The linear regression model

Advanced Methods for Regression and Classification

Rita Selimi

10/21/2024

Load the necessary libraries

```
if (!require(ISLR)) install.packages("ISLR")
```

```
## Loading required package: ISLR
```

```
library(ISLR)
```

1. Data Preparation

```
# Load the College dataset
data(College, package="ISLR")

# head(College)
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

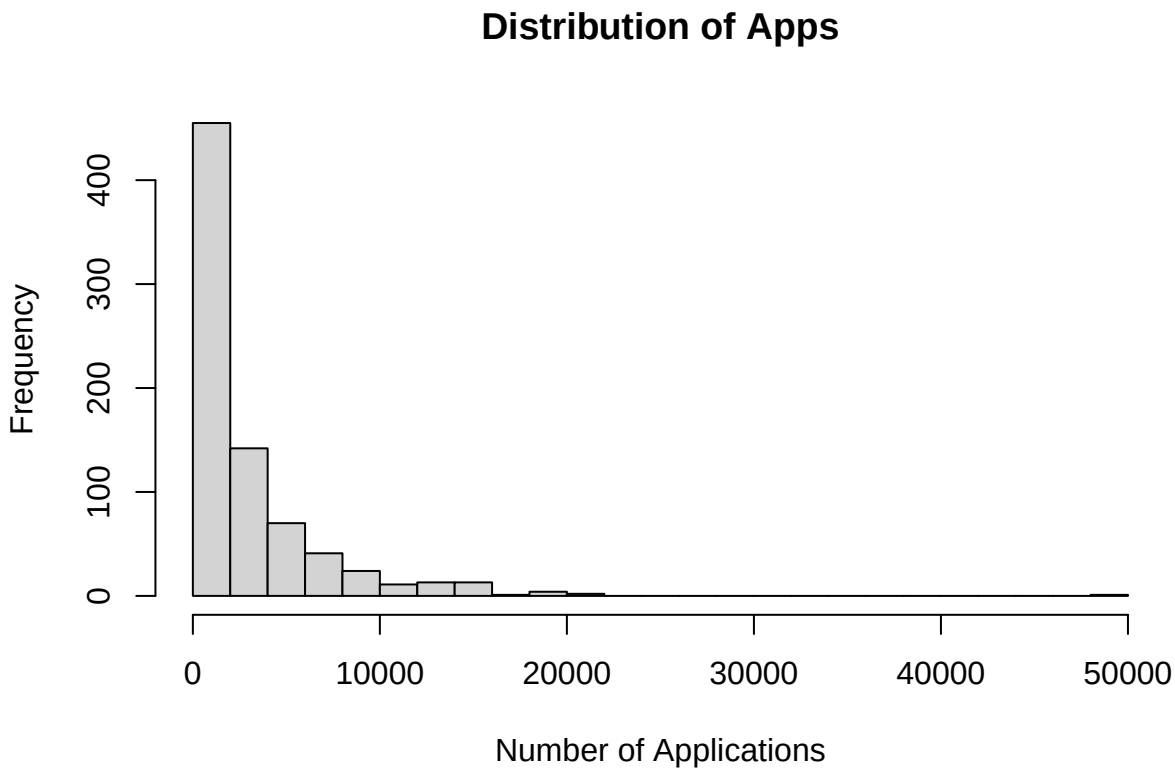
```r
# Check for missing data
print(paste("Number of missing values:", sum(is.na(College))))
```

```
## [1] "Number of missing values: 0"
```

```r
# Summary statistics of the number of applications
summary(College$Apps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      81     776    1558    3002    3624   48094
```

```r
# Visualizing the distribution of Apps to check for skewness
hist(College$Apps,
     main="Distribution of Apps",
     xlab="Number of Applications",
     breaks=30)
```



The `Apps` variable is highly skewed, with most schools having a small number of applications (median = 1558) but a few having very large numbers (maximum = 48094). Using a log transformation for `Apps` makes the data better fit for linear regression, like having normally distributed residuals and consistent variance. This helps the model work more effectively and focus on changes in applications, which is useful when dealing with a wide range of values. I'll proceed with log-transformed `Apps` as the response variable in this model. There is no need to remove missing values using `na.omit(College)` as no missing values were detected in the dataset.

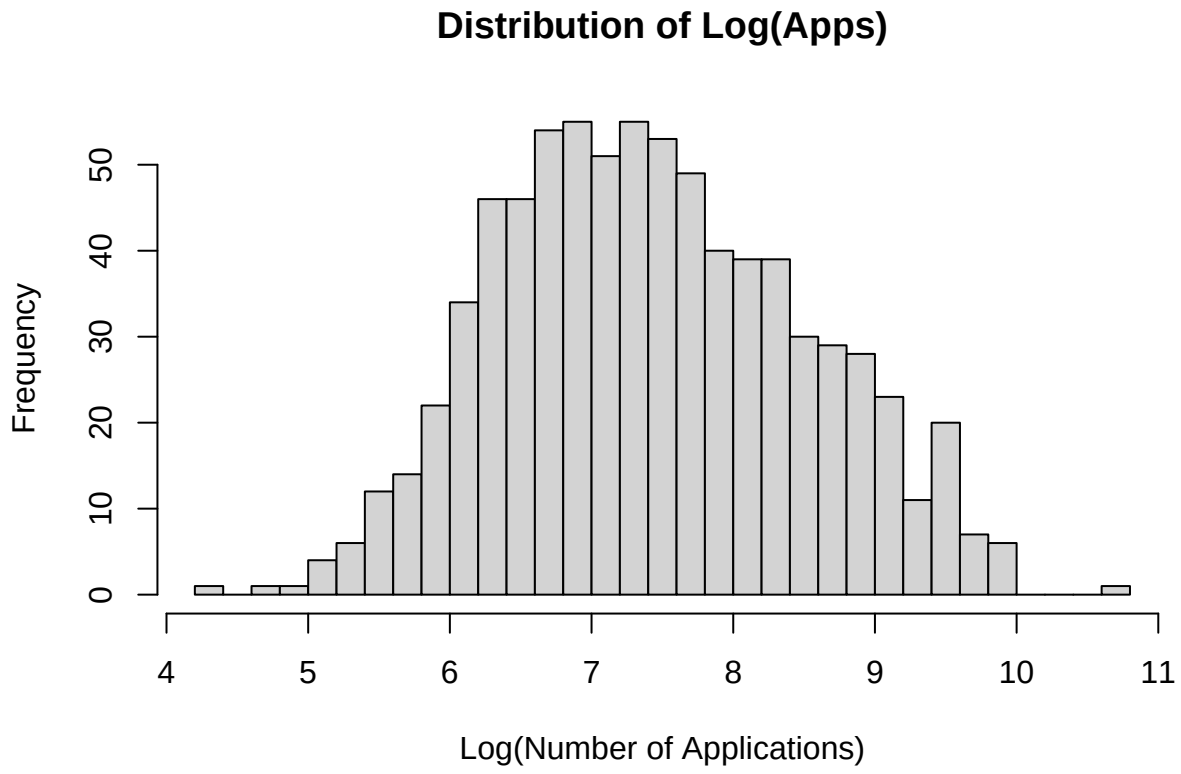
```r
college_clean <- College

# Apply log transformation to the 'Apps' variable
college_clean$Log_Apps <- log(college_clean$Apps)

summary(college_clean$Log_Apps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.394   6.654   7.351   7.427   8.195  10.781
```

```r
# Visualizing the distribution of Apps to check for skewness
hist(college_clean$Log_Apps,
     main = "Distribution of Log(Apps)",
     xlab = "Log(Number of Applications)",
     breaks = 30)
```



After the log transformation of `Apps`, the data now looks much more like a normal (bell-shaped) distribution. The mean (7.427) and median (7.351) are now very close, showing that the transformation has reduced skewness. The log transformation has also lessened the impact of outliers and made the data easier to work with.

