Exercise 1 (2025) — Advanced Methods for Regression and Classification

College data (ISLR)

Olesia Galynskaia 12321492

2025-10-15

Loading and visualizing data

```
# Load data
if (!requireNamespace("ISLR", quietly = TRUE)) install.packages("ISLR")
library(ISLR)
data(College, package = "ISLR")

# Remove missing rows (simplifies modelling)
College <- na.omit(College)

# DS summary
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
summary(College)
```

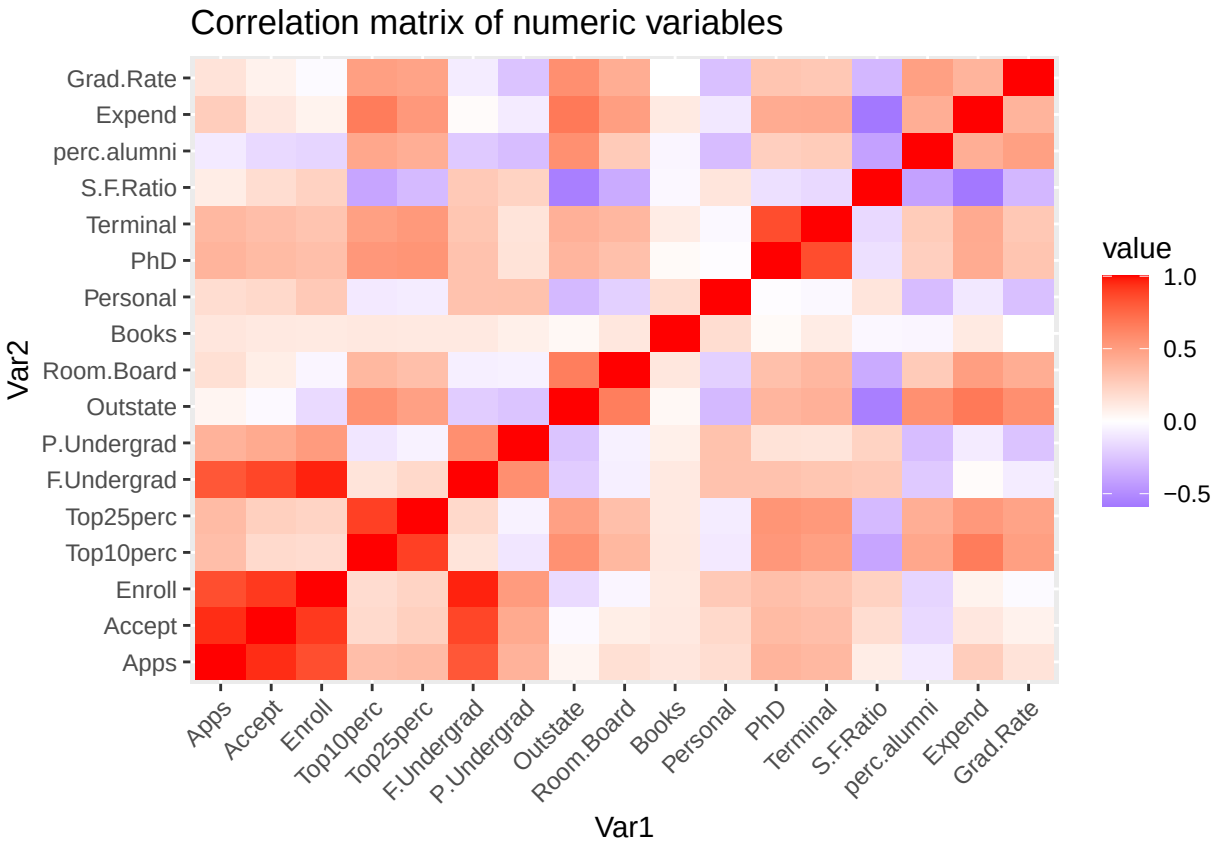
```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

```r
num_vars <- sapply(College, is.numeric)
cor_mat <- cor(College[, num_vars], use="complete.obs")

