2,..., β_k mà còn phải kiểm tra bản chất sự phụ thuộc. Do vậy cần phải biết phân bố xác suất của β_1, β_2,..., β_k. Các phân bố này phụ thuộc vào phân bố của các u_i.
Với các giả thiết OLS, u_i có phân phối N (θ, σ^2). Các hệ số ước lượng tuân theo phân phối chuẩn:

$$\hat{\beta}_j \sim N(\beta_j, Se(\hat{\beta}_j))$$

$$\frac{\hat{\beta}_j - \beta_j}{Se(\hat{\beta}_j)} \sim T(n-k)$$

Ước lượng phương sai sai số dựa vào các phần dư bình phương tối thiểu. Trong đó k là số hệ số có trong phương trình hồi quy đa biến:

$$\hat{\sigma}^2 = \frac{\Sigma e_i^2}{n - k}$$

Ước lượng 2 phía, ta tìm được $t_{\frac{\alpha}{2}}(n-3) = 1 - \alpha$ thoả mãn:

$$P(-t_{\frac{\alpha}{2}}(n-3)) \leq \frac{\hat{\beta}_j - \beta_j}{Se(\hat{\beta}_j)} \leq P(t_{\frac{\alpha}{2}}(n-3))$$

Khoảng tin cậy 1 - α của β_j là:

$$\left[\hat{\beta}_j - t_{\frac{\alpha}{2}}(n-3)Se(\hat{\beta}_j)\right] ; \left[\hat{\beta}_j + t_{\frac{\alpha}{2}}(n-3)Se(\hat{\beta}_j)\right]$$

- Kiểm định giả thiết đối với β_j

Kiểm định ý nghĩa thống kê của các hệ số hồi quy có ý nghĩa hay không: kiểm định rằng biến giải thích có thực sự ảnh hưởng đến biến phụ thuộc hay không. Nói cách khác là hệ số hồi quy có ý nghĩa thống kê hay không.
Có thể đưa ra giả thiết nào đó đối với β_j, chẳng hạn $\beta_j = \beta_j^*$. Nếu giả thiết này đúng thì:

$$T = \frac{\hat{\beta}_j - \beta_j}{Se(\hat{\beta}_j)} \sim T(n-k)$$

Ta có bảng sau:

Loại giả thiết	Giả thiết H_0	Giả thiết đối H_1	Miền bác bỏ
Hai phía	$\beta_i = \beta_i^*$	$\beta_i \neq \beta_i^*$	$\|t\| > t_{\alpha/2}$ (n-k)
Phía phải	$\beta_i \leq \beta_i^*$	$\beta_i > \beta_i^*$	$t > t_\alpha$ (n-k)
Phía trái	$\beta_i \geq \beta_i^*$	$\beta_i < \beta_i^*$	$t < -t_\alpha$ (n-k)

Kiểm định β_j:
$H_0 : \beta_j = 0 <=> x_j$ không tác động
$H_1 : \beta_j \neq 0 <=> x_j$ có tác động
$\beta_j < 0 <=> x_j$ có tác động ngược
$\beta_j > 0 <=> x_j$ có tác động thuận

f. Kiểm định ý nghĩa của mô hình

Trong mô hình hồi quy đa biến, giả thuyết "không" cho rằng mô hình không có ý nghĩa được hiểu là tất cả các hệ số hồi quy riêng đều bằng 0.

Ứng dụng kiểm định Wald (thường được gọi là kiểm định F) được tiến hành cụ thể như sau:
Bước 1: Giả thuyết "không" là H_0: $\beta_2 = \beta_3 = \ldots = \beta_k = 0$.
Giả thuyết đối là H_1: "có ít nhất một trong những giá trị β khác không".
Bước 2: Trước tiên hồi quy Y theo một số hạng không đổi và $X_2, X_3, \ldots, X_k$, sau đó tính tổng bình phương sai số RSS_U, RSS_R. Phân phối F là tỷ số của hai biến ngẫu nhiên phân phối khi bình phương độc lập. Điều này cho ta trị thống kê:

$$F_c = \frac{[RSS_R - RSS_U]/(k-m)}{RSS_U/(n-k)} \sim F(\alpha, k-m, n-k)$$

Vì H_0: $\beta_2 = \beta_3 = \ldots = \beta_k = 0$, nhận thấy rằng trị thống kê kiểm định đối với giả thuyết này sẽ là:

$$F_c = \frac{ESS/(k-1)}{RSS/(n-k)} \sim F(\alpha, k-1, n-k)$$

Bước 3: Tra số liệu trong bảng F tương ứng với bậc tự do (k – 1) cho tử số và (n – k) cho mẫu số, và với mức ý nghĩa α cho trước.
Bước 4: Bác bỏ giả thuyết H_0 ở mức ý nghĩa α nếu $F_c >$ F(α, k-1, n-k). Đối với phương pháp giá trị p, tính giá trị p = P(F $> F_c|H_0$) và bác bỏ giả thuyết H_0 nếu p $< \alpha$.

1.2 Phân tích phương sai

Mục tiêu của phân tích phương sai (Analysis of Variance ANOVA) là so sánh trung bình của nhiều nhóm (tổng thể) dựa trên các trị trung bình của các mẫu quan sát từ các nhóm này, và thông qua kiểm định giả thuyết để kết luận về sự bằng nhau của các trung bình tổng thể này. Trong nghiên cứu, phân tích phương sai được dùng như một công cụ để xem xét ảnh hưởng của một yếu tố nguyên nhân (định tính) đến một yếu tố kết quả (định lượng). Trong chương này chúng ta đề cập đến hai mô hình phân tích phương sai: phân tích phương sai một yếu tố và hai yếu tố. Cụm từ yếu tố ở đây ám chỉ số lượng yếu tố nguyên nhân ảnh hưởng đến yếu tố kết quả đang nghiên cứu.

1.2.1 Phân tích phương sai một yếu tố

Phân tích phương sai một yếu tố (One-way ANOVA) là phân tích ảnh hưởng của một yếu tố nguyên nhân (dạng biến định tính) ảnh hưởng đến một yếu tố kết quả (dạng biến định lượng) đang nghiên cứu. Ta đi vào lý thuyết như sau:
a.Trường hợp k tổng thể có phân phối chuẩn và phương sai bằng nhau
Giả sử rằng chúng ta muốn so sánh trung bình của k tổng thể dựa trên những mẫu ngẫu nhiên độc lập gồm $n_1, n_2, ..., n_k$ quan sát từ k tổng thể này. Cần ghi nhớ ba giả định sau đây về các nhóm tổng thể được tiến hành phân tích ANOVA:

- Các tổng thể này có phân phối bình thường;
- Các phương sai tổng thể bằng nhau;
- Các quan sát được lấy mẫu là độc lập nhau.

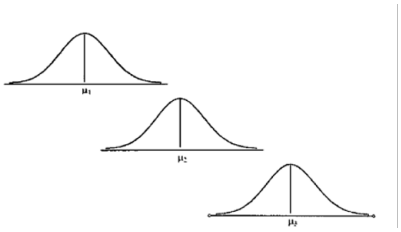
Nếu trung bình của các tổng thể được kí hiệu là $\mu_1, \mu_2, ..., \mu_k$ thì khi các giả định trên được đáp ứng, mô hình phân tích phương sai một yếu tố ảnh hưởng được mô tả dưới dạng kiểm định giả thuyết như sau:

$$H_0 : \mu_1 = \mu_2 = ... = \mu_k$$

Giả thuyết H_0 cho rằng trung bình của k tổng thể đầu bằng nhau (về mặt nghiên cứu liên hệ thì giả thuyết này cho rằng yếu tố nguyên nhân không có tác động gì đến vấn đề ta đang nghiên cứu). Và giả thuyết đối là

H_1: Tồn tại ít nhất một cặp trung bình tổng thể khác nhau

Hai giả định đầu tiên để tiến hành phân tích phương sai được mô tả như hình dưới đây, bạn thấy ba tổng thể đều có phân phối bình thường với mức độ phân tán tương đối giống nhau, nhưng ba vị trí chênh lệch của chúng cho thấy ba trị trung bình khác nhau. Rõ ràng là nếu bạn thực sự có các giá trị của 3 tổng thể và biểu diễn được phân phối của chúng như hình dưới thì bạn không cần phải làm gì nữa mà kết luận được ngay là bạn bác bỏ H_0 hay 3 tổng thể này có trị trung bình khác nhau.



Nhưng bạn chỉ có mẫu đại diện được quan sát, nên để kiểm định giả thuyết này, ta thực hiện các bước sau:
Bước 1: Tính các trung bình mẫu của các nhóm (xem như đại diện của các tổng thể)

Trước hết ta xem cách tính các trung bình mẫu từ những quan sát của k mẫu ngẫu nhiên độc lập (kí hiệu $\overline{x_1}, \overline{x_2}, \ldots, \overline{x_k}$) và trung bình chung của k mẫu quan sát (kí hiệu $\bar{x}$) từ trường hợp tổng quát như sau:

Tổng thể			
1	2	…	K
x_{11}	x_{21}	…	x_{k1}
x_{12}	x_{22}	…	x_{k2} (x_{ij})
…	…	…	…
x_{1n_1}	x_{2n_2}	…	x_{kn_k}

Tính trung bình mẫu của từng nhóm $\overline{x_1}, \overline{x_2}, \ldots, \overline{x_k}$ theo công thức

$$\overline{x_i} = \frac{\sum_{j=1}^{n_i} x_{ij}}{n_i} \ (i = 1,2, \ldots, k)$$

Và trung bình chung của k mẫu (trung bình chung của toàn bộ mẫu khảo sát):

$$\bar{x} = \frac{\sum_{i=1}^{k} n_i \overline{x_i}}{\sum_{i=1}^{k} n_i}$$

Dĩ nhiên bạn có thể tính trung bình chung của k mẫu theo cách khác là: cộng tất cả các x_{ij} trên Bảng 1 lại rồi đem chia cho $\sum n_i$ với (i=1, 2,…,k). Kết quả là như nhau

Bước 2: Tính các tổng các chênh lệch bình phương (hay gọi tắt là tổng bình phương) Tính tổng các chênh lệch bình phương trong nội bộ nhóm SSW1 và tổng các chênh lệch bình phương giữa các nhóm SSG