```r
# Identify outliers using the IQR method
Q1 <- quantile(college_clean$Log_Apps, 0.25)
Q3 <- quantile(college_clean$Log_Apps, 0.75)
IQR_value <- Q3 - Q1
```

```r
lower_bound <- Q1 - 1.5 * IQR_value
upper_bound <- Q3 + 1.5 * IQR_value

# Find outliers
outliers <- college_clean$Log_Apps[
  college_clean$Log_Apps < lower_bound
  | college_clean$Log_Apps > upper_bound]
print(paste("Number of outliers:", outliers))
```

```
## [1] "Number of outliers: 10.7809127081884"
```

One outlier was identified from Rutgers at New Brunswick, and I decided to keep it since the high number of applications is realistic for a large, well-known public university.

2. Estimating the Full Regression Model

Splitting the data into training and test sets

```r
# Split the data into training and test sets (2/3 training, 1/3 test)
set.seed(12332281)
train_indices <- sample(1:nrow(college_clean), size = 2/3 * nrow(college_clean))
train_data <- college_clean[train_indices, ]
test_data <- college_clean[-train_indices, ]
```

a) Fit the full regression model on the training data

```r
# Fit the linear model on the training data, excluding Apps, Accept, and Enroll
res <- lm(Log_Apps ~ . - Apps - Accept - Enroll, data = train_data)
summary(res)
```

```
##
## Call:
## lm(formula = Log_Apps ~ . - Apps - Accept - Enroll, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.14787 -0.31515  0.04083  0.35495  1.73590
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.483e+00  2.571e-01  17.435  < 2e-16 ***
## PrivateYes  -6.071e-01  8.949e-02  -6.783 3.31e-11 ***
## Top10perc    4.095e-03  3.627e-03   1.129 0.259389
## Top25perc   -4.966e-04  2.854e-03  -0.174 0.861944
## F.Undergrad  1.208e-04  8.377e-06  14.421  < 2e-16 ***
## P.Undergrad  1.493e-05  2.474e-05   0.604 0.546341
## Outstate     4.743e-05  1.212e-05   3.914 0.000103 ***
## Room.Board   1.203e-04  3.190e-05   3.772 0.000181 ***
```

```
## Books        2.235e-04  1.459e-04   1.532 0.126263
## Personal    -9.795e-07  3.911e-05  -0.025 0.980031
## PhD          4.673e-03  2.949e-03   1.584 0.113745
## Terminal     1.013e-03  3.253e-03   0.311 0.755702
## S.F.Ratio    3.616e-02  7.991e-03   4.525 7.53e-06 ***
## perc.alumni -7.520e-03  2.647e-03  -2.841 0.004677 **
## Expend       1.838e-05  7.536e-06   2.439 0.015082 *
## Grad.Rate    1.176e-02  1.939e-03   6.062 2.65e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5563 on 502 degrees of freedom
## Multiple R-squared:  0.7175, Adjusted R-squared:  0.7091
## F-statistic: 85.02 on 15 and 502 DF,  p-value: < 2.2e-16
```

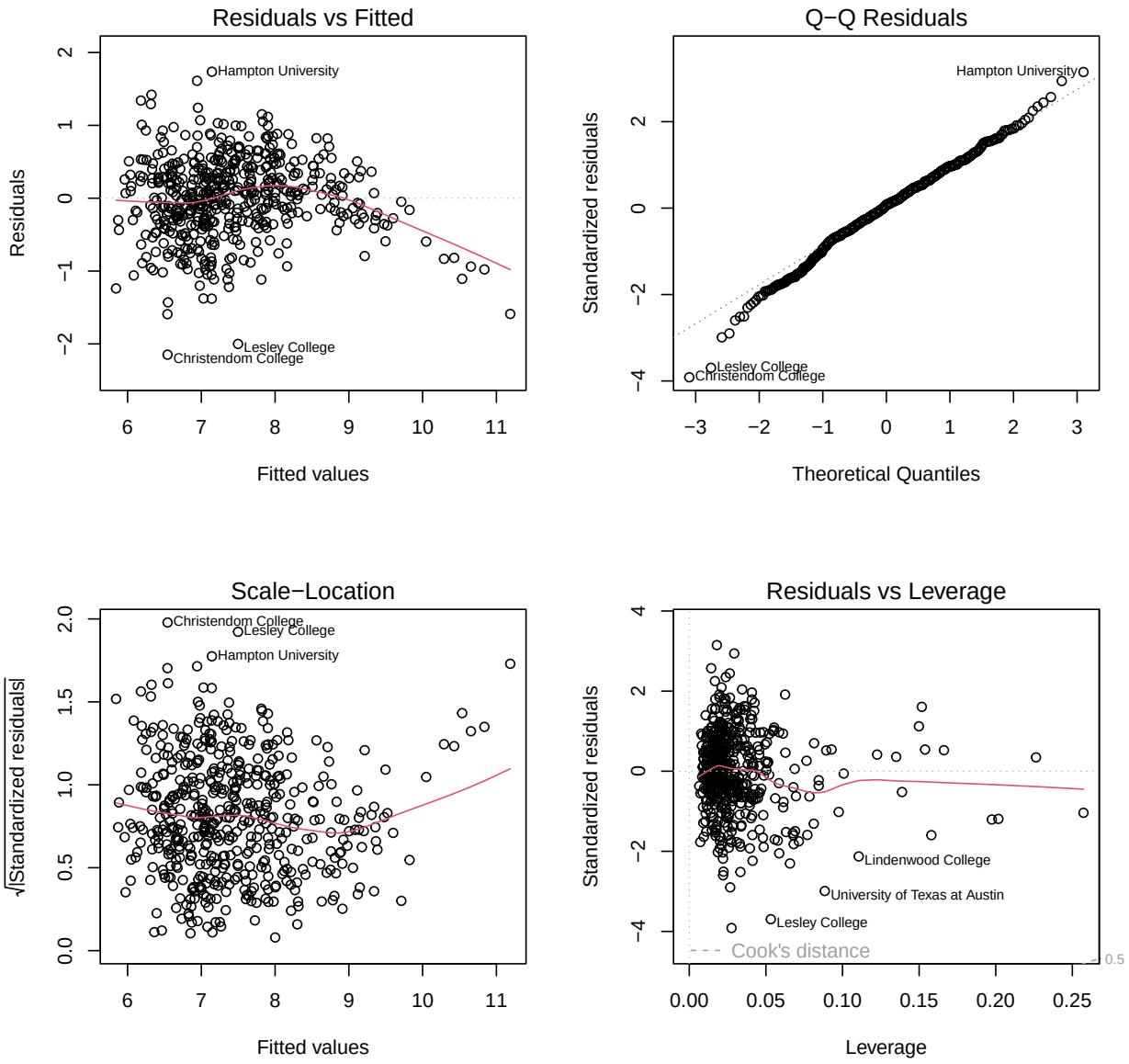
Variables with a p-value below 0.05 are significant contributors to the model. Significant predictors include: `PrivateYes`, `F.Undergrad`, `Outstate`, `Room.Board`, `S.F.Ratio`, `perc.alumni`, `Expend`, and `Grad.Rate`. Insignificant predictors include: `Top10perc`, `Top25perc`, `P.Undergrad`, `Books`, `PhD`, `Personal`, and `Terminal`.

The residuals show how well the model fits the data. The median residual (0.04) being close to zero means most predictions are near the actual values.

The coefficients show how each variable affects `Log_Apps`. For example, private schools has an estimate of -0.607, meaning they receive fewer applications than public ones, and having more full-time students, higher out-of-state tuition, and higher room and board costs all slightly increase applications.

The model explains approximately **71%** of the variation in why schools receive more or fewer applications, which indicates a strong fit. Even after adjusting for the number of predictors, the model still explains **70%** of the variation. This suggests that the model performs well. On average, the predictions are off by about **0.55 on the log scale**. Additionally, the model is highly significant, with a p-value of less than **2.2e-16**.

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



Residuals vs Fitted Plot (Linearity and Constant Variance): This plot shows if the residuals are randomly scattered around 0, checks the linearity assumption and helps identify patterns in the residuals. **What we see**: There's a slight curve, meaning the model might not fully capture the linear relationship. Also, the spread increases a bit for higher fitted values, suggesting the variance isn't completely constant.

Q-Q Plot (Normality of Residuals): This plot checks if the residuals follow a normal distribution. Points should fall close to the diagonal line. **What we see**: Most points follow the line, but some deviations appear at the ends, showing a few outliers or extreme values.

Scale-Location Plot (Constant Variance): This checks if the residuals have consistent spread (constant variance). A flat red line with points randomly spread out. **What we see**: The line curves slightly, and residuals spread more for larger fitted values, but it's not too extreme.

Residuals vs Leverage Plot (Influential Points): - This shows if any data points have a big influence on the model. Most points should be in the middle, with no strong outliers. **What we see**: A few points, have higher influence, but they don't seem to affect the model much.

b) Comparing Observed vs Predicted Values (Training and Test Data)

The formula for the **Least Squares Estimator**:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

