Advanced Methods for Regression and Classification

Exercise 1

Bosse Behrens , st.id: 12347333

Task 1

```
library("dplyr")
library("ISLR")
```

```
data(College,package="ISLR")
?College
```

```
## starte den http Server für die Hilfe fertig
```

```
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 ...
```

```
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
```

```
sum(is.na(College))
```

```
## [1] 0
```

There are no missing values in the College data but for the "Texas A&M University at Galveston" the value for PhD (Pct. of faculty with Ph.D.'s) is at 103, which is obviously not a realistic possible percentage. For "Cazenovia College" the Graduation rate is at 118, which is also not possible. Therefore we delete these two observations since we don't know if it's just a typo and both values should be 100, or the data for the whole observation is corrupted. Also for "Private" we have values "yes" and "no" which we have to convert into a numeric varible for regression. We will use a dummy variable with 0 for "no" and 1 for "yes". Furthermore we want to transform very skewed variables such as "Apps". Looking at the summary, we see for the column "Apps" the max value is way more than 20 times the median value. This means there are some very large outliers that can make the distribution of the attribute very right-skewed which could lead to more variance since the large outliers have a great influence. therefore it makes sense to log-transform "Apps".

```
College <- College[!rownames(College)
         %in% c("Texas A&M University at Galveston", "Cazenovia College"), ]
College <- College %>%
  mutate(Private = recode(Private, "Yes" = 1, "No" = 0))
College_clean <- College %>%
  mutate(
    log_Apps = log(Apps))%>%
  select(-Apps)
```

Task 2

a)

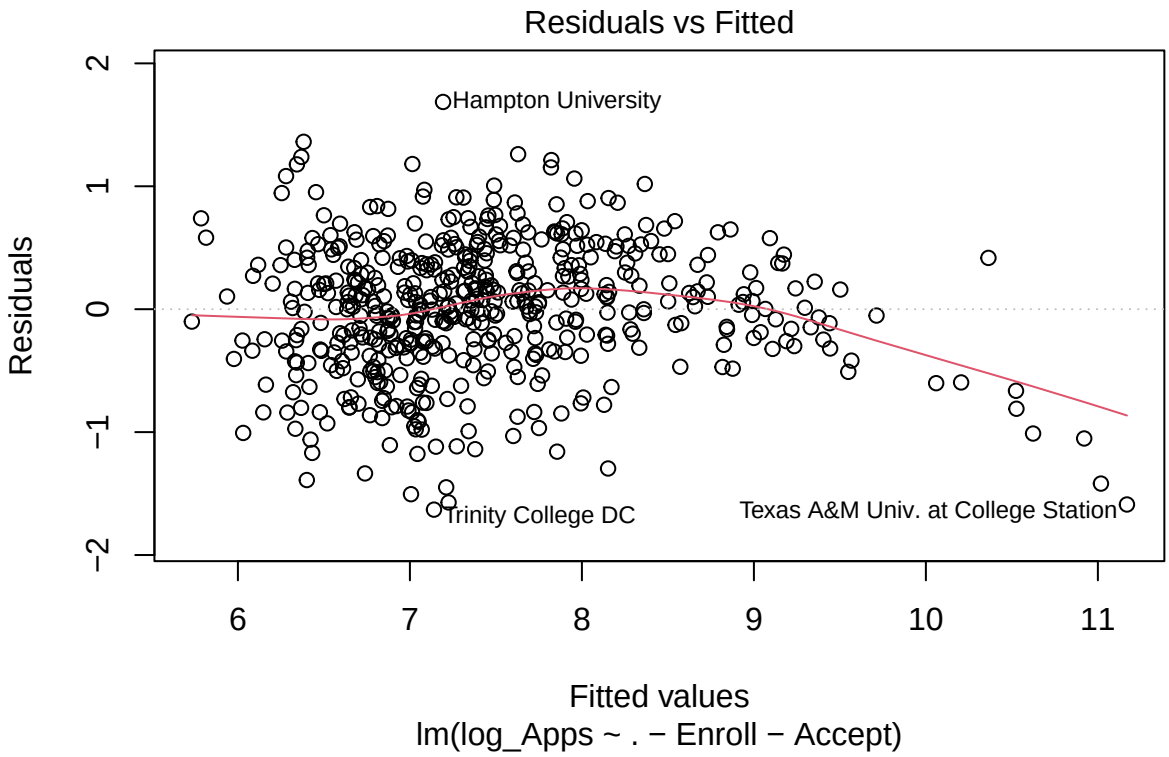
```
set.seed(12347333)
n <- nrow(College_clean)
train <- sample(1:n, round((2/3) * n))
test<-(1:n)[-train]
model_lm <- lm(log_Apps ~ . - Enroll - Accept, data = College_clean,
               subset = train)
```

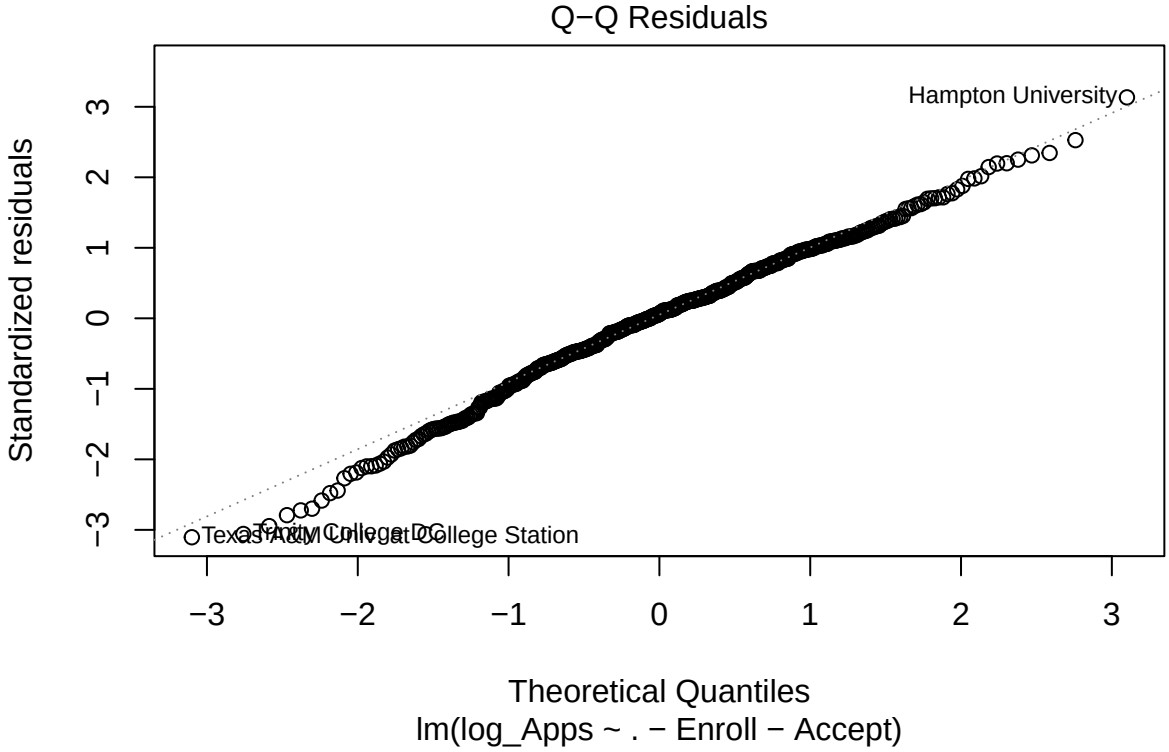
Summarizing and plotting residuals of fitted model.

