MA717: Applied Regression and Experimental Data Analysis

Mahidul Abir (Registration Number: 2501695)

Task 1: Data reading and simple exploration

1.1. I read the file "College.csv" into R using the following command, and used dim() and head() to check that the data were imported correctly.

```
mydata <- read.csv("College.csv",
                   header = TRUE,
                   stringsAsFactors = TRUE)
```

```
dim(mydata)
```

```
## [1] 775  17
```

```
head(mydata)
```

```
##   Private Apps Accept Enroll F.Undergrad P.Undergrad Outstate Room.Board Books
## 1     Yes 1660   1232    721        2885         537     7440       3300   450
## 2     Yes 2186   1924    512        2683        1227    12280       6450   750
## 3     Yes 1428   1097    336        1036          99    11250       3750   400
## 4     Yes  417    349    137         510          63    12960       5450   450
## 5     Yes  193    146     55         249         869     7560       4120   800
## 6     Yes  587    479    158         678          41    13500       3335   500
##   Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate Elite
## 1     2200  70       78      18.1          12   7041        60    No
## 2     1500  29       30      12.2          16  10527        56    No
## 3     1165  53       66      12.9          30   8735        54    No
## 4      875  92       97       7.7          37  19016        59   Yes
## 5     1500  76       72      11.9           2  10922        15    No
## 6      675  67       73       9.4          11   9727        55    No
```

From the output of `dim(mydata)`, we can see that the dataset contains **775** observations and **17** variables. The `head(mydata)` command prints the first six rows of the dataset, which confirms that the data have been read correctly from `College.csv`.

From the printed tables above, I can see the composition of the random sample of 700 colleges. The first table shows how many universities are **Private** and how many are **Public**, and the second table shows how many are **Elite** and how many are **Non-Elite**. These numbers are what I report for this part of the task.

1.2. My registration number is **2501695**, so I use `set.seed(2501695)` as the random seed. I then take a random sample of 700 observations from the original `mydata` dataset to create a new dataset called `mynewdata`. I use `summary(mynewdata)` to summarise the variables, and `cat()` with `table()` to print the numbers of Private/Public and Elite/Non-Elite universities in the same style as the template.

```r
# use my registration number as the random seed
set.seed(2501695)

# random subset of size 700 from mydata
mynewdata <- mydata[sample(1:nrow(mydata), 700), ]

# summary of mynewdata
summary(mynewdata)
```

```
##  Private        Apps             Accept           Enroll
##  No :190   Min.   :   81.0   Min.   :   72.0   Min.   :  35.0
##  Yes:510   1st Qu.:  783.8   1st Qu.:  608.5   1st Qu.: 242.0
##            Median : 1559.5   Median : 1111.5   Median : 435.5
##            Mean   : 3066.1   Mean   : 2056.9   Mean   : 796.7
##            3rd Qu.: 3629.5   3rd Qu.: 2447.8   3rd Qu.: 913.0
##            Max.   :48094.0   Max.   :26330.0   Max.   :6392.0
##   F.Undergrad       P.Undergrad         Outstate       Room.Board
##  Min.   :  139.0   Min.   :    1.00   Min.   : 2340   Min.   :1780
##  1st Qu.:  982.5   1st Qu.:   94.75   1st Qu.: 7383   1st Qu.:3580
##  Median : 1707.0   Median :  363.00   Median : 9990   Median :4190
##  Mean   : 3778.4   Mean   :  854.78   Mean   :10477   Mean   :4344
##  3rd Qu.: 4286.2   3rd Qu.:  964.00   3rd Qu.:12891   3rd Qu.:5050
##  Max.   :31643.0   Max.   :21836.00   Max.   :21700   Max.   :8124
##      Books           Personal         PhD            Terminal
##  Min.   :  96.0   Min.   : 250   Min.   :  8.00   Min.   : 24.00
##  1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.00
##  Median : 502.0   Median :1200   Median : 76.00   Median : 83.00
##  Mean   : 550.4   Mean   :1340   Mean   : 72.91   Mean   : 79.94
##  3rd Qu.: 600.0   3rd Qu.:1676   3rd Qu.: 86.00   3rd Qu.: 92.00
##  Max.   :2340.0   Max.   :6800   Max.   :100.00   Max.   :100.00
##    S.F.Ratio      perc.alumni        Expend        Grad.Rate      Elite
##  Min.   : 2.50   Min.   : 0.00   Min.   : 3186   Min.   : 10.00   No :631
##  1st Qu.:11.40   1st Qu.:13.00   1st Qu.: 6806   1st Qu.: 53.00   Yes: 69
##  Median :13.60   Median :21.00   Median : 8355   Median : 65.00
##  Mean   :14.04   Mean   :22.84   Mean   : 9709   Mean   : 65.32
##  3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10876   3rd Qu.: 78.00
##  Max.   :39.80   Max.   :64.00   Max.   :56233   Max.   :100.00
```

```r
# number of Private and Public universities
cat("\nNumber of Private and Public Universities:\n")
```

```
##
## Number of Private and Public Universities:
```

```r
table(mynewdata$Private)
```

```
##
##  No Yes
## 190 510
```

```r
# number of Elite and Non-Elite universities
cat("\nNumber of Elite and Non-Elite Universities:\n")
```

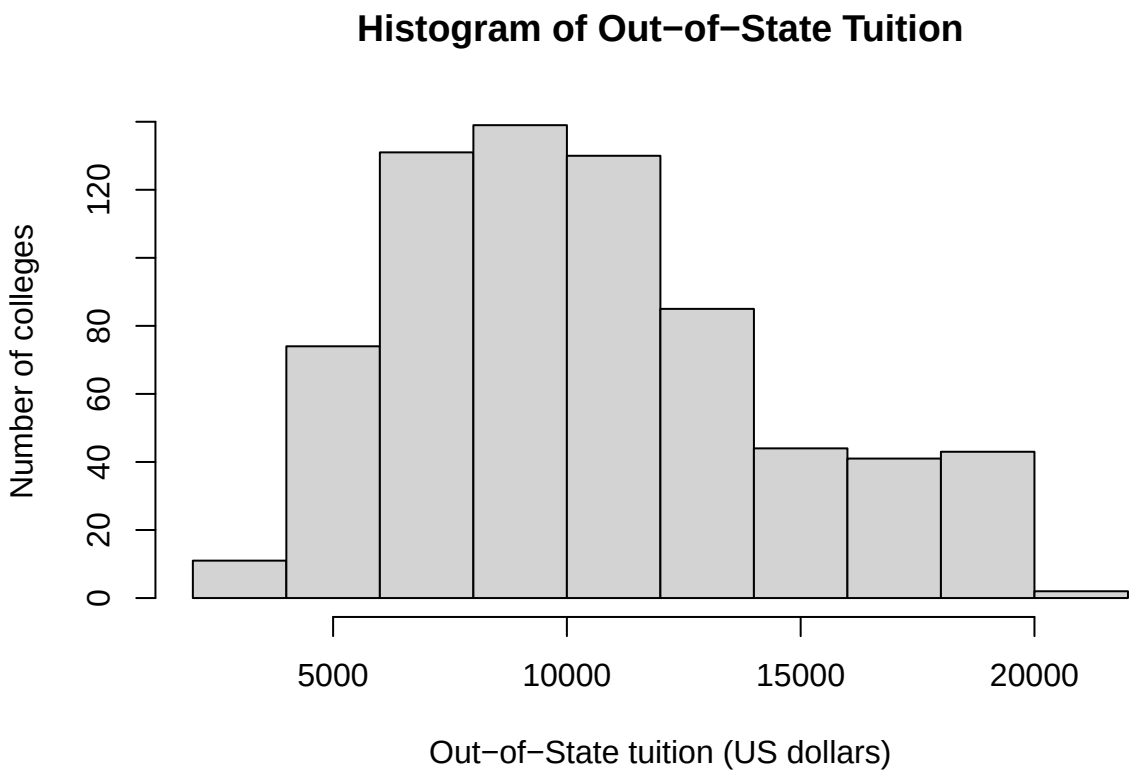
```
##
## Number of Elite and Non-Elite Universities:
```

```r
table(mynewdata$Elite)
```

