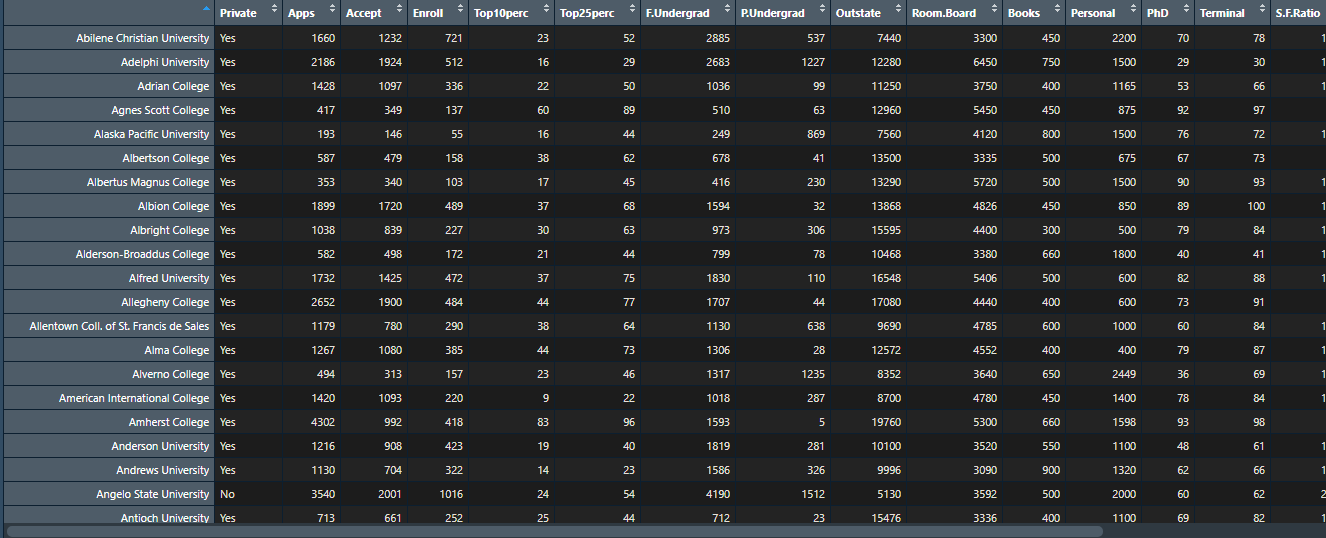
**Mini Project 1- College Dataset- Linear Regression**

>> install.packages("ISLR") >> library(ISLR) >> View(College)

Our dataset was present in ISLR package. So, we first installed the package and then extracted the dataset from it by the above line of codes.

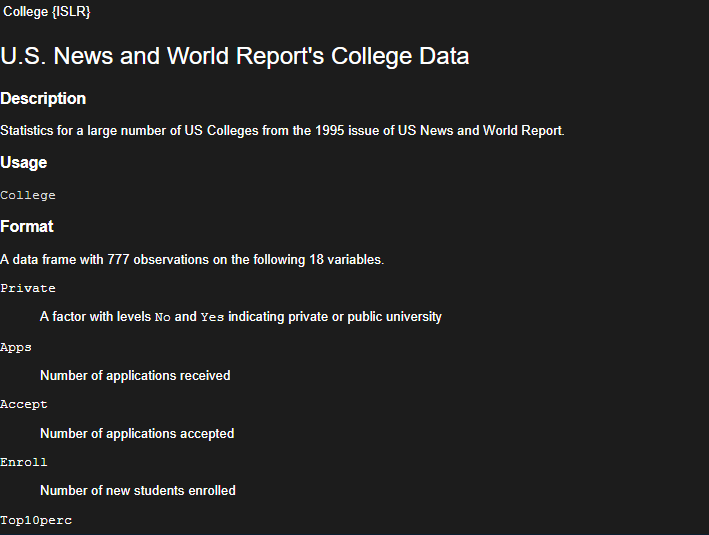
**View()** function visualizes the dataset in a table format, helping the user to have glimpse of the total number of entries that is there in the dataset along with all the columns/variables of the dataset.



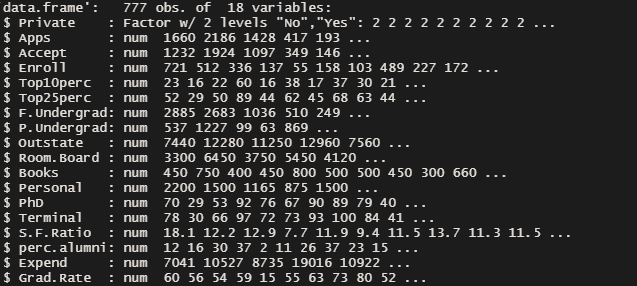
>>help(College) >>str(College) >>dim(College) >>names(College)

We also used few basic operational functions to get some knowledge/overview of the dataset that we have to work on.

