BANA4040 Predictive Analytics

Phan Nu Quynh Huong

2025-03-12

Problem 1

In this problem, we will perform model selection in R. The data used for this problem is stored in "data.xlsx". In the related study, a personnel officer in a governmental agency administered four Assignment № 1 Page 1 newly developed aptitude tests to each of 25 applicants for entry-level clerical positions in the agency. For purpose of the study, all 25 applicants were accepted for positions irrespective of their test scores. After a probationary period, each applicant was rated for proficiency on the job. The data file include the job proficiency score (y, the first column) and scores on the four tests (refer as t1;t2;t3;t4 thereafter). As there are no column headers for this data file, so be sure to assign appropriate column headings for the dataframe after import.

Part 1

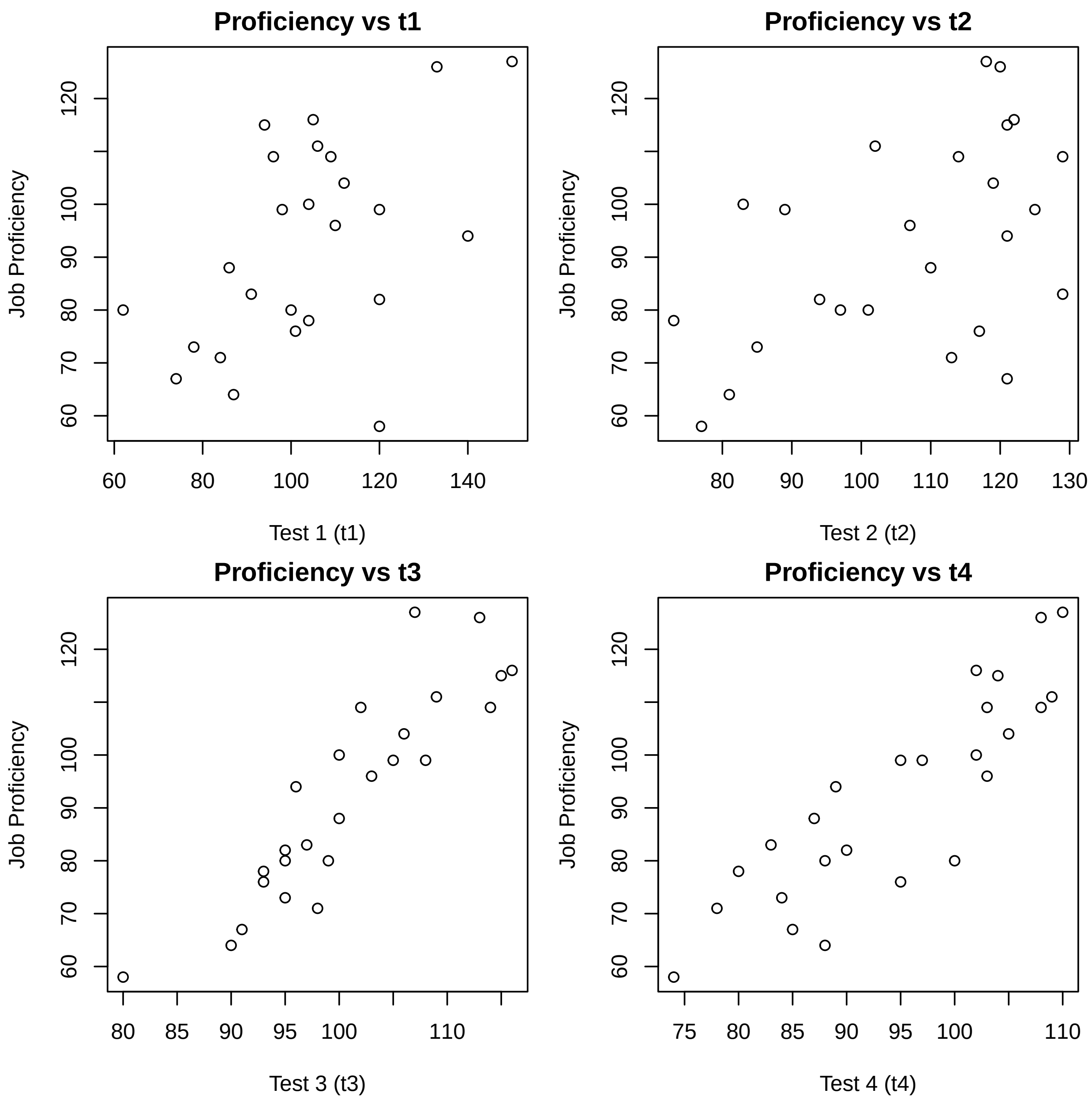
Graphical summaries: Before performing any of the model selection techniques, it is always a good idea to generate some graphical summaries of the data. For example, scatterplots of the response variable proficiency against each predictor individually. What do these plots suggest? ## Load Data

```
# Sử dụng thư viện readxl để đọc dữ liệu từ file Excel
library(readxl)

# Đọc dữ liệu từ file "data.xlsx"
# Tham số col_names = FALSE có nghĩa là file không có hàng tiêu đề nên chúng ta sẽ tự đặt tên sau này.
jobs <- read_excel("data.xlsx", col_names = FALSE)
#> New names:
#> * `` -> `...1`
#> * `` -> `...2`
#> * `` -> `...3`
#> * `` -> `...4`
#> * `` -> `...5`

# Gán tên cho các cột của dataframe:
# - "proficiency": độ chuyên môn của công việc
# - "t1", "t2", "t3", "t4": điểm số của các bài kiểm tra tương ứng.
colnames(jobs) <- c("proficiency", "t1", "t2", "t3", "t4")

# Hiển thị tóm tắt thống kê của dữ liệu (bao gồm min, max, median, mean, ...)
summary(jobs)
#>   proficiency          t1               t2               t3
#>  Min.   : 58.0   Min.   : 62.0   Min.   : 73.0   Min.   : 80.0
#>  1st Qu.: 78.0   1st Qu.: 91.0   1st Qu.: 94.0   1st Qu.: 95.0
#>  Median : 94.0   Median :104.0   Median :113.0   Median :100.0
#>  Mean   : 92.2   Mean   :103.4   Mean   :106.7   Mean   :100.8
#>  3rd Qu.:109.0   3rd Qu.:112.0   3rd Qu.:121.0   3rd Qu.:107.0
#>  Max.   :127.0   Max.   :150.0   Max.   :129.0   Max.   :116.0
#>        t4
#>  Min.   : 74.00
#>  1st Qu.: 87.00
#>  Median : 95.00
#>  Mean   : 94.68
#>  3rd Qu.:103.00
#>  Max.   :110.00
```

Visualize Data

We create scatterplots to visualize the relationship between job proficiency and each test score.

```
# Thiết lập kích thước biểu đồ để hiển thị rõ ràng
options(repr.plot.width = 20, repr.plot.height = 20)

# Chia vùng vẽ thành 2 hàng x 2 cột để hiển thị 4 biểu đồ cùng lúc.
# Tham số mfrow = c(2, 2) điều chỉnh lưới vẽ biểu đồ.
# Tham số mar = c(4, 4, 2, 1) thiết lập lề (margin) cho biểu đồ: dưới, trái, trên, phải.
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))

# Biểu đồ phân tán: điểm số Test 1 (t1) vs độ chuyên môn (proficiency)
plot(jobs$t1, jobs$proficiency,
     xlab = "Test 1 (t1)",      # Nhãn trục X: điểm của bài kiểm tra t1
     ylab = "Job Proficiency",  # Nhãn trục Y: độ chuyên môn
     main = "Proficiency vs t1")# Tiêu đề biểu đồ

# Biểu đồ phân tán: điểm số Test 2 (t2) vs độ chuyên môn
plot(jobs$t2, jobs$proficiency,
     xlab = "Test 2 (t2)",
     ylab = "Job Proficiency",
     main = "Proficiency vs t2")

# Biểu đồ phân tán: điểm số Test 3 (t3) vs độ chuyên môn
plot(jobs$t3, jobs$proficiency,
     xlab = "Test 3 (t3)",
     ylab = "Job Proficiency",
     main = "Proficiency vs t3")

# Biểu đồ phân tán: điểm số Test 4 (t4) vs độ chuyên môn
plot(jobs$t4, jobs$proficiency,
     xlab = "Test 4 (t4)",
     ylab = "Job Proficiency",
     main = "Proficiency vs t4")
```



```r
# Sau khi hoàn thành vẽ 4 biểu đồ, đặt lại vùng vẽ về mặc định (1 hàng, 1 cột)
par(mfrow = c(1, 1))
```

Conclusion

From a quick visual inspection, the plots suggest that both t3 and t4 have a relatively strong, positively sloped relationship with the proficiency measure (i.e., as the test scores go up, so does proficiency, and the points fall roughly along a line). By contrast, t1 and t2 still appear to be positively correlated with proficiency but not as strongly or as cleanly as t3 and t4.

In more detail:

t1 vs. proficiency: The points show a positive trend overall, but the relationship looks fairly scattered. There is some upward slope, yet more variability around the trend line compared to t3 and t4.

t2 vs. proficiency: Similar to t1, there is an upward trend but it is not as tight.

t3 vs. proficiency: The scatterplot indicates a strong, roughly linear relationship: as t3 increases, proficiency tends to increase in a fairly straight line.

t4 vs. proficiency: The points also exhibit a clear, strong positive trend, though perhaps with slightly more spread than t3.

In short, the plots suggest that t3 and t4 are likely to be the strongest individual predictors of proficiency, whereas t1 and t2 show weaker but still positive relationships.