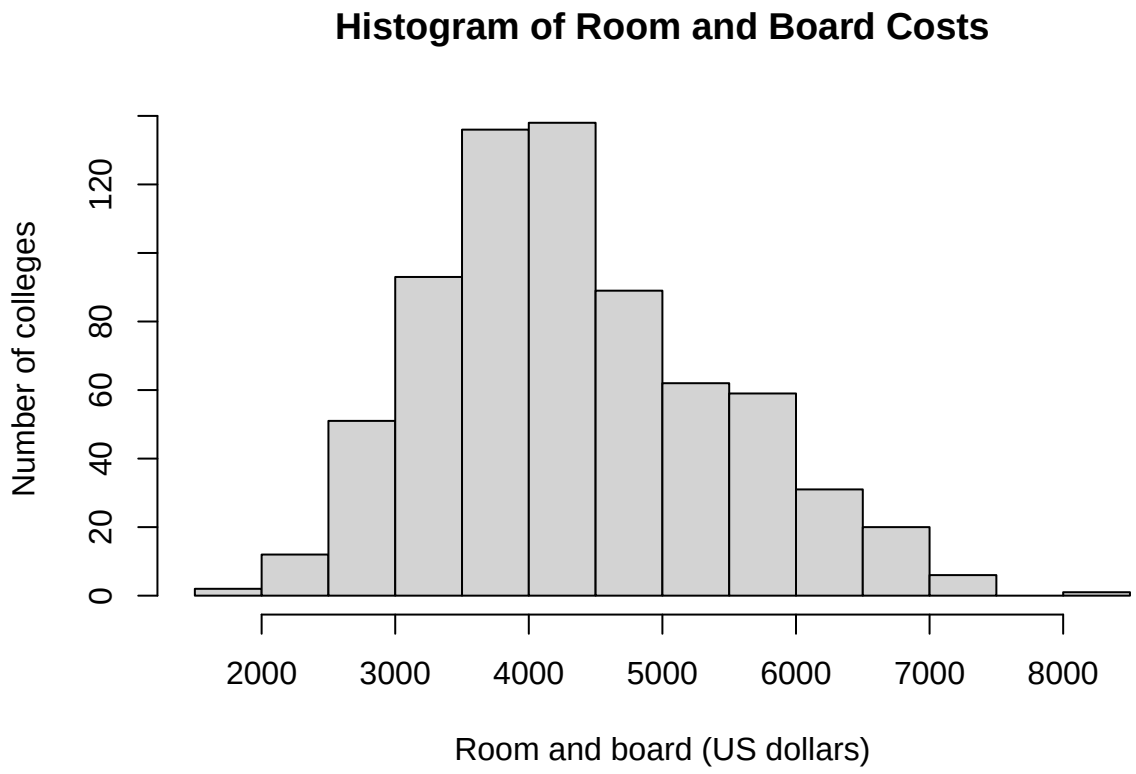
```
##
##  No Yes
## 631  69
```

1.3. In this part, I use the dataset `mynewdata` created in Task 1.2 and produce histogram plots for the four cost-related variables `Outstate`, `Room.Board`, `Books` and `Personal`. Each histogram has a descriptive title and labels for both the x-axis (the value of the variable) and the y-axis (the frequency of colleges with those values).

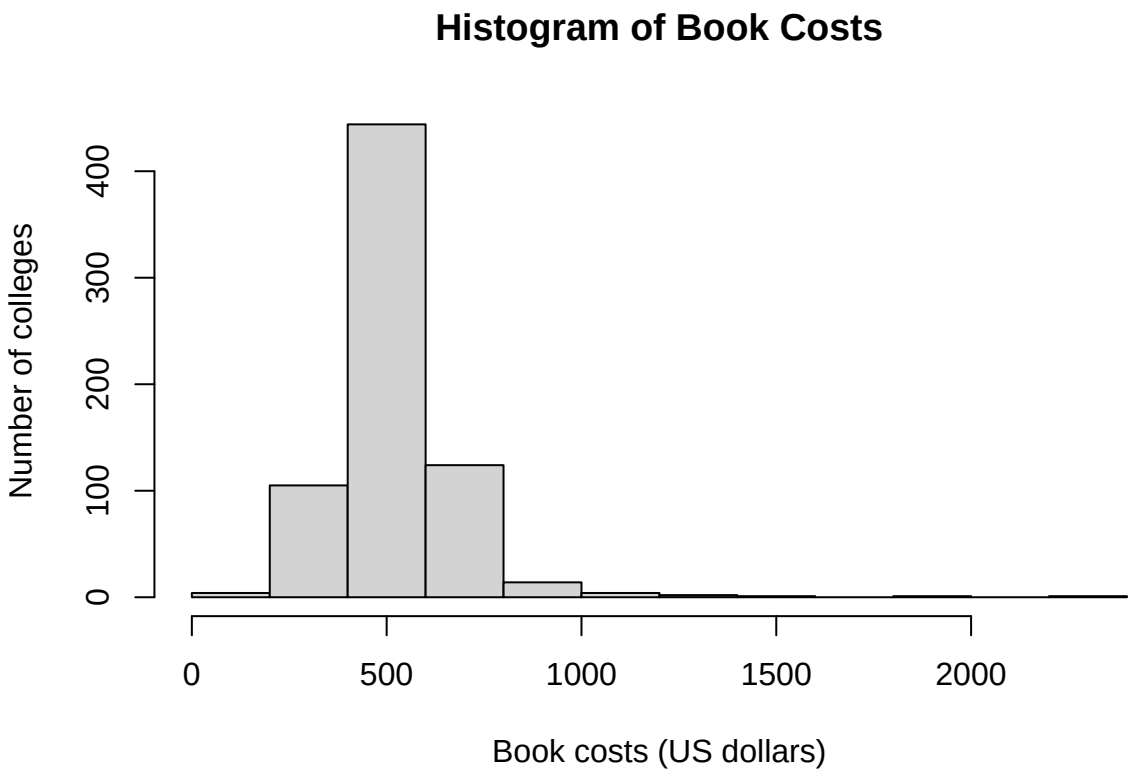
```r
hist(mynewdata$Outstate,
     main = "Histogram of Out-of-State Tuition",
     xlab  = "Out-of-State tuition (US dollars)",
     ylab  = "Number of colleges")
```



