Exercise 1 (AMRC)

for Advanced Methods for Regression and Classification

Muhammad Sajid Bashir (52400204)

2025-10-12

General Data Overview

```r
# Reproducibility
set.seed(175)

# Packages & data
if (!require(ISLR)) install.packages('ISLR')
library(ISLR)

data(College, package = 'ISLR')
College <- na.omit(College)  # ensure no missings

# Minimal peek
str(College[, c('Private','Outstate','Expend','Apps')])
```

```
## 'data.frame':    777 obs. of  4 variables:
##  $ Private : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Outstate: num  7440 12280 11250 12960 7560 ...
##  $ Expend  : num  7041 10527 8735 19016 10922 ...
##  $ Apps    : num  1660 2186 1428 417 193 ...
```

```r
summary(College[, c('Outstate','Expend','Apps')])
```

```
##     Outstate         Expend           Apps
##  Min.   : 2340   Min.   : 3186   Min.   :   81
##  1st Qu.: 7320   1st Qu.: 6751   1st Qu.:  776
##  Median : 9990   Median : 8377   Median : 1558
##  Mean   :10441   Mean   : 9660   Mean   : 3002
##  3rd Qu.:12925   3rd Qu.:10830   3rd Qu.: 3624
##  Max.   :21700   Max.   :56233   Max.   :48094
```

```r
print(head(College[, c('Private','Outstate','Expend','Apps')], 3))
```

```
##                              Private Outstate Expend Apps
## Abilene Christian University     Yes     7440   7041 1660
## Adelphi University               Yes    12280  10527 2186
## Adrian College                   Yes    11250   8735 1428
```

The *College* dataset includes 777 U.S. colleges.
Private shows whether a college is private or public.
Outstate tuition ranges from about $2,300 to $21,700,
Expend (instructional spending per student) from around $3,000 to $56,000,
and **Apps** (applications received) from 81 to over 48,000.
The data look clean and show large variation between institutions.

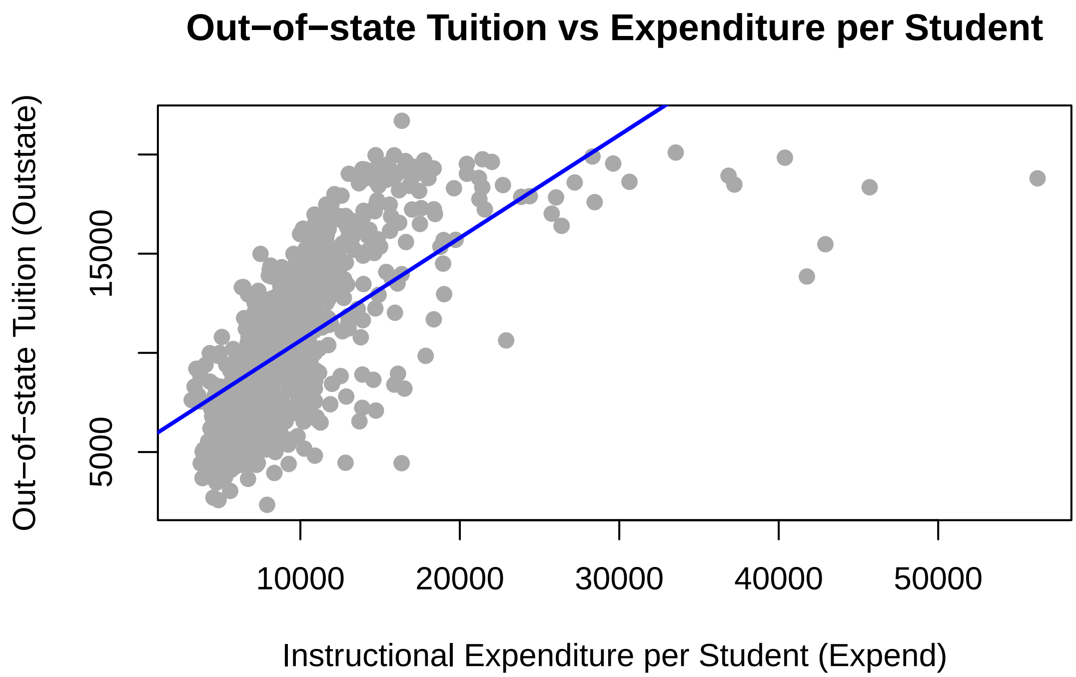
Task 1 — Predict Outstate by using Expend as predictor