Part 2

Performing all possible regressions: Follow the steps of the code in class, fit all possible regression models (4 predictors will generate 16 different models). For each model, record the following information for model selection purpose: p, number of parameters, R2, R2a,p,P RESSp, AICp, BICp, and Mallows Cp

```
model_stats <- function(fit, MSE_full){
  # fit: đối tượng lm (mô hình hồi quy)
  # MSE_full: Mean Squared Error của mô hình đầy đủ dùng để tính Mallows' Cp

  n <- length(fit$residuals)    # Số lượng quan sát
  p <- length(coef(fit))        # Số tham số (bao gồm intercept)

  # Tính hệ số xác định R² và R² hiệu chỉnh từ kết quả summary của mô hình
  r2     <- summary(fit)$r.squared
  adjr2  <- summary(fit)$adj.r.squared

  # Tính tổng bình phương sai số (SSE) của mô hình fit
  sse_p  <- sum(resid(fit)^2)

  # Tính chỉ số PRESS: PRESS = Σ[(e_i / (1 - h_ii))^2]
  # Trong đó: e_i là phần dư, h_ii là giá trị hat (đo lường ảnh hưởng của mỗi quan sát)
  hat    <- lm.influence(fit)$hat  # Lấy giá trị hat của mô hình
  res    <- resid(fit)            # Lấy phần dư của mô hình
  press  <- sum((res/(1 - hat))^2)

  # Tính AIC và BIC của mô hình
  aic_val <- AIC(fit)
  bic_val <- BIC(fit)

  # Tính Mallows' Cp với công thức: Cp = SSE_p / MSE_full - (n - 2*p)
  cp_val  <- sse_p / MSE_full - (n - 2*p)

  # Trả về vector chứa các chỉ số
  return(c(p, r2, adjr2, press, aic_val, bic_val, cp_val))
}
```

```
# Xây dựng mô hình hồi quy đầy đủ với tất cả các biến dự báo
lm_full <- lm(proficiency ~ t1 + t2 + t3 + t4, data = jobs)

# Tính tổng bình phương sai số (SSE) của mô hình đầy đủ
SSE_full <- sum(resid(lm_full)^2)
```

```r
# Số lượng quan sát trong dữ liệu
n <- nrow(jobs)

# Số tham số của mô hình đầy đủ (4 biến dự báo + intercept)
p_full <- length(coef(lm_full))

# Tính Mean Squared Error (MSE) của mô hình đầy đủ
MSE_full <- SSE_full / (n - p_full)
```

```r
# Tạo dataframe rỗng để lưu kết quả
results <- data.frame(
  Model    = character(),
  p        = numeric(),  # Số tham số (bao gồm intercept)
  R2       = numeric(),
  AdjR2    = numeric(),
  PRESS    = numeric(),
  AIC      = numeric(),
  BIC      = numeric(),
  Cp       = numeric(),
  stringsAsFactors = FALSE
)

# Danh sách các biến dự báo
preds <- c("t1", "t2", "t3", "t4")

# Duyệt qua các tập con của biến dự báo với số lượng biến từ 0 đến 4
for (k in 0:4) {
  # Lấy tất cả tổ hợp k phần tử từ vector preds
  subset_list <- combn(preds, k)

  if (k == 0) {
    # Trường hợp k = 0: mô hình chỉ có intercept (không có biến dự báo)
    form <- as.formula("proficiency ~ 1")
    fit <- lm(form, data = jobs)
    stats <- model_stats(fit, MSE_full)

    results <- rbind(
      results,
      data.frame(
        Model = "Intercept Only",
        p     = stats[1],
        R2    = stats[2],
        AdjR2 = stats[3],
        PRESS = stats[4],
        AIC   = stats[5],
        BIC   = stats[6],
        Cp    = stats[7],
        stringsAsFactors = FALSE
      )
    )

  } else {
    # Trường hợp k > 0: duyệt qua từng tổ hợp của các biến dự báo
    for (i in 1:ncol(subset_list)) {
      vars <- subset_list[, i]  # Lấy tập hợp các biến dự báo cho mô hình hiện tại

      # Tạo công thức hồi quy từ các biến được chọn
      form <- as.formula(paste("proficiency ~", paste(vars, collapse = " + ")))

      # Xây dựng mô hình hồi quy với công thức trên
      fit <- lm(form, data = jobs)
      stats <- model_stats(fit, MSE_full)

      # Thêm kết quả của mô hình vào dataframe results
      results <- rbind(
        results,
        data.frame(
          Model = paste(vars, collapse = " + "),
          p     = stats[1],
          R2    = stats[2],
          AdjR2 = stats[3],
          PRESS = stats[4],
          AIC   = stats[5],
          BIC   = stats[6],
          Cp    = stats[7],
          stringsAsFactors = FALSE
        )
      )
    }
  }
}
```

```r
# In kết quả các chỉ số thống kê của các mô hình con
knitr::kable(results, caption = "Bảng kết quả các mô hình dự báo")
```

Table 1: Bảng kết quả các mô hình dự báo

Model	p	R2	AdjR2	PRESS	AIC	BIC	Cp
Intercept Only	1	0.0000000	0.0000000	9824.2188	222.2491	224.6868	515.964627
t1	2	0.2646184	0.2326452	7791.5994	216.5649	220.2216	375.344689
t2	2	0.2470147	0.2142762	7991.0964	217.1563	220.8130	384.832454
t3	2	0.8047247	0.7962344	2064.5976	183.4155	187.0721	84.246496
t4	2	0.7558329	0.7452170	2548.6349	189.0015	192.6581	110.597414
t1 + t2	3	0.4641948	0.4154853	6444.0411	210.6495	215.5250	269.780029
t1 + t3	3	0.9329956	0.9269043	760.9744	158.6741	163.5496	17.112978
t1 + t4	3	0.8152656	0.7984716	2109.8967	184.0282	188.9037	80.565307
t2 + t3	3	0.8060733	0.7884436	2206.6460	185.2422	190.1177	85.519650
t2 + t4	3	0.7832923	0.7635916	2491.7979	188.0189	192.8944	97.797790
t3 + t4	3	0.8772573	0.8660988	1449.6001	173.8075	178.6830	47.153985
t1 + t2 + t3	4	0.9340931	0.9246779	831.1521	160.2613	166.3556	18.521465
t1 + t2 + t4	4	0.8453581	0.8232664	1885.8454	181.5830	187.6774	66.346500
t1 + t3 + t4	4	0.9615422	0.9560482	471.4520	146.7942	152.8886	3.727399
t2 + t3 + t4	4	0.8789698	0.8616797	1570.5610	175.4562	181.5506	48.231020
t1 + t2 + t3 + t4	5	0.9628918	0.9554702	518.9885	147.9011	155.2144	5.000000

Part 3

Which model is the "best"? Recall the criteria we explored, find the best model according to each criteria (R2 , R2a,p, P RESSp, AICp, BICp, and Mallows Cp). Which variable is suggested to be exluded? Are the results surprising?

Model Selection Summary

Best Model According to Each Criterion

Criterion	Best Model	Value
Highest R^2	`t1 + t2 + t3 + t4`	0.9629
Highest Adjusted R^2	`t1 + t3 + t4`	0.956
Lowest PRESS	`t1 + t3 + t4`	471.45
Lowest AIC	`t1 + t3 + t4`	146.79
Lowest BIC	`t1 + t3 + t4`	152.89
Mallows' Cp closest to p	`t1 + t3 + t4`	3.73

Conclusion: The model `t1 + t3 + t4` consistently stands out as the best or near-best across most criteria — especially Adjusted R^2, PRESS, AIC, BIC, and Mallows' Cp.

Suggested Variable to Exclude

- **Variable Excluded:** `t2`
- **Reason:** Models including `t2` do not improve (and sometimes worsen) the main selection criteria once `t1`, `t3`, and `t4` are already in the model.

Are the Results Surprising?

Not really! Based on earlier single-predictor analyses:

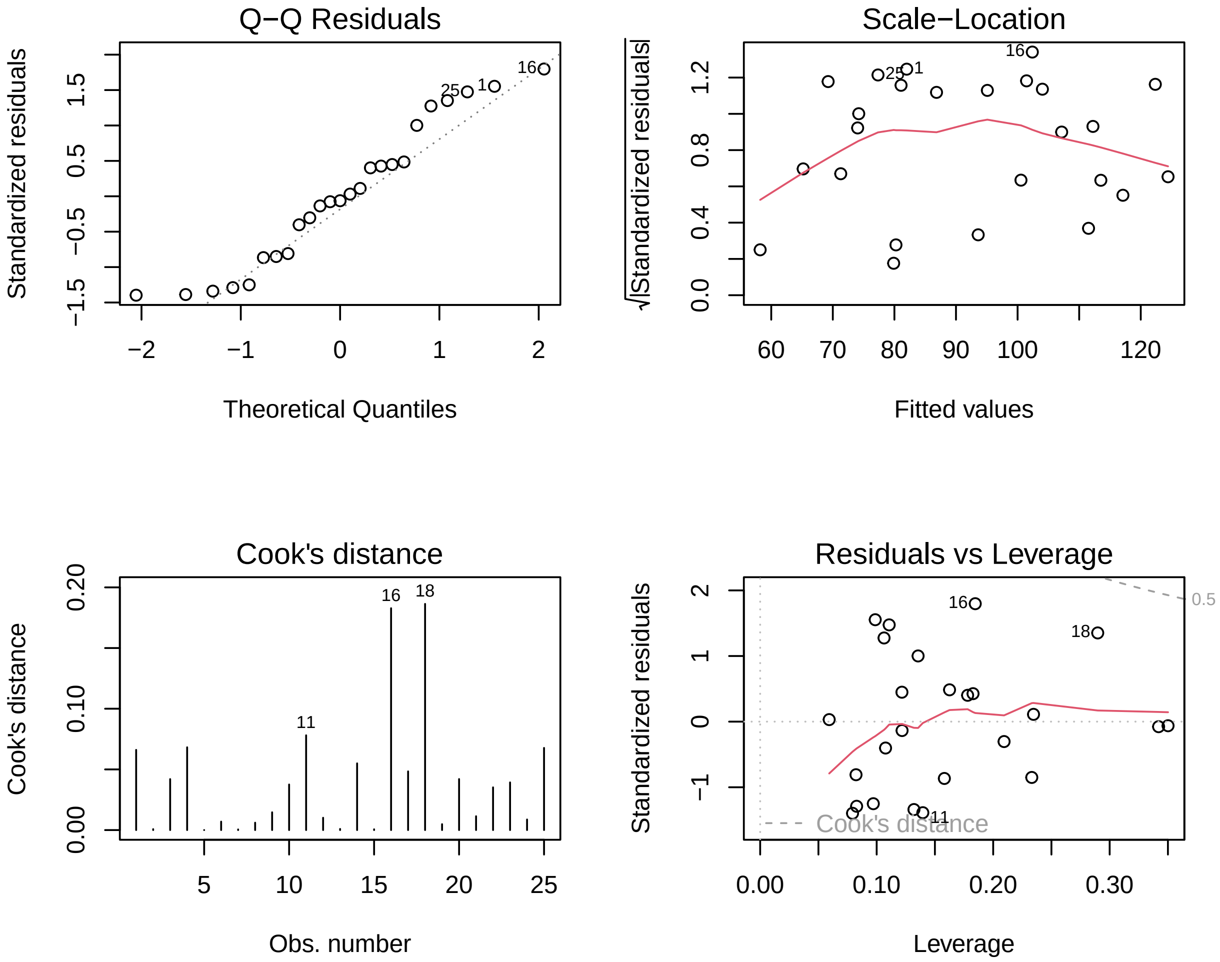
- `t3` and `t4` were the **strongest individual predictors**.
- `t1` was **weaker on its own**, but combined with `t3` and `t4`, the model improved substantially — likely because `t1` captures some additional variance not explained by the other two.
- `t2` never seemed particularly strong and still **does not help** when `t3` and `t4` are already present.

