# Final Writeup

Statistical Methods for Data Analysis

Jay Sayre

Marlboro College - Spring 2013

## Introduction

Much of this project was inspired by the earlier work I had done for my Statistics class taken at Marlboro College with Matt Ollis in Spring 2012. That project explored general trend finding and the use of statistical techniques applied to a data set containing information about previous Marlboro College students. Then,we tried to answer whether studying broadly in the first two years meant that a student's outcomes would improve. Ultimately, our results came back inconclusive, but nevertheless we came away with some interesting findings.

Since then, I wanted to learn some more advanced statistical techniques, including linear regression, multiple regression, and other data mining methods, and try to revisit a similar data set. Most of this project has mainly been based around playing with new techniques and theory I have learned, applied to a very similar data set composed of information gathered about academics at Marlboro, but there has always been at least a vague question underpinning it. I wanted to answer the question of what predicts student success at Marlboro, a question that has been both difficult and fun to explore. Before beginning, my difficulties with this question were certainly not unique, as many academic institutions have tried and would love to find a definite answer to this, Marlboro College included. Although my findings do not provide any definitive profile for who will be the most successful here, I still have the hope that some of the insights gained through this research may be beneficial.

## Information Received

The information I received was an Excel spreadsheet containing information on students who have entered Marlboro College from the year 2000 on. In this time period, the spreadsheet contained entries for 1334 such students, including current students, alumni and students who have withdrawn from the college. Each column represented a different piece of information available about students. I will list them below:

Entry Date, the Current Status of the Student, Graduation Date (if applicable), Ethnicity, Gender, Home Zip Code, Citizenship Status, Whether the student was a first-year college student or transfer, Admissions track (either regular admission or early action), if Financial Aid was requested, Admissions Referral Source, High School grade point average [GPA] (for home-school students, this number was often made up based upon an admissions counselor's judgment), Marlboro GPA, Plan of Concentration Field(s) of Study, Plan Grade, ACT Test Scores for all ACT's submitted, which included the Composite Score, Date the Test was administered, Date the Marlboro College Admissions Office Received the Test, English, Math, Reading, Science and Writing Scores. It also included information on AP tests for however many AP's a student decided to take (up to 5), including the Date the Test was Administered, Date the Marlboro College Admissions Office Received the Test, Test Score, and Test Type. SAT test data was available as well for all tests taken by students, containing the Date the Test was Administered, Date the Marlboro College Admissions Office Received the Test, Critical Reading, Math, and Writing Scores.

I also received information about every course a student had taken during their time at Marlboro, which included course numbers, course titles, number of credits attempted per course, the number of credits earned, and the class standing the student had by the time when they took the course. This, at least at first, seemed to be a fairly comprehensive list of information about students at Marlboro.

However, through my group work in the Painting by Numbers class, Elisabeth Joffe, Emma Rusbarsky and I became interested in being able to quantitatively address the effect that Marlboro College has upon one's writing ability. This lead us to seek out even more data on students at Marlboro, particularly Initial Writing Placement Scores and Clear Writing Requirement data. This data came in the form of a column for the initial writing placement a student received, and then a column for the clear writing requirement score each time a student submitted.

This is all the data that culminated in this final project, and the corresponding writeup. Once this information was assembled into one spreadsheet, there ended up being 1334 rows and 73 columns, meaning there are 97,382 entries to examine. This is certainly a daunting task, but ultimately, the role of a statistical researcher (or student learning statistics, in this case) is to distinguish the signal, or what we are looking for, from the noise inherent in data sets as large as this. Fortunately (or unfortunately), there did seem to be plenty that could be discarded in the hopes of predicting anything useful.

## Assembling a Data Set

It would have been useful to have had the data come from one source, but unfortunately this wasn't the case. Therefore, a considerable amount of time was spent learning Excel functions to match up student information together. In the interest of anonymity, randomized numerical student ids were generated instead of using student names to index entries, and some part of the time was spent putting sources received from various sources together. Ultimately, having multiple data sources was problematic and may have introduced some errors in this analysis for several reasons.

The first reason is reliability of some of the data sets we/I obtained, particularly the information on writing assessments at Marlboro College. John Sheehy has been collecting writing information at least since the year 2000, but especially in the case of initial writing assessment data, has had little intentions to do anything with said data. Anecdotally, before this point, I was informed that initial writing assessments were not even recorded at all, out of the interests of not having student's scores follow them around through their Marlboro career. Therefore, it is unlikely that much care went into the precision of data entry, especially compared to information received from the Registrar's office. This is evidenced by the fact that writing score information was only available for 1022 entries out 1334 students total in the time period. The missing students appeared to separated randomly, not simply missing for a given period of years. This led Bill Mortimer to joke that this was information obtained in a memory stick under the floorboard of John Sheehy's car, an apt description.

The second is that in order to utilize the writing information, I had, in some cases to manually correct errors in the spreadsheet I was given, on a best guess basis. While this increases the likelihood that writing information for any particular student is incorrect, it is unlikely that this affects writing information in the aggregate by any reasonable amount. This is due to my presumption that my best guesses were usually correct, and that there were only 182 entries to correct.

The last source for potential error, then, lies in the difficulty for outside observers to cross validate the information I have computed. Fans of the R-statistical package often criticize the use of Excel to perform statistical analyses because it requires those wishing to check the validity of statistical experiments to have to repeat the same steps the original researcher did in computation. Often times, this is extraordinarily difficult, if not impossible. More discussion of this will be continued in the Methodology section.

## Methodology

This project was completed through use of LibreOffice Calc (similar to Excel), and the R statistical computing package. As a result, much of what I have done can (fairly easily) be

retraced and computed independently. Throughout this writeup, I will refer to what I have done in R, and therefore, attached are the R-script used (final.R) and the spreadsheet the data was analyzed from (export.csv and export.xls). Therefore, before beginning, it is worth mentioning what I have done that cannot be retraced easily (or, what I did in Excel). The first, as already mentioned, is compiling the information taken from different sources into one grand data sheet. One of the great parts about R is that when a researcher receives new data, it is very simple to run the same statistical tests previously done on another data set. Because of this, I have tried to write R code which allows for the numbers to be re-crunched when more student information comes in year after year. Then, in order to accomplish this, there simply needs to be one person who is responsible for collecting relevant student information year after year, so that compiling information from different sources won't be necessary (more than that later).

The second is the removal of information that seemed extraneous to the question at hand. Although I won't mention all of what I removed, data like the date the Marlboro College Admissions office received a test score seemed irrelevant and therefore was removed from the final spreadsheet. Finally, I wrote various formulas to match and re-categorize the data given to me. In an attempt to figure out if a student had taken a Writing Seminar or not, I had to look through the course titles of every class that student had taken. Unfortunately, not every class listed as a Writing Seminar was titled as such, and therefore, I had to look through the Marlboro College course catalog online, which contains information on courses offered through the year 2002, to find and label the courses that were designated Writing Seminars. Again, this was the only other part of the process that was non-reproducible, and could be easily updated if one wished to put the data into the format I have used.

The rest of the data analysis happened through use of the R statistical computing package. Part of the beauty of R is that all code ran can relatively easily be reproduced and cross-validated. In each of the attached R-scripts, a different statistical technique is applied to the data, and each will be discussed in its own section in this writeup. For each, when new data about students comes in, each can be re-ran to see whether these results are changing over time, or whether results appear to be holding up. Each particular technique will be described in a later section.

# Difficulties of Using Observational vs Experimental Data

Ultimately, Marlboro data presents huge challenges to anyone wishing to find trends or causal effects on how does well here. Most of this difficulty is due to the fact that basic measures do not adequately summarize an individual, a result no doubt confirmed in college admissions offices everywhere. However, some of the difficulty is due to the fact that most of this information was collected not for the purpose of statistical analysis, rather created for (and a result of) daily institutional use. As such, there is plenty of information that if stored and archived, would have been fantastic to incorporate into the scope of this project,

but ultimately, was lost/thrown out for various reasons. One such example is the writing score on a prospective student's application. The admissions office scores both the writing submission an applicant submits from previous classes and how well they feel the applicant would fare in college, and use these numbers in their decision to accept or deny a student. Once the incoming class arrives at Marlboro, though, these scores are thrown out, out of concern that they might follow a student permanently through their time here. This is a valid concern, and the opportunity for a fresh start should not be de-emphasized. However, this would make for a great measurement to analyze, and see how well the admissions office scores end up being as predictors of student success. Other challenges include gaps in the data due to institutional procedural changes and in many cases, a lack of data entry in the first place. These and other reasons make data analysis a very difficult prospect at Marlboro.

However, these concerns are not unique. Although some foresight into the collection of data would be helpful in terms designing real life experiments and data analysis, Marlboro College is not alone in having more observational data than it does experimental. Indeed, in the so called era of **big data**, billions of terabytes of raw, unprocessed observational data is available on institutions and just about everything, little of which has even begun to be processed and analyzed. Therefore, this writeup builds from the vast amount of data analysis literature aimed more at fixing the often unique problems found in datasets than looking at perfectly designed experimental data.