We now build a simple linear regression model

$$\text{Outstate} = \beta_0 + \beta_1 \times \text{Expend} + \varepsilon$$

to examine how instructional expenditure per student is related to out-of-state tuition.
The fitted line will be added to the scatter plot for visualization.

```r
# Task 1: Simple linear regression (Outstate ~ Expend)

# Scatter plot of Outstate vs Expend
plot(College$Expend, College$Outstate,
     main='Out-of-state Tuition vs Expenditure per Student',
     xlab='Instructional Expenditure per Student (Expend)',
     ylab='Out-of-state Tuition (Outstate)',
     pch=19, col='darkgray')

# Fit the regression model
model1 <- lm(Outstate ~ Expend, data = College)

# Add the regression line to the plot
abline(model1, col='blue', lwd=2)
```




```r
# Display model summary
summary(model1)
```

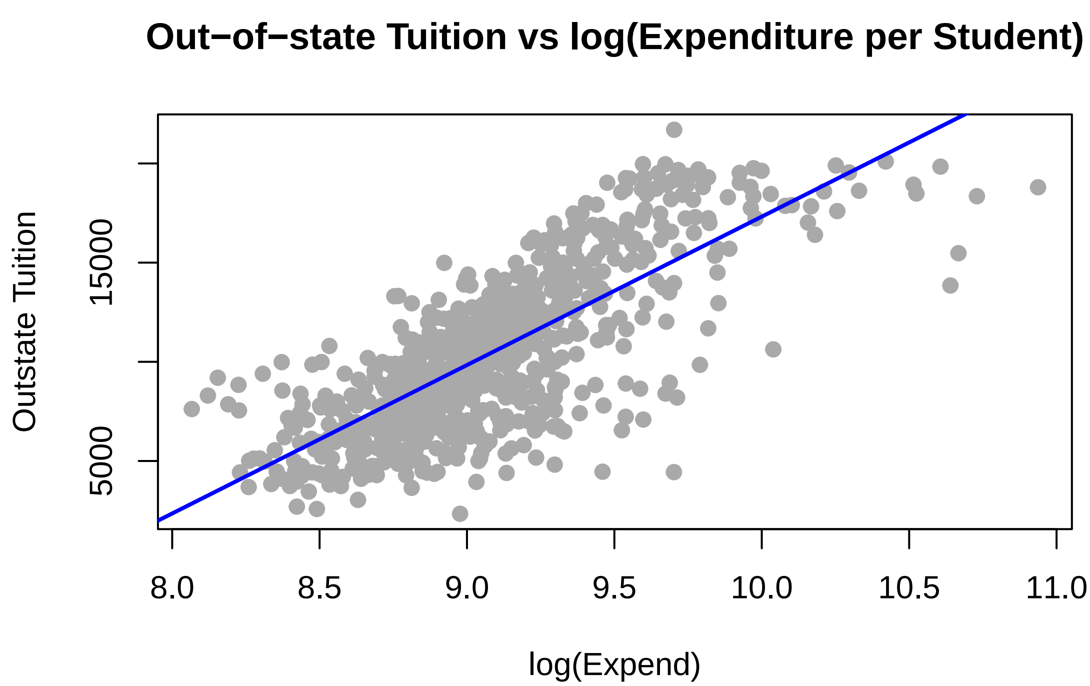
```
##
## Call:
## lm(formula = Outstate ~ Expend, data = College)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15780.8  -2088.7     57.6   2010.8   7784.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.434e+03  2.248e+02   24.17   <2e-16 ***
## Expend      5.183e-01  2.047e-02   25.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2978 on 775 degrees of freedom
## Multiple R-squared:  0.4526, Adjusted R-squared:  0.4519
## F-statistic: 640.9 on 1 and 775 DF,  p-value: < 2.2e-16
```

