# Các kỹ thuật hiệu chỉnh mô hình

Nguyễn Thanh Tùng, **Trần Thị Ngân**
Khoa Công nghệ thông tin – Đại học Thủy lợi
*tungnt@tlu.edu.vn*, ***ngantt@tlu.edu.vn***

# Hiệu chỉnh mô hình

- Một cách cụ thể hơn, ta sẽ tìm cách *di chuyển* nghiệm của bài toán tối ưu hàm tổn thất *tới một điểm gần nó*. Hướng di chuyển sẽ là hướng ***làm cho mô hình ít phức tạp hơn*** mặc dù giá trị của hàm tổn thất có tăng lên một chút.

# Mô hình có điều chỉnh (Generalized model)

# Nhắc lại: Hồi quy tuyến tính đa biến

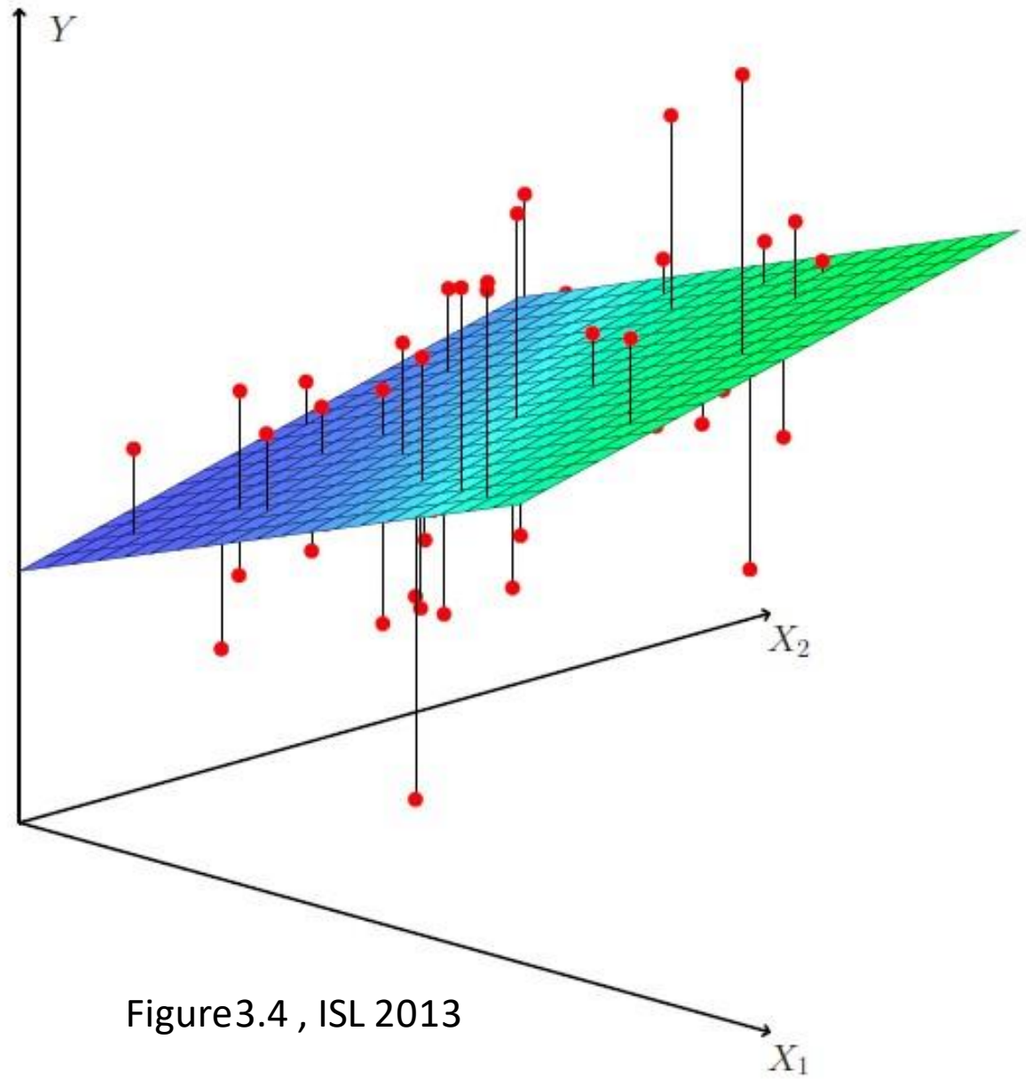



Figure3.4 , ISL 2013

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2$$

# Trường hợp phức tạp

Khi có quá nhiều biến đầu vào

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

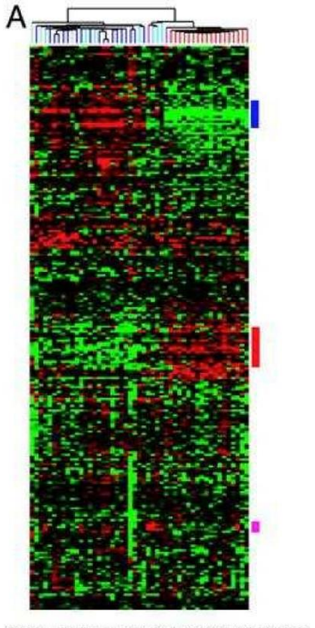
Hoặc khi có tương tác giữa các biến đầu vào

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot (X_1 X_2) + \beta_4 \cdot X_1^2 + \beta_5 \cdot X_2^2 + \beta_6 \cdot \log(X_1 / X_2) + \beta_7 \cdot \sin(X_1 - X_2)$$

# Trường hợp phức tạp

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

Gene expression arrays



# Điều gì xảy ra?

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

Câu hỏi: Ta có 8 biến và có hàng trăm mẫu. Hai biến ($X_3$ và $X_4$) có tương quan yếu với Y (do đó cũng có vai trò nhỏ cho dự đoán), tuy nhiên chúng có tương quan cao với các biến khác. Điều gì xảy ra khi diễn giải các hệ số β của hai biến $X_3$ và $X_4$?

# Đa cộng tuyến (Multi-collinearity)

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

