College {ISLR2 Package}

MATH-50028-001: STATISTICAL LEARNING

May 10, 2024

Introduction [5 points]

The dataset "College" is chosen from ISLR2 package. This college dataset, includes information on a range of US institutions, is taken from the US News & World Report publication from 1995. It appears to have been put together to provide a summary of many characteristics that would be of interest to academic scholars, policy officials, and prospective students who are involved in the planning and study of higher education.

There are 18 variables associated with each of the 777 observations in the dataset, which are likely to represent distinct institutions or universities. These variables provide a range of information about the colleges, such as the number of applications received, enrollment figures, percentages of high-achieving students, undergraduate populations (part-time and full-time), costs (tuition, books, room and board, personal expenses), faculty qualifications, student-to-faculty ratios, percentages of alumni donations, spending per student, etc.

The dataset appears to focus on US colleges and universities as they are included in the US News and World Report from 1995, which may provide a thorough picture of these establishments. However, there is no specific description of the sample technique, which may lead to concerns over selection bias. It's not apparent, for example, if all institution types such as community colleges, technical schools, and recently created universities are included or if some institution types such as large research universities or liberal arts colleges are over-represented. Due to this ambiguity, analyses may be biased toward particular types of universities and may not give a complete picture of all higher education institutions in the United States at that particular moment.

Research Question:

Which institutional characteristics significantly influence the enrollment decisions of students in college?

Loading and preparing the data

```
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.3.3
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data(College)
head(College)
```

```
##                              Private Apps Accept Enroll Top10perc Top25perc
## Abilene Christian University     Yes 1660   1232    721        23        52
## Adelphi University               Yes 2186   1924    512        16        29
## Adrian College                   Yes 1428   1097    336        22        50
## Agnes Scott College              Yes  417    349    137        60        89
## Alaska Pacific University        Yes  193    146     55        16        44
## Albertson College                Yes  587    479    158        38        62
##                              F.Undergrad P.Undergrad Outstate Room.Board Books
## Abilene Christian University        2885         537     7440       3300   450
## Adelphi University                  2683        1227    12280       6450   750
## Adrian College                      1036          99    11250       3750   400
## Agnes Scott College                  510          63    12960       5450   450
## Alaska Pacific University            249         869     7560       4120   800
## Albertson College                    678          41    13500       3335   500
##                              Personal PhD Terminal S.F.Ratio perc.alumni Expend
## Abilene Christian University     2200  70       78      18.1          12   7041
## Adelphi University               1500  29       30      12.2          16  10527
## Adrian College                   1165  53       66      12.9          30   8735
## Agnes Scott College               875  92       97       7.7          37  19016
## Alaska Pacific University        1500  76       72      11.9           2  10922
## Albertson College                 675  67       73       9.4          11   9727
##                              Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
## Agnes Scott College                 59
```

```
## Alaska Pacific University           15
## Albertson College                   55
```

```
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 ...
```

```
# Converting 'Private' column to numeric i.e 1 for Yes, 0 for No
College$Private <- as.numeric(College$Private == "Yes")

# Checking for missing values
College <- na.omit(College)

# Printing the summary
summary(College)
```

```
##     Private            Apps           Accept          Enroll
##  Min.   :0.0000   Min.   :   81   Min.   :   72   Min.   :  35
##  1st Qu.:0.0000   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242
##  Median :1.0000   Median : 1558   Median : 1110   Median : 434
##  Mean   :0.7272   Mean   : 3002   Mean   : 2019   Mean   : 780
##  3rd Qu.:1.0000   3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902
##  Max.   :1.0000   Max.   :48094   Max.   :26330   Max.   :6392
##    Top10perc       Top25perc      F.Undergrad     P.Undergrad
##  Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0
##  1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0
```

```
##  Median :23.00   Median : 54.0   Median : 1707   Median :  353.0
##  Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3
##  3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0
##  Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0
##     Outstate       Room.Board       Books           Personal
##  Min.   : 2340   Min.   :1780   Min.   :  96.0   Min.   : 250
##  1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850
##  Median : 9990   Median :4200   Median : 500.0   Median :1200
##  Mean   :10441   Mean   :4358   Mean   : 549.4   Mean   :1341
##  3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700
##  Max.   :21700   Max.   :8124   Max.   :2340.0   Max.   :6800
##       PhD            Terminal       S.F.Ratio      perc.alumni
##  Min.   :  8.00   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00
##  1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00
##  Median : 75.00   Median : 82.0   Median :13.60   Median :21.00
##  Mean   : 72.66   Mean   : 79.7   Mean   :14.09   Mean   :22.74
##  3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00
##  Max.   :103.00   Max.   :100.0   Max.   :39.80   Max.   :64.00
##      Expend        Grad.Rate
##  Min.   : 3186   Min.   : 10.00
##  1st Qu.: 6751   1st Qu.: 53.00
##  Median : 8377   Median : 65.00
##  Mean   : 9660   Mean   : 65.46
##  3rd Qu.:10830   3rd Qu.: 78.00
##  Max.   :56233   Max.   :118.00
```

Split the Data into Training and Test Sets

It is essential to divide the data into training and testing sets for the following reasons: model tuning, over fitting avoidance, and model evaluation. The amount of indices to sample, "0.7 * nrow(College)". 30% of the data is test samples, while the remaining 70% is train samples.

```r
set.seed(123)

indices <- sample(1:nrow(College), 0.7 * nrow(College))
train <- College[indices, ]
test <- College[-indices, ]

# Checking the size of each set
cat("Training set rows:", nrow(train))
```

```
## Training set rows: 543
```

```r
cat("Test set rows:", nrow(test))
```

```
## Test set rows: 234
```

```r
# Printing the Summary of the training data
summary(train)
```

```
##     Private            Apps             Accept            Enroll
##  Min.   :0.0000   Min.   :   81.0   Min.   :   72.0   Min.   :  51
##  1st Qu.:0.0000   1st Qu.:  792.5   1st Qu.:  619.5   1st Qu.: 255
##  Median :1.0000   Median : 1584.0   Median : 1199.0   Median : 438
##  Mean   :0.7182   Mean   : 2914.4   Mean   : 1941.9   Mean   : 766
##  3rd Qu.:1.0000   3rd Qu.: 3739.5   3rd Qu.: 2459.5   3rd Qu.: 912
##  Max.   :1.0000   Max.   :21804.0   Max.   :18744.0   Max.   :6392
##    Top10perc       Top25perc       F.Undergrad     P.Undergrad
##  Min.   : 2.00   Min.   :  9.00   Min.   :  139   Min.   :    1.0
##  1st Qu.:15.00   1st Qu.: 40.00   1st Qu.: 1048   1st Qu.:   94.5
##  Median :23.00   Median : 53.00   Median : 1707   Median :  382.0
##  Mean   :27.42   Mean   : 55.35   Mean   : 3637   Mean   :  882.5
##  3rd Qu.:35.00   3rd Qu.: 69.00   3rd Qu.: 4228   3rd Qu.:  966.5
##  Max.   :96.00   Max.   :100.00   Max.   :31643   Max.   :21836.0
##     Outstate       Room.Board       Books           Personal
##  Min.   : 2580   Min.   :1780   Min.   : 110.0   Min.   : 250
##  1st Qu.: 7164   1st Qu.:3588   1st Qu.: 475.5   1st Qu.: 850
##  Median : 9900   Median :4200   Median : 525.0   Median :1200
##  Mean   :10397   Mean   :4353   Mean   : 553.6   Mean   :1352
##  3rd Qu.:12938   3rd Qu.:5000   3rd Qu.: 600.0   3rd Qu.:1700
##  Max.   :21700   Max.   :8124   Max.   :2000.0   Max.   :6800
##       PhD            Terminal        S.F.Ratio      perc.alumni
##  Min.   : 14.00   Min.   : 24.00   Min.   : 2.50   Min.   : 0.00
##  1st Qu.: 62.00   1st Qu.: 71.00   1st Qu.:11.50   1st Qu.:13.00
##  Median : 75.00   Median : 82.00   Median :13.80   Median :21.00
##  Mean   : 72.89   Mean   : 79.82   Mean   :14.26   Mean   :22.75
##  3rd Qu.: 86.00   3rd Qu.: 92.00   3rd Qu.:16.70   3rd Qu.:32.00
##  Max.   :103.00   Max.   :100.00   Max.   :39.80   Max.   :60.00
##     Expend        Grad.Rate
##  Min.   : 3186   Min.   : 10.00
##  1st Qu.: 6634   1st Qu.: 54.00
##  Median : 8367   Median : 65.00
##  Mean   : 9627   Mean   : 65.55
##  3rd Qu.:10796   3rd Qu.: 78.00
##  Max.   :56233   Max.   :118.00
```

Statistical learning strategies and methods [15 points]

Backward selection to reduce the number of redundant or irrelevant features.

