Final Project

David Pahmer

2022-06-3

Introduction

For my project, I studied correlations that may exist between outcomes of computer science programs and predictors of student success, from among several variables. It is reasonable to expect that one could predict student success from the outset by building a profile of the student and determining if that profile points to success in the computer science program or not. This would certainly be valuable for deans, advisors, and students themselves- not necessarily as informing school policy, but in guiding the advisors in directing the students along pathways that hold greater promise of success.
Although it would have been very delightful to find a subtle combination of factors that predict success with great accuracy but that elude the scrutinizer due to their complexity; however, in this case the results of the study revealed at best a mildly suggestive model, which used primarily factors that are eminently expected.

Approach

For this project, I gathered student data whose predictor variables are:

- HS grades
- standardized test scores, including SAT, ACT - the students' first semester GPA
- whether the student got credit for AP: math or computers

To identify the population, I selected students whose first term on campus was between fall 2010 and fall 2018, who declared their major to be computer science at any time, or any of those students who took their first computer course within two years of arriving on campus.
This population was split into two groups: students beginning from 2010-2015 for the training dataset, and students beginning from 2015-2018 for the testing set. (Come to think of it, I didn't need to do it that way, but too late now. . .)

For the outcome, dependent variable I created a "success" variable that is defined by completion of the computer science major with major GPA of at least 3.0. Although I intended to split the HS grades into math vs. all else, I discovered that the HS transcript is not kept in the database. So I was only be able to use the aggregate HS gpa.

The thinking here is that students who succeed in the computer science program demonstrate prowess in math, as indicated on the SAT or HS grades or AP exams, or generally do well in their academic studies, as indicated by their first-term GPA, which is sometimes used as a predictor of success in other contexts. Typically, students do not have exposure to programming or other fundamentals of computer science prior to admission to college, so there is no track record of their CS aptitude, so we are forced to look at other, more indirect, predictors.

Analysis

The major tasks associated with this project were filtering the datasets to conform to the specifications of the project, such as identifying students who received credit for the math or computer AP, and coding it as T/F; identifying students' first term on campus and filtering out the ones we wanted; determining which term was the first one they took a computer class in, and whether it was early enough in their program that they might pursue the major; cleaning up the HS transcript data. Some of this work was done before bringing the data into R, and some is done below- especially ensuring unique student identifier keys.

Let's look at the coding:

```
setwd ("C:/Users/pahme/OneDrive/Documents")
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caTools)
library(useful)
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)
library(class)

lgpa <- read_excel("lgpa.xlsx")
grad.bach <- read_excel(("grad_bach.xlsx"))
hs_gpa <- read_excel("hs gpa.xlsx")
tgpa <- read_excel("tgpa.xlsx")
ftoc <- read_excel("ftoc.xlsx")
sat <- read_excel("sat act.xlsx")
ap <- read_excel("ap exam.xlsx")
firstcom <- read_excel("first com course.xlsx")
major <- read_excel("declared com major.xlsx")

# unique student identifer is "pidm"

grad.bach$inc <- (grad.bach$majr_code_1=="COM" | grad.bach$majr_code_1_2=="COM")
str (firstcom)
```

```
## tibble [1,305 x 3] (S3: tbl_df/tbl/data.frame)
##  $ pidm     : num [1:1305] 6454 180858 180877 180905 180915 ...
##  $ term_code: num [1:1305] 201109 201009 201001 201206 201001 ...
##  $ title    : chr [1:1305] "Theory of Computation" "Intro to Computer Science" "Intro To Operating Sy
```

```
firstcom <- firstcom[,1:2]
# because we don't need the title.

str(ftoc)
```

```
## tibble [8,781 x 3] (S3: tbl_df/tbl/data.frame)
##  $ pidm          : num [1:8781] 387 1374 31576 88179 89684 ...
##  $ term_code_ftoc: num [1:8781] 201701 201209 201501 201909 202009 ...
##  $ camp_code     : num [1:8781] 1 1 1 1 1 2 1 1 1 1 ...
```

```
profile <- subset(ftoc,term_code_ftoc > 201009 & term_code_ftoc <201809,
                  select=c(pidm, term_code_ftoc))

# make sure pidm and term_code are unique!
tgpa <- tgpa[!duplicated(tgpa[,c("pidm","term_code")]),]
names(tgpa)[names(tgpa)=='term_code'] <- 'term_gpa'


# same for AP
ap <- ap[!duplicated(ap$pidm),]

#same for the lgpa
lgpa.u <- lgpa[do.call(order,lgpa),c("pidm","levl_code","lgpa")]
lgpa.u <- lgpa.u[!duplicated(lgpa.u$pidm),c("pidm","lgpa")]
```

