The College Scorecard

Varun Nadgir

September 27, 2017

Capstone Project

Introduction

The **College Scorecard** is a service meant to help prospective students make their college decision. Whether by comparing size, popular majors, or comparing costs to the national average, the site's goal is to help the user find a good fit. For my Springboard Capstone Project, I used the **dataset** made available by the College Scorecard to try and find additional ways to help users in their decision.

The Data

Available in a .zip file from the link above, the data is split into 19 .csv files - one for each academic year from 1996-'97 to 2014-'15. Each file contains 1,744 recorded data points (columns) and about 7,500 schools (rows). My first step was to add a DATAYEAR column to indicate the academic year and then merging the 19 files into one large .csv, which I called **fulldata.csv**. From this 2 GB file, I would create subsets for plotting and studying trends. Next, I had to refer to **the data dictionary** to understand the column names and some of the placeholder values used. Once I merged the files and had a basic understanding of what data was available, I began to create some plots and documented my initial findings in my **data story**. As a note, my studies have been on US schools only. The dataset includes records of US territories as well, but I have filtered them out when creating subsets for plotting/modeling.

Deliverables

My first item will be using linear regression models to determine what variables are the most influential on the cost of a school, and to potentially predict what the cost of a school may be in the future. This would be useful in two ways. It could help students who are on the fence about going to college immediately after high school by suggesting a decline in cost. If a student sees that their ideal school is likely to be cheaper in two years, they may make the decision to find entry-level work or go traveling before going to college. It could also help the schools by indicating areas of their budget that are influencing the cost of attendance. Of course, the goals of each school are different and they may not be interested in reducing cost, but if a school is experiencing a decline in applications, cost of attendance is likely to be something they look at.

The second item is a recommendation tool that works similarly to clustering methods used by media services such as Netflix, YouTube, and iTunes. Clustering based on things like location, cost, and SAT scores, a student can find options that are close to their top choice. Mentality is a very important part of finding success at school, and feeling out of place in freshman year can be quite discouraging. If their top choice is a far reach school, or it is too expensive/too far, then finding alternatives would hopefully help them to be satisfied in their decision.

My final item will be a basic UI that allows the user to explore the dataset on their own. Although the data will need to be curated and shaved down to a size that a standard internet browser can handle, my hope is that it will provide some transparency between students and universities. As an example, in my **data story**, I explored SAT averages and admission rates. Even though one could reasonably guess how they are related (higher SAT scores ~ lower admission rates), being able to plot the data and draw a conclusion from a graph is much more convincing. By putting this power in the hands of students and their families, they should be able to make much more educated decisions.

Data Preparation

From **fulldata.csv**, I created and saved a subset of the columns that would make repeated data manipulation more manageable. I named it **subdata** and it contains various *school identification data* (name, ID, location, level of institution), *admission data* (student demographics, SAT scores), *education data* (majors, completion rates), and *financial data* (cost, aid, debt, repayment).

From this subset, I can pick the variables for the cost modeling, as well as the variables for clustering - both tools will get specific subsets made so that the data is organized and loading it will be easier in the future. A combination of the original source data and my curated data will go into the UI plotting tool, keeping in mind that the dataset should stay a reasonable size but also have enough depth for the user to gain whatever insights they can.

Cost Prediction

For convenience, I began by changing the DATAYEAR column from indicating the year range ("1996-'97") to simply having the start year ("1996"). Using the datayear, I created the training file using years 1996 to 2013 and the test file with just the 2014 data. Since the cost is known for 2014, we can compare afterwards to see how good the model is beyond the R^2 score alone.

From **subdata**, I created a subset of up to 14 columns that I felt would be related to the attendance cost of a school. This included columns like "CONTROL" (whether the school is public or private), "UGDS" (the undergraduate student population), "PFTFAC" (percentage of faculty that are full time), and of course "COSTT4_A" (cost of attendance for a 4-year academic year school). Upon checking the summary() of the training set, I noticed that the MAX of "UGDS" was unnaturally high. This was because there were a few records of the University of Pheonix's online campus, which is capable of having about 200,000 students. Since this was skewing the data, I removed its related "OPEID6" from the training and test sets. I also removed all records in the training set that did not have a "COSTT4_A" entry, since they would not contribute to the model. This filtering had an interesting byproduct - apparently years 2008 and prior did not have "COSTT4_A" records, so removing the NA entries also happened to filter the dataset to years 2009 and beyond. After taking care to turn appropriate columns into the numeric data type, I moved on to make small experimental models.

The first model I tried was on Cost by Undergrad Pop., Average Faculty Salary, and % of Faculty that are Full-Time. This, however, only gave an R^2 of 0.2198, which is not that great. However, the model summary suggests that these three variables are still significant (using asterisk notation), so it is a step in the right direction. The 2nd model I tried used 12 independent variables to model Cost. It gave an R^2 of 0.8253, but still showed that three of the variables had little to no significance. After removing these, this was the final model I came to:

Cost by Datayear, Highest degree offered, Pub/Priv, Avg SAT, Undergrad Pop., Avg Faculty Salary, % of Faculty that are Full-Time, Median Debt, and Avg Family Income

```
##
## Call:
## lm(formula = COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + SAT_AVG +
##     UGDS + AVGFACSAL + PFTFAC + DEBT_MDN + FAMINC, data = df.train,
##     na.action = na.omit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -42436  -2843    163   3267  37049
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -1.510e+06  1.013e+05 -14.914  < 2e-16 ***
## DATAYEAR     7.357e+02  5.040e+01  14.597  < 2e-16 ***
## HIGHDEG      7.880e+02  1.277e+02   6.171 7.19e-10 ***
## CONTROL      1.522e+04  1.606e+02  94.793  < 2e-16 ***
## SAT_AVG      1.140e+01  8.111e-01  14.059  < 2e-16 ***
## UGDS        -1.843e-01  1.155e-02 -15.954  < 2e-16 ***
## AVGFACSAL    1.709e+00  4.681e-02  36.509  < 2e-16 ***
## PFTFAC      -1.352e+03  2.673e+02  -5.056 4.39e-07 ***
## DEBT_MDN     2.558e-01  1.992e-02  12.838  < 2e-16 ***
## FAMINC       1.043e-01  4.618e-03  22.588  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5086 on 6962 degrees of freedom
##   (12666 observations deleted due to missingness)
## Multiple R-squared:  0.825,  Adjusted R-squared:  0.8247
## F-statistic:  3646 on 9 and 6962 DF,  p-value: < 2.2e-16
```