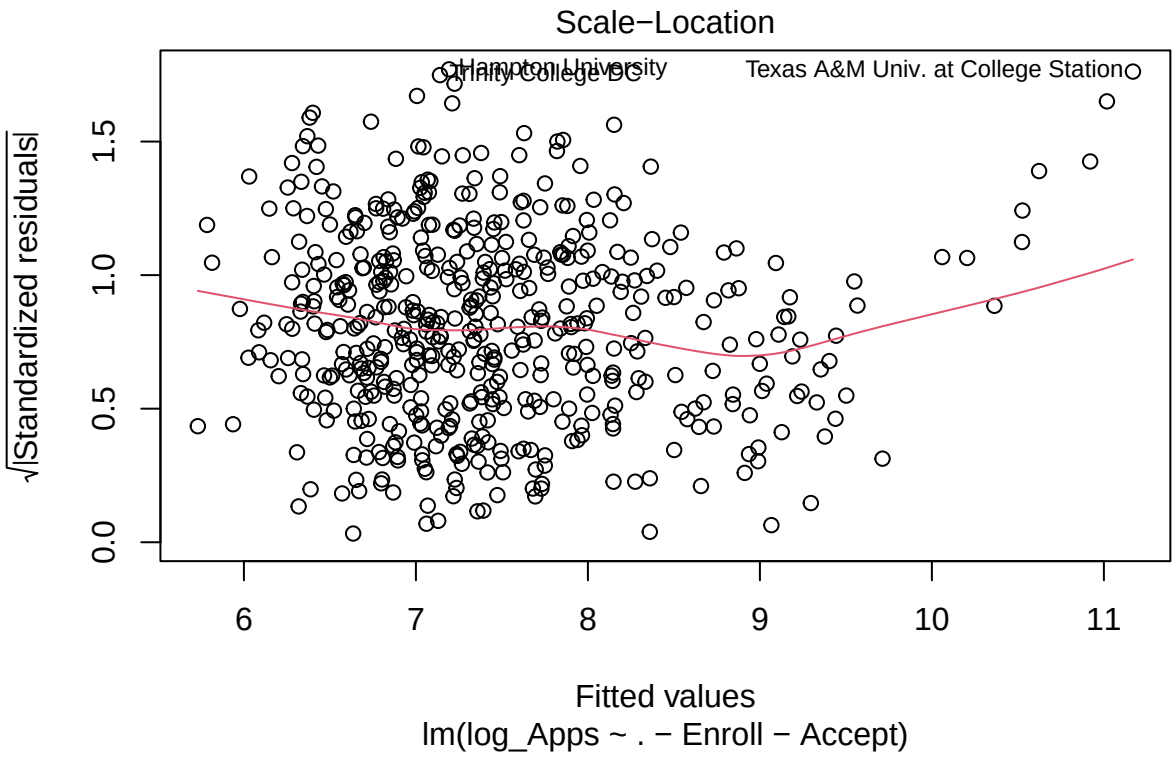
```
summary(model_lm)
```

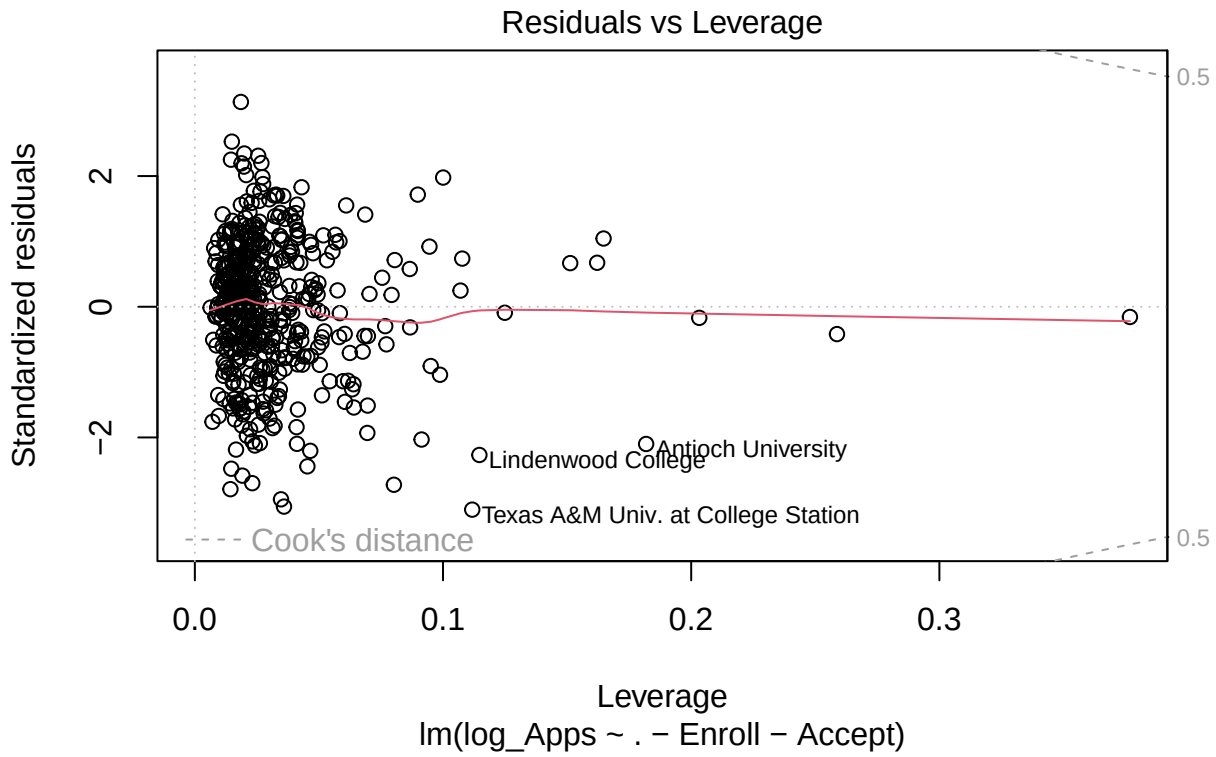
```
##
## Call:
## lm(formula = log_Apps ~ . - Enroll - Accept, data = College_clean,
##     subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63062 -0.31976  0.03628  0.37212  1.68602
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.176e+00  2.644e-01  15.791  < 2e-16 ***
## Private     -5.332e-01  8.861e-02  -6.017 3.43e-09 ***
## Top10perc   -1.496e-03  3.641e-03  -0.411 0.681334
## Top25perc    5.067e-03  2.882e-03   1.758 0.079374 .
## F.Undergrad  1.066e-04  7.688e-06  13.867  < 2e-16 ***
## P.Undergrad  1.676e-05  1.779e-05   0.942 0.346412
## Outstate     4.866e-05  1.240e-05   3.924 9.93e-05 ***
## Room.Board   6.957e-05  3.144e-05   2.213 0.027378 *
## Books        2.195e-04  1.422e-04   1.544 0.123337
## Personal     1.631e-05  3.945e-05   0.413 0.679534
## PhD          7.668e-03  2.905e-03   2.640 0.008555 **
## Terminal     7.859e-04  3.134e-03   0.251 0.802124
## S.F.Ratio    5.015e-02  9.119e-03   5.499 6.09e-08 ***
## perc.alumni -9.278e-03  2.639e-03  -3.516 0.000477 ***
## Expend       3.055e-05  7.121e-06   4.290 2.15e-05 ***
## Grad.Rate    9.560e-03  1.923e-03   4.971 9.18e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.543 on 501 degrees of freedom
## Multiple R-squared:  0.7325, Adjusted R-squared:  0.7245
## F-statistic: 91.44 on 15 and 501 DF,  p-value: < 2.2e-16
```

```
plot(model_lm)
```


Residuals vs Fitted
Hampton University
Trinity College DC
Texas A&M Univ. at College Station
Residuals
Fitted values
lm(log_Apps ~ . − Enroll − Accept)