The below code illustrates statistical learning methods and feature engineering strategies for modeling using a college enrollment dataset. Firstly, the category 'Private' variable is converted to a numeric format in order to prepare the dataset. After determining the skewness of continuous data, we apply modification to highly skewed variables like "Apps" and "Accept." It improves the interpretability and efficiency of the model by normalizing "Outstate" tuition fees and "Room.Board" prices to have a mean of zero and a standard deviation of one. Furthermore, 'AcceptanceRate' is a new feature that may be able to capture more subtle influences on enrollment.

An iterative process of backward elimination that begins with a complete model with all variables and newly designed features, with the goal of identifying the most important predictors by eliminating the least significant ones. In order to improve model performance and reduce complexity, the final model is adjusted to contain those variables that have a meaningful influence on the goal variable, 'Enroll'. This method combines sophisticated feature engineering approaches to improve model accuracy and interpretability with statistical learning concepts for model creation.

```
# Load the MASS package for backward selection
library(MASS)  # For backward selection using stepAIC
```

```
## Warning: package 'MASS' was built under R version 4.3.2
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
library(e1071)  # For skewness function
```

```
## Warning: package 'e1071' was built under R version 4.3.2
```

```
# Applying log transformation to continuous variables that are skewed
# Checking for skewness in continuous variables, assuming continuous variables
# like 'Apps', 'Accept', etc.

# Converting 'Private' variable to numeric
```

```r
train$Private <- as.numeric(train$Private == "Yes")
test$Private <- as.numeric(test$Private == "Yes")

continuous_vars <- sapply(train, function(x) is.numeric(x) && length(unique(x)) > 2)

# Calculating skewness for these continuous variables
skewness_values <- sapply(train[, continuous_vars, drop = FALSE], skewness)

# Printing skewness values
print(skewness_values)
```

```
##        Apps      Accept      Enroll   Top10perc   Top25perc F.Undergrad
##   2.2852781   2.8409804   2.7606163   1.5031175   0.3111121   2.4899021
## P.Undergrad    Outstate  Room.Board       Books    Personal         PhD
##   6.0580970   0.4830387   0.4809163   2.6851916   1.8161836  -0.6693195
##    Terminal   S.F.Ratio perc.alumni      Expend   Grad.Rate
##  -0.8462937   0.6922152   0.5193089   3.5310641  -0.1361789
```

```r
# Applying log transformation where it makes sense (e.g., 'Apps', 'Accept' if they are skewed)
train$LogApps <- log(train$Apps + 1)
train$LogAccept <- log(train$Accept + 1)
test$LogApps <- log(test$Apps + 1)
test$LogAccept <- log(test$Accept + 1)

# Normalizing 'Outstate' tuition fees and 'Room.Board' to
#have mean of 0 and standard deviation of 1
train$NormOutstate <- scale(train$Outstate)
train$NormRoomBoard <- scale(train$Room.Board)
test$NormOutstate <- scale(test$Outstate)
test$NormRoomBoard <- scale(test$Room.Board)

# Creating a new feature that captures the acceptance rate
train$AcceptanceRate <- train$Accept / train$Apps
test$AcceptanceRate <- test$Accept / test$Apps

# Modifying full model to include mandatory terms and allow other terms to compete
mandatory_terms <- lm(Enroll ~ Apps + Accept + Outstate + Room.Board + Grad.Rate,
                      data = train)
full_model <- update(mandatory_terms, . ~ . + F.Undergrad + P.Undergrad + PhD +
                       perc.alumni + LogApps + LogAccept )

# Performing backward stepwise selection starting from the full model
backward_step <- stepAIC(full_model, scope = list(lower = formula(mandatory_terms),
                 upper = formula(full_model)), direction = "backward", trace = FALSE)
```

```r
selected_features <- names(coef(backward_step))
selected_features <- selected_features[-1]  # Removing intercept
cat("Final selected features after backward selection:\n")
```

```
## Final selected features after backward selection:
```

```r
print(selected_features)
```

```
## [1] "Apps"        "Accept"      "Outstate"    "Room.Board"  "Grad.Rate"
## [6] "F.Undergrad" "P.Undergrad" "perc.alumni"
```

The skewness values for different characteristics from a college dataset are included in the output, along with information on which variables are kept after modeling using forward elimination. The values for Accept and P. Undergrad exhibit strong positive skewness, suggesting right-skewed distributions, whereas the other values indicate the degree of asymmetry in the distribution of each characteristic around its mean. A negative skewness, on the other hand, indicates tails expanding towards lower values in variables like PhD and Terminal. Significant predictors like Apps, Accept, Outstate, Room.Board, Grad.Rate, F.Undergrad, P.Undergrad, and perc.alumni were chosen for the final model using the stepAIC function to execute the backward elimination strategy. The explanatory power and statistical prediction efficiency of the model are increased since these factors are thought to have a significant effect on college enrollment forecasts.

Plotting the best model obtained according to Cp, BIC, and adjusted R2

Using Cp, BIC, and modified R2 metrics, the algorithm builds models with progressively more combinations of chosen characteristics and assesses the statistical merits of each model. Iteratively adding one feature at a time, it evaluates each model's performance before determining and presenting the top model based on each criterion. Plots are also included to provide a visual comparison of how these metrics alter as more variables are added.

```r
# Creating models with different combinations of selected features
models <- lapply(1:length(selected_features), function(i) {
  formula <- as.formula(paste("Enroll ~", paste(selected_features[1:i], collapse = " + ")))
  lm_model <- lm(formula, data = train)
  return(lm_model)
})

# Calculating Cp, BIC, and adjusted R^2 for each model
metrics <- sapply(models, function(model) {
  AIC <- AIC(model)
  Cp <- AIC + 2 * length(model$coef) * (nrow(train) / (nrow(train) - length(model$coef) - 1))
  BIC <- AIC + log(nrow(train)) * length(model$coef)
  adj_R2 <- summary(model)$adj.r.squared
```

```
  return(c(Cp, BIC, adj_R2))
})

# Finding the model with the minimum Cp, BIC, and maximum adjusted R^2
best_model_idx <- list(Cp = which.min(metrics[1, ]),
                       BIC = which.min(metrics[2, ]),
                       adj_R2 = which.max(metrics[3, ]))

# Printing the best model according to Cp, BIC, and adjusted R^2
cat("Best model according to Cp:\n")
```

```
## Best model according to Cp:
```

```
print(models[[best_model_idx$Cp]])
```

```
##
## Call:
## lm(formula = formula, data = train)
##
## Coefficients:
## (Intercept)         Apps       Accept     Outstate   Room.Board    Grad.Rate
##   147.57586      0.01505      0.12066     -0.00254     -0.03539     -0.30799
## F.Undergrad  P.Undergrad  perc.alumni
##     0.12990     -0.01653      3.64875
```

```
cat("\nBest model according to BIC:\n")
```

```
##
## Best model according to BIC:
```

```
print(models[[best_model_idx$BIC]])
```

```
##
## Call:
## lm(formula = formula, data = train)
##
## Coefficients:
## (Intercept)         Apps       Accept     Outstate   Room.Board    Grad.Rate
##   147.57586      0.01505      0.12066     -0.00254     -0.03539     -0.30799
## F.Undergrad  P.Undergrad  perc.alumni
##     0.12990     -0.01653      3.64875
```

```r
cat("\nBest model according to adjusted R^2:\n")
```

```
##
## Best model according to adjusted R^2:
```

```r
# Create models with different combinations of selected features
print(models[[best_model_idx$adj_R2]])
```

```
##
## Call:
## lm(formula = formula, data = train)
##
## Coefficients:
## (Intercept)         Apps       Accept     Outstate   Room.Board    Grad.Rate
##   147.57586      0.01505      0.12066     -0.00254     -0.03539     -0.30799
## F.Undergrad  P.Undergrad  perc.alumni
##     0.12990     -0.01653      3.64875
```