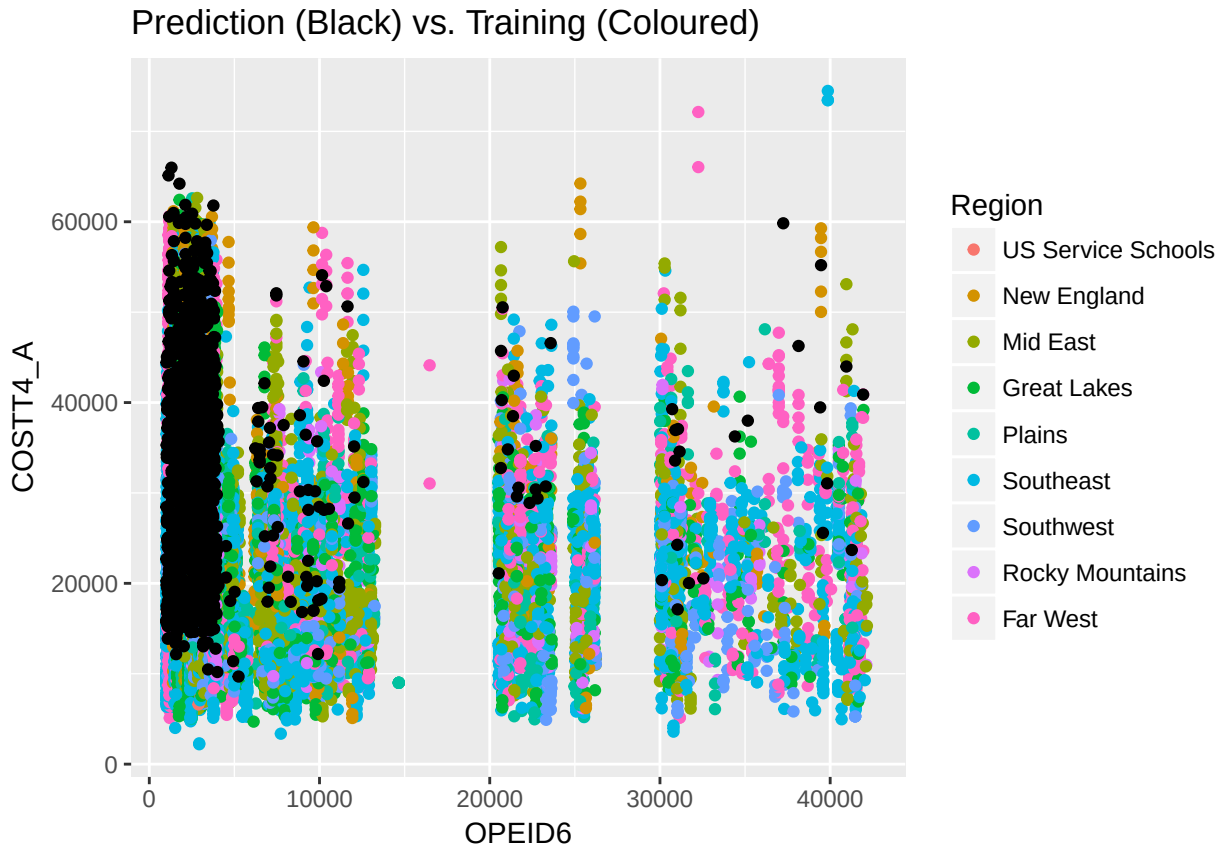
With an R^2 of 0.825, this model is almost equally as good as the second model, but now all of the variables used are significant. In other words, the variables that likely have the most influence on the cost of attendance of a school are the ones in this model. This is saved as **cost.mod** and will be used to predict on the test set.

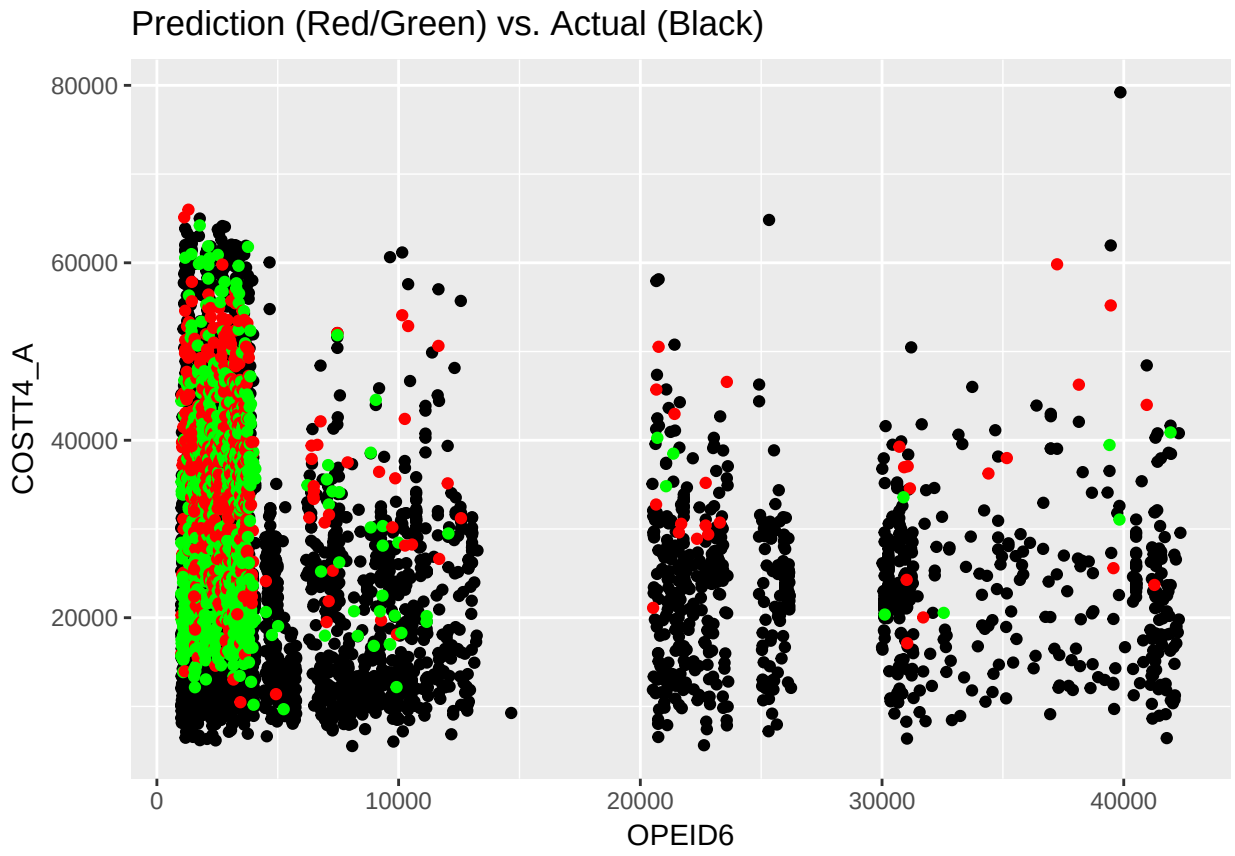
Summary of test data after predictions

```
##      OPEID6          INSTNM             DATAYEAR        REGION
##  Min.   : 1002   Length:3888        Min.   :2014   Min.   :0.000
##  1st Qu.: 2470   Class :character   1st Qu.:2014   1st Qu.:3.000
##  Median : 3810   Mode  :character   Median :2014   Median :5.000
##  Mean   : 9543                      Mean   :2014   Mean   :4.408
##  3rd Qu.:10727                      3rd Qu.:2014   3rd Qu.:6.000
##  Max.   :42345                      Max.   :2014   Max.   :8.000
##
##     HIGHDEG        CONTROL         SAT_AVG          UGDS
##  Min.   :0.00   Min.   :1.000   Min.   : 720   Min.   :    0
##  1st Qu.:2.00   1st Qu.:1.000   1st Qu.: 973   1st Qu.:  446
##  Median :3.00   Median :2.000   Median :1039   Median : 1486
##  Mean   :2.97   Mean   :1.833   Mean   :1059   Mean   : 3893
##  3rd Qu.:4.00   3rd Qu.:2.000   3rd Qu.:1120   3rd Qu.: 4294
##  Max.   :4.00   Max.   :3.000   Max.   :1545   Max.   :77657
##                                 NA's   :2595   NA's   :1
##    AVGFACSAL         PFTFAC          DEBT_MDN         FAMINC
##  Min.   :  332   Min.   :0.0000   Min.   : 1354   Min.   :     0
##  1st Qu.: 4914   1st Qu.:0.2958   1st Qu.: 8414   1st Qu.: 24289
##  Median : 6102   Median :0.5296   Median :12500   Median : 35595
##  Mean   : 6351   Mean   :0.5630   Mean   :12954   Mean   : 43701
##  3rd Qu.: 7553   3rd Qu.:0.8564   3rd Qu.:17500   3rd Qu.: 58456
##  Max.   :20650   Max.   :1.0000   Max.   :37500   Max.   :152100
##  NA's   :190     NA's   :520      NA's   :232     NA's   :15
##     COSTT4_A        PREDICT           DIFF
##  Min.   : 5536   Min.   : 9691   Min.   :-15605.9
##  1st Qu.:14634   1st Qu.:24010   1st Qu.: -2884.8
##  Median :23200   Median :34152   Median :   568.9
##  Mean   :25242   Mean   :33199   Mean   :   560.2
##  3rd Qu.:31789   3rd Qu.:40676   3rd Qu.:  3819.4
```

```
##  Max.   :79212   Max.   :65985   Max.   : 26866.1
##                  NA's   :2608    NA's   :2608
```