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(log_Apps ~ . − Enroll − Accept)
Hampton University
Texas A&M Univ at College Station
Trinity College DC
−3
−2
−1
0
1
2
3


Scale−Location
Hampton University
Trinity College DC
Texas A&M Univ. at College Station
√|Standardized residuals|
0.0
0.5
1.0
1.5
6
7
8
9
10
11
Fitted values
lm(log_Apps ~ . − Enroll − Accept)



The test of significance for the variable sin the models yields that Intercept, Private, F.Undergrad, Outstate, S.F.Ratio, perc.alumni, Expend and Grad.Rate all are very significant with a p-value of < 0.001. Additionally PhD is significant on a level of < 0.01, Room.Board on < 0.05 and Top25perc on < 0.1. The model assumptions are partly fulfilled but not perfectly so. In the residuals vs fitted values plot the residuals seem mostly randomly distributed around zero but have a slight curvature down for the higher fitted values. This could implicate some linear dependence between some of the variables. The qq-plot shows the residuals are not quite normally distributed since the line is curves down for the more negative residuals. The scale-location plot shows again the residuals are mostly homoscedastic but curves slightly upwards for the higher fitted values. This means variance is not perfectly the same for all levels of fitted values. The residuals vs leverage plot shows there are some observation that are clearly identifiable from the rest but still within limits of Cook's distance.

b)

Getting the design matrix and calculating the estimators.

```
train_data <- College_clean[train, ] #model.matrix has no subset functionality
X <- model.matrix(log_Apps ~ . - Enroll - Accept, data = train_data)
y <- train_data[ ,"log_Apps"]
ls_estimators <- solve(t(X)%*%X)%*%t(X)%*%y
```

Comparing the results to the lm() model.

```
lm_model_estimators <- coef(model_lm)
df_estimators <- data.frame(
  ls_estimators = ls_estimators,
  lm_model_estimators = lm_model_estimators,
  absolute_difference = abs(ls_estimators - lm_model_estimators))
```

```r
print(df_estimators)
```

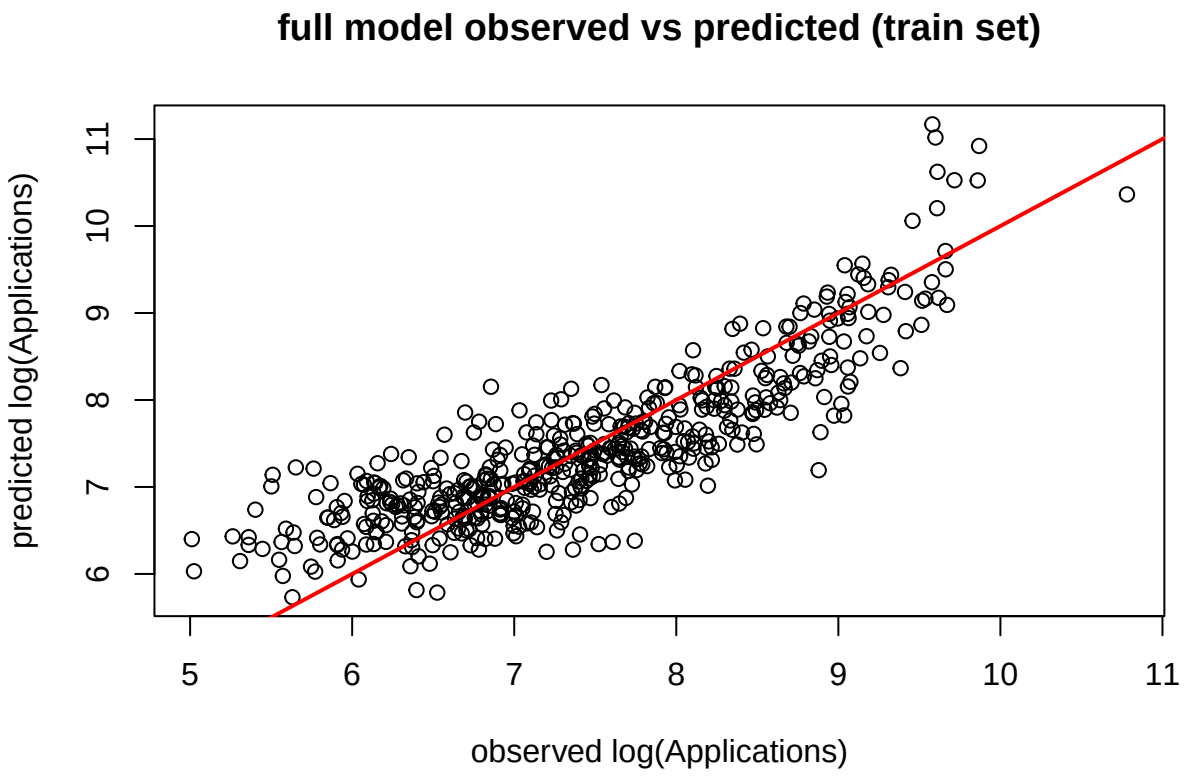
```
##             ls_estimators lm_model_estimators absolute_difference
## (Intercept)  4.175665e+00        4.175665e+00        8.704149e-14
## Private     -5.331641e-01       -5.331641e-01        6.551426e-13
## Top10perc   -1.496228e-03       -1.496228e-03        2.690318e-14
## Top25perc    5.066767e-03        5.066767e-03        2.156608e-14
## F.Undergrad  1.066148e-04        1.066148e-04        2.875846e-17
## P.Undergrad  1.676234e-05        1.676234e-05        2.838916e-17
## Outstate     4.865798e-05        4.865798e-05        3.554828e-17
## Room.Board   6.957279e-05        6.957279e-05        1.897354e-17
## Books        2.195331e-04        2.195331e-04        1.694337e-16
## Personal     1.630742e-05        1.630742e-05        1.521949e-17
## PhD          7.668322e-03        7.668322e-03        1.116121e-14
## Terminal     7.858566e-04        7.858566e-04        1.227631e-14
## S.F.Ratio    5.014991e-02        5.014991e-02        1.431494e-14
## perc.alumni -9.278385e-03       -9.278385e-03        2.069525e-15
## Expend       3.054856e-05        3.054856e-05        6.288373e-18
## Grad.Rate    9.559997e-03        9.559997e-03        3.764350e-15
```

Reading up on the documentation, R is handling binary variables by automatically transforming them into values 0 and 1 alphabetically (which we already did). The coefficient can be interpreted as the change of the predicted value if the binary variable is "yes" ($= 1$) since it will then add $1 \cdot \beta_{binary}$ to the predicted value, otherwise 0.
Comparing the results we can observe they are not exactly the same but within a very small tolerance ranges between 10^{-13} and 10^{-18}. This is probably due to the inner calculations and machine precision.