Tổng các chênh lệch bình phương trong nội bộ nhóm (SSW) được tính bằng cách cộng các chênh lệch bình phương giữa các giá trị quan sát với trung bình mẫu của từng nhóm, rồi sau đó lại tính tổng cộng kết quả tất cả các nhóm lại. SSW phản ánh phần biến thiên của yếu tố kết quả do ảnh hưởng của các yếu tố khác, chứ không phải do yếu tố nguyên nhân đang nghiên cứu (là yếu tố dùng để phân biệt các tổng thể/ nhóm đang so sánh)

Tổng các chênh lệch bình phương của từng nhóm dược tính theo công thức:

$$\text{Nhóm 1: } SS_1 = \sum_{j=1}^{n_1} (x_{1j} - \overline{x_1})^2$$

$$\text{Nhóm 2: } SS_2 = \sum_{j=1}^{n_2} (x_{2j} - \overline{x_2})^2$$

Tương tự như vậy ta tính cho đến nhóm thứ k được SSk. Vậy tổng các chênh lệch bình phương trong nội bộ các nhóm được tính như sau:

$$SSW = SS_1 + SS_2 + \ldots + SS_k$$

Hay viết tổng quát theo công thức ta có:

$$SSW = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \bar{x_i})^2$$

Tổng các chênh lệch bình phương giữa các nhóm (SSG) được tính bằng cách cộng các chênh lệch được lấy bình phương giữa các trung bình mẫu của từng nhóm với trung bình chung của k nhóm (các chênh lệch này đều được nhận thêm với số quan sát tương ứng của từng nhóm). SSG phản ánh phần biến thiên của yếu tố kết quả do ảnh hưởng của yếu tố nguyên nhân đang nghiên cứu.

$$SSG = \sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})^2$$

Tổng các chênh lệch bình phương toàn bộ SST được tính bằng cách cộng tổng các chênh lệch đã lấy bình phương giữa từng giá trị quan sát của toàn bộ mẫu nghiên cứu (x_{ij}) với trung bình chung toàn bộ ($\overline{x}$) SST phản ánh biến thiên của yếu tố kết quả do ảnh hưởng của tất cả các nguyên nhân.

$$SST = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \bar{x})^2$$

Có thể dễ dàng chứng minh là tổng các chênh lệch bình phương toàn bộ bằng tổng cộng tổng các chênh lệch bình phương trong nội bộ các nhóm và tổng các chênh lệch bình phương giữa các nhóm.

$$\text{SST} = \text{SSW} + \text{SSG}$$

Như vậy công thức trên cho thấy, SST là toàn bộ biến thiên của yếu tố kết quả đã được phân tích thành 2 phần: phần biến thiên do yếu tố đang nghiên cứu tạo ra (SSG) và phần biến thiên còn lại do các yếu tố khác không nghiên cứu ở đây tạo ra (SSW). Nếu phần biến thiên do yếu tố nguyên nhân đang xét tạo ra căng "đáng kể" so với phần biến thiên do các yếu tố khác không xét tạo ra, thì chúng ta càng có cơ sở để bác bỏ H0 và kết luận là yếu tố nguyên nhân đang nghiên cứu ảnh hưởng có ý nghĩa đến yếu tố kết quả.
Bước 3: Tính các phương sai (là trung bình của các chênh lệch bình phương)
Các phương sai được tính bằng cách lấy các tổng các chênh lệch bình phương chia cho bậc tự do tương ứng.
Tình phương sai trong nội bộ nhóm (MSW) bằng cách lấy tổng các chênh lệch bình phương trong nội bộ các nhóm (SSW) chia cho bậc tự do tương ứng là n-k (n là số quan sát, k là số nhóm so sánh). MSW là ước lượng phần biến thiên của yếu tố kết quả do các yếu tố khác gây ra (hay giải thích)

$$\text{MSW} = \frac{SSW}{n-k}$$

Tính phương sai giữa các nhóm (MSG) bằng cách lấy tổng các chênh lệch bình phương giữa các nhóm chia cho bậc tự do tương ứng là k – 1. MSG là ước lượng phần biến thiên của yếu tố kết quả do yếu tố nguyên nhân đang nghiên cứu gây ra (hay giải thích được).

$$\text{MSG} = \frac{SSG}{k-1}$$

Bước 4: Kiểm định giả thuyết
Giả thuyết về sự bằng nhau của k trung bình tổng thể được quyết định dựa trên tỉ số của hai phương sai: phương sai giữa các nhóm (MSG) và phương sai trong nội bộ nhóm (MSW), TI số này được gọi là tỷ số F vì nó tuân theo qui luật Fisher– Snedecor với bậc tự do là k - 1 ở tử số và n - k ở mẫu số F = $\frac{MSG}{MSW}$
Ta bác bỏ giả thuyết H_0cho rằng trị trung bình của k tổng thể bằng nhau khi:

$$F > F_{(k-1;n-k);\alpha}$$

$F_{(k-1;n-k);\alpha}$ là giá trị giới hạn tra từ bảng tra số 8 với bậc tự do tra theo cột số k-1 và hàng n-k, nhớ chọn bảng có mức ý nghĩa phù hợp.
Sau đây là dạng bảng kết quả tổng quát của ANOVA khi phân tích bằng chương trình Excel hay SPSS.
Bảng gốc bằng tiếng Anh:

Source of Variation	Sum of squares (SS)	Degree of Freedom (df)	Mean Squares (MS)	F ratio
Between – groups	SSG	k - 1	$MSG = \frac{SSG}{k-1}$	$F = \frac{MSG}{MSW}$
Within – groups	SSW	n- k	$MS = \frac{SSW}{n-k}$	
Total	SST	n - 1		

Tạm dịch sang tiếng Việt:

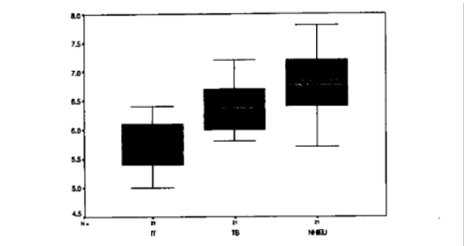
Nguồn biến thiên	Tổng chênh lệch bình phương (SS)	Bậc tự do (df)	Phương sai (MS)	Tỉ số F
Giữa các nhóm	SSG	k - 1	$MSG = \frac{SSG}{k-1}$	$F = \frac{MSG}{MSW}$
Trong nội bộ các nhóm	SSW	n - k	$MSW = \frac{SSW}{k-1}$	
Toàn bộ	SST	n - 1		

Ý nghĩa của công thức và logic của các tính toán trong bảng trên cần được hiểu rõ để có thể vận dụng và giải thích các kết quả phân tích một cách súc tích.

1.2.2 Kiểm tra các giả định của phân tích phương sai

Chúng ta có thể kiểm tra nhanh các giả định này bằng đồ thị. Histogram là phương pháp tốt nhất để kiểm tra giả định về phân phối bình thường của dữ liệu nhưng nó đòi hỏi một số lượng quan sát khá lớn. Biểu đồ thân lá hay biểu đồ hộp và râu là một thay thế tốt trong tình huống số quan sát ít hơn. Nếu công cụ đồ thị cho thấy tập dữ liệu mẫu khá phù hợp với phân phối bình

thường đã thỏa mãn. Hình dưới mô tả biểu đồ hộp râu cho tập dữ liệu mẫu về ba nhóm sinh viên trong tập dữ liệu của chúng ta. Đồ thị cho thấy ngoại trừ nhóm có thời gian tự học TB có hình dáng phân phối của dữ liệu hơi lệch sang trái, còn hai nhóm còn lại có phân phối khá cân đối. Với số quan sát không nhiều thì biểu hiện như thế này của dữ liệu là khả quan và có thể chấp nhận được.
Để khảo sát giả định bằng nhau của phương sai, biểu đồ hộp và râu cũng cho cảm nhận ban đầu nhanh chóng, với ba biểu đồ này, mức độ phân tán của dữ liệu trong trong mỗi tập dữ liệu mẫu không khác biệt nhau nhiều.



Một phương pháp kiểm định tham số chắc chắn hơn cho giả định phương sai bằng nhau là kiểm định Levene về phương sai của các tổng thể. Kiểm định này xuất phát từ giả thuyết sau.
$H_0 : \sigma_1^2 = \sigma_2^2 = ... = \sigma_k^2$
H_1 : Không phải tất cả các phương sai đều bằng nhau
Để quyết định chấp nhận hay bác bỏ H_0 ta tính toán giá trị kiểm định F theo công thức:

$$F_{max} = \frac{s^2_{max}}{s^2_{min}}$$

Trong đó s^2_{max} là phương sai lớn nhất trong các nhóm nghiên cứu và s^2_{min} là phương sai nhỏ nhất trong các nhóm nghiên cứu.
Giá trị F tính được được đem so sánh với giá trị $F_{((k;df);\alpha)}$ tra được từ bảng phân phối Hartley F_max (là bảng số 5 trong phần phụ lục). Trong đó k là số nhóm so sánh, bậc tự do df tính theo công thức df $= (\overline{n} - 1)$. Trong tình huống các nhóm n_i khác nhau thì $\overline{n} = \frac{\Sigma n_i}{k}$ (chú ý là nếu kết quả tính $\overline{n}$ là số thập phân thì ta lấy phần nguyên).
Quy tắc quyết định:
$F_{max} > F_{(k;df);\alpha}$ thì ta bác bỏ H_0 cho rằng phương sai bằng nhau và ngược lại.
Nếu chúng ta không chắc chắn về các giả định hoặc nếu kết quả kiểm định cho thấy các giả định hoặc nếu kết quả kiểm định cho thấy các giả định không được thỏa mãn thì một phương pháp kiểm định thay thế cho ANOVA là phương pháp kiểm định phi tham số Krusksl-Wallis sẽ được áp dụng. Tuy nhiên trong ví dụ này ở đây, ta có thể xem các giả định để tiến hành phân tích phương sai đã được thỏa mãn.

