Table of Contents

List of Figures

List of Tables

1. Multiple Linear Regression Model to Predict `Outstate`

1.1. Fitting the Multiple Linear Regression Model

The dataset used in this assignment is the "College" dataset from the "ISLR2" library in R, which contains various statistics for numerous colleges in the United States. The dataset is organised as a data frame with 777 observations (rows) and 18 variables (columns). Notably, the dataset has no null values in any of its variables. The variables and their description are detailed in Table 1.1 below.

Table 1.1: Description of the variables from the "College" dataset

No.	Variable	Description	Type
1.	`Private`	A factor with levels `No` and `Yes` indicating whether the college is private or public.	Qualitative
2.	`Apps`	Number of applications received.	Quantitative
3.	`Accept`	Number of applications accepted.	Quantitative
4.	`Enroll`	Number of new students enrolled.	Quantitative
5.	`Top10perc`	Percentage of new students from the top 10% of their high school class.	Quantitative
6.	`Top25perc`	Percentage of new students from the top 25% of their high school class.	Quantitative
7.	`F.Undergrad`	Number of full-time undergraduates.	Quantitative
8.	`P.Undergrad`	Number of part-time undergraduates.	Quantitative
9.	`Outstate`	The tuition fees charged to students who are not residents of the state where the college is located.	Quantitative
10.	`Room.Board`	Room and board costs.	Quantitative
11.	`Books`	Estimated book costs.	Quantitative
12.	`Personal`	Estimated personal spending.	Quantitative
13.	`PhD`	Percentage of faculty with Ph.D.'s.	Quantitative
14.	`Terminal`	Percentage of faculty with terminal degrees.	Quantitative
15.	`S.F.Ratio`	Student/faculty ratio.	Quantitative
16.	`perc.alumni`	Percentage of alumni who donate.	Quantitative
17.	`Expend`	Instructional expenditure per student.	Quantitative
18.	`Grad.Rate`	Graduation rate.	Quantitative

For the first part of this assignment, only six (6) variables will be used: `Apps`, `Accept`, `Enroll`, `Top10perc`, and `Top25perc` as predictors, and `Outstate` as the response variable. All these variables are quantitative. In a multiple linear regression model with five (5) predictors and one (1) response variable, the formula can be written as:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4 + \hat{\beta}_5 x_5$$

where:

- $\hat{y}$ is the response variable `Outstate`,
- $\hat{\beta}_0$ is the intercept,
- $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4, \hat{\beta}_5$ are the coefficients for each predictor,
- x_1, x_2, x_3, x_4, x_5 are the predictors `Apps`, `Accept`, `Enroll`, `Top10perc`, and `Top25perc`, respectively.

Figure 1.1 displays the R code used to fit a multiple linear regression model for predicting `Outstate` using the predictors `Apps`, `Accept`, `Enroll`, `Top10perc`, and `Top25perc`.

```r
# Load the necessary library
library(ISLR2)

# Load the dataset
data("College")

# Check for missing values in the `Outstate` column
sum(is.na(College$Outstate))

# Fit the original multiple linear regression model
model = lm(Outstate ~ Apps + Accept + Enroll + Top10perc + Top25perc, data = College)
```

Figure 1.1: R code for fitting a multiple linear regression model using the "College" dataset from the "ISLR2" library

1.2. Interpretation of Model Results

Figure 1.2 presents the summary output of the fitted multiple linear regression model, highlighting the statistical significance and impact of each predictor on `Outstate` in the "College" dataset.

```
> summary(model)

Call:
lm(formula = Outstate ~ Apps + Accept + Enroll + Top10perc +
    Top25perc, data = College)

Residuals:
     Min       1Q   Median       3Q      Max
-11662.3  -1903.5     56.5   1817.4  10950.7

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 7468.66848  393.30646  18.989  < 2e-16 ***
Apps          -0.38616    0.09713  -3.976 7.68e-05 ***
Accept         1.58318    0.18680   8.475  < 2e-16 ***
Enroll        -3.61420    0.28242 -12.797  < 2e-16 ***
Top10perc    161.76486   14.48621  11.167  < 2e-16 ***
Top25perc    -12.61806   12.28241  -1.027    0.305
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2980 on 771 degrees of freedom
Multiple R-squared:  0.4548,    Adjusted R-squared:  0.4512
F-statistic: 128.6 on 5 and 771 DF,  p-value: < 2.2e-16
```

Figure 1.2: Summary of the multiple linear regression model

As depicted in Figure 1.2, the fitted multiple linear regression model can be written in the following formula:

$$\hat{y} = 7468.66848 - 0.38616x_1 + 1.58318x_2 - 3.61420x_3 + 161.76486x_4 - 12.61806x_5$$

where:

- $\hat{y}$ is the response variable `Outstate`,
- x_1, x_2, x_3, x_4, x_5 are the predictors `Apps`, `Accept`, `Enroll`, `Top10perc`, and `Top25perc`, respectively.

1.2.1. Significance of the Model

According to Figure 1.2 (line 24), the overall F-statistic for the model is 128.6 with a p-value of less than 2.2×10^{-16}. The hypothesis being tested is:

$$H_0: \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$$
$$H_1: At\ least\ one\ \beta_i \neq 0$$

At a 5% significance level ($\alpha = 0.05$), the p-value of the model is smaller than α. Therefore, the null hypothesis is rejected at the 5% level. This indicates that the model is statistically significant, meaning that at least one of the predictors is significantly related to the response variable `Outstate`.

In addition, the Residual Standard Error (RSE) is 2980. The Multiple R-squared value is 0.4548, suggesting that approximately 45.48% of the variability in the response variable `Outstate` can be explained by the model.

Next, the significance of each predictor will be evaluated.

1.2.2. Significance of the Coefficient of `Apps` ($\hat{\beta}_1$)

As shown in Figure 1.2 (line 14), the coefficient of `Apps` ($\hat{\beta}_1$) is -0.38616 with a p-value of 7.68×10^{-5}. The hypothesis being tested is:

$$H_0: \beta_1 = 0$$

$$H_1: \beta_1 \neq 0$$

At a 5% significance level ($\alpha = 0.05$), the p-value for the coefficient of `Apps` ($\hat{\beta}_1$) is smaller than α. Therefore, the null hypothesis is rejected at the 5% level. This indicates a statistically significant negative relationship between the predictor `Apps` and the response variable `Outstate`.

1.2.3. Significance of the Coefficient of `Accept` ($\widehat{\beta}_2$)

Based on Figure 1.2 (line 15), the coefficient of `Accept` ($\hat{\beta}_2$) is 1.58318 with a p-value of less than 2×10^{-16}. The hypothesis being tested is:

$$H_0: \beta_2 = 0$$
$$H_1: \beta_2 \neq 0$$

At a 5% significance level ($\alpha = 0.05$), the p-value for the coefficient of `Accept` ($\hat{\beta}_2$) is smaller than α. Therefore, the null hypothesis is rejected at the 5% level. This indicates a statistically significant positive relationship between the predictor `Accept` and the response variable `Outstate`.