After checking that all the other tables are unique students, we can join them all

```
profile <- profile %>% left_join(major, by='pidm')
names(profile)[names(profile)=='term_code'] <- 'term_major'

names(firstcom)[names(firstcom)=='term_code'] <- 'term_com'

profile <- profile %>% left_join(firstcom, by='pidm')

profile <- profile %>% left_join(ap, by='pidm')
# convert this to T/F
profile$subj_code <- as.logical(!is.na(profile$subj_code))

profile <- profile %>% left_join(grad.bach, by='pidm')

profile <- profile %>% left_join(hs_gpa, by='pidm')

# It looks like gpa is a char; change to number
profile$gpa <- as.numeric(profile$gpa)
```

```
## Warning: NAs introduced by coercion
```

```
profile <- profile %>% left_join(lgpa.u, by='pidm')
profile <- profile %>% left_join(sat, by='pidm')

profile <- profile %>% left_join(tgpa, by='pidm')
# Only keep the records that the term gpa is same as the ftoc term
profile <- profile[profile$term_gpa==profile$term_code_ftoc,]
```

Now let's filter down to the eligible population:

```
# first com course within two years of starting
interval <- profile$term_com - profile$term_code_ftoc

eligible <- interval<=200

# Declared COM major
eligible2 <- eligible==TRUE | !is.na(profile$term_major)

profile <- profile[which(eligible2==TRUE),]
```

define success as graduated in COM with GPA 3.0 or better

```
success <- profile$inc==TRUE & profile$lgpa >=3
success[is.na(success)] <- FALSE

profile$success <- success==TRUE


profile.train <- profile[which(profile$term_code_ftoc<= 201509),]
profile.test <- profile[which(profile$term_code_ftoc > 201509),]
```

Now the work: analysis

```
# First model- logistic regression with many variables:
model.1 <- glm(success ~ subj_code + gpa + tgpa + sat_math + act_math +
                 sat_verbal , data = profile.train,  family = binomial())
summary(model.1)
```

```
##
## Call:
## glm(formula = success ~ subj_code + gpa + tgpa + sat_math + act_math +
##     sat_verbal, family = binomial(), data = profile.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.49285  -0.47286  -0.31129  -0.06359   1.95323
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.873e+00  2.285e+01  -0.082    0.935
## subj_codeTRUE  1.606e+00  1.600e+00   1.004    0.315
## gpa           -1.778e-01  2.289e-01  -0.777    0.437
## tgpa           6.164e+00  4.829e+00   1.276    0.202
## sat_math       8.859e-05  1.539e-02   0.006    0.995
## act_math       6.936e-02  2.467e-01   0.281    0.779
## sat_verbal    -1.444e-02  1.196e-02  -1.207    0.227
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 27.034  on 29  degrees of freedom
## Residual deviance: 19.960  on 23  degrees of freedom
##   (361 observations deleted due to missingness)
## AIC: 33.96
##
## Number of Fisher Scoring iterations: 7
```

```
# ok, nothing significant!

# Second model- logistic regression without the ACT, because too many missing values:
model.2 <- glm(success ~ subj_code + gpa + tgpa + sat_math +
                 sat_verbal , data = profile.train,    family = binomial())
summary(model.2)
```

```
##
## Call:
## glm(formula = success ~ subj_code + gpa + tgpa + sat_math + sat_verbal,
##     family = binomial(), data = profile.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1654  -0.6373  -0.4453  -0.3037   2.7475
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -7.842229   4.139093  -1.895  0.05814 .
## subj_codeTRUE  0.031950   0.400821   0.080  0.93647
## gpa           -0.021410   0.051522  -0.416  0.67774
## tgpa           0.747985   0.696130   1.074  0.28260
## sat_math       0.010650   0.003370   3.160  0.00158 **
## sat_verbal    -0.003314   0.002407  -1.377  0.16854
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 248.11  on 304  degrees of freedom
## Residual deviance: 228.51  on 299  degrees of freedom
##   (86 observations deleted due to missingness)
## AIC: 240.51
##
## Number of Fisher Scoring iterations: 5
```

```
# maybe something- if we focus on the SAT math score

model.3 <- glm(success ~ sat_math, data=profile.train, family=binomial())
summary(model.3)
```

```
##
## Call:
## glm(formula = success ~ sat_math, family = binomial(), data = profile.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8294  -0.6319  -0.4742  -0.3376   2.5083
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.967170   1.800431  -4.425 9.64e-06 ***
## sat_math     0.008846   0.002511   3.523 0.000427 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 252.94  on 308  degrees of freedom
## Residual deviance: 238.35  on 307  degrees of freedom
##   (82 observations deleted due to missingness)
## AIC: 242.35
##
## Number of Fisher Scoring iterations: 5
```

```
pred.m3 <- predict(model.3, profile.test, type="response")

(checkmodel3 <- table(actual=profile.test$success, prediction= pred.m3 > .1))
```