The predict() formula returns a column that I named "PREDICT" and, by taking the difference of "PREDICT" and "COSTT4_A", we can see whether the prediction was too high or too low. This summary tells us that the extreme cases were -$15,605 and $26,866 off the mark. What is more disappointing, however, is the number of NAs in the prediction output. To see where the model is falling short, I can check two plots. The first plot overlays the prediction values (in black) over the training data. This helps to see if the predicted values stayed in the same range and may indicate another reason why some predictions weren't made. The second plot overlays the prediction values over the actual 2014 values, and they are coloured green if within $4000 of the actual cost and red if predicted too far. If the plot is mostly green, that would be a good sign.





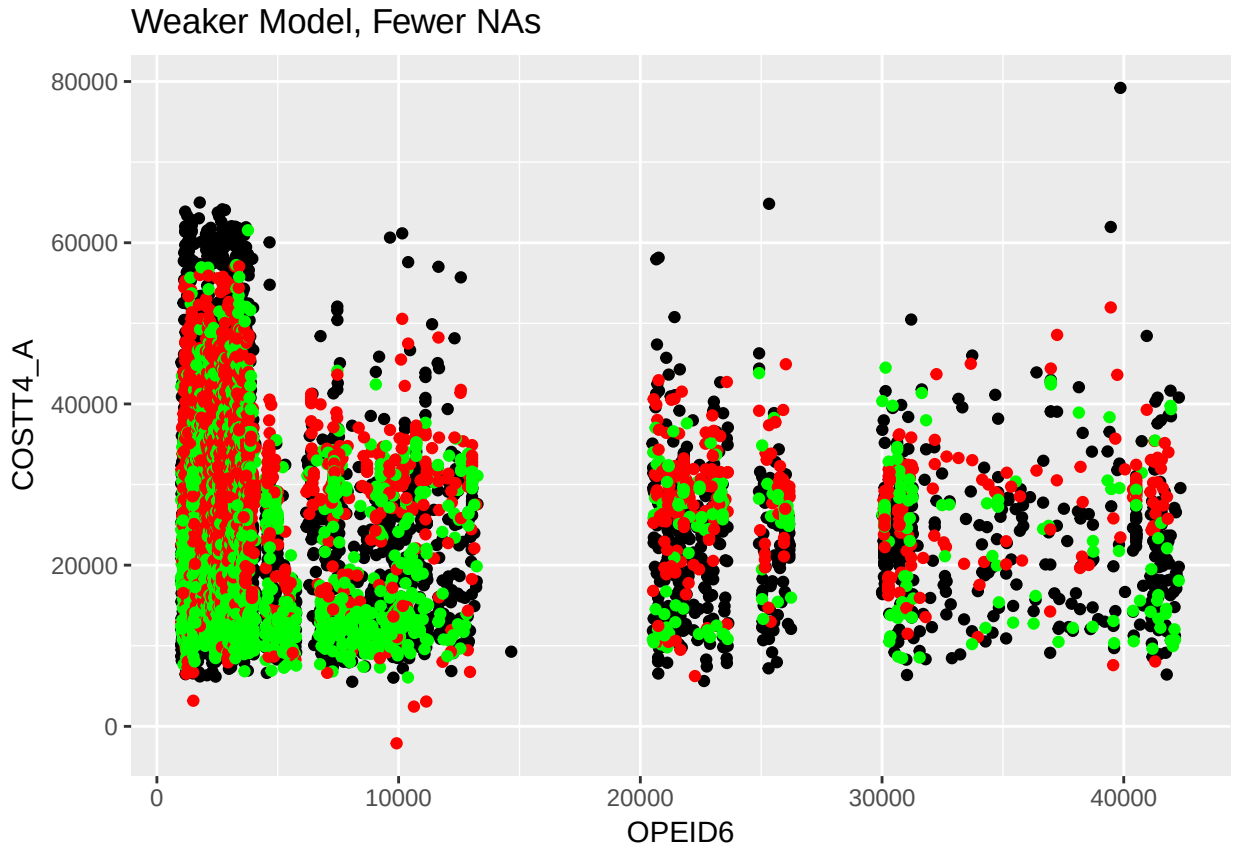
According to these plots, the predictions were mostly succesful for schools in the 0 ~ 3000 range for "OPEID6", while predictions beyond that are very sporadic. Looking at the dataset and sorting by "OPEID6" shows that there are many NAs in the independent variables, causing the prediction to fail for those cases. This is unfortunate, since this causes about 2,600 failed predictions despite a fairly strong model. This leaves us with a few different options. We could remove the independent variable(s) with the most NAs from the model, which risks weakening the model but will yield more successful predictions. Another option could be to replace NAs with another value, such as the mean for that column. This doesn't hurt the model at all, but the predictions for the schools with approximated values have a chance to be completely off.