**help(College)** function gave us the information about the dataset like its source, meaning of all the variables that are there in the dataset, description of the dataset etc.



**str(College)** function gave the structure of the dataset telling that it is a data frame with 777 rows and 18 columns. It also gave information about the class of each variable of the dataset i.e. whether it is a factor or a numeric data.



**dim(College)** function again gave us the information about the number of rows and columns🡪 **[777 x 18]**

**names(College)** function displayed the names of all the variables present in the dataset.

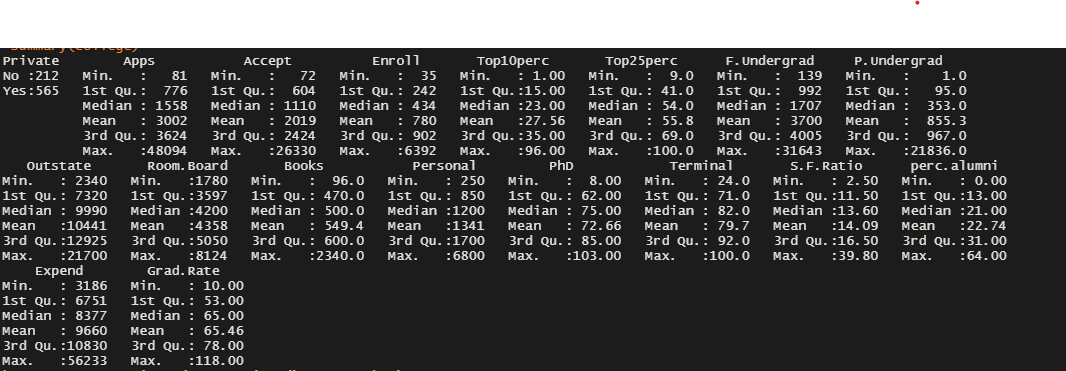
>> sum(is.na(College)) >>class(College[,1])

**sum(is.na(College))**  The above line of code was used to check the presence of NA/ NULL values in the dataset, which came out to be zero.

**class(College[,1])** displayed the category of 1st variable of College dataset i.e. “ Apps” which is a factor.

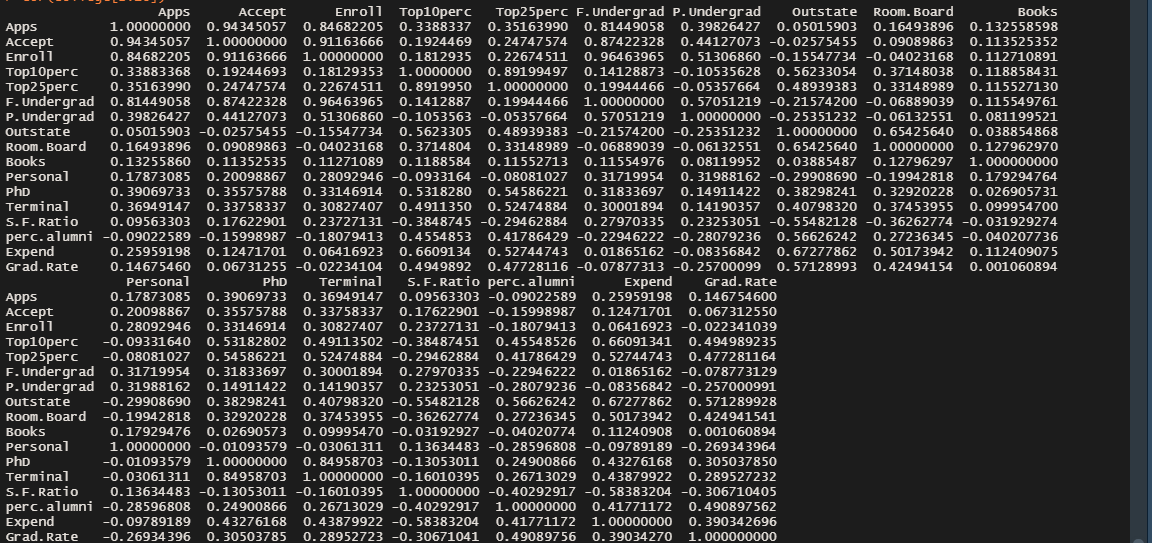
>> summary (College) >> cor(College[2:18])

**summary (College)** Summary of a dataset provided all the statistical information about each variable of the dataset like its mean, median, minimum value, maximum value, 1st quartile and 3rd quartile.



**cor(College[2:18])** function displayed the correlation between respective variables.

The value of correlation lies between -1 to 1 (-1 being lowest and negatively correlated and 1 being highest and highly correlated, If the value is 0, the two variables are independent and there is no correlation.) indicating how the variables are related to each other or the extent to which two variables are linearly related (meaning they change together at a constant rate).