2 Tiền xử lí số liệu

2.1 Nhập dữ liệu

```
# Doc du lieu tu file All_GPUs.csv
All_GPUs=read.csv("D:\\BTL_XSTK\\All_GPUs.csv", na.strings = c("","N/A","\n-","\n","\n- "),
     header = TRUE)
```

Được kết quả như hình:

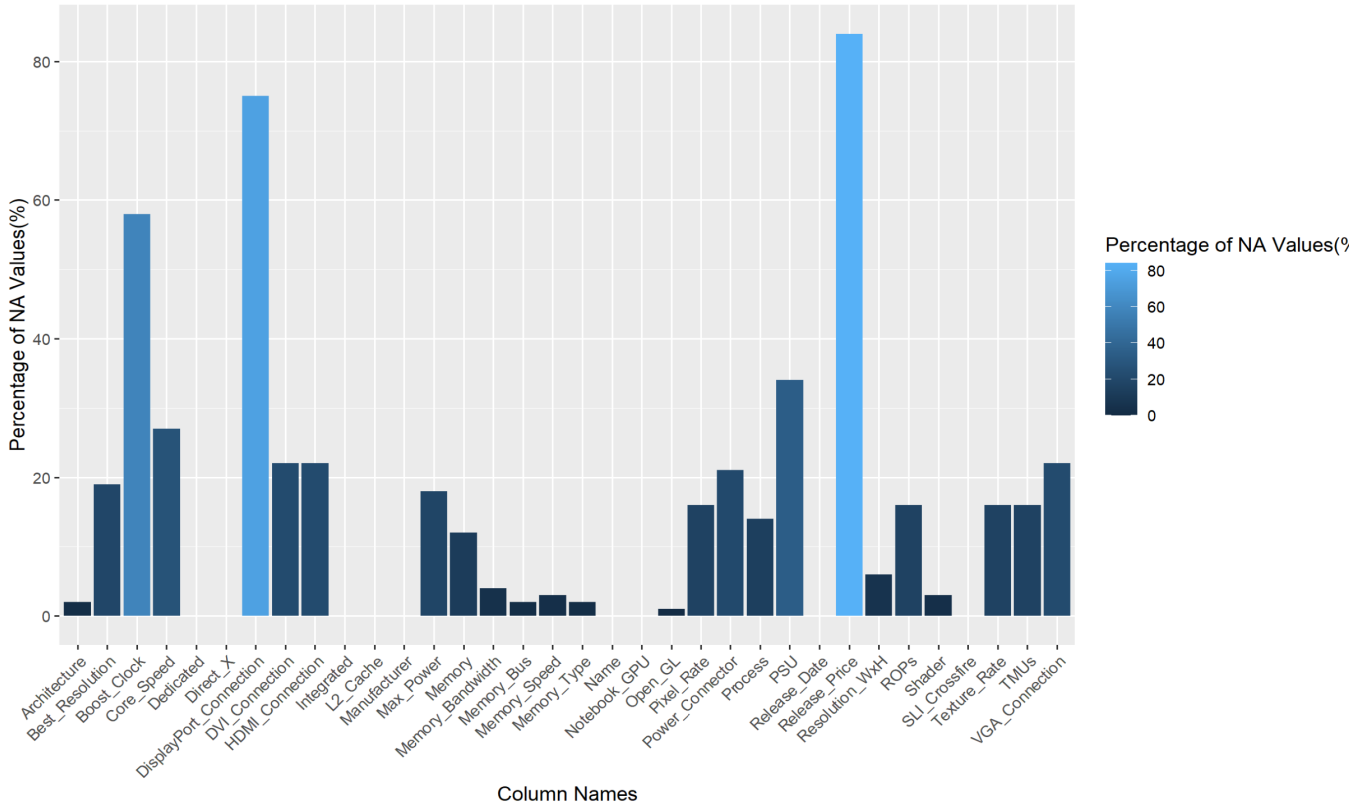
	Architecture	Best_Resolution	Boost_Clock	Core_Speed	DVI_Connection	Dedicated	Direct_X	DisplayP
1	Tesla G92b	NA	NA	738 MHz	2	Yes	DX 10.0	
2	R600 XT	1366 x 768	NA	NA	2	Yes	DX 10	
3	R600 PRO	1366 x 768	NA	NA	2	Yes	DX 10	
4	RV630	1024 x 768	NA	NA	2	Yes	DX 10	
5	RV630	1024 x 768	NA	NA	2	Yes	DX 10	
6	RV630	1024 x 768	NA	NA	2	Yes	DX 10	
7	R700 RV790 XT	1920 x 1080	NA	870 MHz	1	Yes	DX 10.1	
8	R600 GT	1024 x 768	NA	NA	2	Yes	DX 10	
9	Pitcairn XT GL	1920 x 1080	NA	NA	0	Yes	DX 11.2	
10	RV100	NA	NA	NA	NA	Yes	DX 7	
11	NV28GL A2	NA	NA	NA	2	Yes	DX 8.1	
12	Fermi GF110	1920 x 1080	NA	650 MHz	2	Yes	DX 12.0	

Trong đó:

- Lệnh "read.csv" Đọc dữ liệu từ file All_GPUs.csv rồi lưu vào biến All_GPUs
- Lệnh "na.strings =..." là chuyển đổi các giá trị trống thành giá trị không hợp lệ (NA)
- Lệnh "header=TRUE" để xác định dòng đầu tiên được dùng làm tiêu đề cho cột dữ liệu.

2.2 Chọn bộ dữ liệu

Để dự đoán các thông số kĩ thuật ảnh hưởng đến bộ xử lí đồ họa, nhóm quyết định thống kê các giá trị không hợp lệ (NA) ứng với mỗi cột thông số để xác định xem thông số nào nhiều dữ liệu khiếm khuyết.
Kết quả



Các cột thông số có dữ liệu kiểu kí tự, phần trăm khiếm khuyết $<5\%$ và $>50\%$ sẽ bị loại bỏ. Tuy nhiên theo thực tế việc đánh giá hiệu năng của GPU phụ thuộc nhiều vào giá trị Memory Bus, Memory Bandwidth, Memory Speed vì vậy nhóm quyết định chọn Rcác thông số kỹ thuật cần thiết là **Core_Speed, Max_Power, Memory, Memory_Bandwidth, Memory_Bus, Memory_Speed, Process, Pixel_Rate** để nghiên cứu mối tương quan giữa các thông số kỹ thuật được chọn.

```
#Chon bo du lieu can thiet
new_GPU_data=All_GPUs[,c("Core_Speed","Max_Power","Memory","Memory_Bandwidth","Memory_Bus",
"Memory_Speed","Process","Pixel_Rate")]
```

	Core_Speed	Max_Power	Memory	Memory_Bandwidth	Memory_Bus	Memory_Speed	Process	Pixel_Rate
1	738 MHz	141 Watts	1024 MB	64GB/sec	256 Bit	1000 MHz	55nm	12 GPixel/s
2	NA	215 Watts	512 MB	106GB/sec	512 Bit	828 MHz	80nm	12 GPixel/s
3	NA	200 Watts	512 MB	51.2GB/sec	256 Bit	800 MHz	80nm	10 GPixel/s
4	NA	NA	256 MB	36.8GB/sec	128 Bit	1150 MHz	65nm	3 GPixel/s
5	NA	45 Watts	256 MB	22.4GB/sec	128 Bit	700 MHz	65nm	3 GPixel/s
6	NA	50 Watts	256 MB	35.2GB/sec	128 Bit	1100 MHz	65nm	3 GPixel/s
7	870 MHz	190 Watts	2048 MB	134.4GB/sec	256 Bit	1050 MHz	55nm	14 GPixel/s
8	NA	150 Watts	256 MB	51.2GB/sec	256 Bit	800 MHz	80nm	7 GPixel/s
9	NA	150 Watts	2048 MB	160GB/sec	256 Bit	1250 MHz	28nm	25 GPixel/s
10	NA	32 Watts	64 MB	2.9GB/sec	64 Bit	366 MHz	NA	NA
11	NA	NA	128 MB	5.2GB/sec	128 Bit	325 MHz	150nm	1 GPixel/s

2.3 Làm sạch dữ liệu

Sau khi đã chọn được bộ dữ liệu:

- Xóa các đơn vị của thông số

```
 #Dua cac du lieu ve dang chuan
new_GPU_data$Core_Speed=as.numeric(gsub("MHz","",new_GPU_data$Core_Speed))
new_GPU_data$Max_Power=as.numeric(gsub("Watts","",new_GPU_data$Max_Power))
new_GPU_data$Memory=as.numeric(gsub("MB","",new_GPU_data$Memory))
new_GPU_data$Memory_Bus=as.numeric(gsub("Bit","",new_GPU_data$Memory_Bus))
new_GPU_data$Memory_Speed=as.numeric(gsub("MHz","",new_GPU_data$Memory_Speed))
new_GPU_data$Process=as.numeric(gsub("nm","",new_GPU_data$Process))
new_GPU_data$Pixel_Rate=as.numeric(gsub("GPixel/s","",new_GPU_data$Pixel_Rate))

#Memory_Bandwidth bi sai don vi chuan, chuyen tu MB/s ve GB/s
temp = grep("MB/sec",new_GPU_data$Memory_Bandwidth)
for(i in temp) new_GPU_data$Memory_Bandwidth[i] =
     as.numeric(gsub("MB/sec","",new_GPU_data$Memory_Bandwidth[i]))/1024
temp = grep("GB/sec",new_GPU_data$Memory_Bandwidth)
for (i in temp) new_GPU_data$Memory_Bandwidth[i] =
     as.numeric(gsub("GB/sec","",new_GPU_data$Memory_Bandwidth[i]))
new_GPU_data$Memory_Bandwidth =
     as.numeric(gsub("GB/sec","",new_GPU_data$Memory_Bandwidth))
```

	Core_Speed	Max_Power	Memory	Memory_Bandwidth	Memory_Bus	Memory_Speed	Process	Pixel_Rate
1	738	141	1024	64.0	256	1000	55	12
2	NA	215	512	106.0	512	828	80	12
3	NA	200	512	51.2	256	800	80	10
4	NA	NA	256	36.8	128	1150	65	3
5	NA	45	256	22.4	128	700	65	3
6	NA	50	256	35.2	128	1100	65	3
7	870	190	2048	134.4	256	1050	55	14
8	NA	150	256	51.2	256	800	80	7
9	NA	150	2048	160.0	256	1250	28	25
10	NA	32	64	2.9	64	366	NA	NA
11	NA	NA	128	5.2	128	325	150	1