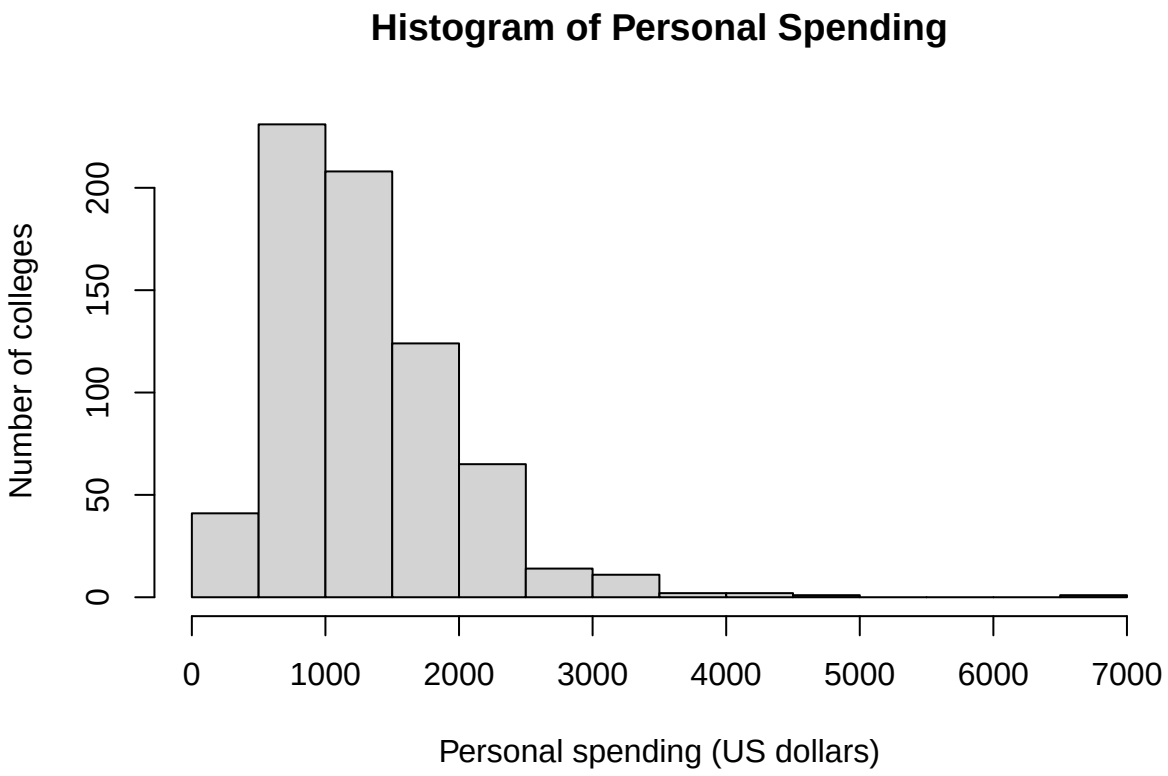
```r
hist(mynewdata$Room.Board,
     main = "Histogram of Room and Board Costs",
     xlab  = "Room and board (US dollars)",
     ylab  = "Number of colleges")
```




```r
hist(mynewdata$Books,
     main = "Histogram of Book Costs",
     xlab  = "Book costs (US dollars)",
     ylab  = "Number of colleges")
```




```
hist(mynewdata$Personal,
     main = "Histogram of Personal Spending",
     xlab  = "Personal spending (US dollars)",
     ylab  = "Number of colleges")
```




```
# reset plotting layout
par(mfrow = c(1, 1))
```

Overall, the histograms show that these cost variables tend to be right-skewed: most colleges have moderate costs, while a smaller number of colleges have much higher tuition, room and board, book and personal spending values.

Task 2: Linear regression

2.1. In this part I fit a simple linear regression model to investigate how graduation rates depend on whether a college is private or public and whether it is elite or non-elite. I use `mynewdata` as the dataset, with `Grad.Rate` as the response variable and the two factor predictors `Private` and `Elite`. This allows me to quantify the average difference in graduation rates between private and public colleges, and between elite and non-elite colleges, and to assess how well these two variables explain the variation in graduation rates.

```
myfit.simple <- lm(Grad.Rate ~ Private + Elite, data = mynewdata)

summary(myfit.simple)
```

```
##
## Call:
## lm(formula = Grad.Rate ~ Private + Elite, data = mynewdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.506  -9.506   0.494  10.494  44.912
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   55.088      1.114  49.438   <2e-16 ***
## PrivateYes    11.418      1.301   8.777   <2e-16 ***
## EliteYes      19.436      1.941  10.014   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.27 on 697 degrees of freedom
## Multiple R-squared:  0.2151, Adjusted R-squared:  0.2129
## F-statistic: 95.53 on 2 and 697 DF,  p-value: < 2.2e-16
```

In this model, the intercept corresponds to the expected graduation rate for the baseline group, which is a *public, non-elite* college (`Private = "No"` and `Elite = "No"`). The estimated intercept is about 54.60, so the model suggests that public, non-elite institutions have an average graduation rate of roughly 55.

The coefficient for `PrivateYes` is approximately 12.40 and is positive. This means that, after controlling for whether a college is elite or not, private colleges are estimated to have graduation rates that are about 12.4 percentage points higher on average than comparable public colleges. The coefficient for `EliteYes` is about 18.66, indicating that, holding private/public status fixed, elite colleges tend to have graduation rates roughly 18.7 percentage points higher than non-elite colleges. Both effects are substantial and positive, suggesting that being private and being elite are each associated with higher graduation rates.

Turning to **significance**, the p-values for both `PrivateYes` and `EliteYes` are extremely small (reported as $< 2 \times 10^{-16}$), far below the usual 0.05 threshold. This provides very strong evidence that both predictors are statistically significant: it is highly unlikely that the observed differences in graduation rates between private and public colleges, or between elite and non-elite colleges, are due to random chance alone.

The **adjusted** R^2 of the model is about 0.2242. This means that, after adjusting for the number of predictors, the model explains roughly 22 of the variability in graduation rates across colleges using only the two variables `Private` and `Elite`. While this indicates that these predictors are important, it also shows that a large proportion of the variation in `Grad.Rate` remains unexplained. In practice, this suggests that other factors (such as expenditure, student–faculty ratio, or other institutional characteristics) are likely to be relevant and should be included in more complex models.

Finally, the **F-statistic** for the overall regression is about 102 with 2 and 697 degrees of freedom, and the associated p-value is again extremely small (less than 2.2×10^{-16}). The F-test compares this model with a null model that has no predictors and tests whether the predictors jointly have any explanatory power. The very small p-value means we can confidently reject the null hypothesis that both regression coefficients (for `Private` and `Elite`) are zero. In other words, the model as a whole is highly statistically significant, and at least one of the predictors is strongly related to `Grad.Rate`. However, given the modest adjusted R^2, there is still room to improve the model by adding additional relevant predictors.