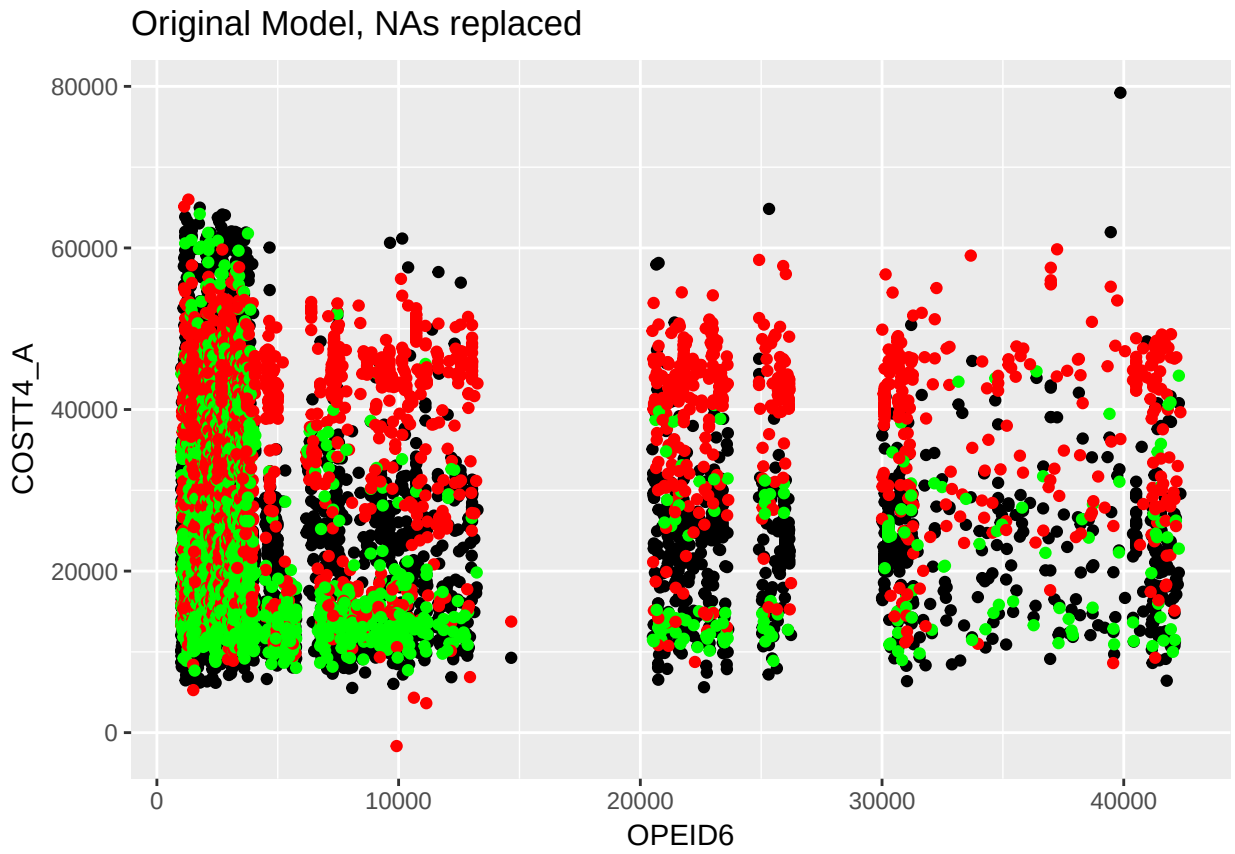
The independent variable with significantly more NAs than the others is "SAT_AVG". We can create a new model that does not include "SAT_AVG" and, it turns out, removing it reveals "PFTFAC" to be an insignificant variable as well. With both removed, the new model has an R^2 of 0.771, which is not bad at all. Checking the summary of this new model shows that there are only about 400 NAs in the prediction this time, which is a massive improvement.

```
##
## Call:
## lm(formula = COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + UGDS +
##     AVGFACSAL + DEBT_MDN + FAMINC, data = df.train, na.action = na.omit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -47320  -3358    -87   3148  49616
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.144e+06  6.731e+04  -17.00   <2e-16 ***
```

```
## DATAYEAR     5.619e+02  3.348e+01   16.78   <2e-16 ***
## HIGHDEG      1.018e+03  5.861e+01   17.36   <2e-16 ***
## CONTROL      8.354e+03  7.065e+01  118.25   <2e-16 ***
## UGDS        -1.755e-01  8.043e-03  -21.82   <2e-16 ***
## AVGFACSAL    1.217e+00  3.119e-02   39.03   <2e-16 ***
## DEBT_MDN     2.816e-01  1.408e-02   19.99   <2e-16 ***
## FAMINC       2.134e-01  2.913e-03   73.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5660 on 17310 degrees of freedom
##   (2320 observations deleted due to missingness)
## Multiple R-squared:  0.771,  Adjusted R-squared:  0.7709
## F-statistic:  8327 on 7 and 17310 DF,  p-value: < 2.2e-16
```