# Correlation map
install.packages("reshape2")
```

```
## package 'reshape2' successfully unpacked and MD5 sums checked
```

```
##
## The downloaded binary packages are in
##  C:\Users\lesya\AppData\Local\Temp\Rtmp276Q2e\downloaded_packages
```

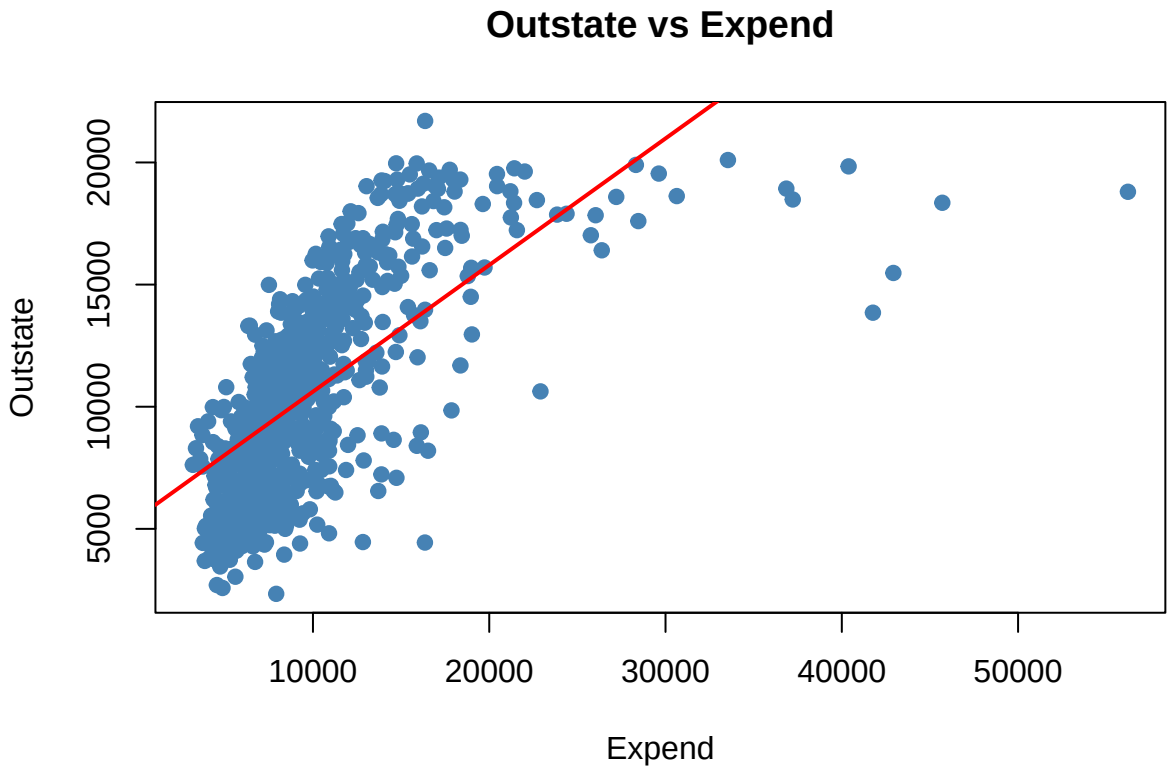
```
library(ggplot2)
library(reshape2)

corr_df <- melt(cor_mat)
ggplot(corr_df, aes(Var1, Var2, fill=value)) +
  geom_tile() +
  scale_fill_gradient2(low="blue", high="red", mid="white", midpoint=0) +
  theme(axis.text.x = element_text(angle=45, hjust=1)) +
  labs(title="Correlation matrix of numeric variables")
```



1) Outstate ~ Expend (simple linear regression)

```
m1 <- lm(Outstate ~ Expend, data = College)

