Analysis of College Tuitions

Malcolm Hess

Thursday, April 09, 2015

Contributed by Malcolm Hess. This is based on my project investigating college tuitions.

Links

The code and files for this project can be found here: https://github.com/Mal-Hess/Colleges

If you enjoyed this or would like to see more of my work feel free to check out my github account for more projects.
https://github.com/Mal-Hess/

Background

There is increasing importance in our society to have a college degree from a top university. However the costs for attending a university across the country have been drasticly increasing. This has made going to college for some simply a dream and many that do go to college graduate with massive students debts. This project is designed to analyze what variables if any effect college attendance costs and would be good predictors in estimating tuition.

Objective

- Find a list of top colleges in the U.S. and download variables that can be used to analyze the cost of tuition.
- Clean and organize the data.
- Check for multi-colinearity of variables.

- Build a linear model to see which varaibles if any are significant predictors of tuition cost.
- Perform a PCA analysis of the variables and build a PCA regression model.

Implementation

There are many different college ranking sites and after looking at many options I decided to use Forbe's list of top colleges. I used Forbe's list and data because I feel Forbes is a trustworthy source.

From the site I was able to get the names, ranks, and also links to the individual pages for each of the top 500 colleges on the Forbe's list. Below you can see the link to the top colleges list.

http://www.forbes.com/top-colleges/

Forbes also has a dedicated page for every college in their list. I web scraped data off of these dedicated pages to create my data frame. Below you can see the link of what one of these individual sites looks like.

http://www.forbes.com/colleges/stanford-university/

Conclusion

There are definitely some signficant variables that can be used to predict and estimate the price of a college. Several commonly held beliefs are validated with this research. Most students would rather go to a cheaper school but they are very willing to pay more in order to go to a more challenging, competetive, and desirable school. Unfortunately, more expensive universities also require students receive larger grants and take on bigger loans.

The variables that came out significant in my linear model are:

- %acceptedenrolled :Lower % accepted that enroll leads to higher costs.
- studenttofacultyratio :lower student to faculty ratio leads to increased costs.
- fouryeargrad : Schools with a higher four year graduation rate cost more than low graduation rate universities.

- %onfinancialaid : As the cost of a school increases the amount of students that will be on financial aid increases.
- averageloan : More expensive shcools means that students need bigger loans
- averagegrant : More expensive schools leads to students needing larger grants
- % athlete : Colleges with a lower % of student athletes cost more.

Getting the Data

The list of top colleges is broken down so that there is a page per 100 colleges. I downloaded the first 5 pages manually. Using x-paths I took out the names, links, and ranks for the top 500 colleges. At the end I also saved that object and along with other objects to disk. I have also uploaded these binary data files to my github page so anyone who wishes to recreate this project will not have to download the sites and web scrape them.

```
## Loading required package: XML
## Loading required package: RCurl
## Loading required package: bitops
##
## Attaching package: 'rvest'
##
## The following object is masked from 'package:XML':
##
##     xml
##
##
## Attaching package: 'pls'
##
## The following object is masked from 'package:stats':
```

```
##
##     loadings
```

```
#name of downloaded sites that have Forbes' top colleges lists.
listofcolleges <- c("Forbes1.html", "Forbes2.html", "Forbes3.html", "Forbes4.html", "Forbes5.html")

topcolleges <- NULL

for(i in listofcolleges){

tables <- readHTMLTable(i)
df <- data.frame(tables[27])

parsed <- htmlParse(i)
xpath <- '//*[@id="listbody"]/tr/td/a/@href'
nodes <- getNodeSet(parsed, xpath)
links <- sapply(nodes, toString)



top <- cbind(df, links)
topcolleges <- rbind(topcolleges, top)
}

topcolleges <- topcolleges[,c(-4,-5)]
names(topcolleges) <- c("rank", "name", "state", "full_link")

topcolleges$full_link <- gsub("\\?list=top-colleges", "", topcolleges$full_link)
```

```
#saving topcolleges object:
#save(topcolleges, file="list of top colleges")
#load("list of top colleges")
```

The extractdata function has individual xpaths for all of the information I chose to remove from the individual college sites. This method does require that the sites be uniformly made. The only individual sites that are not uniform have no tuition cost. The extractdata function will be run on those sites too but they will be removed later since they are of no use to my analysis. There are 39 variables in total that are collected. I saved the object 'mat' which has all of these variables to disk.

Below are a few examples of how the extract data function works.

```
extractdata <- function(data) {

  vect <- c()

  #total cost-in state
  xpath <- '//*[@id="tuition"]/ul/li/table[3]/tbody/tr[2]/td[2]'
  nodes <- getNodeSet(data, xpath)
  value <- sapply(nodes, xmlValue)
  if(!is.null(value)){
    vect[1] <- toString(value)
  }else {vect[1] <- NA}

  #total cost-out state
  xpath <- '//*[@id="tuition"]/ul/li/table[3]/tbody/tr[2]/td[3]'
  nodes <- getNodeSet(data, xpath)
  value <- sapply(nodes, xmlValue)
  if(!is.null(value)){
```

```
    vect[2] <- toString(value)
  }else {vect[2] <- NA}

    #Grand total sports revenues
  xpath <- '//*[@id="athletics"]/ul/li/table[4]/tbody/tr[5]/td[4]'
  nodes <- getNodeSet(data, xpath)
  value <- sapply(nodes, xmlValue)
 if(!is.null(value)){
   vect[39] <- toString(value)
 }else {vect[39] <- NA}

 vect <- gsub("\n", "", vect)
 vect <- gsub("\t", "", vect)

  vect
} #End of extract data function

#Download all of the individual college websites.
mat <- matrix(NA, nrow=500, ncol=39)
i <- 1
while(i <= nrow(topcolleges)){

  collegeurl <- topcolleges$full_link[i]
  rawcollege <- getURL(collegeurl)
  nameoffile <- paste0(toString(topcolleges$name[i]), ".Rdata")
  save(rawcollege, file=nameoffile)

  parsed <- htmlParse(rawcollege)
```

```
  mat[i,] <- extractdata(parsed)

i <- i + 1
} #End of while loop

#save(mat, file="variables.Rdata")
#load("variables.Rdata")
```