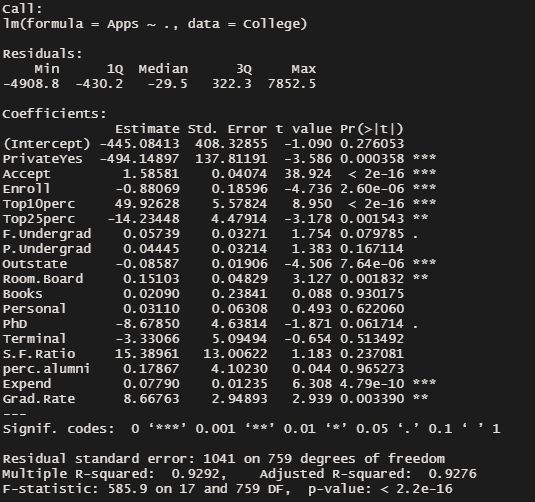


>> attach(College)

with the help of attach() function the College dataset was attached to the R search path. This means that the database is searched by R when evaluating a variable, so objects in the database can be accessed by simply giving their names.

>> full.model =lm(Apps~., data= College) >> summary(full.model)

Applied lm() function on the entire dataset and with the help of summary() function we got to know the following things about our model:



1. Adjusted R-squared: 0.9276 i.e. independent variables are able to explain 92.76% of dependent variable- “Apps”
2. the p-value for most of the variables are less than 0.05 and hence significant except for the “intercept” (i.e. β0) , P.Undergrad, Books, Personal, Terminal, S.F.Ratio and perc.alumni.
3. Overall p-value of the model is also less than 0.05, hence the model is fit to predict the values of dependent variable.

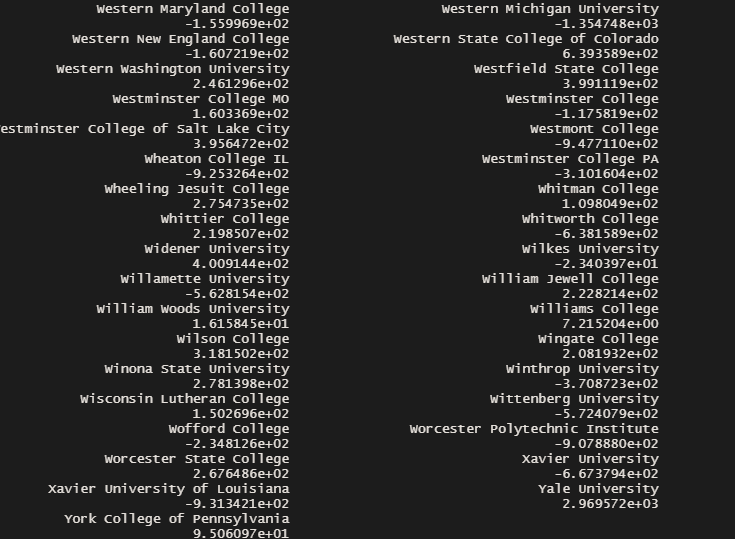
>> predicted\_y= predict(full.model); predicted\_y

Predict() function was used to predict the values of our response variable.



>> residuals(full.model)