```
# Get the design matrix X
X <- model.matrix(Log_Apps ~ . - Apps - Accept - Enroll, data = train_data)
Y <- train_data$Log_Apps

# Compute (X^T X)
XtX <- t(X) %*% X

# Compute (X^T X)^{-1}
XtX_inv <- solve(XtX)

# Compute X^T Y
XtY <- t(X) %*% Y

# Compute the LS estimator (beta coefficients)
beta_hat <- XtX_inv %*% XtY

beta_hat
```

```
##                      [,1]
## (Intercept)  4.482708e+00
## PrivateYes  -6.070692e-01
## Top10perc    4.095398e-03
## Top25perc   -4.965754e-04
## F.Undergrad  1.208087e-04
## P.Undergrad  1.493490e-05
## Outstate     4.742613e-05
## Room.Board   1.203286e-04
## Books        2.235141e-04
## Personal    -9.795192e-07
## PhD          4.672860e-03
## Terminal     1.012579e-03
## S.F.Ratio    3.616429e-02
## perc.alumni -7.520296e-03
## Expend       1.837985e-05
## Grad.Rate    1.175561e-02
```

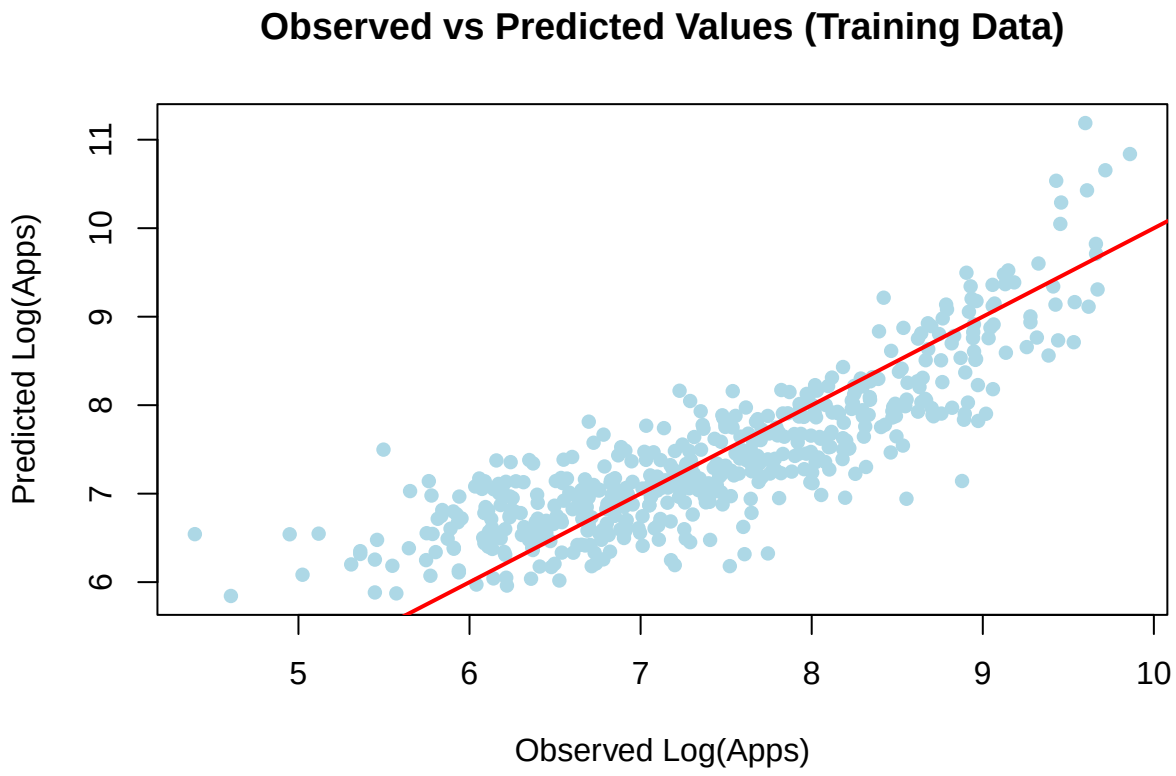
How is R handling binary variables (Private), and how can you interpret the corresponding regression coefficient? R automatically converts binary variables like `Private` (with levels "Yes" and "No") into dummy variables, when used in a regression model. In this case, **PrivateYes** is set to 1 for private schools and 0 for public schools. The coefficient for **PrivateYes** (-0.6232) means that private schools typically receive fewer applications compared to public schools, with the number of applications decreasing by 0.6232 units (on the log scale).

Compare the resulting coefficients with those obtained from lm(). The manually computed coefficients are nearly identical to those obtained from the `lm()` function. For each variable, the differences are extremely small and likely due to rounding, making them unimportant. This confirms that the manual calculation of the least squares (LS) estimator was performed correctly.

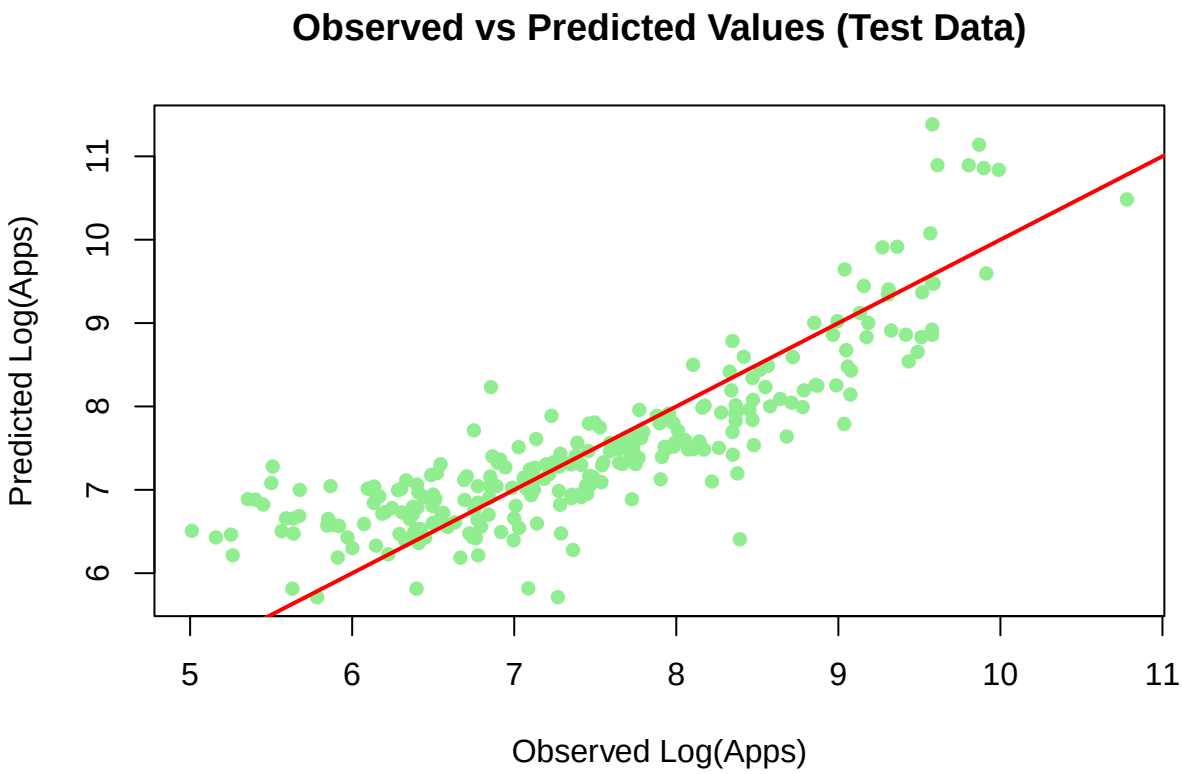
c) Comparing Observed vs Predicted Values Graphically (Training and Test Data)