Cleaning the Data

There was plenty of cleaning to do. Commas and dollar signs had to be removed. Many of the variables I collected were percents, which needed to have the percent symbol removed and then be converted to the actual value.

```
mat$rank <- gsub("#", "", mat$rank)
# returns string w/o leading or trailing whitespace
trim <- function (x) gsub("^\\s+|\\s+$", "", x)
mat$rank <- trim(mat$rank)

topcolleges$rank <- sapply(topcolleges$rank, toString)

mat <- mat[order(mat$rank),]
topcolleges <- topcolleges[order(topcolleges$rank),]

final <- merge(topcolleges, mat, by="rank")

final <- sapply(final, function(x)sub("\\$","", x))
final[,5:42] <- sapply(final[,5:42], function(x)sub("\\,","", x))
final[,5:42] <- sapply(final[,5:42], function(x)sub("\\,","", x))
```

```
percents <- c(12,13,14,15,16,17,18,19, 20,21,22,23,24,25,29 ,33,34,41)

final[,percents] <- sapply(final[,percents], function(x)sub("\\%","", x))

arenumbers <- c(1, 5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,
36,41,42)
arefactors <- c(1, 2,3,37,38,39,40)

final <- as.data.frame(final, stringsAsFactors = FALSE)
final[,arenumbers] <- sapply(final[,arenumbers], as.numeric)
final[,percents] <- sapply(final[,percents], function(x) x/100)

final <- final[order(final$rank),]
final[,arefactors] <- sapply(final[,arefactors], factor)
```

Checking for Multicolinearity

I checked both the covariance and correlation matrix to see if there was any multicolinearity in my data. I expected that there would be more than a few since many of the variables are directly dependent on the same things, for example the cost of in-state schooling and out-of-state schooling. For most universities these are exactly the same; universities where these two costs differ are usually state or city funded schools.

Below, I have highlighted some groups of variables which have a lot of multi-colinearity. I used this information to perform some variable selection on my own by removing variables that had high correlation.

```
load("finaltable.Rdata")
costs <- c(5,8,10,11,15,16)
```

```r
corr1 <- cor(final[,costs],  use = "complete", method = c("pearson", "kendall", "spearman"))

morecovariance <- c(6,32,33,34,35,36,41,42)

corr2 <- cor(final[,morecovariance],  use = "complete", method = c("pearson", "kendall", "spearman"))

corr1
```

```
##                     totalcostinstate undergradpopulation instatetuition
## totalcostinstate           1.0000000          -0.5922777      0.9848453
## undergradpopulation       -0.5922777           1.0000000     -0.6365261
## instatetuition             0.9848453          -0.6365261      1.0000000
## outofstatetuition          0.8986981          -0.4030340      0.8967617
## fulltime                   0.4055471          -0.2863896      0.4324222
## fouryeargrad               0.7213045          -0.4169195      0.7425370
##                     outofstatetuition   fulltime fouryeargrad
## totalcostinstate            0.8986981  0.4055471    0.7213045
## undergradpopulation        -0.4030340 -0.2863896   -0.4169195
## instatetuition              0.8967617  0.4324222    0.7425370
## outofstatetuition           1.0000000  0.4876179    0.7701776
## fulltime                    0.4876179  1.0000000    0.5955393
## fouryeargrad                0.7701776  0.5955393    1.0000000
```

```r
corr2
```

```
##                     totalcostoutstate amountofapplicants     accepted
```

```
## totalcostoutstate         1.00000000          0.1265282 -0.464606399
## amountofapplicants        0.12652816          1.0000000 -0.425793173
## accepted                 -0.46460640         -0.4257932  1.000000000
## acceptedenrolled         -0.15610539          0.1203801 -0.322990640
## sat25                     0.62396060          0.3247551 -0.731295088
## sat75                     0.58940133          0.3189311 -0.665936954
## athlete                   0.25650270         -0.5718996 -0.007584823
## sportsrevenue            -0.07253888          0.5590096 -0.087418206
##                    acceptedenrolled       sat25       sat75      athlete
## totalcostoutstate        -0.1561054  0.62396060  0.58940133  0.256502696
## amountofapplicants        0.1203801  0.32475513  0.31893110 -0.571899603
## accepted                 -0.3229906 -0.73129509 -0.66593695 -0.007584823
## acceptedenrolled          1.0000000  0.31533610  0.35023960 -0.138028775
## sat25                     0.3153361  1.00000000  0.95373535  0.012719589
## sat75                     0.3502396  0.95373535  1.00000000  0.035014489
## athlete                  -0.1380288  0.01271959  0.03501449  1.000000000
## sportsrevenue             0.3256454  0.17851701  0.20500684 -0.486194946
##                    sportsrevenue
## totalcostoutstate    -0.07253888
## amountofapplicants    0.55900965
## accepted             -0.08741821
## acceptedenrolled      0.32564545
## sat25                 0.17851701
## sat75                 0.20500684
## athlete              -0.48619495
## sportsrevenue         1.00000000
```