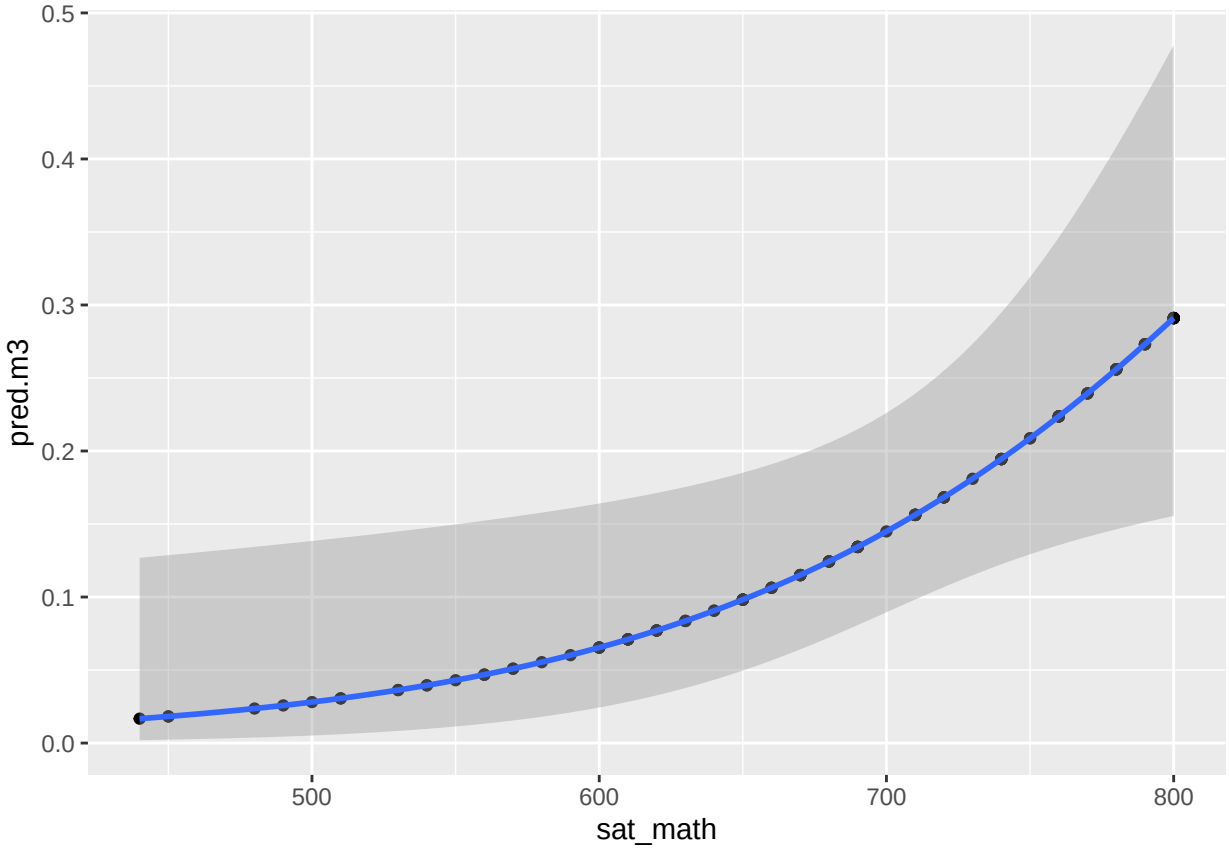
```
##        prediction
## actual  FALSE TRUE
##   FALSE    36   42
##   TRUE      8   33
```

```
# if we set the cutoff at .5 like one would expect- we get the prediction that all fail.

# Accuracy for the training set, sort of
((checkmodel3[1,1]+checkmodel3[2,2])/length(profile.test[,1]))
```

```
## [1] 69
```

```
# Let's plot this logistic regression curve:

ggplot(profile.test, aes(x=sat_math, y=pred.m3)) + geom_point() +
  stat_smooth(method="glm",  method.args=list(family=binomial()))
```

```
## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 72 rows containing non-finite values (stat_smooth).
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning: Removed 72 rows containing missing values (geom_point).
```




```
# We see that everyone has less than 50% expectation of success, for some reason.


# let's try adding also AP credit to this:

model.4 <- glm(success ~ sat_math + subj_code, data=profile.train, family=binomial())
summary(model.4)
```

```
##
## Call:
## glm(formula = success ~ sat_math + subj_code, family = binomial(),
##     data = profile.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8313  -0.6299  -0.4734  -0.3377   2.5074
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -7.935534   1.949671  -4.070  4.7e-05 ***
## sat_math       0.008793   0.002808   3.132  0.00174 **
## subj_codeTRUE  0.016144   0.382416   0.042  0.96633
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 252.94  on 308  degrees of freedom
## Residual deviance: 238.34  on 306  degrees of freedom
##   (82 observations deleted due to missingness)
## AIC: 244.34
##
## Number of Fisher Scoring iterations: 5
```