# Linear Regression

Linear regression is an approach that aims to model a (often times) linear relationship between an explanatory (or x) and response (or y) variable. It takes the general form $y = E(y)+\epsilon$, where $E(y)$ is the expected value for y given x, and $\epsilon$ represents the unexplained variation in y measurements caused by random phenomena, or the "special snowflake" effect for our data. Linear regressions can be used to determine how strong a relationship between two (or more, in the case of multiple regression) variables is, and can often be used as a predictive tool (at least, when our x value we are trying to predict is in our sample range).

Therefore, it is easy to see why linear regression might be a helpful technique to use on Marlboro data - to reveal correlations between variables. Of course, some care must be taken to make sure one does not, as the adage goes, interpret correlation as causation. In order words, just because we have a linear relationship does not imply an actual, real world, relationship. Then, some care was taken to make sure any correlations found actual meant something – which generally just involves use of common sense. For example, one result we found in our group project was that Marlboro GPA "explains" about 60% of the variation in Plan GPA. This does not mean that one's 3 year grade point average causes one to have a different Plan grade. Rather, it is likely that students who are hard workers will remain so on Plan, and likewise for the converse. As such, although the results of linear regressions are illuminating, it is not sufficient to accept them at face value.

In simple linear regression, one can model a linear relationship by the form of the equation given by $y = \beta_0 + \beta_1 x + \epsilon$, where $\beta_0$ is the y-intercept of the line, and $\beta_1$ is the slope of the relationship. After receiving observational (or experimental) data, one can estimate the $\beta$'s through the method of least squares, which aims to find just that - the line which results in least squared vertical distance between it and residuals, or outlying points. After confirming that with a statistical measure of confidence that the model is better than no relationship, one can check how well the linear model fits the data based on the coefficient of determination, or $r^2$ value. Once the $r^2$ value proves a correlation (and we prove some causation), we can use this to generate prediction and confidence intervals for y.

In R, we use the following command to build a linear regression model:

```
> model <- lm(y~x)
```

And then to generate summary statistics about model adequacy:

```
> summary(model)
```

I will explain these summary statistics in the multiple regression section.

Fortunately, it is pretty easy to plot linear regression models with a scatter-plot and line of best fit. It can be easy to how well the model fits the data this way, and determine whether or not there is another relationship happening in the data. The data could be displaying a characteristic where the y-values are increasing exponentially, where it could be useful to plot the y-axis on a logarithmic scale. Alternately, the data could model a curved, or nonlinear, relationship, in which case, we may want to speculate about interaction terms or whether or not higher order models are appropriate. These will be discussed in the multiple regression section.

# Multiple Regression

Multiple regression takes the same theory of linear regression and uses it to model several explanatory variables, instead of just one. The R code to move to a multiple regression model is pretty simple, given by:

```
> model <- lm(y~x_1+x_2+...+x_n)
```

Then, multiple regression models become pretty difficult to plot with just a scatter-plot and line of best fit. Quickly, we have to rely on statistical measurements of adequacy and not graphical representations of best fit relationships.

### Measuring Model Adequacy for Multiple Regression

In R, the summary function generates measures of adequacy for multiple regression models.

```
> summary(model)
```

As before in simple linear regression, R uses the least squares method to fit the regression line to the data. But now, we need a global test to check if the entire model is statistically significant. This test comes in the form of our F-Test and rejection p-value. In order to see if our model is significant, we can compare our f-test to a tabulated f-test based on the df and freedom in the numerator and denominator (which r tells us, but are given by) of $\frac{k \ \mathbf{df}}{n-(k+1) \ \mathbf{df}}$, where n is our sample size, and k is the number of $\beta$'s in our model. If our f-test is greater than the tabulated rejection f value, our model is significant. Similarly, our rejection p-value gives us the confidence level that we have for whether or not our f-value is larger than our tabulated f-value.

In addition, since we now have multiple parameters, it is worth finding some way to determine which ones are significant and which are not. For each $\beta$, we are looking for the likelihood that the true coefficient of the parameter is zero, or $\beta = 0$, and adds nothing to our model. There are two methods of evaluating this in order to try to reject this null hypothesis. They are t statistics and our p value, noted by the R output as t value and $Pr(> |t|)$ respectively. Let's look at a baseline multiple regression model I built to compare some parameters against GPA at Marlboro.

```
> model <- lm(Plan.GPA ~ Marlboro.GPA + Median.Income + Population)
> summary(model)
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.632e-01  1.091e-01   2.412   0.0162 *
Marlboro.GPA   9.611e-01  3.064e-02  31.367   <2e-16 ***
Median.Income  1.662e-07  3.272e-07   0.508   0.6118
Population    -2.318e-07  6.760e-07  -0.343   0.7317
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

So, this model looks at how significant the Population Density of a student's neighborhood, median income of said neighborhood, and Marlboro GPA are in terms of predicting the score one should get on Plan, assuming that whatever relationship we might have, it is linear and that our x's are probabilistically independent and do not exhibit multicollinearity. It's worth mentioning what our t-value is based on. It is given by $\frac{\bar{\beta}_i}{s_{\bar{\beta}_t}}$ or the sample mean of the parameter over its sample standard deviation. Then $Pr(> |t|)$, or the p-value, is based upon the rejection region for a parameter, as given by $|t| > t_{\alpha/2}$, where $t_{\alpha/2}$ is based

upon n-(k+1) degrees of freedom [5, p. 176]. The p-value measures the likelihood that t is actually in this rejection region.

So then, let's look at the p-value for these parameters. The output states that Marlboro GPA is significant at 99.99% confidence level. However, since both .6118 and .7317 ¿ 0.05 (or whatever confidence level you wish to use), it seems pretty unlikely that the Population density or median income of one's area influence Plan grade.

Let's examine the output for the $R^2$ and $R_a^2$ for this model before moving on.

```
Multiple R-squared: 0.6159,Adjusted R-squared: 0.614
```

This suggests that even though Population and Median Income are insignificant, the Marlboro GPA continues to make this model look significant. Therefore, looking at individual parameters is an important step. Although Mendenhall et al. only recommend evaluating the interaction or squared $\beta$'s, since R conducts the test on every variable it's still worth looking at the individual parameters, but obviously put the major emphasis on the p-value of the most important $\beta$'s [5].

I've written more about adequacy tests for multiple regression here.

**Determining what to put in Multiple Regression Models**

Ultimately, regression analysts are in search for the multiple regression model which explains the response variable the "best". Most of the time, this is a guess and check process. However, there is a way to automate the process called, "best subset selection". There are a variety of best subset approaches in R, including forwards and backwards stepwise best subset modeling. In forward best subsets procedures, parameters are gradually added, until all the possible parameters are in the model. In backwards, the converse method is applied. Backwards modeling usually results in higher multiple R-squared values for a given model with k parameters, but requires that a model has more observations than parameters.

The R code below is an example of one such best subsets procedure in both directions:

```
> library(MASS)
> model <- lm(y~x1+x2+x3)
> step <- stepAIC(model, direction="both")
> step$anova # display results
```

Another method is through use of the regsubsets() function:

```
> model <- regsubsets(y~x_1+x_2+x_3, nbest=2)  #nbest specifies how many
  # of the best models will be reported for each subset size
```

The upside of this function is that we can specify how many of the best models we want to see for given number of parameters, k. The reason this is nice is because whenever we plug in an additional $\beta$ into our model, we **always** will have a higher multiple R-squared $R^2$ value. Therefore, regardless of whether or not what we plug in actually has that much value, it will appear that our model is enormously predictive. Then, it is useful to have a model that is *parsimonious*, or has the maximum predictive value for the smallest number of explanatory variables. Therefore, it is useful to look for the most predictive models with the fewest $\beta$'s, and the "nbest=" helps with this.

The downside of automated best subsets procedures is that they require the size of the sample to not change between each model. This is not a problem with a complete data set. However, with the missing entries in our data set (going back to the challenges of observational data), it is difficult to use stepwise regression procedures.

In *A Second Course in Statistics: Regression Analysis*, Mendenhall et al. mention that one way to improve regression models is to hypothesize whether there are non-linear relationships in the data due to higher-order terms or interaction terms. Interaction terms represent where two explanatory variables may have some underlying relationship. Higher order variables may explain relationships where there can be an increasing/decreasing effect of an explanatory variable on the dependent variable [5]. In simple linear regression, these relationships are easier for one to postulate about and test out. However, in multiple regression, these relationships are difficult to plot graphically, and sometimes very difficult to guess. Therefore, it would great to have best subsets procedure for checking interaction terms and higher order explanatory variables. To my knowledge, both regsubsets() and stepAIC() do not directly allow for this, and I'm not sure what function does. Regardless, this would result in a massive number of permutations of models to check, and would probably require way more computing power than available on most computers, including my laptop. However, we can plug in individual guesses for interaction terms by using the R code:

```
> xint = x_1*x_2
> model <- regsubsets(y~x_1+x_2+x_3+xint, nbest=2)
```

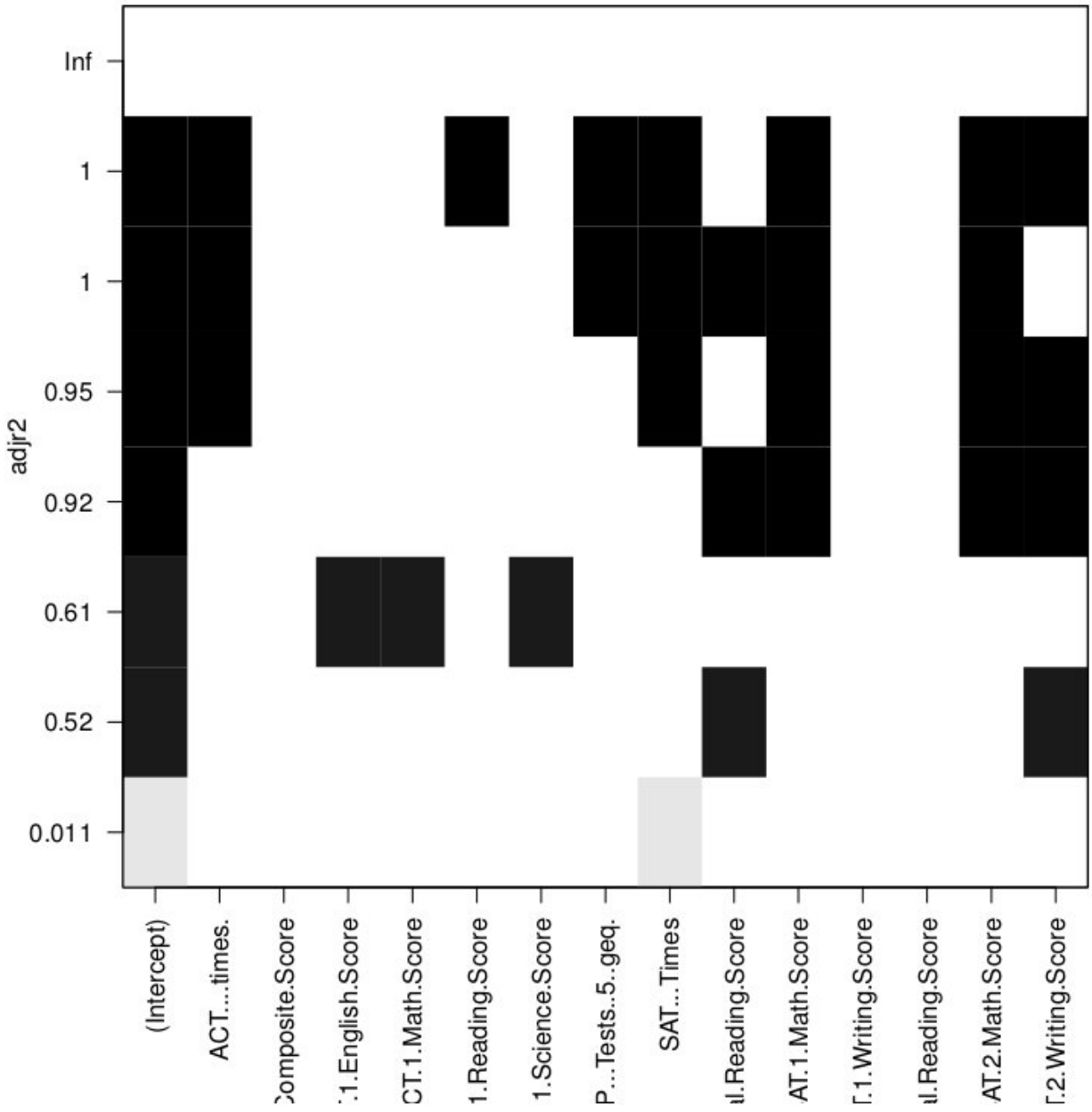
Similarly for higher order variables:

```
> xhigh =(x_1)^2
> model <- regsubsets(y~x_1+x_2+x_3+xhigh, nbest=n)
```

Once we have a few candidates, we can either use ANOVA tests to compare various models or can plot which combinations of parameters lead to various r-squared values, using the r command:

```
> plot(model, scale="adjr2")
```

Which gives us this image:




Ultimately, these methods can be used to generate useful and significant multiple regression models aimed at predicting student success at Marlboro.

# Logistic Regression

Logistic regression aims to adapt the techniques and theory of linear regression to use where a researcher is aiming to find a relationship between explanatory variables and a categorical,

qualitative response variable. One example of where this is useful is trying to model what characteristics are correlated with whether a student withdraws or not.

Then, let

$$y = \begin{cases} 1 & \textbf{if student withdraws} \\ 0 & \textbf{if student graduates} \end{cases}$$

Our logistic regression model, aims to predict $E(y)$, or the probability that a student withdraws, $P(\textbf{withdraw})$, based on the explanatory variables input into our model. Our model for the probability that a student withdraws, then, is $E(y) = \frac{\textbf{exp}(\beta_0+\beta_1 x_1+\beta_2 x_2 \ldots \beta_n x_n)}{1+\textbf{exp}(\beta_0+\beta_1 x_1+\beta_2 x_2 \ldots \beta_n x_n)}$. This is sometimes alternatively written as $ln(\frac{P(\textbf{withdraw})}{1-P(\textbf{withdraw})}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots \beta_n x_n$, where $\frac{P(\textbf{withdraw})}{1-P(\textbf{withdraw})} = \frac{P(\textbf{withdraw})}{P(\textbf{graduate})}$, representing the odds of $y = 1$ occurring, and is called the *log-odds model* [5, p. 497].

In R, estimates of the explanatory variables are generated through maximum likelihood estimation, where test statistics for individual parameters and overall model adequacy have approximate chi-square $\chi^2$ distributions. Now, let's look at generating logistic regression models in R.

The following R-code generates a logistic regression model.

```
> model <- glm(y ~ x_1 + x_2+ x_3, family=binomial)
> summary(model)
```

As before for linear regression, the summary() function provides us with measures for model and parameter significance. This generates multiple statistics, including residual and null deviance. A high residual deviance shows that the parameters are unlikely to have modelled the logistic function for P(withdraw)=1, on a given $\chi^2$ distribution and number of degrees of freedom. Then, $Pr(> |z|)$ gives a measurement similar to our p-value for the parameters of linear regression. Estimates are chi-square estimations of the effects of each parameter, $\beta_i$. These estimates can be interpreted as percentage change in the odds of a student withdrawing, P(Withdraw)/P(Graduate), for every 1 unit increase in $x_1$, holding all other $x$'s fixed, if this estimate is transformed into $e^{\beta_i} - 1$.

Analysis of the usefulness of each of the parameters is given by the R code:

```
> anova(model, test="Chisq")
```

Where the ANOVA Test tries adding the factors in one by one, and measuring the difference in residual significance for each one, allowing one to determine which inputs are the most significant. Then, if one wants to compare their original model to a new one developed through an ANOVA test (or guess and check), one can determine the confidence threshold of a new model being better than a old one by the R code:

```
> test <- pchisq(oldmodelresidualdeviance - newmodelresidualdeviance, 1)
> percent(test * 100) #A R function I wrote turning a number into a percentage
```

Where we are given the confidence threshold of our result [4].

# Non-parametric Regression

Although use of interaction terms and higher-order linear models in linear regression allows linear regression to be plotted to non-linear relationships, it often times can be easier to begin with the assumption that the relationship is non-linear and use non-parametric regression models instead. The main object of non-parametric regression, then is not estimate the various parameters of a graph, but rather to the estimate the regression function itself. Therefore, such methods are frequently used for trend finding, rather than prediction.

The general non-parametric regression model, then, is written $y_i = s_i(x_i) + \epsilon_i$. All models, use some sort of smoother $s_i$ in order to define how "fitted" the function or curve will be to each data point, which are estimated by either cubic smoothing splines or thin-plate smoothing splines. This non-parametric model can be scaled up to have multiple explanatory factors, given by the additive regression model, $y_i = \alpha + s_1(x_{i1}) + s_2(x_{i2}) + \cdots + s_k(x_{ik}) + \epsilon_i$ [5, pp. 513]. In multiple non-parametric regression, it is difficult to use this for trend finding (since multiple dimension graphs are hard to read) and there is no analogy to the multiple $R^2$ value, so the utility of these is hard to measure. Ultimately, one can use ANOVA tests to see whether non-parametric models with more parameters are more significant than those with less, in order to determine significance of a larger model.

There are many different ways/functions that employ some sort of non-parametric regression, but for models with multiple explanatory variables, we can use the R code:

```
> model <- loess(y ~ x_1 + x_2, span=0.5, degree=1)
```

In this case, span defines the fixed proportion of the data set that each local regression function covers. Therefore, the larger the span the smoother the resulting non-parametric regression will be.