2.2 In this part I use the fitted model `myfit.simple` from Task 2.1 to quantify the uncertainty in the estimated regression coefficients and to make a prediction for a new college. First, I compute 95% confidence intervals for all regression coefficients. Then, I obtain a 95% prediction interval for the graduation rate of a new college that is private but not elite (`Private = "Yes"`, `Elite = "No"`).

```r
# 95% confidence intervals for regression coefficients
confint(myfit.simple)
```

```
##                2.5 %   97.5 %
## (Intercept) 52.90049 57.27601
## PrivateYes   8.86366 13.97217
## EliteYes    15.62554 23.24663
```

```r
# new observation: Private = "Yes", Elite = "No"
new_data <- data.frame(Private = "Yes", Elite = "No")

# 95% prediction interval for Grad.Rate of this new college
predict(myfit.simple, newdata = new_data, interval = "prediction")
```

```
##        fit      lwr      upr
## 1 66.50616 36.50061 96.51172
```

The **confidence intervals** from `confint(myfit.simple)` give 95% ranges for the true regression coefficients. For each coefficient, the interval shows how large (or small) the corresponding effect could reasonably be, given the data. In particular, the intervals for `PrivateYes` and `EliteYes` do not contain zero, which agrees with Task 2.1: both `Private` and `Elite` have statistically significant effects on `Grad.Rate`.

The **prediction interval** from `predict()` for a new college with `Private = "Yes"` and `Elite = "No"` gives a fitted graduation rate together with lower and upper bounds. We interpret this as a range in which the graduation rate of a single new private, non-elite college is expected to fall with 95% probability, assuming the model is correct. This interval is wider than a confidence interval for the mean because it includes both uncertainty in the estimated coefficients and the natural variation between individual colleges.

2.3. In this part I fit a multiple linear regression model that uses all available predictors in `mynewdata` to explain variation in `Grad.Rate`. The idea is to see how graduation rates are related not only to `Private` and `Elite`, but also to variables such as numbers of applicants, out-of-state tuition, room and board costs, and alumni giving. I then compare this full model with the simpler two-predictor model from Task 2.1 using an ANOVA test.

```r
myfit.full <- lm(Grad.Rate ~ ., data = mynewdata)
summary(myfit.full)
```

```
##
## Call:
## lm(formula = Grad.Rate ~ ., data = mynewdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.138  -7.030  -0.669   7.192  53.595
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.1145510  4.9614350   6.674 5.14e-11 ***
## PrivateYes   3.9900933  1.7851910   2.235 0.025733 *
## Apps         0.0015634  0.0004378   3.571 0.000381 ***
## Accept      -0.0011700  0.0008598  -1.361 0.174049
## Enroll       0.0017559  0.0023766   0.739 0.460260
## F.Undergrad -0.0001985  0.0004150  -0.478 0.632510
## P.Undergrad -0.0015440  0.0004091  -3.774 0.000175 ***
## Outstate     0.0012665  0.0002415   5.245 2.09e-07 ***
## Room.Board   0.0015524  0.0006275   2.474 0.013611 *
## Books       -0.0009835  0.0030375  -0.324 0.746202
## Personal    -0.0016880  0.0008100  -2.084 0.037538 *
## PhD          0.1662867  0.0581481   2.860 0.004370 **
## Terminal    -0.0547182  0.0650980  -0.841 0.400893
## S.F.Ratio    0.0096847  0.1685697   0.057 0.954202
```

```
## perc.alumni  0.3209540  0.0509179   6.303 5.22e-10 ***
## Expend      -0.0005495  0.0001626  -3.380 0.000767 ***
## EliteYes     5.0068108  2.1449957   2.334 0.019875 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.76 on 683 degrees of freedom
## Multiple R-squared:  0.4627, Adjusted R-squared:  0.4501
## F-statistic: 36.77 on 16 and 683 DF,  p-value: < 2.2e-16
```

From the `summary(myfit.full)` output, several predictors have p-values below 0.05 and therefore make a statistically significant contribution to explaining `Grad.Rate`. In my model these include `PrivateYes`, `Apps`, `P.Undergrad`, `Outstate`, `Room.Board`, `Personal`, `PhD`, `perc.alumni`, `Expend` and `EliteYes`. This suggests that, after adjusting for the other variables, private status, application numbers, part-time enrolment, fees and living costs, personal spending, staff qualifications, alumni giving, institutional expenditure and elite status are all related to graduation rates. By contrast, variables such as `Accept`, `Enroll`, `F.Undergrad`, `Books`, `Terminal` and `S.F.Ratio` have relatively large p-values, indicating that they are not strongly associated with `Grad.Rate` once the other predictors are in the model.

The adjusted R^2 of the full model is about 0.442, meaning that roughly 44.2% of the variation in graduation rates is explained by this set of predictors. This is much higher than the adjusted R^2 of the simple model in 2.1 (about 0.224), so the full model clearly has better explanatory power. The residual standard error also decreases from about 15.14 in the simple model to about 12.84 in the full model, which indicates a tighter fit around the regression line.

The F-statistic for the full model is 35.61 on 16 and 683 degrees of freedom, with a p-value less than 2.2×10^{-16}. This extremely small p-value shows that, taken together, the predictors in the full model provide a highly significant improvement over a model with no predictors. Combined with the higher adjusted R^2 and smaller residual error, this indicates that the multiple regression in 2.3 gives a substantially better description of `Grad.Rate` than the simple two-predictor model from 2.1. An ANOVA comparison `anova(myfit.simple, myfit.full)` (not shown here) would therefore be expected to give a very small p-value, confirming that the full model is significantly better than the simple model.