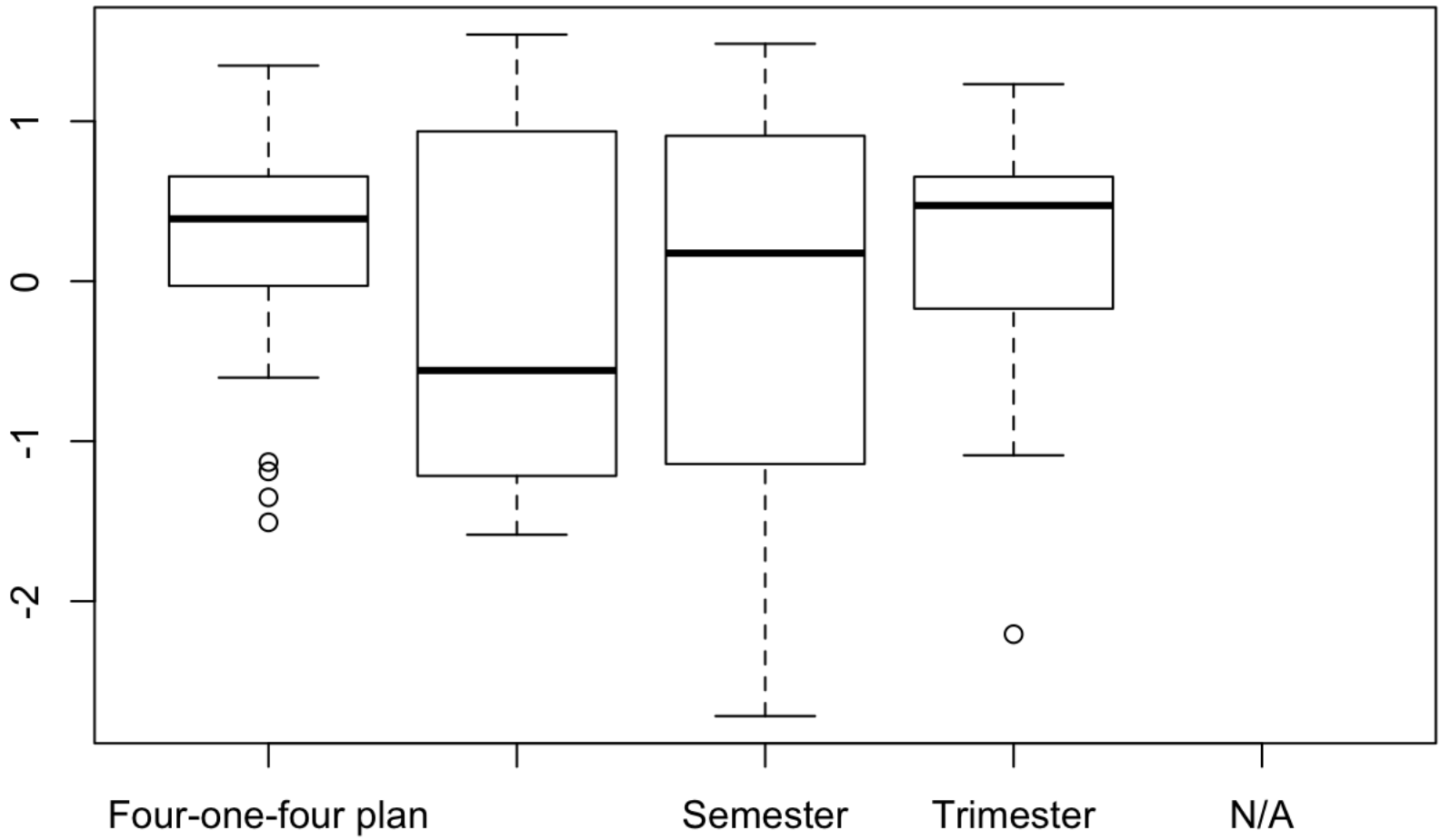
The next step was to center and standardize all of the continous variables. They were scaled to have a mean zero and

standard deviation of 1.