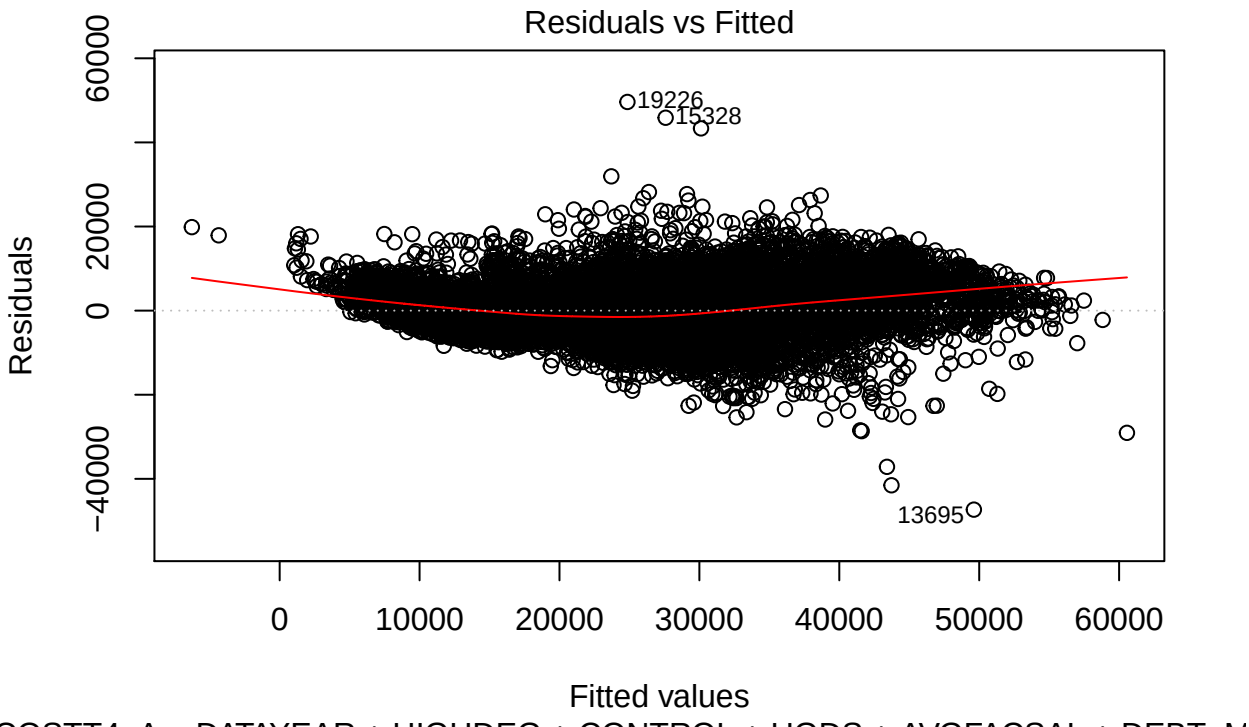
We can compare the performance of this model to the other option, where NAs would have been replaced by the column mean. Is it worth it to have a slightly weaker model for more accurate predictions, or will replacing the NAs solve the problem? Ideally, the colour coding of the plot will indicate the better model.



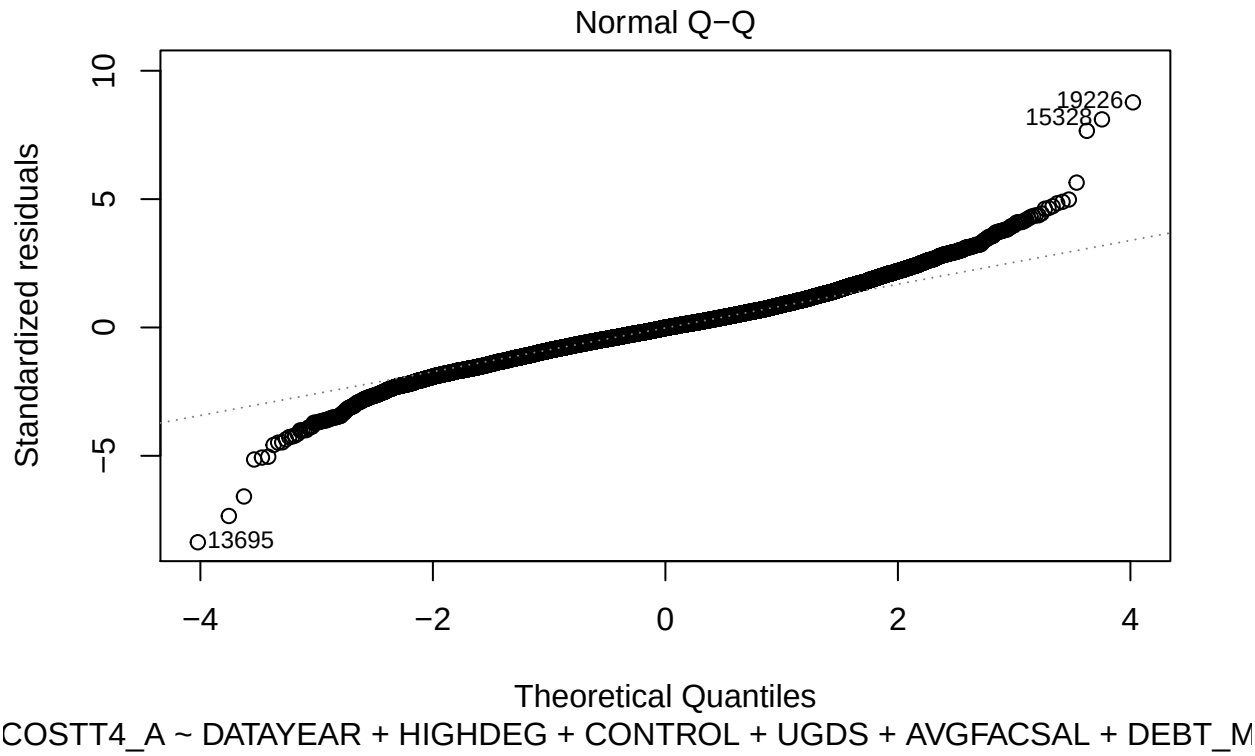


It truly looks as though the weaker model with a few NAs remaining is the better one for prediction. The amount of red in the NA-replacement model shows that many predictions were quite off and that forcing the model to make predictions when there was not enough data was the wrong decision. Giving up the two variables and using a slightly weaker model, while keeping the original data, seems to stay within the range of the actual 2014 data for the most part.