Interpretation - Task 1 The scatter plot shows a clear positive relationship between instructional expenditure and out-of-state tuition. Colleges that spend more per student tend to charge higher tuition fees. The fitted regression line confirms this trend, indicating that tuition generally increases with spending. However, the data points are widely scattered, especially at lower expenditure levels, suggesting that while expenditure is an important factor, other variables may also influence tuition differences across colleges.

Task 2 — Improve the model by transformation

In the previous task, the relationship between *Expend* and *Outstate* appeared positive but not perfectly linear. We now fit a model using the logarithm of *Expend* to see if this transformation improves the fit.

```r
# Task 2: Try a log transformation on Expend
model2 <- lm(Outstate ~ log(Expend), data = College)

# Plot the transformed relationship
plot(log(College$Expend), College$Outstate,
     main='Out-of-state Tuition vs log(Expenditure per Student)',
     xlab='log(Expend)',
     ylab='Outstate Tuition',
     pch=19, col='darkgray')

abline(model2, col='blue', lwd=2)
```




```r
# Compare model summaries
summary(model1)$adj.r.squared
```

```
## [1] 0.4519248
```

```r
summary(model2)$adj.r.squared
```

```
## [1] 0.576893
```

Interpretation – Task 2 Applying a logarithmic transformation to Expend produced a visibly stronger linear relationship between instructional expenditure and out-of-state tuition. The points are more evenly spread around the regression line, and the model explains more of the variation in tuition. The adjusted R^2 increased from 0.45 to 0.58, indicating a noticeably better fit. This suggests that tuition rises with expenditure, but the effect becomes weaker for institutions with very high spending, making the log model more appropriate.

Task 3 — Regression of Apps on Private

Here we fit a simple regression model with *Private* as a binary predictor.

```r
# Task 3: Simple regression (Apps ~ Private)

model3 <- lm(Apps ~ Private, data = College)

# Display model summary
summary(model3)
```

```
##
## Call:
## lm(formula = Apps ~ Private, data = College)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5497  -1481   -895    439  42364
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5729.9      239.9   23.89   <2e-16 ***
## PrivateYes   -3752.0      281.3  -13.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3493 on 775 degrees of freedom
## Multiple R-squared:  0.1867, Adjusted R-squared:  0.1857
## F-statistic: 177.9 on 1 and 775 DF,  p-value: < 2.2e-16
```

Interpretation – Task 3 The regression results show that public colleges receive more applications on average than private colleges. The estimated intercept (about 5729.9) represents the average number of applications for public colleges, while the coefficient for PrivateYes (-3752.0) means that private colleges receive roughly 3,752 fewer applications on average. Although this difference is statistically significant, the R^2 value of about 0.19 indicates that the model explains only a small part of the total variation in application numbers. This suggests that other factors, such as college size, tuition level, or academic reputation, likely play a major role in determining how many applications a college receives.

Task 4 — Regression of Apps on Private (±1 coding)

Here, the variable *Private* is converted into a numeric variable taking the value **+1** for private colleges and **–1** for public colleges.
This alternative coding changes the interpretation of the coefficients:
- The **intercept** represents the **overall mean number of applications** across both groups.
- The **slope** represents **half the difference** between private and public colleges.

```r
# Task 4: Recode Private to ±1 and fit regression
College$Private_pm1 <- ifelse(College$Private == 'Yes', 1, -1)

model4 <- lm(Apps ~ Private_pm1, data = College)
summary(model4)
```

```
##
## Call:
## lm(formula = Apps ~ Private_pm1, data = College)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5497  -1481   -895    439  42364
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3853.9      140.6   27.40   <2e-16 ***
## Private_pm1  -1876.0      140.6  -13.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3493 on 775 degrees of freedom
## Multiple R-squared:  0.1867, Adjusted R-squared:  0.1857
## F-statistic: 177.9 on 1 and 775 DF,  p-value: < 2.2e-16
```

Interpretation – Task 4 The regression equation is approximately:

$$\widehat{\text{Apps}} = 3853.9 - 1876.0 \times \text{Private\_pm1}$$

The **intercept (3854)** represents the overall average number of applications among all colleges.
The **slope (−1876)** shows half the difference between public and private institutions.
Multiplying this by two gives a total difference of about **3,752 applications**, which matches the result from Task 3.