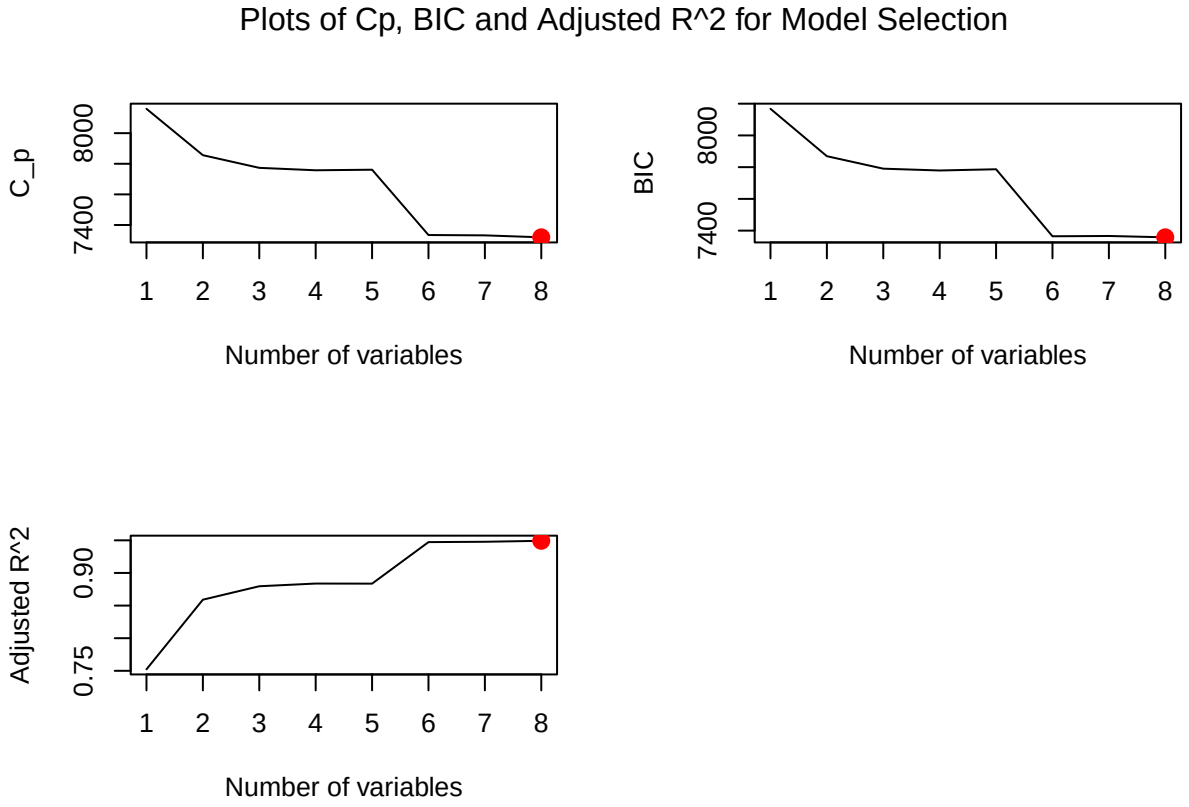
```r
# Plotting
par(mfrow = c(2, 2))

# Plotting Cp values
plot(metrics[1, ], xlab = "Number of variables", ylab = "C_p", type = "l")
points(which.min(metrics[1, ]), min(metrics[1, ]), col = "red", cex = 2, pch = 20)

# Plotting BIC values
plot(metrics[2, ], xlab = "Number of variables", ylab = "BIC", type = "l")
points(which.min(metrics[2, ]), min(metrics[2, ]), col = "red", cex = 2, pch = 20)

# Plotting Adjusted R^2 values
plot(metrics[3, ], xlab = "Number of variables", ylab = "Adjusted R^2", type = "l")
points(which.max(metrics[3, ]), max(metrics[3, ]), col = "red", cex = 2, pch = 20)

mtext("Plots of Cp, BIC and Adjusted R^2 for Model Selection", side = 3, line = -2,
      outer = TRUE)
```



The best model, determined by three distinct criteria (Cp, BIC, and modified R^2), is displayed in detail in the output. The coefficients for each predictor variable are shown next to the intercept in a linear regression formula that has been fitted to the training set of each model. These coefficients show how much each predictor is expected to influence the response variable, "Enroll." The same coefficient values are shown by the three best models, while having different labels, indicating that the variables' effects are consistent across the various model selection criteria.

From the visual representation, we observe that there three metrics Cp, BIC, and Adjusted R2 for Model Selection- 1) Cp is a metric used to address over fitting. It strikes a compromise between the amount of variables in the model and how well it fits the data. A better model is indicated by a lower Cp value. 2) One measure used to deal with over fitting is BIC. A better model is indicated by lower BIC values. 3) The percentage of the dependent variable's variation that the independent variables account for is shown by the R2 statistic, which is employed in regression analysis. Since adding additional variables tends to enhance R2, models with too many variables are penalized by adjusted R^2.

Exploratory Data Analysis (EDA) using training set

The distribution, outliers, and correlations between the important numerical variables in the training dataset are shown using a set of exploratory data analysis (EDA). It uses boxplots to find outliers, scatter plots to investigate associations, histograms to show the distributions, and scatter plots with regression lines to investigate the link between enrollment and other factors in more detail.

```r
# Loading necessary library for visualization
library(ggplot2)
library(corrplot)
```

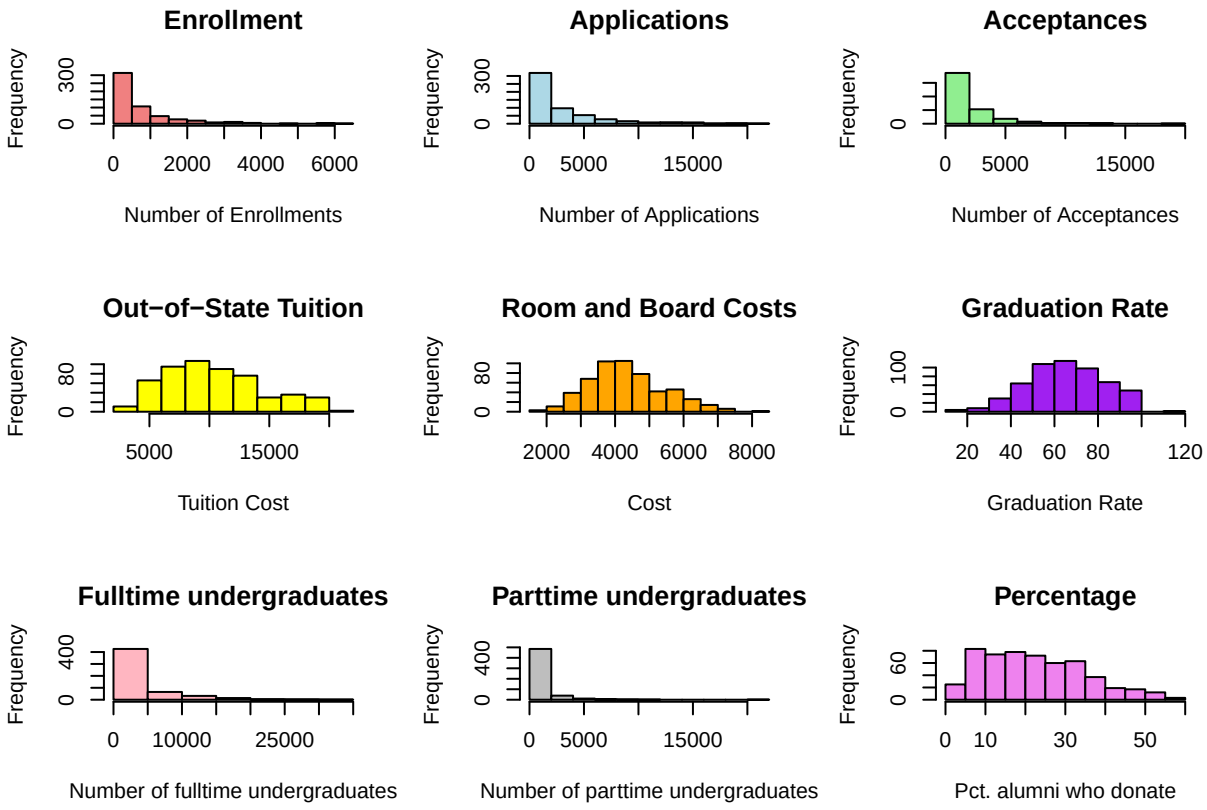
```
## corrplot 0.92 loaded
```

```r
library(gridExtra)
```