Final Conclusion: The best balance of simplicity and predictive accuracy is the **three-predictor model** `t1 + t3 + t4` — excluding `t2`.

Part 4

Fitting the model and model diagnostic For the best model according to R2a,p, fit the corresponding regression model and assess whether model assumptions are met. Do we have a good fit for the model?

```
options(repr.plot.width = 7, repr.plot.height = 6)
par(mfrow = c(2, 2))
# Assuming the data frame is 'jobs' with columns proficiency, t1, t2, t3, t4:
best_model <- lm(proficiency ~ t1 + t3 + t4, data = jobs)
summary(best_model)
#>
#> Call:
#> lm(formula = proficiency ~ t1 + t3 + t4, data = jobs)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -5.4579 -3.1563 -0.2057  1.8070  6.6083
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept) -124.20002    9.87406 -12.578 3.04e-11 ***
#> t1             0.29633    0.04368   6.784 1.04e-06 ***
#> t3             1.35697    0.15183   8.937 1.33e-08 ***
#> t4             0.51742    0.13105   3.948 0.000735 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 4.072 on 21 degrees of freedom
#> Multiple R-squared:  0.9615, Adjusted R-squared:  0.956
#> F-statistic:   175 on 3 and 21 DF,  p-value: 5.16e-15
# plot(best_model, which = 1)  # Residuals vs Fitted
plot(best_model, which = 2)  # Normal Q-Q plot
plot(best_model, which = 3)  # Scale-Location plot (spread vs. fitted)
plot(best_model, which = 4)  # Cook's distance
plot(best_model, which = 5)  # Residuals vs Leverage
```



Conclusion

From the four diagnostic plots, yes, the model looks to be fitting reasonably well overall. Here are some more detailed observations:

Q–Q Plot The residuals lie fairly close to the diagonal, indicating that the normality assumption is not severely violated. There is a bit of deviation in the upper tail (observations such as #16, #19, #25), but it's not drastic for a real-world data set.

Scale–Location (Spread vs. Fitted) Plot Although the red loess line isn't perfectly flat, the variation in residuals doesn't show a strong funnel shape or any major nonconstant variance. A slight rise or fall in the red line can happen; as long as it's not extremely pronounced, homoscedasticity is not a big concern.

Cook's Distance Observations #11, #16, and #18 stand out somewhat, but their Cook's distance (roughly 0.2 at most) is still well below the usual "rule-of-thumb" cutoff of 1. This suggests no single point is exerting a dramatically outsized influence on the regression fit.

Residuals vs. Leverage Points #16 and #18 have relatively higher leverage (meaning these points lie a bit farther out in predictor space), but the standardized residuals are not huge. The dashed lines for Cook's distance confirm they're not excessively influential overall.

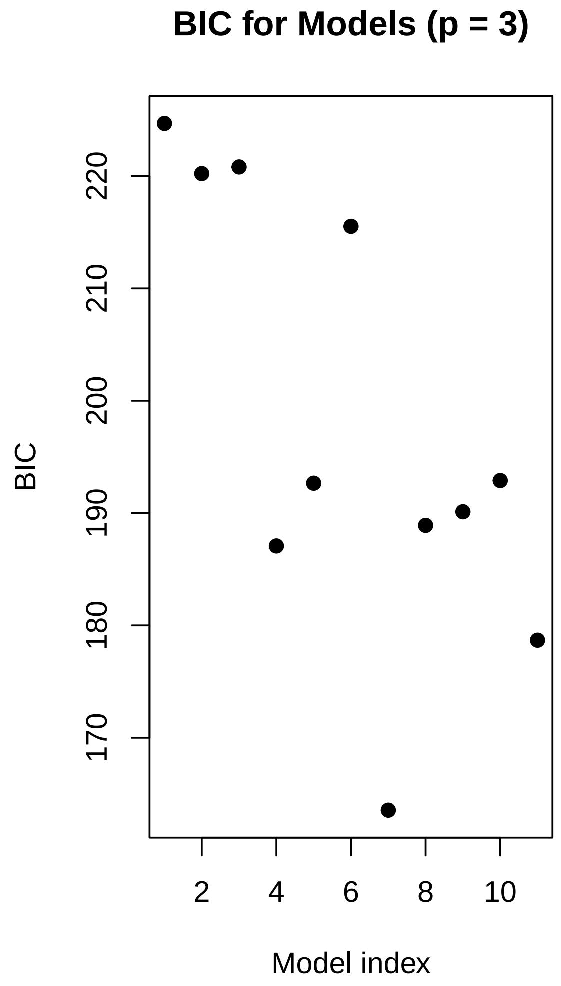
Putting this all together, there are no severe violations of normality, constant variance, or linearity, and no single outlier is dominating the regression. Hence, we can conclude that the model assumptions are reasonably satisfied and we do have a good fit.

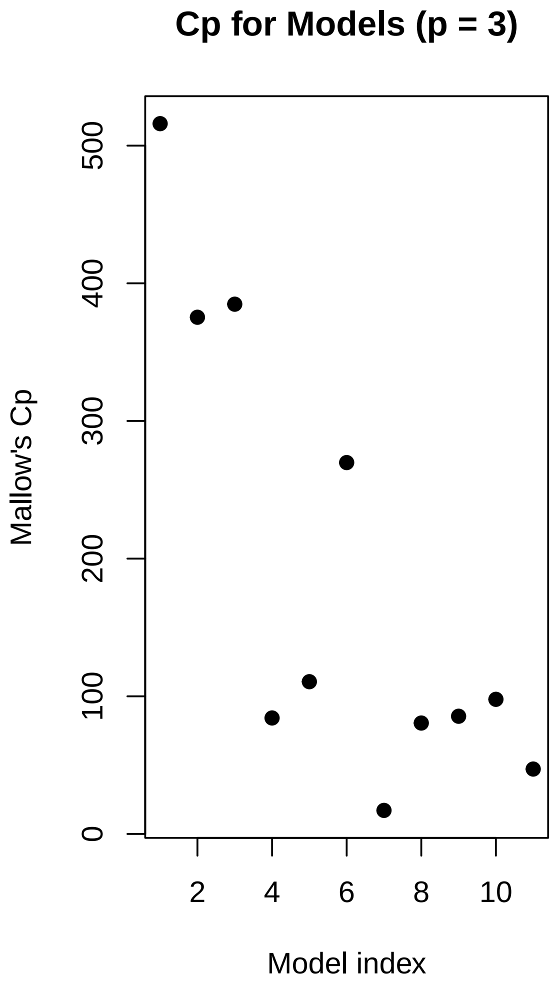
Part 5

Which model is the best with a given number of predictors? Suppose we only want to consider models with two predictors (or less). (Perhaps due to budget cuts the agency cannot administer as many aptitude tests). Suppose we want to finnd the “best” model with two predictors (or less), using the BICp criterion. Hints: use the plot and identify function. Do the same for Mallow’s Cp criterion.

```r
# 1. Suppose we have the following data for models with p   3:
modelNames <- c("Intercept only",
                "t1", "t2", "t3", "t4",
                "t1 + t2", "t1 + t3", "t1 + t4",
                "t2 + t3", "t2 + t4", "t3 + t4")

p <- c(1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3)  # total parameters (intercept + #predictors)

BICvals <- c(224.6868, 220.2216, 220.8130, 187.0721, 192.6581,
             215.5250, 163.5496, 188.9037, 190.1177, 192.8944, 178.6830)

Cpvals  <- c(515.9646, 375.3447, 384.8325, 84.2465, 110.5974,
             269.7800, 17.1130, 80.5653, 85.5196, 97.7978, 47.1540)
```

```r
par(mfrow = c(1,2))  # Chia vùng vẽ thành 1 hàng, 2 cột
options(repr.plot.width = 15, repr.plot.height = 15)

# Vẽ BIC
plot(BICvals,
     xlab = "Model index",
     ylab = "BIC",
     main = "BIC for Models (p   3)",
     pch = 19)

# Vẽ Cp
plot(Cpvals,
     xlab = "Model index",
     ylab = "Mallow's Cp",
     main = "Cp for Models (p   3)",
     pch = 19)

# Reset lại vùng vẽ về mặc định (tránh ảnh hưởng các lệnh sau)
par(mfrow = c(1,1))
identify(BICvals, labels = modelNames)
#> integer(0)
identify(Cpvals, labels = modelNames)
```





```
#> integer(0)
```

Model Selection with Two or Fewer Predictors

1. Identify the Candidate Models

For models with **two predictors or fewer**, the possible models include:

- **No predictors (Intercept-only):** `p = 1`
- **One predictor:** `t1`, `t2`, `t3`, or `t4` (`p = 2`)
- **Two predictors:** `t1 + t2`, `t1 + t3`, `t1 + t4`, `t2 + t3`, `t2 + t4`, `t3 + t4` (`p = 3`)

The table below provides **BIC** and **Mallows' Cp** for each model.

