Eindopdracht_deel_2

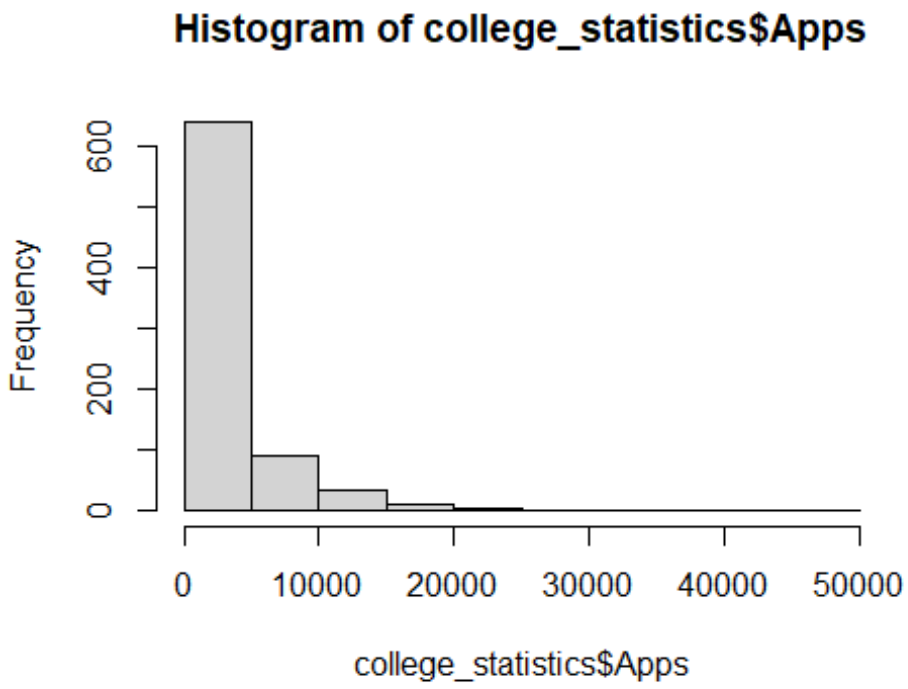
Mohammed Al Hor

2022-10-16

```
library(MASS)
library(car)
library(lmtest)
library(HH)
library(ggplot2)
```

4 (a) Voer eerst een test uit voor de hypothese dat het aantal aanmeldingen een normale verdeling volgt. Wat is je conclusie? Is deze conclusie van belang voor het verder modelleren van deze variabele?

```
summary(college_statistics$Apps)

hist(college_statistics$Apps)
```



Both the histogram and qq plot indicate that this variable is not normally distributed (Positive skewness in histogram and not a straight line in the qqplot). Let put this to the test using the Shapiro-Wilk test for normality.

H0: Apps is normally distributed

Ha: Apps is not normally distributed

```
shapiro.test(college_statistics$Apps)

##
##  Shapiro-Wilk normality test
##
## data:  college_statistics$Apps
## W = 0.65408, p-value < 2.2e-16
```

The p-value is very small (less than 0.05), which means the null hypothesis can be rejected and the data is not normally distributed. However, because we have a decent sample size (777 observations) OLS remains a statistically sound method to use.

4 (b) Deel de data eerst op willekeurige manier op in een "estimation" en "test" sample. Neem 600 universiteiten in de estimation sample. Zorg ervoor dat deze opdeling reproduceerbaar is. Hint: de R-functies set.seed en sample kunnen hiervoor gebruikt worden.

```
set.seed(123)

train_ind <- sample(seq_len(nrow(college_statistics)), size=600)

college_statistics_est <- college_statistics[train_ind,]
college_statistics_test <- college_statistics[-train_ind,]
```

First, we set the seed so the resulting dataframe can be reproduced. Then, we take a random sample of 600 observations. Using indexing we make an estimation dataframe and a test dataframe.

4 (c) Maak eerst een lineair model voor het aantal aanmeldingen. Gebruik hiervoor alleen de estimation sample.

```
fit1 <- lm(Apps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad
+ Outstate + Room.Board + Books + Personal + PhD + Terminal +
           S.F.Ratio + perc.alumni + Expend + Grad.Rate , data =
college_statistics_est)
summary(fit1)

##
## Call:
## lm(formula = Apps ~ Private + Top10perc + Top25perc + F.Undergrad +
##     P.Undergrad + Outstate + Room.Board + Books + Personal +
##     PhD + Terminal + S.F.Ratio + perc.alumni + Expend + Grad.Rate,
##     data = college_statistics_est)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5663.8  -693.2  -105.4   500.2  6501.9
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.826e+03  6.449e+02  -2.831 0.004803 **
## PrivateYes  -7.248e+02  2.243e+02  -3.231 0.001303 **
## Top10perc    2.875e+01  8.509e+00   3.379 0.000775 ***
## Top25perc   -8.082e+00  7.006e+00  -1.154 0.249097
## F.Undergrad  6.271e-01  2.015e-02  31.117  < 2e-16 ***
## P.Undergrad -1.620e-01  4.825e-02  -3.358 0.000837 ***
## Outstate     4.967e-02  2.893e-02   1.717 0.086458 .
## Room.Board   3.141e-01  7.245e-02   4.335 1.72e-05 ***
## Books        3.526e-01  3.951e-01   0.892 0.372530
## Personal    -1.508e-01  1.016e-01  -1.485 0.138210
## PhD         -3.465e+00  7.471e+00  -0.464 0.642987
## Terminal    -7.226e+00  7.951e+00  -0.909 0.363806
## S.F.Ratio    5.523e+00  2.019e+01   0.274 0.784464
## perc.alumni -2.260e+01  6.405e+00  -3.529 0.000450 ***
## Expend       9.556e-02  1.903e-02   5.023 6.78e-07 ***
## Grad.Rate    1.916e+01  4.635e+00   4.133 4.11e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1435 on 584 degrees of freedom
## Multiple R-squared:  0.8275, Adjusted R-squared:  0.8231
## F-statistic: 186.8 on 15 and 584 DF,  p-value: < 2.2e-16
```

Pretty straightforward, we use the lm() function to make a linear model. Accept and Enroll are omitted, these are obviously dependent on the amount of Apps.

4 (d) Pas backward elimination toe om het aantal variabelen terug te brengen.