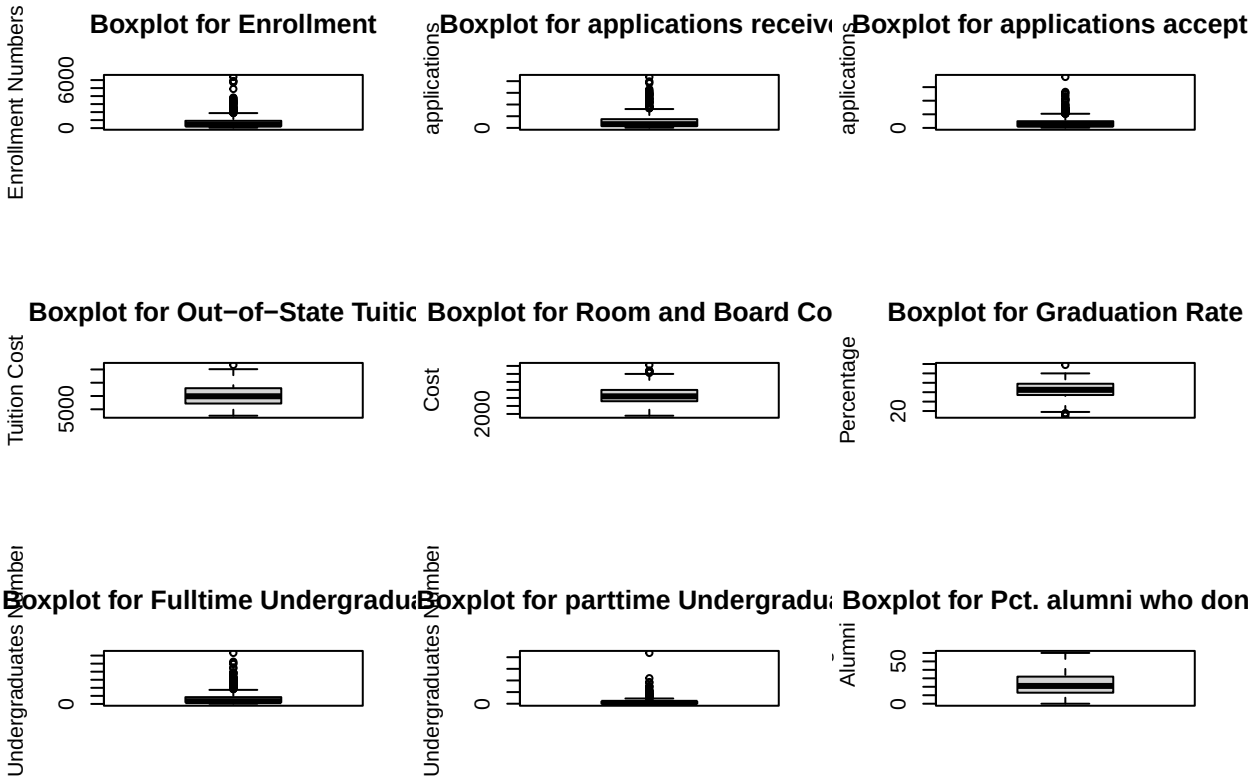
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

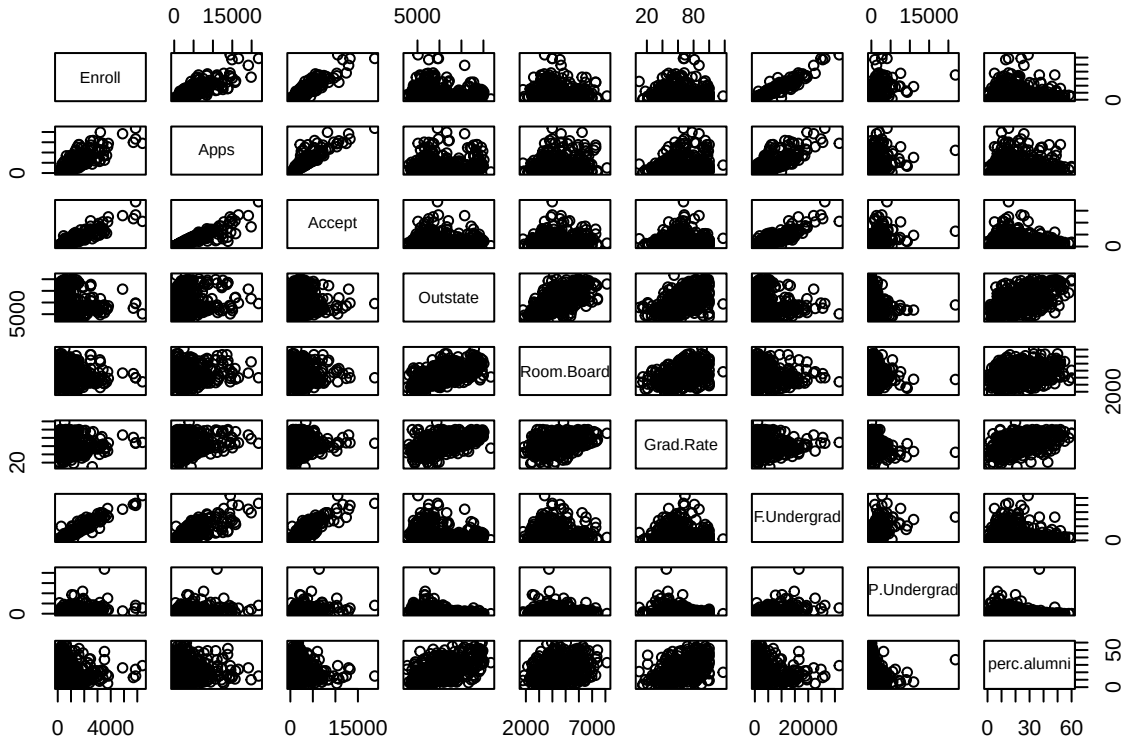
```r
# Plotting histograms for key numerical variables
par(mfrow = c(3, 3))
hist(train$Enroll, main = "Enrollment", xlab = "Number of Enrollments", col = "lightcoral")
hist(train$Apps, main = "Applications", xlab = "Number of Applications", col = "lightblue")
hist(train$Accept, main = "Acceptances", xlab = "Number of Acceptances", col = "lightgreen")
hist(train$Outstate, main = "Out-of-State Tuition", xlab = "Tuition Cost", col = "yellow")
hist(train$Room.Board, main = "Room and Board Costs", xlab = "Cost", col = "orange")
hist(train$Grad.Rate, main = "Graduation Rate", xlab = "Graduation Rate", col = "purple")
hist(train$F.Undergrad, main = "Fulltime undergraduates", xlab = "Number of fulltime undergradua
     col = "lightpink")
hist(train$P.Undergrad , main = "Parttime undergraduates", xlab = "Number of parttime undergradu
     col = "grey")
hist(train$perc.alumni, main = "Percentage", xlab = "Pct. alumni who donate", col = "violet")
```




```r
# Boxplots to check for outliers
boxplot(train$Enroll, main = "Boxplot for Enrollment", ylab = "Enrollment Numbers")
boxplot(train$Apps, main = "Boxplot for applications received", ylab = "Number of
        applications")
boxplot(train$Accept, main = "Boxplot for applications accepted", ylab = "Number of
        applications")
boxplot(train$Outstate, main = "Boxplot for Out-of-State Tuition", ylab = "Tuition Cost")
boxplot(train$Room.Board, main = "Boxplot for Room and Board Costs", ylab = "Cost")
boxplot(train$Grad.Rate, main = "Boxplot for Graduation Rate", ylab = "Percentage")
boxplot(train$F.Undergrad, main = "Boxplot for Fulltime Undergraduates", ylab = "Fulltime
        Undergraduates Numbers")
boxplot(train$P.Undergrad, main = "Boxplot for parttime Undergraduates", ylab = "Parttime
        Undergraduates Numbers")
boxplot(train$perc.alumni, main = "Boxplot for Pct. alumni who donate", ylab = "Percentage of
        Alumni")
```




```
# Scatter plots to see relationships between key variables
pairs(~Enroll + Apps + Accept + Outstate + Room.Board + Grad.Rate + F.Undergrad +
        P.Undergrad + perc.alumni, data = train)
```