2. Best Model by BIC

Considering only models with **p** **3**, the smallest **BIC** is for:

- **Model:** `t1 + t3`
- **BIC:** 163.55

This is **substantially lower** than any other model with one or two predictors (e.g., `t3` alone has **BIC 187.07**).

Conclusion: Using the **BIC** criterion, the best choice is the model **`t1 + t3`**.

---

3. Best Model by Mallows' Cp

Similarly, examining **Mallows' Cp** values for models with **p** **3**, the smallest value is again for:

- **Model:** `t1 + t3`
- **Cp:** 17.11

This is **noticeably smaller** than the next-best single- or two-predictor model.

Conclusion: By **Mallows' Cp**, the best model (with up to two predictors) is also **`t1 + t3`**.

---

4. Final Conclusion

Whether judged by **BIC** or **Mallows' Cp**, if we are constrained to using **two or fewer predictors**, the **`t1 + t3`** model is the **clear winner**.

Part 6

Valdation: One of the validation techniques is to compare the SSEp and P RESSp of the models under considerations. Ideally, these two values should be close for the same model. Let us use model 11 as an example. First, we fit this model using lm(), and obtain the SSEp and P RESSp statistics. The SSEp can be computed based on the component sigma in the output of summary function. And the P RESSp statistic of a model can be computed using function press() in the DAAG package. Make sure to install the package before you run the commands.

```r
#-----------------------------------------------
# 1) Chuẩn bị dữ liệu và gói cần thiết
#-----------------------------------------------
# install.packages("DAAG") # cài nếu chưa có
library(DAAG)
# install.packages("readxl") # cài nếu cần đọc Excel
library(readxl)

# Giả sử bạn đã đọc dữ liệu vào jobs_data:
jobs_data <- read_excel("data.xlsx", col_names = FALSE)
```

```r
#> New names:
#> * `` -> `...1`
#> * `` -> `...2`
#> * `` -> `...3`
#> * `` -> `...4`
#> * `` -> `...5`
colnames(jobs_data) <- c("y", "t1", "t2", "t3", "t4")

#-----------------------------------------------
# 2) Tạo danh sách tất cả các tổ hợp biến
#-----------------------------------------------
predictors <- c("t1", "t2", "t3", "t4")

# Hàm combn() sẽ lấy tất cả tổ hợp k phần tử từ danh sách,
# lapply(...) để duyệt k = 1..4, rồi unlist(..., recursive=FALSE) gộp lại thành một list.
subset_list <- unlist(
  lapply(1:length(predictors), function(k) {
    combn(predictors, k, simplify = FALSE)
  }),
  recursive = FALSE
)

#-----------------------------------------------
# 3) Khởi tạo khung kết quả
#-----------------------------------------------
results <- data.frame(
  model = character(),
  SSE   = numeric(),
  PRESS = numeric(),
  stringsAsFactors = FALSE
)

# Số dòng (số quan sát)
n <- nrow(jobs_data)

#-----------------------------------------------
# 4) Vòng lặp: ước lượng từng mô hình, tính SSE & PRESS
#-----------------------------------------------
for (vars in subset_list) {
  # Tạo chuỗi công thức, ví dụ "y ~ t1 + t3" v.v.
  formula_str <- paste("y ~", paste(vars, collapse = " + "))

  # Fit mô hình
  fit <- lm(as.formula(formula_str), data = jobs_data)

  # Tính SSE:
  # - Cách 1: Dùng công thức SSE = sigma^2 * (n - p),
  #           trong đó p = số tham số (kể cả intercept).
  p <- length(coef(fit))               # số tham số ước lượng
  rse <- summary(fit)$sigma            # residual standard error
  SSEp <- rse^2 * (n - p)

  # - Cách 2 (tương đương): SSEp = sum(resid(fit)^2)
  # SSEp <- sum(resid(fit)^2)

  # Tính PRESS:
  PRESSp <- press(fit)  # từ gói DAAG

  # Lưu kết quả
  results <- rbind(results, data.frame(
    model = formula_str,
    SSE   = SSEp,
    PRESS = PRESSp
  ))
}

#-------------------------------------------------
# 5) Xem kết quả
#-------------------------------------------------
knitr::kable(results, caption = "Bảng kết quả")
```

Table 3: Bảng kết quả

model	SSE	PRESS
y ~ t1	6658.1453	7791.5994
y ~ t2	6817.5291	7991.0964
y ~ t3	1768.0228	2064.5976
y ~ t4	2210.6887	2548.6349
y ~ t1 + t2	4851.1799	6444.0411
y ~ t1 + t3	606.6574	760.9744
y ~ t1 + t4	1672.5853	2109.8967
y ~ t2 + t3	1755.8127	2206.6460
y ~ t2 + t4	1962.0716	2491.7979
y ~ t3 + t4	1111.3126	1449.6001
y ~ t1 + t2 + t3	596.7207	831.1521
y ~ t1 + t2 + t4	1400.1275	1885.8454
y ~ t1 + t3 + t4	348.1970	471.4520
y ~ t2 + t3 + t4	1095.8078	1570.5610
y ~ t1 + t2 + t3 + t4	335.9775	518.9885

Part 7

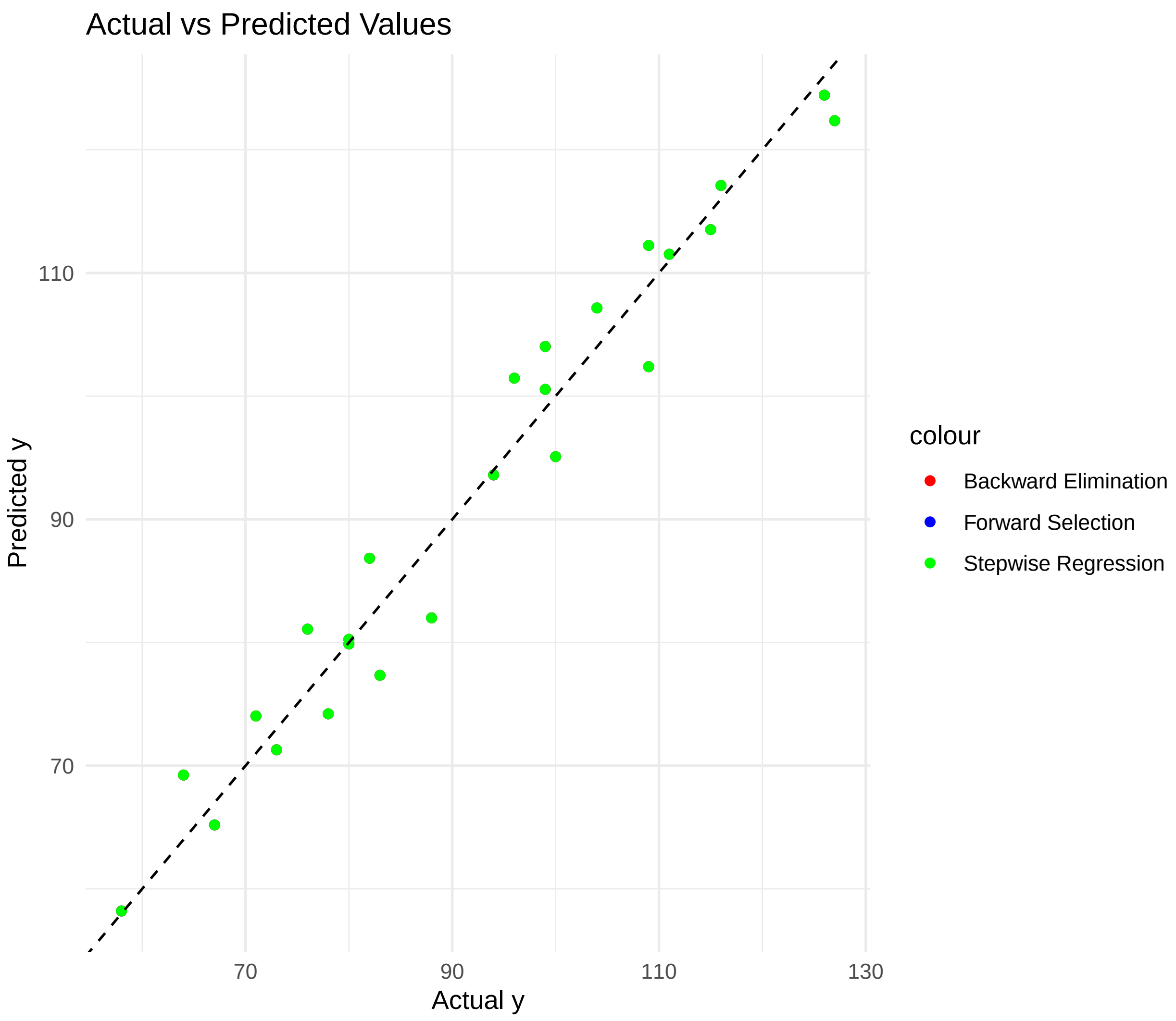
Automated search procedure: Next we explore forward selection, backward elimination, and stepwise regression. These are performed using the step() function. In forward selection, to start from a model with just the intercept and end with a model with all variables. Do the same with backward elimination. What model(s) is chosen with these procedures? It is important to bear in mind that the same model is not always going to be selected by all of these procedures, and the choice of starting model might impact the result.

```
library(readxl)
library(ggplot2)
library(MASS)

# Đọc dữ liệu
jobs_data <- read_excel("data.xlsx", col_names = FALSE)
```