# Plot data and regression line
plot(College$Expend, College$Outstate,
     main = "Outstate vs Expend",
```

```
     xlab = "Expend",
     ylab = "Outstate",
     pch = 19, col = "steelblue")
abline(m1, col = "red", lwd = 2)
```

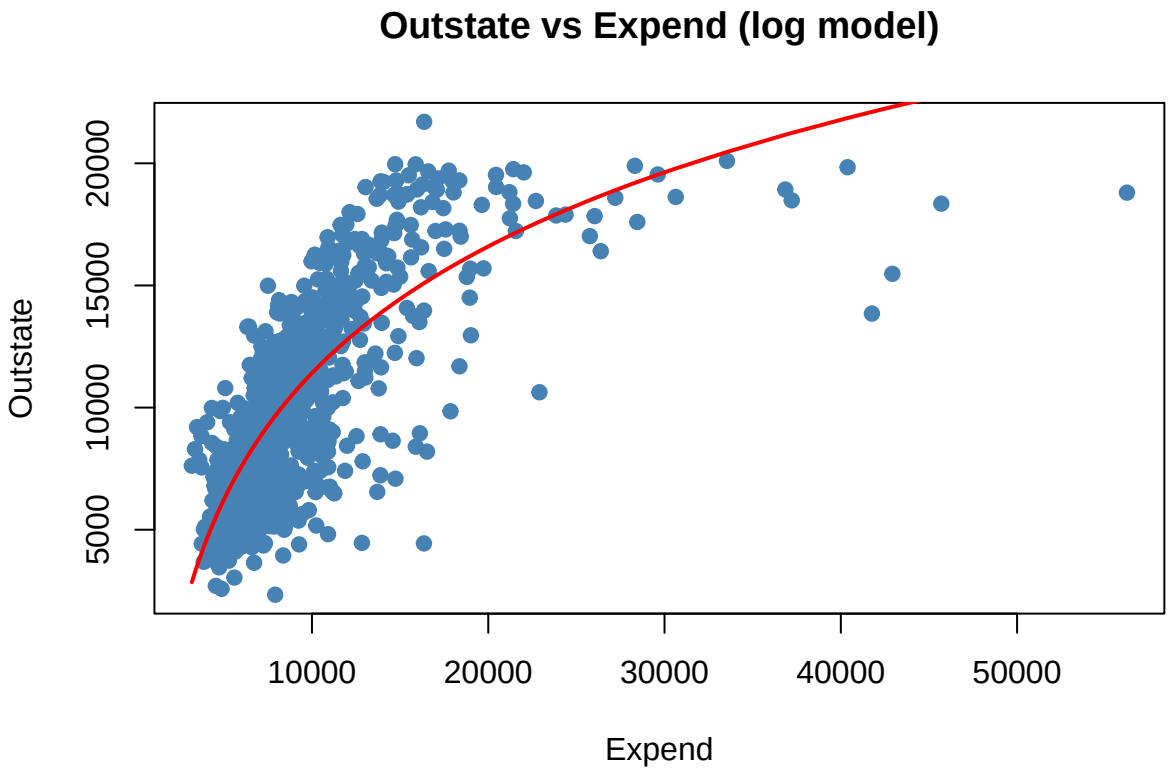


Comment:
There is an almost clear positive linear relationship between *Expend* and *Outstate*.
Colleges that spend more per student tend to charge higher out-of-state tuition. The spread of points increases for higher expenditures, indicating non-constant variance.

2) Improving the model

```
# Try to make the model follow the data more closely
m2 <- lm(Outstate ~ log(Expend), data = College)

# Compare visually
plot(College$Expend, College$Outstate,
     main = "Outstate vs Expend (log model)",
     xlab = "Expend",
     ylab = "Outstate",
     pch = 19, col = "steelblue")
```