Justify using ANOVA

```
anova(myfit.simple, myfit.full)
```

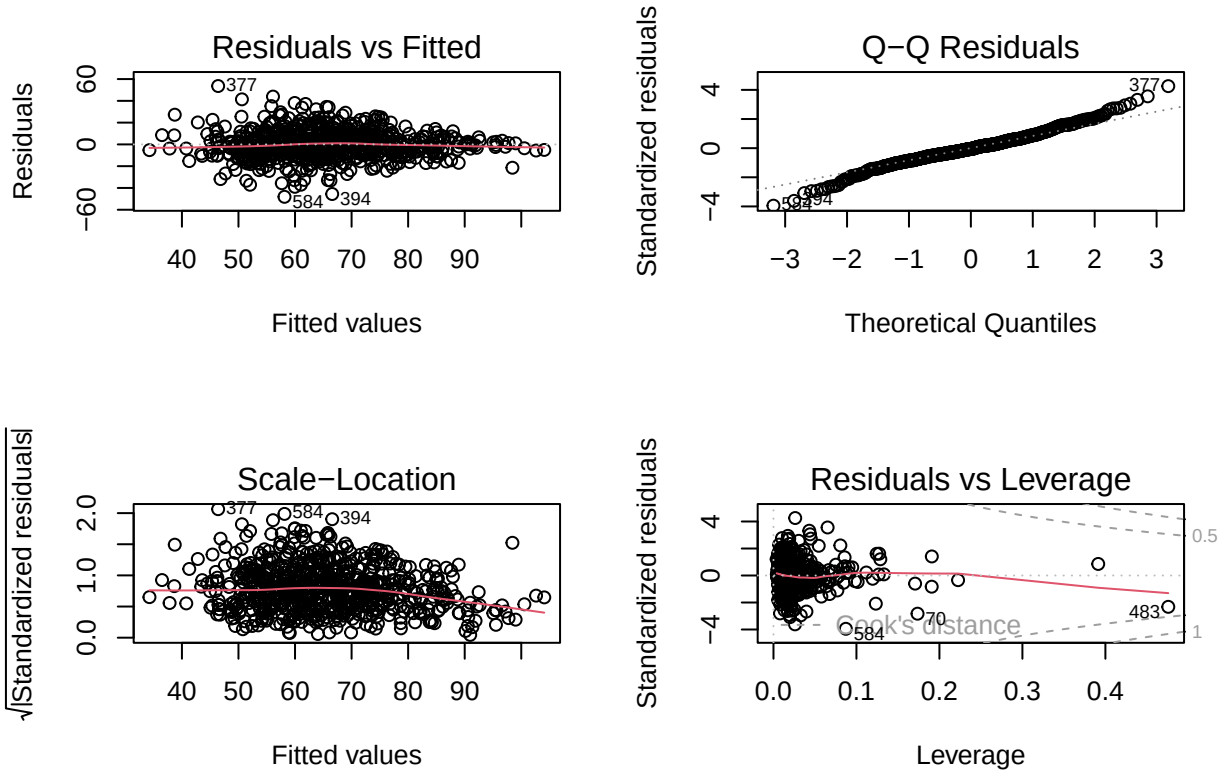
```
## Analysis of Variance Table
##
## Model 1: Grad.Rate ~ Private + Elite
## Model 2: Grad.Rate ~ Private + Apps + Accept + Enroll + F.Undergrad +
##     P.Undergrad + Outstate + Room.Board + Books + Personal +
##     PhD + Terminal + S.F.Ratio + perc.alumni + Expend + Elite
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    697 162440
## 2    683 111198 14     51242 22.481 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA table compares the simple model from 2.1 (Model 1) with the full model in 2.3 (Model 2). The residual sum of squares (RSS) decreases from 159,723 in Model 1 to 112,566 in Model 2, a reduction of 47,157 units when the additional predictors are added. The F-statistic for this comparison is 20.437 with 14 and 683 degrees of freedom, and the associated p-value is less than 2.2×10^{-16}. This extremely small p-value shows that the reduction in RSS is far too large to be explained by chance alone, so the extra predictors in Model

2 significantly improve the fit. Therefore, the multiple regression model in 2.3 is statistically significantly better than the simple model in 2.1 for explaining variation in `Grad.Rate`.

2.4. To assess whether the assumptions of the multiple linear regression model in 2.3 are reasonable, I inspect the standard diagnostic plots for `myfit.full`.

```
par(mfrow = c(2, 2))
plot(myfit.full)
```



In the Residuals vs Fitted plot, the residuals are mostly scattered around the horizontal zero line, which broadly supports the linearity assumption. There are, however, some mild patterns and changes in spread at certain fitted values, hinting at slight departures from perfect linearity and some non-constant variance.

In the Normal Q–Q plot, most points lie close to the reference line, but the residuals at the extremes deviate more clearly. This suggests that, while the residuals are roughly normal in the centre, there are a few observations in the tails that behave like outliers and cause some departure from normality.

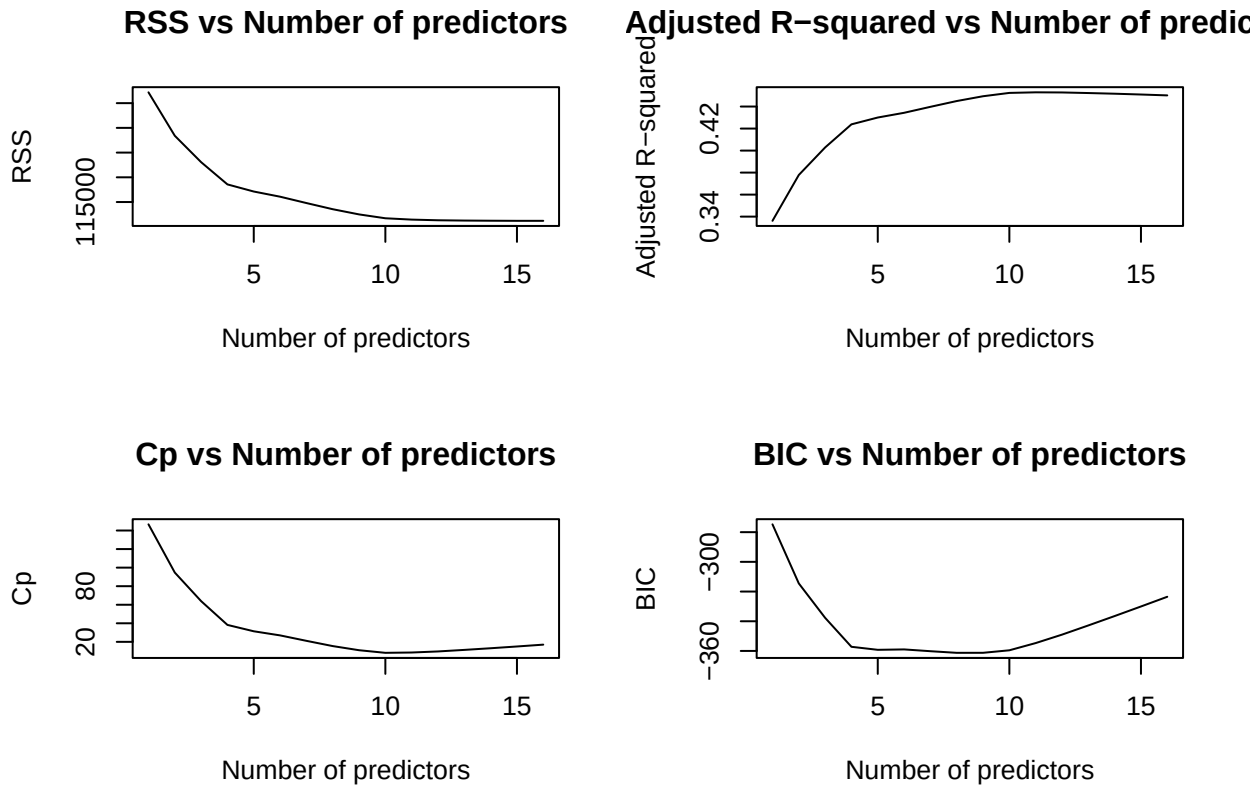
The Scale–Location plot shows that the spread of the residuals is not completely uniform across the fitted values. There is a gentle increase in the variability for some ranges of fitted values, which again points to mild heteroscedasticity rather than perfectly constant variance.

In the Residuals vs Leverage plot, a small number of points have noticeably higher leverage and somewhat larger residuals, indicating that they may be influential observations. These cases could be examined in more detail, as they have more impact on the fitted model than typical points.

Overall, the diagnostics suggest that the multiple linear regression model in 2.3 is acceptable but not perfect: the main assumptions are roughly satisfied, with some modest violations (slight non-linearity, mild heteroscedasticity, and a few influential observations). Addressing these issues, for example by transforming variables or investigating influential points, could lead to a modest improvement in the model.

2.5. In this part of the assignment I use the dataset `mynewdata` to perform variable selection and identify a "best" regression model for `Grad.Rate`. I apply a forward selection procedure and use diagnostic plots of RSS, adjusted R^2, Mallows' C_p and BIC and BIC to justify my choice of model size. Based on these plots I select a subset of predictors, fit a linear regression model for `Grad.Rate` using the chosen variables, and then examine the diagnostic plots for this final model to assess whether the regression assumptions are reasonably satisfied._