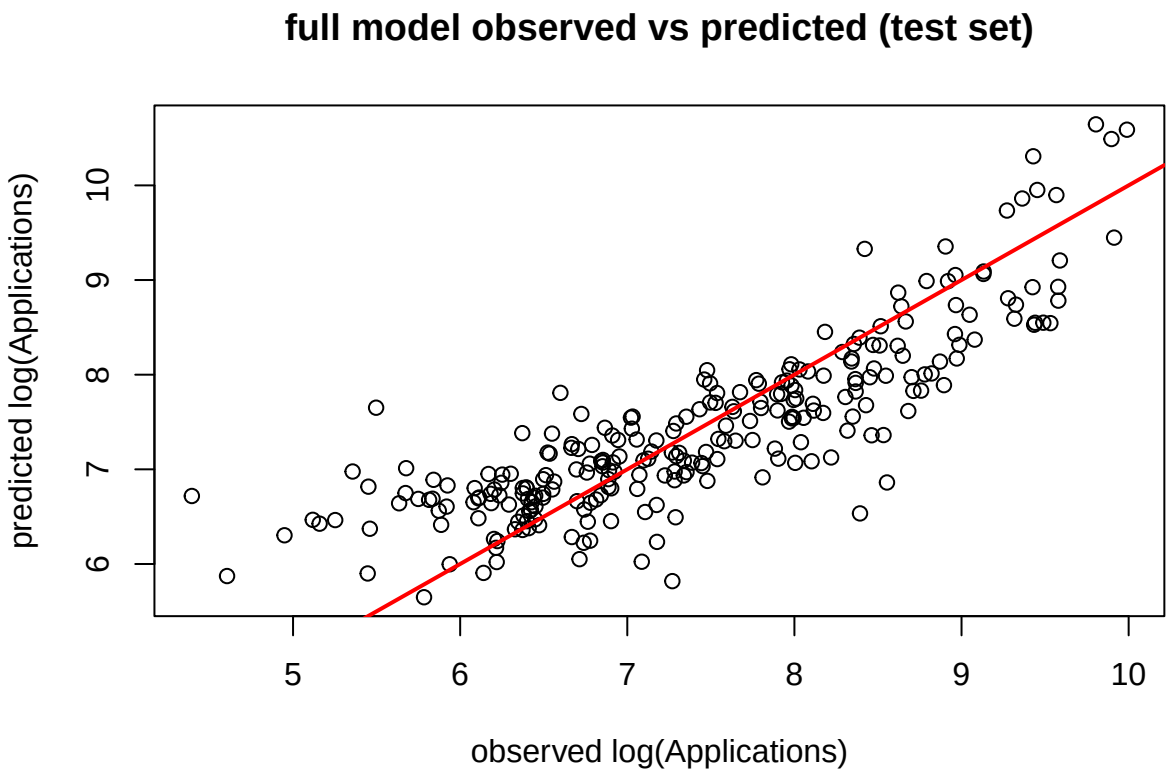
c)

```r
pred_lm_train <- predict(model_lm, newdata = College_clean[train, ])
pred_lm_test <- predict(model_lm, newdata = College_clean[test, ])
observed_values_train <- College_clean[train,"log_Apps"]
observed_values_test <- College_clean[test,"log_Apps"]

plot(observed_values_train, pred_lm_train,
     main = "full model observed vs predicted (train set)",
     xlab = "observed log(Applications)",
     ylab = "predicted log(Applications)")
abline(0, 1, col = "red", lwd = 2)
```




```
plot(observed_values_test, pred_lm_test,
     main = "full model observed vs predicted (test set)",
     xlab = "observed log(Applications)",
     ylab = "predicted log(Applications)")
abline(0, 1, col = "red", lwd = 2)
```



The prediction performance of the model and its predictions are okay but not perfect. Looking at the plots both for the training and testing data a more fitting curve through the data points plotted (observations vs predictions) would be slightly curves. The points do not seem totally randomly distributed around the $x = y$ line. Especially on the ends of the range of the values the points for very small and large values seem to differ. This implicates that one or more of the model assumptions is not completely fulfilled.

d)

```
n_train <- length(observed_values_train)
n_test <- length(observed_values_test)
RMSE_train <- sqrt((1/n_train)*sum((observed_values_train - pred_lm_train)^2))
RMSE_test <- sqrt((1/n_test)*sum((observed_values_test - pred_lm_test)^2))
print(RMSE_train)
```

```
## [1] 0.5345712
```

```
print(RMSE_test)
```

```
## [1] 0.6148493
```

The RSME values are 0.5345712 for the training set and 0.6148493 for the testing set. Since the models was fitted on the training data this is to be expected. The log-transformed values for "Apps" ranges in values form around 5 to 9. The RMSEs values seem therefore alright and that the models fits fairly acurate enough even if not perfect.

Task 3

a)

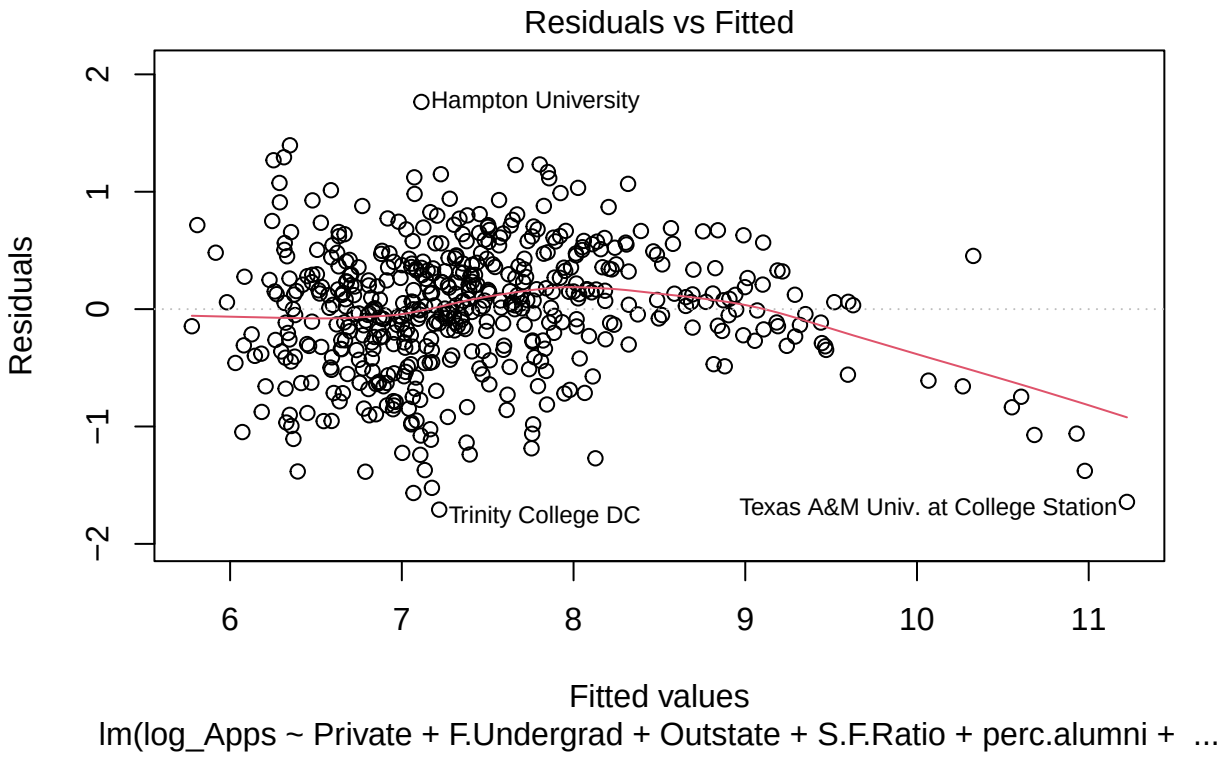
The variables that had p-values of < 0.05 (except intercept) were Private, F.Undergrad, Outstate, S.F.Ratio, perc.alumni, Expend, and Grad.Rate, PhD and Room.Board.

