

Bài giảng R, số 7

-Xử lý khuyết dữ liệu cho mô hình hồi quy-

TS.Tô Đức Khánh

23/06/2024

Trong phần này, ta sẽ tìm hiểu các bước xử lý dữ liệu khuyết khi làm việc với mô hình hồi quy tuyến tính:

1. Chuẩn bị dữ liệu:
 - kiểm tra % dữ liệu bị khuyết;
 - điều tra dạng dữ liệu bị khuyết và cơ chế khuyết có thể: MCAR, MAR hoặc MNAR;
 - áp dụng multiple imputation để tạo ra nhiều tập dữ liệu ước đoán khác nhau.
2. Xây dựng và huấn luyện mô hình: sử dụng các thuật toán xây dựng mô hình hồi quy trên các tập dữ liệu ước đoán.
3. Tổng hợp kết quả ước lượng trên các tập dữ liệu để tạo thành một kết quả chung.
4. Đánh giá độ chính xác của mô hình.
5. Áp dụng mô hình và giám sát.

1 Xác định loại khuyết dữ liệu

Trong R, để xem biến nào bị khuyết và tỷ lệ khuyết là bao nhiêu, đồng thời để hiểu dạng khuyết, chúng ta có thể sử dụng hai lệnh `aggr()` và `marginplot()` trong gói VIM:

- `aggr()` cung cấp hai biểu đồ, một biểu đồ hiển thị tỷ lệ bị thiếu, một biểu đồ khác hiển thị dạng khuyết;
- `marginplot()` hiển thị biểu đồ phân tán gồm hai biến với các ô bổ sung dựa trên các giá trị còn thiếu và giá trị quan sát được. Nếu giả định về dữ liệu MCAR là chính xác thì chúng tôi kỳ vọng các ô màu đỏ và xanh lam sẽ rất giống nhau.

Ví dụ 1: ta xét tập dữ liệu `AIDA_expamle.csv` chứa dữ liệu về các công ty hoạt động trong lĩnh vực trang sức và thời trang tại Ý. Các biến liên quan tới lợi nhuận (`profit`), doanh thu (`revenue`), số lượng nhân viên (`employees`), lợi nhuận trên tài sản (`roa`), lợi nhuận trên vốn chủ sở hữu (`roe`), trụ sở công ty (`province`).

```
aida <- read_csv(file = "datasets/AIDA_example.csv")
aida <- aida |> janitor::clean_names()
glimpse(aida)

## # Rows: 1,248
## # Columns: 15
## $ cid      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, ~
## $ province <chr> "Alessandria", "Milano", "Milano", "Alessandria", "Vicenza", "V~
## $ revenue   <dbl> 207918, 160615, 146019, 141401, 126609, 121147, 89497, 82429, 7~
## $ employees <dbl> 454, 259, 403, 304, 84, 132, 57, 59, 67, 110, 75, 102, 12, 104, ~
## $ total_asset <dbl> 165588, 97147, 127722, 224664, 45174, 48912, 16707, 23785, 8979~
```