```
pred.m4 <- predict(model.4, profile.test, type="response")

(checkmodel4 <- table(actual=profile.test$success, prediction= pred.m4 > .1))
```

```
##        prediction
## actual  FALSE TRUE
##   FALSE    36   42
##   TRUE      8   33
```

```
# Accuracy for the training set, sort of
((checkmodel4[1,1]+checkmodel4[2,2])/length(profile.test[,1]))
```

```
## [1] 69
```

```
# So it didn't add anything.


model.5 <- glm(success ~ sat_math + tgpa, data=profile.train, family=binomial())
summary(model.5)
```

```
##
## Call:
## glm(formula = success ~ sat_math + tgpa, family = binomial(),
##     data = profile.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8496  -0.6300  -0.4607  -0.3204   2.6584
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.379320   2.362045  -3.971 7.16e-05 ***
## sat_math     0.007587   0.002748   2.761  0.00576 **
## tgpa         0.619028   0.611384   1.013  0.31130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 252.94  on 308  degrees of freedom
## Residual deviance: 237.19  on 306  degrees of freedom
##   (82 observations deleted due to missingness)
## AIC: 243.19
##
## Number of Fisher Scoring iterations: 5
```

```
pred.m5 <- predict(model.5, profile.test, type="response")

(checkmodel5 <- table(actual=profile.test$success, prediction= pred.m5 > .1))
```

```
##        prediction
## actual  FALSE TRUE
##   FALSE    36   42
##   TRUE      7   34
```

```
# Accuracy for the training set, sort of
((checkmodel5[1,1]+checkmodel5[2,2])/length(profile.test[,1]))
```

```
## [1] 70
```