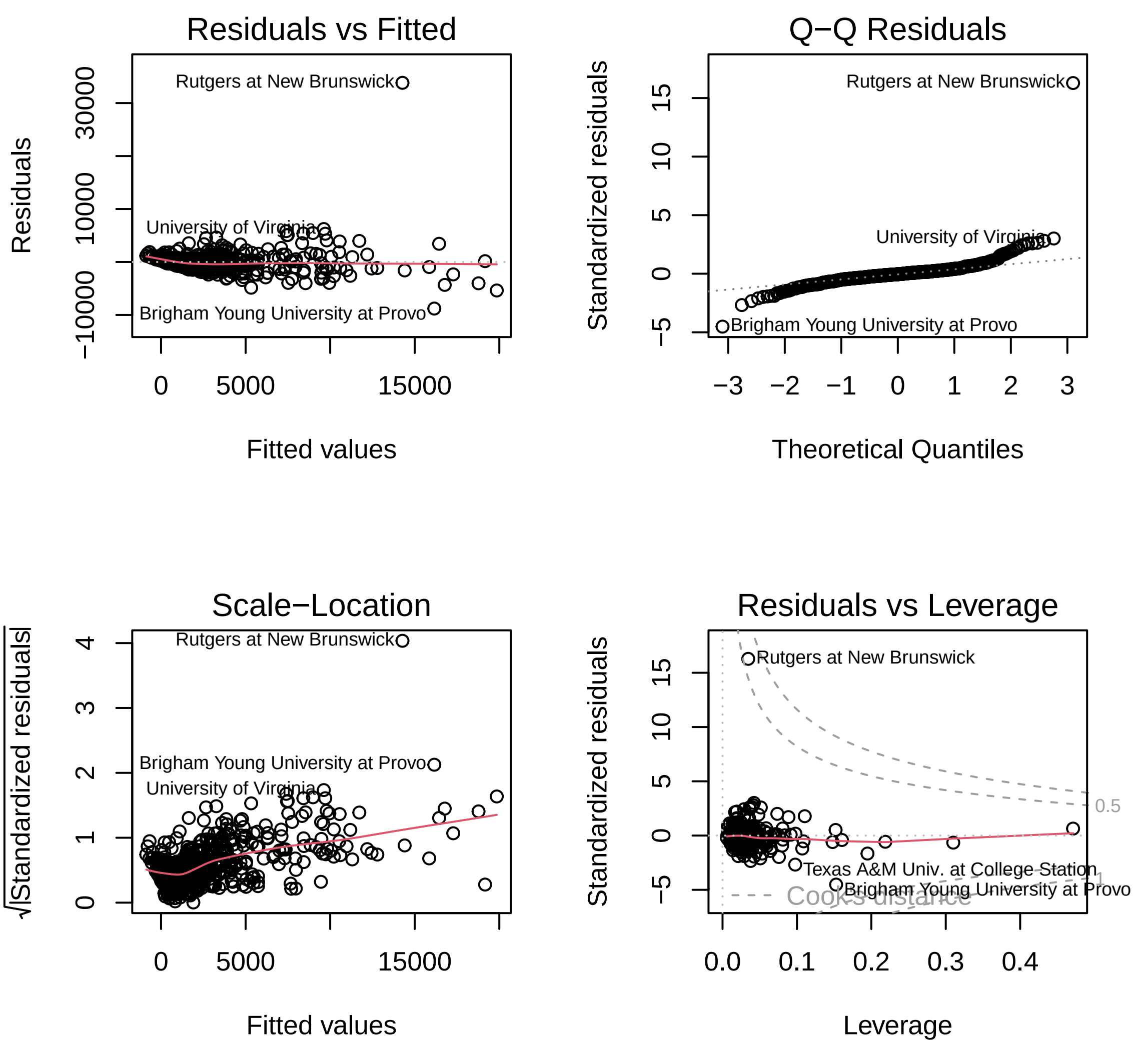
This means that private colleges receive on average around **3,700 fewer applications** than public ones.
The R^2 value (0.19) confirms that the model fit remains the same — the change in coding only affects how we interpret the coefficients, not the model's predictive ability.