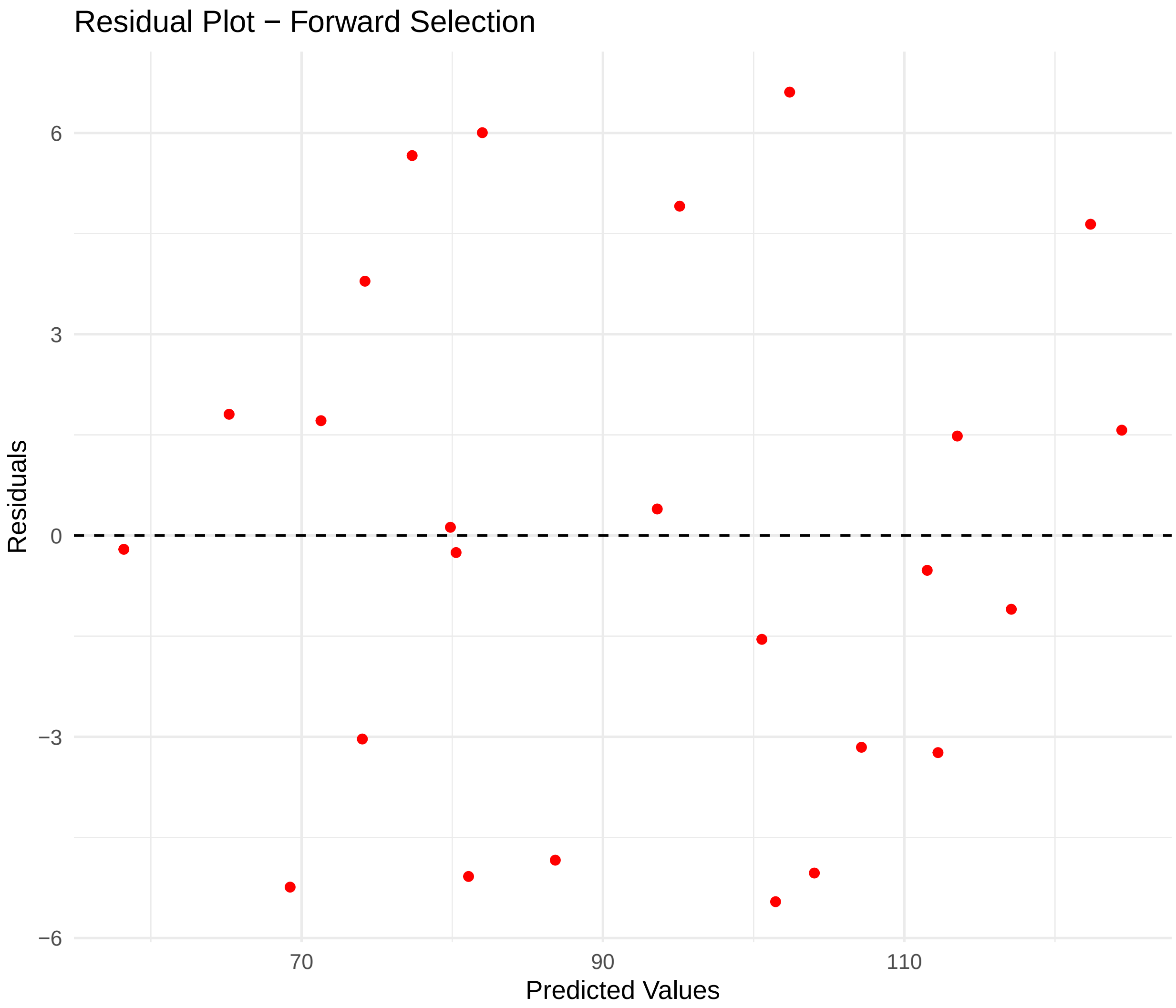
```
#> New names:
#> * `` -> `...1`
#> * `` -> `...2`
#> * `` -> `...3`
#> * `` -> `...4`
#> * `` -> `...5`
colnames(jobs_data) <- c("y", "t1", "t2", "t3", "t4")

# Mô hình null (chỉ có intercept) và mô hình đầy đủ
null_model <- lm(y ~ 1, data = jobs_data)
full_model <- lm(y ~ t1 + t2 + t3 + t4, data = jobs_data)

# Forward Selection
forward_model <- step(null_model, scope = list(lower = null_model, upper = full_model), direction = "fo
#> Start:  AIC=149.3
#> y ~ 1
#>
#>        Df Sum of Sq    RSS    AIC
#> + t3    1    7286.0 1768.0 110.47
#> + t4    1    6843.3 2210.7 116.06
#> + t1    1    2395.9 6658.1 143.62
#> + t2    1    2236.5 6817.5 144.21
#> <none>              9054.0 149.30
#>
#> Step:  AIC=110.47
#> y ~ t3
#>
#>        Df Sum of Sq     RSS     AIC
#> + t1    1   1161.37  606.66  85.727
#> + t4    1    656.71 1111.31 100.861
#> <none>              1768.02 110.469
#> + t2    1     12.21 1755.81 112.295
#>
#> Step:  AIC=85.73
#> y ~ t3 + t1
#>
#>        Df Sum of Sq    RSS    AIC
#> + t4    1   258.460 348.20 73.847
#> <none>              606.66 85.727
#> + t2    1     9.937 596.72 87.314
#>
#> Step:  AIC=73.85
#> y ~ t3 + t1 + t4
#>
#>        Df Sum of Sq    RSS    AIC
#> <none>              348.20 73.847
#> + t2    1     12.22 335.98 74.954

# Backward Elimination
backward_model <- step(full_model, direction = "backward")
#> Start:  AIC=74.95
#> y ~ t1 + t2 + t3 + t4
#>
```

```
#>        Df Sum of Sq     RSS     AIC
#> - t2    1     12.22  348.20  73.847
#> <none>               335.98  74.954
#> - t4    1    260.74  596.72  87.314
#> - t1    1    759.83 1095.81 102.509
#> - t3    1   1064.15 1400.13 108.636
#>
#> Step:  AIC=73.85
#> y ~ t1 + t3 + t4
#>
#>        Df Sum of Sq     RSS     AIC
#> <none>               348.20  73.847
#> - t4    1    258.46  606.66  85.727
#> - t1    1    763.12 1111.31 100.861
#> - t3    1   1324.39 1672.59 111.081

# Stepwise Regression
stepwise_model <- step(null_model, scope = list(lower = null_model, upper = full_model), direction = "bo
#> Start:  AIC=149.3
#> y ~ 1
#>
#>        Df Sum of Sq    RSS    AIC
#> + t3    1    7286.0 1768.0 110.47
#> + t4    1    6843.3 2210.7 116.06
#> + t1    1    2395.9 6658.1 143.62
#> + t2    1    2236.5 6817.5 144.21
#> <none>              9054.0 149.30
#>
#> Step:  AIC=110.47
#> y ~ t3
#>
#>        Df Sum of Sq    RSS     AIC
#> + t1    1    1161.4  606.7  85.727
#> + t4    1     656.7 1111.3 100.861
#> <none>              1768.0 110.469
#> + t2    1      12.2 1755.8 112.295
#> - t3    1    7286.0 9054.0 149.302
#>
#> Step:  AIC=85.73
#> y ~ t3 + t1
#>
#>        Df Sum of Sq    RSS     AIC
#> + t4    1     258.5  348.2  73.847
#> <none>               606.7  85.727
#> + t2    1       9.9  596.7  87.314
#> - t1    1    1161.4 1768.0 110.469
#> - t3    1    6051.5 6658.1 143.618
#>
#> Step:  AIC=73.85
#> y ~ t3 + t1 + t4
#>
#>        Df Sum of Sq     RSS     AIC
#> <none>               348.20  73.847
#> + t2    1     12.22  335.98  74.954
#> - t4    1    258.46  606.66  85.727
#> - t1    1    763.12 1111.31 100.861
#> - t3    1   1324.39 1672.59 111.081

# Thêm dự đoán vào dữ liệu
jobs_data$pred_forward <- predict(forward_model, jobs_data)
jobs_data$pred_backward <- predict(backward_model, jobs_data)
jobs_data$pred_stepwise <- predict(stepwise_model, jobs_data)
```

```
# Biểu đồ so sánh giá trị thực tế và dự đoán
ggplot(jobs_data, aes(x = y)) +
  geom_point(aes(y = pred_forward, color = "Forward Selection")) +
  geom_point(aes(y = pred_backward, color = "Backward Elimination")) +
  geom_point(aes(y = pred_stepwise, color = "Stepwise Regression")) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  labs(title = "Actual vs Predicted Values", x = "Actual y", y = "Predicted y") +
  theme_minimal() +
  scale_color_manual(values = c("red", "blue", "green"))
```