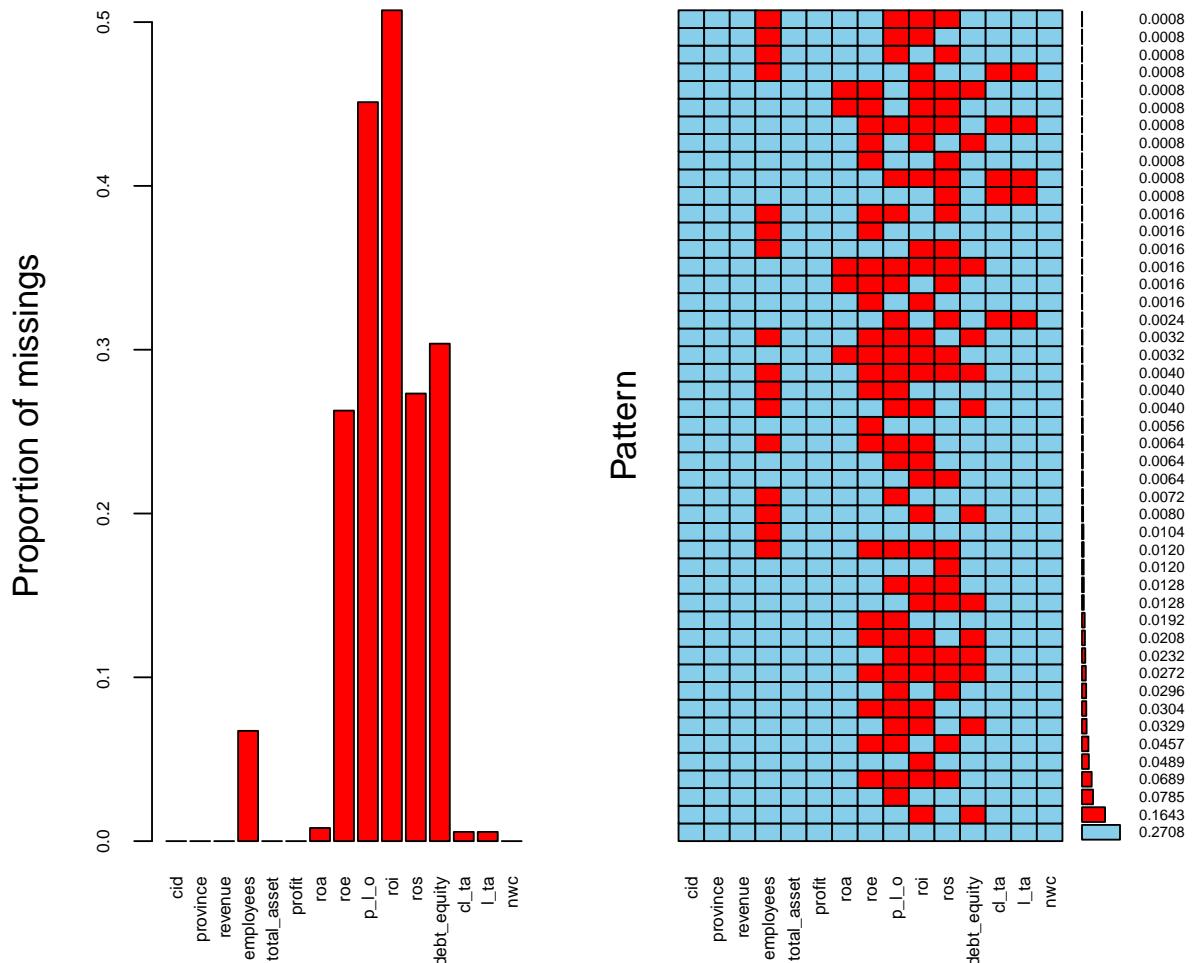
```

## $ profit      <dbl> 10691, 5348, 3341, -871, 3164, 1823, 339, 732, 5573, -5158, 179~
## $ roa        <dbl> 11.09, 10.07, 16.75, 1.10, 10.22, 6.12, 3.79, 1.02, 9.63, -32.5~
## $ roe        <dbl> 25.97, 29.73, 4.56, -1.03, 13.12, 9.30, 5.93, 12.75, 17.25, NA, ~
## $ p_l_o       <dbl> 58.24, 54.64, 15.62, -35.38, 68.54, 60.86, 53.51, 301.52, 64.43~
## $ roi         <dbl> NA, NA, 29.19, 1.74, 10.91, 6.73, 4.71, 1.24, 16.84, NA, NA, 15~
## $ ros         <dbl> 8.82, 6.08, 14.51, 1.72, 3.64, 2.46, 0.71, 0.29, 11.58, -5.44, ~
## $ debt_equity <dbl> 0.00, 0.00, 0.00, 0.56, 0.46, 1.27, 1.36, 2.42, 0.59, 0.00, 0.0~
## $ cl_ta       <dbl> 1.00, 0.81, 1.00, 0.87, 0.56, 0.89, 1.00, 0.94, 0.94, 0.87, 1.0~
## $ l_ta        <dbl> 0.00, 0.19, 0.00, 0.13, 0.44, 0.11, 0.00, 0.06, 0.06, 0.13, 0.0~
## $ nwc         <dbl> 4332, 25171, 54151, 56141, 31643, 6754, 3988, 1467, 35589, 98, ~