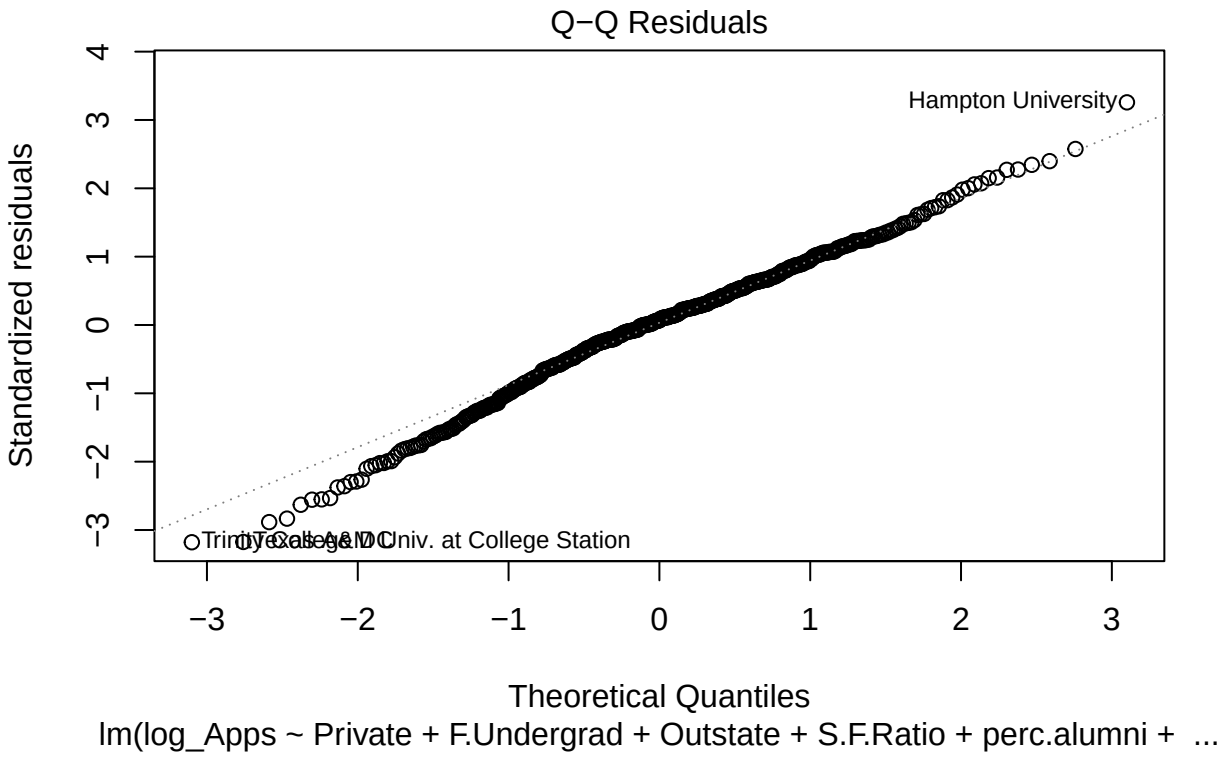
```
model_lm_red <- lm(log_Apps ~ Private + F.Undergrad + Outstate + S.F.Ratio +
                     perc.alumni + Expend + Grad.Rate + PhD + Room.Board,
                   data = College_clean, subset = train)
```