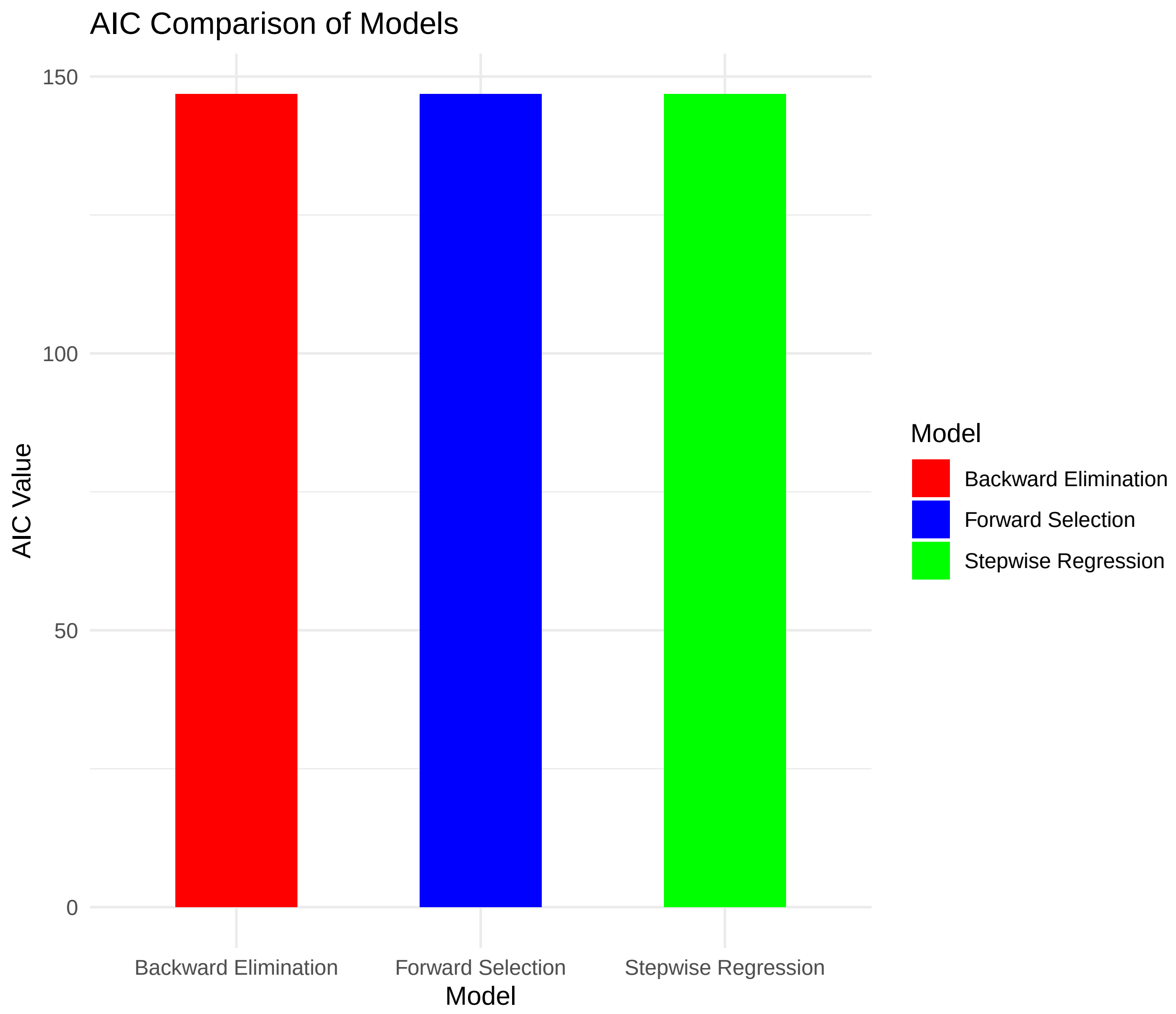
```r
# Biểu đồ Residuals để kiểm tra mô hình
ggplot(jobs_data, aes(x = pred_forward, y = residuals(forward_model))) +
  geom_point(color = "red") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residual Plot - Forward Selection", x = "Predicted Values", y = "Residuals") +
  theme_minimal()
```



So sánh AIC của các mô hình

```r
aic_values <- data.frame(
  Model = c("Forward Selection", "Backward Elimination", "Stepwise Regression"),
  AIC = c(AIC(forward_model), AIC(backward_model), AIC(stepwise_model))
)
```

```
ggplot(aic_values, aes(x = Model, y = AIC, fill = Model)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(title = "AIC Comparison of Models", y = "AIC Value") +
  theme_minimal() +
  scale_fill_manual(values = c("red", "blue", "green"))
```



```
  # In summary các mô hình
cat("Forward Selection Model Summary:\n")
#> Forward Selection Model Summary:
print(summary(forward_model))
#>
#> Call:
#> lm(formula = y ~ t3 + t1 + t4, data = jobs_data)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -5.4579 -3.1563 -0.2057  1.8070  6.6083
#>
#> Coefficients:
```

```
#>              Estimate Std. Error t value Pr(>|t|)    
#> (Intercept) -124.20002    9.87406 -12.578 3.04e-11 ***
#> t3             1.35697    0.15183   8.937 1.33e-08 ***
#> t1             0.29633    0.04368   6.784 1.04e-06 ***
#> t4             0.51742    0.13105   3.948 0.000735 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> Residual standard error: 4.072 on 21 degrees of freedom
#> Multiple R-squared:  0.9615, Adjusted R-squared:  0.956 
#> F-statistic:   175 on 3 and 21 DF,  p-value: 5.16e-15

cat("\nBackward Elimination Model Summary:\n")
#> 
#> Backward Elimination Model Summary:
print(summary(backward_model))
#> 
#> Call:
#> lm(formula = y ~ t1 + t3 + t4, data = jobs_data)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -5.4579 -3.1563 -0.2057  1.8070  6.6083 
#> 
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)    
#> (Intercept) -124.20002    9.87406 -12.578 3.04e-11 ***
#> t1             0.29633    0.04368   6.784 1.04e-06 ***
#> t3             1.35697    0.15183   8.937 1.33e-08 ***
#> t4             0.51742    0.13105   3.948 0.000735 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> Residual standard error: 4.072 on 21 degrees of freedom
#> Multiple R-squared:  0.9615, Adjusted R-squared:  0.956 
#> F-statistic:   175 on 3 and 21 DF,  p-value: 5.16e-15

cat("\nStepwise Regression Model Summary:\n")
#> 
#> Stepwise Regression Model Summary:
print(summary(stepwise_model))
#> 
#> Call:
#> lm(formula = y ~ t3 + t1 + t4, data = jobs_data)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -5.4579 -3.1563 -0.2057  1.8070  6.6083 
#> 
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)    
#> (Intercept) -124.20002    9.87406 -12.578 3.04e-11 ***
#> t3             1.35697    0.15183   8.937 1.33e-08 ***
#> t1            0.29633    0.04368   6.784 1.04e-06 ***
#> t4            0.51742    0.13105   3.948 0.000735 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 4.072 on 21 degrees of freedom
#> Multiple R-squared:  0.9615, Adjusted R-squared:  0.956
#> F-statistic:   175 on 3 and 21 DF,  p-value: 5.16e-15

# In bảng AIC
aic_values <- data.frame(
  Model = c("Forward Selection", "Backward Elimination", "Stepwise Regression"),
  AIC = c(AIC(forward_model), AIC(backward_model), AIC(stepwise_model))
)
knitr::kable(aic_values)
```

Model	AIC
Forward Selection	146.7942
Backward Elimination	146.7942
Stepwise Regression	146.7942

Part 8

Searching all models based on specified conditions: Here we introduce a more powerful model searching function: regsubsets() from the leaps package. This function allows user to specify certain predictors that must always be considered, certain predictors that must always be excluded, and the maximum number of predictors to be considered. A drawback of this function is that it only considers R2 a,p, Mallow's Cp, BICp. Find the best models in terms of these criteria.

```
# install.packages("leaps")   # nếu chưa cài
library(leaps)
library(readxl)
# Ví dụ: Dữ liệu có cột: y, t1, t2, t3, t4
jobs_data <- read_excel("data.xlsx")
colnames(jobs_data) <- c("y", "t1", "t2", "t3", "t4")
# Tìm mô hình tốt nhất theo mọi số biến (từ 1 đến 4) trong {t1, t2, t3, t4}
# nbest=1 nghĩa là chỉ lấy 1 mô hình tốt nhất cho mỗi số biến.
fit_sub <- regsubsets(y ~ t1 + t2 + t3 + t4,
                      data = jobs_data,
                      method = "exhaustive",  # duyệt tất cả tổ hợp
                      nbest = 1,
                      nvmax = 4)              # tối đa 4 biến (có thể ít hơn)
summary_sub <- summary(fit_sub)

# summary_sub$which  => Ma trận TRUE/FALSE cho biết biến nào được chọn
# summary_sub$rsq    => R^2
# summary_sub$adjr2  => Adjusted R^2
# summary_sub$cp     => Mallow's Cp
# summary_sub$bic    => BIC
# 1) Mô hình tốt nhất theo Adjusted R^2 (lớn nhất)
best_adjr2_index <- which.max(summary_sub$adjr2)
```