1.2.4. Significance of the Coefficient of `Enroll` ($\widehat{\beta}_3$)

Referring to Figure 1.2 (line 16), the coefficient of `Enroll` ($\hat{\beta}_3$) is -3.61420 with a p-value of less than 2×10^{-16}. The hypothesis being tested is:

$$H_0: \beta_3 = 0$$
$$H_1: \beta_3 \neq 0$$

At a 5% significance level ($\alpha = 0.05$), the p-value for the coefficient of `Enroll` ($\hat{\beta}_3$) is smaller than α. Therefore, the null hypothesis is rejected at the 5% level. This indicates a statistically significant negative relationship between the predictor `Enroll` and the response variable `Outstate`.

1.2.5. Significance of the Coefficient of `Top10perc` ($\hat{\beta}_4$)

As indicated by Figure 1.2 (line 17), the coefficient of `Top10perc` ($\hat{\beta}_4$) is 161.76486 with a p-value of less than 2×10^{-16}. The hypothesis being tested is:

$$H_0: \beta_4 = 0$$
$$H_1: \beta_4 \neq 0$$

At a 5% significance level ($\alpha = 0.05$), the p-value for the coefficient of `Top10perc` ($\hat{\beta}_4$) is smaller than α. Therefore, the null hypothesis is rejected at the 5% level. This indicates a statistically significant positive relationship between the predictor `Top10perc` and the response variable `Outstate`.

1.2.6. Significance of the Coefficient of `Top25perc` ($\hat{\beta}_5$)

Figure 1.2 (line 18) reveals that the coefficient of `Top25perc` ($\hat{\beta}_5$) is -12.61806 with a p-value of 0.305. The hypothesis being tested is:

$$H_0: \beta_5 = 0$$
$$H_1: \beta_5 \neq 0$$

At a 5% significance level ($\alpha = 0.05$), the p-value for the coefficient of `Top25perc` ($\hat{\beta}_5$) is larger than α. Therefore, the null hypothesis fails to be rejected at the 5% level. This indicates that there is no statistically significant relationship between the predictor `Top25perc` and the response variable `Outstate`.

In summary, the overall model is statistically significant, with several predictors showing strong statistical significance in explaining the variability in the response variable `Outstate`. Four out of five predictors (`Apps`, `Accept`, `Enroll`, and `Top10perc`) are statistically significant, whereas `Top25perc` is not.

1.3. Diagnostic Plots

To further assess the model's adequacy, it is essential to examine the diagnostic plots. These plots can help identify issues such as non-linearity, heteroscedasticity, and influential observations. Figure 1.3 shows the R code for generating diagnostic plots and a residual plot for the multiple linear regression model. Figure 1.4 displays some diagnostic plots for the multiple linear regression model. There include the Residuals vs. Fitted plot, the QQ plot, the Scale-Location plot, and the Residuals vs. Leverage plot. Figure 1.5 presents the studentised residual plot.