```

Ta nhận thấy rằng, có các giá trị khuyết, NA, hiện diện trong dữ liệu. Để kiểm tra phần trăm bị khuyết trong dữ liệu cũng như các biến bị khuyết và xu hướng, ta dùng hàm `aggr()` như sau:

```
aggr(aida, ylab = c("Proportion of missings", "Pattern"), number = TRUE,
      cex.axis = 0.6, cex.numbers = 0.5)
```



Biểu đồ đầu tiên cho thấy số lượng % bị khuyết của biến trong dữ liệu:

- `roi` bị khuyết nhiều nhất, với khoảng hơn 50% đối tượng;

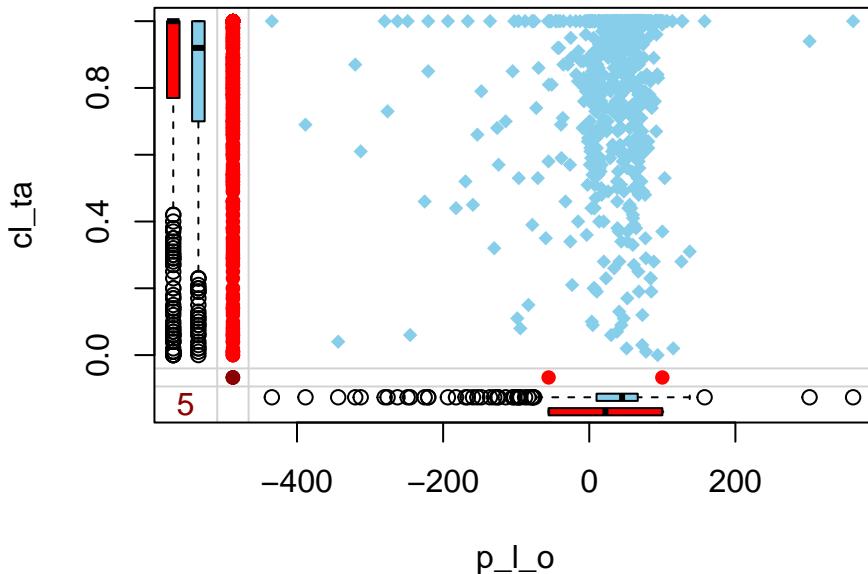
- `p_l_o` có khoảng 46% dữ liệu bị khuyết;
- `province`, `revenue`, `total_asset`, `profit` và `nwc` là không bị khuyết.

Biểu đồ thứ hai cho ta biết về xu hướng khuyết của các biến:

- chỉ có khoảng 27% dữ liệu là đầy đủ quan sát cho tất cả các biến;
- khoảng 16.43% số lượng quan sát là bị khuyết thông tin của `roi` và `debt_equity`;
- khoảng 7.85% dữ liệu chỉ thiếu thông tin của `p_l_o`;
- và những dòng còn lại hiển thị các dạng khuyết khác.

Đi vào cụ thể hơn, ta quan sát khả năng khuyết của từng cặp biến. Chẳng hạn, ta xét `p_l_o` và `cl_ta`, sử dụng hàm `marginplot()` như sau:

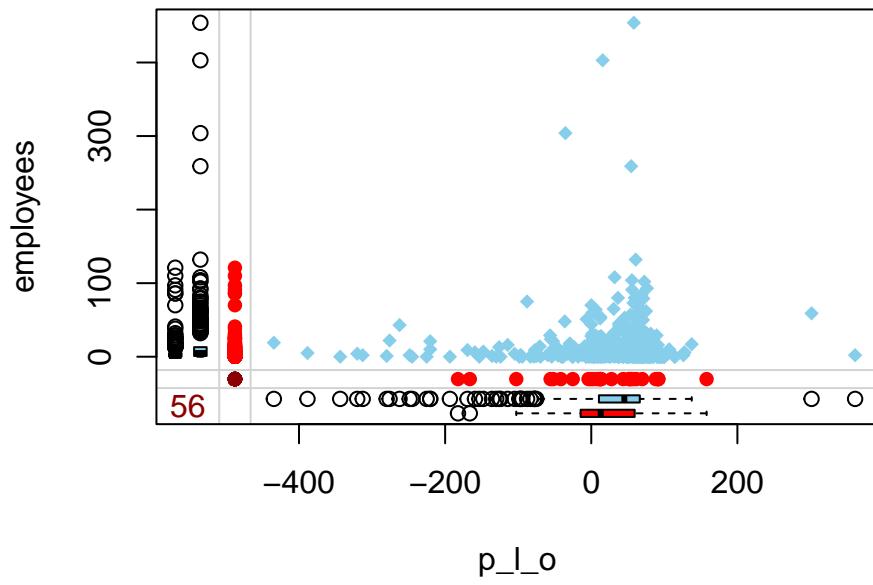
```
marginplot(aida[, c("p_l_o", "cl_ta")], pch = c(18, 16))
```



Dựa trên đồ thị, ta thấy rằng các biểu đồ hộp là không trùng khớp nhau, do đó, cơ chế khuyết dữ liệu của `p_l_o` và `cl_ta` không thể là MCAR (missing completely at random).

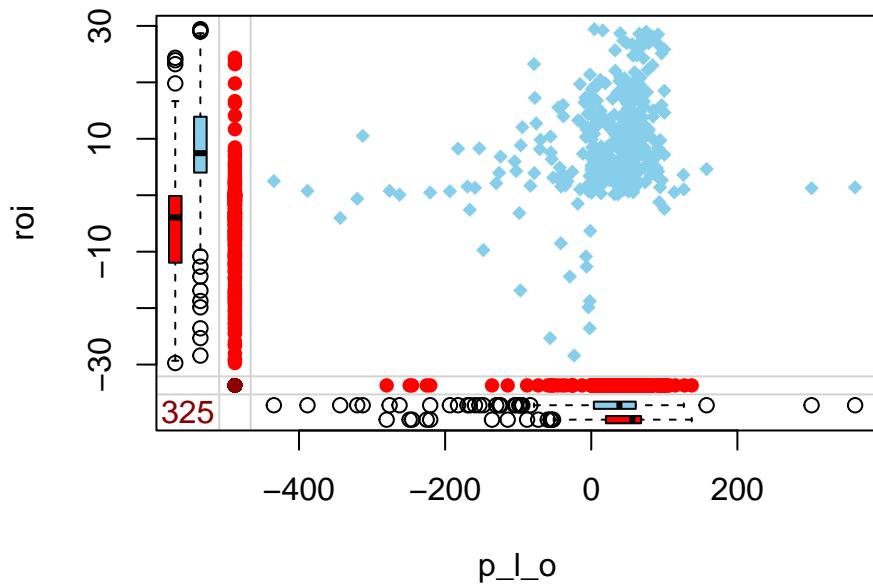
Ta cũng có kết luận tương tự cho `employees`:

```
marginplot(aida[, c("p_l_o", "employees")], pch = c(18, 16))
```



Đối với `roi`, ta cũng có kết luận tương tự:

```
marginplot(aida[, c("p_l_o", "roi")], pch = c(18, 16))
```



2 Xử lý dữ liệu khuyết

2.1 Xử lý MCAR

Nếu:

- % khuyết của dữ liệu là thấp (< 10%);
- cơ chế khuyết dữ liệu là MCAR

thì ta có thể xử lý bằng cách loại đi tất cả các quan sát có biến bị khuyết, chỉ sử dụng phần gồm tất cả các quan sát của tất cả các biến. Trong R, ta dùng hàm

```
na.omit(data)
```