```
train_pred <- predict(res, newdata = train_data)
test_pred <- predict(res, newdata = test_data)

# Plot for Training Data
plot(train_data$Log_Apps, train_pred,
     main = "Observed vs Predicted Values (Training Data)",
     xlab = "Observed Log(Apps)",
     ylab = "Predicted Log(Apps)",
     pch = 16, col = "lightblue")
abline(0, 1, col = "red", lwd = 2)
```




```
# Plot for Test Data
plot(test_data$Log_Apps, test_pred,
     main = "Observed vs Predicted Values (Test Data)",
     xlab = "Observed Log(Apps)",
     ylab = "Predicted Log(Apps)",
     pch = 16, col = "lightgreen")
abline(0, 1, col = "red", lwd = 2)
```



What do you think about the prediction performance of your model? The model performs reasonably well for both training and test data, with predicted values generally aligning with observed values along the red 45-degree line. However, there is some scatter, especially for lower and higher values of `Log(Apps)`, indicating that the model may have trouble accurately predicting applications for schools with extreme numbers of applications (either very few or very many).This could be due to outliers or non-linear relationships that the linear model doesn't capture well.

The test data shows slightly more scatter than the training data, suggesting some potential overfitting. This means the model might be too specifically tuned to the training data and not generalizing as well to unseen data.

d) Compute RMSE for both Training and Test Data

To compute the **Root Mean Square Error (RMSE)** separately for the training and test data, you can use the following formula:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

```
# Compute RMSE for training data
train_rmse <- sqrt(mean((train_data$Log_Apps - train_pred)^2))
cat("Training RMSE:", train_rmse, "\n")
```

```
## Training RMSE: 0.5476839
```

```r
# Compute RMSE for test data
test_rmse <- sqrt(mean((test_data$Log_Apps - test_pred)^2))
cat("Test RMSE:", test_rmse)
```

```
## Test RMSE: 0.6040236
```

Compare the values. What do you conclude? The Training RMSE is 0.54, and the Test RMSE is 0.60. The test RMSE being slightly higher than the training RMSE indicates that the model performs slightly better on the data it was trained on but still generalizes reasonably well to unseen data. The increase in RMSE for the test data suggests a small degree of overfitting: the model may be fitting the training data a bit too closely, but the difference between the two RMSE values isn't drastic.

3. Estimating the Reduced Model

(a) Reduced Model: Check for Significance

```r
# Fit the reduced linear model (excluding insignificant variables from 2(a))
reduced_model <- lm(Log_Apps ~ Private + F.Undergrad + Outstate + Room.Board
                    + S.F.Ratio + perc.alumni + Expend + Grad.Rate,
                    data = train_data)

summary(reduced_model)
```

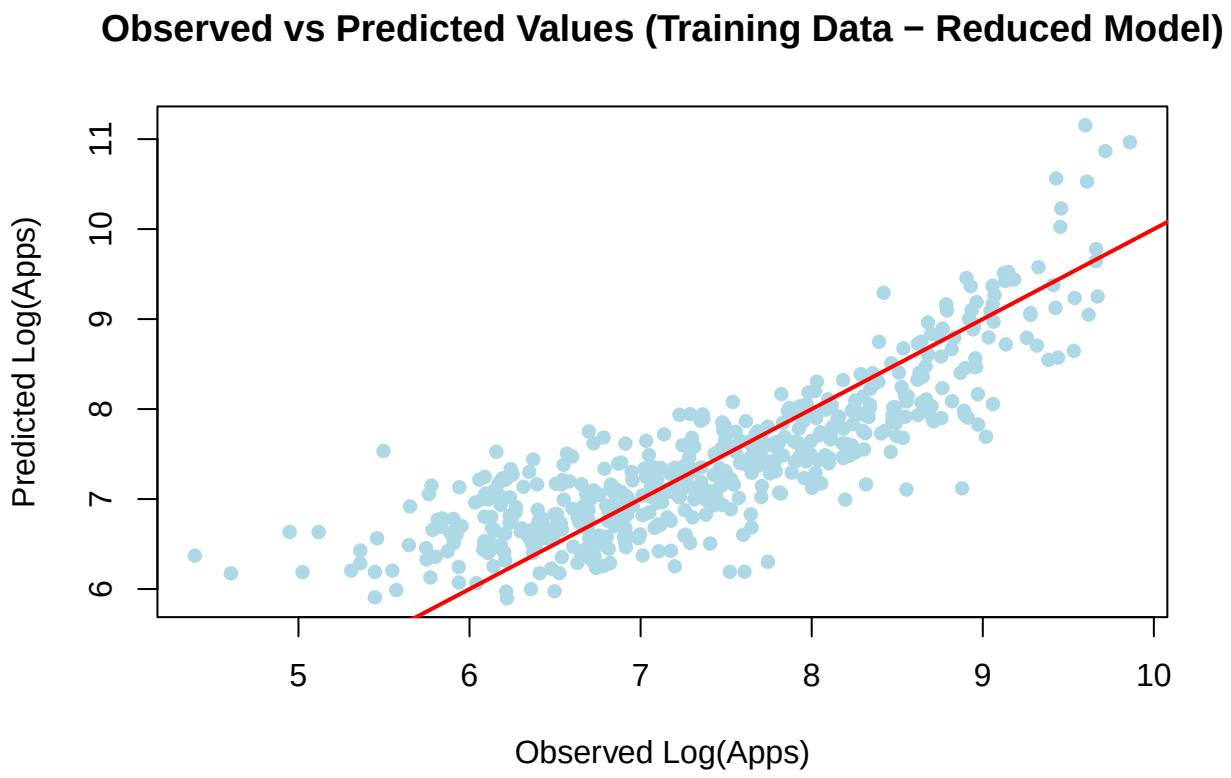
```
##
## Call:
## lm(formula = Log_Apps ~ Private + F.Undergrad + Outstate + Room.Board +
##     S.F.Ratio + perc.alumni + Expend + Grad.Rate, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.03634 -0.30692  0.03467  0.36209  1.75926
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.787e+00  1.998e-01  23.958  < 2e-16 ***
## PrivateYes  -6.854e-01  8.535e-02  -8.030 6.82e-15 ***
## F.Undergrad  1.309e-04  7.059e-06  18.547  < 2e-16 ***
## Outstate     5.795e-05  1.173e-05   4.939 1.07e-06 ***
## Room.Board   1.343e-04  3.072e-05   4.372 1.49e-05 ***
## S.F.Ratio    3.774e-02  8.014e-03   4.709 3.22e-06 ***
## perc.alumni -6.066e-03  2.589e-03  -2.343   0.0195 *
## Expend       2.693e-05  6.865e-06   3.922 9.97e-05 ***
## Grad.Rate    1.241e-02  1.832e-03   6.777 3.41e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5616 on 509 degrees of freedom
## Multiple R-squared:  0.7082, Adjusted R-squared:  0.7036
## F-statistic: 154.4 on 8 and 509 DF,  p-value: < 2.2e-16
```

Are now all input variables significant in the model? Why is this not to be expected in general? Yes, in this reduced model, all input variables are significant, but we can notice perc.alumni has lower significance now. Each variable has a p-value below 0.05, which means they all contribute meaningfully. However, it is not always expected that all variables will remain significant after reducing the model. This is because of, multicollinearity where some variables might be highly correlated with others, making it difficult for the model to separate their effects, or random variation in which there can be randomness in the data that causes some variables to appear significant in one model but not in another

(b) Visualize the Fit and Predictions for the Reduced Model

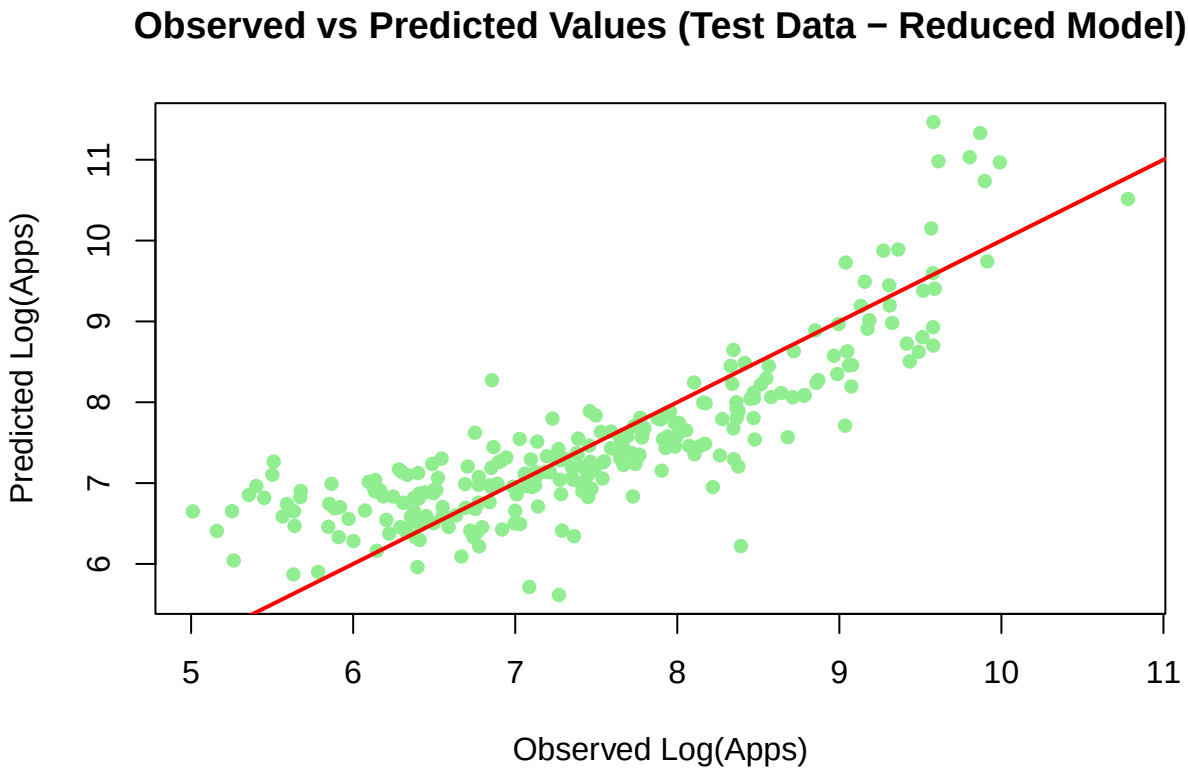
```
train_pred_reduced <- predict(reduced_model, newdata = train_data)
test_pred_reduced <- predict(reduced_model, newdata = test_data)

