Regression And Stat Models - Assignment 6

Nathan Pasder

2025-06-05

Contents

Question 1

(a) Write the matrix X used in the model

We define the matrix $X \in \mathbb{R}^{n \times 8}$, where each row corresponds to one observation (a team-season), and each column corresponds to one of the predictors or interactions. The formula used in the R model is:

```
lm(fans ~ budget * league)
```

This expands to the standard form of a linear model with interaction terms:

$$\begin{aligned} \text{fans} = \beta_0 + \beta_1 \cdot \text{budget} + \beta_2 \cdot \text{leagueLaLiga} + \beta_3 \cdot \text{leaguePremier} + \beta_4 \cdot \text{leagueSerieA} \\ + \beta_5 \cdot (\text{budget} \cdot \text{leagueLaLiga}) + \beta_6 \cdot (\text{budget} \cdot \text{leaguePremier}) + \beta_7 \cdot (\text{budget} \cdot \text{leagueSerieA}) + \varepsilon \end{aligned} \quad (1)$$

This is based on the key idea that a categorical variable with k levels is represented using $k-1$ dummy variables, and an interaction term like `budget * league` introduces both the main effect and the interaction terms.

This corresponds to the following columns in X:

- Intercept (column of ones)
- budget
- leagueLaLiga (1 if LaLiga, 0 otherwise)
- leaguePremier (1 if Premier League, 0 otherwise)
- leagueSerieA (1 if Serie A, 0 otherwise)
- budget $\times$ leagueLaLiga
- budget $\times$ leaguePremier
- budget $\times$ leagueSerieA

Matrix form:

$$X = \begin{bmatrix} 1 & \text{budget}_1 & L1_1 & P1_1 & S1_1 & \text{budget}_1 \cdot L1_1 & \text{budget}_1 \cdot P1_1 & \text{budget}_1 \cdot S1_1 \\ 1 & \text{budget}_2 & L1_2 & P1_2 & S1_2 & \text{budget}_2 \cdot L1_2 & \text{budget}_2 \cdot P1_2 & \text{budget}_2 \cdot S1_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \text{budget}_n & L1_n & P1_n & S1_n & \text{budget}_n \cdot L1_n & \text{budget}_n \cdot P1_n & \text{budget}_n \cdot S1_n \end{bmatrix}$$

(b) Fill in the missing values in the output

The degrees of freedom in a linear model are given by:

$$\text{df} = n - p$$

Where n is the number of observations and p is the number of estimated parameters (including the intercept).

Given values:

- $n = 1000$ (stated in the question)
- $p = 8$ (1 intercept + 1 budget + 3 league dummies + 3 interaction terms)
- Degrees of freedom: $df = 1000 - 8 = \boxed{992}$
- F-statistic: $F = 1074$, on $\boxed{7}$ and $\boxed{992}$ degrees of freedom

To compute the p-value for leagueSerieA:

- Estimate: 2.409351
- Std. Error: 0.952114
- **Key formula:** $t = \frac{\text{Estimate}}{\text{Std. Error}}$

- $t = \frac{2.409351}{0.952114} \approx 2.531$

The p-value is calculated using the two-tailed probability from the t-distribution:

$$\text{p-value} = 2 \cdot P(T_{df} > |t|)$$

Using a t-distribution calculator with 992 degrees of freedom:

$$P(T_{992} > 2.531) \approx 0.0057 \Rightarrow \text{p-value} \approx 2 \cdot 0.0057 = \boxed{0.0114}$$

To compute R^2:

Adjusted R^2 corrects for the number of predictors and is calculated as:

$$R^2_{adj} = 1 - (1 - R^2) \cdot \frac{n-1}{n-p}$$

Solving for R^2:

$$\begin{aligned} 0.8826 &= 1 - (1 - R^2) \cdot \frac{999}{992} \\ 1 - 0.8826 &= (1 - R^2) \cdot 1.00706 \\ 0.1174 &= 1.00706(1 - R^2) \\ 1 - R^2 &= \frac{0.1174}{1.00706} = 0.1166 \\ R^2 &= 1 - 0.1166 = \boxed{0.8834} \end{aligned}$$

(c) Compute a confidence interval for expected fans of a team in the baseline league with 10M budget