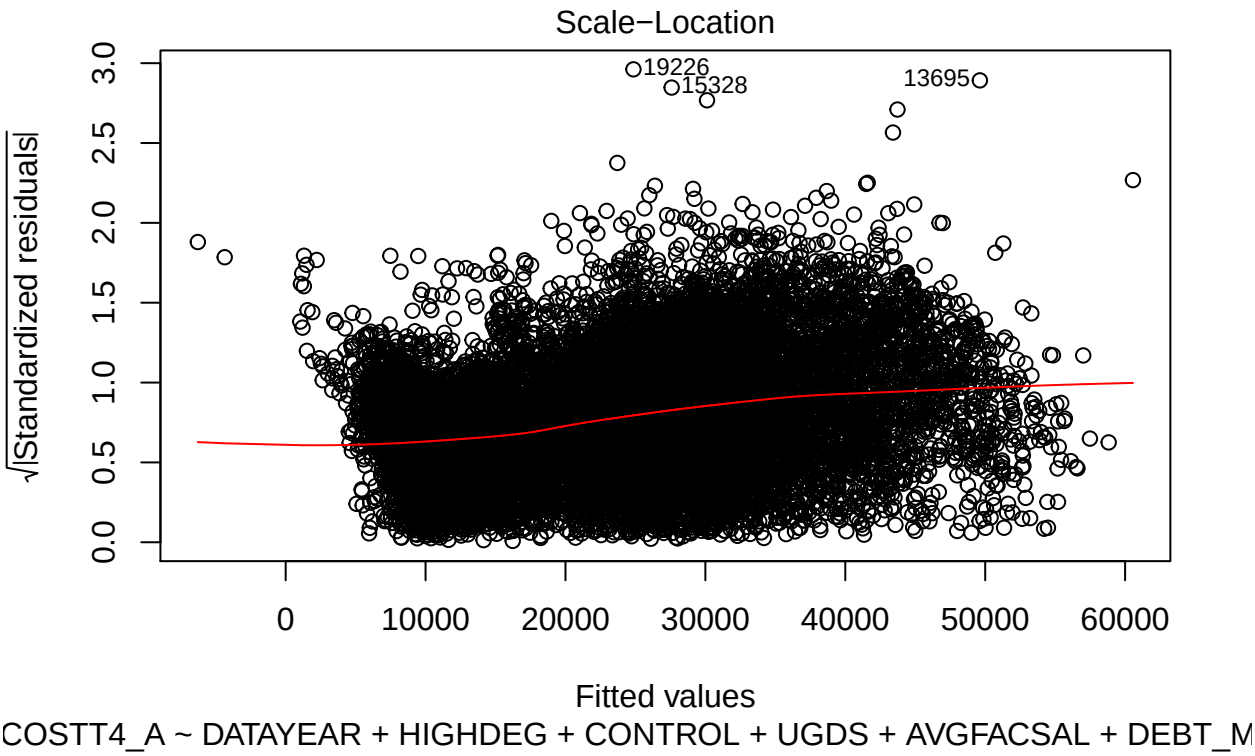
If we plot this new model, we can also use the residuals to judge how well it works.

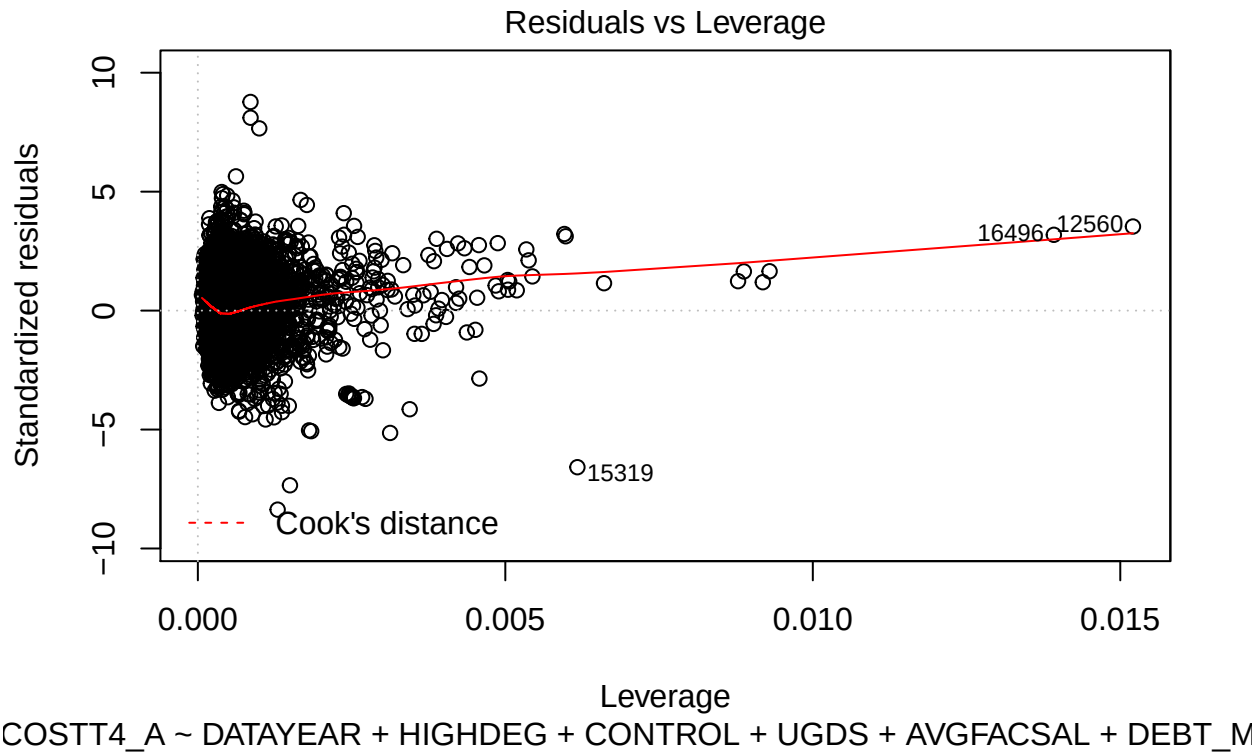


COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + UGDS + AVGFACSAL + DEBT_M



COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + UGDS + AVGFACSAL + DEBT_M


Scale−Location
19226
15328
13695
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
10000
20000
30000
40000
50000
60000
Fitted values
COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + UGDS + AVGFACSAL + DEBT_M



The Residuals vs Fitted plot stands out right away, showing that the residuals generally reside around the fitted line. The Normal Q-Q plot also follows the normal line for the most part, aside from the very bottom and very top of the plot. It seems like this model is fairly strong, and the R^2 and plot of residuals reinforces it.

After revisions, the current conclusion is that Cost is strongly related to the school's highest provided degree, whether it is public or private, the student population size, the average faculty salary, the median student debt, and the average family income. It is unlikely that the school will change between public/private, or that it will stop providing higher degrees, but there is a chance that the school might look into its admission rates. According to the coefficients table, "UGDS" is negatively correlated to "COSTT4_A", which means that school prices go up as student population sizes go down; i.e., fewer people go to the more expensive schools, and the schools that accept the most students can afford to have slightly lower attendance costs since they make it back through sheer numbers.

It should also be mentioned that the $4,000 "acceptable" window I applied to the prediction values was chosen arbitrarily. This window could change depending on the student's budget - someone who is more well-off could widen the window and someone with a tighter budget could shrink it. Despite removing some of the variables that made what seemed to be a stronger model, the final model we reached seems rather good at predicting cost when all data points are present.