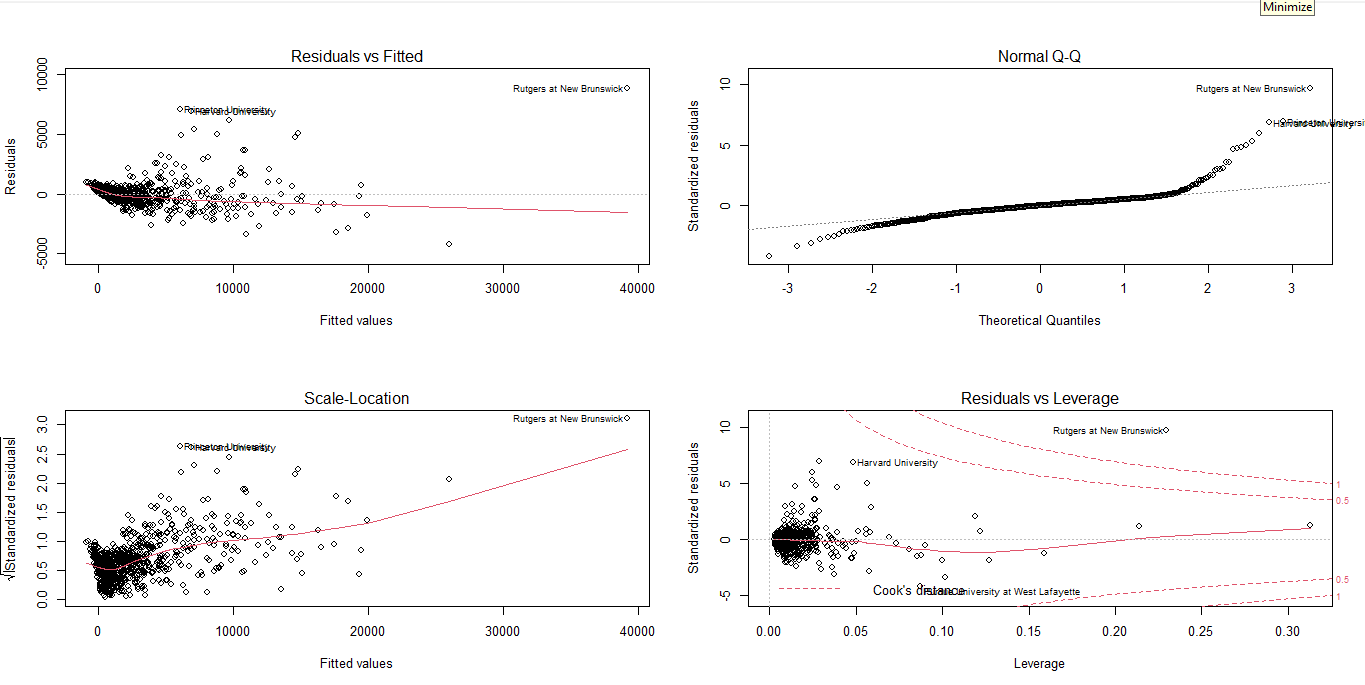
This function tells how different are our predicted values are from the actual y i.e. (y’-y)



>> windows()

>> par(mfrow=c(2,2))

>> plot(reg)



Plot 1 This scatter plot shows the distribution of residuals (errors) vs fitted values (predicted values). It reveals various useful insights including outliers, presence of non-linearity and heteroskedasticity

Here no pattern is evident

Plot 2 This q-q or quantile-quantile is a scatter plot which helps us validate the assumption of normal distribution of residuals in a data set. Using this plot, we can infer if the data comes from a normal distribution

The residuals are not normally distributed

Plot 3 This plot is also used to detect homoskedasticity (assumption of equal variance). It shows how the residual are spread along the range of predictors. It’s similar to residual vs fitted value plot except it uses standardized residual values.

Heteroskedasticity is evident

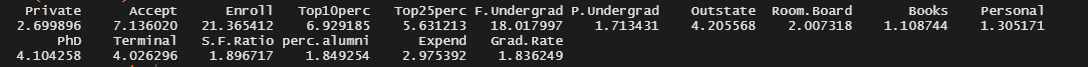
Plot 4 It is also known as Cook’s Distance plot. Cook’s distance attempts to identify the points which have more influence than other points. Such influential points tend to have a sizable impact on the regression line. [points lying above those dotted lines are called influential points that will impact our model by the value mentioned in front of that dotted line i.e 0.5 and 1]

Here we can see that the outliers are present

>> install.packages("car") >>library(car)

In order to use vif() function we needed to install “car ” package

>> vif(full.model)



VIF is used to identify the correlation of one independent variable with a group of other variables. All the variables have VIF values either less than 4 or between 4 to 10 showing less multicollinearity except for the two viz: Enroll(21.365412), F.Undergrad(18.017997). Therefore, the p-value and significance of these independent variables can’t be trusted.

[[We could have simply removed variable- “Enroll” and run the model again to check its significance, but this step was quite time consuming hence we went forth with another approach i.e asking the system to make a significant model for us by the following two approaches:]]

**Approach-1**

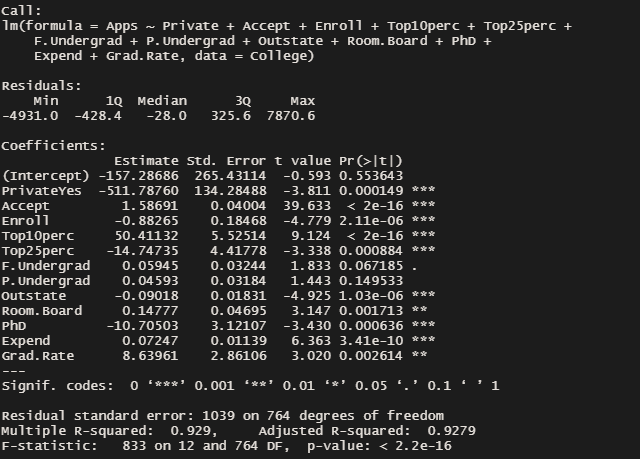
>> install.packages("MASS") >>library(MASS)

We needed MASS library to make use of a function names stepAIC() which chooses the best model by AIC(Akaike information criterion) . It has an option named direction, which can take the following values: “both” (for stepwise regression, both forward and backward selection); “backward” (for backward selection) and “forward” (for forward selection). We used “both” as our value and It returns the best final model containing the most significant (12) independent variables.