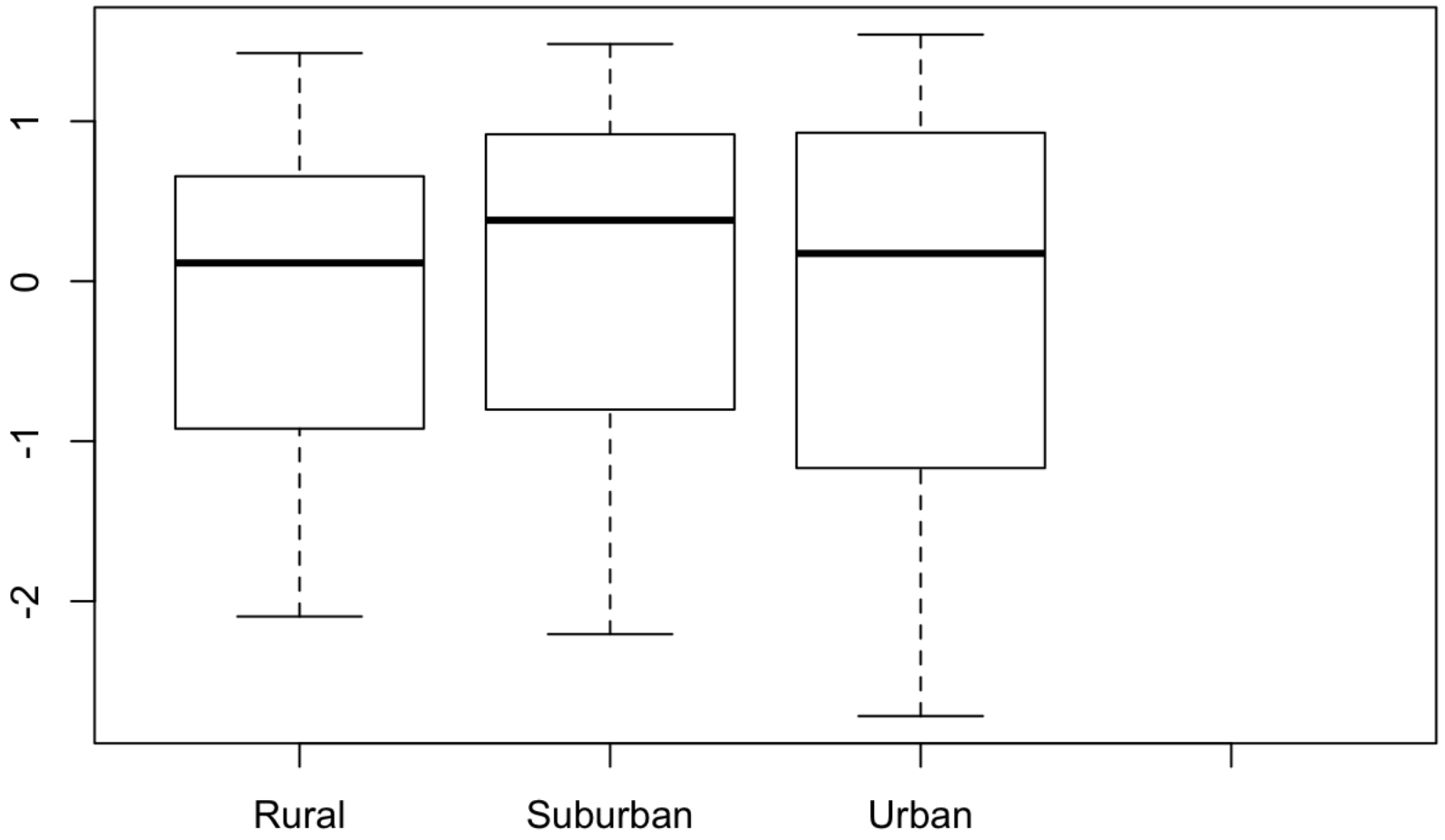
```
normality <- c()

withoutrank <- c(5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36
,41,42)


for(i in withoutrank){
  final[,i] <- scale(final[,i])
}

final$calendar <- sapply(final$calendar, as.factor)
#final$campus_housing <- sapply(final$campushousing, as.factor)
final$setting <- sapply(final$setting, as.factor)
final$religion <- sapply(final$religion, as.factor)
final$state <- sapply(final$state, as.factor)
final$rank <- sapply(final$state, as.numeric)

#save(final, file="finaltable.Rdata")
```

I checked the factor variables, such as calendar and setting, with boxplots. None of them appeared to have significant differences between the groups.

```
load("finaltable.Rdata")
boxplot(totalcostinstate ~ calendar, data=final)
```




```
boxplot(totalcostinstate ~ setting, data=final)
```



Building a Linear Model

I furthered my analysis by creating linear models. I analyzed many and tried a multitude of different approaches particularly

in how I paired variables together interaction terms. Eventually I settled on a varaible combination I called basemodel.

9 variables came out significant:

- %acceptedenrolled
- studenttofacultyratio
- fouryeargrad
- %onfinancialaid
- averageloan
- averagegrant
- % athlete
- studenttofacultyratio:fouryeargrad
- averageloan:averagegrant

```
final_vars <- final[,c(1, 3, 5,7, 9, 12,13, 14,15,16,17,22,28,29,30,31,32,33,34,35,36,37,40,41,42)]

final_complete <- final_vars[!is.na(final_vars$totalcostinstate),]

allmodel <- lm(totalcostinstate ~ ., data=final_complete, na.action=na.omit)

nostate <- lm(totalcostinstate ~ . -state , data=final_complete, na.action=na.omit)

basemodel <- lm(totalcostinstate ~ rank + acceptedenrolled +  studentpopulation * amountofapplicants + subm
itttingsat +
                  studenttofacultyratio * fouryeargrad +  onfinancialaid * recevingloan + fulltime + female
+ white +
                  averageloan * averagegrant + applicationfee + state +
                sat25 * admitted + calendar+ setting + athlete + sportsrevenue, data=final_complete, na.act
ion=na.omit)
```

```
basenostate <- lm(totalcostinstate ~ rank + acceptedenrolled +  studentpopulation * amountofapplicants + su
bmitttingsat +
                  studenttofacultyratio * fouryeargrad +  onfinancialaid * recevingloan + fulltime + female
+ white +
                  averageloan * averagegrant + applicationfee +
                  sat25 * admitted + calendar+ setting + athlete + sportsrevenue, data=final_complete, na.a
ction=na.omit)
```

```
summary(allmodel)
```

```
##
## Call:
## lm(formula = totalcostinstate ~ ., data = final_complete, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68216 -0.19032 -0.00598  0.19507  1.13439
##
## Coefficients: (2 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.1625175  0.1255968   1.294  0.19673
## rank                       0.0045464  0.0069237   0.657  0.51194
## stateCalifornia            0.1246947  0.1087017   1.147  0.25229
## statePennsylvania          0.0144220  0.1089776   0.132  0.89481
## stateNew Jersey           -0.1839316  0.1444795  -1.273  0.20403
## stateConnecticut          -0.2034613  0.1664969  -1.222  0.22271
```

```
## stateNew York             -0.0263634  0.1022460  -0.258  0.79671
## stateRhode Island          0.0468045  0.1801038   0.260  0.79515
## stateMaine                -0.1519118  0.2068253  -0.734  0.46325
## stateMinnesota            -0.2382343  0.1979615  -1.203  0.22981
## stateIndiana              -0.3225994  0.1544326  -2.089  0.03760 *
## stateNew Hampshire        -0.1992118  0.2081800  -0.957  0.33942
## stateIllinois              0.1491222  0.1851869   0.805  0.42135
## stateNorth Carolina       -0.3556972  0.1537357  -2.314  0.02140 *
## stateMaryland             -0.2996135  0.1589441  -1.885  0.06044 .
## stateDistrict of Columbia  0.2193198  0.1856980   1.181  0.23856
## stateVermont              -0.4438346  0.2192118  -2.025  0.04383 *
## stateTexas                -0.1399286  0.1566849  -0.893  0.37258
## stateVirginia             -0.3942696  0.1573162  -2.506  0.01276 *
## stateColorado             -0.0820226  0.2027070  -0.405  0.68605
## stateOhio                 -0.1838027  0.1785850  -1.029  0.30425
## stateMichigan             -0.3843698  0.2262806  -1.699  0.09048 .
## stateOregon                0.0089836  0.1969659   0.046  0.96365
## stateWashington            0.0208630  0.1961795   0.106  0.91538
## stateTennessee            -0.2827363  0.2388965  -1.184  0.23759
## stateIowa                 -0.3874836  0.2296289  -1.687  0.09261 .
## stateMissouri             -0.0843239  0.2475107  -0.341  0.73359
## stateWisconsin            -0.2653907  0.3105398  -0.855  0.39348
## stateUtah                 -0.1979740  0.2880661  -0.687  0.49248
## stateKentucky             -0.4148054  0.2722705  -1.524  0.12874
## stateSouth Carolina       -0.5395040  0.2308142  -2.337  0.02011 *
## stateFlorida              -0.0889259  0.2349601  -0.378  0.70536
## stateGeorgia              -0.4240009  0.2391391  -1.773  0.07729 .
## stateDelaware             -0.7755576  0.3938953  -1.969  0.04993 *
```

```
## stateLouisiana            -0.1443989  0.3136105  -0.460  0.64555
## stateArkansas             -0.6344190  0.2945151  -2.154  0.03207 *
## stateNebraska             -0.1604930  0.3174616  -0.506  0.61356
## stateWyoming              -0.6877355  0.4182051  -1.644  0.10118
## stateOklahoma             -0.1207072  0.3213825  -0.376  0.70750
## stateAlabama              -0.5567694  0.3006045  -1.852  0.06504 .
## stateMontana              -0.4388404  0.3269618  -1.342  0.18061
## stateArizona              -0.1216088  0.3618171  -0.336  0.73704
## stateIdaho                -0.4396005  0.4239943  -1.037  0.30070
## stateMississippi          -0.2878885  0.3809512  -0.756  0.45045
## stateNew Mexico           -0.0890436  0.3902759  -0.228  0.81969
## stateNorth Dakota         -0.1711745  0.3915644  -0.437  0.66233
## stateSouth Dakota         -0.2407394  0.4086379  -0.589  0.55624
## stateWest Virginia        -0.6239566  0.3929077  -1.588  0.11338
## stateHawaii               -0.7191564  0.4573908  -1.572  0.11699
## stateNevada                       NA         NA      NA       NA
## studentpopulation         -0.0076465  0.0494428  -0.155  0.87720
## studenttofacultyratio     -0.1368601  0.0338768  -4.040 6.88e-05 ***
## onfinancialaid             0.1224911  0.0330267   3.709  0.00025 ***
## admitted                  -0.0586726  0.0400265  -1.466  0.14379
## submitttingsat             0.0001619  0.0457153   0.004  0.99718
## fulltime                   0.0096574  0.0258500   0.374  0.70898
## fouryeargrad               0.0843400  0.0429220   1.965  0.05039 .
## female                     0.0366204  0.0304624   1.202  0.23030
## white                      0.0783561  0.0332880   2.354  0.01926 *
## averagegrant               0.7347023  0.0362374  20.275  < 2e-16 ***
## recevingloan              -0.0406829  0.0326877  -1.245  0.21430
## averageloan                0.1536776  0.0238871   6.433 5.26e-10 ***
```

```
## applicationfee            -0.0030021  0.0350412  -0.086  0.93179
## amountofapplicants        -0.0711826  0.0403061  -1.766  0.07846 .
## accepted                         NA         NA      NA       NA
## acceptedenrolled          -0.1887390  0.0290372  -6.500 3.58e-10 ***
## sat25                      0.0016575  0.0783778   0.021  0.98314
## sat75                      0.0406767  0.0712253   0.571  0.56838
## calendarQuarter           -0.1802968  0.1006553  -1.791  0.07432 .
## calendarSemester          -0.1336487  0.0714691  -1.870  0.06251 .
## calendarTrimester         -0.0311334  0.2248986  -0.138  0.89000
## settingSuburban           -0.0006463  0.0654363  -0.010  0.99213
## settingUrban               0.0768585  0.0627593   1.225  0.22172
## athlete                   -0.1095497  0.0361326  -3.032  0.00265 **
## sportsrevenue             -0.0162641  0.0312685  -0.520  0.60337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3121 on 285 degrees of freedom
##   (136 observations deleted due to missingness)
## Multiple R-squared:  0.9257, Adjusted R-squared:  0.9069
## F-statistic: 49.31 on 72 and 285 DF,  p-value: < 2.2e-16
```

```
summary(basenostate)
```

```
##
## Call:
## lm(formula = totalcostinstate ~ rank + acceptedenrolled + studentpopulation *
##     amountofapplicants + submitttingsat + studenttofacultyratio *
```

```
##     fouryeargrad + onfinancialaid * recevingloan + fulltime +
##     female + white + averageloan * averagegrant + applicationfee +
##     sat25 * admitted + calendar + setting + athlete + sportsrevenue,
##     data = final_complete, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79931 -0.21383 -0.00728  0.21194  1.07473
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      0.072166   0.084270   0.856 0.392420
## rank                            -0.004876   0.002004  -2.433 0.015498
## acceptedenrolled                -0.162715   0.030105  -5.405 1.25e-07
## studentpopulation               -0.031487   0.047016  -0.670 0.503519
## amountofapplicants              -0.074512   0.046655  -1.597 0.111209
## submitttingsat                  -0.015972   0.026955  -0.593 0.553907
## studenttofacultyratio           -0.170387   0.038342  -4.444 1.21e-05
## fouryeargrad                     0.128624   0.042745   3.009 0.002823
## onfinancialaid                   0.118629   0.035347   3.356 0.000883
## recevingloan                     0.008722   0.031424   0.278 0.781528
## fulltime                        -0.004692   0.025176  -0.186 0.852285
## female                           0.025076   0.029628   0.846 0.397966
## white                            0.020304   0.028672   0.708 0.479341
## averageloan                      0.138659   0.021692   6.392 5.62e-10
## averagegrant                     0.740347   0.039063  18.953  < 2e-16
## applicationfee                  -0.011229   0.031311  -0.359 0.720093
## sat25                            0.077468   0.046011   1.684 0.093195
```

```
## admitted                              -0.021293   0.039442  -0.540 0.589655
## calendarQuarter                       -0.029908   0.092845  -0.322 0.747559
## calendarSemester                      -0.069816   0.065089  -1.073 0.284231
## calendarTrimester                     -0.168668   0.203134  -0.830 0.406956
## settingSuburban                        0.018370   0.060312   0.305 0.760872
## settingUrban                           0.085042   0.056548   1.504 0.133571
## athlete                               -0.143523   0.037891  -3.788 0.000181
## sportsrevenue                         -0.007042   0.028925  -0.243 0.807813
## studentpopulation:amountofapplicants   0.011704   0.020553   0.569 0.569433
## studenttofacultyratio:fouryeargrad    -0.064511   0.020726  -3.113 0.002018
## onfinancialaid:recevingloan           -0.011754   0.025740  -0.457 0.648218
## averageloan:averagegrant               0.044982   0.021683   2.074 0.038816
## sat25:admitted                         0.026753   0.024050   1.112 0.266787
##
## (Intercept)
## rank                                  *
## acceptedenrolled                      ***
## studentpopulation
## amountofapplicants
## submitttingsat
## studenttofacultyratio                 ***
## fouryeargrad                          **
## onfinancialaid                        ***
## recevingloan
## fulltime
## female
## white
## averageloan                           ***
```

```
## averagegrant                           ***
## applicationfee
## sat25                                  .
## admitted
## calendarQuarter
## calendarSemester
## calendarTrimester
## settingSuburban
## settingUrban
## athlete                                ***
## sportsrevenue
## studentpopulation:amountofapplicants
## studenttofacultyratio:fouryeargrad     **
## onfinancialaid:recevingloan
## averageloan:averagegrant               *
## sat25:admitted
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3199 on 328 degrees of freedom
##   (136 observations deleted due to missingness)
## Multiple R-squared:  0.9101, Adjusted R-squared:  0.9022
## F-statistic: 114.6 on 29 and 328 DF,  p-value: < 2.2e-16
```