School Recommendation

From **subdata**, I created another subset for clustering. Since the plan is to have a way to recommend schools to a student, I filtered down to just the school data from 2014 (meaning the 2014-'15 school year). Since this is the most recent year of data available in the dataset, it felt the most appropriate to use when trying to classify them. With some trial and error, I eventually decided on 9 variables to use, aside from the school

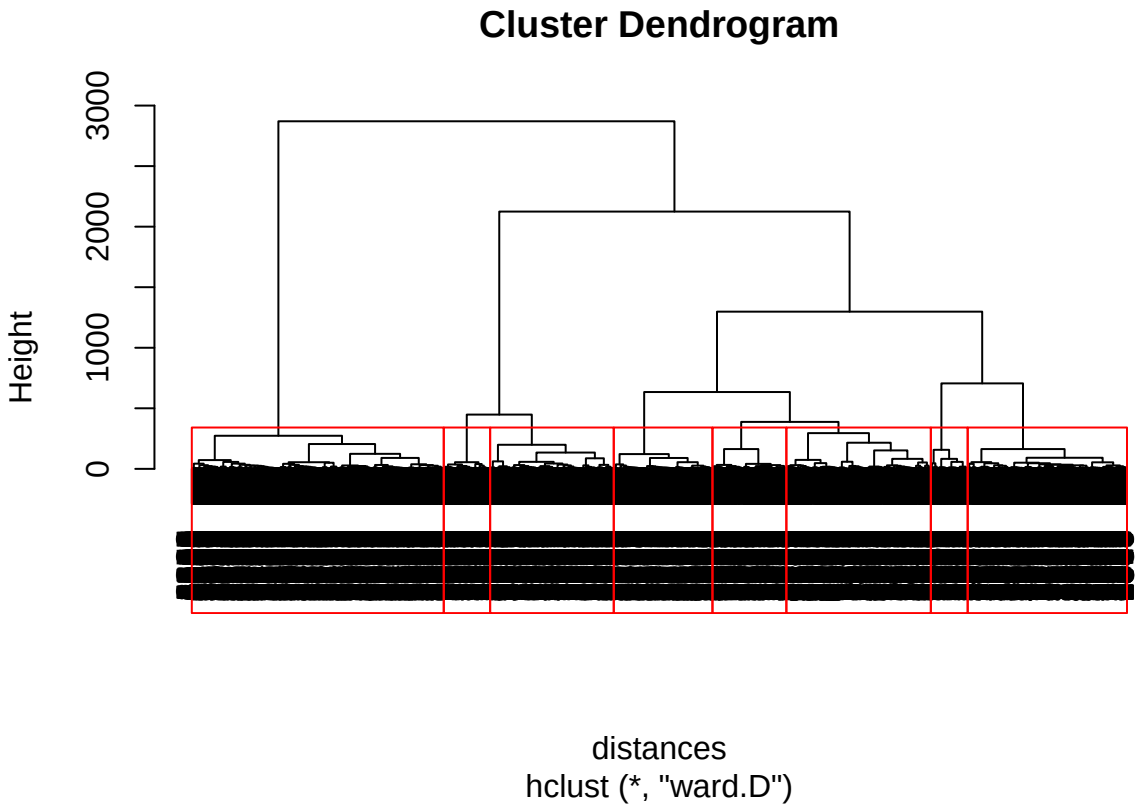
name column. At this point, I made two subsets - **clustdata** for performing data manipulation, and a copy called **schoolCluster2014** which would remain as is until the end when we have clusters to assign.

Proper data cleaning was important here, since calculating Euclidean distance is the first step to building the clustering model. I needed to scale() the numeric variables, so that variables with different ranges do not skew the model. However, before scaling, there were a few changes I had to make to the data.

For population size, family income, and age of entry, I replaced NAs with the mean of the column. The two cost columns were different - one column shows the cost of academic year schools and the other is for program year schools, meaning if one column had an entry, the other would be NA. In this case, it wouldn't make sense to replace NAs with the mean, so I replaced them with 0 instead. With the NAs filled, we can run scale(clustdata[2:6]), where the 2nd to 6th columns are the numeric variables. After handling the NA values, I also had to check the mean() and sd() so I could note them down. I kept these in a spreadsheet for later, since I will need them to change the scaled values back to ones we can make sense of for analysis.

I also transformed the categorical variables into their own binary columns - instead of checking the row for the value in the cell, the value is in the column name and the cell is a 1 or 0. This meant adding 51 columns for the states and DC, 3 columns for public or private (for-profit and non-profit), and 12 columns for different locale types (city, town, etc). After adding the new binary columns, the original columns were removed.

After these changes, we can run dist(clustdata[2:72]) to get the Euclidean distances for the schools. With the distances, we can then create a cluster object, which can be used to plot a dendrogram. This will give us an idea of how many clusters we should create from the dataset.



Using a horizontal line, we can cut the dendrogram to help us decide how many clusters to use. If we cut higher up, we would not have enough clusters to give meaningful recommendations. The red boxes here show a spot that crosses the dendrogram 8 times.

The next step involved a little bit of manual work outside of R. I used tapply() to find the mean occurrences

for the variables in each cluster, copy/pasted the values into a text editor, and made a .csv manually. This file will show us what values are most common in the cluster, allowing us to come up with classifications. For the numeric variables, the value here will tell us the mean (of the scaled value) in that cluster - for example, telling us the mean cost is high or the mean population is low. And the binary columns will just be $\frac{N}{7542}$, where N is the number of "1" values and 7542 is the number of schools - this gives us an easy percent.

Cluster Means

```
## # A tibble: 20 x 9
##               name         `1`          `2`         `3`         `4`
##              <chr>       <dbl>        <dbl>       <dbl>       <dbl>
##  1            UGDS  0.45167960 -0.206787400  3.79701700  0.18535070
##  2        COSTT4_A -0.01254381 -0.169963390  0.37271110  2.43075640
##  3        COSTT4_P -0.65546113 -0.660025890 -0.66002589 -0.66002589
##  4          FAMINC -0.13780205 -0.029948810  0.71906651  2.69492410
##  5       AGE_ENTRY -0.35099900  0.312633600 -0.87248440 -1.58489500
##  6          public  0.97431907  0.001677852  0.96632997  0.00000000
##  7 private_nonprof  0.01167315  0.996644295  0.01683502  1.00000000
##  8 private_forprof  0.01400778  0.001677852  0.01683502  0.00000000
##  9      city_large  0.09338521  0.296979870  0.30976431  0.21657754
## 10        city_mid  0.09805447  0.144295300  0.11784512  0.17379679
## 11      city_small  0.12373541  0.132550340  0.21548822  0.14171123
## 12    suburb_large  0.12451360  0.241610700  0.20202020  0.28877010
## 13      suburb_mid  0.04280156  0.006711409  0.02020202  0.02941176
## 14    suburb_small  0.02101167  0.008389262  0.02020202  0.00802139
## 15     town_fringe  0.02568093  0.008389262  0.02020202  0.03475936
## 16    town_distant  0.10350195  0.020134230  0.04040404  0.04545455
## 17     town_remote  0.11439689  0.016778520  0.05387205  0.03208556
## 18    rural_fringe  0.15642023  0.048657720  0.00000000  0.01336898
## 19   rural_distant  0.04046693  0.018456376  0.00000000  0.01604278
## 20    rural_remote  0.02334630  0.016778523  0.00000000  0.00000000
## # ... with 4 more variables: `5` <dbl>, `6` <dbl>, `7` <dbl>, `8` <dbl>
```