- Theo giả thiết của phương pháp Hồi quy tuyến tính thì các biến độc lập không có mối quan hệ tuyến tính.
- Nếu quy tắc này bị vi phạm thì sẽ có hiện tượng đa cộng tuyến: là hiện tượng **các biến độc lập trong mô hình phụ thuộc tuyến tính lẫn nhau** và thể hiện được dưới dạng hàm số

# Đa cộng tuyến (Multi-collinearity)

Mức độ đa cộng tuyến được đo bởi hệ số VIF (Variance Inflation Factor) là tỷ lệ phương sai trong một mô hình có nhiều biến chia cho phương sai của một mô hình chỉ có một biến

- Ví dụ về đa cộng tuyến trên bộ dữ liệu Boston

| crim | zn | indus | chas | nox | rm | age | dis | rad |
|---|---|---|---|---|---|---|---|---|
| 1.87 | 2.36 | 3.90 | 1.06 | 4.47 | 2.01 | 3.02 | 3.96 | 7.80 |
| tax | ptratio | black | lstat | | | | | |
| 9.16 | 1.91 | 1.31 | 2.97 | | | | | |

# Hồi quy tuyến tính đa biến

Quay lại hồi quy tuyến tính, ta cố gắng để cực tiểu hóa sai số bình phương

$$\sum_{\text{các mẫu}} [Y - (\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2)]^2$$

# Hiệu chỉnh mô hình (Regularization)

## Hồi quy Ridge

Tìm giá trị β để cực tiểu lỗi phạt "penalized", tương đương với

$$\sum_{\text{các mẫu}} [Y - (\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2)]^2 + \lambda \cdot (\beta_0^2 + \beta_1^2 + \beta_2^2)$$

L2

hoặc viết ở dạng khác,

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left( \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 \right)$$

$\lambda$ được gọi là tham số hiệu chỉnh (tuning parameter)