# Visualizing the fit for training data
plot(train_data$Log_Apps, train_pred_reduced,
     main = "Observed vs Predicted Values (Training Data - Reduced Model)",
     xlab = "Observed Log(Apps)",
     ylab = "Predicted Log(Apps)",
     pch = 16, col = "lightblue")
abline(0, 1, col = "red", lwd = 2)
```




```
# Visualizing the fit for test data
plot(test_data$Log_Apps, test_pred_reduced,
     main = "Observed vs Predicted Values (Test Data - Reduced Model)",
```

```
     xlab = "Observed Log(Apps)",
     ylab = "Predicted Log(Apps)",
     pch = 16, col = "lightgreen")
abline(0, 1, col = "red", lwd = 2)
```



The plots from the reduced model, compared to the ones with all variables, have a similar alignment between observed and predicted values along the red line.

(c) Compute the RMSE for the Reduced Model

```
# Compute RMSE for training data (reduced model)
train_rmse_reduced <- sqrt(mean((train_data$Log_Apps - train_pred_reduced)^2))
cat("Training RMSE (Reduced Model):", train_rmse_reduced, "\n")
```

```
## Training RMSE (Reduced Model): 0.5566698
```

```
# Compute RMSE for test data (reduced model)
test_rmse_reduced <- sqrt(mean((test_data$Log_Apps - test_pred_reduced)^2))
cat("Test RMSE (Reduced Model):", test_rmse_reduced)
```

```
## Test RMSE (Reduced Model): 0.6257909
```

When using a reduced model, we would expect the Training RMSE to increase slightly, as the model now has fewer variables to capture the patterns in the data. The test RMSE might remain the same or decrease

because removing insignificant variables can help the model generalize better to unseen data. In this case, the Test RMSE has increased slightly, which indicates that it has slightly lower predictive accuracy for the test data.

The increase in Test RMSE suggests the reduced model may be slightly underfitting the data compared to the full model. However, since the difference is not substantial, the reduced model still performs reasonably well.

(d) Compare the Full and Reduced Models with ANOVA

```
# Perform ANOVA to compare full and reduced models
anova(res, reduced_model)
```

```
## Analysis of Variance Table
##
## Model 1: Log_Apps ~ (Private + Apps + Accept + Enroll + Top10perc + Top25perc +
##     F.Undergrad + P.Undergrad + Outstate + Room.Board + Books +
##     Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend +
##     Grad.Rate) - Apps - Accept - Enroll
## Model 2: Log_Apps ~ Private + F.Undergrad + Outstate + Room.Board + S.F.Ratio +
##     perc.alumni + Expend + Grad.Rate
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    502 155.38
## 2    509 160.52 -7   -5.1404 2.3725 0.02154 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Degrees of Freedom (Res.Df), the full model has fewer residual degrees of freedom (502) than the reduced model (509). This makes sense because Model 1 includes more predictors, which use up more degrees of freedom. **Residual Sum of Squares (RSS)**, the full model has a lower RSS (155.38) compared to the reduced model (160.52), indicating that the full model fits the data slightly better. **F-statistic (F)**: The F-statistic is 2.3725. This measures the ratio of the improvement in fit by the full model to the additional complexity. A higher F-statistic indicates that the full model provides a significantly better fit. **p-value (Pr(>F))**: The p-value is 0.02154, which is less than 0.05. This indicates that the difference between the full and reduced models is significant.