I tested the basenostate model against several other models including a backwards-step-wise model. The basenostate model had one of the lowest AIC and therefore I accept it to be the best of the linear models that I have created.

The VIF check was done to see how much multi-colinearity there might still be between the variables. There is still quite a bit of unexplained correlation between them, however since the VIF is under ten I decided that I can keep them.

```
correlationvars <- c(3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,24,25)
completecases <- final_complete[complete.cases(final_complete),]

basemodel2 <- lm(totalcostinstate ~ rank + acceptedenrolled +  studentpopulation * amountofapplicants + sub
mitttingsat +
                 studenttofacultyratio * fouryeargrad +  onfinancialaid * recevingloan + fulltime + female
+ white +
                 averageloan * averagegrant + applicationfee + state +
                 sat25 * admitted + calendar+ setting + athlete + sportsrevenue, data=completecases)

#stepwise function commented out for R-markdown file.
#step(basemodel2,direction="both")

bestbackward<- lm(formula = totalcostinstate ~ acceptedenrolled + amountofapplicants +
                   studenttofacultyratio + fouryeargrad + onfinancialaid + averageloan +
                   averagegrant + sat25 + admitted + athlete + rank + studenttofacultyratio:fouryeargrad +

                   averageloan:averagegrant + sat25:admitted, data = final_complete, na.action=na.omit)



AIC(allmodel, basemodel, basenostate, bestbackward)
```

```
## Warning in AIC.default(allmodel, basemodel, basenostate, bestbackward):
## models are not all fitted to the same number of observations
```

```
##             df      AIC
```

```
## allmodel     74 248.5047
## basemodel    78 240.0691
## basenostate  31 230.5397
## bestbackward 16 234.1297
```

```
vif(bestbackward)
```

```
##                  acceptedenrolled                  amountofapplicants
##                          1.811757                            2.611288
##              studenttofacultyratio                       fouryeargrad
##                          5.129386                            3.917939
##                   onfinancialaid                        averageloan
##                          2.354751                            1.369731
##                     averagegrant                              sat25
##                          4.151629                            4.505355
##                         admitted                             athlete
##                          3.536566                            2.632722
##                             rank studenttofacultyratio:fouryeargrad
##                          1.452677                            2.589883
##          averageloan:averagegrant                     sat25:admitted
##                          1.503384                            2.850619
```