# Hệ số hiệu chỉnh mô hình Ridge

**Giá trị của $\lambda$:**

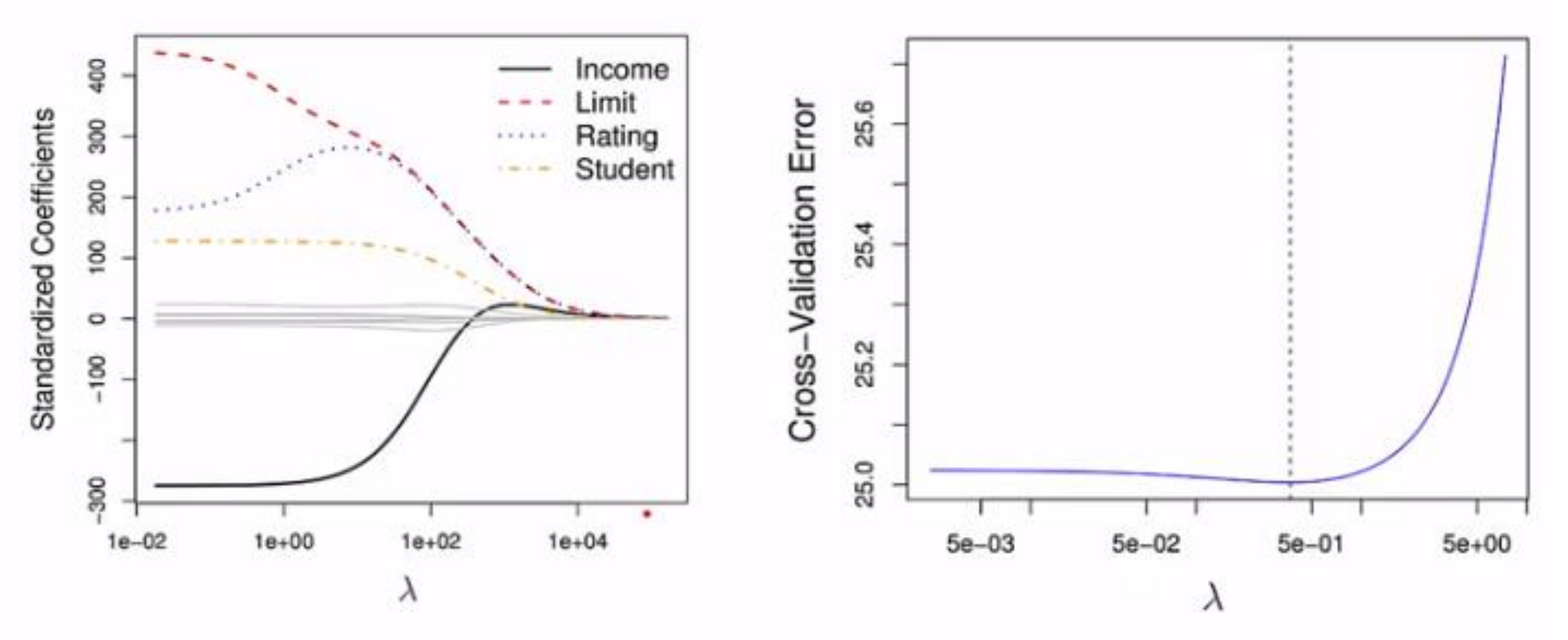
- Lớn: Hầu hết các hệ số $\beta$ giảm về 0 $\rightarrow$ underfitting
- Nhỏ: tương tự Hồi quy tuyến tính thông thường $\rightarrow$ overfitting
- Thường được lựa chọn bằng phương pháp đánh giá chéo (CV)

# Chuẩn hóa các biến đầu vào

- Các biến đầu vào có đơn vị tính và khoảng giá trị khác nhau $\rightarrow$ Các giá trị $\beta$ có thể chênh lệch lớn nên cần chuẩn hóa các biến số trước khi ước lượng mô hình
- **Chuẩn hóa**:

- Lấy giá trị của mỗi biến trừ đi trung bình (mean) của nó
- Lấy kết quả nhận được chia cho độ lệch chuẩn của biến

# Ví dụ: Hồi quy Ridge



# Hiệu chỉnh mô hình

Ta đã xử lý:

- *Underdetermined*
- *Overfitting*
- Đa cộng tuyến (*Multi--collinearity)*

Vậy mô hình thưa là gì (*sparsity)*?