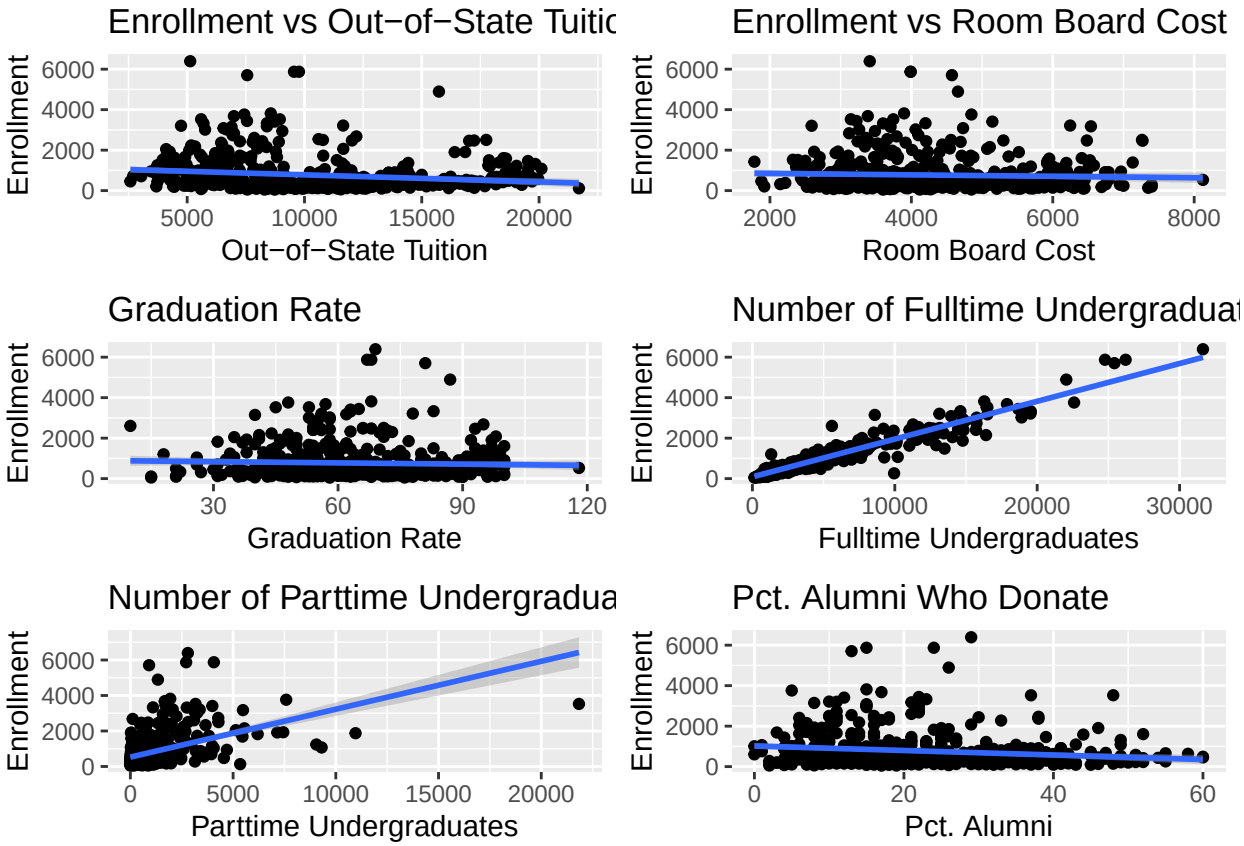

```
# Creating each plot
p1 <- ggplot(train, aes(x = Outstate, y = Enroll)) + geom_point() + geom_smooth(method = "lm") +
  ggtitle("Enrollment vs Out-of-State Tuition") + xlab("Out-of-State Tuition") + ylab("Enrollmen
p2 <- ggplot(train, aes(x = Room.Board, y = Enroll)) + geom_point() + geom_smooth(method = "lm")
  ggtitle("Enrollment vs Room Board Cost") + xlab("Room Board Cost") + ylab("Enrollment")
p3 <- ggplot(train, aes(x = Grad.Rate, y = Enroll)) + geom_point() + geom_smooth(method = "lm")
  ggtitle("Graduation Rate") + xlab("Graduation Rate") + ylab("Enrollment")
p4 <- ggplot(train, aes(x = F.Undergrad, y = Enroll)) + geom_point() + geom_smooth(method = "lm"
  ggtitle("Number of Fulltime Undergraduates") + xlab("Fulltime Undergraduates") + ylab("Enrollm
p5 <- ggplot(train, aes(x = P.Undergrad, y = Enroll)) + geom_point() + geom_smooth(method = "lm"
  ggtitle("Number of Parttime Undergraduates") + xlab("Parttime Undergraduates") + ylab("Enrollm
p6 <- ggplot(train, aes(x = perc.alumni, y = Enroll)) + geom_point() + geom_smooth(method = "lm"
  ggtitle("Pct. Alumni Who Donate") + xlab("Pct. Alumni") + ylab("Enrollment")

grid.arrange(p1, p2, p3, p4, p5, p6, nrow = 3)
```

```
## `geom_smooth()` using formula = 'y ~ x'

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Histogram plot: Histograms for a number of variables, including enrollment, applications, acceptances, and tuition expenses, are shown. To visually identify each histogram, distinct colors are used for each. By highlighting features like skewness and bimodality, these histograms aid in the comprehension of each variable's distribution.

Box plot: The distribution and central trends (median, quartiles) of the data and to detect outliers, the boxplots for the same variables. In order to possibly affect any statistical models constructed from this data, it is necessary to identify data points that differ considerably from the rest.

Scatter plot: The scatter plots with fitted linear regression lines to investigate correlations between enrollment and other factors like the cost of room & board and out-of-state tuition. This stage is crucial for finding any linear trends and correlations that might guide more in-depth statistical research or the creation of models.

The last graph indicates coherent visual overview of the associations between enrollment and other important factors, all scatter plots are finally neatly arranged on a single page using the grid.arrange function from gridExtra. This makes it easier to compare plots across numerous plots. This in-depth visual investigation can support initial data analysis, providing direction for additional statistical testing or predictive modeling.

Correlation matrix and checking for mutli-colinearity with PCA

It illustrates a methodical approach to data analysis that addresses multicollinearity in a dataset and visualizing correlations and using principal component analysis (PCA). First, the script searches for numeric columns in the dataset, eliminates low-variance columns, and creates a correlation matrix using pairwise full observations to handle missing values. In order to minimize multicollinearity problems and simplify the data structure, it performs principal component analysis (PCA) on the dataset to decrease dimensionality and find principal components that capture the most variation. The format of the findings includes the standard deviations, variance proportions, and cumulative proportions of these components.

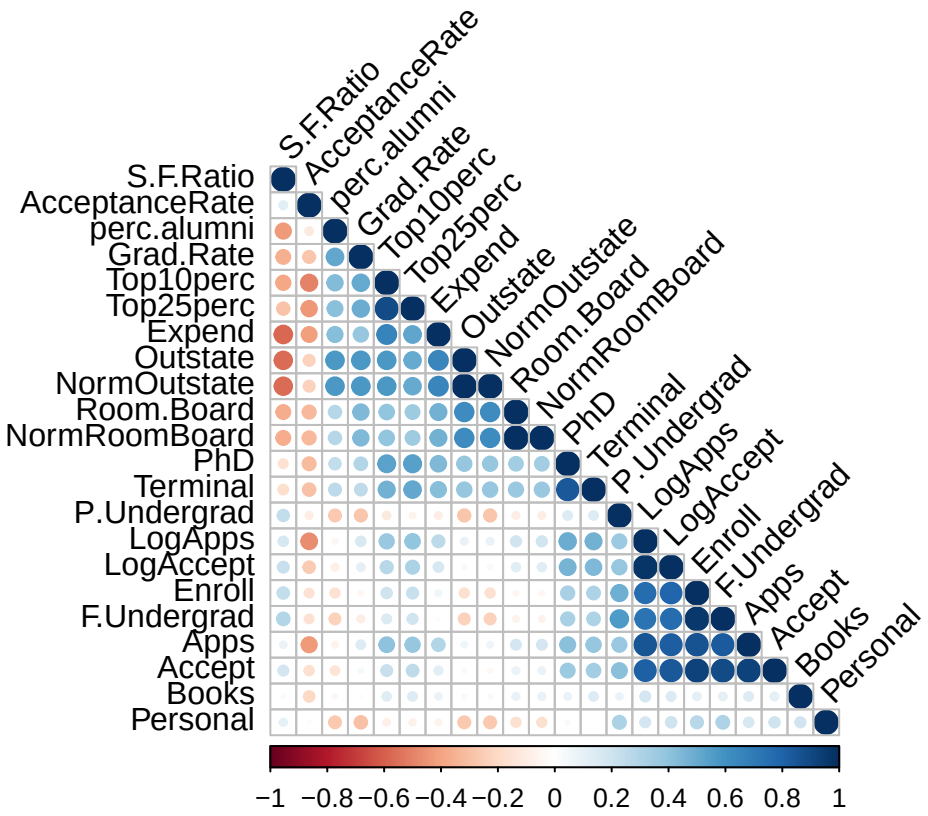
```
library(corrplot)