- Xử lí các giá trị NA. Thông số nào có phần trăm NA >5% thì dùng giá trị trung bình (mean) để thay thế. Nhỏ hơn 5% thì xóa hàng

```
#Thay the NA thanh gia tri trung binh cua cot du lieu do
new_GPU_data$Core_Speed=ifelse(is.na(new_GPU_data$Core_Speed),
     mean(new_GPU_data$Core_Speed,na.rm = TRUE),
                            new_GPU_data$Core_Speed)
new_GPU_data$Max_Power=ifelse(is.na(new_GPU_data$Max_Power),
     mean(new_GPU_data$Max_Power,na.rm = TRUE),
                           new_GPU_data$Max_Power)
new_GPU_data$Memory=ifelse(is.na(new_GPU_data$Memory), mean(new_GPU_data$Memory,na.rm =
     TRUE),
                         new_GPU_data$Memory)
new_GPU_data$Process=ifelse(is.na(new_GPU_data$Process),
     mean(new_GPU_data$Process,na.rm = TRUE),
                          new_GPU_data$Process)
new_GPU_data$Pixel_Rate=ifelse(is.na(new_GPU_data$Pixel_Rate),
     mean(new_GPU_data$Pixel_Rate,na.rm = TRUE),
                            new_GPU_data$Pixel_Rate)

#Xoa lun hang co NA neu %NA < 5%
new_GPU_data=new_GPU_data[!is.na(new_GPU_data$Memory_Speed),]
new_GPU_data=new_GPU_data[!is.na(new_GPU_data$Memory_Bandwidth),]
new_GPU_data=new_GPU_data[!is.na(new_GPU_data$Memory_Bus),]
```

	Core_Speed	Max_Power	Memory	Memory_Bandwidth	Memory_Bus	Memory_Speed	Process	Pixel_Rate
1	738.0000	141.0000	1024.000	64.0	256	1000	55.00000	12.00000
2	946.8939	215.0000	512.000	106.0	512	828	80.00000	12.00000
3	946.8939	200.0000	512.000	51.2	256	800	80.00000	10.00000
4	946.8939	125.5987	256.000	36.8	128	1150	65.00000	3.00000
5	946.8939	45.0000	256.000	22.4	128	700	65.00000	3.00000
6	946.8939	50.0000	256.000	35.2	128	1100	65.00000	3.00000
7	870.0000	190.0000	2048.000	134.4	256	1050	55.00000	14.00000
8	946.8939	150.0000	256.000	51.2	256	800	80.00000	7.00000
9	946.8939	150.0000	2048.000	160.0	256	1250	28.00000	25.00000
10	946.8939	32.0000	64.000	2.9	64	366	31.89704	34.96541
11	946.8939	125.5987	128.000	5.2	128	325	150.00000	1.00000
12	650.0000	250.0000	6144.000	177.6	384	925	40.00000	31.00000

Kiểm tra sau khi thực hiện

```
freq.na(new_GPU_data) #kiem tra con NA hay khong
```

```
                missing %
Core_Speed            0 0
Max_Power             0 0
Memory                0 0
Memory_Bandwidth      0 0
Memory_Bus            0 0
Memory_Speed          0 0
Process               0 0
Pixel_Rate            0 0
```

3 Thống kê mô tả

3.1 Thông tin tổng quát về các biến dữ liệu

```
> summary(new_GPU_data)
   Core_Speed       Max_Power          Memory      Memory_Bandwidth      Memory_Bus
 Min.   : 100.0   Min.   :  1.0   Min.   :   16   Min.   :   0.7812   Min.   :  32.0
 1st Qu.: 850.0   1st Qu.: 60.0   1st Qu.: 1024   1st Qu.:  28.8000   1st Qu.: 128.0
 Median : 946.9   Median :125.6   Median : 2048   Median : 105.8000   Median : 128.0
 Mean   : 946.5   Mean   :126.7   Mean   : 2885   Mean   : 137.1846   Mean   : 207.2
 3rd Qu.:1050.0   3rd Qu.:150.0   3rd Qu.: 4096   3rd Qu.: 194.5000   3rd Qu.: 256.0
 Max.   :1784.0   Max.   :780.0   Max.   :32000   Max.   :1280.0000   Max.   :8192.0
  Memory_Speed      Process         Pixel_Rate
 Min.   : 100   Min.   : 14.00   Min.   :  1.00
 1st Qu.: 800   1st Qu.: 28.00   1st Qu.: 13.00
 Median :1150   Median : 28.00   Median : 31.00
 Mean   :1176   Mean   : 31.75   Mean   : 35.15
 3rd Qu.:1502   3rd Qu.: 40.00   3rd Qu.: 40.00
 Max.   :2127   Max.   :150.00   Max.   :260.00
```

- **Đối với biến Core Speed**
  - Min = 100 : Giá trị nhỏ nhất là 100 MHz
  - Q1 = 850 : Có 75% giá trị lớn hơn 850 MHz
  - Median = 946.9 : Có 50% giá trị lớn hơn 946.9 MHz
  - Mean = 946.5 : Giá trị trung bình của mẫu là 946.5 MHz
  - Q3 = 1050 : Có 25% giá trị lớn 1050 MHz
  - Max = 1784 : Giá trị lớn nhất là 1784 MHz
- **Đối với biến Max Power**
  - Min = 1 : Giá trị nhỏ nhất là 1 Watts
  - Q1 = 60 : Có 75% giá trị lớn hơn 60 Watts
  - Median = 125.6 : Có 50% giá trị lớn hơn 125.6 Watts
  - Mean = 126.7 : Giá trị trung bình của mẫu là 126.7 Watts
  - Q3 = 150 : Có 25% giá trị lớn 150 Watts
  - Max = 780 : Giá trị lớn nhất là 780 Watts
- **Đối với biến Memory**
  - Min = 16 : Giá trị nhỏ nhất là 16 MB
  - Q1 = 1024 : Có 75% giá trị lớn hơn 1024 MB
  - Median = 2048 : Có 50% giá trị lớn hơn 2048 MB
  - Mean = 2885 : Giá trị trung bình của mẫu là 2885 MB
  - Q3 = 4096 : Có 25% giá trị lớn 4096 MB
  - Max = 32000 : Giá trị lớn nhất là 32000 MB
- **Đối với biến Memory Bandwidth**
  - Min = 0.7812 : Giá trị nhỏ nhất là 0.7812 MB
  - Q1 = 28.8 : Có 75% giá trị lớn hơn 28.8 MB

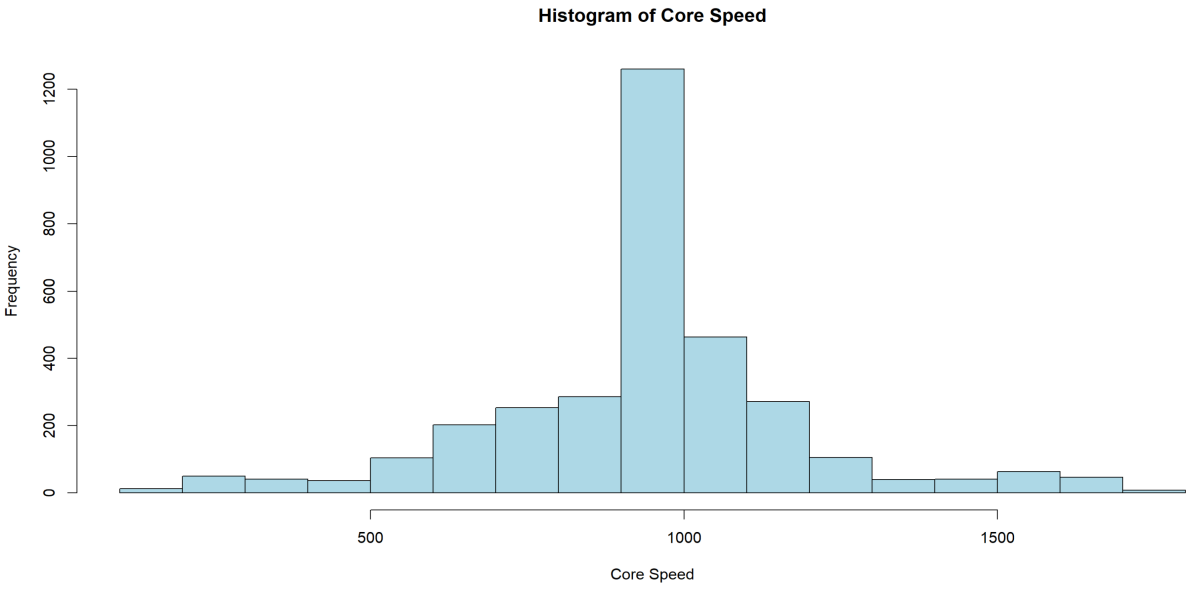
- Median = 105.8 : Có 50% giá trị lớn hơn 105.8 MB
- Mean = 137.1846 : Giá trị trung bình của mẫu là 137.1846 MB
- Q3 = 194.5 : Có 25% giá trị lớn 194.5 MB
- Max = 1280 : Giá trị lớn nhất là 1280 MB

- **Đối với biến Memory Bus**
  - Min = 32 : Giá trị nhỏ nhất là 32 Bit
  - Q1 = 128 : Có 75% giá trị lớn hơn 128 Bit
  - Median = 128 : Có 50% giá trị lớn hơn 128 Bit
  - Mean = 207.2 : Giá trị trung bình của mẫu là 207.2 Bit
  - Q3 = 256 : Có 25% giá trị lớn 256 Bit
  - Max = 8192 : Giá trị lớn nhất là 8192 Bit
- **Đối với biến Memory Speed**
  - Min = 100 : Giá trị nhỏ nhất là 100 MHz
  - Q1 = 800 : Có 75% giá trị lớn hơn 800 Mhz
  - Median = 1150 : Có 50% giá trị lớn hơn 1150 MHz
  - Mean = 1176 : Giá trị trung bình của mẫu là 1176 MHz
  - Q3 = 1502 : Có 25% giá trị lớn 1502 MHz
  - Max = 2127 : Giá trị lớn nhất là 2127 MHz
- **Đối với biến Process**
  - Min = 14 : Giá trị nhỏ nhất là 14 nm
  - Q1 = 28 : Có 75% giá trị lớn hơn 28 nm
  - Median = 28 : Có 50% giá trị lớn hơn 28 nm
  - Mean = 31.75 : Giá trị trung bình của mẫu là 31.75 nm
  - Q3 = 40 : Có 25% giá trị lớn 40 nm
  - Max = 150 : Giá trị lớn nhất là 150 nm
- **Đối với biến Pixel Rate**
  - Min = 1 : Giá trị nhỏ nhất là 1 GPixel/s
  - Q1 = 13 : Có 75% giá trị lớn hơn 13 GPixel/s
  - Median = 31 : Có 50% giá trị lớn hơn 31 GPixel/s
  - Mean = 35.15 : Giá trị trung bình của mẫu là 35.15 GPixel/s
  - Q3 = 40 : Có 25% giá trị lớn 40 GPixel/s
  - Max = 260 : Giá trị lớn nhất là 260 GPixel/s