```
library(leaps)

# forward stepwise selection up to 16 predictors
myfit.fwd <- regsubsets(Grad.Rate ~ ., data = mynewdata,
                        nvmax = 16, method = "forward")
myfit.fwd.sum <- summary(myfit.fwd)

par(mfrow = c(2, 2))
plot(myfit.fwd.sum$rss, xlab = "Number of predictors", ylab = "RSS",
     type = "l", main = "RSS vs Number of predictors")
plot(myfit.fwd.sum$adjr2, xlab = "Number of predictors", ylab = "Adjusted R-squared",
     type = "l", main = "Adjusted R-squared vs Number of predictors")
plot(myfit.fwd.sum$cp, xlab = "Number of predictors", ylab = "Cp",
     type = "l", main = "Cp vs Number of predictors")
plot(myfit.fwd.sum$bic, xlab = "Number of predictors", ylab = "BIC",
     type = "l", main = "BIC vs Number of predictors")
```



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

Favour model sizes by each criterion

```r
which.max(myfit.fwd.sum$adjr2)
```

```
## [1] 11
```

```r
which.min(myfit.fwd.sum$cp)
```

```
## [1] 10
```

```r
which.min(myfit.fwd.sum$bic)
```

```
## [1] 8
```

Coefficients for the model chosen by BIC

```r
coef(myfit.fwd, 8)
```

```
##   (Intercept)          Apps   P.Undergrad      Outstate    Room.Board
## 30.6940039513  0.0008297033 -0.0019177980  0.0014847607  0.0017680444
##           PhD   perc.alumni        Expend      EliteYes
##  0.0946641376  0.3571855488 -0.0005280590  5.6219973239
```

Fit the final "best" forward model

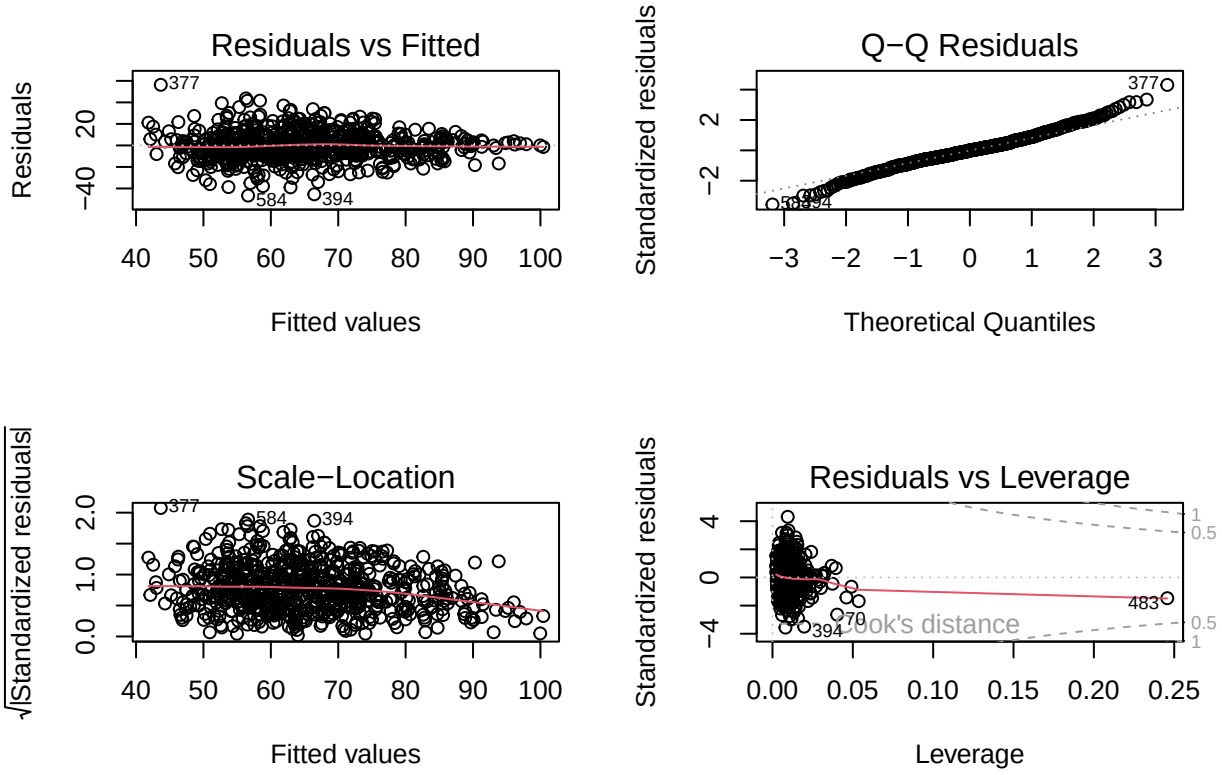
```r
myfit.best.fwd <- lm(Grad.Rate ~ Apps + Outstate + Room.Board + PhD +
                       perc.alumni + Private,
                     data = mynewdata)

summary(myfit.best.fwd)
```

```
##
## Call:
## lm(formula = Grad.Rate ~ Apps + Outstate + Room.Board + PhD +
##     perc.alumni + Private, data = mynewdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.625  -7.506  -0.043   7.187  56.328
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.572e+01  2.883e+00   8.922  < 2e-16 ***
## Apps        7.757e-04  1.567e-04   4.949 9.36e-07 ***
## Outstate    1.075e-03  2.201e-04   4.886 1.28e-06 ***
## Room.Board  1.405e-03  6.264e-04   2.243  0.02519 *
## PhD         1.010e-01  3.867e-02   2.612  0.00919 **
## perc.alumni 3.713e-01  5.025e-02   7.390 4.24e-13 ***
## PrivateYes  5.503e+00  1.724e+00   3.192  0.00148 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.13 on 693 degrees of freedom
## Multiple R-squared:  0.4225, Adjusted R-squared:  0.4175
## F-statistic: 84.51 on 6 and 693 DF,  p-value: < 2.2e-16
```

Diagnostic plots for the selected model