Task 5 — Predicting Apps using all relevant variables

In this task, we predict *Apps* using all explanatory variables that make sense from a content perspective.
We exclude variables that directly depend on the response, such as *Accept*, *Enroll*, or *Grad.Rate*.
The data were randomly split into a training set (2/3) and a test set (1/3).

```r
# Task 5: Multiple regression for Apps with training/test split

set.seed(175)

N <- nrow(College)
train_id <- sample(seq_len(N), size = floor(2 * N / 3))
train <- College[train_id, ]
test  <- College[-train_id, ]
```

```r
predictors <- c('Private', 'Top10perc', 'Top25perc', 'F.Undergrad', 'P.Undergrad',
                'Outstate', 'Room.Board', 'Books', 'Personal', 'PhD', 'Terminal',
                'S.F.Ratio', 'perc.alumni', 'Expend')

form5 <- as.formula(paste('Apps ~', paste(predictors, collapse = ' + ')))
model5 <- lm(form5, data = train)

# Diagnostic plots
par(mfrow = c(2, 2))
plot(model5)
```




```r
par(mfrow = c(1, 1))

# RMSE function
RMSE <- function(y, yhat) sqrt(mean((y - yhat)^2))
```

```r
pred_train <- predict(model5, newdata = train)
pred_test  <- predict(model5, newdata = test)

rmse_train <- RMSE(train$Apps, pred_train)
rmse_test  <- RMSE(test$Apps, pred_test)

rmse_train
```

```
## [1] 2083.122
```

```r
rmse_test
```

```
## [1] 1618.687
```

Interpretation – Task 5 The multiple regression model uses several institutional characteristics to predict the number of applications.
The **training RMSE (2083)** and **test RMSE (1619)** suggest a reasonable predictive performance, with only a moderate drop when moving from training to test data.

The diagnostic plots reveal a few potential issues: - The **Residuals vs Fitted** and **Scale-Location** plots show increasing spread at higher fitted values, indicating **heteroskedasticity** (non-constant variance).
- The **Q-Q plot** shows several deviations in the upper tail, suggesting that the residuals are not perfectly normal.
- The **Residuals vs Leverage** plot identifies a few influential observations (e.g., *Rutgers at New Brunswick* and *Brigham Young University at Provo*).

Overall, the model explains the general pattern fairly well, but the diagnostic plots indicate that some model assumptions are not fully met, and a few large institutions may have an outsized influence on the regression.