>> step.model <- stepAIC(full.model, direction = "both",

trace = FALSE)

>> summary(step.model)



This model was able to explain 92.79% of response variable and all predictors except two (intercept and P.Undergrad) are considered to be significant as there p-value is less than 0.05.

>> vif(step.model)

Although in this case too vif value of Enroll and F.Undergrad was quite high but our adjusted R^2 raised, so we accepted this model as a good fit for our prediction.

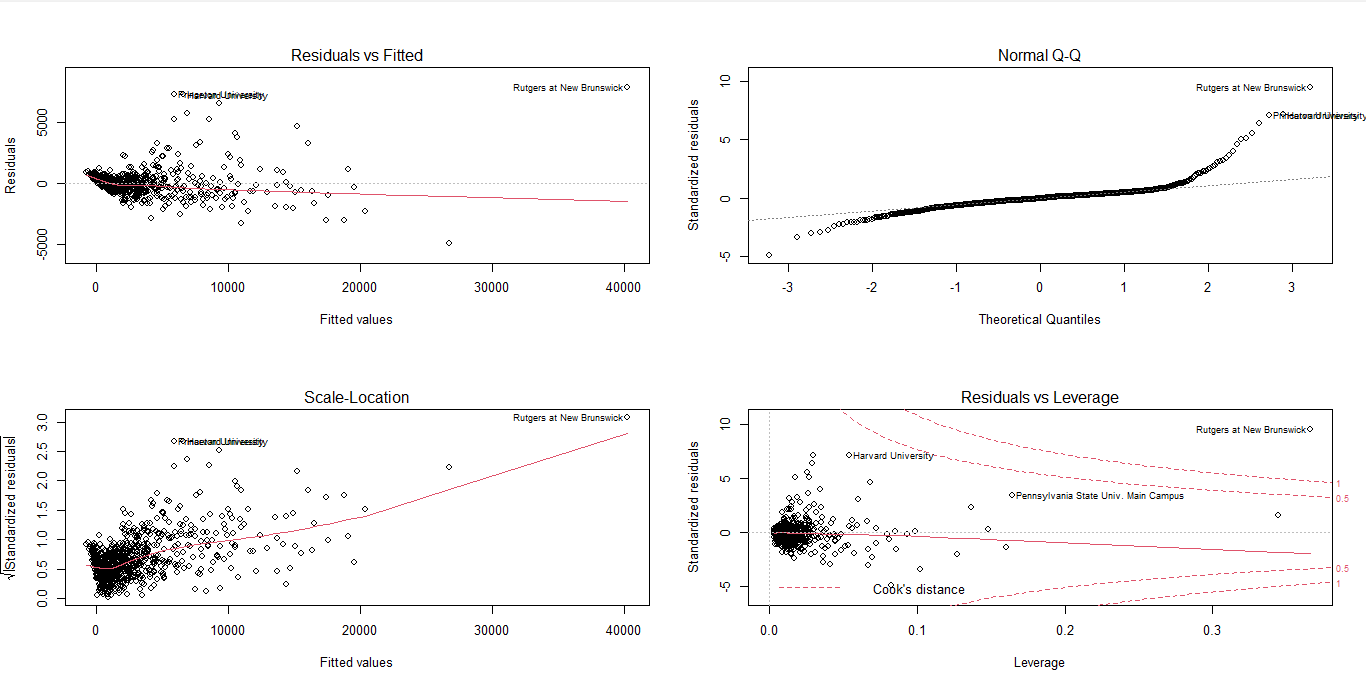
>> predicted\_y1= predict(step.model.reg); predicted\_y1 >> residuals(step.model)

Predicted values of response variable and their variance from actual y variable as done before for full model

>> windows()

>> par(mfrow=c(2,2))

>> plot(step.model)