```
> summary(model)
```

Gives us the significance results, again, where the likelihood of a non-parametric model is maximized when the sum of the squared residuals is minimized, given by $S(\beta) = \sum_{i=1}^{n}[y_i - s_i(x_i)]^2$. Then, ANOVA tests can be used in order to determine which parameters minimize $S(\beta)$ the most.

```
> model <- loess(y ~ x_1 + x_2, span=0.5, degree=1)
> model2 <- loess(y ~ x_2, span=0.7, degree=1) #Span=0.7 since sqrt(0.5)~=0.7
> anova(model2, model) #Tests for adequacy of the x_1 variable
```

The ANOVA test outputs an approximate F-test for the change in the residual sum of squares. This process can be roughly compared to evaluating the change the multiple r-squared $R^2$ value by plugging in different $\beta$'s [1].

# Machine Learning for Data Mining

Machine learning is defined as the non trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. [6] With machine learning, we must must make a distinction between supervised and unsupervised learning techniques. Supervised learning, is primarily concerned with using one or more predictor variables to estimate what a value a response variable may have, or in terms of probability theory, aiming to predict $Pr(Y|X)$ [3, pp.485]. With supervised learning there is a clear measure of success, or lack thereof, that can be used to judge adequacy in particular situations and to compare the effectiveness of different methods over various situations. [3, pp.486] However, the goal of unsupervised learning is to directly infer the properties of a probability density function with the help of a degree-of-error for each observation, where there can be a higher number of x observations than in supervised learning, and the properties of interest are often more complicated than simple location estimates [3, pp.486]. Roughly, unsupervised learning can be used to see trends in large data sets, where supervised learning is much easier to use if one already knows which trends to look for. In large observational data sets like the one we received, it is easy to see why the former would be beneficial, although it might lose the benefit of some predictive value.

There are a variety of machine learning approaches, but undoubtedly one of the most popular is association rule learning. As one of the most prominent methods in both machine learning and data mining [6], I figured that I would focus on this technique instead of a swath of different data mining methods out of preference for understanding one method more in depth and applying it than obtaining a broad overview of the field, which appears to be categorized by a variety of different techniques, largely separated into different categories because of their practical applications. Other popular techniques include cluster analysis, which involves looking for patterns in data sets, and anomaly detection. I'd like to explore these at a later point, but for now I will focus upon association rule mining.

Association rule mining looks for rules found in a data set, and the incidence of those rules in the data. It is a form of unsupervised learning, helpful to a statistical researcher wishing to find patterns in a data set which then can be plugged into a supervised method, such as linear regression. Although association rule mining can be used to observe rules in any data set, it was originally developed to help retailers predict what items a customer is likely to buy given their previous purchases (see: Target knows you're pregnant). Therefore, much of the documentation for association rule mining algorithms is written in a format specifically designed for retail use. However, I will explain the method in a way describing how it is used on the Marlboro data set, indicating that association rule mining can be used

for a variety of purposes and not just retail sales.

# The "arules" package and Apriori Association Rule Mining Algorithm

One of the implementations of association rule mining in R is the "arules" and "arulesViz" package [2], and Apriori algorithm for generating rules. The Apriori algorithm takes a breadth first approach to searching for rules, aimed at being able to be run on less powerful computer, such as mine [6]. Therefore, for this project, I used the arules package. I will begin by showing the inputs arules takes, and the theoretically underpinnings of the arules/apriori algorithm, learned from mainly from the Hahsler et. al paper, Introduction to arulesA Computational Environment for Mining Association Rules and Frequent Item Sets.

### Theory of Association Rule Mining

To begin, association rule mining requires a set of items, or things, that we wish to find rules or associations between. Then, let $I = \{i_1, i_2, \ldots, i_n\}$ be the set of n items. Additionally, let $D = \{s_1, s_2, \ldots s_m\}$ be a set of m students, or in their case transactions, called the database. Then each student in the database, $s_j \in D$, has a unique student ID and for each student row in the database contains a subset of items in I, $s_j \subseteq I$.

Then, a rule is defined by the form $X \Rightarrow Y$ where X, Y are both subsets of items in the set I, $X, Y \subseteq I$, and $X \cap Y = \varnothing$, or both X and Y share no items in common. It is worth giving an example of how this can work. Therefore, let $I = \{$pass clear writing, pass initial writing, graduate, drop out, take writing seminar$\}$. Then $X = \{$take writing seminar, pass initial writing$\}, Y = \{$pass clear writing$\}$. Then $X \Rightarrow Y$ would be a rule that says that if a student passes the initial writing test and takes a writing seminar, they will also pass the clear writing requirement. In large data sets like ours, though, this might lead to an inordinate amount of rules, and therefore, there must be some constraints on significance that can be used to narrow down the amount of rules.

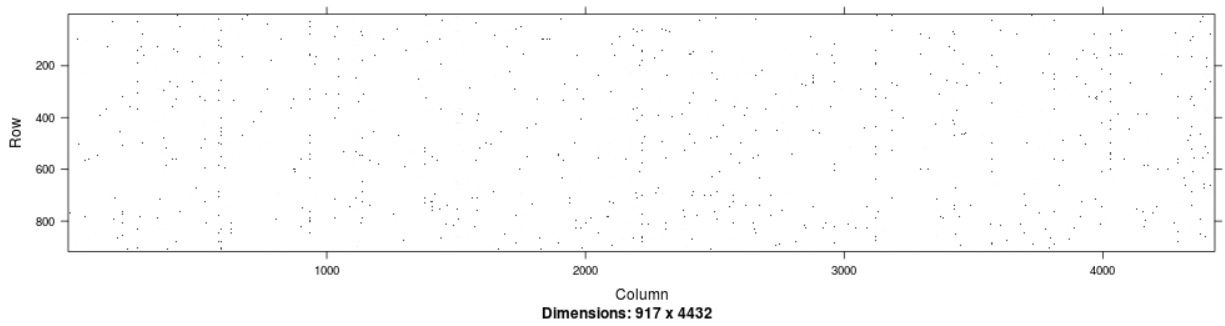
And, of course, there in fact are such constraints. The first is support, or $supp(X)$ of a set of items X. It is defined as the proportion of students in the database which have taken a writing seminar and passing initial writing, over total number of students. It approximates the probability that a student has taken a writing seminar and passed initial writing, or $P(WS \wedge IW)$. The second is the confidence of a rule, written $conf(X \Rightarrow Y)$. Roughly, it approximates the probability of Y given X, or $P(Y|X)$. In this case, it gives the number of students for whom have taken a WS, passed initial writing, and pass clear writing, over the total number of students who have taken a WS and passed initial writing. Then, confidence of a rule is given by $\frac{supp(X \cup Y)}{supp(X)}$.

If one seeks additional measures to pare down the sheer amount of rules, it is possible to use the measure of lift, written as $lift(X \Rightarrow Y)$. It is defined as $\frac{supp(X \cup Y)}{supp(X)supp(Y)}$. It can be interpreted, as Hahsler et al. write, "as the deviation of the support of the whole rule from the support expected under independence given the supports of the [left and right hand side of the equation]", where "greater lift values indicate stronger associations" [2, pp. 3].

**Inputs to arules**

The arules package requires all data to be coerced to a "transactions" (mirroring the retail-oriented focus of the package) data format. As such, transforming one's data into a transactions format is one of the most difficult steps in the process. The transactions data format is a binary incidence matrix, where the cardinality of the item set $I$ is equal to number of columns and the cardinality of the database (or number of students) is equal to the number of rows. Then, each column corresponds to a different item, with 0 representing the student not having the item, and 1 representing the presence of the item.

From the attached *associationrules.R* script, a binary incidence matrix resembles the form:




However, the input does not have to be in this form, as the arules package can convert data into this format itself. The data can be read in .csv format, or from a regular data frame format in R. The "associationrules.R" script goes through a few variations of how data can be input into a transactions format, but let me give an example here.

If this resembles roughly our data set

| $StudentID$ | $items$ |
|---|---|
| 1 | $takenWS, passedCWR, passinitialwriting$ |
| 2 | $takenWS, passedCWR$ |
| 3 | $passedCWR, passinitialwriting$ |
| 4 | $passedcwr, takenWS$ |

then this would be transformed into the transactions format:

| $StudentID$ | $takenWS$ | $passedCWR$ | $passinitialwriting$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 1 | 0 |

There are two separate formats the data set can be in: single and basket format, and both formats do not take headers, which need to be removed beforehand. Single format has one item per row, whereas the basket format can have multiple items per row, separated by columns. What is interesting to note is that in single format, one can enter multiple items per student by simply having multiple rows with the same student id and different items per each row. I used this technique to analyse student course data, which naturally is well suited to this format.

In single format, this R-code will read in our data:

```
> clsdata <- read.transactions(file, sep=",", format="single", cols=c(1, 2))
```

Where the first column "1" refers to the student ID, and the second column "2" refers to the column the item is in. In single format, if there is any information outside of these two defined columns, it is not analysed.

In basket format, this R-code will read in our data:

```
> All<-read.transactions(file, format="basket", cols=1, rm.duplicates = TRUE)
```

In this case, each column represents a different item that one student will have. This is most suited to the table of students and all of their attributes. "cols=1" refers to that the first column is where student id's are stored, but the rest remains undefined. One can also do this without student ids, assuming each row is a different student, with the change "cols=NULL".

**Using arules and arulesViz**

Once one has coerced their data into a transactions format, then one can finally mine for association rules. (All this is commented on in the associationrules.R script). This involves using the apriori algorithm on the transactions data.

Then we can read in our data

```
> clsdata <- read.transactions(file, sep=",", format="single", cols=c(1, 2))
```

and then use the apriori alrgorithm to find rules based on a minimum support and confidence, in this case 0.01 and 0.5, respectively.