Since the first five rows are still scaled, it is hard to make classifications without context. That is why I kept the mean() and sd() of the unscaled values beforehand. To convert back, we just use

$$x = (sd \times z) + m$$

Now we can understand the numeric values since they are in their original units. Using Google Sheets, I did some colour coding to see which values here stood out. Some clusters are a bit more distinct than others, but 8 seems to have been a good choice. Yellow cells are close to the original mean(), green cells are above, and red cells are below. Although the colouring is a bit arbitrary from my end, the values are fortunately different enough to make unique clusters. I also checked the count of each of the clusters to see how many schools were placed into them, which could also help determine if the classifications we decide on make sense.

	A	B	C	D	E	F	G	H	I
1	Name	1	2	3	4	5	6	7	8
2	UGDS	4,524.42	1,318.82	20,810.42	3,227.86	1,564.50	577.13	214.39	941.14
3	COSTT4_A	12,924.10	10,471.14	18,927.25	50,996.31	29,571.40	26,132.23	0.00	962.66
4	COSTT4_P	43.94	0.00	0.00	0.00	9.93	0.00	19,671.15	6,766.69
5	FAMINC	32,916.66	35,228.58	51,284.27	93,638.24	56,997.03	23,962.00	25,441.44	25,229.71
6	AGE_ENTRY	25.09	27.62	23.11	20.40	22.69	29.61	26.84	30.38
7	public	0.974319066	0.001677852	0.966329966	0	0.120361083	0	0.000984252	0.303862661
8	private_nonprof	0.011673152	0.996644295	0.016835017	1	0.739217653	0	0.000492126	0.149356223
9	private_forprof	0.014007782	0.001677852	0.016835017	0	0.140421264	1	0.998523622	0.546781116
10	city_large	0.09338521	0.29697987	0.30976431	0.21657754	0.24272818	0.31030151	0.23228346	0.16909871
11	city_mid	0.09805447	0.1442953	0.11784512	0.17379679	0.06519559	0.16080402	0.13188976	0.06695279
12	city_small	0.12373541	0.13255034	0.21548822	0.14171123	0.12738215	0.11557789	0.14714567	0.06609442
13	suburb_large	0.1245136	0.2416107	0.2020202	0.2887701	0.1935807	0.3542714	0.3400591	0.2085837
14	suburb_mid	0.042801556	0.006711409	0.02020202	0.029411765	0.027081244	0.021356784	0.036417323	0.013733906
15	suburb_small	0.021011673	0.008389262	0.02020202	0.00802139	0.026078235	0	0.021161417	0.01888412
16	town_fringe	0.025680934	0.008389262	0.02020202	0.034759358	0.032096289	0.002512563	0.009350394	0.010300429
17	town_distant	0.10350195	0.02013423	0.04040404	0.04545455	0.14042126	0.01005025	0.03494094	0.05236052
18	town_remote	0.11439689	0.01677852	0.05387205	0.03208556	0.0441324	0.01005025	0.03248031	0.02660944
19	rural_fringe	0.15642023	0.04865772	0	0.01336898	0.03911735	0.01256281	0.01279528	0.06008584
20	rural_distant	0.040466926	0.018456376	0	0.016042781	0.025075226	0.002512563	0.000492126	0.01888412
21	rural_remote	0.023346304	0.016778523	0	0	0.013039117	0	0.000492126	0.006866953

```
## clusterGroups
##    1    2    3    4    5    6    7    8
## 1285  596  297  374  997  796 2032 1165
```

Based on all of these factors, I came up with the following classifications:

1. Public, Medium Pop, Mid Cost (A), Mid Family Income, Mid 20s, Varying Locales
2. Private Non-Prof, Low Pop, Mid Cost (A), Mid Family Income, Mid 20s, Large City/Suburb
3. Public, High Pop, Mid Cost (A), High Family Income, Low 20s, Large City/Suburb
4. Private Non-Prof, Medium Pop, High Cost (A), High Family Income, Low 20s, Large City/Suburb
5. Private Non-Prof, Low Pop, High Cost (A), High Family Income, Low 20s, Large City/Suburb
6. Private For-Prof, Low Pop, High Cost (A), Low Family Income, Upper 20s, Large City/Suburb
7. Private For-Prof, Low Pop, High Cost (P), Low Family Income, Mid 20s, Large City/Suburb
8. Public/Private For-Prof, Low Pop, Mid Cost (P), Low Family Income, Upper 20s, Large City/Suburb

There are some interesting take-aways from this already. It is pretty unsurprising that the majority of public schools are in cluster 1, where student population, cost, and family income are all around average, and cluster 7, where students are joining for-profit schools to try and gain new skills in their mid-20s since their incomes are on the lower side. Cluster 1 is also not dominated by any one locale, though the most are small towns or cities. Clusters 3, 4, and 5 also seem to indicate that the students going to schools in large locales are younger, have higher family incomes than the others, but still have different preferences on population and cost of their school.

Finally, we can append the clusters to the end of the original data. Earlier in the process I made a table called **schoolCluster2014**, to which I can now assign clusters. Now if a student has a school in mind, for example "Drexel University", they can do a lookup for that school and see what cluster it is in. In this case, Drexel is in cluster 4.

Cluster 4

```
## # A tibble: 374 x 10
##                                INSTNM STABBR CONTROL LOCALE  UGDS COSTT4_A
##                                 <chr>  <chr>   <int>  <int> <chr>    <chr>
```

```
##  1        Birmingham Southern College     AL       2     12  1180    45162
##  2                  Samford University     AL       2     21  3033    40900
##  3                Spring Hill College      AL       2     12  1215    44970
##  4                    Hendrix College      AR       2     13  1322    52536
##  5       Art Center College of Design      CA       2     12  1814    57738
##  6                   Biola University      CA       2     21  4364    46162
##  7 California Institute of Technology      CA       2     12   983    58755
##  8     California Lutheran University      CA       2     12  2799    50425
##  9   California Institute of the Arts      CA       2     21   946    57704
## 10                 Chapman University      CA       2     12  6211    58948
## # ... with 364 more rows, and 4 more variables: COSTT4_P <chr>,
## #   FAMINC <chr>, AGE_ENTRY <chr>, CLUSTER <int>
```

With an interactive UI, a student could now sort by population, cost, and compare schools based on their own preferences and needs. This brings us to the next and final tool.

College Scorecard Sandbox