We do the backwards step regression by using the stepAIC function and save the results in a list

```
backresults <- stepAIC(fit1, direction = "backward")
```

```
## Start:  AIC=8738.76
## Apps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + Personal + PhD + Terminal +
##     S.F.Ratio + perc.alumni + Expend + Grad.Rate
##
##                Df  Sum of Sq        RSS    AIC
## - S.F.Ratio     1     154253 1203281442 8736.8
## - PhD           1     443098 1203570287 8737.0
## - Books         1    1640768 1204767958 8737.6
## - Terminal      1    1701690 1204828879 8737.6
## - Top25perc     1    2742081 1205869270 8738.1
## <none>                       1203127189 8738.8
## - Personal      1    4540164 1207667354 8739.0
## - Outstate      1    6075495 1209202684 8739.8
## - Private       1   21507032 1224634222 8747.4
## - P.Undergrad   1   23225746 1226352935 8748.2
## - Top10perc     1   23525430 1226652619 8748.4
```

```
## - perc.alumni  1   25652521 1228779710 8749.4
## - Grad.Rate    1   35187706 1238314895 8754.1
## - Room.Board   1   38716903 1241844092 8755.8
## - Expend       1   51972844 1255100033 8762.1
## - F.Undergrad  1 1994744024 3197871214 9323.3
##
## Step:  AIC=8736.83
## Apps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + Personal + PhD + Terminal +
##     perc.alumni + Expend + Grad.Rate
##
##               Df  Sum of Sq        RSS    AIC
## - PhD          1     414401 1203695843 8735.0
## - Books        1    1665478 1204946920 8735.7
## - Terminal     1    1756106 1205037549 8735.7
## - Top25perc    1    2726599 1206008042 8736.2
## <none>                      1203281442 8736.8
## - Personal     1    4803821 1208085263 8737.2
## - Outstate     1    5985897 1209267339 8737.8
## - Private      1   22982373 1226263815 8746.2
## - P.Undergrad  1   23276267 1226557709 8746.3
## - Top10perc    1   23455551 1226736994 8746.4
## - perc.alumni  1   26103898 1229385340 8747.7
## - Grad.Rate    1   35075382 1238356824 8752.1
## - Room.Board   1   38768998 1242050441 8753.9
## - Expend       1   56786478 1260067921 8762.5
## - F.Undergrad  1 2036163530 3239444972 9329.0
##
## Step:  AIC=8735.04
## Apps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + Personal + Terminal + perc.alumni +
##     Expend + Grad.Rate
##
##               Df  Sum of Sq        RSS    AIC
## - Books        1    1852894 1205548737 8734.0
## - Top25perc    1    2804756 1206500599 8734.4
## <none>                      1203695843 8735.0
## - Personal     1    5001124 1208696967 8735.5
## - Outstate     1    5802887 1209498731 8735.9
## - Terminal     1    7076378 1210772221 8736.6
## - Private      1   22571491 1226267334 8744.2
## - Top10perc    1   23124619 1226820462 8744.5
## - P.Undergrad  1   23579052 1227274895 8744.7
## - perc.alumni  1   25852700 1229548543 8745.8
## - Grad.Rate    1   34675516 1238371359 8750.1
## - Room.Board   1   38800594 1242496437 8752.1
## - Expend       1   56425252 1260121095 8760.5
## - F.Undergrad  1 2035841983 3239537826 9327.1
##
## Step:  AIC=8733.96
```

```
## Apps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Personal + Terminal + perc.alumni +
##     Expend + Grad.Rate
##
##               Df  Sum of Sq        RSS    AIC
## - Top25perc    1    2623869 1208172606 8733.3
## <none>                      1205548737 8734.0
## - Personal     1    4114782 1209663519 8734.0
## - Outstate     1    5679298 1211228035 8734.8
## - Terminal     1    6729520 1212278257 8735.3
## - Private      1   22284578 1227833315 8743.0
## - P.Undergrad  1   23295180 1228843917 8743.4
## - Top10perc    1   23469952 1229018689 8743.5
## - perc.alumni  1   26709653 1232258390 8745.1
## - Grad.Rate    1   34463261 1240011998 8748.9
## - Room.Board   1   40214652 1245763389 8751.7
## - Expend       1   56318465 1261867202 8759.4
## - F.Undergrad  1 2035799498 3241348235 9325.4
##
## Step:  AIC=8733.27
## Apps ~ Private + Top10perc + F.Undergrad + P.Undergrad + Outstate +
##     Room.Board + Personal + Terminal + perc.alumni + Expend +
##     Grad.Rate
##
##               Df  Sum of Sq        RSS    AIC
## <none>                      1208172606 8733.3
## - Personal     1    4300046 1212472653 8733.4
## - Outstate     1    5360095 1213532702 8733.9
## - Terminal     1    8813085 1216985691 8735.6
## - Private      1   21815224 1229987830 8742.0
## - P.Undergrad  1   23418800 1231591407 8742.8
## - perc.alumni  1   28277317 1236449923 8745.1
## - Grad.Rate    1   32910078 1241082685 8747.4
## - Top10perc    1   35623933 1243796539 8748.7
## - Room.Board   1   40153749 1248326355 8750.9
## - Expend       1   66347441 1274520047 8763.3
## - F.Undergrad  1 2033909292 3242081898 9323.5
```

Then, we record and evaluate the model selected by the backwards step regression method and assign this to fit1 using the following code:

```
backmodel <- backresults$call
backmodel
```

```
## lm(formula = Apps ~ Private + Top10perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Personal + Terminal + perc.alumni +
##     Expend + Grad.Rate, data = college_statistics_est)
```

```
# This line evaluates the 'code' of the model
backmodel <- eval(backmodel)
```

```
fit1 <- backmodel
summary(fit1)
```

```
##
## Call:
## lm(formula = Apps ~ Private + Top10perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Personal + Terminal + perc.alumni +
##     Expend + Grad.Rate, data = college_statistics_est)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5662.7  -694.6  -103.6   526.2  6428.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.703e+03  4.608e+02  -3.697 0.000239 ***
## PrivateYes  -7.113e+02  2.183e+02  -3.258 0.001185 **
## Top10perc    2.092e+01  5.024e+00   4.164 3.60e-05 ***
## F.Undergrad  6.265e-01  1.991e-02  31.462  < 2e-16 ***
## P.Undergrad -1.625e-01  4.813e-02  -3.376 0.000784 ***
## Outstate     4.643e-02  2.874e-02   1.615 0.106817
## Room.Board   3.190e-01  7.217e-02   4.421 1.17e-05 ***
## Personal    -1.431e-01  9.893e-02  -1.447 0.148530
## Terminal    -1.088e+01  5.254e+00  -2.071 0.038791 *
## perc.alumni -2.352e+01  6.341e+00  -3.710 0.000227 ***
## Expend       9.815e-02  1.727e-02   5.682 2.09e-08 ***
## Grad.Rate    1.835e+01  4.585e+00   4.002 7.08e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1433 on 588 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8236
## F-statistic: 255.2 on 11 and 588 DF,  p-value: < 2.2e-16
```

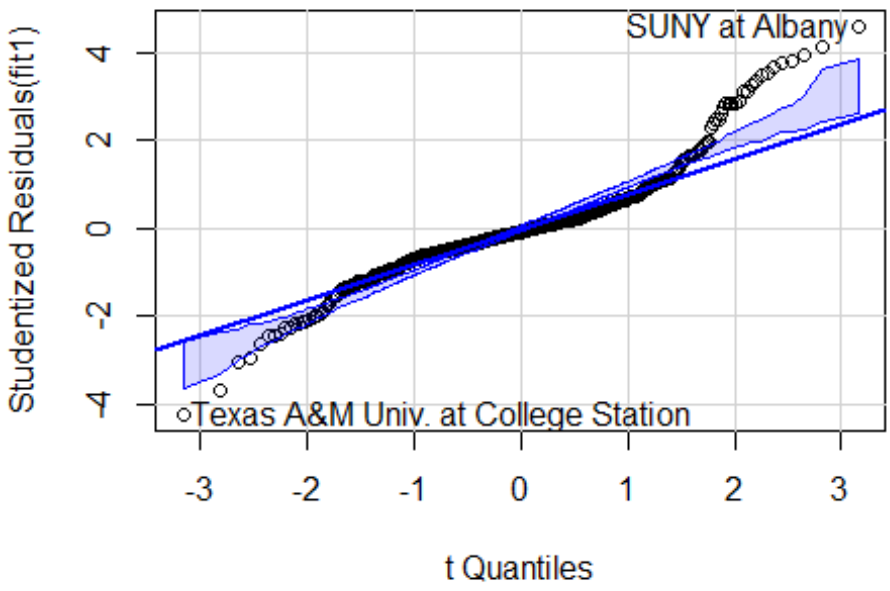
The formula for this model is as follows: formula = Apps ~ Private + Top10perc + F.Undergrad + P.Undergrad + Outstate + Room.Board + Personal + Terminal + perc.alumni + Expend + Grad.Rate

4 (e) Voer diverse toetsen uit om de aannamen van het lineaire model te testen.