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

# Studentised residual plot for the original model
par(mfrow = c(1, 1))
plot(predict(model), rstudent(model))
```

Figure 1.3: R code for generating diagnostic plots and a studentised residual plot for the multiple linear regression model

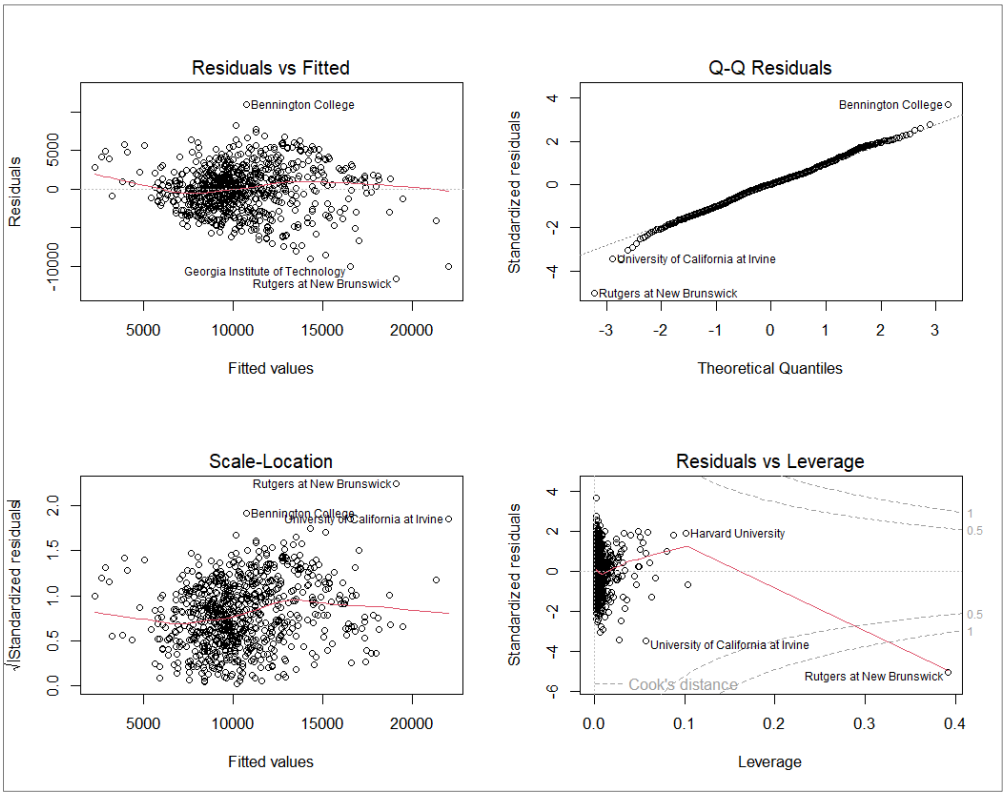

Figure 1.4: Some diagnostic plots for the multiple linear regression model. Top left: Residuals vs. Fitted plot; Top right: QQ plot; Bottom left: Scale-Location plot; Bottom right: Residuals vs. Leverage plot

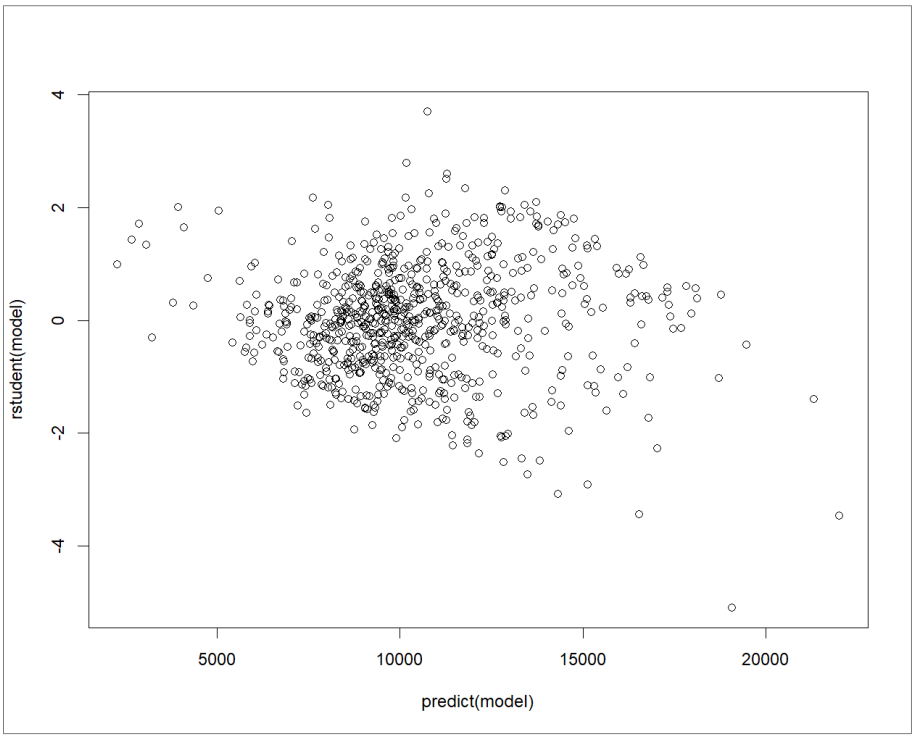

Figure 1.5: Studentised residual plot for the multiple linear regression model

1.3.1. Residuals vs. Fitted Plot

The Residuals vs. Fitted plot is used to check the linearity assumption. If the red line is approximately horizontal and there is no clear pattern in the residuals, the linearity assumption is likely satisfied. Ideally, the residuals should be randomly scattered around the horizontal line (residual $=$ 0), indicating a good fit.

In Figure 1.4, the Residuals vs. Fitted plot reveals a slight pattern, with the residuals showing some curvature. This suggests the presence of non-linear relationships that the model has not captured. Additionally, some outliers are visible, particularly Bennington College, Georgia Institute of Technology, and Rutgers at New Brunswick.

1.3.2. QQ Plot of Residuals

The QQ plot checks the normality of the residuals and assesses whether they follow a normal distribution. The residuals should lie approximately along the 45-degree reference line if they are normally distributed.

According to the QQ plot in Figure 1.4, the residuals generally follow the reference line, indicating that they are approximately normally distributed. However, there are deviations at

both tails, especially the upper tail. Bennington College, University of California at Irvine, and Rutgers at New Brunswick are identified as potential outliers.

1.3.3. Scale-Location Plot

The Scale-Location plot examines the homoscedasticity (constant variance) of the residuals. A horizontal red line with equally spread points indicates homoscedasticity. Ideally, the points should be randomly dispersed without any clear pattern.

Referring to the Scale-Location plot in Figure 1.4, the red line is not perfectly horizontal and shows a slight downward trend, indicating some heteroscedasticity. The spread of residuals varies somewhat across the range of fitted values. This suggests that the assumption of constant variance might be violated to some degree.

1.3.4. Residuals vs. Leverage Plot

The Residuals vs. Leverage plot identifies influential data points that might unduly affect the model. Points outside Cook's distance lines are considered influential.

In Figure 1.4, the Residuals vs. Leverage plot shows that most observations have low leverage. However, a few observations, such as Harvard University and University of California at Irvine, exhibit high leverage or large residuals. Moreover, Rutgers at New Brunswick is outside the Cook's distance lines, indicating it might be an influential point.

1.3.5. Studentised Residual Plot

As seen in the studentised residual plot in Figure 1.5, there are possible outliers, indicated by data points with values greater than 3 and less than -3.

1.4. Issues Identified with the Fit

The diagnostic plots shown earlier in Figure 1.4 and Figure 1.5 reveal several issues with the model fit that require attention. Firstly, the Residuals vs. Fitted plot indicates evidence of non-linearity in the relationship between predictors and the response variable, as shown by the slight curved pattern. This suggests that the current linear model may not fully capture the complexity of the underlying relationships.

The QQ plot indicates a departure from normality in the residuals, particularly at the tails of the distribution. This non-normality can affect the validity of statistical tests and confidence intervals derived from the model, potentially leading to incorrect inferences.

Heteroscedasticity is another concern, visible in the Scale-Location plot. The uneven spread of residuals across fitted values violates the assumption of constant variance, which can lead to unreliable standard errors and confidence intervals.

The Residuals vs. Leverage plot also highlights points with high leverage, particularly Rutgers at New Brunswick, which can have a strong influence on the regression line.

Several data points stand out as potential outliers or highly influential observations, including Bennington College, Georgia Institute of Technology, Rutgers at New Brunswick, University of California at Irvine, and Harvard University. These points may be disproportionately affecting the model's estimates and could be skewing the results.

1.5. Suggested Transformations to Improve the Model

1.5.1. Log Transformation of the Response Variable `Outstate`

To address the identified issues, applying a logarithmic transformation to the response variable `Outstate` can improve the model. This transformation can help stabilise the variance, linearise the relationship, and address heteroscedasticity, making the relationship between the predictors and the response variable more linear. Figure 1.6 shows the R code for fitting the log-transformed multiple linear regression model, while Figure 1.7 displays the summary of the model.

```r
# Log transformation of the response variable `Outstate`
model_log = lm(log(Outstate) ~ Apps + Accept + Enroll + Top10perc + Top25perc, data = College)
```

Figure 1.6: R code for fitting the log-transformed multiple linear regression model

```
> summary(model_log)

Call:
lm(formula = log(Outstate) ~ Apps + Accept + Enroll + Top10perc +
    Top25perc, data = College)

Residuals:
     Min       1Q   Median       3Q      Max
-1.21759 -0.18758  0.04661  0.21695  0.77072