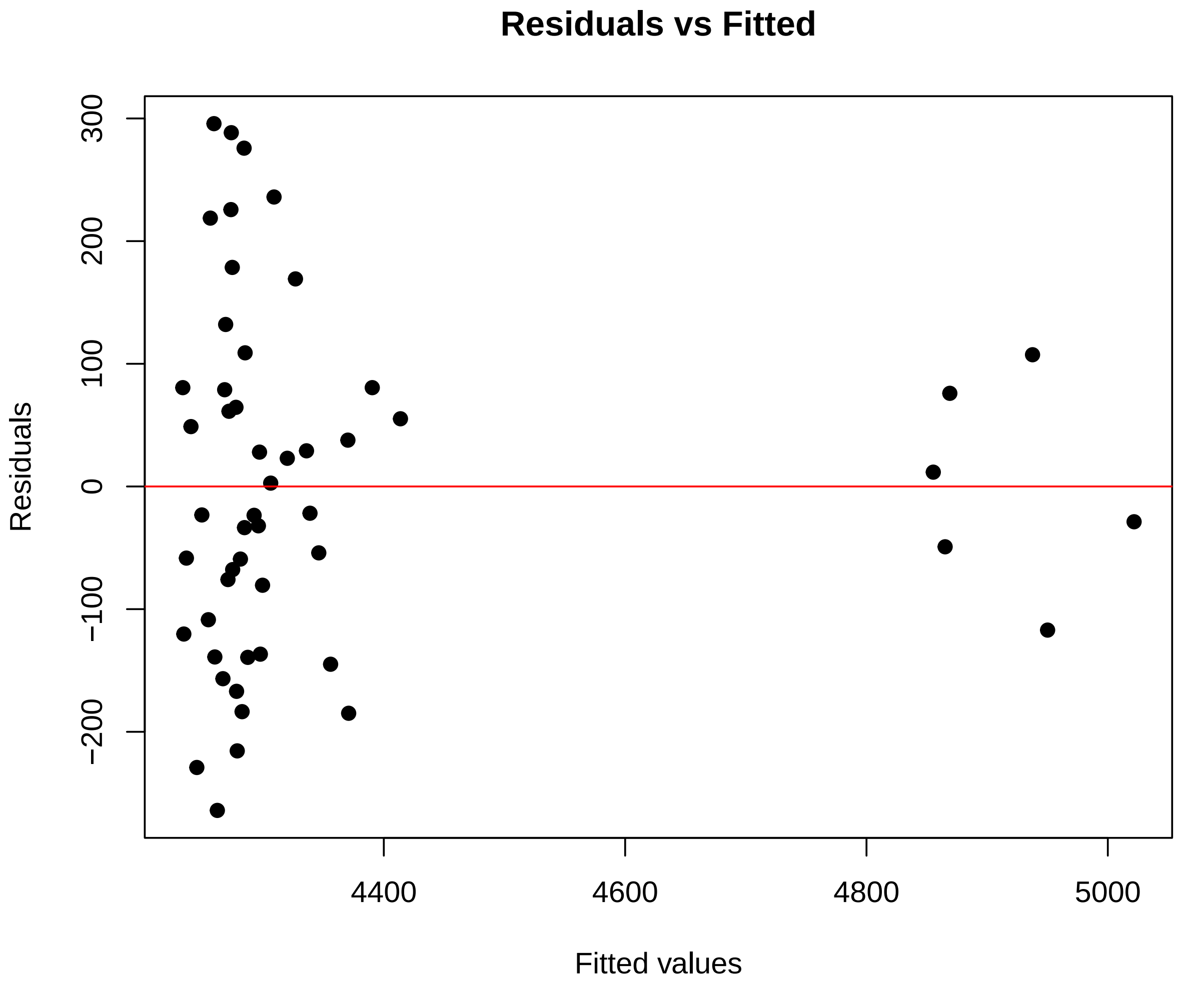
```r
# 2) Mô hình tốt nhất theo Mallow's Cp (nhỏ nhất)
best_cp_index <- which.min(summary_sub$cp)

# 3) Mô hình tốt nhất theo BIC (nhỏ nhất)
best_bic_index <- which.min(summary_sub$bic)

# Kiểm tra biến nào được chọn ở mô hình best_adjr2_index
summary_sub$which[best_adjr2_index, ]
#> (Intercept)          t1          t2          t3          t4
#>        TRUE        TRUE       FALSE        TRUE        TRUE
summary_sub$which[best_cp_index, ]
#> (Intercept)          t1          t2          t3          t4
#>        TRUE        TRUE       FALSE        TRUE        TRUE
summary_sub$which[best_bic_index, ]
#> (Intercept)          t1          t2          t3          t4
#>        TRUE        TRUE       FALSE        TRUE        TRUE
# Lưu ý: force.in / force.out trong leaps thường là chỉ số cột nếu x,y là dạng ma trận
# Nhưng với công thức, một số phiên bản cho phép ta truyền tên trực tiếp (nếu không được, hãy dùng chỉ
fit_sub2 <- regsubsets(y ~ t1 + t2 + t3 + t4,
                       data = jobs_data,
                       method = "exhaustive",
                       nbest = 1,
                       nvmax = 3,
                       force.in = "t1",
                       force.out = "t2")

summary_sub2 <- summary(fit_sub2)
```

Problem 2

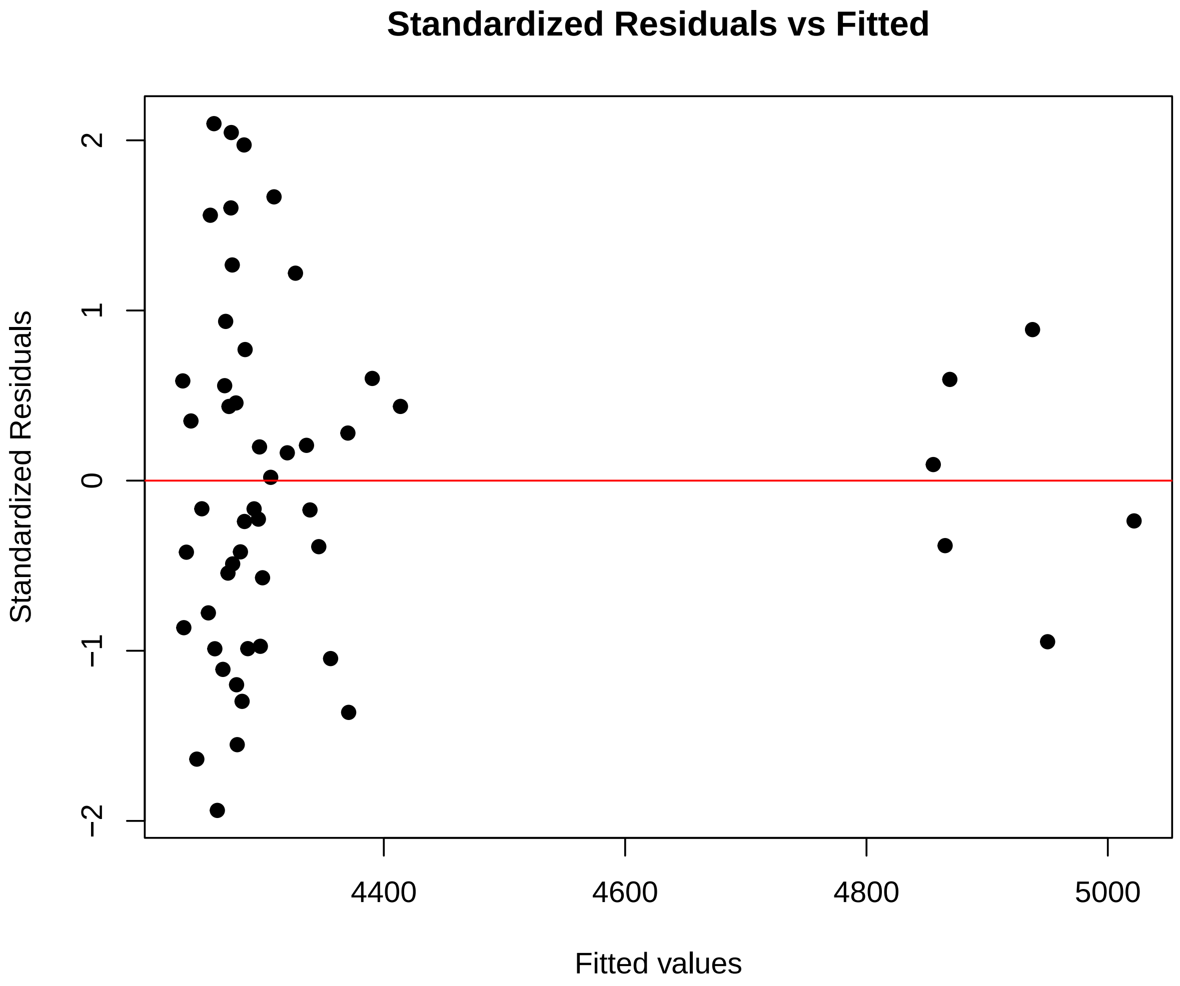
Part 1

```r
library(readxl)
# Đọc dữ liệu từ file "grocery.txt"
data <- read_excel("grocery.xlsx", col_names = TRUE)

# Xây dựng mô hình hồi quy tuyến tính với các biến dự báo: shipped, cost, và holiday
result <- lm(labor ~ shipped + cost + holiday, data = data)

# Thiết lập cửa sổ đồ họa chia thành 1 hàng, 3 cột để vẽ 3 biểu đồ cạnh nhau
par(mfrow = c(1, 3))
```