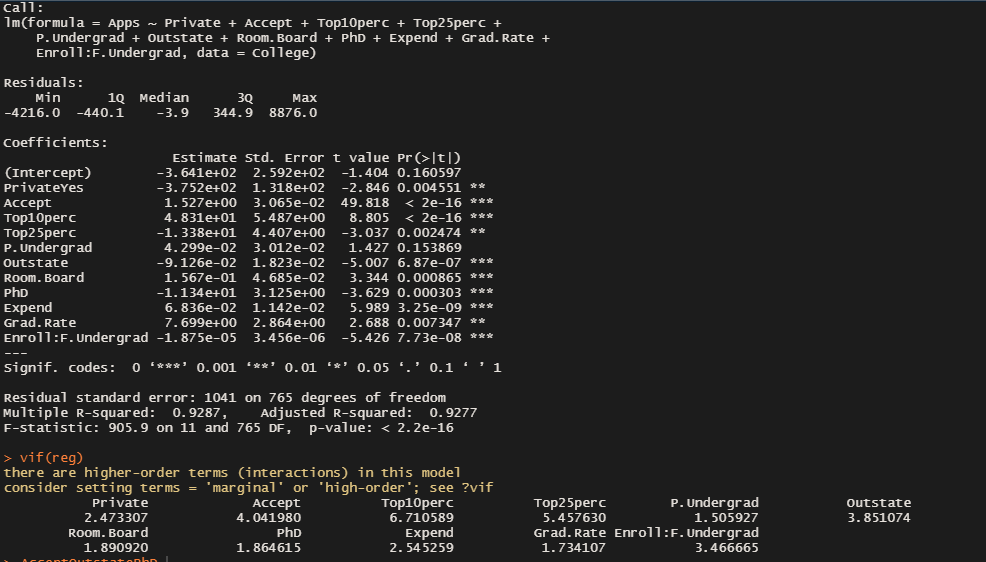


[[Further on we tested various other models by combining Enroll and F.Undergrad and removing either one of them respectively, each time although the vif value came down to a significant level but our adjusted R^2 reduced to 0.9277]]

>> reg= lm(Apps~Private+Accept+Top10perc+Top25perc+P.Undergrad+Outstate+Room.Board +PhD +Expend +Grad.Rate+Enroll:F.Undergrad, data= College)

>> summary(reg)

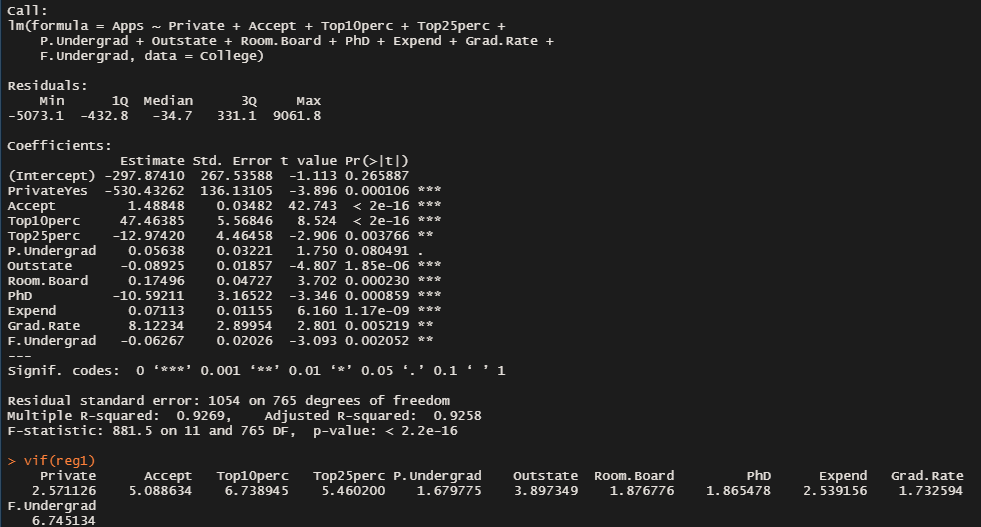
>> vif(reg)



>> reg1= lm(Apps~Private+Accept+Top10perc+Top25perc+P.Undergrad+Outstate+Room.Board +PhD +Expend +Grad.Rate+F.Undergrad, data= College)

>> summary(reg1)

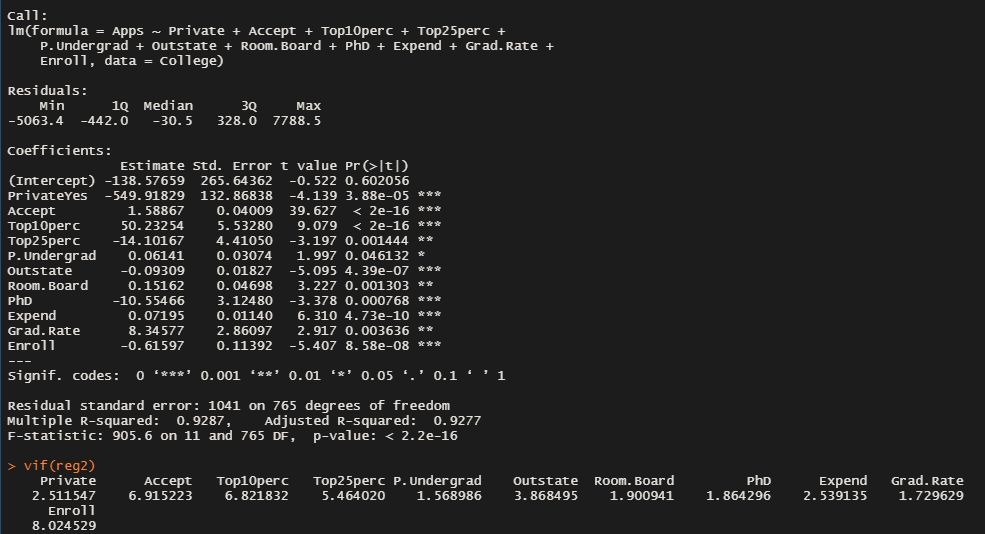
>> vif(reg1)