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.899e+00  4.114e-02 216.330  < 2e-16 ***
Apps        -4.711e-05  1.016e-05  -4.637 4.15e-06 ***
Accept       1.842e-04  1.954e-05   9.429  < 2e-16 ***
Enroll      -4.032e-04  2.954e-05 -13.649  < 2e-16 ***
Top10perc    1.442e-02  1.515e-03   9.516  < 2e-16 ***
Top25perc   -6.596e-04  1.285e-03  -0.513    0.608
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3117 on 771 degrees of freedom
Multiple R-squared:  0.4116,    Adjusted R-squared:  0.4078
F-statistic: 107.9 on 5 and 771 DF,  p-value: < 2.2e-16
```

Figure 1.7: Summary of the log-transformed multiple linear regression model

According to Figure 1.7 (line 24), the overall F-statistic for the model is 107.9 with a p-value of less than 2.2×10^{-16}, indicating that the model is still statistically significant. The significance of individual predictors is largely unchanged, with the same predictors being significant or non-significant as in the original model. The RSE has decreased from 2980 in the original model to 0.3117 (line 22), which is expected due to the log transformation of the response variable `` `Outstate` ``. However, the Multiple R-squared value has slightly decreased from 0.4548 in the original model to 0.4116 (line 23) in the log-transformed model, suggesting that the log transformation does not improve the overall explanatory power of the model.

Next, Figure 1.8 and Figure 1.9 display some of the diagnostic plots for the log-transformed multiple linear regression model.

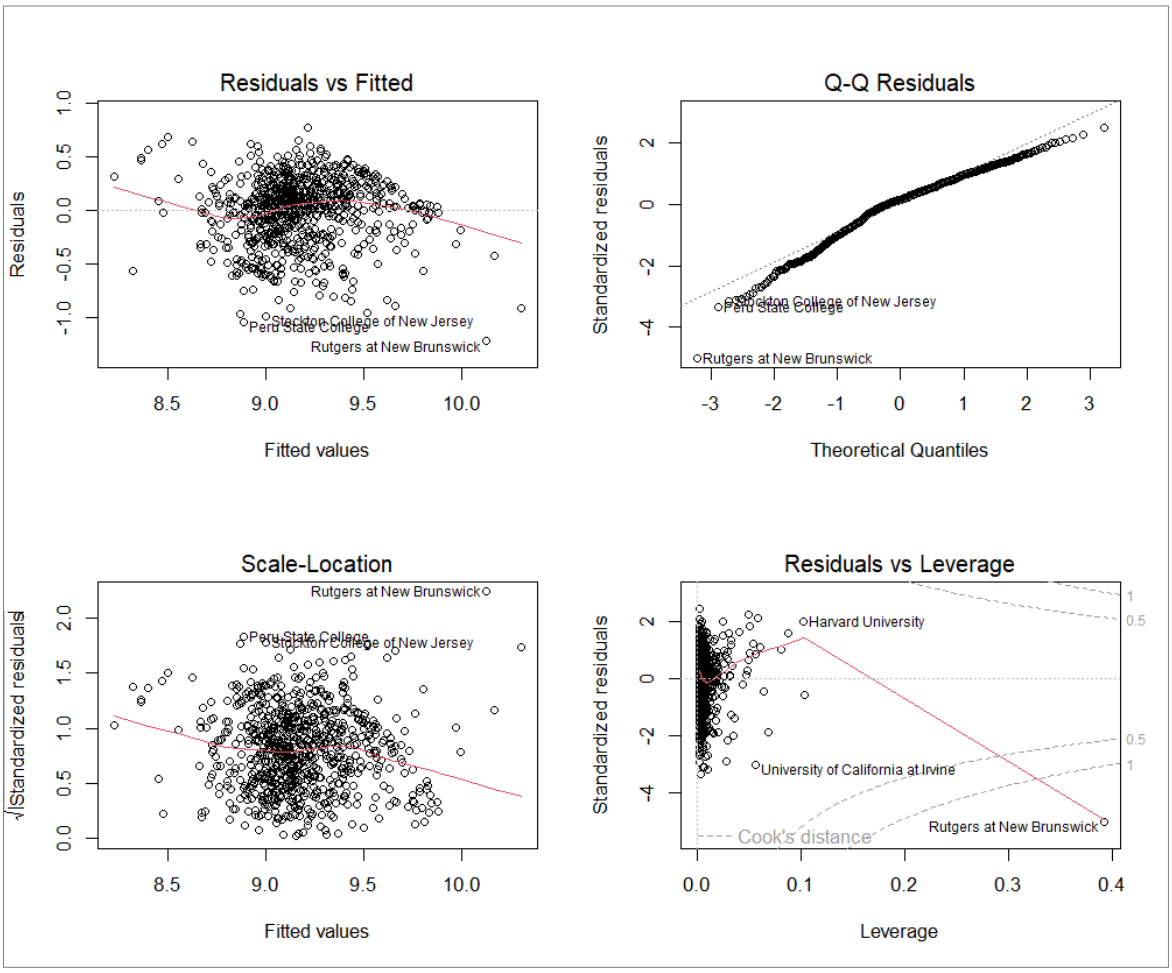

Figure 1.8: Some diagnostic plots for the log-transformed multiple linear regression model. Top left: Residuals vs. Fitted plot; Top right: QQ plot; Bottom left: Scale-Location plot; Bottom right: Residuals vs. Leverage plot

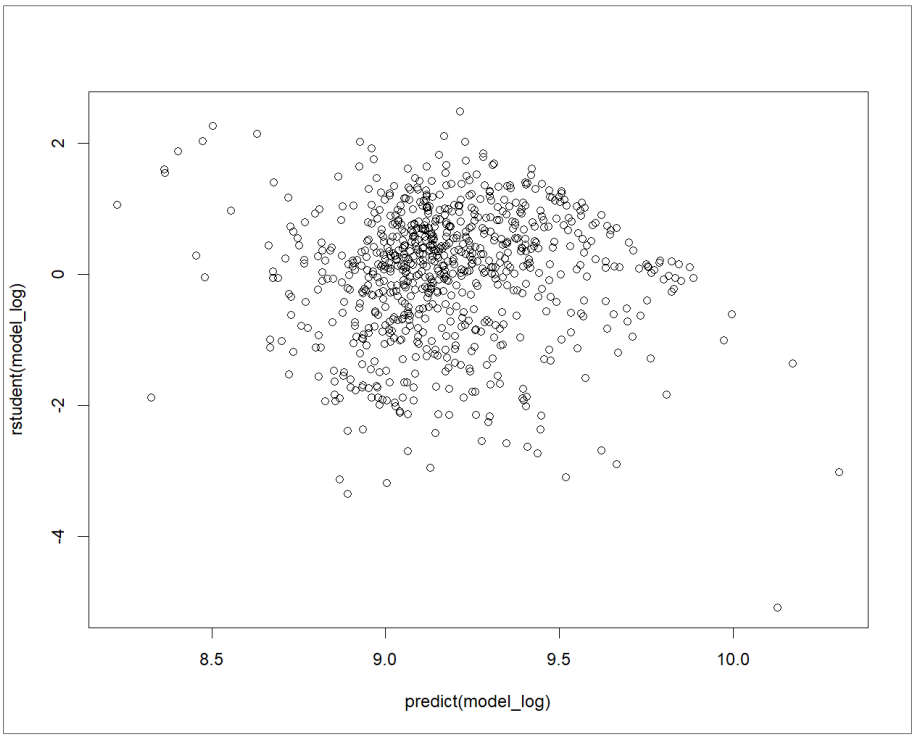

Figure 1.9: Studentised residual plot for the log-transformed multiple linear regression model

Based on the diagnostic plots in Figure 1.8 and Figure 1.9, the log transformation of the response variable `Outstate` has improved the consistency of residual spread across fitted values. The residuals are closer to a normal distribution, particularly in the middle range. The range of residuals is now smaller and more manageable. Nevertheless, some non-linear patterns are still visible in the Residuals vs. Fitted plot. There is still some variation in residual spread, although it is much improved. While less extreme, outliers are still present in the studentised residual plot.

1.5.2. Square Root Transformation of Some Predictors

Another useful transformation is applying a square root transformation to some predictors. This transformation can also help linearise their relationship with the response variable `Outstate` and potentially reduce the impact of outliers in these predictors. Figure 1.10 shows the R code for fitting the multiple linear regression model with the square root transformation of the predictors `Apps`, `Accept`, and `Enroll`, while Figure 1.11 displays the summary of the model.