Now that we have a model, we can start by testing some of the assumptions of this model. First, let's take a look at the qqplot of this model.

Test for normality(A6) using qqplot:

```
qqPlot(fit1)
```




```
shapiro.test(residuals(fit1))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit1)
## W = 0.91962, p-value < 2.2e-16
```

A lot of the datapoints don't fall along the reference line, which means we can assume non normality. We alo observe outliers for Texas A&M Univ. at College Station and SUNY at Albany. Furthermore, the Shpiro-Wilk test on the residuals provides us the evidence to reject normality (p-value is quite small, so we can reject the null hypothesis for normality).

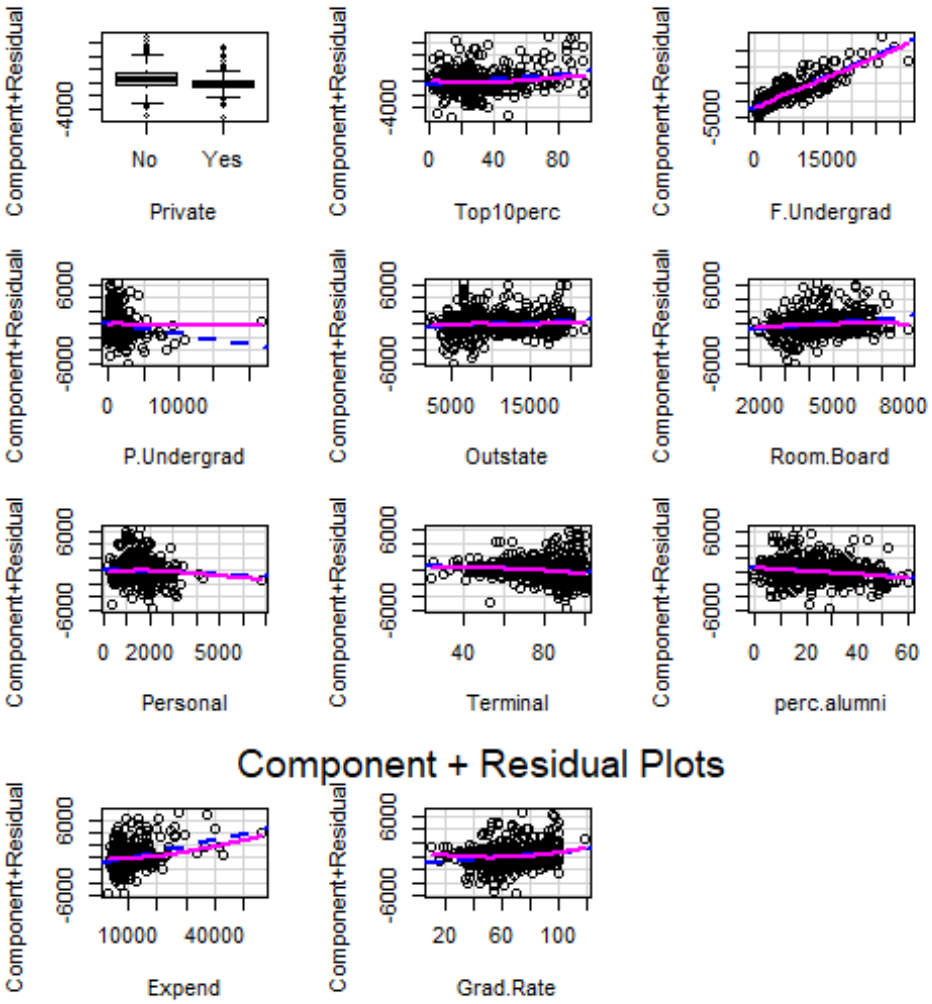
Test for independence(A5) using the Durbin-Watson test.

```
durbinWatsonTest(fit1)

##  lag Autocorrelation D-W Statistic p-value
##    1       0.0359695      1.927191   0.414
##  Alternative hypothesis: rho != 0
```

No autocorrelation detected (p-value is 0.382), which makes sense when we take into account the data we have(not time series data).

Let's check for linearity(A3) using component + residual plot

```
crPlots(fit1)
```



Looking at these plots, we observe some non-linearity (especially in Expend and Personal). Let's put this to the test, using the the resettest from the lmtest package.

```
resettest(fit1, power=2)
```

```
##
##  RESET test
##
## data:  fit1
## RESET = 5.0481, df1 = 1, df2 = 587, p-value = 0.02502
```

H0: linear relation between x and y

Ha: some nonlinearity

As we can see the p-value is quite small (<0.05) which means we can reject the null hypthesis and thus linearity.

Next, let's take a look at Homoskedasticity(A4) using the Breusch-Pagan test:

```
ncvTest(fit1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 394.1075, Df = 1, p = < 2.22e-16
```

H0: constant variances (homoskedasticity)

Ha: non-constant variances (heteroskedasticity)

We can reject H0 of homoskedasticity -> variances are not constant or heteroscedasticity is present.

We can also check for multicolinearity (A7) (No perfect linear relationship in X)

```
vif(fit1)
```

```
##  PrivateYes   Top10perc F.Undergrad P.Undergrad    Outstate  Room.Board
##    2.784858    2.351514    2.419156    1.651071    3.898348    1.834423
##    Personal    Terminal perc.alumni      Expend   Grad.Rate
##    1.260313    1.753760    1.795078    2.401752    1.847167
```

None are larger than 4 (this is the rule of thumb), we can assume no multicolinearity.

Because OLS is quite sensitive to outliers, let's do a quick outlier test.

```
outlierTest(fit1)
```

```
##                                  rstudent unadjusted p-value Bonferroni
p
## SUNY at Albany                   4.597908         5.2295e-06
0.0031377
## Texas A&M Univ. at College Station -4.230119         2.7093e-05
0.0162560
## University of Virginia           4.142904         3.9336e-05
0.0236020
```

As we previously saw in the qqplot there are some outliers, 'SUNY at Albany', 'Texas A&M Univ. at College Station' and 'University of Virginia'.

4 (f) Maak vervolgens een model voor de logaritme van het aantal aanmeldingen (ook weer met backward elimination).

The steps for this are the same as for the previous model, thus I will not go into detail.

```
fit2 <- lm(log(Apps) ~ Private + Top10perc + Top25perc + F.Undergrad +
P.Undergrad + Outstate + Room.Board + Books + Personal + PhD + Terminal +
             S.F.Ratio + perc.alumni + Expend + Grad.Rate , data =
college_statistics_est)
summary(fit2)

##
## Call:
## lm(formula = log(Apps) ~ Private + Top10perc + Top25perc + F.Undergrad +
##     P.Undergrad + Outstate + Room.Board + Books + Personal +
##     PhD + Terminal + S.F.Ratio + perc.alumni + Expend + Grad.Rate,
##     data = college_statistics_est)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.17654 -0.33006  0.02374  0.36395  1.80192
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.341e+00  2.530e-01  17.159  < 2e-16 ***
## PrivateYes  -6.232e-01  8.798e-02  -7.083 4.08e-12 ***
## Top10perc    1.088e-03  3.337e-03   0.326  0.74463
## Top25perc    2.656e-03  2.748e-03   0.967  0.33416
## F.Undergrad  1.131e-04  7.905e-06  14.311  < 2e-16 ***
## P.Undergrad -4.220e-06  1.893e-05  -0.223  0.82365
## Outstate     4.677e-05  1.135e-05   4.123 4.29e-05 ***
## Room.Board   7.242e-05  2.842e-05   2.548  0.01108 *
## Books        4.324e-04  1.550e-04   2.791  0.00543 **
## Personal     5.496e-05  3.983e-05   1.380  0.16819
## PhD          3.506e-03  2.930e-03   1.197  0.23198
## Terminal     2.153e-03  3.119e-03   0.690  0.49035
## S.F.Ratio    4.267e-02  7.917e-03   5.389 1.03e-07 ***
## perc.alumni -7.010e-03  2.512e-03  -2.790  0.00544 **
## Expend       2.990e-05  7.463e-06   4.006 6.97e-05 ***
## Grad.Rate    1.039e-02  1.818e-03   5.714 1.76e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.563 on 584 degrees of freedom
## Multiple R-squared:  0.7157, Adjusted R-squared:  0.7084
## F-statistic: 98.02 on 15 and 584 DF,  p-value: < 2.2e-16
```

```r
# Backwards step regression
backresults <- stepAIC(fit2, direction = "backward")
```