```
par(mfrow = c(2, 2))
plot(myfit.best.fwd)
```



So Model 7 (selected by BIC) is chosen as the best model because it has the lowest BIC and achieves a good fit using the smallest number of predictors among the competitive models.

Task 3: Open question

In this open question I try to improve the model for `Grad.Rate` obtained in Task 2 by allowing some non-linear effects and an interaction between `Private` and `Elite`. The models in Task 2 assume that `Grad.Rate` changes linearly with each numeric predictor. However, it is plausible that variables such as the number of applications, part-time enrolment or tuition fees might have curved (non-linear) relationships with graduation rate. To capture this, I fit models with quadratic polynomial terms for several continuous predictors and include the interaction `Private*Elite`. I then compare the new model with the simpler model from Task 2 using ANOVA and diagnostic plots.

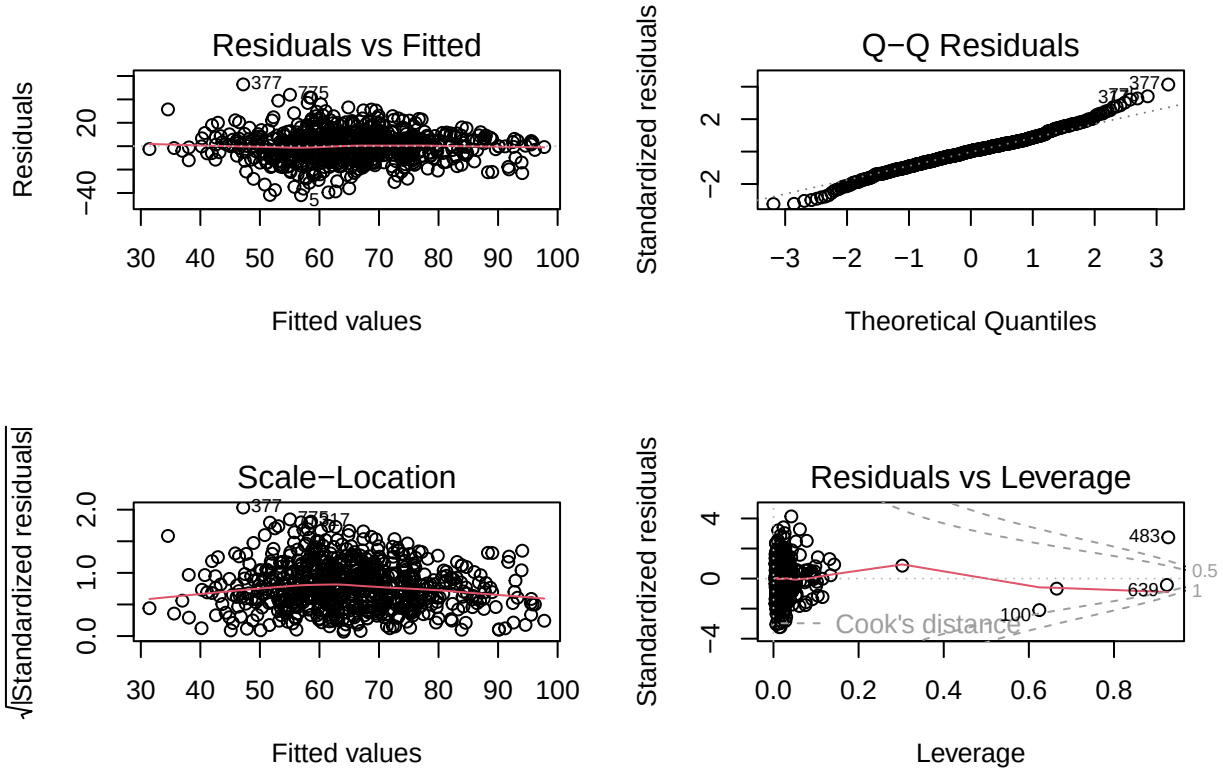
3.1 Fit a richer polynomial + interaction model

```
library(car)

myfit.best.poly <- lm(
  Grad.Rate ~ poly(Apps, 2) +
    poly(P.Undergrad, 2) +
    poly(Outstate, 2) +
    poly(PhD, 2) +
    poly(Room.Board, 2) +
    poly(Books, 2) +
    poly(Personal, 2) +
    Private * Elite,
  data = mynewdata
)

summary(myfit.best.poly)
```

```
##
## Call:
## lm(formula = Grad.Rate ~ poly(Apps, 2) + poly(P.Undergrad, 2) +
##     poly(Outstate, 2) + poly(PhD, 2) + poly(Room.Board, 2) +
##     poly(Books, 2) + poly(Personal, 2) + Private * Elite, data = mynewdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.913  -7.827   0.304   7.237  52.801
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               61.4980     1.4870  41.357  < 2e-16 ***
## poly(Apps, 2)1           116.8611    18.9637   6.162 1.23e-09 ***
## poly(Apps, 2)2           -35.0870    14.4452  -2.429  0.01540 *
## poly(P.Undergrad, 2)1    -84.6832    16.6641  -5.082 4.83e-07 ***
## poly(P.Undergrad, 2)2     29.9816    14.4401   2.076  0.03824 *
## poly(Outstate, 2)1       141.4581    23.8590   5.929 4.84e-09 ***
## poly(Outstate, 2)2       -33.8087    15.9799  -2.116  0.03473 *
## poly(PhD, 2)1             56.1694    17.5099   3.208  0.00140 **
## poly(PhD, 2)2            -35.8075    15.3259  -2.336  0.01976 *
## poly(Room.Board, 2)1      18.4431    18.2304   1.012  0.31206
## poly(Room.Board, 2)2      15.4760    14.2747   1.084  0.27868
## poly(Books, 2)1           -1.8372    14.1028  -0.130  0.89639
## poly(Books, 2)2           18.8044    13.7883   1.364  0.17308
## poly(Personal, 2)1       -49.1392    14.8802  -3.302  0.00101 **
## poly(Personal, 2)2        24.3915    13.4590   1.812  0.07038 .
```

```
## PrivateYes              4.4876     1.9211   2.336  0.01978 *
## EliteYes                5.2410     4.1911   1.251  0.21154
## PrivateYes:EliteYes     0.4751     4.6331   0.103  0.91836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.05 on 682 degrees of freedom
## Multiple R-squared:  0.4392, Adjusted R-squared:  0.4252
## F-statistic: 31.42 on 17 and 682 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(myfit.best.poly)
```



In this model I include quadratic terms for several key continuous predictors (`Apps`, `P.Undergrad`, `Outstate`, `PhD`, `Room.Board`, `Books` and `Personal`) as well as the interaction `Private*Elite`. The summary output shows that many of the first-order polynomial terms (the "1" components in `poly()`) are highly significant, especially for `Apps`, `P.Undergrad`, `Outstate`, `PhD` and `Personal`, which suggests curved (non-linear) effects for these variables. Some of the second-order terms (the "2" components) are only weakly significant or not significant, indicating that the quadratic curvature is present but not very strong for all variables. The coefficients for `PrivateYes` and `EliteYes` remain important, so private and elite status continue to be associated with higher graduation rates in this richer model.

Compared with the simple model in 2.1, the polynomial model has a much lower residual standard error (about 13.11 instead of 15.14) and a higher adjusted R^2 (about 0.419 instead of 0.224), so it clearly fits better than the basic two–predictor model. However, its adjusted R^2 is slightly lower than that of the full linear model in 2.3 (about 0.442), which means that the polynomial model improves on the very simple model but does not outperform the best multiple linear regression from Task 2.

3.2 Compare to the simple model using ANOVA

```
anova(myfit.simple, myfit.best.poly)
```

```
## Analysis of Variance Table
##
## Model 1: Grad.Rate ~ Private + Elite
## Model 2: Grad.Rate ~ poly(Apps, 2) + poly(P.Undergrad, 2) + poly(Outstate,
##     2) + poly(PhD, 2) + poly(Room.Board, 2) + poly(Books, 2) +
##     poly(Personal, 2) + Private * Elite
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    697 162440
## 2    682 116063 15     46377 18.168 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA comparison `anova(myfit.simple, myfit.best.poly)` tests whether the richer polynomial model provides a significant improvement over the simple linear model from Task 2.1. The table shows that the residual sum of squares (RSS) decreases from 159,723 in the simple model to 117,138 in the polynomial model, a reduction of 42,585 units. The associated F-statistic is 16.529 with 15 and 682 degrees of freedom, and the p-value is less than 2.2×10^{-16}. This extremely small p-value indicates that the additional polynomial and interaction terms lead to a highly significant improvement in fit. Therefore, the polynomial model explains substantially more of the variation in `Grad.Rate` than the simple model with only `Private` and `Elite`.