```r
# Square root transformation of the predictors `Apps`, `Accept`, and `Enroll`
model_sqrt = lm(Outstate ~ sqrt(Apps) + sqrt(Accept) + sqrt(Enroll) + Top10perc + Top25perc, data = College)
```

Figure 1.10: R code for fitting the square root-transformed multiple linear regression model

```
> summary(model_sqrt)

Call:
lm(formula = Outstate ~ sqrt(Apps) + sqrt(Accept) + sqrt(Enroll) +
    Top10perc + Top25perc, data = College)

Residuals:
     Min       1Q   Median       3Q      Max
-11330.0  -1808.4     77.6   1767.0  10600.3

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   8529.54     394.15  21.641   <2e-16 ***
sqrt(Apps)     -31.34      15.92  -1.968   0.0494 *
sqrt(Accept)   235.49      23.95   9.831   <2e-16 ***
sqrt(Enroll)  -379.68      23.32 -16.284   <2e-16 ***
Top10perc      152.63      13.83  11.034   <2e-16 ***
Top25perc      -13.89      11.70  -1.187   0.2355
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2837 on 771 degrees of freedom
Multiple R-squared:  0.5059,    Adjusted R-squared:  0.5027
F-statistic: 157.9 on 5 and 771 DF,  p-value: < 2.2e-16
```

Figure 1.11: Summary of the square root-transformed multiple linear regression model

According to Figure 1.11 (line 24), the overall F-statistic for the model is 157.9 with a p-value of less than 2.2×10^{-16}, indicating that the model is still statistically significant. The significance of individual predictors is unchanged, with the same predictors being significant or non-significant as in the original model. The RSE has decreased from 2980 in the original model to 2837 (line 22), indicating a slight improvement in the model's precision. The Multiple R-squared value has increased from 0.4548 in the original model to 0.5059 (line 23) in the square root-transformed model, suggesting that the square root transformation has improved the overall explanatory power of the model.

Next, Figure 1.12 and Figure 1.13 display some of the diagnostic plots for the multiple linear regression model with square root transformation of the predictors `Apps`, `Accept`, and `Enroll`.

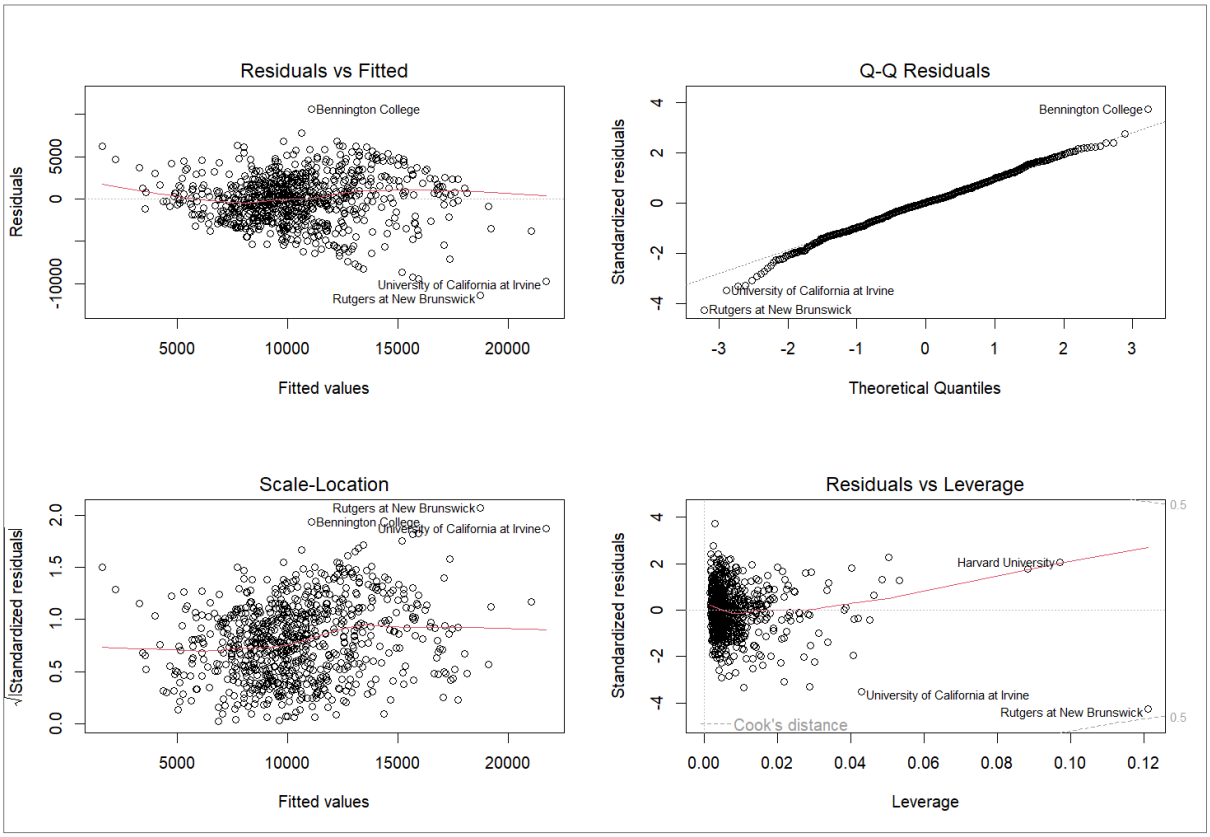

Figure 1.12: Some diagnostic plots for the square root-transformed multiple linear regression model. Top left: Residuals vs. Fitted plot; Top right: QQ plot; Bottom left: Scale-Location plot; Bottom right: Residuals vs. Leverage plot

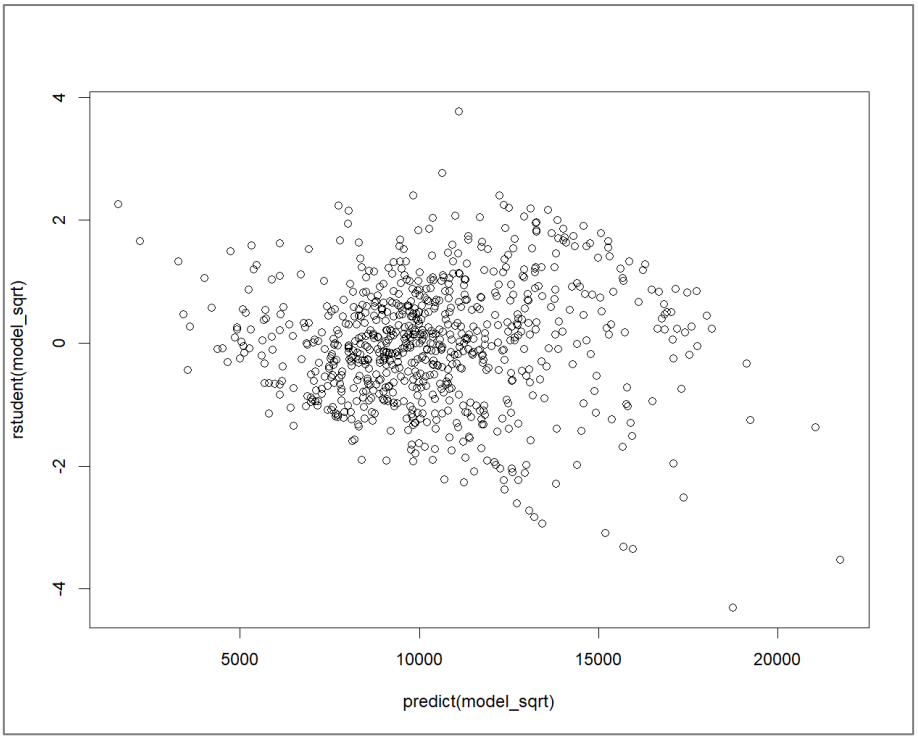

Figure 1.13: Studentised residual plot for the square root-transformed multiple linear regression model

Based on the diagnostic plots in Figure 1.12 and Figure 1.13, a more random scatter points is observed from the Residuals vs. Fitted plot. The QQ plot demonstrates improved normality of residuals, especially in the mid-range, although some deviations persist at the tails. Additionally, some outliers and high-leverage points are still present.

2. Best Subset Selection for Optimal Predictors

2.1. Best Subset Selection Procedure

To perform the best subset selection, all predictors will be utilised to identify the optimal predictors for modelling the response variable `Outstate`. Figure 2.1 shows the R code used to perform best subset selection using all predictors, using the `leaps` library. Figure 2.2 presents the summary output of the best subset selection, providing a clear picture of how predictors are selected across different model sizes.