```r
ord <- order(College$Expend)
lines(College$Expend[ord], fitted(m2)[ord], col = "red", lwd = 2)
```



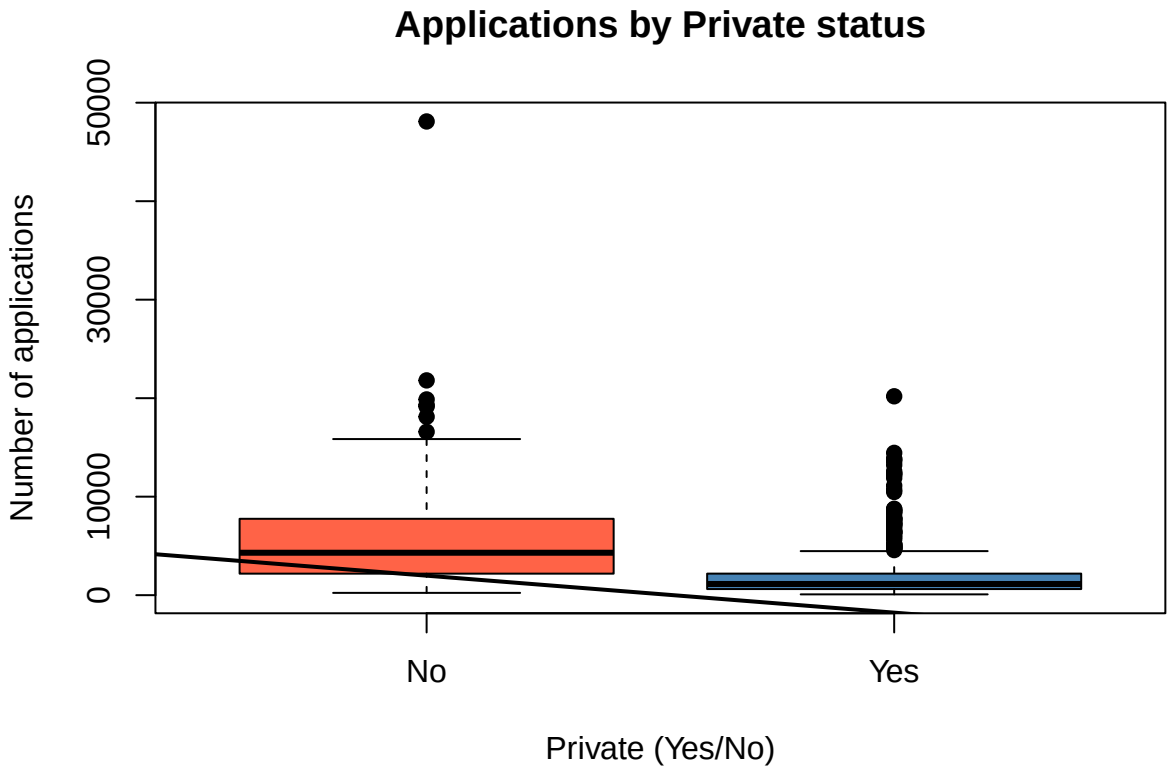
Comment:
In the first model, the spread of points increased for higher expenditures — the variance was not constant.
After taking the logarithm of expend, plot seems to fit the visible trend better. However, the improvement is modest how higher outstate.

3) Apps ~ Private

```r
# Fit regression model
m3 <- lm(Apps ~ Private, data = College)