Output omitted to keep this report as short as possible (see code if you want the output)

```r
# Record the best model selected by the backwards method
# This line takes the model specification as 'code'
backmodel <- backresults$call
backmodel
```

```
## lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
##     Outstate + Room.Board + Books + PhD + S.F.Ratio + perc.alumni +
##     Expend + Grad.Rate, data = college_statistics_est)
```

```r
# This line evaluates the 'code' of the model
backmodel <- eval(backmodel)
summary(backmodel)
```

```
##
## Call:
## lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
##     Outstate + Room.Board + Books + PhD + S.F.Ratio + perc.alumni +
##     Expend + Grad.Rate, data = college_statistics_est)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.21420 -0.33085  0.02215  0.37221  1.76272
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.469e+00  2.226e-01  20.080  < 2e-16 ***
## PrivateYes  -6.243e-01  8.709e-02  -7.169 2.28e-12 ***
## Top25perc    3.449e-03  1.642e-03   2.101  0.03607 *
## F.Undergrad  1.149e-04  7.126e-06  16.123  < 2e-16 ***
## Outstate     4.581e-05  1.122e-05   4.084 5.03e-05 ***
## Room.Board   7.191e-05  2.805e-05   2.564  0.01061 *
## Books        4.810e-04  1.512e-04   3.182  0.00154 **
## PhD          5.110e-03  1.968e-03   2.596  0.00966 **
## S.F.Ratio    4.117e-02  7.842e-03   5.250 2.12e-07 ***
## perc.alumni -7.100e-03  2.486e-03  -2.856  0.00445 **
## Expend       3.112e-05  6.837e-06   4.552 6.47e-06 ***
## Grad.Rate    9.995e-03  1.767e-03   5.656 2.42e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5622 on 588 degrees of freedom
## Multiple R-squared:  0.7146, Adjusted R-squared:  0.7092
## F-statistic: 133.8 on 11 and 588 DF,  p-value: < 2.2e-16
```

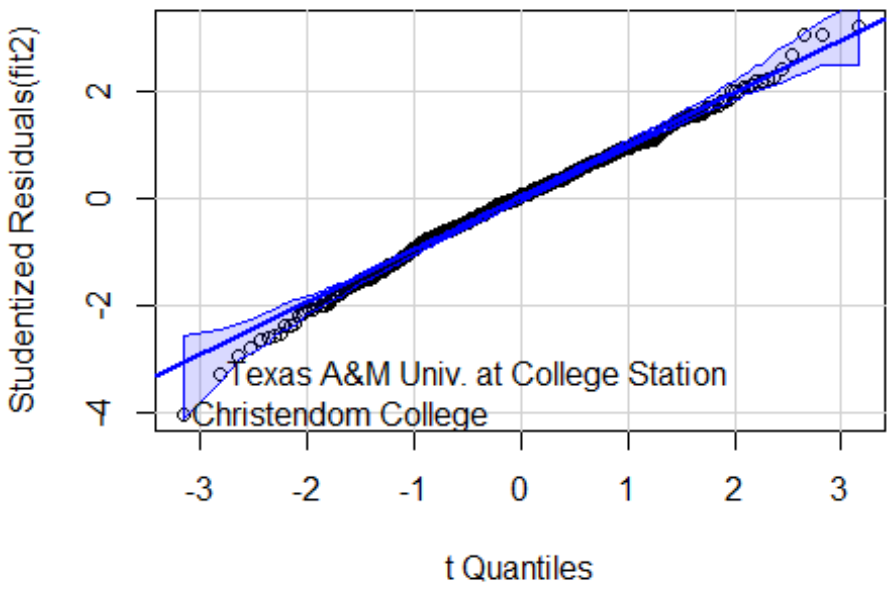
```
fit2 <- backmodel
summary(fit2)

##
## Call:
## lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
##     Outstate + Room.Board + Books + PhD + S.F.Ratio + perc.alumni +
##     Expend + Grad.Rate, data = college_statistics_est)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.21420 -0.33085  0.02215  0.37221  1.76272
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.469e+00  2.226e-01  20.080  < 2e-16 ***
## PrivateYes  -6.243e-01  8.709e-02  -7.169 2.28e-12 ***
## Top25perc    3.449e-03  1.642e-03   2.101  0.03607 *
## F.Undergrad  1.149e-04  7.126e-06  16.123  < 2e-16 ***
## Outstate     4.581e-05  1.122e-05   4.084 5.03e-05 ***
## Room.Board   7.191e-05  2.805e-05   2.564  0.01061 *
## Books        4.810e-04  1.512e-04   3.182  0.00154 **
## PhD          5.110e-03  1.968e-03   2.596  0.00966 **
## S.F.Ratio    4.117e-02  7.842e-03   5.250 2.12e-07 ***
## perc.alumni -7.100e-03  2.486e-03  -2.856  0.00445 **
## Expend       3.112e-05  6.837e-06   4.552 6.47e-06 ***
## Grad.Rate    9.995e-03  1.767e-03   5.656 2.42e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5622 on 588 degrees of freedom
## Multiple R-squared:  0.7146, Adjusted R-squared:  0.7092
## F-statistic: 133.8 on 11 and 588 DF,  p-value: < 2.2e-16
```

Different independent variables are selected by the backwards step regression when we use the log of Apps. formula = log(Apps) ~ Private + Top25perc + F.Undergrad + Outstate + Room.Board + Books + PhD + S.F.Ratio + perc.alumni + Expend + Grad.Rate, data = college_statistics_est

4 (g) Voer opnieuw de diverse toetsen uit om de aannamen van het model te testen.

Test for normality

```
qqPlot(fit2)
```




```
##                Christendom College Texas A&M Univ. at College Station
##                                242                                 488
```

```
shapiro.test(residuals(fit2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit2)
## W = 0.99453, p-value = 0.03029
```

The data points fall along the reference line quite well compared to the previous model. We still see some outliers for 'Christendom College' and 'Texas A&M Univ. at College Station'. A Shapiro-Wilk test for normality of the residuals provides us with evidence that the residuals are not normally distributed(null hypothesis can be rejected with p-value smaller than 0.05).

Test for independence using the Durbin-Watson test.