```r
# Load the necessary library for best subset selection
library(leaps)

# Fit the full model with all predictors to perform best subset selection
regfit.full = regsubsets(Outstate ~ ., data = College, nvmax = 17)
```

Figure 2.1: R code for performing best subset selection

```
> reg.summary = summary(regfit.full)
> reg.summary
Subset selection object
Call: regsubsets.formula(Outstate ~ ., data = College, nvmax = 17)
17 Variables  (and intercept)
            Forced in Forced out
PrivateYes      FALSE      FALSE
Apps            FALSE      FALSE
Accept          FALSE      FALSE
Enroll          FALSE      FALSE
Top10perc       FALSE      FALSE
Top25perc       FALSE      FALSE
F.Undergrad     FALSE      FALSE
P.Undergrad     FALSE      FALSE
Room.Board      FALSE      FALSE
Books           FALSE      FALSE
Personal        FALSE      FALSE
PhD             FALSE      FALSE
Terminal        FALSE      FALSE
S.F.Ratio       FALSE      FALSE
perc.alumni     FALSE      FALSE
Expend          FALSE      FALSE
Grad.Rate       FALSE      FALSE
1 subsets of each size up to 17
Selection Algorithm: exhaustive
          PrivateYes Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Room.Board Books Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate
1  ( 1 )  " "        " "  " "    " "    " "       " "       " "         " "         " "        " "   " "      " " " "      " "        " "          "*"    " "
2  ( 1 )  "*"        " "  " "    " "    " "       " "       " "         " "         " "        " "   " "      " " " "      " "        " "          "*"    " "
3  ( 1 )  "*"        " "  " "    " "    " "       " "       " "         " "         " "        "*"   " "      " " " "      " "        " "          "*"    " "
4  ( 1 )  "*"        " "  " "    " "    " "       " "       " "         " "         " "        "*"   " "      " " " "      " "        "*"          "*"    " "
5  ( 1 )  "*"        " "  " "    " "    " "       " "       " "         " "         " "        "*"   " "      " " "*"      " "        "*"          "*"    " "
6  ( 1 )  "*"        " "  " "    " "    " "       " "       " "         " "         " "        "*"   " "      " " "*"      " "        "*"          "*"    "*"
7  ( 1 )  "*"        " "  " "    " "    " "       " "       " "         " "         " "        "*"   " "      "*" "*"      " "        "*"          "*"    "*"
8  ( 1 )  "*"        " "  "*"    " "    " "       " "       "*"         " "         "*"        " "   " "      "*" "*"      " "        "*"          "*"    "*"
9  ( 1 )  "*"        "*"  "*"    " "    " "       " "       "*"         " "         "*"        " "   " "      "*" "*"      " "        "*"          "*"    "*"
10  ( 1 ) "*"        "*"  "*"    " "    "*"       " "       "*"         " "         "*"        " "   " "      "*" "*"      " "        "*"          "*"    "*"
11  ( 1 ) "*"        "*"  "*"    " "    "*"       " "       "*"         " "         "*"        " "   "*"      "*" "*"      " "        "*"          "*"    "*"
12  ( 1 ) "*"        "*"  "*"    " "    "*"       " "       "*"         " "         "*"        " "   "*"      "*" "*"      "*"        "*"          "*"    "*"
13  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       "*"         " "         "*"        " "   "*"      "*" "*"      "*"        "*"          "*"    "*"
14  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       "*"         " "         "*"        " "   "*"      "*" "*"      "*"        "*"          "*"    "*"
15  ( 1 ) "*"        "*"  "*"    "*"    "*"       " "       "*"         " "         "*"        "*"   "*"      "*" "*"      "*"        "*"          "*"    "*"
16  ( 1 ) "*"        "*"  "*"    "*"    "*"       "*"       "*"         " "         "*"        "*"   "*"      "*" "*"      "*"        "*"          "*"    "*"
17  ( 1 ) "*"        "*"  "*"    "*"    "*"       "*"       "*"         "*"         "*"        "*"   "*"      "*" "*"      "*"        "*"          "*"    "*"
```

Figure 2.2: Summary of the best subset selection

2.2. Models Obtained and Evaluation Metrics

To determine the optimal model size, evaluation metrics such as Adjusted R-squared, Mallows' C_p, and Bayesian Information Criterion (BIC) will be utilised. Figure 2.3 shows a plot of the number of predictors against each of these evaluation metrics. According to the plot, the Adjusted R-squared value increases sharply with the addition of predictors initially, then begins to level off. This suggests that while adding predictors initially enhances the model's explanatory power, the benefits become marginal beyond a certain point. In contrast, the C_p value generally decreases as the number of predictors increases. The BIC value decreases sharply at first, but then shows a slight increase with the addition of more predictors.

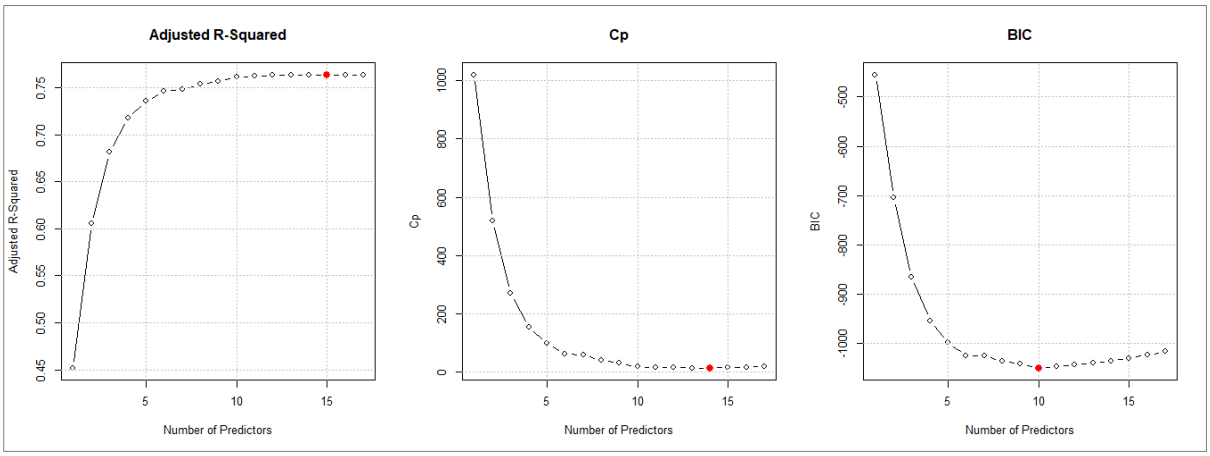

Figure 2.3: Model selection criteria for subset regression applied to the "College" dataset. Left: Adjusted R-squared; Middle: C_p; Right: BIC

2.2.1. Adjusted R-squared

The model with the highest Adjusted R-Squared value is preferred as it indicates a higher proportion of variance explained by the model, adjusted for the number of predictors. Figure 2.4 shows the best model based on the highest Adjusted R-squared value. This model includes 15 out of 17 possible predictors, excluding only `` `P.Undergrad` `` and `` `Top25perc` ``.