# Show coefficients
coef(m3)
```

```
## (Intercept)  PrivateYes
##    5729.920   -3751.991
```

```r
# Plot the two groups with different colors
plot(College$Private, College$Apps,
     main = "Applications by Private status",
     xlab = "Private (Yes/No)",
     ylab = "Number of applications",
     col = c("tomato", "steelblue"),
     pch = 19)
abline(m3, col = "black", lwd = 2)
```



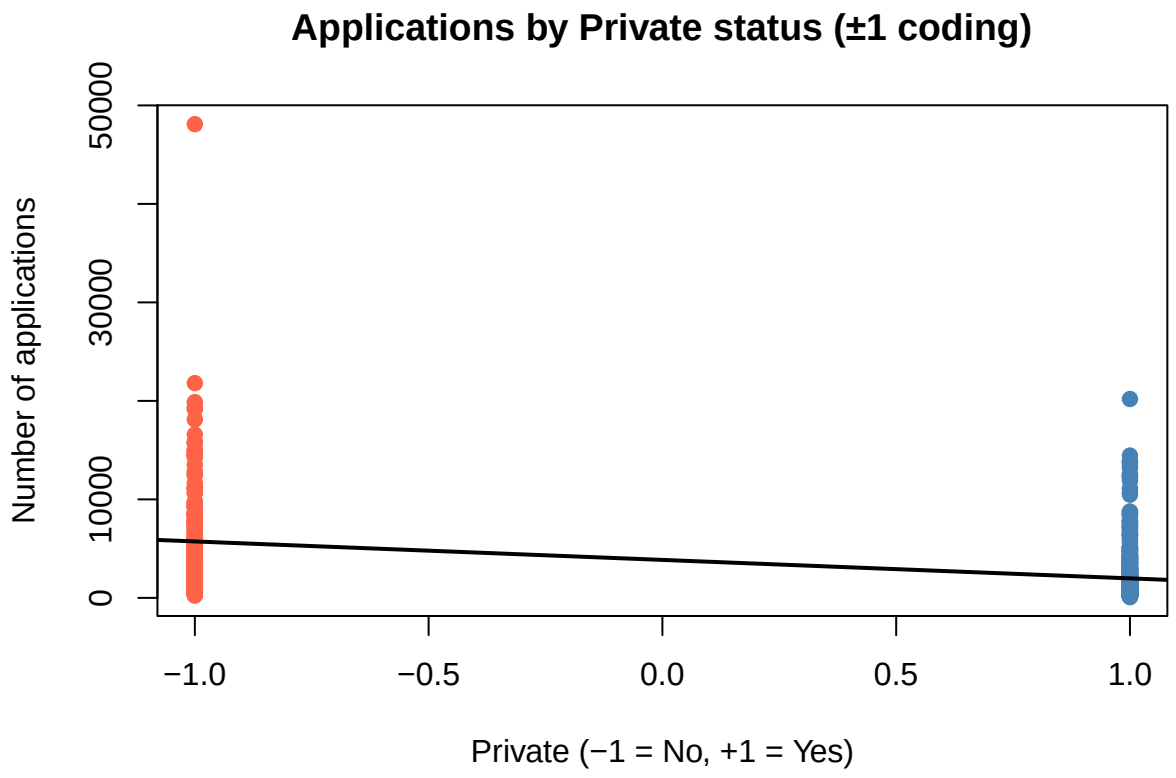
Comment:
The fitted regression model is Apps = 5730 - 3752 * PrivateYes B0 (Intercept) 5730 represents the mean number of applications for public colleges. B1 (PrivateYes) -3752 means that private colleges receive roughly 3,750 fewer applications on average than public ones. The negative slope seen on the plot confirms this difference. However, the boxplot also shows several outliers — a few public and private colleges that receive an unusually high number of applications compared to the rest. These outliers increase the spread of the data and may influence the fitted regression line.

4) Convert Private to ±1 and fit regression

```r
College$Private_pm1 <- ifelse(College$Private == "Yes", 1, -1)

# Fit regression model

m4 <- lm(Apps ~ Private_pm1, data = College)

# Show coefficients
coef(m4)
```

```
## (Intercept) Private_pm1
##    3853.925   -1875.995
```

```r
# Plot again
plot(College$Private_pm1, College$Apps,
     main = "Applications by Private status (±1 coding)",
     xlab = "Private (-1 = No, +1 = Yes)",
     ylab = "Number of applications",
     col = c("tomato", "steelblue")[as.numeric(College$Private)],
     pch = 19)