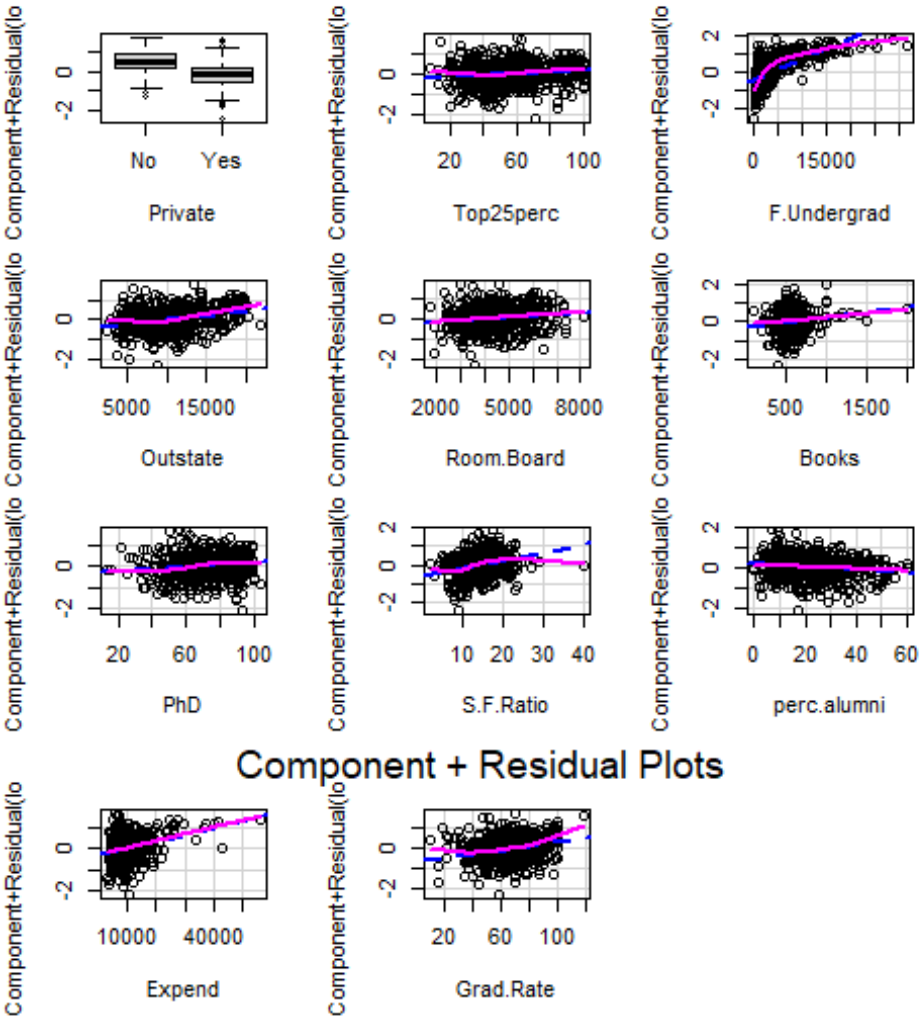
```
durbinWatsonTest(fit2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.06271588      1.873683   0.104
##  Alternative hypothesis: rho != 0
```

No autocorrelation detected (p-value is 0.122), which makes sense when we take into account the data we have(not time series data).

Let's check for linearity(A3) using component + residual plot

```
crPlots(fit2)
```



Looking at these plots, we observe more 'non-linearity' than in the previous mdoel. (especially in F.Undergrad, S.F.Ratio, Grad.Rate). Let's put this to the test, using the the resettest from the lmtest package.

H0: linear relation between x and y Ha: some nonlinearity

```
resettest(fit2, power=2)
```

```
##
##  RESET test
##
## data:  fit2
## RESET = 84.075, df1 = 1, df2 = 587, p-value < 2.2e-16
```

As we can see the p-value is quite small (<0.05) which means we can reject the null hypthesis and thus linearity.

Next, let's take a look at Homoskedasticity(A4) using the Breusch-Pagan test:

```
ncvTest(fit2)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.550253, Df = 1, p = 0.2131
```

H0: constant variances (homoskedasticity) Ha: non-constant variances (heteroskedasticity) We fail to reject the null-hypothesis (p-value > 0.05), and thus can assume homoskedacticity is present.

A7 Multicolinearity (No perfect linear relationship in X)

```
vif(fit2)
```

```
##  PrivateYes   Top25perc F.Undergrad    Outstate  Room.Board       Books
##    2.881235    2.024971    2.014211    3.858887    1.801658    1.046697
##         PhD   S.F.Ratio perc.alumni      Expend   Grad.Rate
##    1.885400    1.887194    1.794143    2.446720    1.784017
```

None are larger than 4 (this is the rule of thumb), we can assume no multicolinearity.

Because OLS is quite sensitive to outliers, let's do a quick outlier test.

```
outlierTest(fit2)
```

```
##                      rstudent unadjusted p-value Bonferroni p
## Christendom College -4.035229         6.1767e-05      0.03706
```

As we previously saw in the qqplot we have one observation that was relatively far from the reference line;'Christendom College'.

4 (h) Welk van de twee modellen heeft de voorkeur?

Looking at the previous test we performed on the two models we see that the log model has a better fit (qqplot), the variances are constant (homoskedacticity) and all the variables are significant. However, let's compare these two models using 'Goodness-of-fit Measures'. Akaike's information criterion can be calculated using the AIC function in R. The smaller the AIC the better the fit of the model.

```
AIC(fit1,fit2)
```

```
##      df       AIC
## fit1 13 10437.993
## fit2 13  1025.521
```

We see that model 2 outperforms model 1, the log-likelihood value is much lower, 1025 for model 2 and 10437 for model 1. Therefore we pick model 2.

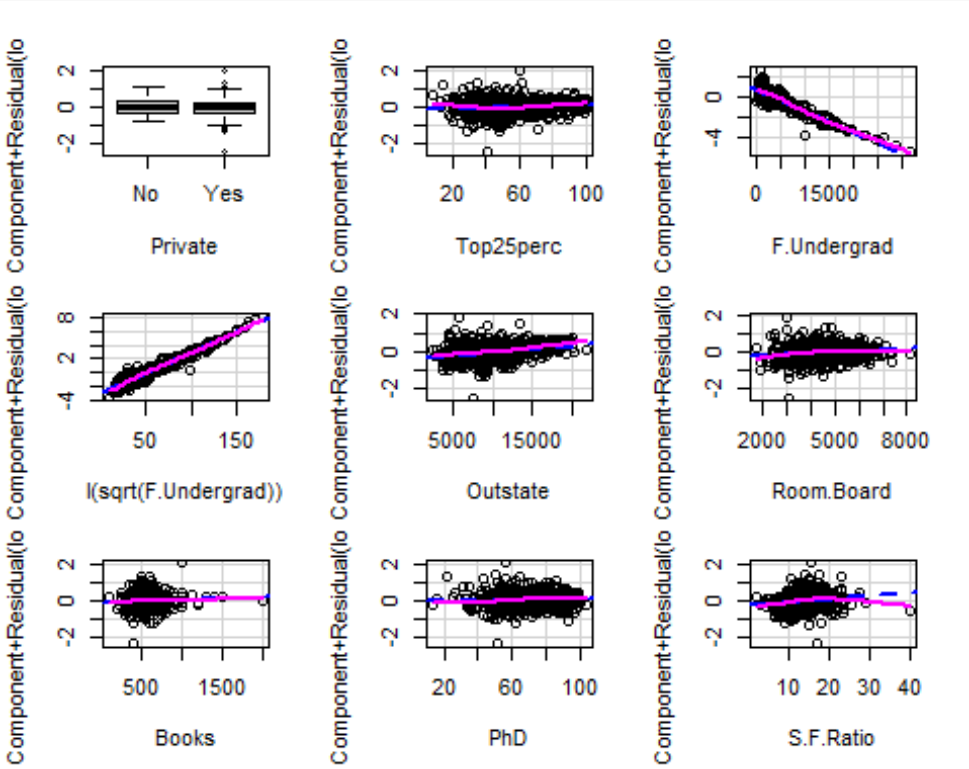
4 (i) Probeer het gekozen model nog verder te verbeteren: denk aan het toevoegen van transformaties van verklarende variabelen.