3.2 Biểu đồ - Đồ thị

Nhóm sử dụng 2 biểu đồ tần suất (Histogram) để nhìn rõ sự phân phối giữa các biến độc lập và biểu đồ phân tán (Scatter Plots để so sánh sự phân phối giữa các biến)

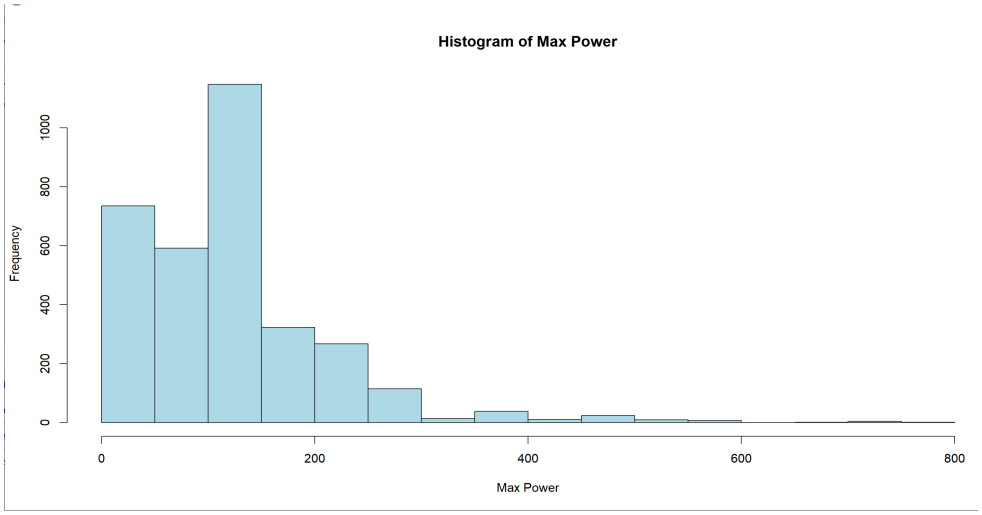
```
hist(new_GPU_data$Core_Speed,xlab="Core Speed",ylab="Frequency",main="Histogram of Core
    Speed")
```



Biểu đồ tần suất của biến Core Speed

Nhận xét:

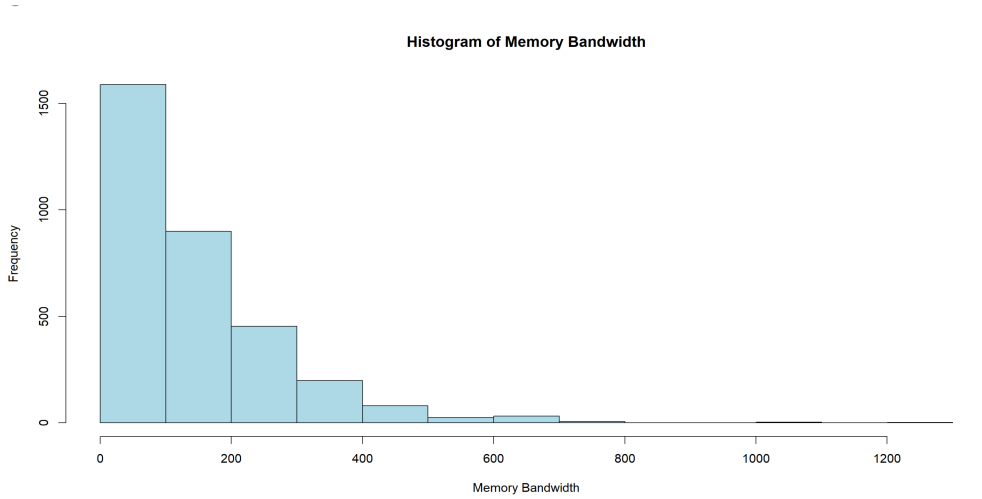
- Phần lớn GPU có Core Speed từ 500 MHz tới 1300 Mhz.
- Số GPU có Core Speed từ 900MHz tới 1000MHz có tỉ lệ cao nhất
- Số GPU có Core Speed từ 1700MHz tới 1800MHz có tỉ lệ thấp nhất
- Biến Core_Speed không phải phân phối chuẩn (biểu đồ tần suất không phải hình chuông)



Biểu đồ tần suất của biến Max_Power

Nhận xét:

- Phần lớn GPU có Max Power từ 0 tới 250 Watts.
- Số GPU có Max Power từ 100 - 150 Watts có tỉ lệ cao nhất
- GPU có Max Power > 500 Watts rất ít được sử dụng
- Biến Max_Power không phải phân phối chuẩn (biểu đồ tần suất không phải hình chuông)

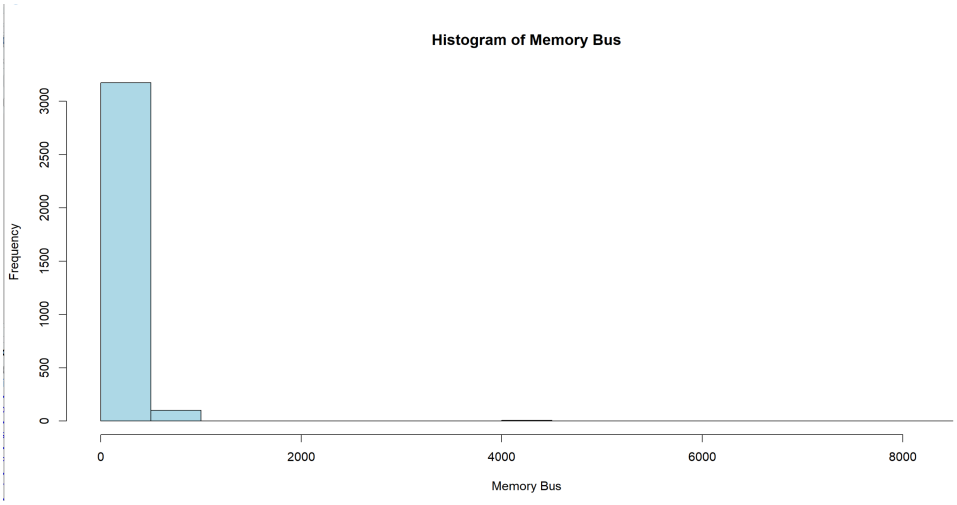


Biểu đồ tần suất của biến Memory Bandwidth

Nhận xét:

- Phần lớn GPU có Memory Bandwidth từ 0 - 500 MB/sec, GPU có Memory Bandwidth < 500 MB/sec được sử dụng phổ biến

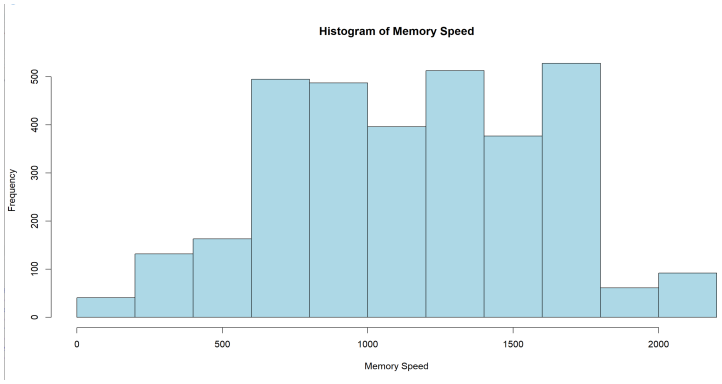
- Số GPU có Memory Bandwidth từ 0 - 100 MB/sec có tỉ lệ cao nhất
- Biến Memory_Bandwidth không phải phân phối chuẩn (biểu đồ tần suất không phải hình chuông)



Biểu đồ tần suất của biến Memory_Bus

Nhận xét:

- Hầu hết GPU chỉ có Memory Bus nằm trong khoảng từ 0 - 100 Bit.
- Số GPU có Memory Bus từ 0 - 50 Bit chiếm tỉ lệ cao nhất, sau đó là Memory Bus từ 50 - 100 Bit chiếm tỉ lệ nhỏ
- Biến Memory_Bus không phải phân phối chuẩn (biểu đồ tần suất không phải hình chuông)

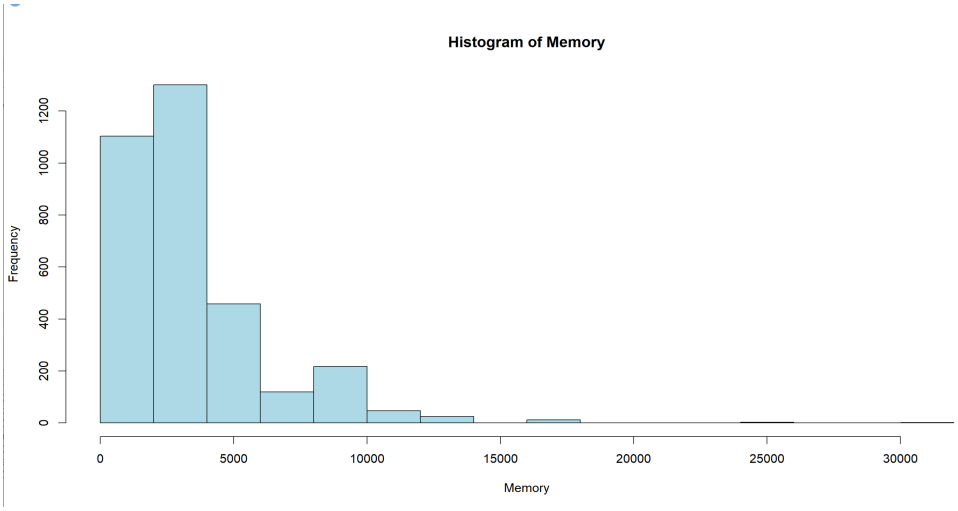


Biểu đồ tần suất của biến Memory Speed

Nhận xét:

- Phần lớn GPU có Memory Speed nằm trong khoảng từ 600 - 1800 MHz

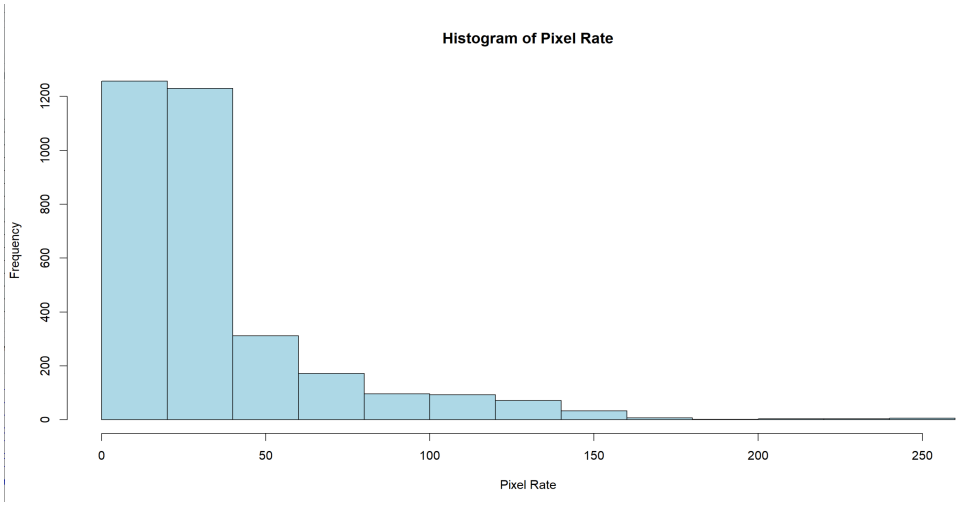
- Số GPU có Memory Speed từ 1600MHz - 1800 MHz có tỉ lệ cao nhất, số GPU có Memory Speed từ 0 - 200 MHz có tỉ lệ thấp nhất.
- Sự phân bố không đồng đều cho thấy có thể do sự khác biệt trong các mô hình hoặc cấu hình hay giá cả của GPUs nên sẽ có Memory Speed khác nhau
- Biến Memory_Speed không phải phân phối chuẩn (biểu đồ tần suất không phải hình chuông)



Biểu đồ tần suất của biến Memory

Nhận xét:

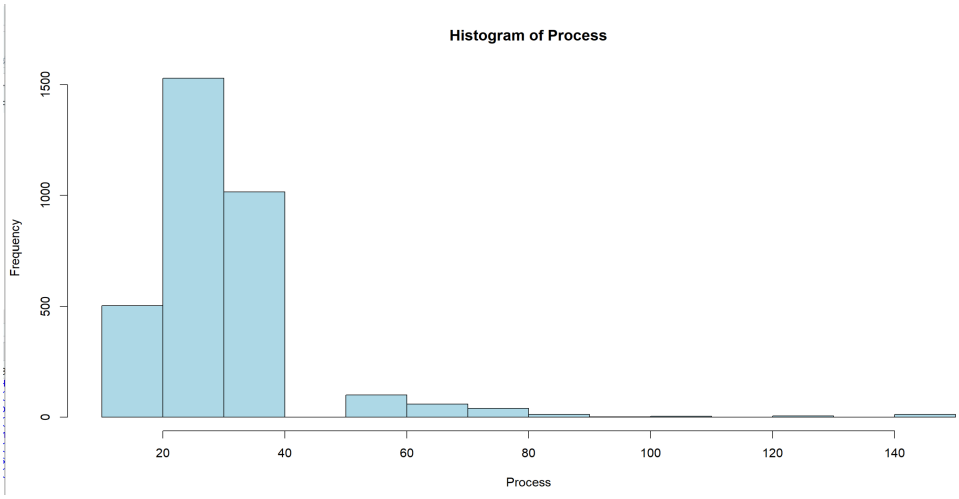
- Phần lớn GPU có bộ nhớ trong khoảng từ 0 - 10000 MB.
- Số GPU có bộ nhớ từ 2000 MB - 4000 MB chiếm tỉ lệ cao nhất
- GPU có bộ nhớ $>$ 10000 MB ít được sử dụng hơn, có thể do có ít nhu cầu sử dụng GPU với bộ nhớ lớn.
- Biến Memory không phải phân phối chuẩn (biểu đồ tần suất không phải hình chuông)



Biểu đồ tần suất của biến Pixel_Rate

Nhận xét:

- Phần lớn GPU có Pixel Rate từ 0 - 60 GPixel/s.
- GPU có Pixel Rate từ 0 - 40 GPixel/s chiếm tỉ trọng lớn nhất
- GPU có Pixel Rate $>$ 140 GPixel/s rất ít phổ biến
- Biến Pixel_Rate không phải phân phối chuẩn (biểu đồ tần suất không phải hình chuông)

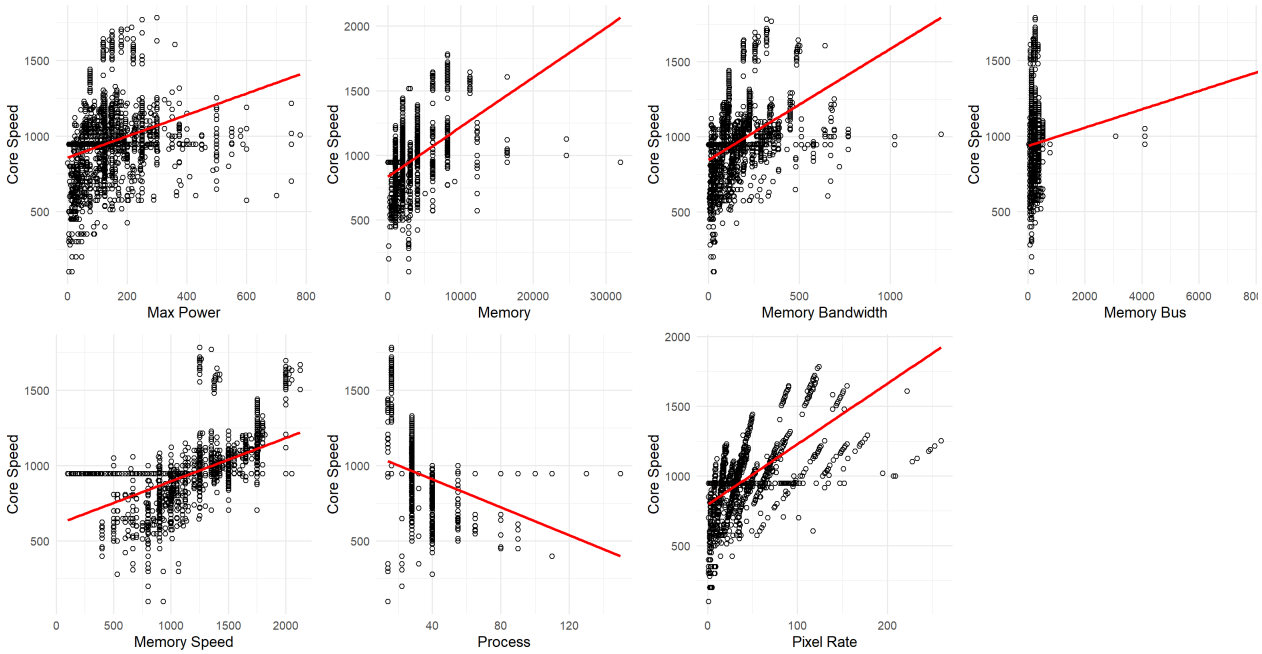


Biểu đồ tần suất của biến Process

Nhận xét:

- Phần lớn GPU có khả năng xử lý số lượng Process từ 0 - 40 nm.
- Số GPU có khả năng xử lý số lượng process từ 20 nm - 40 nm chiếm tỉ lệ cao nhất

- Số GPU có khả năng xử lý số lượng process > 40 nm không nhiều hoặc không sử dụng phổ biến
- Biến Process không phải phân phối chuẩn (biểu đồ tần suất không phải hình chuông)



Đồ thị phân tán thể hiện sự phân phối của Core Speed với các biến độc lập

Nhận xét:

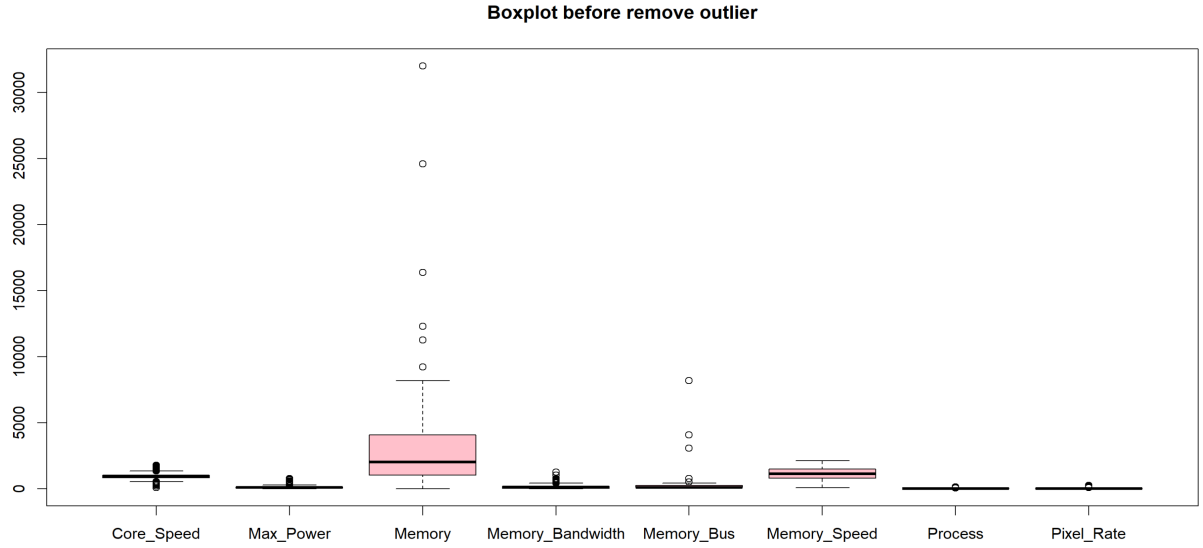
- Dựa trên đồ thị phân tán giữa các biến Max Power, Memory Bandwidth, Memory Speed, Pixel Rate và biến Core Speed ta thấy được mối quan hệ tuyến tính giữa 4 biến trên và Core Speed là đồng biến và phân tán ở mức đồ dày đặc
- Đồ thị phân tán giữa Memory, Memory Bus và Core Speed cho ta thấy 2 biến này có mối quan hệ tuyến tính với Core Speed và là mối quan hệ đồng biến, phân tán ở mức độ không dày đặc
- Đồ thị phân tán giữa Process và Core Speed cho thấy giữa hai biến có mối quan hệ tuyến tính và là mối quan hệ nghịch biến, phân tán ở mức độ không dày đặc.