abline(m4, col = "black", lwd = 2)
```



Comment: The fitted regression model is Apps = 3854 - 1876 * Private_pm1 B0 (Intercept) 3854 represents the overall mean number of applications across both college types. B1 (Private±1) -1876 means that private colleges receive around 1,876 fewer applications on average than public ones. The negative slope on the plot confirms this difference. n this version, the variable Private was recoded to numeric values (-1 and +1), so the plot now shows points instead of boxplots. This change allows the linear relationship to be displayed directly as a continuous regression line. A few outliers can still be seen, mostly among public colleges with unusually high numbers of applications.

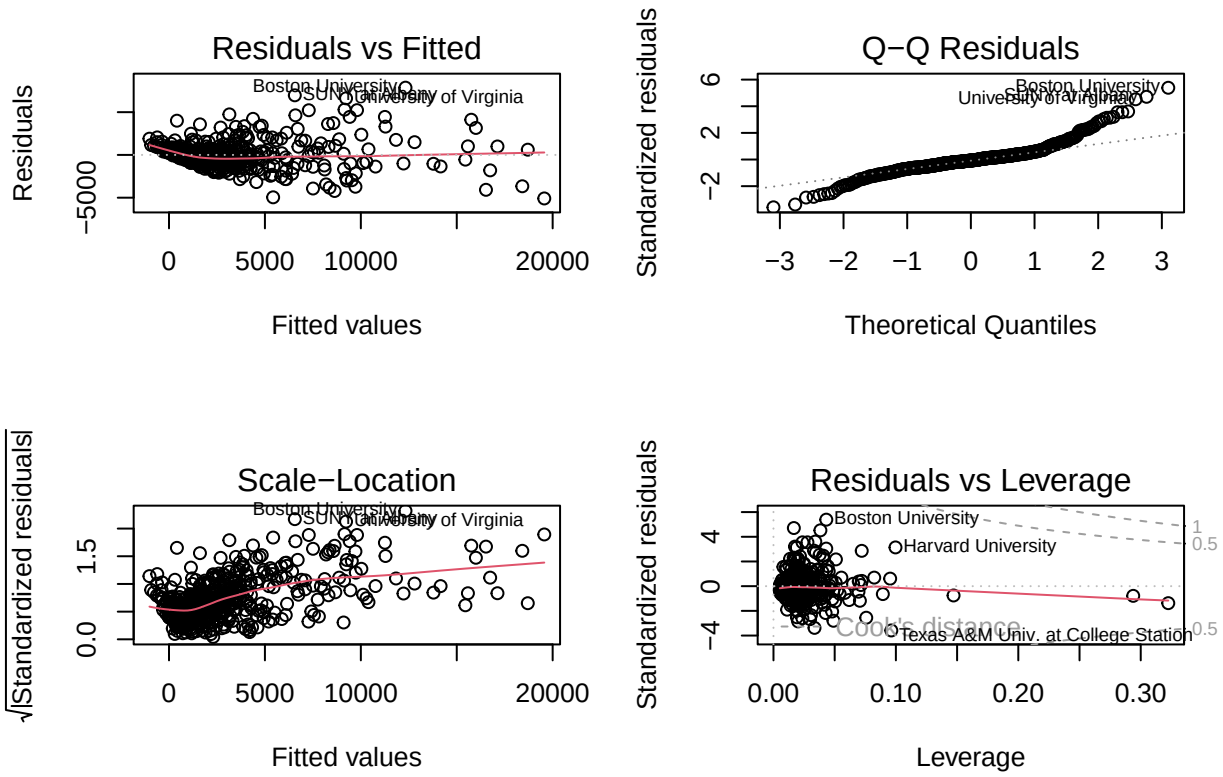
5) Predict Apps using meaningful predictors + RMSE (train/test)

```
set.seed(42)

# Train / test split ( 2/3 : 1/3)
n <- nrow(College)
train_idx <- sample(seq_len(n), size = floor(n * 2/3))
train <- College[train_idx, ]
test  <- College[-train_idx, ]

# Fit with content-wise predictors
m5 <- lm(
  Apps ~ Private + Top10perc + F.Undergrad + Outstate +
         Room.Board + Books + Personal + PhD +
         S.F.Ratio + perc.alumni + Expend,
  data = train
)

# Predictions
pred_train <- predict(m5, newdata = train)
pred_test  <- predict(m5, newdata  = test)

# RMSE
rmse_train <- sqrt(mean((train$Apps - pred_train)^2))
rmse_test  <- sqrt(mean((test$Apps  - pred_test )^2))
rmse_train; rmse_test
```

```
## [1] 1475.064
```

```
## [1] 2646.462
```

```
# Diagnostic plots
par(mfrow = c(2, 2))
plot(m5)
```




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

Comment:
We selected predictors that plausibly drive the number of applications (quality, costs, size, type, reputation) and removed variables that are consequences or near-duplicates (Accept, Enroll, Grad.Rate; Top25perc vs. Top10perc; Terminal vs. PhD; X as an ID).
The diagnostic plots check key regression assumptions (linearity, homoscedasticity, normality, influential points). The Residuals vs Fitted and Scale-Location plots show slight heteroscedasticity, meaning residuals spread increases with fitted values. The Q-Q plot indicates nearly normal errors except for a few outliers. The Residuals vs Leverage plot highlights a few influential colleges (e.g. Boston University, Harvard, Texas A&M). RMSE (train 1475, test 2646) suggests reasonable but not perfect generalization — the model explains main trends but is affected by variance and outliers.

6) Same model as (5) but with scaled predictors (variance = 1)

```
set.seed(42)

# split
n <- nrow(College)
```

```r
train_idx <- sample(seq_len(n), size = floor(n * 2/3))
train <- College[train_idx, ]
test  <- College[-train_idx, ]