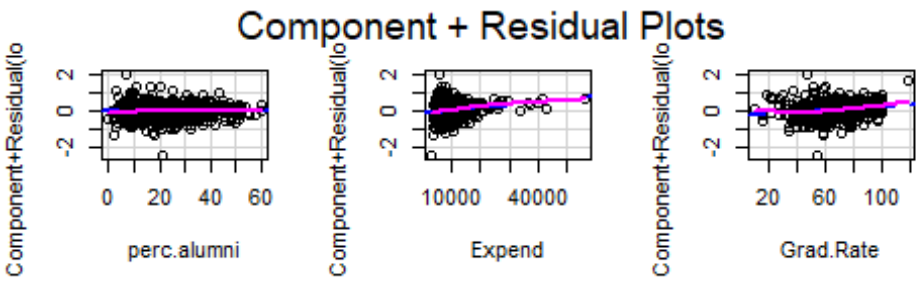
It is apparent, from the crPlot, that the relationship between the log(Apps) and the F.Undergrad variable is not linear. Let's try to do a square root transformation on this variable and compare the models.

```
fit3 <- lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
I(sqrt(F.Undergrad)) +
             Outstate + Room.Board + Books + PhD + S.F.Ratio + perc.alumni +
             Expend + Grad.Rate, data = college_statistics_est)

summary(fit3)
```

```
##
## Call:
## lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
##     I(sqrt(F.Undergrad)) + Outstate + Room.Board + Books + PhD +
##     S.F.Ratio + perc.alumni + Expend + Grad.Rate, data =
college_statistics_est)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.39100 -0.24284  0.01382  0.26962  2.03651
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.492e+00  1.736e-01  20.117  < 2e-16 ***
## PrivateYes            -4.956e-02  7.085e-02  -0.699 0.484517
## Top25perc              2.324e-03  1.236e-03   1.881 0.060485 .
## F.Undergrad           -2.111e-04  1.624e-05 -13.004  < 2e-16 ***
## I(sqrt(F.Undergrad))   5.959e-02  2.801e-03  21.273  < 2e-16 ***
## Outstate               3.614e-05  8.448e-06   4.278  2.2e-05 ***
## Room.Board             5.399e-05  2.111e-05   2.557 0.010808 *
```

```
## Books                  1.627e-04  1.147e-04   1.418 0.156598
## PhD                    1.268e-03  1.491e-03   0.850 0.395662
## S.F.Ratio              1.656e-02  6.010e-03   2.755 0.006049 **
## perc.alumni            3.102e-04  1.902e-03   0.163 0.870506
## Expend                 1.514e-05  5.197e-06   2.914 0.003701 **
## Grad.Rate              4.846e-03  1.351e-03   3.587 0.000362 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4228 on 587 degrees of freedom
## Multiple R-squared:  0.8388, Adjusted R-squared:  0.8355
## F-statistic: 254.6 on 12 and 587 DF,  p-value: < 2.2e-16

crPlots(fit3)
```





The crPlots presents us with much better results, we see a more linear relationship for F.Undergrad Let's check this using AIC

```
AIC(fit2,fit3)
```

```
##      df       AIC
## fit2 13 1025.5207
## fit3 14  684.6197
```

The AIC value goes from 1025 to 684, the transformation results in a much better model!

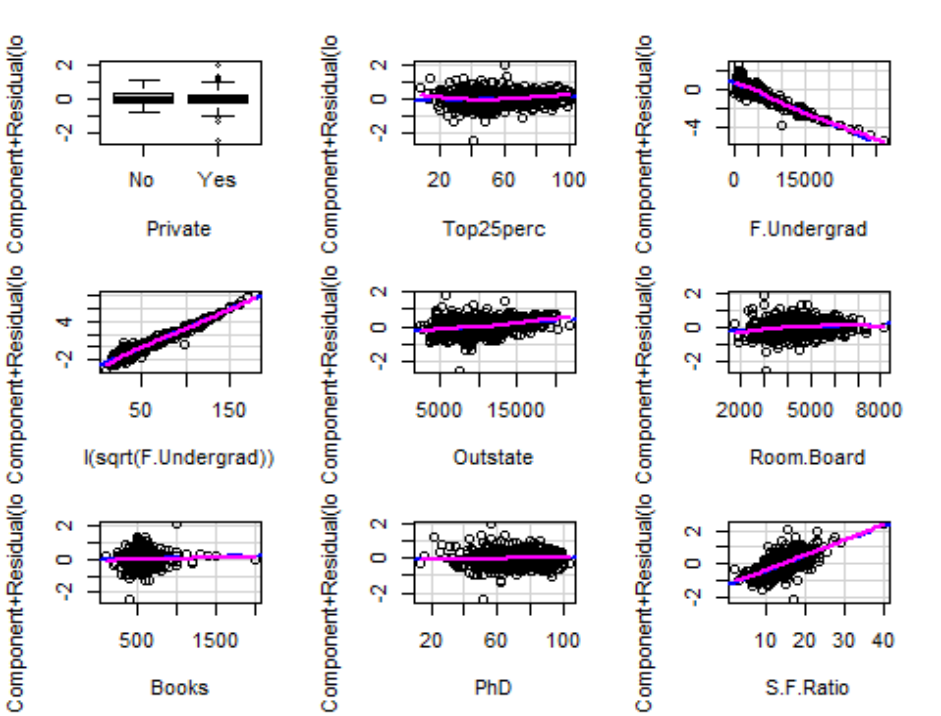
Let's build on this model and perform a quadratic transformation on the S.F.Ratio (see crPlot for non linear relationship).

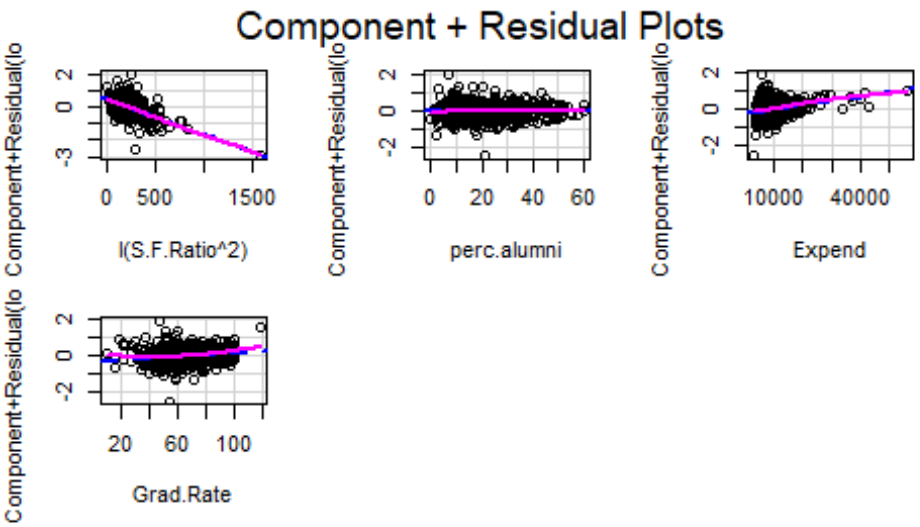
```
fit4 <- lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
I(sqrt(F.Undergrad)) +
             Outstate + Room.Board + Books + PhD + S.F.Ratio + I(S.F.Ratio^2)
+ perc.alumni +
             Expend + Grad.Rate, data = college_statistics_est)
summary(fit4)
```

```
##
## Call:
## lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
##     I(sqrt(F.Undergrad)) + Outstate + Room.Board + Books + PhD +
##     S.F.Ratio + I(S.F.Ratio^2) + perc.alumni + Expend + Grad.Rate,
##     data = college_statistics_est)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.40208 -0.25394  0.00937  0.26862  1.99871
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.909e+00  2.308e-01  12.601  < 2e-16 ***
## PrivateYes             -4.553e-02  7.007e-02  -0.650 0.516060
## Top25perc               2.463e-03  1.223e-03   2.015 0.044375 *
## F.Undergrad            -2.087e-04  1.607e-05 -12.990  < 2e-16 ***
## I(sqrt(F.Undergrad))    5.912e-02  2.773e-03  21.323  < 2e-16 ***
## Outstate                3.547e-05  8.356e-06   4.245 2.54e-05 ***
## Room.Board              5.735e-05  2.090e-05   2.744 0.006255 **
## Books                   1.572e-04  1.134e-04   1.386 0.166328
## PhD                     4.130e-04  1.492e-03   0.277 0.782047
## S.F.Ratio               9.064e-02  2.050e-02   4.421 1.17e-05 ***
## I(S.F.Ratio^2)         -2.172e-03  5.752e-04  -3.776 0.000176 ***
## perc.alumni             2.506e-04  1.881e-03   0.133 0.894052
## Expend                  2.343e-05  5.588e-06   4.193 3.18e-05 ***
## Grad.Rate               4.664e-03  1.337e-03   3.489 0.000521 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4181 on 586 degrees of freedom
## Multiple R-squared:  0.8426, Adjusted R-squared:  0.8392
## F-statistic: 241.4 on 13 and 586 DF,  p-value: < 2.2e-16
```

```
crPlots(fit4)
```