```
> # Print the best model based on Adjusted R-squared and display the coefficients
> cat("Best model by Adjusted R-squared: ", which.max(reg.summary$adjr2), "\n")
Best model by Adjusted R-squared:  15
> print(coef(regfit.full, which.max(reg.summary$adjr2)))
  (Intercept)    PrivateYes          Apps        Accept        Enroll     Top10perc
-1.647456e+03  2.263196e+03 -2.994326e-01  8.022893e-01 -5.366481e-01  2.427876e+01
  F.Undergrad    Room.Board         Books      Personal           PhD      Terminal
-9.509565e-02  8.836488e-01 -4.637800e-01 -2.267671e-01  1.134333e+01  2.417116e+01
    S.F.Ratio   perc.alumni        Expend     Grad.Rate
-4.638875e+01  4.155348e+01  2.009265e-01  2.364651e+01
```

Figure 2.4: The best model based on Adjusted R-squared

2.2.2. Mallows' C_p

The model with the lowest C_p value is considered the best, indicating the most appropriate trade-off between goodness-of-fit and model complexity. Figure 2.5 shows the best model based on the lowest C_p value. This model includes 14 out of 17 possible predictors, closely resembling the Adjusted R-squared model, but excluding the predictor `` `Books` ``.

```
> # Print the best model based on Cp and display the coefficients
> cat("Best model by Cp: ", which.min(reg.summary$cp), "\n")
Best model by Cp:  14
> print(coef(regfit.full, which.min(reg.summary$cp)))
  (Intercept)    PrivateYes          Apps        Accept        Enroll     Top10perc
-1.817040e+03  2.256946e+03 -2.999022e-01  8.023519e-01 -5.372545e-01  2.365529e+01
  F.Undergrad    Room.Board      Personal           PhD      Terminal     S.F.Ratio
-9.569936e-02  8.741819e-01 -2.478418e-01  1.269506e+01  2.297296e+01 -4.700560e+01
  perc.alumni        Expend     Grad.Rate
 4.195006e+01  2.003912e-01  2.383197e+01
```

Figure 2.5: The best model based on C_p

2.2.3. Bayesian Information Criterion (BIC)

The model with the lowest BIC value is preferred, as BIC imposes a larger penalty for model complexity compared to Adjusted R-squared and C_p, thus favouring more parsimonious models. Figure 2.6 shows the best model based on the lowest BIC value. This model includes only 10 predictors, the fewest among all three metrics.

```
# Print the best model based on BIC and display the coefficients
cat("Best model by BIC: ", which.min(reg.summary$bic), "\n")
print(coef(regfit.full, which.min(reg.summary$bic)))

> # Print the best model based on BIC and display the coefficients
> cat("Best model by BIC: ", which.min(reg.summary$bic), "\n")
Best model by BIC:  10
> print(coef(regfit.full, which.min(reg.summary$bic)))
  (Intercept)    PrivateYes          Apps        Accept     Top10perc   F.Undergrad
-3253.0008956  2317.2502512    -0.3020726     0.7609174    24.9103848    -0.1887515
   Room.Board      Terminal   perc.alumni        Expend     Grad.Rate
    0.9064012    33.9920350    44.0986438     0.2146827    25.2864716
```

Figure 2.6: The best model based on BIC

2.3. Interpretation of Best Models

In summary, all three models include the predictors `PrivateYes`, `Apps`, `Accept`, `Top10perc`, `F.Undergrad`, `Room.Board`, `Terminal`, `perc.alumni`, `Expend`, and `Grad.Rate`. These variables are likely the most important predictors of the response variable `Outstate`. The optimal number of predictors should range between 10 and 15, as indicated by the BIC metric (suggesting a minimum of 10 predictors) and the Adjusted R-squared metric (suggesting a maximum of 15 predictors).

3. Evaluation of Polynomial Regression Models for Predicting `Top10perc`

This section of the assignment involves using the predictor `Apps` to evaluate the predictive performance of polynomial regression models in predicting the response variable `Top10perc` across different polynomial degrees. Various cross-validation techniques, including Holdout Method, Leave-One-Out Cross-Validation (LOOCV), and k-Fold Cross-Validation, will be employed to assess the models' performance. The Mean Squared Error (MSE) will be calculated for each model using the three cross-validation approaches. The performance of each polynomial model will be evaluated based on the MSE, and the results will be summarised and compared to identify the polynomial degree that offers the best predictive performance.

3.1. Fitting Polynomial Regression Models with Different Cross-Validation Approaches

Figure 3.1 presents the R code for fitting polynomial regression models using the Holdout Method, LOOCV, and k-Fold Cross-Validation. On line 2, a seed value of 1 is set to ensure reproducibility of the results. A function named `calculate_mse` is then created on line 5, which takes five arguments: `data`, `response`, `predictor`, `method`, and `k`.

On line 10, a numeric vector `mse_values` is initialised to store the MSE values for each polynomial degree from 1 to 10. In the Holdout Method, beginning on line 13, the dataset is split into training (70%) and testing (30%) sets. For each polynomial degree from 1 to 10, a polynomial regression model is fitted on the training set (lines 20-22). The model's predictions are then made on the test set, and the MSE is calculated by comparing the predicted values to the actual values in the test set (lines 25-28).

For both LOOCV and k-Fold Cross-Validation, starting on line 33, the function loops through each polynomial degree, fitting a polynomial regression model on the entire dataset. For LOOCV, the MSE is computed using the `cv.glm` function (lines 39). The function `cv.glm` returns an object that includes the element `delta`, where `delta[1]` represents the estimated cross-validation error (MSE in this case). For k-Fold Cross-Validation, the dataset is divided into k folds (with k set to 10 in this analysis), and the `cv.glm` function is also used to compute the MSE for each fold (line 44). On line 50, the

function returns a vector of MSE values corresponding to each polynomial degree for the specified cross-validation approach.

Finally, the `calculate_mse` function is applied to the "College" dataset on lines 54-56 to predict the response variable `Top10perc` using `Apps` as the predictor. The function is executed three times, each with a different cross-validation approach: Holdout Method, LOOCV, and *k*-Fold Cross-Validation.