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

$\beta_2 \to 0$, $\beta_4 \to 0$, $\beta_7 \to 0$

# Mô hình thưa (Sparsity)

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

$\beta_2 \to 0$, $\beta_4 \to 0$, $\beta_7 \to 0$

- Dùng cho lựa chọn biến (Feature selection)
- Thời gian tính toán lâu (computational efficiency)

# Mô hình thưa (Sparsity)

## Lasso

"**L**east **a**bsolute **s**hrinkage and **s**election **o**perator"

$$\sum_{samples} [Y - (\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2)]^2 + \lambda \cdot (|\beta_0| + |\beta_1| + |\beta_2|)$$

L1

Mô hình giống như hồi quy Rigde nhưng ***khác hàm phạt***

# Hiệu chỉnh mô hình

**Hồi quy Ridge:**

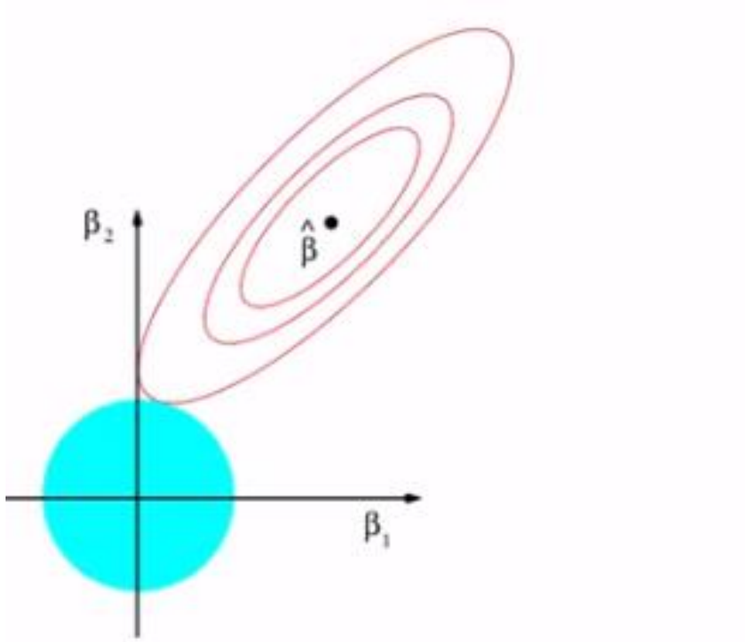
- Giữ lại tất cả các biến số, chỉ làm giảm các hệ số ước lượng về 0

**Hồi quy Lasso**

- Giúp lựa chọn các biến có ý nghĩa, các biến số khác được gán bằng 0

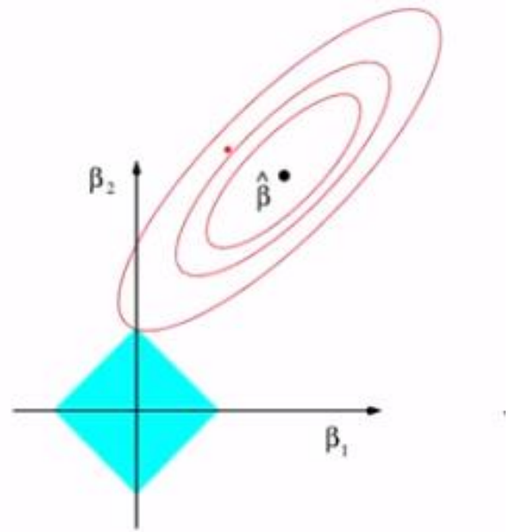
# Hiệu chỉnh mô hình

## Hồi quy Ridge




$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

## Hồi quy Lasso




$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

# Mục tiêu khác: Mô hình thưa

## Lasso

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \equiv \hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 \quad s.t. \sum_{j=1}^{p} |\beta_j| \leq t$$

## Ridge

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \equiv \hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 \quad s.t. \sum_{j=1}^{p} \beta_j^2 \leq t$$

# Ví dụ về Ridge và Lasso trên R

# Thư viện và dữ liệu

- Thư viện: car (carData), ridge
- Dữ liệu: Longley chứa các biến số kinh tế vĩ mô khác nhau của Hoa Kỳ:

- GNP.Deflator - Giảm lạm phát GNP
- GNP – Tổng sản lượng quốc gia
- Unemployed - Số người thất nghiệp
- Armed.Forces - Quy mô lực lượng vũ trang
- Population - Dân số
- YEAR - Năm (1947 - 1962)
- Employed - Tổng số việc làm