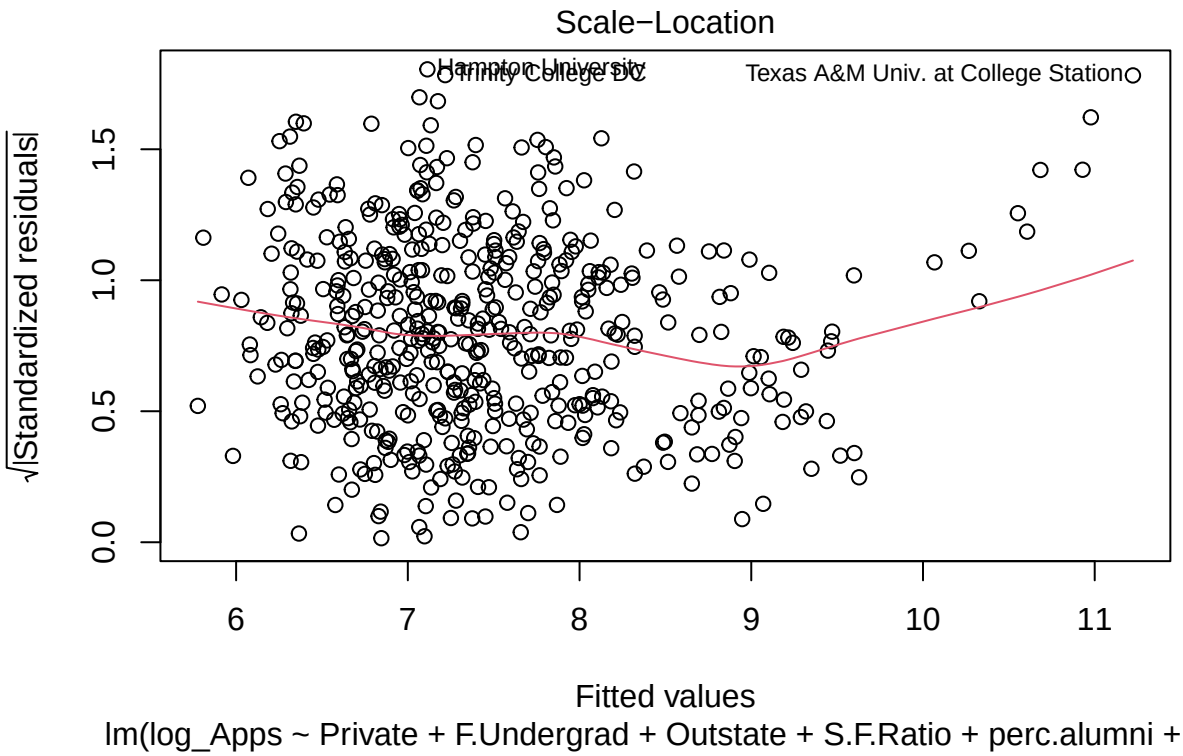
```
summary(model_lm_red)
```

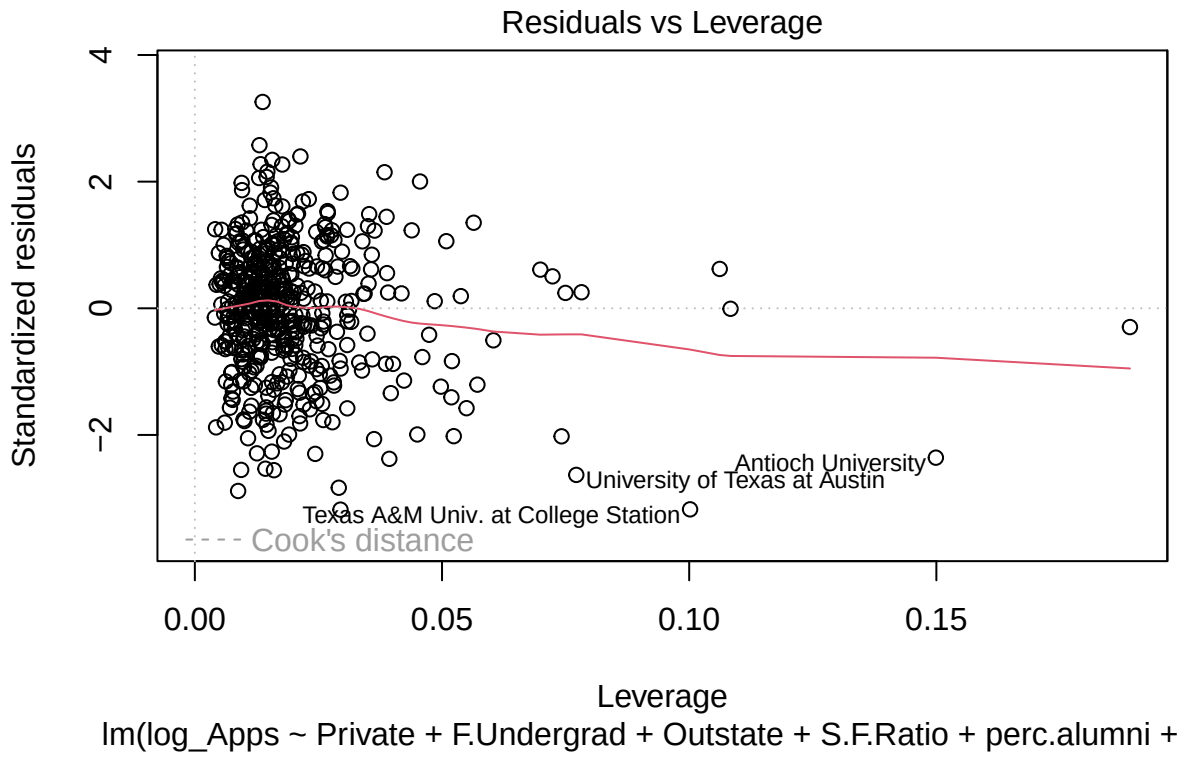
```
##
## Call:
## lm(formula = log_Apps ~ Private + F.Undergrad + Outstate + S.F.Ratio +
##     perc.alumni + Expend + Grad.Rate + PhD + Room.Board, data = College_clean,
##     subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.70827 -0.31361  0.03717  0.35135  1.76484
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.420e+00  2.232e-01  19.802  < 2e-16 ***
## Private     -5.362e-01  8.835e-02  -6.069 2.52e-09 ***
## F.Undergrad  1.148e-04  6.723e-06  17.080  < 2e-16 ***
## Outstate     5.052e-05  1.229e-05   4.112 4.57e-05 ***
## S.F.Ratio    4.877e-02  9.131e-03   5.341 1.40e-07 ***
## perc.alumni -8.732e-03  2.606e-03  -3.351 0.000864 ***
## Expend       3.265e-05  6.503e-06   5.020 7.15e-07 ***
## Grad.Rate    9.975e-03  1.801e-03   5.540 4.87e-08 ***
## PhD          9.184e-03  1.931e-03   4.757 2.57e-06 ***
## Room.Board   7.465e-05  3.042e-05   2.454 0.014472 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5454 on 507 degrees of freedom
## Multiple R-squared:  0.7269, Adjusted R-squared:  0.722
## F-statistic: 149.9 on 9 and 507 DF,  p-value: < 2.2e-16
```

```
plot(model_lm_red)
```


Residuals vs Fitted
Residuals
Fitted values
lm(log_Apps ~ Private + F.Undergrad + Outstate + S.F.Ratio + perc.alumni + ...
Hampton University
Trinity College DC
Texas A&M Univ. at College Station
2
1
0
−1
−2
6
7
8
9
10
11


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(log_Apps ~ Private + F.Undergrad + Outstate + S.F.Ratio + perc.alumni + ...
Hampton University
Texas A&M Univ. at College Station
Trinity College CT


Scale−Location
Hampton University
Trinity College DC
Texas A&M Univ. at College Station
√|Standardized residuals|
0.0
0.5
1.0
1.5
6
7
8
9
10
11
Fitted values
lm(log_Apps ~ Private + F.Undergrad + Outstate + S.F.Ratio + perc.alumni + ...

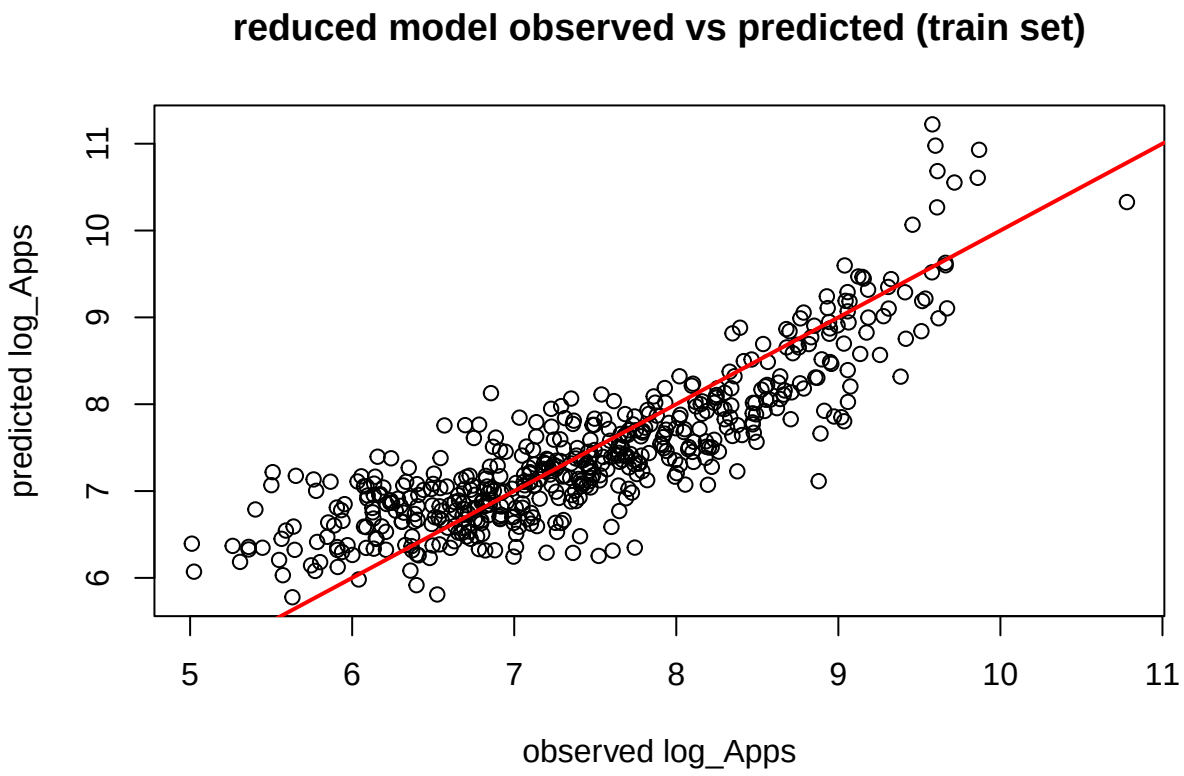


Now all predictors have a p-value of < 0.05 and are thereby statistically significant at the 5%-level. In general, this should not be expected. Reasons can be collinearity in the predictors due to which the model might not be able to get the individual effects. Also some predictors might first be marked as significant due to overfitting (training data fits by chance very well) or random variation in the data, but don't have a true effect on the predictions. After reducing the model they might not be calculated as significant anymore.