4. Variable Selection

(a) Forward Selection

Starting from the empty model and adding variables that improve the model.

```
# Perform forward stepwise selection (start from the empty model)
forward_model <- step(lm(Log_Apps ~ 1, data = college_clean),
                      scope = formula(res),
                      direction = "forward")
```

```
## Start:  AIC=111.49
## Log_Apps ~ 1
##
##               Df Sum of Sq    RSS     AIC
```

```
## + F.Undergrad  1    481.70 412.88 -487.29
## + PhD          1    233.99 660.58 -122.12
## + Terminal     1    208.39 686.19  -92.58
## + Private      1    203.44 691.14  -86.99
## + Top25perc    1    141.33 753.25  -20.12
## + P.Undergrad  1    137.84 756.74  -16.53
## + Top10perc    1    118.64 775.94    2.94
## + Expend       1     64.95 829.62   54.92
## + Grad.Rate    1     35.80 858.78   81.75
## + Room.Board   1     35.43 859.14   82.08
## + Personal     1     27.37 867.20   89.34
## + Books        1     23.24 871.34   93.04
## + S.F.Ratio    1     22.14 872.44   94.01
## + Outstate     1     10.25 884.32  104.53
## + perc.alumni  1      3.52 891.05  110.42
## <none>                     894.58  111.49
##
## Step:  AIC=-487.29
## Log_Apps ~ F.Undergrad
##
##               Df Sum of Sq    RSS     AIC
## + PhD          1    76.845 336.03 -645.30
## + Terminal     1    67.736 345.14 -624.52
## + Outstate     1    66.068 346.81 -620.78
## + Top10perc    1    61.937 350.94 -611.58
## + Grad.Rate    1    59.847 353.03 -606.96
## + Top25perc    1    58.750 354.13 -604.55
## + Expend       1    58.543 354.34 -604.10
## + Room.Board   1    55.987 356.89 -598.51
## + perc.alumni  1    10.538 402.34 -505.38
## + Books        1     5.290 407.59 -495.31
## + Personal     1     3.327 409.55 -491.57
## + S.F.Ratio    1     2.229 410.65 -489.49
## <none>                     412.88 -487.29
## + Private      1     0.913 411.97 -487.01
## + P.Undergrad  1     0.904 411.97 -486.99
##
## Step:  AIC=-645.3
## Log_Apps ~ F.Undergrad + PhD
##
##               Df Sum of Sq    RSS     AIC
## + Grad.Rate    1   24.8818 311.15 -703.08
## + Room.Board   1   20.7318 315.30 -692.78
## + Outstate     1   19.4632 316.57 -689.66
## + Expend       1   17.2075 318.83 -684.15
## + Top10perc    1   15.1031 320.93 -679.03
## + Top25perc    1   13.2669 322.77 -674.60
## + Books        1    5.7222 330.31 -656.65
## + Terminal     1    2.7804 333.25 -649.76
## + Private      1    2.0106 334.02 -647.97
## <none>                     336.03 -645.30
## + Personal     1    0.5454 335.49 -644.57
## + S.F.Ratio    1    0.4105 335.62 -644.25
## + P.Undergrad  1    0.3429 335.69 -644.10
```

```
## + perc.alumni  1    0.0396 335.99 -643.39
##
## Step:  AIC=-703.08
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate
##
##               Df Sum of Sq    RSS     AIC
## + Private      1   12.4231 298.73 -732.74
## + Room.Board   1    9.3376 301.81 -724.75
## + Expend       1    8.1692 302.98 -721.75
## + Books        1    5.3741 305.78 -714.62
## + Outstate     1    5.1362 306.02 -714.01
## + perc.alumni  1    4.2588 306.89 -711.79
## + Top10perc    1    4.0030 307.15 -711.14
## + S.F.Ratio    1    3.4667 307.69 -709.78
## + Top25perc    1    3.1689 307.98 -709.03
## + Terminal     1    1.7011 309.45 -705.34
## <none>                     311.15 -703.08
## + P.Undergrad  1    0.5514 310.60 -702.46
## + Personal     1    0.1819 310.97 -701.53
##
## Step:  AIC=-732.74
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private
##
##               Df Sum of Sq    RSS     AIC
## + Outstate     1   22.4824 276.25 -791.53
## + Room.Board   1   18.6299 280.10 -780.77
## + Expend       1   16.1984 282.53 -774.05
## + Top10perc    1    8.2523 290.48 -752.50
## + Books        1    6.5562 292.17 -747.98
## + Top25perc    1    5.9478 292.78 -746.36
## + Terminal     1    1.9716 296.76 -735.88
## + perc.alumni  1    1.5326 297.20 -734.73
## <none>                     298.73 -732.74
## + S.F.Ratio    1    0.4864 298.24 -732.00
## + P.Undergrad  1    0.2698 298.46 -731.44
## + Personal     1    0.0451 298.68 -730.85
##
## Step:  AIC=-791.53
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private + Outstate
##
##               Df Sum of Sq    RSS     AIC
## + Room.Board   1    6.1818 270.06 -807.12
## + perc.alumni  1    5.8002 270.45 -806.02
## + S.F.Ratio    1    5.5201 270.73 -805.22
## + Books        1    5.2250 271.02 -804.37
## + Expend       1    3.5688 272.68 -799.63
## + Top10perc    1    2.5384 273.71 -796.70
## + Top25perc    1    2.2682 273.98 -795.94
## <none>                     276.25 -791.53
## + Personal     1    0.5269 275.72 -791.02
## + P.Undergrad  1    0.3295 275.92 -790.46
## + Terminal     1    0.1566 276.09 -789.97
##
## Step:  AIC=-807.12
```

```
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private + Outstate +
##     Room.Board
##
##               Df Sum of Sq    RSS     AIC
## + S.F.Ratio    1    5.6788 264.39 -821.63
## + perc.alumni  1    3.9169 266.15 -816.47
## + Books        1    3.9146 266.15 -816.46
## + Top10perc    1    3.2061 266.86 -814.40
## + Expend       1    2.8722 267.19 -813.42
## + Top25perc    1    2.8562 267.21 -813.38
## <none>                     270.06 -807.12
## + Personal     1    0.6139 269.45 -806.88
## + P.Undergrad  1    0.0704 270.00 -805.32
## + Terminal     1    0.0131 270.05 -805.15
##
## Step:  AIC=-821.63
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private + Outstate +
##     Room.Board + S.F.Ratio
##
##               Df Sum of Sq    RSS     AIC
## + Expend       1    8.8682 255.52 -846.14
## + Top10perc    1    5.3900 259.00 -835.63
## + Books        1    4.2283 260.16 -832.16
## + Top25perc    1    4.0372 260.35 -831.59
## + perc.alumni  1    3.0295 261.36 -828.58
## + Personal     1    1.0439 263.34 -822.70
## <none>                     264.39 -821.63
## + P.Undergrad  1    0.0681 264.32 -819.83
## + Terminal     1    0.0412 264.35 -819.75
##
## Step:  AIC=-846.14
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private + Outstate +
##     Room.Board + S.F.Ratio + Expend
##
##               Df Sum of Sq    RSS     AIC
## + perc.alumni  1    3.7354 251.78 -855.58
## + Books        1    3.2629 252.25 -854.12
## + Top25perc    1    2.1726 253.34 -850.77
## + Top10perc    1    1.7005 253.82 -849.33
## + Personal     1    0.7281 254.79 -846.36
## <none>                     255.52 -846.14
## + P.Undergrad  1    0.1185 255.40 -844.50
## + Terminal     1    0.0218 255.50 -844.21
##
## Step:  AIC=-855.58
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private + Outstate +
##     Room.Board + S.F.Ratio + Expend + perc.alumni
##
##               Df Sum of Sq    RSS     AIC
## + Top25perc    1    3.2170 248.57 -863.57
## + Books        1    3.0101 248.77 -862.93
## + Top10perc    1    2.5738 249.21 -861.57
## <none>                     251.78 -855.58
## + Personal     1    0.4234 251.36 -854.89
```

```
## + Terminal     1    0.0996 251.68 -853.89
## + P.Undergrad  1    0.0623 251.72 -853.77
##
## Step:  AIC=-863.57
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private + Outstate +
##     Room.Board + S.F.Ratio + Expend + perc.alumni + Top25perc
##
##               Df Sum of Sq    RSS     AIC
## + Books        1   2.42921 246.14 -869.20
## <none>                     248.57 -863.57
## + Personal     1   0.40084 248.16 -862.83
## + P.Undergrad  1   0.18896 248.38 -862.16
## + Top10perc    1   0.08271 248.48 -861.83
## + Terminal     1   0.04057 248.53 -861.70
##
## Step:  AIC=-869.2
## Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private + Outstate +
##     Room.Board + S.F.Ratio + Expend + perc.alumni + Top25perc +
##     Books
##
##               Df Sum of Sq    RSS     AIC
## <none>                     246.14 -869.20
## + P.Undergrad  1  0.158606 245.98 -867.70
## + Personal     1  0.143347 245.99 -867.66
## + Top10perc    1  0.058804 246.08 -867.39
## + Terminal     1  0.000747 246.14 -867.21
```

```
summary(forward_model)
```

```
##
## Call:
## lm(formula = Log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private +
##     Outstate + Room.Board + S.F.Ratio + Expend + perc.alumni +
##     Top25perc + Books, data = college_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.26973 -0.32758  0.02424  0.37942  1.84521
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.370e+00  1.931e-01  22.632  < 2e-16 ***
## F.Undergrad  1.129e-04  5.832e-06  19.366  < 2e-16 ***
## PhD          6.787e-03  1.710e-03   3.969 7.88e-05 ***
## Grad.Rate    9.985e-03  1.560e-03   6.400 2.71e-10 ***
## PrivateYes  -6.254e-01  7.446e-02  -8.399  < 2e-16 ***
## Outstate     4.830e-05  1.013e-05   4.766 2.24e-06 ***
## Room.Board   7.913e-05  2.564e-05   3.086 0.002099 **
## S.F.Ratio    4.124e-02  7.062e-03   5.840 7.71e-09 ***
## Expend       2.854e-05  6.216e-06   4.592 5.14e-06 ***
## perc.alumni -8.086e-03  2.184e-03  -3.703 0.000229 ***
## Top25perc    4.174e-03  1.458e-03   2.862 0.004320 **
## Books        3.491e-04  1.270e-04   2.748 0.006142 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5672 on 765 degrees of freedom
## Multiple R-squared:  0.7249, Adjusted R-squared:  0.7209
## F-statistic: 183.2 on 11 and 765 DF,  p-value: < 2.2e-16
```