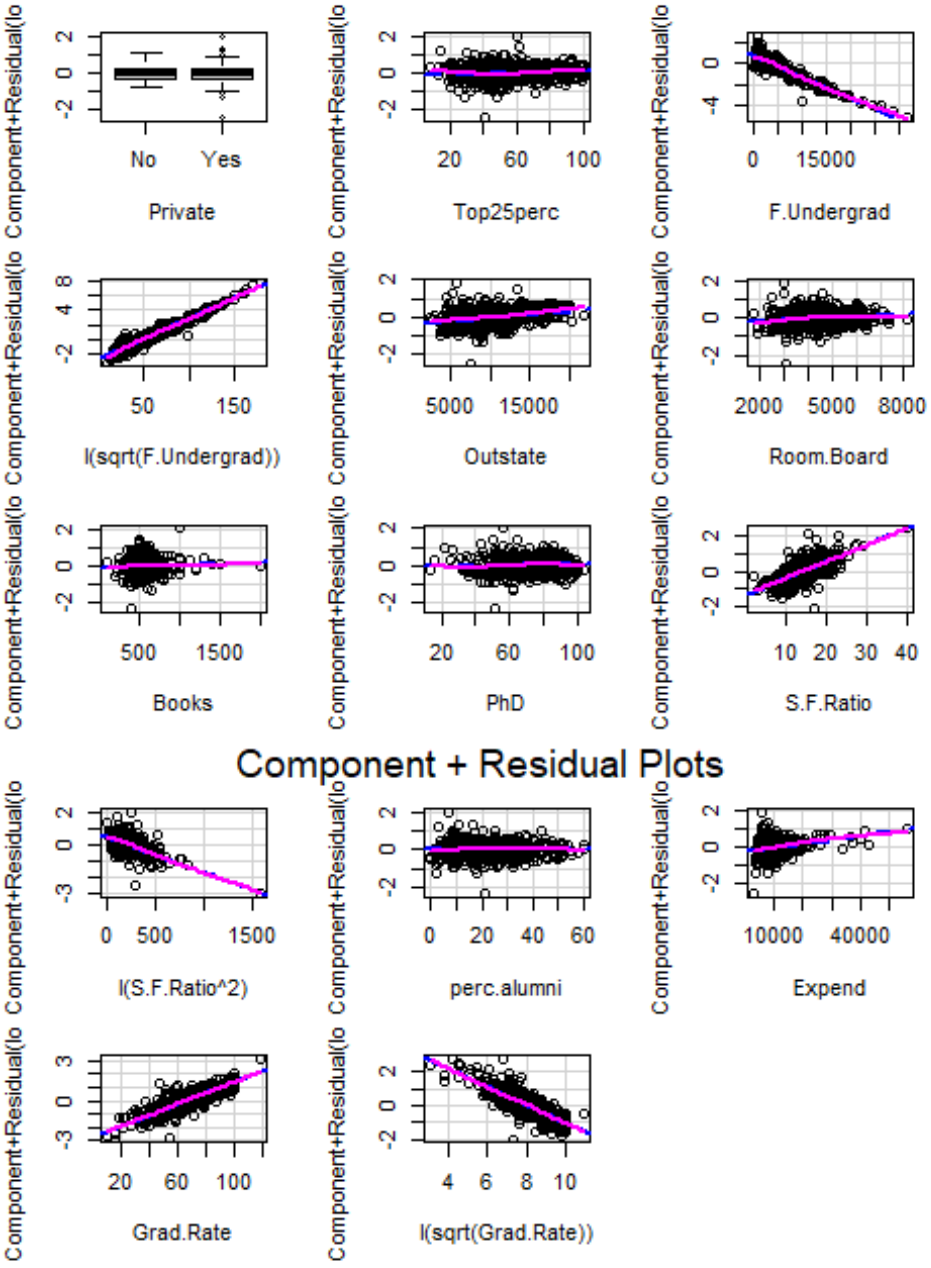

```
AIC(fit3,fit4)

##      df      AIC
## fit3 14 684.6197
## fit4 15 672.1987
```

For the sake of keeping this report brief, I'm not going to show the crPlots for each of the transformation. The crPlot of S.F.Ratio is more linear and when we look at the AIC, we can see a slight decrease. The model is improving. Let's continue:

Next, we perform a quadratic transformation on 'Grad.Rate'.

```
fit5 <- lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
I(sqrt(F.Undergrad)) +
             Outstate + Room.Board + Books + PhD + S.F.Ratio + I(S.F.Ratio^2)
+ perc.alumni +
             Expend + Grad.Rate+ I(sqrt(Grad.Rate)), data =
college_statistics_est)

summary(fit5)

##
## Call:
## lm(formula = log(Apps) ~ Private + Top25perc + F.Undergrad +
##     I(sqrt(F.Undergrad)) + Outstate + Room.Board + Books + PhD +
##     S.F.Ratio + I(S.F.Ratio^2) + perc.alumni + Expend + Grad.Rate +
##     I(sqrt(Grad.Rate)), data = college_statistics_est)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.36646 -0.25131  0.01409  0.26443  2.01728
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.983e+00  5.567e-01   8.951  < 2e-16 ***
## PrivateYes           -4.827e-02  6.916e-02  -0.698 0.485474
## Top25perc             2.277e-03  1.207e-03   1.886 0.059770 .
## F.Undergrad          -2.040e-04  1.590e-05 -12.828  < 2e-16 ***
## I(sqrt(F.Undergrad))  5.847e-02  2.741e-03  21.333  < 2e-16 ***
## Outstate              3.597e-05  8.247e-06   4.362 1.53e-05 ***
## Room.Board            5.654e-05  2.063e-05   2.741 0.006305 **
## Books                 1.450e-04  1.120e-04   1.295 0.195815
## PhD                   7.807e-04  1.475e-03   0.529 0.596849
## S.F.Ratio             9.326e-02  2.024e-02   4.607 5.02e-06 ***
## I(S.F.Ratio^2)       -2.244e-03  5.679e-04  -3.951 8.72e-05 ***
## perc.alumni          -9.288e-05  1.858e-03  -0.050 0.960152
## Expend                2.159e-05  5.533e-06   3.902 0.000106 ***
## Grad.Rate             4.036e-02  8.841e-03   4.566 6.07e-06 ***
## I(sqrt(Grad.Rate))   -5.492e-01  1.345e-01  -4.084 5.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4126 on 585 degrees of freedom
## Multiple R-squared:  0.847,  Adjusted R-squared:  0.8433
## F-statistic: 231.3 on 14 and 585 DF,  p-value: < 2.2e-16

crPlots(fit5)
```




```
AIC(fit4,fit5)
##      df      AIC
## fit4 15 672.1987
## fit5 16 657.3329
```

Again, the model goes from an AIC of 672 to 657, a slight improvement. At this point I'm content with the model, more transformation will marginally decrease the AIC and thus I am moving on to the next question.