4 Thống kê suy diễn

Vấn đề nhóm đặt ra ở bộ dữ liệu này là tìm hiểu những nhân tố nào ảnh hưởng đến hiệu năng của GPUs. Đặc biệt, trong các biến của file dữ liệu, nhân tố Core Speed đại diện cho hiệu năng của GPUs. Ta tiến hành xây dựng mô hình hồi quy tuyến tính đa biến để giải thích vấn đề này.

4.1 Xử lí outlier

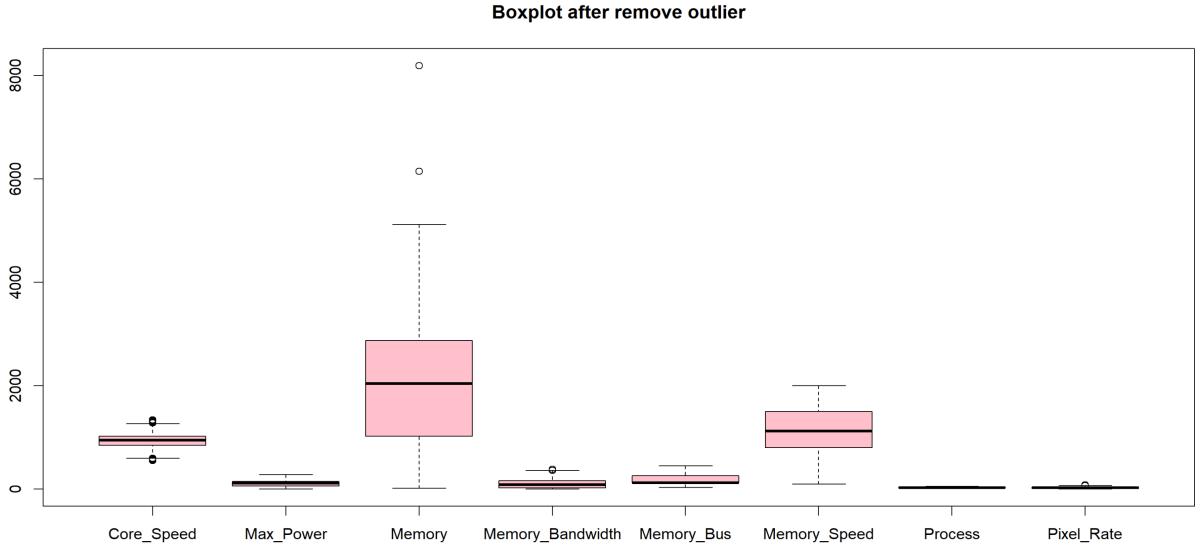
Trước khi xây dựng hồi quy, ta phải loại bỏ bớt các giá trị ngoại lai (outlier) để thu về được mô hình hiệu quả tốt nhất




```r
#xu li outlier dung tu phan vi

iqr_range <- apply(new_GPU_data, 2, function(x) {
  iqr <- IQR(x)
  lower <- quantile(x, 0.25) - 1.5 * iqr
  upper <- quantile(x, 0.75) + 1.5 * iqr
  c(lower, upper)
})

# Loai bo outlier tu DataFrame
use_GPUs_df <- new_GPU_data
for (i in 1:ncol(new_GPU_data)) {
  use_GPUs_df <- use_GPUs_df[!use_GPUs_df[, i] < iqr_range[1, i] & !use_GPUs_df[, i] >
      iqr_range[2, i], ]
}
```



4.2 Mô hình hồi quy tuyến tính

4.2.1 Phương pháp xây dựng mô hình

- Bước 1: Thực hiện phân tích hồi quy với các biến đã chọn với lệnh:
  - **lm()** để tạo ra mô hình hồi quy tuyến tính với 1 biến phụ thuộc và tất cả các biến còn lại là biến độc lập.
  - **summary()** liệt kê các thông tin tính toán cần thiết để giải quyết vấn đề bài toán
- Bước 2: Đọc kết quả và chọn ra các biến có giá trị thống kê. Ý nghĩa của bảng số summary:
  - Cột Estimate: Hệ số beta của mô hình hồi quy
  - Cột Std.Error: Độ lệch chuẩn ước lượng với hệ số beta tương ứng.
  - Cột t-value = Estimate/Std.Error: Là giá trị T trong kiểm định giả thuyết
  - Cột Pr(>|t|): Giá trị p-value trong kiểm định giả thuyết
- Thông thường người ta kiểm định p-value theo quy tắc:
  - p-value $>$ 0.05: biến không mang giá trị thống kê
  - p-value $\leq$ 0.05 : biến mang ý nghĩa thống kê
- Bước 3: Phân tích và loại bỏ các biến không có ý nghĩa thống kê Tiến hành phân tích và loại bỏ các biến không có ý nghĩa thống kê, tiếp tục xây dựng mô hình đến khi tìm được mô hình sao cho R^2 hiệu chỉnh cao

4.2.2 Xây dựng mô hình

Xét mô hình hồi quy tuyến tính bao gồm **Core_Speed** là biến phụ thuộc, và các biến độc lập bao gồm **Max_Power, Memory_Bandwidth, Memory_Bus, Memory_Speed, Process**. Dùng lệnh **lm()** để thực thi mô hình hồi quy tuyến tính bội. Mô hình 1 gồm các biến:

- **Core_Speed**: biến liên tục (biến phụ thuộc)
- **Max_Power**: biến liên tục
- **Memory**: biến liên tục
- **Memory_Bandwidth**: biến liên tục
- **Memory_Bus**: biến liên tục
- **Memor_Speed**: biến liên tục
- **Process**: biến liên tục
- **Pixel_Rate**: biến liên tục

```r
    Model1<- lm(Core_Speed~ Max_Power+Memory + Memory_Bandwidth + Memory_Bus
           + Memory_Speed + Process + Pixel_Rate ,data = use_GPUs_df) #Co 7 bien
summary(Model1)
```

Kết quả: Phân tích mô hình 1:

```
Residuals:
    Min      1Q  Median      3Q     Max
-409.72  -60.69    9.44   60.35  317.01

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      828.513337  17.584333  47.117  < 2e-16 ***
Max_Power          0.335829   0.054295   6.185 7.23e-10 ***
Memory            -0.014870   0.001734  -8.577  < 2e-16 ***
Memory_Bandwidth  -1.055022   0.078833 -13.383  < 2e-16 ***
Memory_Bus        -0.022827   0.046406  -0.492    0.623
Memory_Speed       0.233652   0.009212  25.364  < 2e-16 ***
Process           -5.392291   0.341223 -15.803  < 2e-16 ***
Pixel_Rate         4.564081   0.169995  26.848  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 98.04 on 2482 degrees of freedom
Multiple R-squared:  0.5665,    Adjusted R-squared:  0.5652
F-statistic: 463.3 on 7 and 2482 DF,  p-value: < 2.2e-16
```

Output của Model 1

- Sai số tiêu chuẩn (Residual standard error) là 98.04
- Hệ số R^2 hiệu chỉnh bằng 0.5665, nghĩa là 56.65% sự biến thiên trong hiệu năng GPU được giải thích bởi các biến Max_Power, Memory, Memory_Bandwidth, Memory_Bus, Memory_Speed, Process, Pixel_Rate
- Kết quả trên cho chúng ta thông tin về hệ số góc của các biến độc lập và chúng có tác dụng đến Core_Speed như thế nào
- Loại biến không có ý nghĩa thống kê là biến Memory Bus (do p-value ≤ 0.05)

Mô hình 2 gồm các biến Max_Power, Memory, Memory_Bandwidth, Memory_Speed, Process, Pixel_Rate

```
    Model2 <- lm (Core_Speed~ Max_Power + Memory + Memory_Bandwidth +
                  Memory_Speed + Process + Pixel_Rate, data = use_GPUs_df) #Co 6 bien
summary(Model2)
```

Kết quả: Phân tích mô hình 2:

```
Residuals:
    Min      1Q  Median      3Q     Max
-408.86  -60.73    9.34   60.42  319.81

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      826.176019  16.927594  48.806  < 2e-16 ***
Max_Power          0.331943   0.053709   6.180 7.45e-10 ***
Memory            -0.014700   0.001699  -8.654  < 2e-16 ***
Memory_Bandwidth  -1.081770   0.057068 -18.956  < 2e-16 ***
Memory_Speed       0.235489   0.008420  27.969  < 2e-16 ***
Process           -5.425880   0.334269 -16.232  < 2e-16 ***
Pixel_Rate         4.567251   0.169847  26.890  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 98.03 on 2483 degrees of freedom
Multiple R-squared:  0.5664,    Adjusted R-squared:  0.5654
F-statistic: 540.6 on 6 and 2483 DF,  p-value: < 2.2e-16
```

Output của Model 2

- Sai số tiêu chuẩn (Residual standard error) là 98.03
- Hệ số R^2 hiệu chỉnh bằng 0.5665, nghĩa là 56.65% sự biến thiên trong hiệu năng GPU được giải thích bởi các biến Max_Power, Memory, Memory_Bandwidth, Memory_Speed, Process, Pixel_Rate
- Kết quả trên cho chúng ta thông tin về hệ số góc của các biến độc lập và chúng có tác dụng đến Core_Speed như thế nào

- Các hệ số p-value đều ≤ 0.05 nên không loại biến nào

Cả hai mô hình đều có hệ số R^2 hiệu chỉnh bằng 0.5665 nhưng do ở mô hình 2, tất cả các biến đều có ý nghĩa thống kê nên nhóm quyết định chọn mô hình 2 để thực hiện dự đoán.