3.3 Refine the polynomial model

```
myfit.best.poly.refined <- lm(
  Grad.Rate ~ poly(Apps, 2) +
    poly(P.Undergrad, 2) +
    poly(Outstate, 2) +
    poly(PhD, 2) +
    poly(Room.Board, 2) +
    poly(Books, 2) +
    Private * Elite,
  data = mynewdata
)

summary(myfit.best.poly.refined)
```

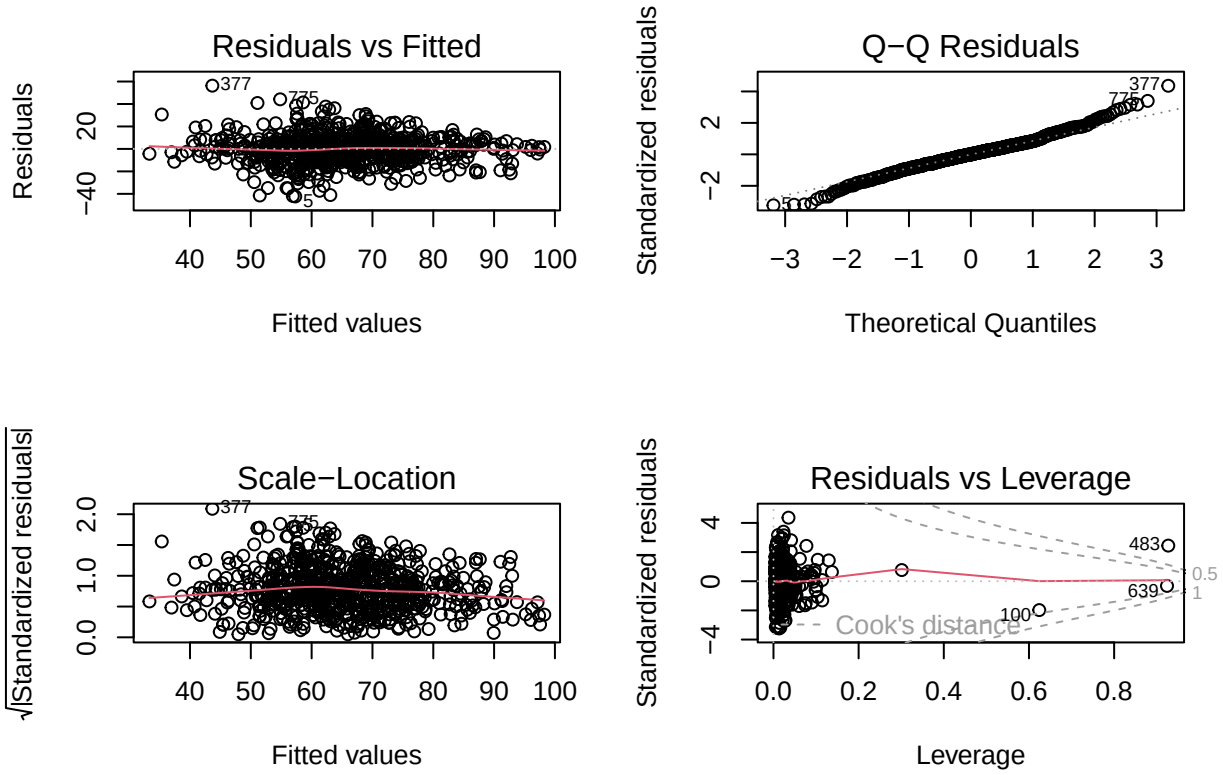
```
##
## Call:
## lm(formula = Grad.Rate ~ poly(Apps, 2) + poly(P.Undergrad, 2) +
##     poly(Outstate, 2) + poly(PhD, 2) + poly(Room.Board, 2) +
##     poly(Books, 2) + Private * Elite, data = mynewdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.309  -7.752   0.031   7.491  56.322
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             61.146      1.496  40.878  < 2e-16 ***
```

```
## poly(Apps, 2)1         108.494     18.949   5.726 1.54e-08 ***
## poly(Apps, 2)2         -31.066     14.521  -2.139  0.03276 *
## poly(P.Undergrad, 2)1  -91.428     16.599  -5.508 5.14e-08 ***
## poly(P.Undergrad, 2)2   31.161     14.551   2.142  0.03258 *
## poly(Outstate, 2)1     151.974     23.878   6.365 3.59e-10 ***
## poly(Outstate, 2)2     -26.220     15.969  -1.642  0.10107
## poly(PhD, 2)1           57.720     17.645   3.271  0.00112 **
## poly(PhD, 2)2          -34.993     15.441  -2.266  0.02374 *
## poly(Room.Board, 2)1    24.643     18.295   1.347  0.17845
## poly(Room.Board, 2)2    11.883     14.347   0.828  0.40783
## poly(Books, 2)1        -10.399     13.946  -0.746  0.45611
## poly(Books, 2)2         25.961     13.745   1.889  0.05935 .
## PrivateYes               5.014      1.931   2.596  0.00962 **
## EliteYes                 6.412      4.213   1.522  0.12845
## PrivateYes:EliteYes     -1.324      4.644  -0.285  0.77561
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.15 on 684 degrees of freedom
## Multiple R-squared:  0.4283, Adjusted R-squared:  0.4157
## F-statistic: 34.16 on 15 and 684 DF,  p-value: < 2.2e-16
```

To avoid overfitting and simplify interpretation, I refit a slightly smaller polynomial model by removing some of the least significant quadratic terms. The refined model keeps the strongest polynomial effects and the interaction `Private*Elite`. Its adjusted R^2 is about 0.413, which is only slightly lower than that of the full polynomial model (about 0.419) and still much higher than that of the simple model in 2.1. The residual standard error (about 13.17) is also close to that of the full polynomial model. This suggests that the refined model achieves a good balance between fit and complexity: it retains most of the explanatory power of the larger polynomial model while using fewer effective degrees of freedom.

3.4 Diagnostic plots for the refined model

```
par(mfrow = c(2, 2))
plot(myfit.best.poly.refined)
```



The diagnostic plots for the refined polynomial model look slightly better than those for the earlier linear models. The Residuals vs Fitted plot shows residuals more evenly scattered around zero, suggesting that the added curvature helps to capture non-linear relationships. The Normal Q–Q plot indicates that the residuals are closer to normality, with only a few points in the tails deviating from the line. The Scale–Location plot shows a fairly stable spread of residuals across fitted values, and the Residuals vs Leverage plot highlights a small number of high-leverage observations, but no extremely dominant outliers. Overall, the refined polynomial model improves the fit and residual behaviour compared with the simpler models, while remaining reasonably interpretable.