For the reference group (baseline league — the one not explicitly listed in dummy variables), all dummy variables are 0. The prediction model simplifies to:

$$\hat{Y} = \beta_0 + \beta_1 \cdot \text{budget}$$

Substituting values:

$$\hat{Y} = 42.271730 + 0.220877 \cdot 10 = 44.4805$$

The formula for a 95% confidence interval for the expected value:

$$\hat{Y} \pm t_{0.975,\, df} \cdot SE(\hat{Y})$$

Since $SE(\hat{Y})$ is not given, we leave it in symbolic form. Note that $SE(\hat{Y})$ would be calculated as:

$$SE(\hat{Y}) = \sqrt{x_0^T (X^T X)^{-1} x_0 \cdot \hat{\sigma}^2}$$

where $x_0 = [1, 10, 0, 0, 0, 0, 0, 0]^T$ for the baseline league with budget = 10M. ## (d) Explain the source of the difference in marginal effect of budget

In interaction models, the slope of one variable (e.g. budget) depends on the category of another variable (e.g. league). The marginal effect is:

$$\text{slope}_{\text{league}} = \beta_1 + \text{interaction term}$$

From the output:

$$\begin{aligned}
\text{Baseline league (reference)} &: 0.2209 \\
\text{LaLiga} &: 0.2209 + 0.025686 &&= \boxed{0.2466} \\
\text{Premier League} &: 0.2209 - 0.198629 &&= \boxed{0.0223} \\
\text{Serie A} &: 0.2209 - 0.025224 &&= \boxed{0.1957}
\end{aligned}$$

Interpretation:

- The slope is highest in LaLiga, suggesting higher return on investment.
- It is lowest in Premier League, suggesting diminishing returns or fan saturation.

(e) Compare models with and without interaction terms

We compare two models:

```
fm1 <- lm(fans ~ budget * league, data = df)
fm2 <- lm(fans ~ budget + league, data = df)
```

The operator `*` in R expands to both main effects and interaction terms:

$$\texttt{budget * league} = \texttt{budget + league + budget:league}$$

This means `fm1` allows the effect (slope) of budget on fan attendance to vary across leagues, while `fm2` assumes the effect of budget is constant (same slope) across all leagues — only the intercepts change.

Interpretation of plot:

In the plot, we see two leagues — Premier (black circles) and LaLiga (red triangles) — showing different patterns:

- LaLiga shows a clear increasing relationship: higher budget ⇒ higher fans per match.
- Premier shows a flatter trend: increasing budget does not raise fan attendance much.

Conclusion:

- `fm1` captures this difference in slopes using interaction terms.
- `fm2` forces a single slope for both leagues, which is incorrect here.

Therefore: The interaction model (`fm1`) is more appropriate because the relationship between budget and attendance depends on the league. The difference in slope is essential and cannot be ignored.

Question 2

We investigate whether each response variable y_k follows the classical linear model assumptions using the file `ex6_q3_data.csv`.

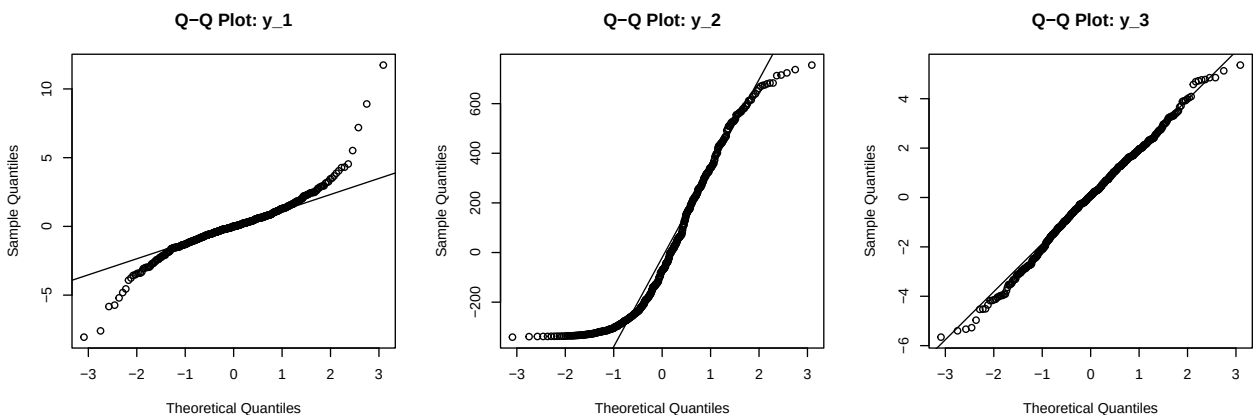
(a) Fit linear models

```r
df <- read.csv("ex6_q3_data.csv")

models <- lapply(1:3, function(k) {
  y <- df[[paste0("y_", k)]]
  lm(y ~ x, data = df)
})
names(models) <- paste0("y_", 1:3)
```