# predictors from 5
vars_use <- c("Private","Top10perc","F.Undergrad","Outstate",
              "Room.Board","Books","Personal","PhD",
              "S.F.Ratio","perc.alumni","Expend")

# numeric columns to scale
num_to_scale <- intersect(vars_use, names(train)[sapply(train, is.numeric)])

# scale on train, keep center/scale to apply to test
sc_train_mat <- scale(train[, num_to_scale])
center <- attr(sc_train_mat, "scaled:center")
sdev   <- attr(sc_train_mat, "scaled:scale")

train_sc <- train
train_sc[, num_to_scale] <- sc_train_mat

test_sc <- test
# apply the SAME center/scale learned on train
test_sc[, num_to_scale] <- sweep(test_sc[, num_to_scale], 2, center, "-")
test_sc[, num_to_scale] <- sweep(test_sc[, num_to_scale], 2, sdev,   "/")

# fit model on scaled predictors (same formula as in 5)
form6 <- as.formula(
  paste("Apps ~", paste(vars_use, collapse = " + "))
)
m6 <- lm(form6, data = train_sc)

# coefficients (standardized for numeric X's)
coef(m6)
```

```
## (Intercept)  PrivateYes   Top10perc F.Undergrad    Outstate  Room.Board
##  3379.63224  -539.48927   454.22813  2969.56812   286.81688   394.62213
##       Books    Personal         PhD   S.F.Ratio perc.alumni      Expend
##   -10.04627  -176.69151  -146.56445   111.16999  -236.73668   501.88209
```

```r
# sort by absolute size (excluding intercept) to see "most influential"
coefs <- coef(m6)
imp <- sort(abs(coefs[names(coefs) != "(Intercept)"]), decreasing = TRUE)
imp
```

```
## F.Undergrad  PrivateYes      Expend   Top10perc  Room.Board    Outstate
##  2969.56812   539.48927   501.88209   454.22813   394.62213   286.81688
```

```
## perc.alumni    Personal         PhD   S.F.Ratio       Books
##   236.73668   176.69151   146.56445   111.16999    10.04627
```

```r
# RMSE
pred_tr6 <- predict(m6, newdata = train_sc)
pred_te6 <- predict(m6, newdata = test_sc)
rmse_tr6 <- sqrt(mean((train_sc$Apps - pred_tr6)^2))
rmse_te6 <- sqrt(mean((test_sc$Apps  - pred_te6)^2))
rmse_tr6; rmse_te6
```