(b) Backward Selection

Starting from the full model and removing variables that have the least impact on the model.

```
# Perform backward stepwise selection (start from the full model)
backward_model <- step(res, direction = "backward")
```

```
## Start:  AIC=-591.73
## Log_Apps ~ (Private + Apps + Accept + Enroll + Top10perc + Top25perc +
##     F.Undergrad + P.Undergrad + Outstate + Room.Board + Books +
##     Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend +
##     Grad.Rate) - Apps - Accept - Enroll
##
##               Df Sum of Sq    RSS     AIC
## - Personal     1     0.000 155.38 -593.73
## - Top25perc    1     0.009 155.39 -593.70
## - Terminal     1     0.030 155.41 -593.63
## - P.Undergrad  1     0.113 155.49 -593.36
## - Top10perc    1     0.395 155.77 -592.42
## <none>                     155.38 -591.73
## - Books        1     0.726 156.10 -591.32
## - PhD          1     0.777 156.16 -591.15
## - Expend       1     1.841 157.22 -587.63
## - perc.alumni  1     2.499 157.88 -585.47
## - Room.Board   1     4.404 159.78 -579.25
## - Outstate     1     4.741 160.12 -578.16
## - S.F.Ratio    1     6.339 161.72 -573.02
## - Grad.Rate    1    11.373 166.75 -557.14
## - Private      1    14.242 169.62 -548.30
## - F.Undergrad  1    64.371 219.75 -414.18
##
## Step:  AIC=-593.73
## Log_Apps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + PhD + Terminal + S.F.Ratio +
##     perc.alumni + Expend + Grad.Rate
##
##               Df Sum of Sq    RSS     AIC
## - Top25perc    1     0.009 155.39 -595.70
## - Terminal     1     0.030 155.41 -595.63
## - P.Undergrad  1     0.113 155.49 -595.35
## - Top10perc    1     0.395 155.77 -594.42
## <none>                     155.38 -593.73
## - Books        1     0.739 156.12 -593.27
## - PhD          1     0.777 156.16 -593.15
## - Expend       1     1.845 157.22 -589.62
## - perc.alumni  1     2.509 157.89 -587.43
## - Room.Board   1     4.447 159.82 -581.11
```

```
## - Outstate     1     4.766 160.14 -580.08
## - S.F.Ratio    1     6.377 161.75 -574.89
## - Grad.Rate    1    11.621 167.00 -558.37
## - Private      1    14.271 169.65 -550.21
## - F.Undergrad  1    66.195 221.57 -411.90
##
## Step:  AIC=-595.7
## Log_Apps ~ Private + Top10perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + PhD + Terminal + S.F.Ratio +
##     perc.alumni + Expend + Grad.Rate
##
##               Df Sum of Sq    RSS     AIC
## - Terminal     1     0.026 155.41 -597.61
## - P.Undergrad  1     0.115 155.50 -597.32
## <none>                     155.39 -595.70
## - Books        1     0.736 156.12 -595.25
## - PhD          1     0.776 156.16 -595.12
## - Top10perc    1     0.798 156.19 -595.05
## - Expend       1     1.951 157.34 -591.24
## - perc.alumni  1     2.517 157.91 -589.37
## - Room.Board   1     4.480 159.87 -582.98
## - Outstate     1     4.760 160.15 -582.07
## - S.F.Ratio    1     6.391 161.78 -576.82
## - Grad.Rate    1    11.667 167.05 -560.20
## - Private      1    14.288 169.68 -552.13
## - F.Undergrad  1    66.694 222.08 -412.71
##
## Step:  AIC=-597.61
## Log_Apps ~ Private + Top10perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + PhD + S.F.Ratio + perc.alumni +
##     Expend + Grad.Rate
##
##               Df Sum of Sq    RSS     AIC
## - P.Undergrad  1     0.116 155.53 -599.23
## <none>                     155.41 -597.61
## - Top10perc    1     0.789 156.20 -596.99
## - Books        1     0.800 156.21 -596.95
## - Expend       1     1.952 157.37 -593.15
## - PhD          1     2.218 157.63 -592.27
## - perc.alumni  1     2.491 157.91 -591.37
## - Room.Board   1     4.633 160.05 -584.39
## - Outstate     1     4.886 160.30 -583.58
## - S.F.Ratio    1     6.378 161.79 -578.78
## - Grad.Rate    1    11.644 167.06 -562.19
## - Private      1    14.544 169.96 -553.27
## - F.Undergrad  1    67.026 222.44 -413.87
##
## Step:  AIC=-599.23
## Log_Apps ~ Private + Top10perc + F.Undergrad + Outstate + Room.Board +
##     Books + PhD + S.F.Ratio + perc.alumni + Expend + Grad.Rate
##
##               Df Sum of Sq    RSS     AIC
## <none>                     155.53 -599.23
## - Top10perc    1     0.727 156.26 -598.81
```

```
## - Books         1     0.790 156.32 -598.60
## - Expend        1     1.973 157.50 -594.69
## - PhD           1     2.353 157.88 -593.45
## - perc.alumni   1     2.620 158.15 -592.57
## - Outstate      1     4.844 160.37 -585.34
## - Room.Board    1     4.945 160.47 -585.01
## - S.F.Ratio     1     6.400 161.93 -580.34
## - Grad.Rate     1    11.606 167.13 -563.95
## - Private       1    14.697 170.23 -554.45
## - F.Undergrad   1    88.384 243.91 -368.14
```

```
summary(backward_model)
```

```
##
## Call:
## lm(formula = Log_Apps ~ Private + Top10perc + F.Undergrad + Outstate +
##     Room.Board + Books + PhD + S.F.Ratio + perc.alumni + Expend +
##     Grad.Rate, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.15496 -0.31245  0.04207  0.35535  1.72620
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.500e+00  2.286e-01  19.684  < 2e-16 ***
## PrivateYes  -6.125e-01  8.858e-02  -6.915 1.42e-11 ***
## Top10perc    3.406e-03  2.215e-03   1.538  0.12476
## F.Undergrad  1.231e-04  7.258e-06  16.957  < 2e-16 ***
## Outstate     4.754e-05  1.198e-05   3.970 8.23e-05 ***
## Room.Board   1.246e-04  3.105e-05   4.011 6.96e-05 ***
## Books        2.277e-04  1.420e-04   1.604  0.10942
## PhD          5.434e-03  1.964e-03   2.767  0.00587 **
## S.F.Ratio    3.622e-02  7.939e-03   4.563 6.34e-06 ***
## perc.alumni -7.610e-03  2.606e-03  -2.920  0.00366 **
## Expend       1.870e-05  7.380e-06   2.534  0.01158 *
## Grad.Rate    1.148e-02  1.869e-03   6.145 1.62e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5544 on 506 degrees of freedom
## Multiple R-squared:  0.7173, Adjusted R-squared:  0.7111
## F-statistic: 116.7 on 11 and 506 DF,  p-value: < 2.2e-16
```