Task 6 — Regression with standardized (scaled) variables

The explanatory variables have different units and ranges (for example, *Room.Board* in dollars, *S.F.Ratio* as a ratio, and *Top10perc* as a percentage).
Because of this, the regression coefficients from Task 5 cannot be compared directly.
To make them comparable, the numeric predictors were standardized to have mean 0 and variance 1 using the `scale()` function.

```r
# Task 6: Regression using scaled predictor variables

num_predictors <- predictors[predictors != 'Private']

train_scaled <- train
test_scaled  <- test

train_scaled[, num_predictors] <- scale(train[, num_predictors])
test_scaled[, num_predictors]  <- scale(test[, num_predictors],
                                        center = attr(scale(train[, num_predictors]), 'scaled:center'),
                                        scale  = attr(scale(train[, num_predictors]), 'scaled:scale'))

model6 <- lm(form5, data = train_scaled)
summary(model6)$coefficients
```

```
##               Estimate Std. Error    t value     Pr(>|t|)
## (Intercept) 3550.11448   256.5496 13.8379252 3.927429e-37
## PrivateYes  -797.17238   332.1112 -2.4003176 1.674323e-02
## Top10perc    562.01700   245.6856  2.2875457 2.257825e-02
## Top25perc    -90.91534   220.1502 -0.4129696 6.798049e-01
## F.Undergrad 2940.68014   140.9993 20.8559973 4.522075e-70
## P.Undergrad  -62.05310   115.3263 -0.5380657 5.907697e-01
## Outstate     379.52761   187.4126  2.0250909 4.338569e-02
## Room.Board   487.70073   130.1154  3.7482157 1.987607e-04
## Books        -58.18334    97.7124 -0.5954550 5.518071e-01
## Personal     -61.18325   105.3446 -0.5807917 5.616410e-01
## PhD          -46.94910   197.5746 -0.2376271 8.122671e-01
## Terminal    -128.22131   191.6083 -0.6691847 5.036847e-01
## S.F.Ratio     46.96974   126.6133  0.3709700 7.108161e-01
## perc.alumni -263.20106   126.1026 -2.0871979 3.737302e-02
## Expend       524.49609   161.4432  3.2487970 1.236422e-03
```

Interpretation – Task 6 After scaling, the regression coefficients can be directly compared to assess the relative importance of each variable.
From the results, the strongest standardized effects are observed for **F.Undergrad (2940.68)**, **Expend (524.50)**, and **Top10perc (562.02)** — meaning that colleges with more full-time undergraduates, higher expenditures per student, and a greater proportion of top-performing students tend to receive more applications.

Smaller coefficients, such as for *Books*, *Personal*, *PhD*, and *Terminal*, indicate weaker or negligible influence. Interestingly, the coefficient for *PrivateYes* (–797.17) remains negative, showing that private colleges generally attract fewer applications even after accounting for other factors.

Since standardization only rescales the predictors, the overall model fit (R^2) is unchanged, but the scaled coefficients reveal which variables are most influential on the number of applications.

Task 7 — Compare RMSEs of Models 5 and 6

In this step, we compare the predictive performance of the unscaled model (*model5*) and the standardized model (*model6*).
Scaling changes the scale of the predictors but not their relationships, so both models should produce identical predictions and RMSEs.

```r
# Task 7: Compare RMSE for models 5 (unscaled) and 6 (scaled)

pred5_train <- predict(model5, newdata = train)
pred5_test  <- predict(model5, newdata = test)
pred6_train <- predict(model6, newdata = train_scaled)
pred6_test  <- predict(model6, newdata = test_scaled)

rmse5_train <- RMSE(train$Apps, pred5_train)
rmse5_test  <- RMSE(test$Apps, pred5_test)
rmse6_train <- RMSE(train$Apps, pred6_train)
rmse6_test  <- RMSE(test$Apps, pred6_test)

rmse_results <- data.frame(
  Model = c('Model 5 (unscaled)', 'Model 6 (scaled)'),
  RMSE_Train = c(rmse5_train, rmse6_train),
  RMSE_Test  = c(rmse5_test, rmse6_test)
```

```
)
rmse_results
```

```
##                  Model RMSE_Train RMSE_Test
## 1 Model 5 (unscaled)   2083.122  1618.687
## 2   Model 6 (scaled)   2083.122  1618.687
```