>> reg2= lm(Apps~Private+Accept+Top10perc+Top25perc+P.Undergrad+Outstate+Room.Board +PhD +Expend +Grad.Rate+Enroll, data= College)

>> summary(reg2)

>> vif(reg2)



***[[Comparing all the three models we got to know that “reg” has adjusted R^2 value as 0.9277, p-value of intercept and P.Undergrad close to 0.05 and vif values of all the variables significant enough to consider a good fit for prediction.]]***

**Approach-2**

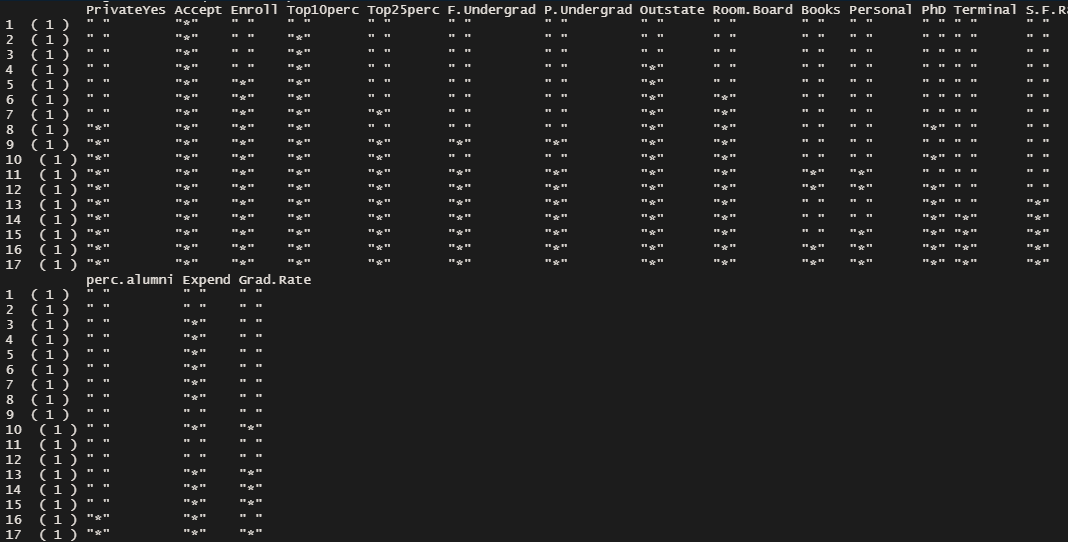
>> regressionfits= regsubsets(Apps~., data=College, method ="seqrep", nvmax = 17)

>> regressionfits.summary = summary(regressionfits);regressionfits.summary

It returns multiple models with different size up to nvmax() [a tuning parameter used to specify the the maximal number of predictors to incorporate in the model].

You need to compare the performance of the different models for choosing the best one.

regsubsets() has the option method, which can take the values “backward”, “forward” and “seqrep” (seqrep = sequential replacement, combination of forward and backward selections).



>> names(regressionfits.summary)

As mentioned above regsubsets() has these following performace measures to estimate which model id the best fir to predict the values of our response variable.

**which**: tells which all predictors we have taken for different nvmax values and provides the output in a tabular format with Boolean output [True for the variables that ae included in that model and False for those that are not included]