(b) Show Q-Q plots of residuals

```r
par(mfrow = c(1, 3))
for (k in 1:3) {
  qqnorm(residuals(models[[k]]), main = paste("Q-Q Plot: y_", k, sep = ""))
  qqline(residuals(models[[k]]))
}
```



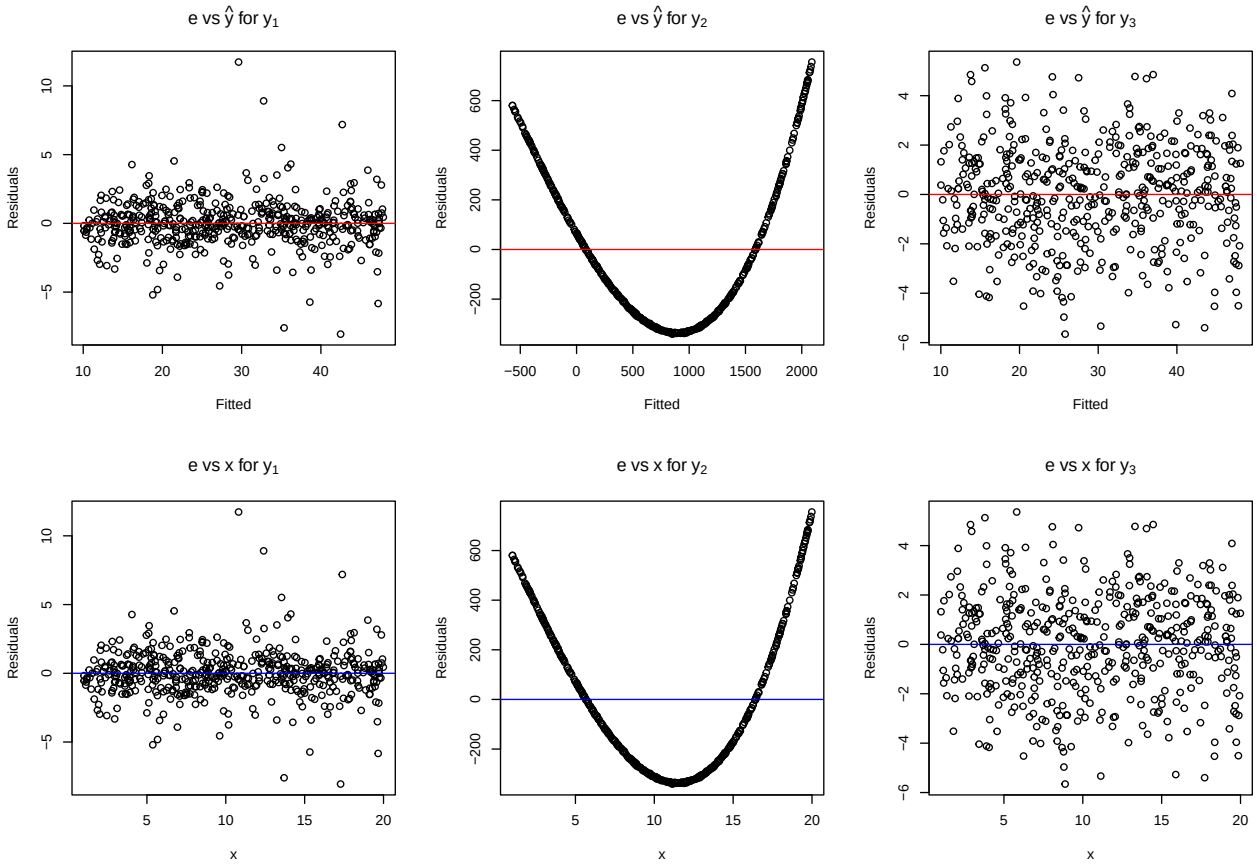
(c) Residual vs. fitted & residual vs. x plots

```r
par(mfrow = c(2, 3))

# Residuals vs. fitted
for (k in 1:3) {
  res <- residuals(models[[k]])
  fit <- fitted(models[[k]])
  plot(fit, res,
       main = bquote(e ~ "vs" ~ hat(y) ~ "for" ~ y[.(k)]),
       xlab = "Fitted", ylab = "Residuals")
  abline(h = 0, col = "red")
}

# Residuals vs. x
for (k in 1:3) {
  res <- residuals(models[[k]])
  plot(df$x, res,
       main = bquote(e ~ "vs" ~ x ~ "for" ~ y[.(k)]),
       xlab = "x", ylab = "Residuals")
  abline(h = 0, col = "blue")
```

```
}
```