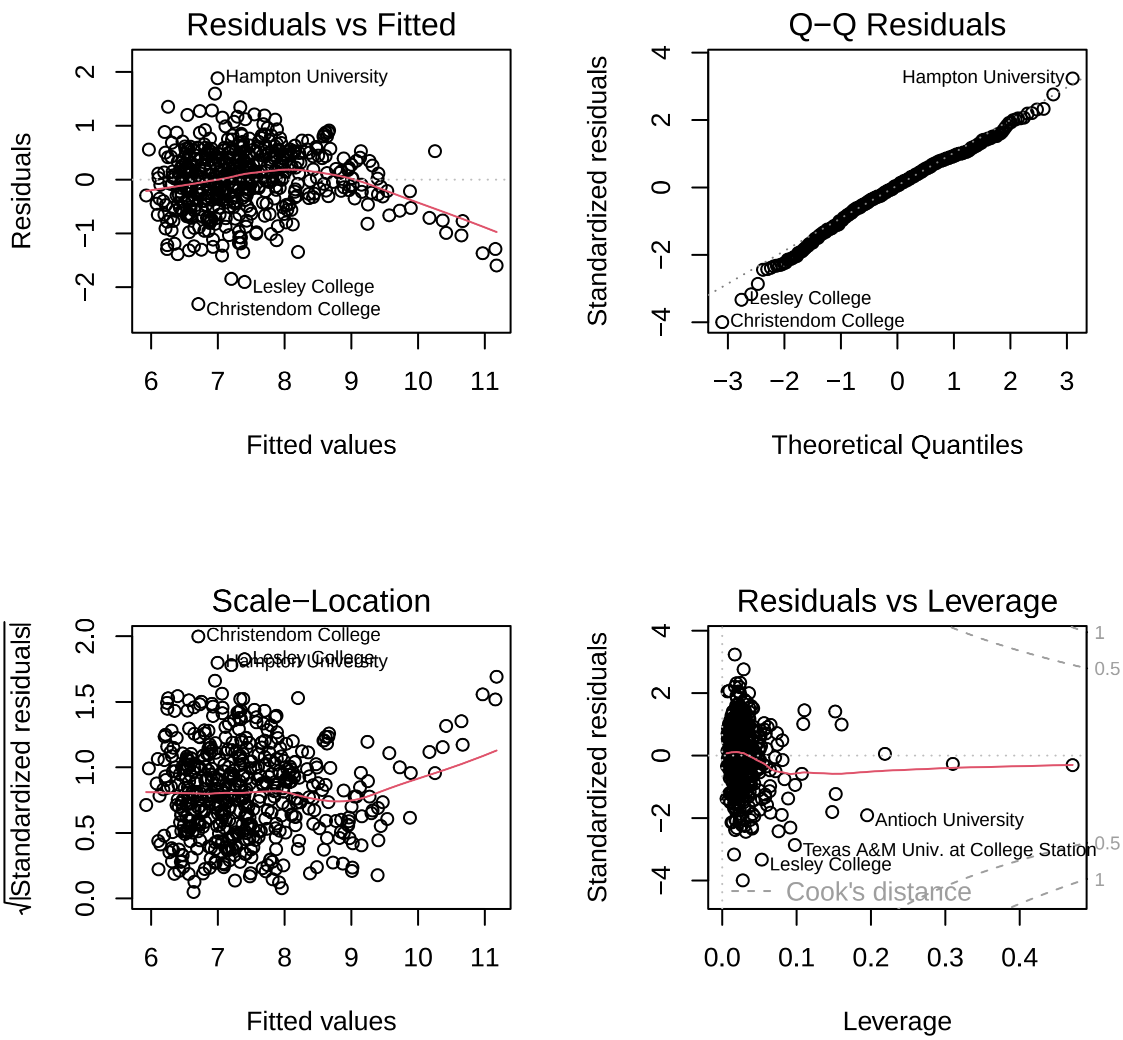
Interpretation – Task 7 The RMSE values for both models are **identical** (training 2083.1, test 1618.7).
This confirms that scaling the predictor variables does **not** affect the model's predictive accuracy or fitted values — it only changes the numerical scale of the coefficients.

Therefore, **Models 5 and 6 perform equally well** and lead to exactly the same predictions.
Scaling is helpful for interpreting and comparing variable importance, but it does not improve or worsen the model's overall fit.