# Filtering to numeric columns
numeric_data <- train[, sapply(train, is.numeric)]

# Removing columns with zero variance or not enough variance
numeric_data <- numeric_data[, sapply(numeric_data, function(x) sd(x, na.rm = TRUE) > 0)]

# Handling NAs by pairwise complete observation
cor_matrix <- cor(numeric_data, use = "pairwise.complete.obs")

# Plotting the correlation matrix
corrplot(cor_matrix, method = "circle", type = "lower", order = "hclust",
         tl.col = "black", tl.srt = 45)
```




```
# PCA to handle multicollinearity
pca_result <- prcomp(College, scale. = TRUE)
importance_df <- summary(pca_result)$importance
importance_df <- as.data.frame(t(importance_df))
colnames(importance_df) <- c("Standard Deviation", "Proportion of Variance",
                             "Cumulative Proportion")
rownames(importance_df) <- paste("PC", 1:nrow(importance_df), sep = "")
print(importance_df)
```

```
##      Standard Deviation Proportion of Variance Cumulative Proportion
## PC1           2.3366500                0.30333                0.30333
## PC2           2.2485744                0.28089                0.58422
## PC3           1.0906806                0.06609                0.65031
## PC4           1.0276812                0.05867                0.70899
## PC5           0.9671631                0.05197                0.76095
## PC6           0.9209743                0.04712                0.80807
## PC7           0.7851744                0.03425                0.84232
## PC8           0.7665981                0.03265                0.87497
## PC9           0.7294071                0.02956                0.90453
## PC10          0.6402131                0.02277                0.92730
## PC11          0.5981814                0.01988                0.94718
```

```
## PC12          0.5540899               0.01706                 0.96424
## PC13          0.4307633               0.01031                 0.97454
## PC14          0.4093531               0.00931                 0.98385
## PC15          0.3790965               0.00798                 0.99184
## PC16          0.2964863               0.00488                 0.99672
## PC17          0.1898153               0.00200                 0.99872
## PC18          0.1515867               0.00128                 1.00000
```

The correlation analysis reveals insights into factors influencing Enrollment. Notably, Enrollment shows minimal correlation with Personal, books, S.F. Ratio is relatively weak, and improvement surcharge at Accept, Perc.alumni, Grad.Rate. This suggests that while these factors may have some influence, their impact on Enrollment. Understanding these correlations aids in optimizing Enrollment calculations and providing transparent decisions in college systems.

The standard deviations, proportions of variance, and cumulative proportions of variance explained by each principal component (PC) are shown in this Principal Component Analysis (PCA) output. The data is primarily characterized by the first two components (PC1 and PC2), which together account for approximately 58.4% of the total variance in the dataset.

Predictive analysis and results [15 points]

Linear Regression with k-fold cross validation resampling Initially, performing Linear Regression with k-fold cross validation resampling on selected features such as Apps, Accept, Outstate, Room board cost, graduate rate, fulltime undergrad, parttime undergrad, percent alumni and check for evaluation metrics on test data.

the required R packages are loaded, followed by stats for more statistical functions and caret for machine learning techniques. Tools for creating, adjusting, and assessing predictive models are included in these programs. In order to make sure that the outcomes are constant after several runs, a repeatable random seed is established using set.seed(123). A 10-fold cross-validation is specified in the trainControl() definition of the training control settings. By dividing the data into ten subsets and using nine of them for training and one for validation throughout each iteration, this technique helps to consistently evaluate model performance and decrease overfitting. Apps, Accept, Outstate, Room.Board, Grad.Rate, F.Undergrad, P.Undergrad, and perc.alumni are some of the indicators that the model formula uses to predict Enroll. Then, using the established formula and cross-validation parameters, a linear model (lm) is fitted to the training data (train dataset) using the train() function from the caret package. This function optimizes the model's parameters as needed in addition to fitting the model. To examine the model coefficients and other statistical metrics, which shed light on the importance and impact of each predictor in the model, the fitted model summary is printed.

```
# Loading the necessary packages
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.2
```

```
## Loading required package: lattice
```

```
library(stats)

# Setting up cross-validation method: 10-fold cross-validation
set.seed(123)  # for reproducibility
train_control <- trainControl(
  method = "cv",  # cross-validation
  number = 10   # number of folds
)

# Defining the model formula with selected features
formula <- Enroll ~ Apps + Accept + Outstate + Room.Board + Grad.Rate + F.Undergrad +
  P.Undergrad + perc.alumni

# Fitting the model
model <- train(
  formula,
  data = train,
  method = "lm",  # linear model
  trControl = train_control
)

# Summary of the model to check coefficients and model statistics
print(summary(model$finalModel))
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1166.22   -60.48    -2.39    52.48  1390.70
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 147.575856  45.819770   3.221 0.001356 **
## Apps          0.015046   0.007156   2.103 0.035964 *
## Accept        0.120664   0.013103   9.209  < 2e-16 ***
## Outstate     -0.002540   0.003429  -0.741 0.459206
## Room.Board   -0.035388   0.010369  -3.413 0.000692 ***
## Grad.Rate    -0.307995   0.650758  -0.473 0.636202
## F.Undergrad   0.129903   0.005306  24.481  < 2e-16 ***
## P.Undergrad  -0.016531   0.006900  -2.396 0.016935 *
## perc.alumni   3.648754   0.894124   4.081 5.17e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 199.1 on 534 degrees of freedom
## Multiple R-squared:   0.95,  Adjusted R-squared:  0.9493
## F-statistic:  1268 on 8 and 534 DF,  p-value: < 2.2e-16
```

```
# Using the model to make predictions on the test set
predictions <- predict(model, newdata = test)

# Calculating MSE
mse <- mean((predictions - test$Enroll)^2)
cat("Test Mean Square Error (MSE):", mse, "\n")
```

```
## Test Mean Square Error (MSE): 55139.46
```

```
# Calculating RMSE
rmse <- sqrt(mse)
cat("Test Root Mean Square Error (RMSE):", rmse, "\n")
```