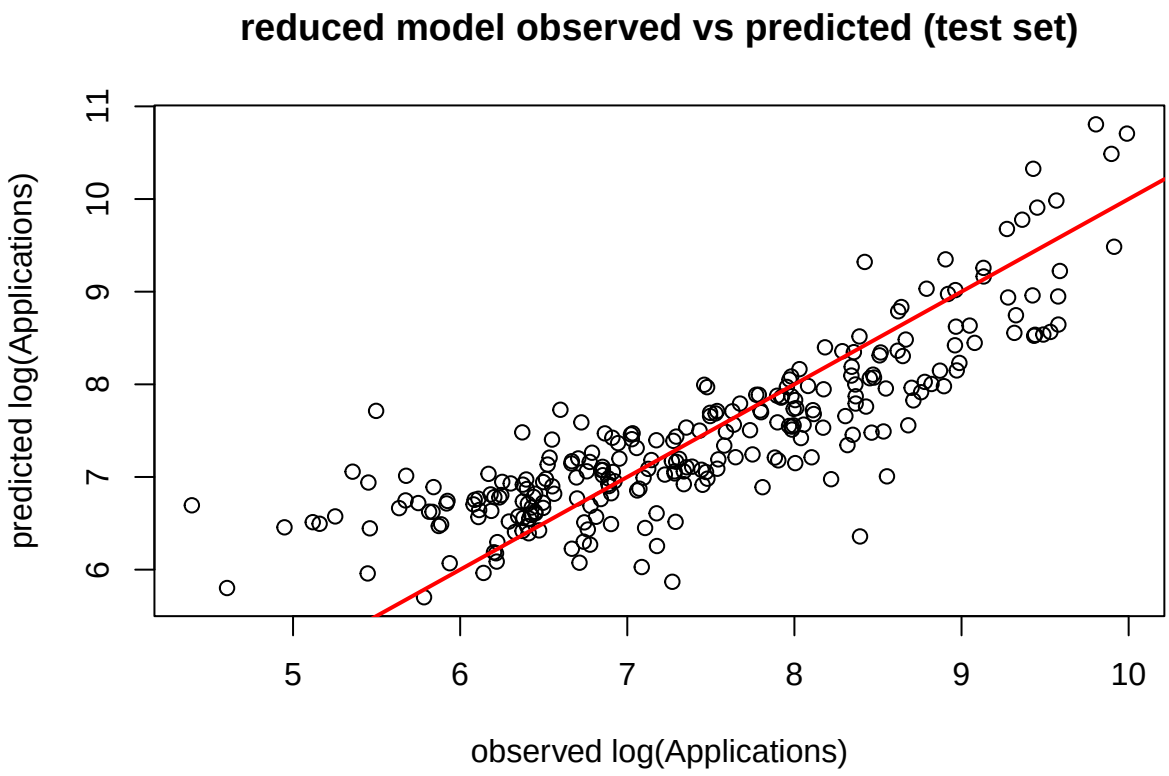
b)

```
pred_lm_train_red <- predict(model_lm_red, newdata = College_clean[train, ])
pred_lm_test_red <- predict(model_lm_red, newdata = College_clean[test, ])
observed_values_train <- College_clean[train,"log_Apps"]
observed_values_test <- College_clean[test,"log_Apps"]

plot(observed_values_train, pred_lm_train_red,
     main = "reduced model observed vs predicted (train set)",
     xlab = "observed log_Apps",
     ylab = "predicted log_Apps")
abline(0, 1, col = "red", lwd = 2)
```




```
plot(observed_values_test, pred_lm_test_red,
      main = "reduced model observed vs predicted (test set)",
     xlab = "observed log(Applications)",
     ylab = "predicted log(Applications)")
abline(0, 1, col = "red", lwd = 2)
```



c)

```
RMSE_train_red <- sqrt((1/n_train)*sum((observed_values_train -
                                          pred_lm_train_red)^2))
RMSE_test_red <- sqrt((1/n_test)*sum((observed_values_test -
                                        pred_lm_test_red)^2))
print(RMSE_train_red)
```

```
## [1] 0.540125
```

```
print(RMSE_test_red)
```

```
## [1] 0.6217448
```

It could have been expected that when only using the previously as significant marked variables for the regression the regression itself will be far better and way less random noise be caught by omitting the insignificant ones. The results of the reduced regression show this is not true. They look very similar to the full regression on all predictors with almost the same plots and RSME values for both training and testing data.

d)

```
anova(model_lm_red, model_lm)
```

```
## Analysis of Variance Table
##
## Model 1: log_Apps ~ Private + F.Undergrad + Outstate + S.F.Ratio + perc.alumni +
```

```
##     Expend + Grad.Rate + PhD + Room.Board
## Model 2: log_Apps ~ (Private + Accept + Enroll + Top10perc + Top25perc +
##     F.Undergrad + P.Undergrad + Outstate + Room.Board + Books +
##     Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend +
##     Grad.Rate) - Enroll - Accept
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1    507 150.83
## 2    501 147.74  6    3.0858 1.744 0.1089
```

The RSS for both models has only a very small difference. Also the F-statistic is relatively low and the p-value for the F-test is > 0.1. This means there cannot be a significantly improved fitting of the model be observed by using the full model with more predictors. Therefore the smaller model should be used if at all, because there is no significant difference and the model should be held as small as it needs to be.

Task 4

```
model_lm_step_back <- step(model_lm)
```

```
empty_model <- lm(log_Apps ~ 1, data = College_clean[train, ])
model_lm_step_forward <- step(empty_model,
                              scope = list(lower = empty_model,
                                           upper = model_lm),
                              direction = "forward")
```

```
summary(model_lm_step_back)
```

```
##
## Call:
## lm(formula = log_Apps ~ Private + Top25perc + F.Undergrad + Outstate +
##     Room.Board + Books + PhD + S.F.Ratio + perc.alumni + Expend +
##     Grad.Rate, data = College_clean, subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65694 -0.31586  0.03803  0.36946  1.69861
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.267e+00  2.349e-01  18.164  < 2e-16 ***
## Private     -5.395e-01  8.774e-02  -6.148 1.59e-09 ***
## Top25perc    3.981e-03  1.707e-03   2.333 0.020059 *
## F.Undergrad  1.104e-04  6.845e-06  16.124  < 2e-16 ***
## Outstate     4.765e-05  1.228e-05   3.879 0.000119 ***
## Room.Board   7.374e-05  3.060e-05   2.409 0.016333 *
## Books        2.396e-04  1.373e-04   1.744 0.081690 .
## PhD          8.310e-03  1.999e-03   4.157 3.79e-05 ***
## S.F.Ratio    5.003e-02  9.078e-03   5.511 5.68e-08 ***
## perc.alumni -9.238e-03  2.606e-03  -3.544 0.000430 ***
## Expend       2.969e-05  6.539e-06   4.541 7.01e-06 ***
## Grad.Rate    8.993e-03  1.854e-03   4.850 1.65e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5417 on 505 degrees of freedom
```

```
## Multiple R-squared:  0.7317, Adjusted R-squared:  0.7258
## F-statistic: 125.2 on 11 and 505 DF,  p-value: < 2.2e-16
```

```
summary(model_lm_step_forward)
```

```
##
## Call:
## lm(formula = log_Apps ~ F.Undergrad + PhD + Grad.Rate + Private +
##     Outstate + perc.alumni + S.F.Ratio + Expend + Room.Board +
##     Top25perc + Books, data = College_clean[train, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65694 -0.31586  0.03803  0.36946  1.69861
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.267e+00  2.349e-01  18.164  < 2e-16 ***
## F.Undergrad  1.104e-04  6.845e-06  16.124  < 2e-16 ***
## PhD          8.310e-03  1.999e-03   4.157 3.79e-05 ***
## Grad.Rate    8.993e-03  1.854e-03   4.850 1.65e-06 ***
## Private     -5.395e-01  8.774e-02  -6.148 1.59e-09 ***
## Outstate     4.765e-05  1.228e-05   3.879 0.000119 ***
## perc.alumni -9.238e-03  2.606e-03  -3.544 0.000430 ***
## S.F.Ratio    5.003e-02  9.078e-03   5.511 5.68e-08 ***
## Expend       2.969e-05  6.539e-06   4.541 7.01e-06 ***
## Room.Board   7.374e-05  3.060e-05   2.409 0.016333 *
## Top25perc    3.981e-03  1.707e-03   2.333 0.020059 *
## Books        2.396e-04  1.373e-04   1.744 0.081690 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5417 on 505 degrees of freedom
## Multiple R-squared:  0.7317, Adjusted R-squared:  0.7258
## F-statistic: 125.2 on 11 and 505 DF,  p-value: < 2.2e-16
```