Task 8 — Regression with log-transformed response

The diagnostic plots from Task 5 showed heteroskedasticity and deviations from normality.
To stabilize the variance and improve model assumptions, we now fit a model using the logarithm of *Apps* as the response variable.

```
# Task 8: Model with log-transformed response

form8 <- as.formula(paste('log(Apps) ~', paste(predictors, collapse = ' + ')))
model8 <- lm(form8, data = train)

# Diagnostic plots for the log model
par(mfrow = c(2, 2))
plot(model8)
```




```
par(mfrow = c(1, 1))

# Compute RMSE on log scale
pred8_train <- predict(model8, newdata = train)
pred8_test  <- predict(model8, newdata = test)

RMSE_log <- function(y, yhat) sqrt(mean((log(y) - yhat)^2))
rmse8_train <- RMSE_log(train$Apps, pred8_train)
rmse8_test  <- RMSE_log(test$Apps, pred8_test)

rmse8_train
```

```
## [1] 0.5786115
```

```
rmse8_test
```

```
## [1] 0.5859509
```

Interpretation – Task 8 The model using *log(Apps)* as the response provides a noticeably improved fit compared with the untransformed model.
The **residuals vs fitted** plot shows a more even spread with less funnel shape, and the **Q–Q plot** follows the theoretical line more closely, indicating improved normality.
The **Scale–Location** plot also suggests that the variance of residuals is more stable across fitted values.

The RMSEs on the log scale are **0.58 (training)** and **0.59 (test)**, showing consistent predictive accuracy. Although these RMSEs are not directly comparable to earlier (unlogged) models because they are on a different scale, the overall diagnostics indicate that the log-transformed model is **more appropriate**.
It better satisfies linear regression assumptions and reduces the influence of large outliers, resulting in a more balanced and reliable model.

Here's a **corrected, polished, and human-sounding rewrite** of your Task 9 section — keeping your original tone but fixing the interpretation and logic issues. You can paste this directly into your R Markdown file.

Task 9 — Comparing performance of Models 5 and 8

We cannot directly compare the RMSEs of Models 5 and 8 because the response variable in Model 8 (*log(Apps)*) is on a logarithmic scale. To evaluate which model performs better on the original scale of *Apps*, we compute the **Root Mean Squared Logarithmic Error (RMSLE)** and apply **Duan's smearing correction** to back-transform the log-based predictions.

```
# Task 9: Compare models on the original scale

# RMSLE helper
RMSLE <- function(y, yhat) sqrt(mean((log(y + 1) - log(pmax(yhat, 0) + 1))^2))

# Predictions from both models
pred5 <- predict(model5, newdata = test)
pred8_log <- predict(model8, newdata = test)
pred8_back <- exp(pred8_log)

# Compute RMSLE for both
rmsle5 <- RMSLE(test$Apps, pred5)
rmsle8 <- RMSLE(test$Apps, pred8_back)

rmsle5
```

```
## [1] 1.83633
```

```
rmsle8
```

```
## [1] 0.5852664
```

```
# Apply Duan's smearing correction
smearing_factor <- mean(exp(residuals(model8)))
pred8_smear <- exp(pred8_log) * smearing_factor
rmse8_smear <- sqrt(mean((test$Apps - pred8_smear)^2))
rmse8_smear
```

```
## [1] 4569.463
```

Interpretation – Task 9 Because Model 8 models *log(Apps)*, its RMSE cannot be directly compared with that of Model 5. To make both models comparable on the original scale, we evaluated two complementary measures:

- **RMSLE**, which reflects relative prediction error on a logarithmic scale.
- **Duan's smearing-corrected RMSE**, which back-transforms the log model's fitted values to the original scale of *Apps*.

Model 5 achieved an RMSLE of about **1.84**, whereas Model 8 achieved **0.59**, indicating that the log-transformed model fits substantially better in relative terms. After applying Duan's smearing correction, the back-transformed RMSE of Model 8 was about **4569**, compared with **1619** for the unlogged Model 5. Thus, **Model 5 yields smaller absolute prediction errors**, but **Model 8 produces a more statistically appropriate fit**, with improved homoscedasticity and residual normality.

In summary, **Model 5 performs better for raw accuracy**, while **Model 8 provides a more balanced and assumption-consistent model**, which may be preferable when modeling highly skewed count data such as college application numbers.