```
## [1] 1475.064
```

```
## [1] 2646.462
```

Comment:
After scaling all predictors to unit variance, the regression coefficients become directly comparable. The largest standardized effects are for F.Undergrad, Private, and Expend, indicating that the number of full-time undergraduates, institutional type, and expenditures have the strongest influence on the number of applications. Variables such as Books, PhD, and S.F.Ratio show much smaller effects, suggesting limited predictive relevance. The RMSE values for the training (1475) and test data (2646) remain identical to the unscaled model, confirming that scaling affects coefficient comparability but not model performance.

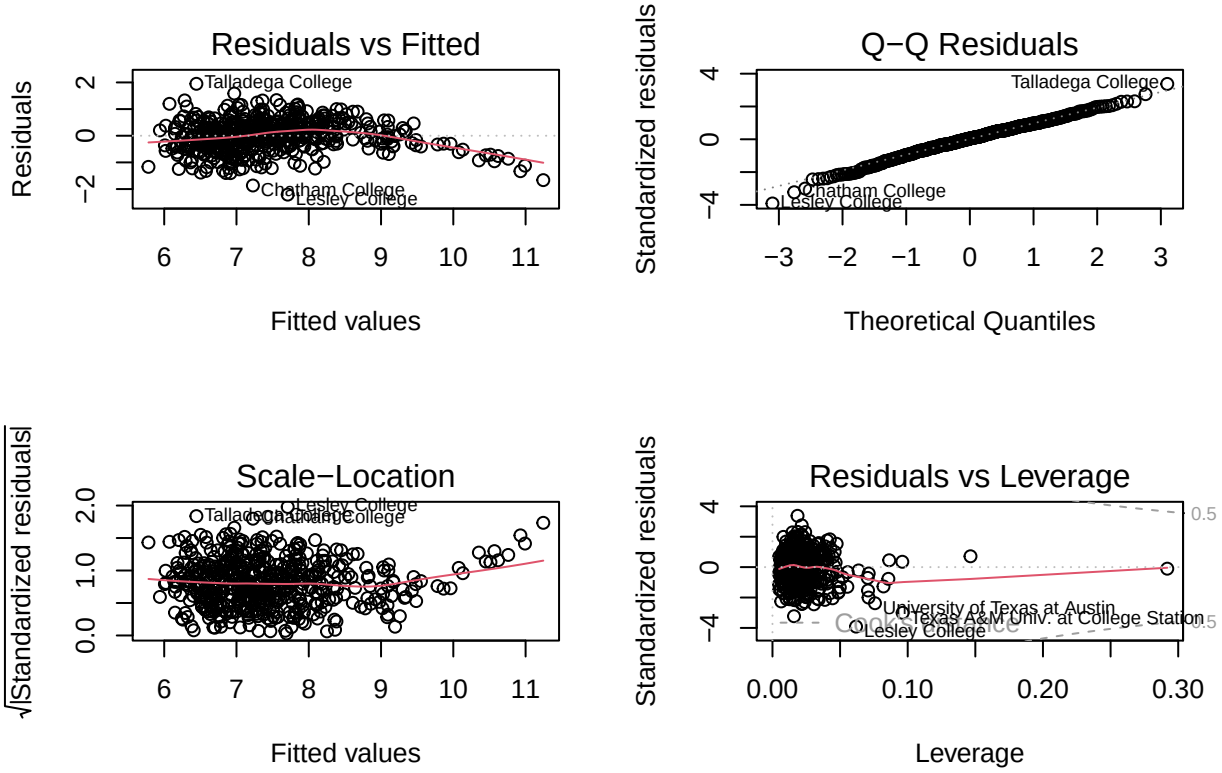
7) RMSE is already calculated in 5 and 6

Comment: The RMSE values for models 5 and 6 are the same for both training (1475) and test data (2646). This means that both models perform equally well and give identical predictions. Scaling the variables does not change how well the model fits or predicts — it only affects the size of the coefficients, making them easier to compare.

8) Log-transformed response

```r
set.seed(42)

# train/test split
train_idx <- sample(1:nrow(College), size = floor(2/3 * nrow(College)))
train_log <- College[train_idx, ]
test_log  <- College[-train_idx, ]

# log-transform response
train_log$log_Apps <- log(train_log$Apps + 1)
test_log$log_Apps  <- log(test_log$Apps + 1)
```

```r
# fit model (same predictors as in 5)
m8 <- lm(log_Apps ~ Private + Top10perc + F.Undergrad + Outstate + Room.Board +
           Personal + PhD + S.F.Ratio + perc.alumni + Expend,
         data = train_log)

# diagnostic plots
par(mfrow = c(2, 2))
plot(m8)
```




```r
# RMSE (on log-scale)
pred_train_log <- predict(m8, newdata = train_log)
pred_test_log  <- predict(m8, newdata = test_log)
rmse_train_log <- sqrt(mean((train_log$log_Apps - pred_train_log)^2))
rmse_test_log  <- sqrt(mean((test_log$log_Apps - pred_test_log)^2))

rmse_train_log; rmse_test_log
```

```
## [1] 0.5761923
```

```
## [1] 0.5953341
```

Comment: After applying a log transformation to the response variable, the residuals appear more evenly distributed around zero, and the variance is more constant across fitted values. The Q-Q plot shows that residuals follow the normal line much more closely, indicating improved normality. Outliers such as "Talladega College" and "Lesley College" are still visible, but their influence is reduced. The RMSE values for the training (0.58) and test data (0.60) confirm a consistent model fit. Compared to the untransformed model, this one better satisfies linear regression assumptions — especially homoscedasticity and normality — making it more appropriate overall.

9) How to compare models

We cannot directly compare the RMSE values of models 5 and 8 because they are on different scales. Model 5 predicts Apps, while model 8 predicts log(Apps). To compare them fairly, we would need to back-transform the log predictions (exp(pred) - 1) and then compute RMSE on the original scale, or use other metrics like R^2 or AIC. Based on residual plots, the log model (model 8) seems to fit better, since it has more constant variance and more normally distributed residuals.