4.2.3 Phân tích sự tác động của các yếu tố lên hiệu năng GPUs:

Như vậy, mô hình hồi quy tuyến tính về ảnh hưởng của các nhân tố đến hiệu năng Core_Speed của GPUs được cho bởi biểu thức sau:

Core_Speed = 826.176019 + 0.331943 x Max_Power -0.014700 x Memory - 1.081770 x Memory_Bandwidth + 0.235489 x Memory Speed - 5.425880 x Process + 4.567251 x Pixel_Rate

Trước hết, ta thấy p-value tương ứng với thống kê F bé hơn 2.2 x 10^{-16}, có ý nghĩa rất cao. Điều này chỉ ra rằng, ít nhất có một biến dự báo trong mô hình có ý nghĩa giải thích rất cao cho biến Core_Speed.
Để xét ảnh hưởng cụ thể từng biến độc lập, ta xét trọng số (hệ số β_i và p-value tương ứng). Ta thấy rằng p-value tương ứng với các biến đều bé hơn 2 x 10^{-6} (ngoài trừ biến Max_Power có p-value= 7.45 x 10^{-10}), điều này nói lên rằng ảnh hưởng của các biến này có ý nghĩa rất cao lên hiệu năng GPUs.
Mặt khác, hệ số hồi quy β_i của một biến dự báo cũng có thể được xem như ảnh hưởng trung bình lên biến phụ thuộc hiệu năng GPUS Core_Speed khi tăng một đơn vị của biến dự báo đó, giả sử rằng các biến dự báo không đổi. Cụ thể, $\beta_4 = 0.235489$ thì khi giá trị của Memory Speed tăng lên 1 MHz, ta có thể kỳ vọng tốc độ Core_Speed của GPU sẽ tăng lên 0.234589 MHZ về mặt trung bình (giả sử rằng các biến dự báo không đổi). Tương tự, đối với các biến còn lại trong mô hình.
Tóm lại, qua số liệu thu được từ mô hình Model 2, ta có thể kết luận là đa số các biến độc lập đều có mối quan hệ tuyến tính với biến phụ thuộc

4.3 Dự đoán

Dùng lệnh **predict()** để tiến hành dự báo cho biến **Core_Speed**

```
cp=use_GPUs_df$Core_Speed
mp=use_GPUs_df$Max_Power
m=use_GPUs_df$Memory
mba=use_GPUs_df$Memory_Bandwidth
mbu=use_GPUs_df$Memory_Bus
ms=use_GPUs_df$Memory_Speed
p=use_GPUs_df$Process
pr=use_GPUs_df$Pixel_Rate

data_predict=data.frame(mp,m,mba,ms,p,pr)
P<-lm(cp~.,data=data_predict)
summary(P)
predict_CoreS=predict(P)
p=data.frame(predict_CoreS,cp)
```

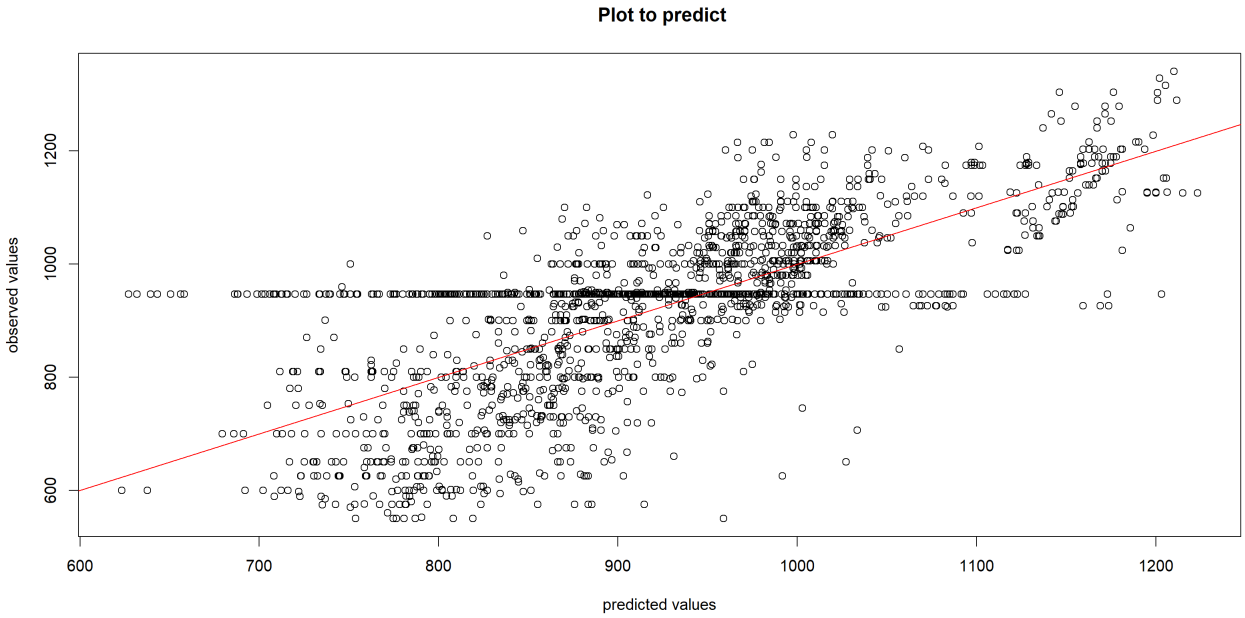
Một phần giá trị dự báo so với thực tế:

	predict_CoreS	cp
1	780.5666	738.0000
2	726.5314	870.0000
3	929.3967	946.8939
4	905.5355	946.8939
5	769.0998	650.0000
6	898.8967	705.0000
7	932.0549	946.8939
8	910.7253	1050.0000
9	857.7631	732.0000
10	778.3643	575.0000
11	781.3518	575.0000
12	823.5227	575.0000
13	900.4671	837.0000
14	881.6103	902.0000
15	928.2826	928.0000
16	900.4671	837.0000
17	900.4671	837.0000

Giá trị dự báo với trong file dữ liệu

So sánh kết quả và nhận xét:

```
x<- (1:50)
y<-x
kql = lm(y~x)
par(mfrow=c(1,1))
plot(p,xlab="predicted values",ylab="observed values",main=" Plot to predict")
abline(kql,col="red")
```



Biểu đồ dự báo cho biến Core_Speed

Nhận xét: Biểu đồ cho ta kết quả các giá trị dự báo cho hiệu năng **Core_Speed** của GPUS sai lệch không quá nhiều so với dữ liệu trong file, bằng chứng là nhiều điểm biểu diễn giá trị dự đoán và giá trị thực tế gần như nằm trên đường màu đỏ. Vì vậy mô hình hồi quy tuyến tính **Model 2** tương đối ổn, và có ý nghĩa thống kê

4.4 Thực hiện dự báo cho hiệu năng Core_Speed của GPUS

Dựa trên mô hình hồi quy ta đã xây dựng, dự đoán **Core_Speed** của một con chip GPU có thông số sau:

- **Max_Power** = 150 Watts
- **Memory** = 2048 MB
- **Memory_Bandwidth** = 200 GB/sec
- **Memory_Bus** = 256 Bit
- **Memory_Speed** = 900 MHz
- **Process** = 40 nm
- **Pixel_Rate** = 42 GPixel/s

```
    #Du doan co so lieu
X <- data.frame("Max_Power"=150,"Memory"=2048,
       "Memory_Bandwidth"=200,"Memory_Bus"=256,
              "Memory_Speed"=900,"Process"=40, "Pixel_Rate"=42)
predict_X <- predict(Model2,X,interval ="confidence")
head(predict_X)
```

Kết quả:

```
       fit     lwr      upr
1 816.2374 804.288 828.1869
```

Nhận xét: Dựa vào kết quả dự báo, ta nhận được:

- Giá trị Core_Speed hiệu năng trung bình là 816.2374 MHz.
- Khoảng tin cậy so với giá trị dự báo là (804.288;828.1869)

5 Thảo luận và Mở rộng

Trong bài tập lớn, đối với các biến định lượng chứa tỉ lệ dữ kiệu khuyết nhỏ hơn 5%, nhóm sẽ loạt bỏ các quan sát (object) chứa dữ liệu kiếm khuyết, đối với tỉ lệ khiếm khuyết trên 5% và dưới 50% thì sẽ dùng phương pháp thế trung vị lên các ô NA trong cột.
Dưới đây là một số ưu điểm và nhược điểm của phương pháp này:
Ưu điểm:

- Dễ triển khai: Phương pháp thế trung vị là phương pháp đơn giản và dễ triển khai. Bạn chỉ cần tính giá trị trung vị của biến và thay thế các giá trị thiếu bằng giá trị này.
- Không thay đổi phân phối: Khi sử dụng trung vị để thay thế các giá trị thiếu, phân phối của biến không bị thay đổi. Điều này có ý nghĩa khi sử dụng các phương pháp thống kê dựa trên giả định về phân phối như phân phối chuẩn.
- Ổn định: Phương pháp thế trung vị ít bị ảnh hưởng bởi giá trị ngoại lệ (outliers). Vì nó dựa trên giá trị trung vị, các giá trị ngoại lệ không có ảnh hưởng lớn đến kết quả

Nhược điểm

- Mất thông tin: Khi thay thế các giá trị thiếu bằng trung vị, ta mất đi thông tin về biến và sự biến động của dữ liệu. Việc này có thể làm giảm khả năng phân biệt và khả năng dự đoán của mô hình.
- Gây độc lập: Phương pháp thế trung vị không khai thác các mối quan hệ hoặc sự tương quan giữa các biến. Điều này có thể dẫn đến mất mát thông tin liên quan đến các biến khác trong mô hình.
- Không phù hợp cho biến có phân phối không đối xứng: Trong trường hợp biến có phân phối không đối xứng, việc thay thế bằng trung vị có thể làm biến đổi phân phối và làm sai lệch kết quả.
- Giảm độ biến thiên: Khi thay thế các giá trị thiếu bằng trung vị, độ biến thiên của biến có thể bị giảm do giá trị trung vị là một giá trị cố định.

Có thể sử dụng phương pháp tối ưu hơn là K-nearest neighbors (KNN) để xử lí NA nhưng do phương pháp khá phức tạp nên nhóm không xử dụng.

6 Nguồn dữ liệu và code

Nguồn dữ liệu: https://www.kaggle.com/datasets/iliassekkaf/computerparts?resource=downlo
Nguồn code: https://drive.google.com/file/d/1ECgjvbvcMPQ8mvuShM1aV04aoQ6JG09R/view?usp=sharing

7 Tài liệu tham khảo

1. Nguyễn Đình Huy (chủ biên), Nguyễn Bá Thi, Giáo trình Xác suất và Thống kê, 2018
2. Peter Dalgaard, Introductory Statistics with R, Second Edition (2008)