Ví dụ: ta áp dụng các xử lý này cho dữ liệu aida:

```
aida_cc <- na.omit(aida)
glimpse(aida_cc)
```

```
## Rows: 338
## Columns: 15
## $ cid      <dbl> 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23~
## $ province <chr> "Milano", "Alessandria", "Vicenza", "Vicenza", "Trev~
## $ revenue   <dbl> 146019, 141401, 126609, 121147, 89497, 82429, 74621, 60774, 578~
## $ employees <dbl> 403, 304, 84, 132, 57, 59, 67, 102, 12, 104, 48, 93, 11, 65, 50~
## $ total_asset <dbl> 127722, 224664, 45174, 48912, 16707, 23785, 89794, 49069, 3358, ~
## $ profit    <dbl> 3341, -871, 3164, 1823, 339, 732, 5573, 4912, 65, 675, -525, 36~
## $ roa        <dbl> 16.75, 1.10, 10.22, 6.12, 3.79, 1.02, 9.63, 13.79, 4.90, 7.17, ~
## $ roe        <dbl> 4.56, -1.03, 13.12, 9.30, 5.93, 12.75, 17.25, 11.36, 2.31, 12.8~
## $ p_l_o     <dbl> 15.62, -35.38, 68.54, 60.86, 53.51, 301.52, 64.43, 72.60, 39.21~
## $ roi        <dbl> 29.19, 1.74, 10.91, 6.73, 4.71, 1.24, 16.84, 15.65, 5.90, 11.25~
## $ ros        <dbl> 14.51, 1.72, 3.64, 2.46, 0.71, 0.29, 11.58, 11.09, 0.28, 2.14, ~
## $ debt_equity <dbl> 0.00, 0.56, 0.46, 1.27, 1.36, 2.42, 0.59, 0.00, 0.00, 1.05, 0.6~
## $ cl_ta      <dbl> 1.00, 0.87, 0.56, 0.89, 1.00, 0.94, 0.94, 1.00, 1.00, 0.68, 0.7~
## $ l_ta       <dbl> 0.00, 0.13, 0.44, 0.11, 0.00, 0.06, 0.06, 0.00, 0.00, 0.32, 0.2~
## $ nwc        <dbl> 54151, 56141, 31643, 6754, 3988, 1467, 35589, 42851, 2431, 2094~
```

kết quả là một tập dữ liệu nhỏ hơn, chỉ chừ 338 quan sát, tức là 27% dữ liệu ban đầu. với bộ dữ liệu mới này, ta có thể tiến hành các quy trình xử lý khác nhau cho các nhiệm vụ khác nhau.

2.2 Xử lý MAR

Khi cơ chế khuyết dữ liệu là MAR (missing at random), ta áp dụng các thuật toán của multiple imputation để tạo dữ liệu mô phỏng để điền vào những vị trí khuyết của dữ liệu. Trong R, ta sử dụng hàm `mice()` được cung cấp bởi thư viện `mice`:

```
mice(data, m, method, printFlag = TRUE)
```

trong đó,

- `data`: bộ dữ liệu cần xử lý;
- `m`: số lần thực hiện mô phỏng dữ liệu;
- `method`: vector chứa tên các phương pháp multiple imputation dữ liệu tương ứng cho từng biến, bao gồm, "pmm" - predictive mean matching (dành cho biến dạng số); "norm" - Bayesian linear regression (dành cho biến dạng số); "logreg" - logistic regression imputation (thích hợp với biến nhị phân); "polyreg" - polytomous regression imputation (dành cho biến định tính không có thứ tự); "polr" -

proportional odds model (dành cho biến có thứ tự); các lựa chọn khác được trình bày trong trang trợ giúp của hàm (`help("mice")`).

Nếu ta chỉ muốn áp dụng 1 phương pháp duy nhất cho tất cả các biến (cùng dạng biến) thì đổi số `method` chỉ gồm 1 phần tử duy nhất tương ứng tên của phương pháp.