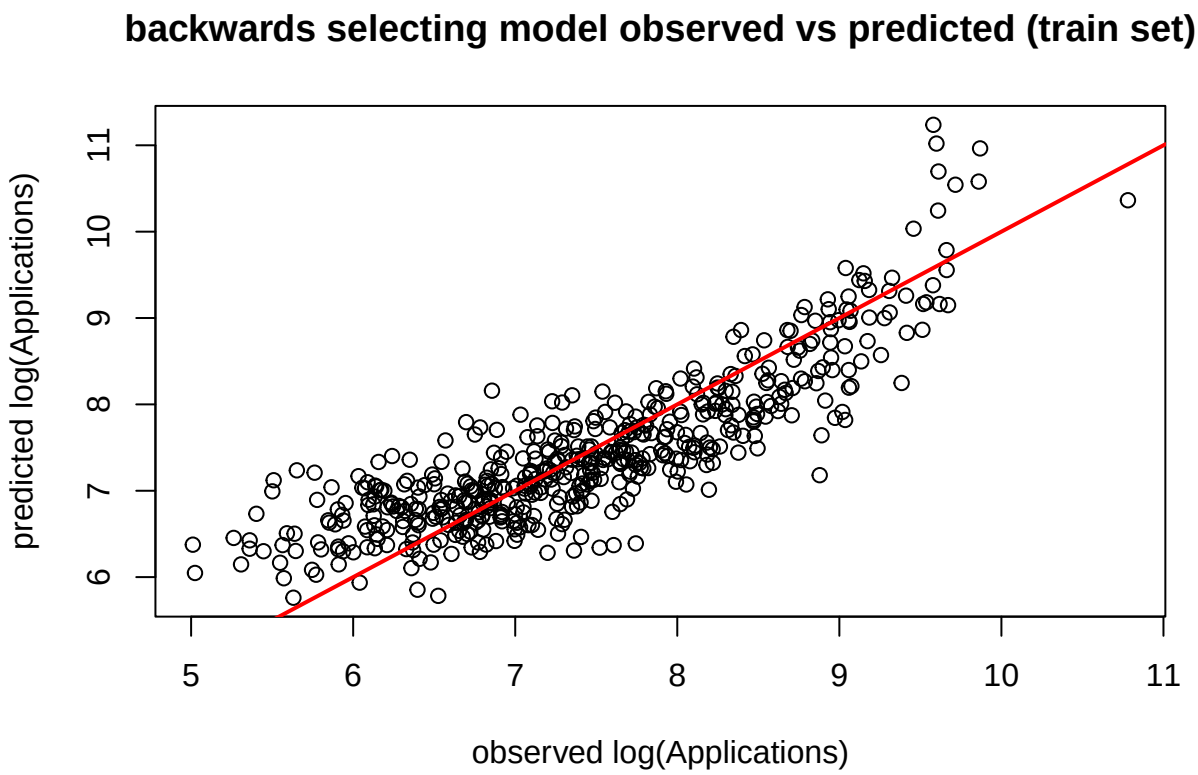
As w can observe, both forwards and backwards selection of the predictos result in exactly the same model. Therefore we only need to calculate the RMSE and visualise for one.

```
pred_lm_train_back <- predict(model_lm_step_back,
                              newdata = College_clean[train, ])
pred_lm_test_back <- predict(model_lm_step_back,
                             newdata = College_clean[test, ])
```

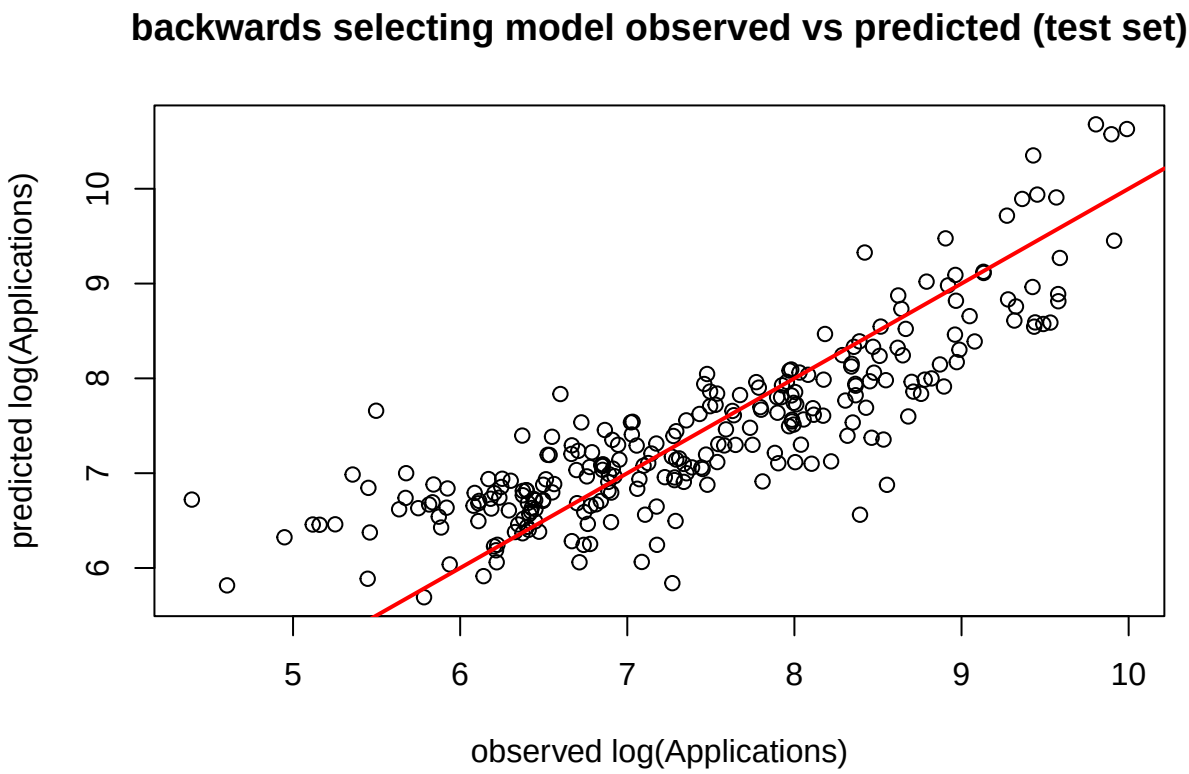
```
RMSE_train_back <- sqrt((1/n_train)*sum((observed_values_train -
                                           pred_lm_train_back)^2))
RMSE_test_back <- sqrt((1/n_test)*sum((observed_values_test -
                                         pred_lm_test_back)^2))
RMSE_back <- c(RMSE_train_back, RMSE_test_back)
RMSE_full <- c(RMSE_train, RMSE_test)
RMSE_red <- c(RMSE_train_red, RMSE_test_red)
results <- cbind(RMSE_full, RMSE_red, RMSE_back)
rownames(results) <- c("training set", "testing set")
colnames(results) <- c("full model", "reduced model", "backwards selection model")
print(results)
```

```
##              full model reduced model backwards selection model
## training set  0.5345712     0.5401250                 0.5353381
## testing set   0.6148493     0.6217448                 0.6127111
```

```
plot(observed_values_train, pred_lm_train_back,
     main = "backwards selecting model observed vs predicted (train set)",
     xlab = "observed log(Applications)",
     ylab = "predicted log(Applications)")
abline(0, 1, col = "red", lwd = 2)
```




```
plot(observed_values_test, pred_lm_test_back,
     main = "backwards selecting model observed vs predicted (test set)",
     xlab = "observed log(Applications)",
     ylab = "predicted log(Applications)")
abline(0, 1, col = "red", lwd = 2)
```



The RMSE values and the plots of predicted and observed values are again very similar to our models from before. If we take a look at the predictions, we can observe that most are the same as in the reduced model, except Top25perc and Books (which are not at that high significance levels). Therefore the similar results are not surprising. From this we can conclude that all models already include most of the significant predictors that hold most of the information and a simple linear regression can not further improve our results.