(c) Compute RMSE for Both Models

```
# Predict values using forward model
forward_pred <- predict(forward_model, newdata = college_clean)

# Predict values using backward model
backward_pred <- predict(backward_model, newdata = college_clean)
```

```r
# Compute RMSE for forward model
forward_rmse <- sqrt(mean((college_clean$Log_Apps - forward_pred)^2))
cat("Forward Selection RMSE:", forward_rmse, "\n")
```

```
## Forward Selection RMSE: 0.5628301
```

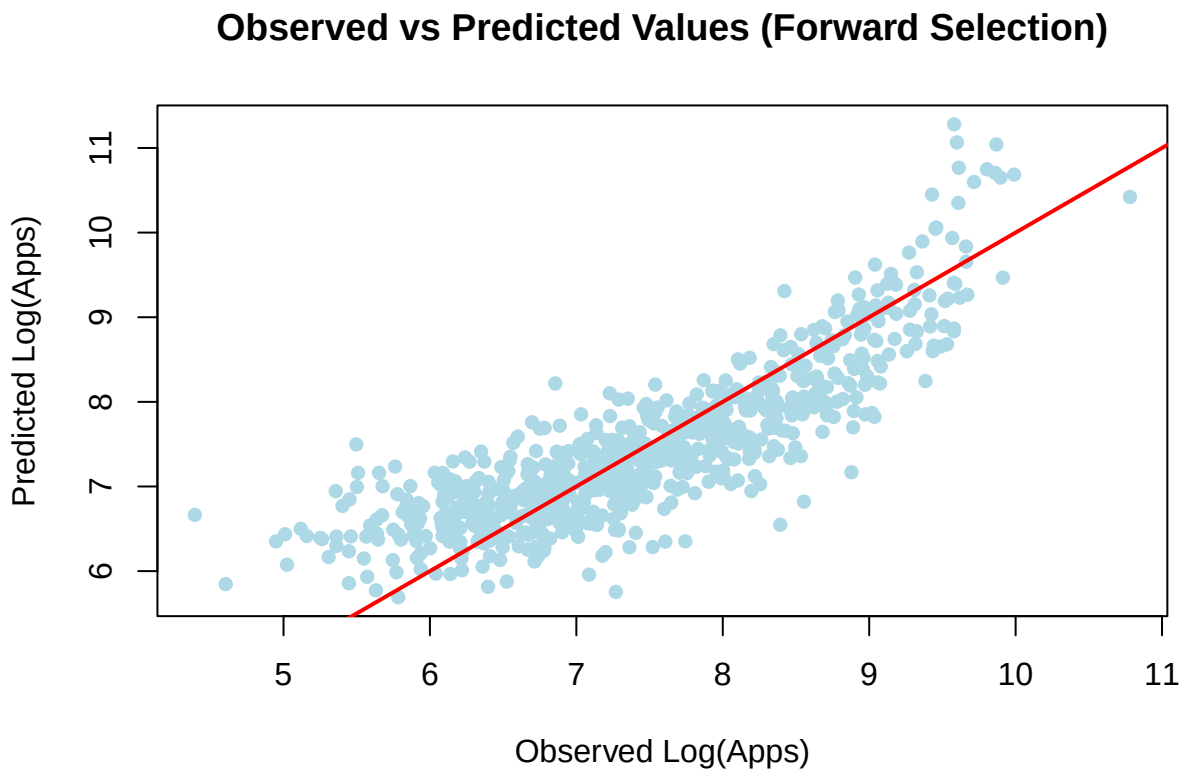
```r
# Compute RMSE for backward model
backward_rmse <- sqrt(mean((college_clean$Log_Apps - backward_pred)^2))
cat("Backward Selection RMSE:", backward_rmse, "\n")
```

```
## Backward Selection RMSE: 0.5671309
```

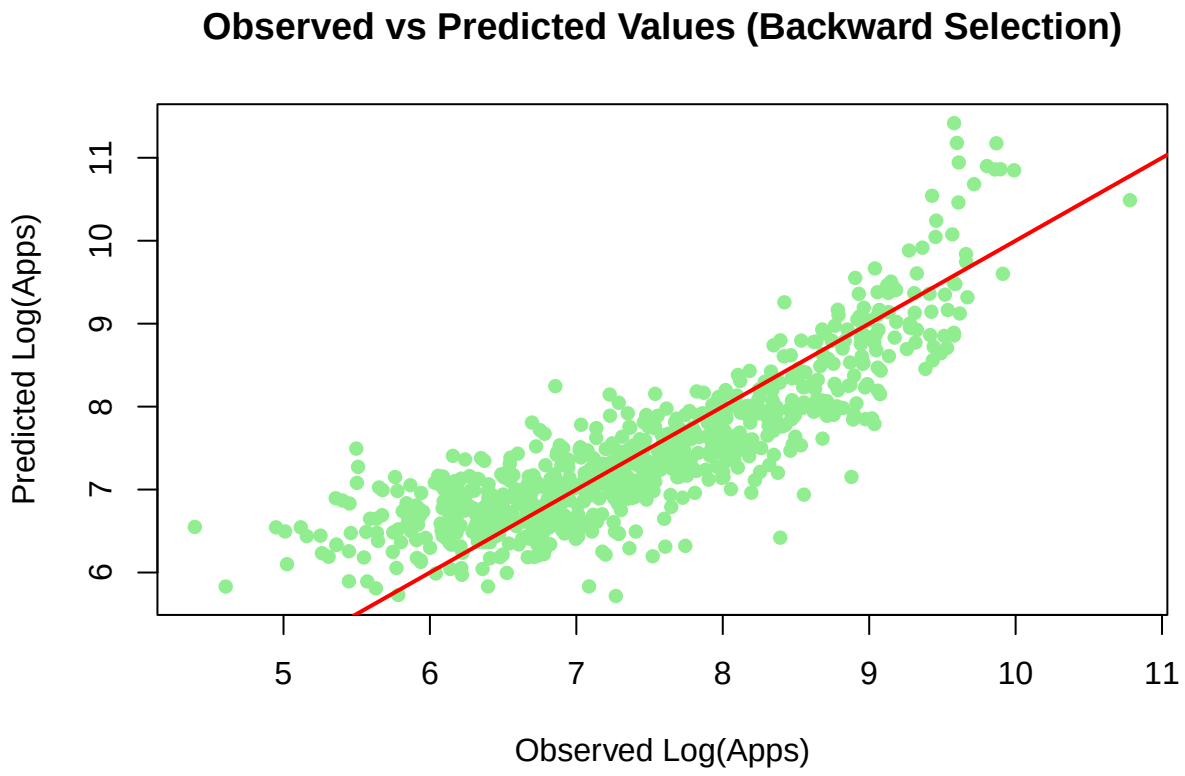
The **Forward Selection RMSE** (0.5628) is slightly lower than the **Backward Selection RMSE** (0.5671), indicating that the forward-selected model performs better in predicting the number of applications.

(d) Visualize the Fit and Predictions for Both Models

```r
# Plot observed vs predicted for forward model
plot(college_clean$Log_Apps, forward_pred,
     main = "Observed vs Predicted Values (Forward Selection)",
     xlab = "Observed Log(Apps)",
     ylab = "Predicted Log(Apps)",
     pch = 16, col = "lightblue")
abline(0, 1, col = "red", lwd = 2)
```



```r
# Plot observed vs predicted for backward model
plot(college_clean$Log_Apps, backward_pred,
     main = "Observed vs Predicted Values (Backward Selection)",
     xlab = "Observed Log(Apps)",
     ylab = "Predicted Log(Apps)",
     pch = 16, col = "lightgreen")
abline(0, 1, col = "red", lwd = 2)
```



Both the **Backward Selection** and **Forward Selection** models produce similar results, but the Forward Selection model fits the data slightly better. This is shown by its tighter alignment of predicted values with the observed values along the red line.