Ví dụ: ta áp dụng phương pháp "pmm" cho tất cả các biến dạng số trong dữ liệu `aida` (không chứa "cid" và "province"):

```
aida_numeric <- aida |> select(!c("cid", "province"))
```

hàm `mice()` được sử dụng như sau:

```
imp_aida <- mice(aida_numeric, method = "pmm", m = 6, printFlag = TRUE)
```

```
##  
## iter imp variable  
## 1 1 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 1 2 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 1 3 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 1 4 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 1 5 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 1 6 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 2 1 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 2 2 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 2 3 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 2 4 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 2 5 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 2 6 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 3 1 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 3 2 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 3 3 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 3 4 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 3 5 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 3 6 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 4 1 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 4 2 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 4 3 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 4 4 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 4 5 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 4 6 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 5 1 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 5 2 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 5 3 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 5 4 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 5 5 employees roa roe p_l_o roi ros debt_equity cl_ta  
## 5 6 employees roa roe p_l_o roi ros debt_equity cl_ta  
  
## Warning: Number of logged events: 1
```

Ở đây, ta thực hiện 6 lần tạo mẫu ($m = 6$). Kết quả là

```
imp_aida
```

```
## Class: mids  
## Number of multiple imputations: 6  
## Imputation methods:  
##     revenue   employees total_asset      profit          roa          roe        p_l_o
```

```

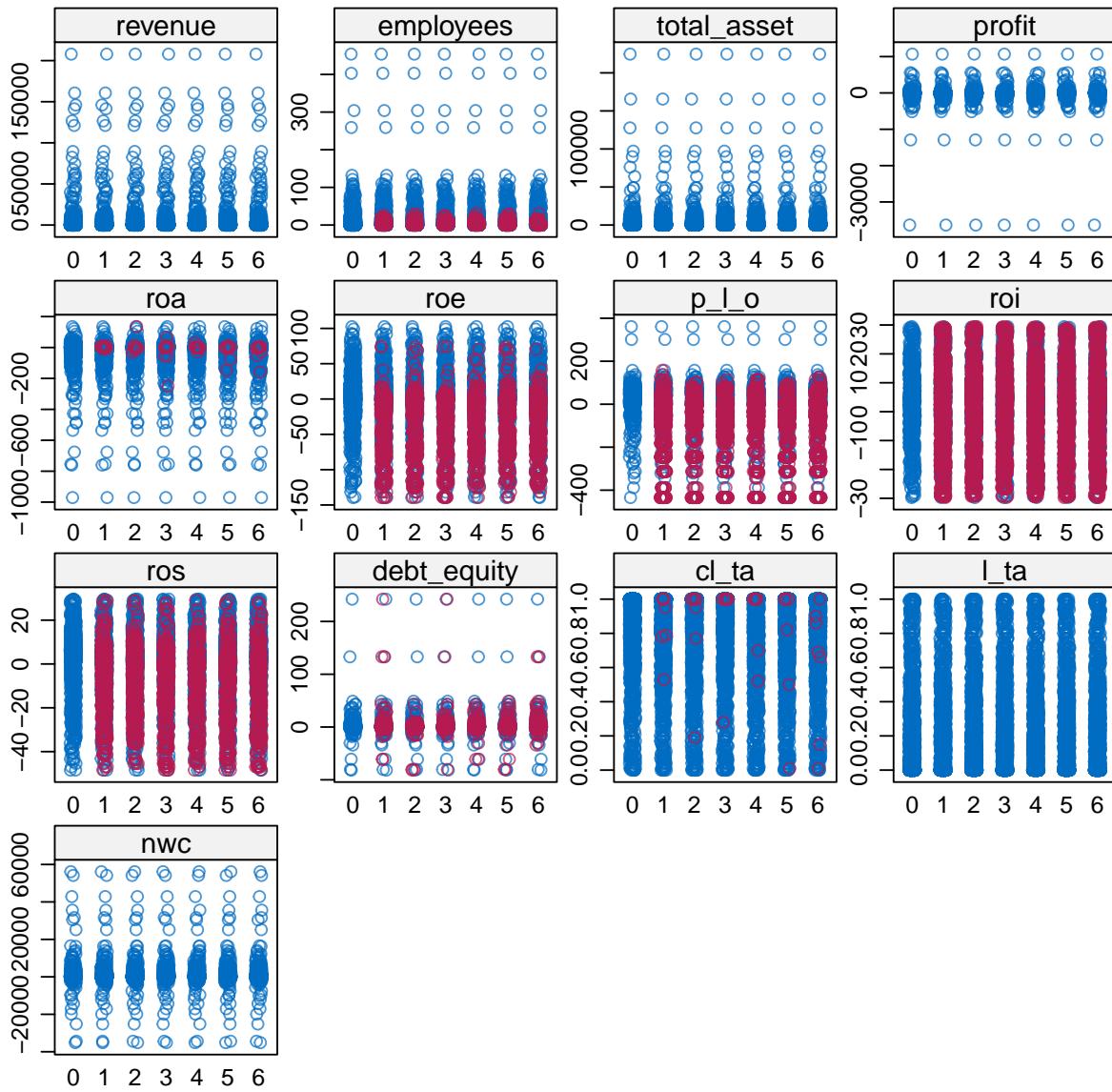
##      ""      "pmm"      ""      ""      "pmm"      "pmm"      "pmm"
##      roi      ros debt_equity      cl_ta      l_ta      nwc
##      "pmm"      "pmm"      "pmm"      "pmm"      ""      ""
## PredictorMatrix:
##      revenue employees total_asset profit roa roe p_l_o roi ros debt_equity
## revenue          0           1           1       1   1   1       1   1   1       1
## employees        1           0           1       1   1   1       1   1   1       1
## total_asset      1           1           0       1   1   1       1   1   1       1
## profit          1           1           1       0   1   1       1   1   1       1
## roa              1           1           1       1   0   1       1   1   1       1
## roe              1           1           1       1   1   0       1   1   1       1
##      cl_ta l_ta nwc
## revenue          1           0           1
## employees        1           0           1
## total_asset      1           0           1
## profit          1           0           1
## roa              1           0           1
## roe              1           0           1
## Number of logged events: 1
##      it im dep      meth out
## 1  0  0      collinear l_ta