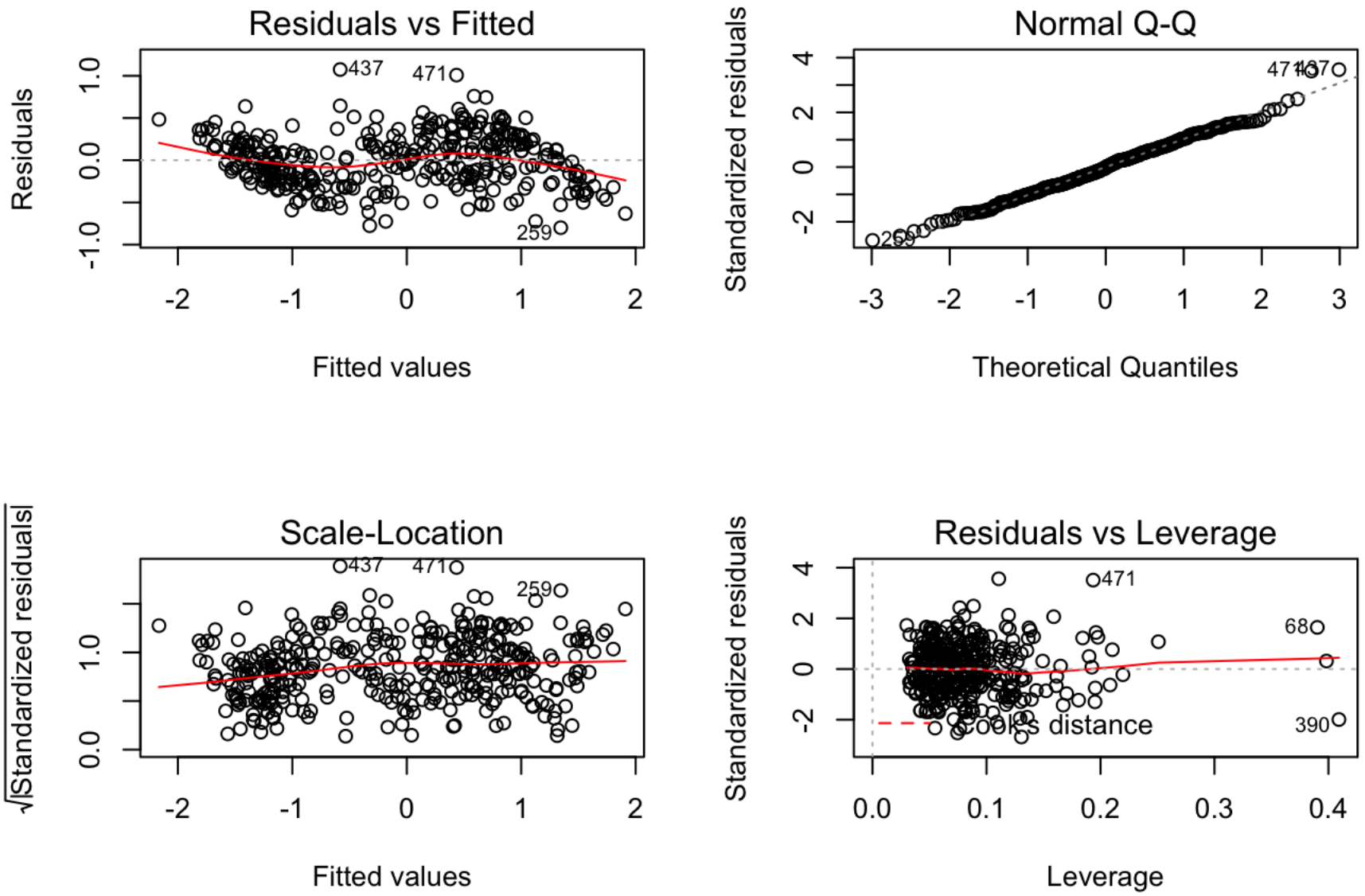
```
vif(basenostate)
```

```
##                                  GVIF Df GVIF^(1/(2*Df))
## rank                         2.243968  1        1.497988
```

```
## acceptedenrolled                       2.632692  1       1.622557
## studentpopulation                      8.952299  1       2.992039
## amountofapplicants                     8.712312  1       2.951663
## submitttingsat                         2.218226  1       1.489371
## studenttofacultyratio                  5.834599  1       2.415491
## fouryeargrad                           7.063791  1       2.657779
## onfinancialaid                         4.168972  1       2.041806
## recevingloan                           3.492328  1       1.868777
## fulltime                               2.120014  1       1.456027
## female                                 1.536320  1       1.239484
## white                                  3.014621  1       1.736266
## averageloan                            1.548974  1       1.244578
## averagegrant                           5.607234  1       2.367960
## applicationfee                         3.591846  1       1.895217
## sat25                                  7.573959  1       2.752083
## admitted                               5.488443  1       2.342743
## calendar                               1.610333  3       1.082645
## setting                                1.665537  2       1.136027
## athlete                                4.381156  1       2.093121
## sportsrevenue                          3.378866  1       1.838169
## studentpopulation:amountofapplicants 4.297501  1       2.073041
## studenttofacultyratio:fouryeargrad   3.104316  1       1.761907
## onfinancialaid:recevingloan          2.279416  1       1.509774
## averageloan:averagegrant             1.614464  1       1.270616
## sat25:admitted                         4.079609  1       2.019804
```

```
#qqPlot(basenostate, main="QQ Plot")
#resid <- resid(basenostate)
```

```
par(mfrow=c(2,2))
plot(basenostate)
```