```
> rules = apriori(New, parameter=list(support=0.01, confidence=0.5))
```

In this case, we have 165 rules, a relatively small amount. However, it is not uncommon to have six figure numbers of rules, and therefore, we need some way to pare them down.

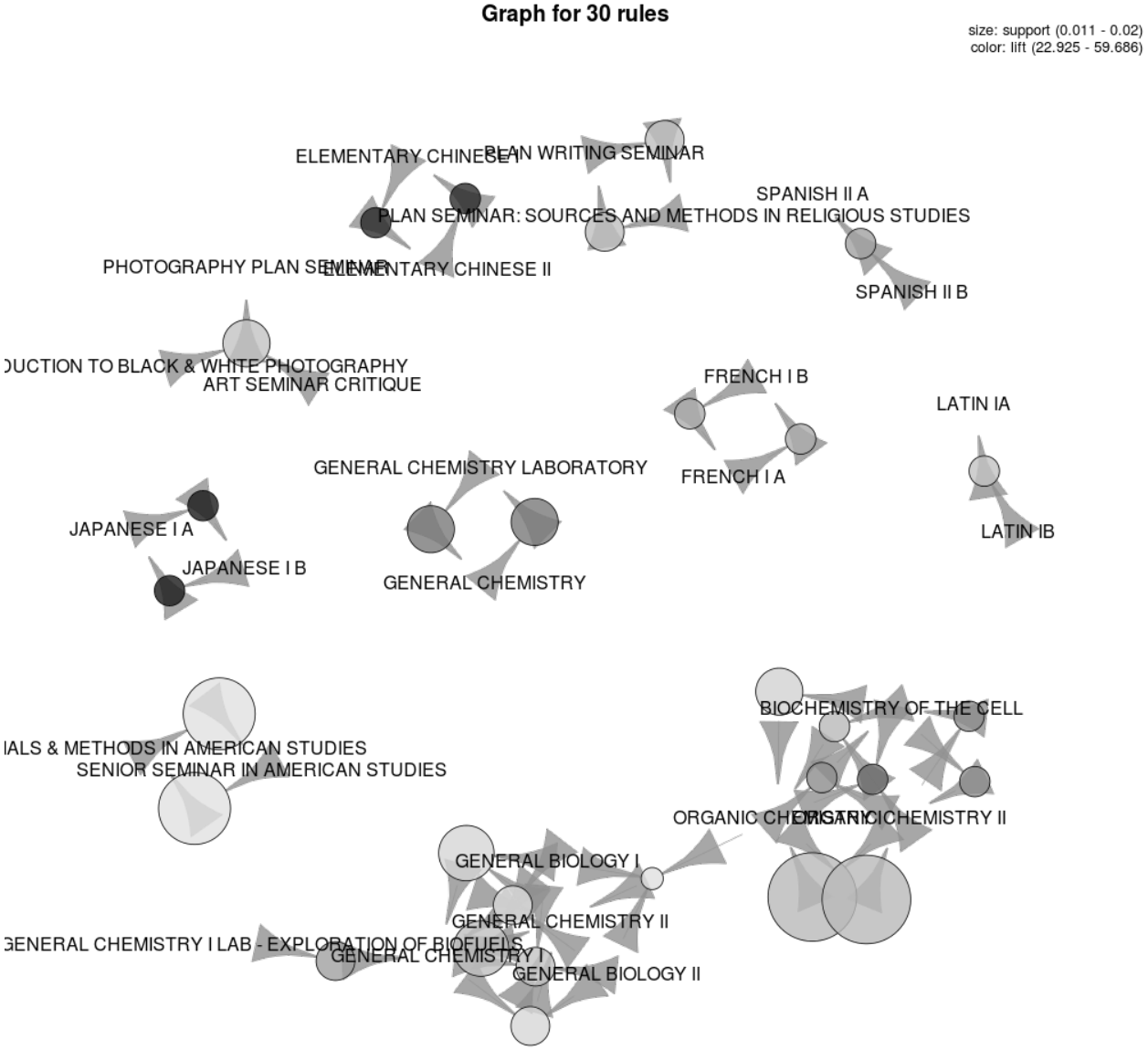
```
> subrules = rules[quality(rules)$confidence > 0.8]
> subrules2 = head(sort(rules, by="lift"), 30)
```

Are two ways to achieve this. Then, there are two things we can now do with these set of rules. One is to visualize them, and the other is to export them so we can look at rules for specific items, or any other purpose.

One way to visualize them is a sort of word-cloud esque graph which uses color to indicate lift and the size of the bubbles to indicate support. I'm personally preferential towards this. The R-code to generate it is:

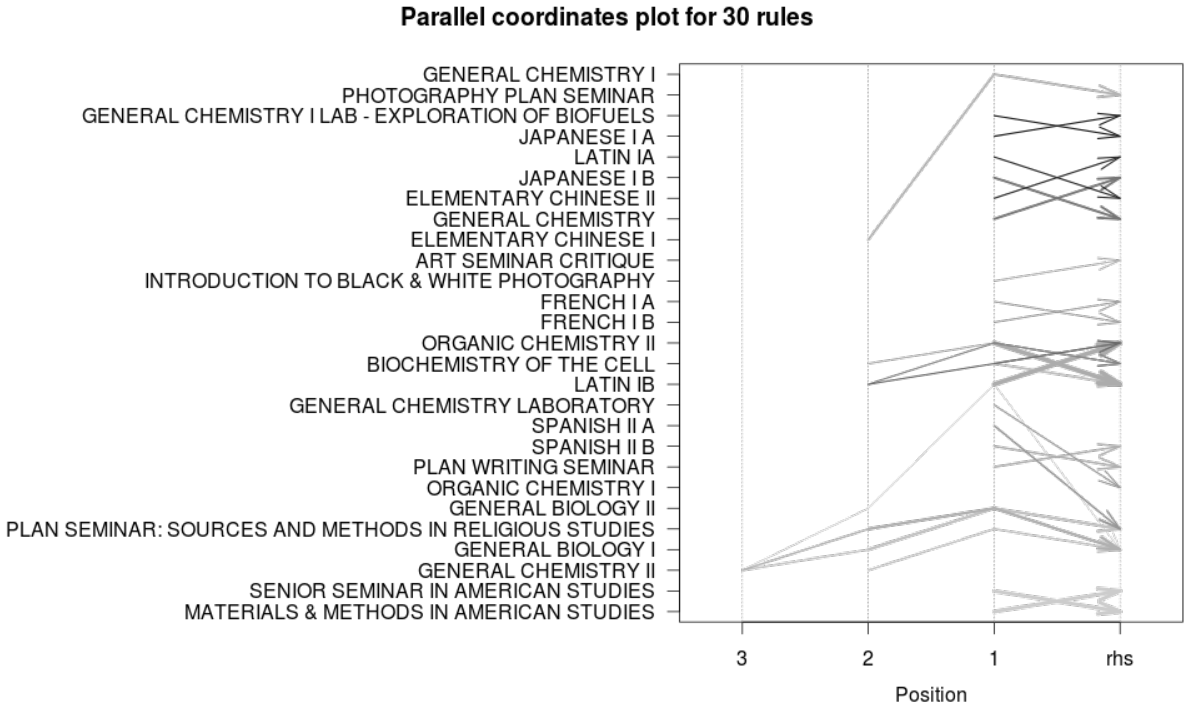
```
> plot(subrules2, method="graph", control=list(type="items"))
```

Which gives us:




Another method is plotting the most important rules on a graph that shows which items lead to each other. This is generated by:

```
> plot(subrules2, method="paracoord", control=list(reorder=TRUE))
```




This helps us see maximal item sets for a rule, which are not proper subsets of any other frequent item set.

Alternately, these rules can also be exported to a data frame so they can be used to look at specific rules using:

```
> rulesdf <- as(subrules, "data.frame")
```

And then, one can one their R tool of choice to narrow down to rules they care about, or find other ways of plotting this information. I'll leave this exploration to the reader, since I haven't found anything else to use to look at specific rules, other than exporting back into a csv, and then using CTRL-f in Excel. I'm sure there is a better way in R.

### Thoughts on Using Association Rule Mining

I believe that this can be used to help a researcher find interesting patterns in a data set, as an unsupervised learning tool. After finding these patterns, one could then plug results in to a supervising learning technique including linear or (in most cases, given the categorical variables) logistic regression. Ultimately, for this project, Association Rule Mining probably does not have enough information to determine really interesting results, but that could be changed. Unfortunately, all the inputs into the arules package have to be factors, so numeric information like GPA or SAT scores would have to be put into a categorical format like (low, middle, high) in order to be used in the arules package. I have not done this, but ultimately, I think it could result in some interesting results if one did.

# Conclusions

Each conclusion in this corresponds to a piece in the "final.R" script. The "final.R" script has numbers that will match numbers listed with each conclusion here. However, the "final.R" script has some sections that do not correspond to conclusions here, and instead give examples of each statistical method applied to Marlboro data.

### (1) - Revisiting the Painting by Numbers Project

Let's revisit this result and show some of the work behind it:


This result was generated through a simple linear regression model generated through the code:

```
> model <- lm(Plan.GPA~Marlboro.GPA)
>  summary(model)
```

First, the linear model deletes any entries (rows of students) that do not have Plan grades, which reduces the dataset down considerably, taking out 699 entries.

Then, the summary() call gives us our $r^2$ value of 0.6182, with a p-value that indicates a higher than 99.99% confidence level of statistical significance. This informs us that approximately 61.82% of our sample variation in Plan grade is "explained" by one's 3-year Marlboro GPA.

Also, we have a Residual standard error of 0.2576, as the sample standard deviation of our error $\epsilon$. Then we should expect most of our observed y-values (Plan GPA's) to fall within 2(0.2576) of the their predicted values given by the regression line (y=0.96x+0.25, in this case). Therefore, we have a small estimated standard deviation of $\epsilon$, a good thing for any model.

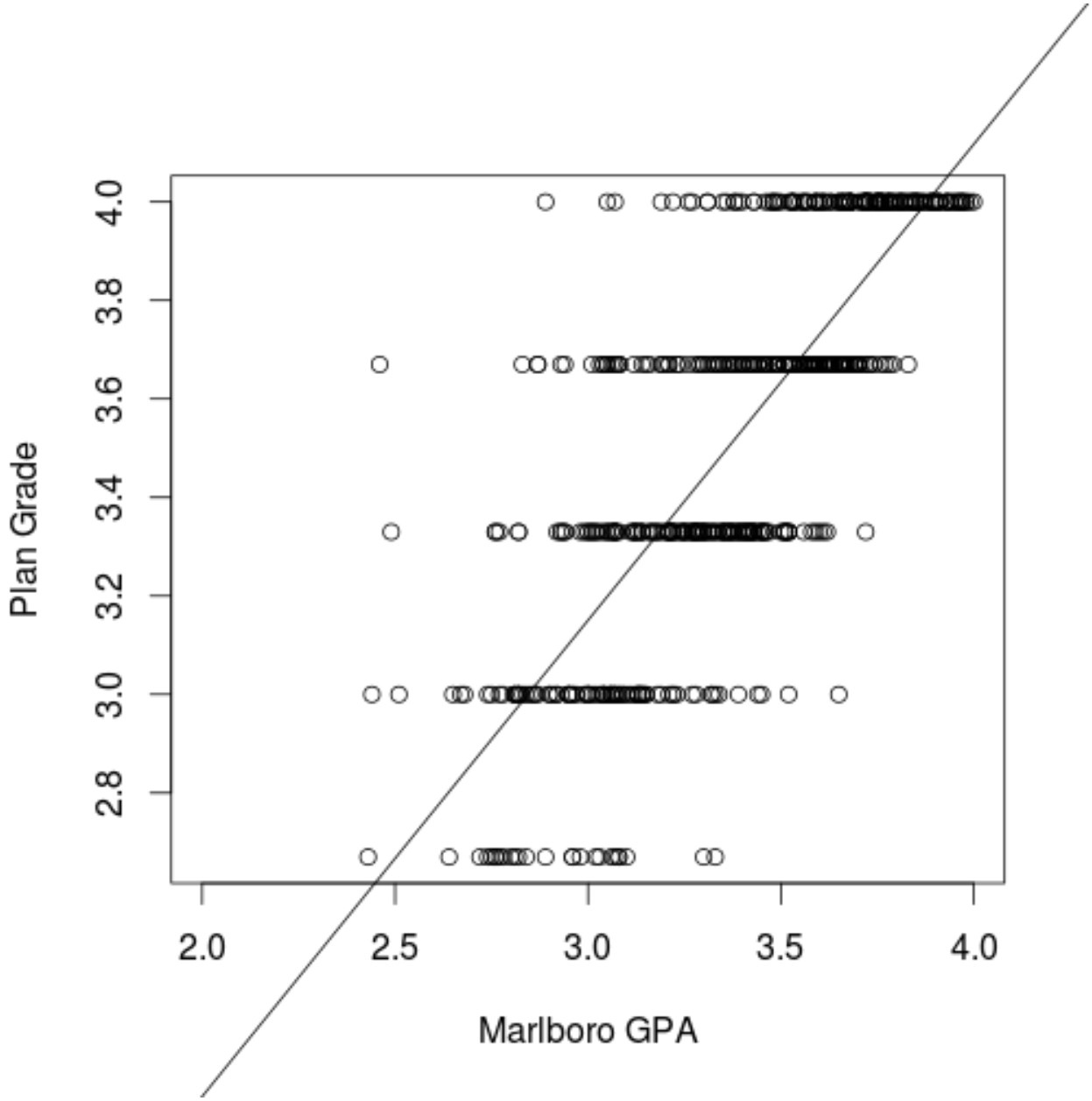
Another measure of model adequacy is given by the coefficient of variation, or the ratio between the estimated standard deviation of the error (given by s) and the sample mean (given by $\bar{y}$, given as a percentage by the formula $C.V. = 100(s/\bar{y})$. Generally, regression analysts prefer for this to be smaller than 10%. Plugging in with a sample mean of 3.526, we get a C.V. of $\cong 7.32$, which is lower than 10%.

Therefore, it appears that this is a statistically significant model. On the subject of correlation not implying causation, as stated earlier, we need to remind outside observers that this does not mean that 60% of your Plan Grade is based on your Marlboro GPA. Phrasing this correlation appropriately for our graph was one of the challenges we faced.

Since this is a simple linear regression, it is fairly easy to graph. I've tried two methods here, one plotting the best fit line obtained in the regression model, and another, using a non-parametric loess line, in order to do more trendfinding, and see how closely related the two linear and non-linear methods end up being.

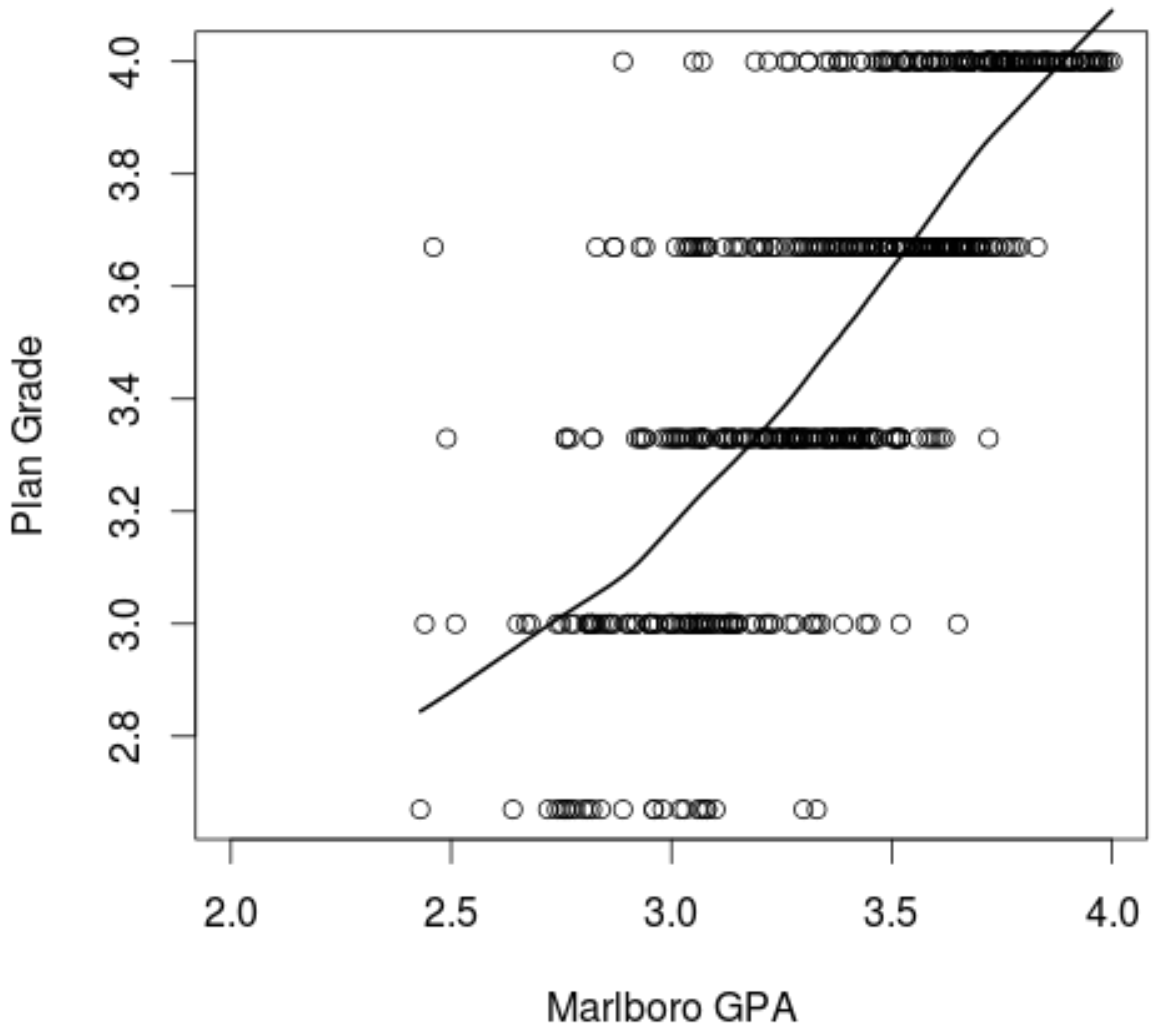
This plots our data and linear regression model:

```
> plot(x, y)
> abline(model)
```




This plots our data using a non-parametric model with the same parameters:

```
> plot(x, y)
> lines(lowess(x, y, f=0.5, iter=0), lwd=2)
```




They both look like they come to a similar result – that there is definitely an upward result, but for any particular student, given the spread of the points, this relationship is certainly not set in stone.

**(2) Building an "Ultimate" Model to Predict Student Success**

In order to determine which explanatory variables measure student success (or Marlboro GPA) the most, we can use best subsets procedures to plug in all the information provided in the massive data set I received. Unfortunately, though, it was not possible to simply "plug"

everything into a subset selection procedure, since for all the test scores, there was a large number of students who didn't submit any standardized test at all, since Marlboro is test optional. Therefore, I had only 268 students to look at the predictive effects of tests/test scores.

Let's begin with all the explanatory variables that didn't involve standardized tests. They were all plugged into a multiple regression regression and run through the stepAIC() method of subset selection, but only in a forward selection process, since there are missing entries in some of the columns that causes backwards methods to throw errors. This is given by the R code:

```
> fit2 <- lm(Marlboro.GPA~Number.of.WS.s+Pass.CWR.+Calculated.GPA+
  Referral.Source+Admissions.Plan+Fin..Aid.Requested+Program+
  Citizenship.Status+Ethnicity+Highest.Student.Level.for.WS+
  X..Portfolio.Submissions+Enter.Date+Writing.Placement+Placement.Factor+
  Highest.CWR.Score+Older.CWR.Score+Oldest.CWR.Score+Gender, data=alldata)
> step <- stepAIC(fit2, direction="forward")
> summary(step)
```

This revealed that out of all the possible explanatory variables we have, only 3 of them are statistically significant, and they are: Highest.CWR.Score, Writing.Placement[Up], and Referral.SourceReferral[Teacher]. This is really surprising and demonstrates just how much noise there can be in looking for a signal. Then, we can plug in Highest CWR Score and Writing Placement into a model (leaving out Source Referral because it only applies to a small subset of students), to obtain summary statistics about our model.

```
> model <- lm(Marl.GPA~CWR.Score+Writing.Caret)
> summary(model)
```

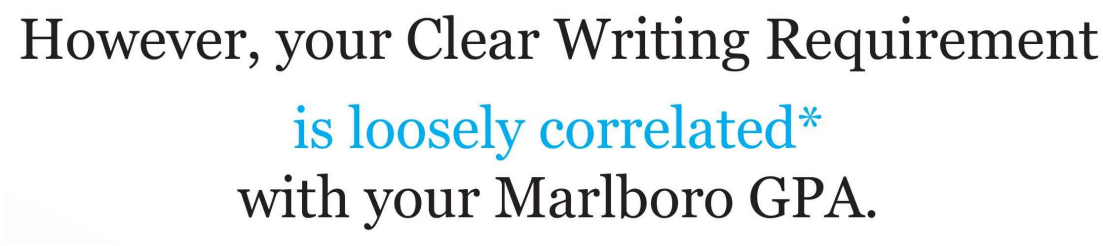
The function summary() then outputs:

```
Coefficients:             Estimate Std. Error t value Pr(>|t|)
(Intercept)               1.93884    0.06428  30.163  < 2e-16 ***
Highest.CWR.Score         0.36395    0.01791  20.318  < 2e-16 ***
Writing.PlacementSide     0.18023    0.04682   3.849 0.000125 ***
Writing.PlacementUp       0.29253    0.05873   4.981 7.29e-07 ***
Residual standard error: 0.58 on 1148 degrees of freedom
Multiple R-squared:  0.2814,Adjusted R-squared:  0.2789
F-statistic: 112.4 on 4 and 1148 DF,  p-value: < 2.2e-16
```

Then, we see that all of these variables are statistically significant, and have an overall multiple R-squared $R^2 = 0.2814$. Furthermore, this model is fairly parsimonious, and it

seems to account for a non-neglible amount of sample variation in one's Marlboro grade point average.

This, then, is also the underpinnings of another group result:




If one prefers non-parametric regression:

```
> model<-gam(Marl.GPA~Writing.Caret+CWR.Score, family=gaussian)
> summary(model)
```

Gives us a very similar result:

```
Parametric coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept)               1.93884    0.06428  30.163  < 2e-16 ***
Writing.PlacementSide     0.18023    0.04682   3.849 0.000125 ***
Writing.PlacementUp       0.29253    0.05873   4.981 7.29e-07 ***
Highest.CWR.Score         0.36395    0.01791  20.318  < 2e-16 ***
R-sq.(adj) =  0.279   Deviance explained = 28.1%
GCV score = 0.33789  Scale est. = 0.33642   n = 1153
```

For test scores, a similar best subsets process was repeated just for students who had taken those tests. As the result of this process, it was determined that the two most important quantitative predictors of Marlboro GPA are SAT Writing and ACT Reading. Independently, we can look at the results of the summaries for each.

For ACT Reading:

```
> test3 <-lm(Marlboro.GPA~ACT.1.Reading.Score)
> summary(test3)
```

Gives us:

```
Multiple R-squared:  0.05041,Adjusted R-squared:  0.03404
F-statistic: 3.079 on 1 and 58 DF,  p-value: 0.08458
```

We have a result that while technically significant, is only so on a 90% confidence interval given the p-value of 0.08458. I'd consider this to be interesting but not a particularly memorable result.

Then, for SAT Writing:

```
> test5 <-lm(Marlboro.GPA~SAT.2.Writing.Score)
> summary(test5)
```

Gives us:

```
Multiple R-squared:  0.1089,Adjusted R-squared:  0.1033
F-statistic: 19.43 on 1 and 159 DF,  p-value: 1.917e-05
```

This time, SAT Writing is statistically significant with 99.99% confidence, and looks as if it explains 10% of sample variation in Marlboro GPA. Maybe Marlboro should require SAT Writing, then.

The only other results that proved to be interesting, stemmed from conversations I had with other students in the Painting by Numbers class. They mentioned that it seemed likely that students who had submitted standardized test scores did so in order to use their good test scores to "signal" that they were qualified applicants, and Marlboro's real average test scores might be lower than they are in reality. This is impossible to prove. But I wondered – does submitting a test at all make you more likely to have a higher GPA? Therefore, I made up factors for whether a student had submitted standardized tests, and threw in Gender (based of earlier exploratory data analysis) into a multiple regression model to find out. Of course, R handles categorical explanatory variables correctly, so no changes have to be made. These were plugged into a best subsets procedure, which let us know that: Gender, Whether a student has taken AP tests, and whether a student has taken SAT tests, (but not ACT tests, oddly enough) have statistically significant results "explaining" Marlboro GPA.

We can evaluate this model here:

```
> model <- lm(Marlboro.GPA~Gender+Taken.AP.s.+Taken.SAT.)
> summary(model)
```

Which gives us:

```
Coefficients:  Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.15562    0.03113 101.365  < 2e-16 ***
GenderMale     -0.11118    0.04397  -2.529  0.01156 *
Taken.AP.s.Yes  0.39223    0.12123   3.235  0.00124 **
Taken.SAT.Yes  -0.35050    0.06103  -5.743 1.15e-08 ***
Residual standard error: 0.7989 on 1328 degrees of freedom
Multiple R-squared:  0.03222,Adjusted R-squared:  0.03004
F-statistic: 14.74 on 3 and 1328 DF,  p-value: 1.915e-09
```

This indicates that although statistically significant, these explanatory factors explain little of the sample variation in Marlboro GPA. Then, I'd be fairly safe saying that Gender

and whether one has taken standardized tests has little effect on GPA.

Then, although we don't have our ultimate model, we still have some interesting results.

### (3) Using Logistic Regression to Determine What makes Students Leave

In order to use regression analysis to predict a categorical response variable, such as withdraw status, we must use logistic regression. Then, for Marlboro data, we could try to model whether grade point average predicts whether or not one will withdraw.

In R, we can do this with:

```
> model <- glm(Current.Status~Marlboro.GPA, family=binomial)
> summary(model)

 Coefficients:  Estimate Std. Error z value Pr(>|z|)
  (Intercept)    5.3015     0.4420   11.99   <2e-16 ***
  Marlboro.GPA  -1.6418     0.1344  -12.21   <2e-16 ***
      Null deviance: 1843.5  on 1331  degrees of freedom
  Residual deviance: 1588.1  on 1330  degrees of freedom
  AIC: 1592.1
```

For this model, the high residual deviance should scare us, indicating that this model doesn't really do a great job estimating withdraw status, although it is somewhat lower than the null deviance. Then, in order to figure out what predicts withdraw status, we need to hypothesize about better models. Still, this is an interesting result - that Marlboro GPA, on its own, does not particularly predict whether or not a student will leave.

Now, we can try adding in a variable, particularly whether or not a student has requested financial aid. Then, our model is now:

```
> model <- glm(Current.Status~Fin.Aid+Marlb.GPA, family=binomial)
> summary(model)

  Coefficients:          Estimate Std. Error z value Pr(>|z|)
  (Intercept)             5.7494     0.5060   11.36   <2e-16 ***
  Marlboro.GPA           -1.9147     0.1560  -12.28   <2e-16 ***
  Fin..Aid.RequestedYes   3.4854     0.3023   11.53   <2e-16 ***
      Null deviance: 1843.5  on 1331  degrees of freedom
  Residual deviance: 1318.0  on 1329  degrees of freedom
  AIC: 1324
```

Therefore, adding in "financial aid requested" as an explanatory variable reduces residual deviance by quite a bit. Additionally, both of these variables are statistically significant, with a very low probability that they do not pass a $\chi^2$ distribution test. Although we still have to convert these into a number that makes sense, based off the estimates, we see that increases in Marlboro GPA have a decrease the odds of withdraw status, while financial aid requests increase the odds of a withdraw status. However, here, it might be nice to compare how much deviance each specific variable explains. We can check this with:

```
> anova(model)
                      Df Deviance Resid. Df Resid. Dev
NULL                                  1331     1843.5
Marlboro.GPA          1   255.33      1330     1588.1
Fin..Aid.Requested    1   270.09      1329     1318.0
```

Which shows us that Financial Aid Requested explains slightly more variation who drops out or not, but only by a small margin. Therefore, it looks as if the two most important factors in whether a student leaves are their grade point average and if they requested financial aid.

### (4) Predicting the Average Applicant

This portion was completed in the associationrules.R script, starting from line 89.

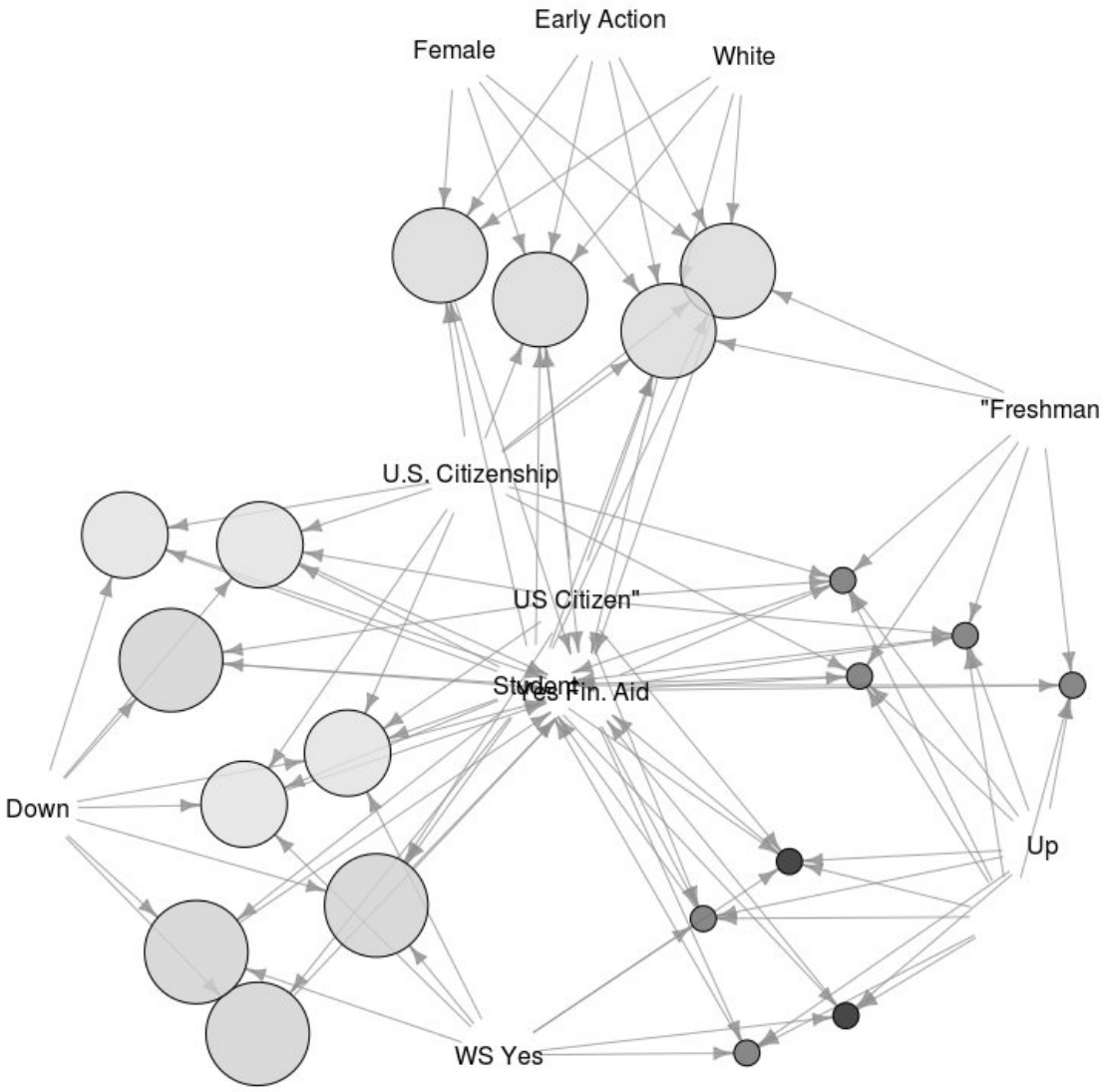
Association rule mining can be used for more than simple rules like which classes frequently lead to others. In this case, it can be also used to determine the most frequent items, or characteristics at Marlboro. This is only one more use of this, but I bet there are some more pretty interesting applications of this here. As such, qualitative categorical Marlboro data was read in a basket format through the code:

```
> All <- read.transactions(file, format="basket", cols=1)
> rules4 = apriori(All, parameter=list(support=0.01, confidence=0.8))
```

And the plotted after being subsetted using the following line:

```
> plot(subrulesall3, method="graph", control=list(type="items"))
```

Which gives us:




Ok. So we have that a frequent Marlboro student is Female, Early Action, White, a Freshman, and a U.S. Citizen. This is sort of a silly result, but it demonstrates the capabilities of association rule mining to produce some interesting findings.

# Recommendations

Ultimately, it appears that while some of my results are interesting, they tell a story that is, at the very least, incomplete. In, my opinion, far more interesting and useful results

about Marlboro students (prospective, current, and past) are out there, and can be revealed through statistical techniques mentioned here. Undoubtedly, there needs to be a increased commitment to storage of institutional data for use in such analyses. The loss of information like H.S. GPA, reader scores, and whatever else that was lost is unfortunate, because it would have potentially resulted in some interesting findings. Bill Mortimer attributed this to the fact that the data I received was the result of day-to-day institutional use, and for data used in day-to-day use, there was little oversight aimed at making sure valuable information was kept for long-term institutional analysis. In other words, the fact that I received observational, and not experimental data likely had adverse effects upon my results. This led, Bill, and certainly me as well, to the conclusion that having an institutional researcher would be great to mitigate data loss, corruption, and to balance use of data for daily purposes and for long term institutional analysis. Also, it would let someone else take up this daunting task.

# References


[1] John Fox. Nonparametric regression: Appendix to an r and s-PLUS companion to applied regression. *Encyclopedia of Statistics in Behavioral Science*, January 2002.

[2] Michael Hahsler, Bettina Grn, Kurt Hornik, and Christian Buchta. Introduction to arulesA computational environment for mining association rules and frequent item sets. *The Comprehensive R Archive Network*, 2009.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, second edition.

[4] Christopher Manning. *Logistic regression (with R)*. Apontamentos de Quantitative and Probabilistic Explanation in Linguistics, The Stanford NLP Group, Stanford University, 2007.

[5] William Mendenhall and Terry Sincich. *A Second Course in Statistics: Regression Analysis*. Pearson, seventh edition.

[6] Ansaf Salleb-Aouissi. Mining frequent patterns and association rules, October 2008.