```

Những biến không bị khuyết dữ liệu thì sẽ có phương pháp trống "".

Để kiểm tra xem liệu các điểm dữ liệu được tạo có phù hợp với dữ liệu hiện tại hay không, ta sử dụng hàm `stripplot()` để miêu tả dữ liệu được quan sát và dữ liệu được tái tạo

```
stripplot(imp_aida)
```



Ta nhận thấy các điểm (vòng tròn) màu đỏ (tức là các điểm dữ liệu được tái tạo) là trùng khớp với dữ liệu đã được quan sát. Do đó, kết quả thu được là hợp lý.

Thực hành 1: Thử các phương pháp imputation khác và nhận xét.

3 Multiple imputation và Hồi quy tuyến tính

Khi dữ liệu bị khuyết, ta cần phải kiểm tra % bị khuyết của dữ liệu, cũng như cơ chế khuyết của dữ liệu. Công việc này được hoàn thành ở các phần trước đó.

MCAR: Nếu ta nhận thấy % khuyết dữ liệu là thấp, và cơ chế khuyết là MCAR, khi đó, ta có thể loại bỏ các đối tượng quan sát có dữ liệu bị khuyết bằng hàm `na.omit()` và thực hiện các bước xây dựng mô hình trên dữ liệu mới.

MAR: Trong trường hợp, cơ chế khuyết là MAR, khi đó, ta cần phải tái tạo lại dữ liệu khuyết và thực hiện

các bước như sau:

Step 1. sử dụng hàm `mice()` để tái tạo lại dữ liệu khuyết

```
imp_data <- mice(data, method = "name method", m = M, maxit = N, seed = x,  
printFlag = FALSE)
```

Step 2. Sử dụng hàm `with()`, để thực hiện ước lượng mô hình hồi quy trên từng bộ dữ liệu được tái tạo:

```
md_imp_data <- with(imp_data, lm(y ~ x1 + x2 + ... , ...))
```

hoặc ta có thể dùng hàm `lm.mids()`:

```
md_imp_data <- lm.mids(y ~ x1 + x2 + ... , data = imp_data)
```

Step 3. Gộp chung các kết quả ước lượng mô hình trên từng bộ dữ liệu tái tạo, lại thành 1 kết quả chung, bằng việc sử dụng hàm `pool()`, trong khi, sử dụng hàm `pool.r.squared()` sẽ tổng hợp kết quả R^2 -adjusted.

```
summary(pool(md_imp_data))  
pool.r.squared(md_imp_pmm)
```

Step 4. Kết quả phân tích trên từng bộ dữ liệu tái tạo được lưu trữ trong danh sách `analysis` của dữ liệu, chẳng hạn

```
plot(md_imp_pmm$analyses[[j]])
```

với j từ 1 tới M là số lần tái tạo bộ dữ liệu. Ta có thể thực hiện việc đánh giá mô hình trên từng kết quả.