# Thực hiện

```
library (car)
library (ridge)
data(longley, package="datasets")  # initialize data
head (longley)#Hiển thị 6 bản ghi đầu tiên của dữ liệu
```

```
     GNP.deflator     GNP Unemployed Armed.Forces Population Year Employed
1947         83.0 234.289      235.6        159.0    107.608 1947   60.323
1948         88.5 259.426      232.5        145.6    108.632 1948   61.122
1949         88.2 258.054      368.2        161.6    109.773 1949   60.171
1950         89.5 284.599      335.1        165.0    110.929 1950   61.187
1951         96.2 328.975      209.9        309.9    112.075 1951   63.221
1952         98.1 346.999      193.2        359.4    113.270 1952   63.639
```

# Thực hiện

```
inputData <- data.frame (longley)
colnames(inputData)[1] <- "response"  #Thay đổi tiêu đề cột 1
#GNP.deflator đổi thành response
XVars <- inputData[, -1] # Lấy dữ liệu longley trừ cột đầu tiên
round(cor(XVars), 2)#Tính hệ số tương quan giữa các biến
```

| | GNP | Unemployed | Armed.Forces | Population | Year | Employed |
|---|---|---|---|---|---|---|
| GNP | 1.00 | 0.60 | 0.45 | 0.99 | 1.00 | 0.98 |
| Unemployed | 0.60 | 1.00 | -0.18 | 0.69 | 0.67 | 0.50 |
| Armed.Forces | 0.45 | -0.18 | 1.00 | 0.36 | 0.42 | 0.46 |
| Population | 0.99 | 0.69 | 0.36 | 1.00 | 0.99 | 0.96 |
| Year | 1.00 | 0.67 | 0.42 | 0.99 | 1.00 | 0.97 |
| Employed | 0.98 | 0.50 | 0.46 | 0.96 | 0.97 | 1.00 |

# Chuẩn bị dữ liệu training và testing

```
set.seed(100) # Để nhân rộng kết quả

trainingIndex <- sample(1:nrow(inputData), 0.8*nrow(inputData))

trainingData <- inputData[trainingIndex, ] # training data

testData <- inputData[-trainingIndex, ] # test data
```

# Mô hình hồi quy tuyến tính

lmMod <- lm(response ~ ., trainingData)
summary (lmMod)

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)  7652.25192 7028.03242   1.089   0.3259
GNP             0.39214    0.14881   2.635   0.0462 *
Unemployed      0.06462    0.03887   1.663   0.1573
Armed.Forces    0.01573    0.01861   0.845   0.4366
Population     -2.33550    0.99029  -2.358   0.0649 .
Year           -3.83113    3.65618  -1.048   0.3427
Employed        0.53060    1.50698   0.352   0.7391
```

# Mô hình hồi quy tuyến tính

vif(lmMod)

```
> vif(lmMod)
       GNP    Unemployed Armed.Forces    Population          Year      Employed
 1523.74714      93.07635      10.74587     350.58472    2175.29221    182.93609
>
```

Có sự đa cộng tuyến đáng kể giữa GNP & Year và Population & Employed, với các hệ số âm trong Population và Employed. Các biến này có thể không đóng góp nhiều để giải thích biến phụ thuộc

# Đánh giá mô hình hồi quy tuyến tính

*#Dự báo trên tập testing*

predicted <- predict (lmMod, testData)

compare <- cbind (actual=testData$response, predicted)

#Tính độ chính xác của mô hình

Accuracy_LM=mean (apply(compare, 1, min)/apply(compare, 1, max))

Accuracy_LM

```
> Accuracy_LM
[1] 0.9876802
```

# Mô hình Ridge

```
RidgeMod <- linearRidge(response ~ ., data = trainingData)  # XD mô hình

predicted1 <- predict(RidgeMod, testData)  # Dự báo trên testing

compare1 <- cbind (actual=testData$response, predicted1)

Accuracy_RM=mean (apply(compare1, 1, min)/apply(compare1, 1, max))

Accuracy_RM
```

```
> Accuracy_RM
[1] 0.9910763
```

# Câu hỏi?