**rsq**: gives the r-square value for different models [the greater the better]

**rss**: gives residual sum of square values for different models [the lower the better]

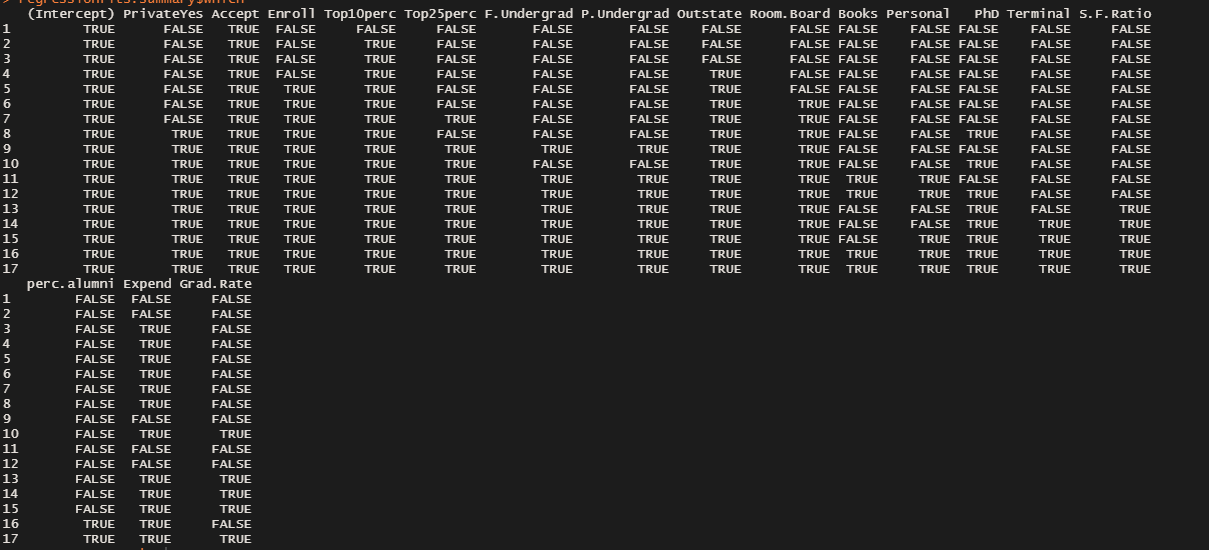
**adjr2**: gives adjusted r-square values for different models [the greater the better]

**cp or AIC**: Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection.

If a model is more than 2 AIC units lower than another, then it is considered significantly better than that model. [the lower the better]

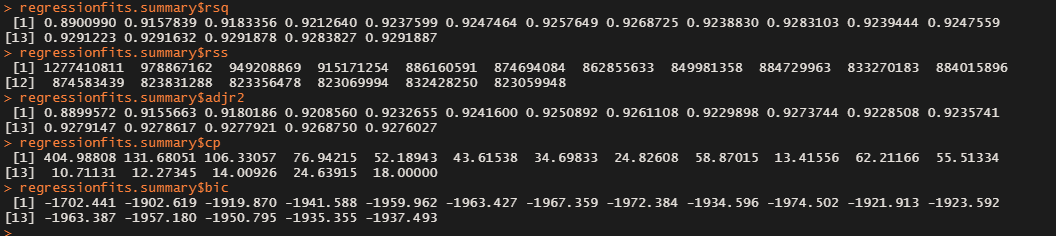


>> regressionfits.summary$which



>> regressionfits.summary$rsq >> regressionfits.summary$rss >>regressionfits.summary$adjr2 >> regressionfits.summary$cp ( AIC

)>> regressionfits.summary$bic

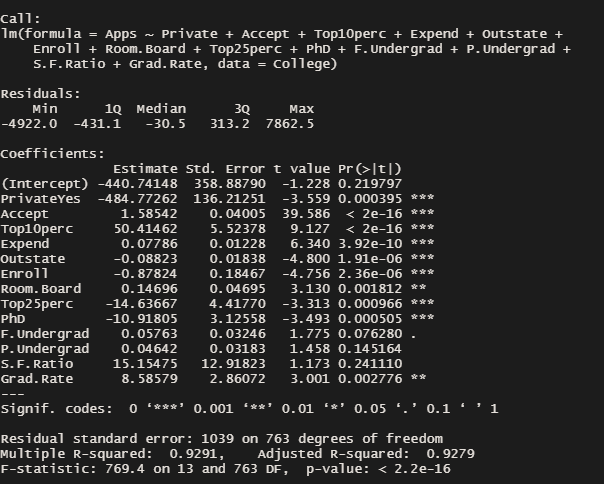


On the basis of above-mentioned performance metrics, we got to a conclusion that model with nvmax value 13 is the best fir model for prediction of response variable

>> regsubset.model.reg= lm(Apps~ Private+Accept+Top10perc+Expend+Outstate+Enroll+Room.Board+Top25perc+PhD+ F.Undergrad+ P.Undergrad+ S.F.Ratio+ Grad.Rate, data=College )

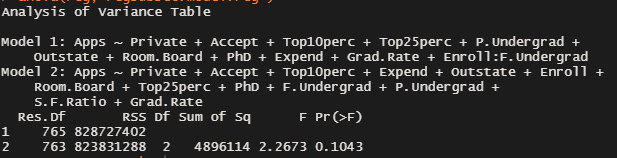
>> summary(regsubset.model.reg)

Created the model using the inputs from above performance measures containing 13 appropriate variables for a precise prediction model



>> anova(regsubset.model.reg, reg )

Compared the two models created using two different approaches to check if they vary from each other or not



From the p-value [>0.05] of F-statistic value [2.2673] we can interpret that there is not much difference in both the models.