```r
# Set the seed for reproducibility
set.seed(1)

# Function to perform cross-validation and calculate MSE for different polynomial degrees
calculate_mse = function(data, response, predictor, method = "holdout", k = 10) {
    # Define the range of polynomial degrees to evaluate
    degrees = 1:10

    # Initialise a numeric vector to store MSE values for each polynomial degree
    mse_values = numeric(length(degrees))

    # Calculate MSE using Holdout Method
    if (method == "holdout") {
        # Split the data into training (70%) and testing (30%) sets
        train_index = sample(1:nrow(data), 0.7 * nrow(data))
        train_data = data[train_index, ]
        test_data = data[-train_index, ]

        # Loop through polynomial degrees from 1 to 10
        for (degree in degrees) {
            # Fit the model using polynomial regression of the specified degree
            model = lm(as.formula(paste(response, "~ poly(", predictor, ",", degree, ")")), data = train_data)

            # Predict the response variable on the test data
            predictions = predict(model, newdata = test_data)

            # Calculate MSE for the current degree and store it in `mse_values`
            mse_values[degree] = mean((test_data[[response]] - predictions)^2)
        }

    # Calculate MSE using Leave-One-Out Cross-Validation (LOOCV) or k-Fold Cross-Validation
    } else if (method == "loocv" || method == "kfold") {
        for (degree in degrees) {
            # Fit the model using polynomial regression of the specified degree
            model = glm(as.formula(paste(response, "~ poly(", predictor, ",", degree, ")")), data = data)

            # Calculate MSE using Leave-One-Out Cross-Validation (LOOCV)
            if (method == "loocv") {
                mse_values[degree] = cv.glm(data, model)$delta[1]

            # Calculate MSE using k-Fold Cross-Validation
            } else if (method == "kfold") {
                set.seed(1)
                mse_values[degree] = cv.glm(data, model, K = k)$delta[1]
            }
        }
    }

    # Return the vector of MSE values for each polynomial degree
    return(mse_values)
}

# Apply the function for each cross-validation approach
holdout_mse = calculate_mse(College, "Top10perc", "Apps", method = "holdout")
loocv_mse = calculate_mse(College, "Top10perc", "Apps", method = "loocv")
kfold_mse = calculate_mse(College, "Top10perc", "Apps", method = "kfold", k = 10)
```

Figure 3.1: R code for fitting polynomial regression models with Holdout Method, LOOCV, and k-Fold Cross-Validation

3.2. Results of Cross-Validation Analysis

Figure 3.2 shows the R code for creating and printing the data frames of MSE results from different cross-validation approaches: Holdout Method, LOOCV, and k-Fold Cross-Validation.

```r
# Function to create a results data frame
create_results_df = function(mse_values) {
    return(data.frame(
        Degree = 1:10,
        Model = c("linear", paste0("polyn", 2:10)),
        MSE = format(round(mse_values, 4), scientific = FALSE)
    ))
}

# Create separate data frames for each method
holdout_results = create_results_df(holdout_mse)
loocv_results = create_results_df(loocv_mse)
kfold_results = create_results_df(kfold_mse)

# Print the results
print("Cross-Validation Approach 1 - Holdout Method Results:")
print(holdout_results, row.names = FALSE)

print("Cross-Validation Approach 2 - Leave-One-Out Cross-Validation Results:")
print(loocv_results, row.names = FALSE)

print("Cross-Validation Approach 3 - k-Fold Cross-Validation Results:")
print(kfold_results, row.names = FALSE)
```

Figure 3.2: R code for creating and printing data frames of MSE results from different cross-validation approaches

3.2.1. Holdout Method

Figure 3.3 presents the MSE results for different polynomial degrees using the Holdout Method. The MSE values fluctuate slightly across different polynomial degrees, with no clear trend of improvement or deterioration as the degree increases. The differences in MSE values between models are relatively small. Although the lowest MSE value (263.6213) is achieved with a 7^{th}-degree polynomial, high-degree polynomials are generally not preferred due to their complexity. Therefore, for a better balance between performance and simplicity, a 1^{st}-degree (linear) polynomial is preferred in this case due to its relatively low MSE value (268.6026).

```
> print("Cross-Validation Approach 1 - Holdout Method Results:")
[1] "Cross-Validation Approach 1 - Holdout Method Results:"
> print(holdout_results, row.names = FALSE)
 Degree   Model      MSE
      1  linear 268.6026
      2  polyn2 272.8774
      3  polyn3 273.0092
      4  polyn4 272.5021
      5  polyn5 267.9493
      6  polyn6 264.8389
      7  polyn7 263.6213
      8  polyn8 264.1425
      9  polyn9 266.9854
     10 polyn10 266.7922
```

Figure 3.3: MSE results of Holdout Method for different polynomial degrees

3.2.2. Leave-One-Out Cross-Validation (LOOCV)

Figure 3.4 shows the MSE results for different polynomial degrees using LOOCV. The MSE values remain relatively stable for degrees 1 to 3 but increase dramatically for higher degrees. This sharp increase in MSE value for degrees 4 and above indicates severe overfitting. The lowest MSE value (270.7443) is achieved with a 2^{nd}-degree (quadratic) polynomial.

```
> print("Cross-Validation Approach 2 - Leave-One-Out Cross-Validation Results:")
[1] "Cross-Validation Approach 2 - Leave-One-Out Cross-Validation Results:"
> print(loocv_results, row.names = FALSE)
 Degree   Model                 MSE
      1  linear            279.2049
      2  polyn2            270.7443
      3  polyn3            298.0793
      4  polyn4          64064.3748
      5  polyn5         755961.7297
      6  polyn6       30441711.2458
      7  polyn7      600861401.9378
      8  polyn8    45755638786.6521
      9  polyn9  7267589666073.3701
     10 polyn10 62477129874981.4375
```

Figure 3.4: MSE results of LOOCV for different polynomial degrees

3.2.3. k-Fold Cross-Validation

Figure 3.5 displays the MSE results for different polynomial degrees using k-fold Cross-Validation. The results are very similar to those obtained with LOOCV. There is a similar pattern of a dramatic increase in MSE values for higher-degree polynomials. The lowest MSE value (270.4458) is also achieved with a 2^{nd}-degree polynomial.

```
> print("Cross-Validation Approach 3 - k-Fold Cross-Validation Results:")
[1] "Cross-Validation Approach 3 - k-Fold Cross-Validation Results:"
> print(kfold_results, row.names = FALSE)
 Degree   Model                MSE
      1  linear           279.0846
      2  polyn2           270.7692
      3  polyn3           278.4125
      4  polyn4         92345.7905
      5  polyn5       1177624.5625
      6  polyn6      29819037.3692
      7  polyn7    1386992245.5845
      8  polyn8   30666942191.6457
      9  polyn9 4715990022122.0264
     10 polyn10 21172942606563.3203
```

Figure 3.5: MSE results of k-Fold Cross-Validation for different polynomial degrees

In summary, the cross-validation analysis suggests that a quadratic (2^{nd} degree) polynomial regression model of response variable `` `Top10perc` `` on the predictor `` `Apps` `` provides the best balance between model complexity and predictive performance. Higher-degree polynomials (4 and above) show clear signs of overfitting and should be avoided. The consistency between LOOCV and k-Fold Cross Validation results provides strong evidence for these conclusions.