```
## Test Root Mean Square Error (RMSE): 234.8179
```

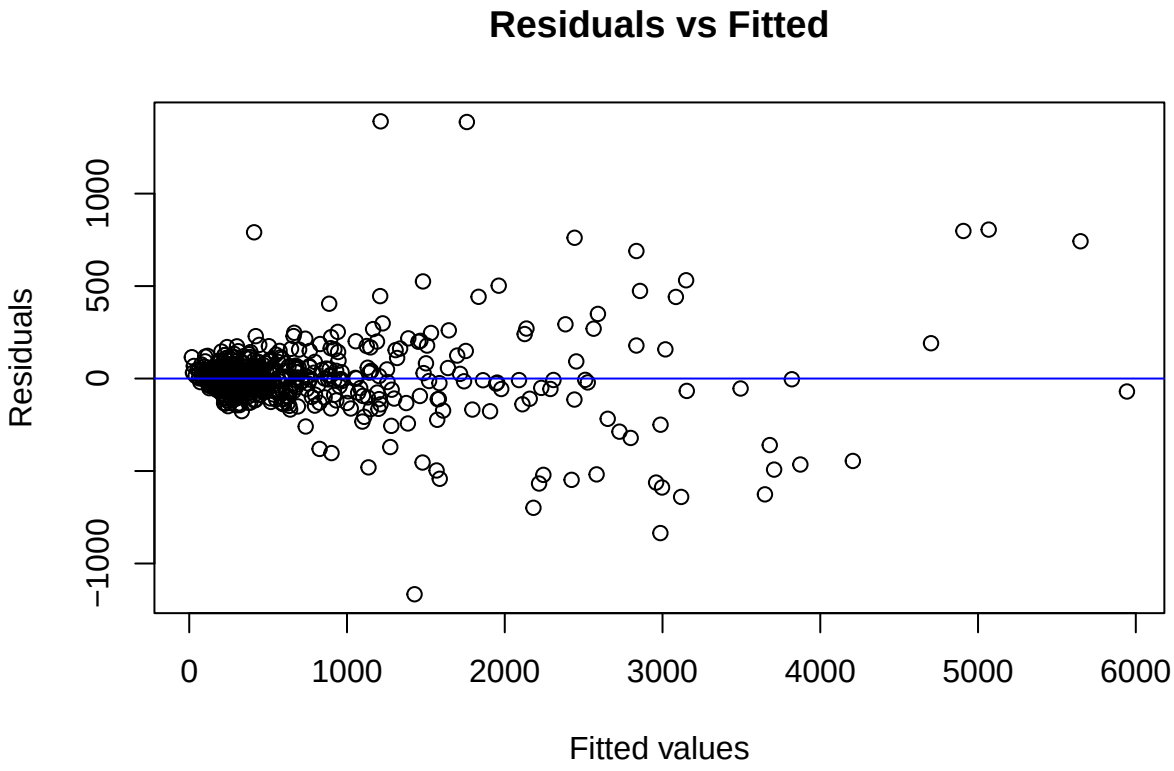
```
# Calculating MAE
mae <- mean(abs(predictions - test$Enroll))
cat("Test Mean Absolute Error (MAE):", mae, "\n")
```

```
## Test Mean Absolute Error (MAE): 107.1901
```

```
# Calculating R-Squared
r_squared <- summary(model$finalModel)$r.squared
cat("Test R-squared:", r_squared, "\n")
```

```
## Test R-squared: 0.9500031
```

```
# Residuals vs Fitted
plot(model$finalModel$fitted.values, resid(model$finalModel),
     xlab = "Fitted values", ylab = "Residuals", main = "Residuals vs Fitted")
abline(h = 0, col = "blue")
```



The result of a linear regression model that forecasts college enrollment (Enroll) based on a number of institutional parameters (Apps, Accept, Outstate, Room.Board, Grad.Rate, F.Undergrad, P.Undergrad, perc.alumni) produces this output. The expected impact of each predictor on enrollment is displayed in the coefficients section. For instance, an estimated 0.12 units more enrollment will result from a one unit rise in Accept. This signify the degree of statistical significance, while asterisks highlight the relevance of each predictor.

The model is used to forecast enrollment for a test dataset once it has been trained. Evaluating test data performance by these metrics, Mean Squared Error (MSE) and Root Mean Square Error (RMSE): These metrics indicate the average squared difference and the root of this difference between observed and predicted enrollment values, respectively, providing a measure of model accuracy.

Mean Absolute Error (MAE): This metric represents the average absolute difference between predicted and actual enrollments, offering another perspective on prediction accuracy.

R-squared (R^2): This statistic measures the proportion of variance in the dependent variable that is predictable from the independent variables, giving an indication of the goodness of fit.

Furthermore, a test dataset is used to construct test metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared, which show how well the model performs on unobserved data. The model's overall high R-squared value (0.95) means that the institutional features it includes can account for around 95% of the variance in college attendance. The model's performance is further validated by the test error metrics, where low mistake rates show that the model is good at predicting enrollment. The test's mean square error

(MSE), which is 55139.46 and quite high, can be decreased by using Random Forest Regression, an advanced statistical learning method.

The model is evaluated by creating a plot of residuals against fitted values. Non-linearity, uneven error variances, and outliers may all be found with the aid of this figure. In order to visually evaluate the residuals' randomness a critical step in verifying the linear regression's underlying assumptions a plot's horizontal line at zero is helpful illustrates the correlation between a linear regression model's fitted values and residuals. The residuals in this particular instance appear to be haphazardly dispersed about zero, lacking any discernible pattern. This implies that there is no bias in the predictions and that the mistakes in the model are random.

The residuals in this case are ideally randomly distributed about zero on the y-axis in a linear regression model. This indicates that the model is operating effectively.The model is not very well adapted to the available data, thanks to this methodical approach, and it may be enhanced by demonstrating its effectiveness on fresh, untested data, making it a reliable tool for predictive analysis in an educational setting. To create a trustworthy prediction model that is verified using exacting statistical techniques, the procedure described above is crucial.

Random Forest Regression with k-fold cross validation resampling

Now, performing Random Forest Regression with k-fold cross validation resampling on selected features such as Apps, Accept, Outstate, Room board cost, graduate rate, fulltime undergrad, parttime undergrad, percent alumni and check for evaluation metrics on test data, as the linear regression didn't give better results.

A Random Forest regression model for forecasting college enrollment based on several institutional variables may be built, evaluated, and interpreted according to the code. In order to generate and analyze Random Forest models, the script first loads the randomForest package. For consistent outcomes in the model fitting procedure, a repeatable seed (set.seed(123)) is established. An approach for 10-fold cross-validation is established by the trainControl function included in the caret package. To prevent overfitting and provide a reliable estimate of model performance, this method rotates through 10 subsets of the dataset, utilizing 9 for training and 1 for validation. With predetermined predictors and response (Enroll), the Random Forest model is trained via the train function. The accuracy and stability of the ensemble are increased by configuring the model to develop 500 trees (ntree = 500). To evaluate each predictor variable's relevance within the model, the importance = TRUE parameter is incorporated. The summary of the fitted model is produced to assess the overall model and variable significance statistics, revealing the elements that have the biggest effects on enrollment.

```
# Loading necessary libraries
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.2
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
# Setting up cross-validation method: 10-fold cross-validation
set.seed(123)  # for reproducibility
train_control <- trainControl(
  method = "cv", # cross-validation
  number = 10   # number of folds
)

# Fitting the Random Forest model
rf_model <- train(
  Enroll ~ Apps + Accept + Outstate + Room.Board + Grad.Rate + F.Undergrad +
    P.Undergrad + perc.alumni,
  data = train,
  method = "rf",  # random forest
  trControl = train_control,
  ntree = 500,    # number of trees
  importance = TRUE
)

# Summary of the model to check model statistics
print(rf_model)
```

```
## Random Forest
##
## 543 samples
##   8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 489, 489, 488, 489, 489, 490, ...
```

```
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   2     245.2198  0.9322285  120.2401
##   5     220.0307  0.9410184  107.1193
##   8     224.1272  0.9369704  106.4937
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 5.
```

```
# Using the model to make predictions on the test set
predictions <- predict(rf_model, newdata = test)

# Calculating Mean Squared Error (MSE)
mse_rf <- mean((test$Enroll - predictions)^2)
cat("Test Mean Square Error (MSE) for Random Forest:", mse_rf)
```

```
## Test Mean Square Error (MSE) for Random Forest: 34562.79
```

```
# Calculating RMSE
rmse_rf <- sqrt(mse_rf)
cat("Test Root Mean Square Error (RMSE) for Random Forest:", rmse_rf, "\n")
```