Ngoài ra, ta có thể sử dụng hàm

```
complete(imp_aida, action)
```

để truy xuất 1 bộ dữ liệu đã được tái tạo. Trong đó, `imp_aida` là tên kết quả dữ liệu được tái tạo; `action` là 1 số tương ứng với lần tái tạo thứ j . Sau đó, ta có thể xây dựng, ước lượng mô hình trên dữ liệu này (thay vì trên tất cả M tập dữ liệu tái tạo).

Ví dụ: ta thực hiện việc ước lượng mô hình dự đoán lợi nhuận (`profit`) dựa vào các biến `revenue`, `employees`, `ros` và `nwc`:

```
aida_1 <- aida[, c("profit", "revenue", "employees", "ros", "nwc")]
```

Thực hiện tái tạo kết quả trong 50 lần.

```
imp_aida <- mice(aida_1, method = "pmm", m = 50, printFlag = FALSE)  
imp_aida
```

```
## Class: mids  
## Number of multiple imputations: 50  
## Imputation methods:  
##   profit   revenue employees      ros      nwc  
##   ""        ""       "pmm"      "pmm"      "  
## PredictorMatrix:  
##   profit revenue employees ros nwc  
## profit      0      1      1  1  1  
## revenue     1      0      1  1  1  
## employees   1      1      0  1  1  
## ros         1      1      1  0  1  
## nwc         1      1      1  1  0
```

Áp dụng mô hình hồi quy cho 50 dữ liệu được tái tạo.

```
md1_imp_pmm <- with(imp_aida, lm(profit ~ revenue + employees + ros + nwc))
```

Kết quả ước lượng mô hình cho dữ liệu được tái tạo thứ 15

```
summary(md1_imp_pmm$analyses[[15]])
```

```
##  
## Call:  
## lm(formula = profit ~ revenue + employees + ros + nwc)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -31311.1    -32.5     34.8    143.6   11841.6  
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -59.059299  35.357042 -1.670 0.095098 .  
## revenue      0.027006   0.005009   5.392 8.34e-08 ***  
## employees   -16.035502   2.527189  -6.345 3.11e-10 ***  
## ros          9.163158   2.517279   3.640 0.000284 ***  
## nwc          0.113749   0.010273  11.072 < 2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 1140 on 1243 degrees of freedom  
## Multiple R-squared:  0.1538, Adjusted R-squared:  0.1511  
## F-statistic: 56.49 on 4 and 1243 DF, p-value: < 2.2e-16
```

Gộp chung 50 kết quả ước lượng mô hình hồi quy tuyến tính:

```
pool_md1_imp_pmm <- pool(md1_imp_pmm)  
summary(pool_md1_imp_pmm)
```

```
##           term   estimate   std.error statistic      df   p.value  
## 1 (Intercept) -38.50753868 38.161049215 -1.009080 715.4638 3.132776e-01  
## 2   revenue    0.02670926  0.005069512  5.268605 1185.5995 1.631640e-07  
## 3  employees   -15.86527237  2.598737539 -6.104992 1067.5679 1.437001e-09  
## 4      ros     6.31057227  3.274768779  1.927028 145.6026 5.592362e-02  
## 5      nwc     0.11293155  0.010499196 10.756210 1169.8117 8.511213e-26
```

Kết quả R^2 -adjusted chung của 50 mô hình:

```
pool.r.squared(md1_imp_pmm, adjusted = TRUE)
```

```
##           est      lo 95      hi 95      fmi  
## adj R^2 0.1464276 0.1109051 0.1850535 0.0485383
```

Thực hành 2: Áp dụng hàm `complete()` để lấy ra một dữ liệu tái tạo bất kỳ và áp dụng các bước xây dựng và đánh giá mô hình hồi quy tuyến tính.