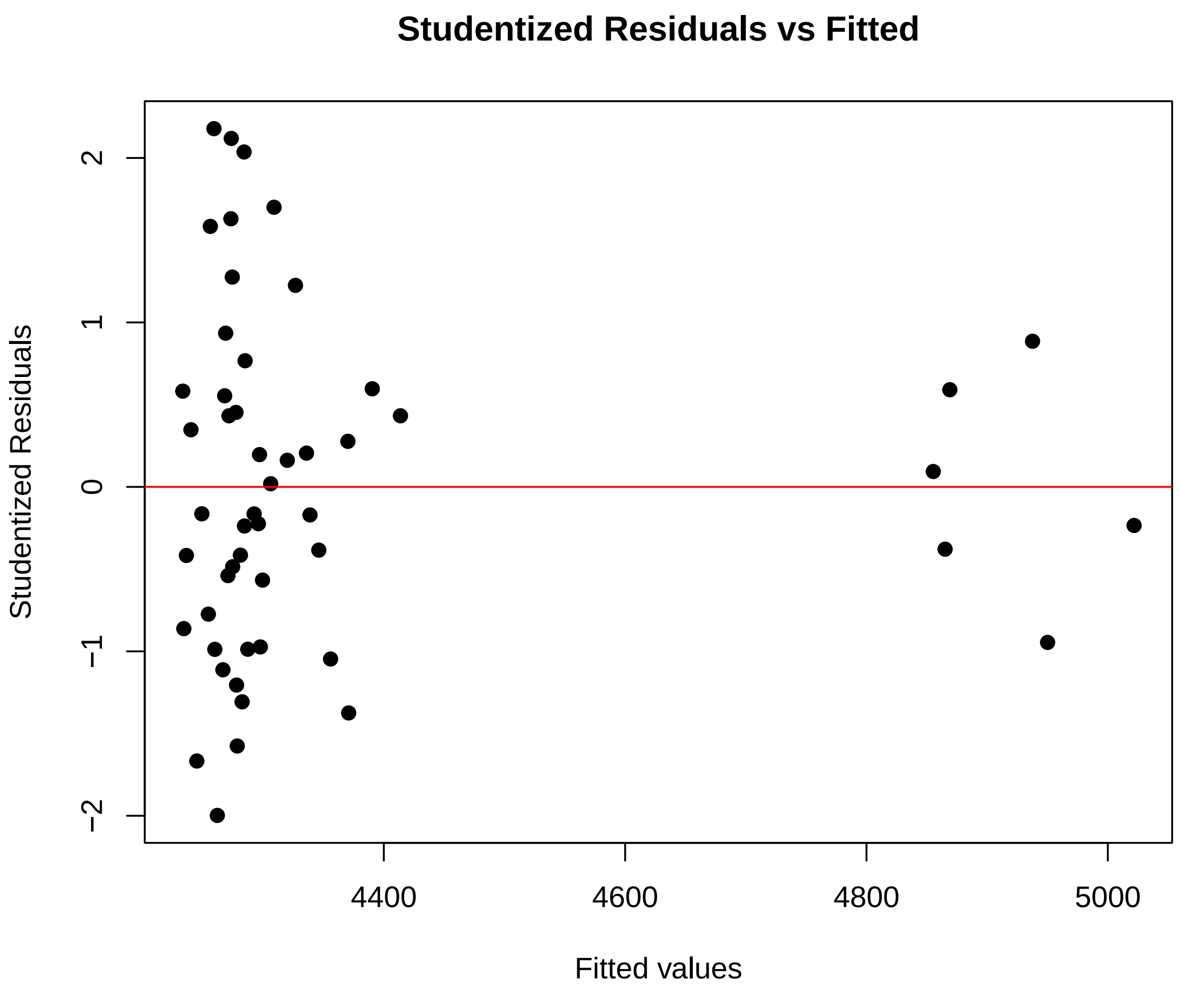
```r
options(repr.plot.width = 7, repr.plot.height = 6)
# Biểu đồ Residuals vs Fitted Values
plot(result$fitted.values, result$residuals,
     main = "Residuals vs Fitted",
     xlab = "Fitted values", ylab = "Residuals",
     pch = 19)
abline(h = 0, col = "red")
```



```r
options(repr.plot.width = 7, repr.plot.height = 6)
# Biểu đồ Standardized Residuals vs Fitted Values
plot(result$fitted.values, rstandard(result),
     main = "Standardized Residuals vs Fitted",
     xlab = "Fitted values", ylab = "Standardized Residuals",
     pch = 19)
abline(h = 0, col = "red")
```



```r
options(repr.plot.width = 7, repr.plot.height = 6)
# Biểu đồ Studentized Residuals vs Fitted Values
plot(result$fitted.values, rstudent(result),
     main = "Studentized Residuals vs Fitted",
     xlab = "Fitted values", ylab = "Studentized Residuals",
     pch = 19)
abline(h = 0, col = "red")
```



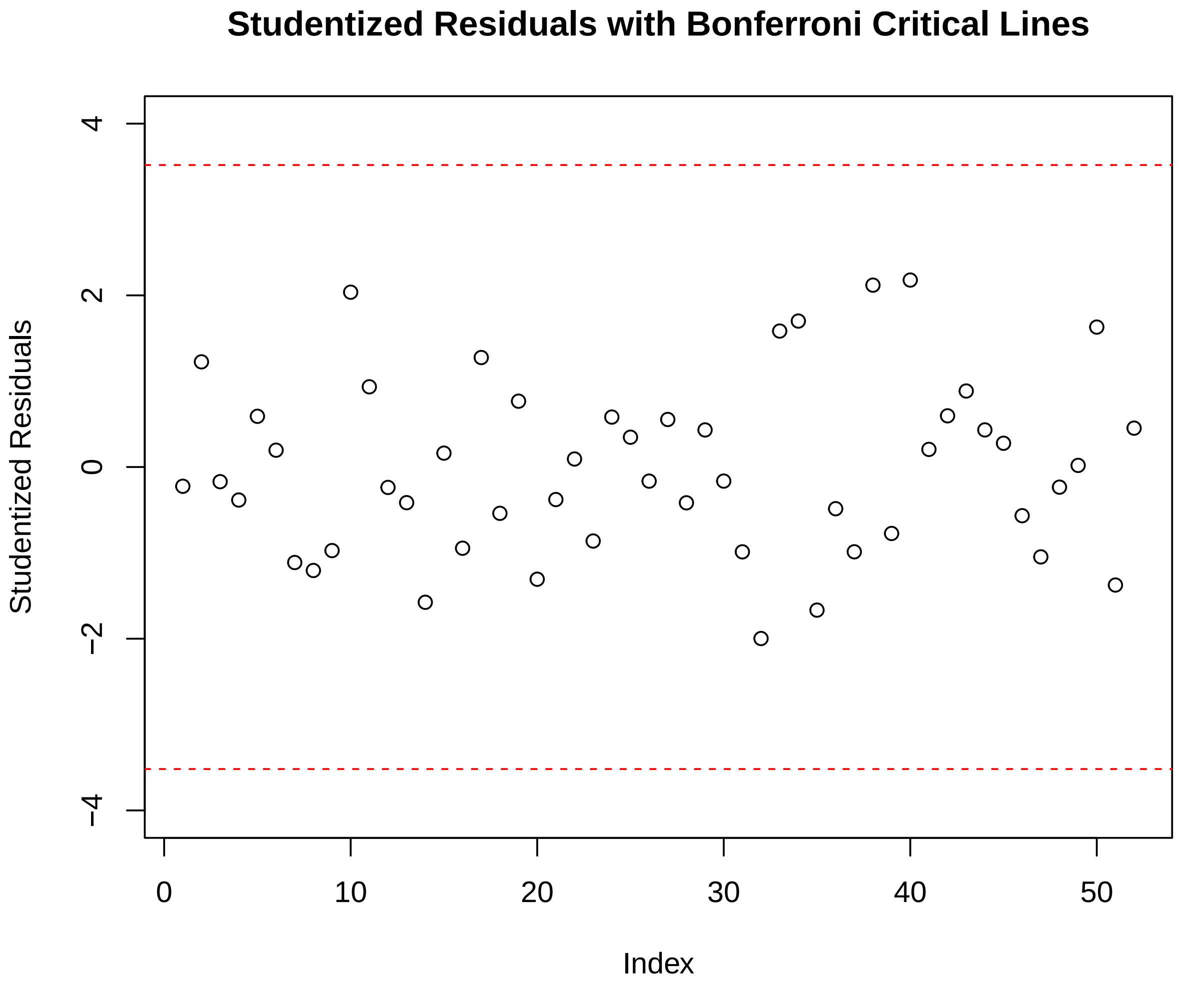
```
# Reset lại layout đồ họa về mặc định
par(mfrow = c(1,1))
```

Part 2

```
library(readxl)
data <- read_excel("grocery.xlsx", col_names = TRUE)

# Xây dựng mô hình hồi quy first order với các biến dự đoán: shipped, cost, và holiday
fit <- lm(labor ~ shipped + cost + holiday, data = data)

# Lấy số quan sát (n) và số biến dự báo (p)
n <- nrow(data)
p <- length(coef(fit)) - 1  # trừ đi hệ số intercept

# Tính phần dư studentized (rstudent)
student.res <- rstudent(fit)
```

```
# --- (a) Sắp xếp các phần dư studentized và in ra
sorted_res <- sort(student.res)
print("Sorted studentized residuals:")
#> [1] "Sorted studentized residuals:"
print(sorted_res)
#>          32          35          14          51          20           8
#> -1.99766654 -1.66686277 -1.57563574 -1.37470514 -1.30688333 -1.20529304
#>           7          47          31          37           9          16
#> -1.11220903 -1.04682238 -0.98793522 -0.98726030 -0.97317140 -0.94585538
#>          23          39          46          18          36          28
#> -0.86199416 -0.77401415 -0.56690329 -0.53946536 -0.48548061 -0.41694583
#>          13           4          21          12          48           1
#> -0.41516269 -0.38465346 -0.37866873 -0.23775605 -0.23443689 -0.22408724
#>           3          26          30          49          22          15
#> -0.17058921 -0.16427705 -0.16381817  0.01909697  0.09348669  0.16177701
#>           6          41          45          25          29          44
#>  0.19612403  0.20535372  0.27680521  0.34737819  0.43222641  0.43246959
#>          52          27          24           5          42          19
#>  0.45278959  0.55395260  0.58204380  0.59079243  0.59660183  0.76695374
#>          43          11           2          17          33          50
#>  0.88556630  0.93459516  1.22549009  1.27571169  1.58403006  1.63020460
#>          34          10          38          40
#>  1.70041654  2.03651764  2.11878596  2.17827186

# Tính giá trị tới hạn theo phương pháp Bonferroni
# Sử dụng  = 0.05. Giá trị tới hạn: qt(1 -  /(2*n), df = n - p - 1)
alpha <- 0.05
crit <- qt(1 - alpha/(2*n), df = n - p - 1)
print("Critical value (Bonferroni):")
#> [1] "Critical value (Bonferroni):"
print(crit)
#> [1] 3.518198
```

```
options(repr.plot.width = 7, repr.plot.height = 6)
# --- (b) Vẽ biểu đồ phần dư studentized và overlay các đường giới hạn tới hạn
plot(student.res,
     ylab = "Studentized Residuals",
     main = "Studentized Residuals with Bonferroni Critical Lines",
     ylim = c(-4, 4))
abline(h = crit, col = "red", lty = 2)
abline(h = -crit, col = "red", lty = 2)
```



```
# --- (c) Liệt kê các quan sát có |studentized residual| > giá trị tới hạn
outliers <- which(abs(student.res) > crit)
print("Indices of potential outliers (|t_i| > critical value):")
#> [1] "Indices of potential outliers (|t_i| > critical value):"
print(outliers)
#> named integer(0)
```