4 (j) Hoe interpreteer je de coefficienten in het model dat je uiteindelijk hebt gevonden? Wees hierbij heel precies. Welke factoren zijn uiteindelijk het meest van belang?

Because we did a log transformation on 'Apps' in this model, we cannot directly interpret the coefficients. What this means is; if the variable (x) increases by 1 the number of 'Apps' increases approximately by 100*coefficient %.

The most important variables are as follows: Grad.Rate, S.F.Ratio, F.Undergrad, Outstate and Expend.

4 (k) Gebruik het uiteindelijke model om voorspellingen te maken voor de waarnemingen in de estimation en de test sample.

We use the predict function to make predictions on the test and estimation sets. Exp() is used on these predictions to get the real number of Apps, instead of the log values.

```
college_statistics_test$predict <- exp(predict(fit5, newdata =
college_statistics_test))
college_statistics_est$predict <- exp(predict(fit5, newdata =
college_statistics_est))
```

4 (l) Vergelijk de voorspelkracht (mbv. mean squared error) van het model op de estimation sample met die op de test sample. Wat concludeer je?

First let's calculate the mean squared error of both predictions.

```
test_MSE <- mean((college_statistics_test$Apps -
college_statistics_test$predict)^2)
est_MSE <- mean((college_statistics_est$Apps -
college_statistics_est$predict)^2)
```

The MSE for the test set is 8.509.323, for the estimation set it's 1.849.956. The model obviously has more explanatory power for the data it was built on. Outside of this sample this power decreases.

5. In dit onderdeel voer je een ANOVA analyse uit op de relatie tussen de student faculty ratio en het percentage studenten uit de top 25% van de high school. Volg de volgende stappen:

5 (a) Maak een factor met drie levels op basis van de variabele Top25perc. De levels zijn: laag (minder dan 20%)/midden/hoog (meer dan 40%)

The code below creates a categorical variable from conditions, in the last line this variable is converted to a factor.

```
college_statistics$catTop25perc[as.numeric(college_statistics$Top25perc)<20]=
"laag"
college_statistics$catTop25perc[as.numeric(college_statistics$Top25perc)>=20
& as.numeric(college_statistics$Top25perc)<=40]="midden"
college_statistics$catTop25perc[as.numeric(college_statistics$Top25perc)>40]=
"hoog"
college_statistics$catTop25perc <- as.factor(college_statistics$catTop25perc)
```

5 (b) Voer de ANOVA analyse uit en geef je conclusie(s). Presenteer de ANOVA resultaten zowel numeriek als grafisch.

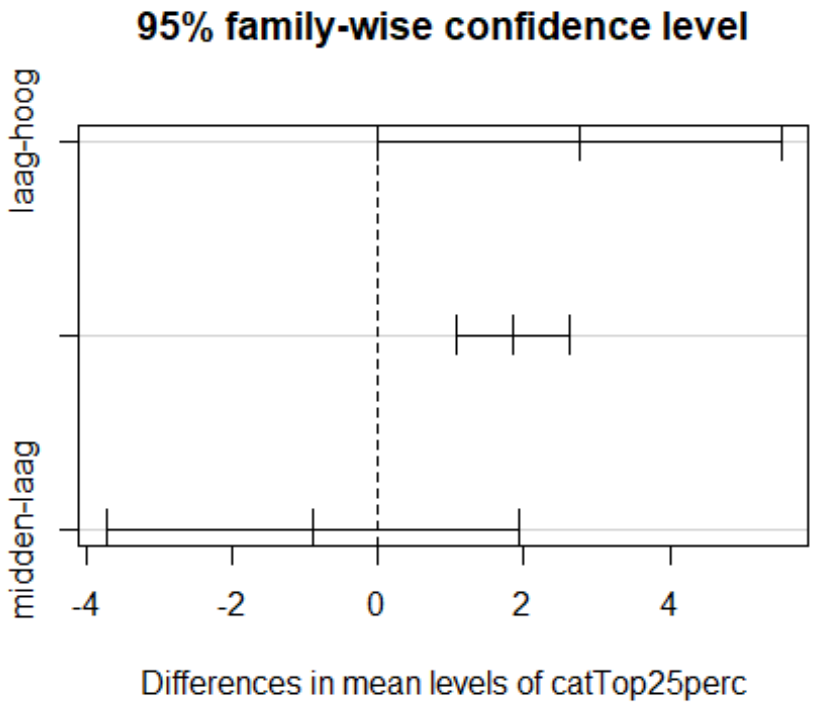
First, we do a One-way ANOVA with catTop25perc on S.F.Ratio

```
aov1 <- aov(S.F.Ratio~catTop25perc, data=college_statistics)
summary(aov1)

##               Df Sum Sq Mean Sq F value   Pr(>F)
## catTop25perc   2    533  266.40   17.74 2.94e-08 ***
## Residuals    774  11626   15.02
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H0 : The mean of the dependent variable is the same across all groups We can reject the null hypothesis (p-value < 0.05), thus there are significant differences between groups.

Some visualizations of the results.

```
TukeyHSD(aov1)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = S.F.Ratio ~ catTop25perc, data = college_statistics)
##
## $catTop25perc
##                   diff         lwr      upr     p adj
## laag-hoog    2.7520788 -0.01757942 5.521737 0.0519049
## midden-hoog  1.8565738  1.08101717 2.632130 0.0000001
## midden-laag -0.8955051 -3.72213044 1.931120 0.7373750

plot(TukeyHSD(aov1))
```



5 (c) Onderzoek of er outliers zijn. Pas de analyse aan als dat nodig is.

```
outlierTest(aov1)

##                              rstudent unadjusted p-value Bonferroni p
## Indiana Wesleyan University 6.965011         7.0276e-12   5.4605e-09
```

We find an outlier for Indiana Wesleyan University. Let's drop this observation and try again.

```
college_statistics_outlier <- college_statistics[rownames(college_statistics)
!= 'Indiana Wesleyan University',]
```

Run the model again.

```
aov2 <- aov(S.F.Ratio~catTop25perc, data=college_statistics_outlier)
summary(aov2)

##               Df Sum Sq Mean Sq F value   Pr(>F)
## catTop25perc   2    557  278.73    19.7 4.54e-09 ***
## Residuals    773  10939   14.15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H0 : The mean of the dependent variable is the same across all groups Again, We can reject the null hypothesis (p-value < 0.05), thus there are significant differences between groups.

Some visuals:

```
TukeyHSD(aov2)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = S.F.Ratio ~ catTop25perc, data =
college_statistics_outlier)
##
## $catTop25perc
##                   diff        lwr      upr     p adj
## laag-hoog    2.7968298  0.1084056 5.485254 0.0392212
## midden-hoog  1.9013248  1.1483759 2.654274 0.0000000
## midden-laag -0.8955051 -3.6391823 1.848172 0.7237357

plot(TukeyHSD(aov2))
```