(d) Explanation — Is there evidence of deviation?

Detailed Analysis:

- y_1: The Q-Q plot shows heavy tails indicating non-normal residuals. The residual vs. fitted plot shows a clear funnel pattern, suggesting **heteroscedasticity** (non-constant variance) - variance increases with fitted values.
- y_2: The Q-Q plot shows points closely following the theoretical line, indicating approximately normal residuals. Residual plots show random scatter around zero with fairly constant spread, suggesting the **classical assumptions are satisfied**.
- y_3: While the Q-Q plot appears reasonably normal, the residuals vs. fitted plot shows a clear **curved/parabolic pattern**, indicating a **non-linear relationship** between x and y_3 that is not captured by the linear model.

Conclusion: Models for y_1 and y_3 show clear deviations from classical linear model assumptions, while y_2 appears to satisfy the assumptions.

(e) Standardized residuals and outlier detection

We compute the standardized residuals using the formula:

$$r_i = \frac{e_i - \bar{e}}{SD_e}$$

```r
std_residuals <- lapply(1:3, function(k) {
  e <- residuals(models[[k]])
  (e - mean(e)) / sd(e)
})
names(std_residuals) <- paste0("y_", 1:3)

# Check for values with |r_i| > 2
outlier_indices <- lapply(std_residuals, function(r) which(abs(r) > 2))
print("Outlier positions (|r_i| > 2):")
```

```
## [1] "Outlier positions (|r_i| > 2):"
```

```r
outlier_indices
```

```
## $y_1
##  12  17  33 101 112 114 136 139 160 175 182 233 265 295 297 320 337 370 410 422
##  12  17  33 101 112 114 136 139 160 175 182 233 265 295 297 320 337 370 410 422
## 429 430 467
## 429 430 467
##
## $y_2
##  24  87 126 139 193 248 277 297 327 340 347 356 380 391 400 401 434 461 491 496
##  24  87 126 139 193 248 277 297 327 340 347 356 380 391 400 401 434 461 491 496
##
## $y_3
##  15  57  60  86 109 110 121 142 166 177 189 199 206 258 299 341 366 380 415 440
##  15  57  60  86 109 110 121 142 166 177 189 199 206 258 299 341 366 380 415 440
## 446 493 496 497 499
## 446 493 496 497 499
```

```r
# Summary of outlier analysis
cat("Summary of Outlier Detection:\n")
```

```
## Summary of Outlier Detection:
```

```r
for (k in 1:3) {
  outliers <- outlier_indices[[k]]
  cat("y_", k, ": ", length(outliers), " observations with |r_i| > 2")
  if (length(outliers) > 0) {
    cat(" (positions: ", paste(outliers, collapse=", "), ")")
  }
  cat("\n")
}
```

```
## y_ 1 :  23  observations with |r_i| > 2 (positions:  12, 17, 33, 101, 112, 114, 136, 139, 160, 175, 
## y_ 2 :  20  observations with |r_i| > 2 (positions:  24, 87, 126, 139, 193, 248, 277, 297, 327, 340, 
## y_ 3 :  25  observations with |r_i| > 2 (positions:  15, 57, 60, 86, 109, 110, 121, 142, 166, 177, 18
```

```r
# Check for extreme outliers (|r_i| > 3)
cat("\nExtreme outliers (|r_i| > 3):\n")
```

```
##
## Extreme outliers (|r_i| > 3):
```

```r
for (k in 1:3) {
  extreme <- which(abs(std_residuals[[k]]) > 3)
  cat("y_", k, ": ", length(extreme), " extreme outliers\n")
```

```
}
```

```
## y_ 1 :  9  extreme outliers
## y_ 2 :  0  extreme outliers
## y_ 3 :  0  extreme outliers
```