```
## Test Root Mean Square Error (RMSE) for Random Forest: 185.9107
```

```
# Calculate Mean Absolute Error (MAE)
mae_rf <- mean(abs(predictions - test$Enroll))
cat("Test Mean Absolute Error (MAE) for Random Forest:", mae_rf)
```

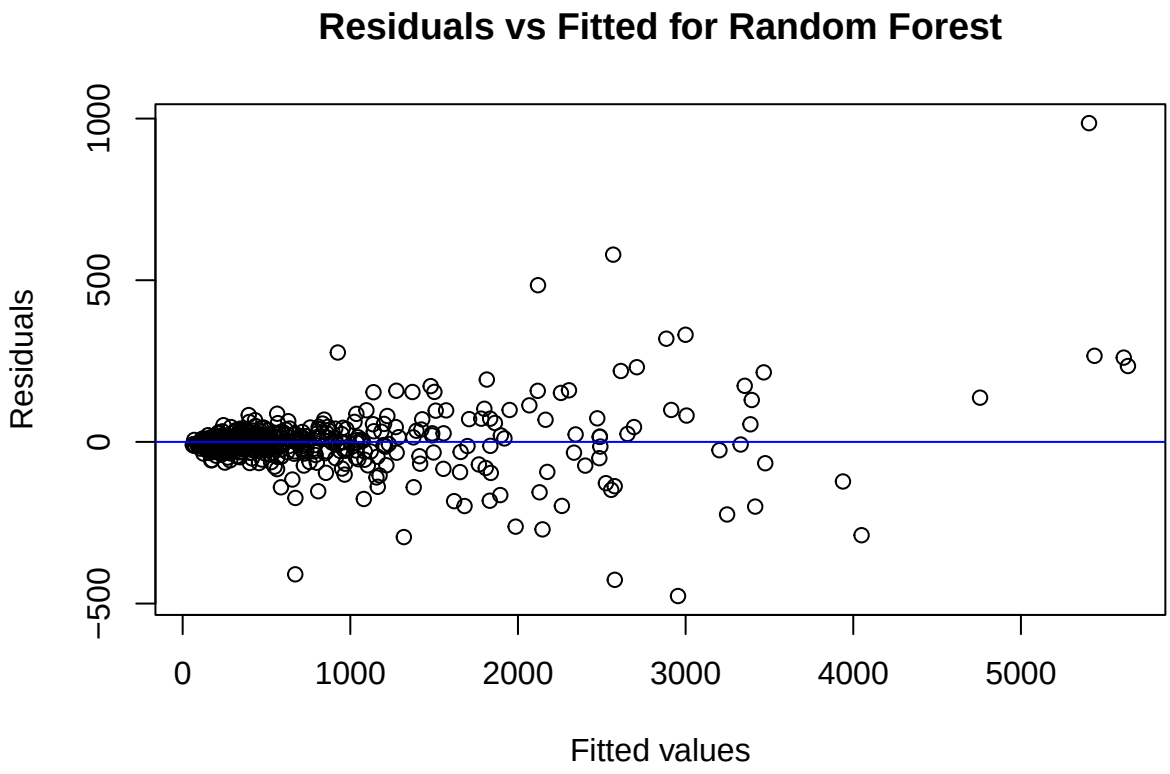
```
## Test Mean Absolute Error (MAE) for Random Forest: 95.73832
```

```
# Calculate R-squared (R²)
sse_rf <- sum((predictions - test$Enroll)^2)
sst_rf <- sum((test$Enroll - mean(test$Enroll))^2)
r_squared_rf <- 1 - sse_rf / sst_rf
cat("Test R-squared for Random Forest:", r_squared_rf)
```

```
## Test R-squared for Random Forest: 0.9671346
```

```
# Plot Residuals vs Fitted
fitted_values <- predict(rf_model, newdata = train)
residuals <- train$Enroll - fitted_values
plot(fitted_values, residuals,
```

```
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted for Random Forest")
abline(h = 0, col = "blue")  # Horizontal line at 0
```



This output displays the performance characteristics of a Random Forest regression model that was used to assess the predicted accuracy of the model using unseen data. Since it is expressed in squared units, it can be difficult to interpret the magnitude of the error directly in relation to the original variable. The Mean Squared Error (MSE) of 34,562.79 is a measure of the average squared difference between the actual and predicted enrollment values, this metric is improved by Random Forest Regression when compared to Linear Regression which indicates a better modeling technique. However, because it is expressed in the same units as the outcome variable (enrollment), the Root Mean Square Error (RMSE), which represents the mean departure from the actual enrollments, is around 185.91.An easy indicator of average error per prediction is provided by the Mean Absolute Error (MAE) of 95.74, which expresses the average absolute difference between the values that were predicted and the actual values. Last but not least, the model's R-squared value of 0.967 indicates that it correctly explains almost 96.7% of the variation in college enrollment, demonstrating a very high degree of prediction accuracy and model fit.

To visually evaluate the fit of the model, a plot of the residuals vs the fitted values for the training dataset is made. Plotting such patterns as heteroscedasticity, outliers, or other anomalies that may impact the model's performance is made easier with the use of this graphic. In this graphic, biases in the residual distribution are indicated by the horizontal line at zero.On the y-axis, the residuals seem to be randomly distributed around zero, with no clear trend. This shows that

the predictions are not biased and that the mistakes in the model are random. The residuals in this case are ideally dispersed randomly about zero in a random forest model. This is a positive indication of the model's effectiveness.

This method guarantees a comprehensive examination of the Random Forest model's predictive power and dependability. The process establishes the model's efficacy and pinpoints important college enrollment determinants, offering academic institutions insightful information by analyzing the model using cross-validation and a variety of statistical metrics.

Conclusion [5 points]

In order to effectively quantify the relationship between Enrollment and predictors such as the number of applications received, acceptance rates, out-of-state tuition, room and board costs, graduation rates, and the proportion of undergraduates enrolled full-time and part-time, as well as the percentage of alumni who donate.

By evaluating each predictor's significance using t- and p-values in linear regression models, it is possible to determine which factors are statistically significant and how much of an impact they have on enrollment rates. For instance, the model's coefficients show how enrollment is expected to be impacted by changes in these characteristics, such as tuition hikes or changes in graduation rates.

In contrast, Random Forest offers a significance score for every characteristic that indicates how well it predicts enrollment. It is especially helpful in identifying non-linear correlations and feature interactions. The result of this model, which shows a high R-squared value, indicates that the model has strong predictive ability and can explain a significant amount of the variance in enrollment.

Linear Regression Modeling

Scope: By estimating coefficients that indicate the change in the dependent variable for a one-unit change in an independent variable, linear regression models are mainly used to understand the relationship between the dependent variable and one or more independent variables. This model is very helpful in situations when decision-makers need to know how certain factors will effect enrollment since it gives a clear understanding of how each predictor impacts enrollment.

Generalizability: The assumptions of linearity, normalcy, independence, and homoscedasticity of residuals might restrict the generalizability of linear regression models. The model may not produce accurate predictions if certain presumptions are broken. In comparable educational environments or institutions, linear models may function quite effectively and offer clearly interpreted forecasts when certain presumptions are satisfied.

Limitations: Dependency on Assumptions: In order to do a linear regression analysis, data must satisfy certain assumptions, such as linearity, normalcy, homoscedasticity, and the absence of multicollinearity. These assumptions may not always hold true in actual data, which might result in inaccurate or biased estimations. Sensitivity to Outliers: Outliers can significantly affect the

model coefficients and distort the findings in linear models, which makes them extremely sensitive to them.

Improvements: Robust Regression Techniques: Reliable results may be obtained by using robust regression techniques, which are less susceptible to outliers and assumption violations. Feature Transformation: By applying transformations (square root, logarithmic, etc.) to variables, data may occasionally be normalized and variance stabilized, making it more suited for linear modeling. Diagnostic and Corrective methods: Model accuracy may be increased by using corrective methods like eliminating outliers or employing penalized regression approaches, as well as by routinely doing residual analysis to find assumptions breaches.

Random Forest Regression Modeling

Scope: Based on decision trees, Random Forest is a potent ensemble learning method that works well with nonlinear relationships and feature interactions without the need for transformation or adherence to assumptions. It can efficiently manage a large number of features and their interactions, making it especially helpful for complicated datasets where linear connections between variables are not expected.

Generalizability: Since Random Forest models do not rely on the premise of a linear connection, they often offer higher predictive accuracy and durability than linear models. Because of their ensemble nature, which employs averaging to increase forecast accuracy and reduce over-fitting, they are also less prone to overfit. But compared to linear regression, the predictions made by Random Forest models are more interpreted with less clarity. This makes it difficult to directly use Random Forest results for policy-making, because knowing how changing any one variable would affect things is essential.

Limitations: Model Complexity and Interpretability: Random Forest is capable of modeling intricate nonlinear interactions, but it is not as interpretable as more straightforward models such as linear regression, which makes it challenging to determine the importance and function of individual predictors. Overfitting in Specific Contexts: Random Forest can overfit even with generally adequate control over fitting, particularly when working with noisy data or when the number of trees is not calibrated to perfection. Computationally intensive: Training Random Forest models may be costly and time-consuming in terms of computing, particularly when dealing with big datasets and a lot of trees.

Improvements: Feature significance Analysis: By determining which factors have the most influence, using built-in techniques to assess feature significance helps help alleviate the interpretability problem. Model tuning: You may improve the performance of the model and avoid overfitting by experimenting with settings like the number of trees, max depth of trees, and the amount of characteristics examined at each split. Hybrid Models: By combining linear and random forest models in an ensemble approach, one may take use of both the interpretability of linear regression and the resilience of random forest.