PCA Analysis

I went on to do some principle component analysis. I created two models a principle component regression and a principle least squared regression. From the summary of the two different models I concluded that the plsr model was the better model since it accounted for more variance of tuition using fewer components.

```
temp1 <- final_complete[,correlationvars]
temp2 <- complete.cases(temp1)
complete_continous <- temp1[temp2,]

pcrModel = pcr(totalcostinstate ~ .,
               data=complete_continous, rotation='Varimax')

plsrModel = plsr(totalcostinstate ~ .,
                 data=complete_continous, rotation='Varimax')

summary(pcrModel)
```

```
## Data:    X dimension: 358 20
##  Y dimension: 358 1
## Fit method: svdpc
## Number of components considered: 20
## TRAINING: % variance explained
##                  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X                  33.55    57.46    67.21     74.4    78.31    81.78
## totalcostinstate   18.12    69.05    70.52     75.2    80.05    80.51
##                  7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
```

```
## X                  84.73    87.18    89.34     91.33     93.03     94.48
## totalcostinstate   81.03    81.22    83.10     83.64     83.72     84.06
##                  13 comps  14 comps  15 comps  16 comps  17 comps
## X                   95.74     96.90     97.87     98.72     99.33
## totalcostinstate    84.33     86.16     86.20     89.28     89.69
##                  18 comps  19 comps  20 comps
## X                   99.84    100.00    100.00
## totalcostinstate    90.28     90.28     90.28
```

```
summary(plsrModel)
```

```
## Data:    X dimension: 358 20
##  Y dimension: 358 1
## Fit method: kernelpls
## Number of components considered: 20
## TRAINING: % variance explained
##                  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X                  27.68    54.00    63.51    69.09    75.04    79.27
## totalcostinstate   72.79    78.75    85.19    88.07    89.29    90.00
##                  7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
## X                  81.37    83.40    85.43     87.51     89.47     91.18
## totalcostinstate   90.22    90.26    90.28     90.28     90.28     90.28
##                  13 comps  14 comps  15 comps  16 comps  17 comps
## X                   92.36     93.65     95.10     96.15     97.68
## totalcostinstate    90.28     90.28     90.28     90.28     90.28
##                  18 comps  19 comps  20 comps
## X                   98.92    100.00    101.03
```

```
## totalcostinstate     90.28     90.28     90.28
```

Looking at the components in the plsr model does not provide as clear a division in the variable groups as I had hoped it would. The first three components account for 85% of the variance however there are no heavily weighted variables in those components.

```
loadings <- as.data.frame(plsrModel[1])
names(loadings) <- c("PC1", "PC2", "PC3", "PC4", "PC5",
                     "PC6", "PC7", "PC8", "PC9", "PC10",
                     "PC11", "PC12", "PC13", "PC14", "PC15",
                     "PC16", "PC17", "PC18", "PC19", "PC20")
loadings[,1:4]
```

```
##                              PC1         PC2          PC3          PC4
## studentpopulation     -0.109227177 -0.13247315 -0.084003997 -0.054424022
## studenttofacultyratio -0.157633325 -0.17903796 -0.186795459 -0.178490106
## onfinancialaid        -0.002175640  0.04663756  0.090658152  0.127694358
## admitted              -0.079316002 -0.04942137 -0.034004490  0.003108021
## submitttingsat         0.024462929  0.02769213  0.047849908 -0.035662190
## fulltime               0.076262686  0.06154948  0.011118641 -0.035621427
## fouryeargrad           0.149970214  0.15838493  0.172068230  0.173504533
## female                 0.016310117  0.04088839  0.084067893  0.086471688
## white                 -0.008592882 -0.02288397 -0.114104138 -0.097627393
## averagegrant           0.179390846  0.23391336  0.382768266  0.549394466
## recevingloan           0.012559655  0.05884842  0.081709792  0.052696562
## averageloan            0.031938951  0.07908287  0.154324981  0.149663084
## applicationfee         0.037787564  0.02607660  0.074119072  0.046593060
## amountofapplicants    -0.036416327 -0.05308647  0.008144206 -0.014705521
```

```
## accepted              -0.079316002 -0.04942137 -0.034004490  0.003108021
## acceptedenrolled      -0.030474866 -0.08139991 -0.155334237 -0.137900511
## sat25                  0.096826009  0.07079162  0.060663811  0.066881732
## sat75                  0.089737648  0.06181666  0.045938612  0.059897892
## athlete                0.090919190  0.09505547  0.026144603 -0.022638994
## sportsrevenue         -0.068620845 -0.08626613 -0.039375629  0.022949119
```

Future Work

Using this plsr model I intend to check which schools are over-priced and which ones are good deals in relation to the variables that I have found to be significant.