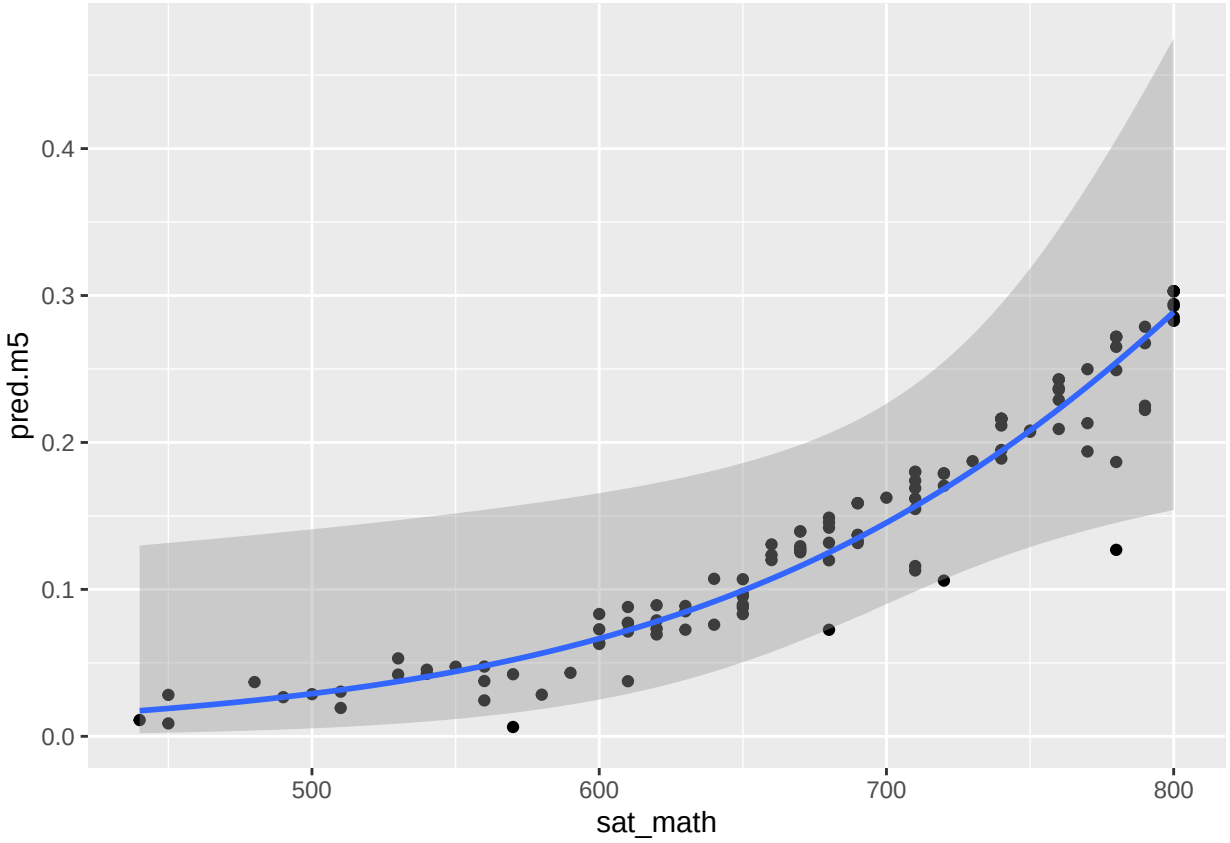
```
# Let's plot this logistic regression curve:

ggplot(profile.test, aes(x=sat_math, y=pred.m5)) + geom_point() +
  stat_smooth(method="glm",  method.args=list(family=binomial()))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 72 rows containing non-finite values (stat_smooth).
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning: Removed 72 rows containing missing values (geom_point).
```




```
# Let's try it with an interaction:

model.5 <- glm(success ~ sat_math * tgpa, data=profile.train, family=binomial())
summary(model.5)
```

```
##
## Call:
## glm(formula = success ~ sat_math * tgpa, family = binomial(),
##     data = profile.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8288  -0.6428  -0.4887  -0.2873   3.0176
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -31.103569  25.667630  -1.212    0.226
## sat_math        0.039675   0.037347   1.062    0.288
## tgpa            6.438077   6.842817   0.941    0.347
## sat_math:tgpa  -0.008572   0.009919  -0.864    0.387
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 252.94  on 308  degrees of freedom
## Residual deviance: 236.34  on 305  degrees of freedom
##   (82 observations deleted due to missingness)
## AIC: 244.34
##
## Number of Fisher Scoring iterations: 6
```

```
pred.m5 <- predict(model.5, profile.test, type="response")

(checkmodel5 <- table(actual=profile.test$success, prediction= pred.m5 > .1))
```

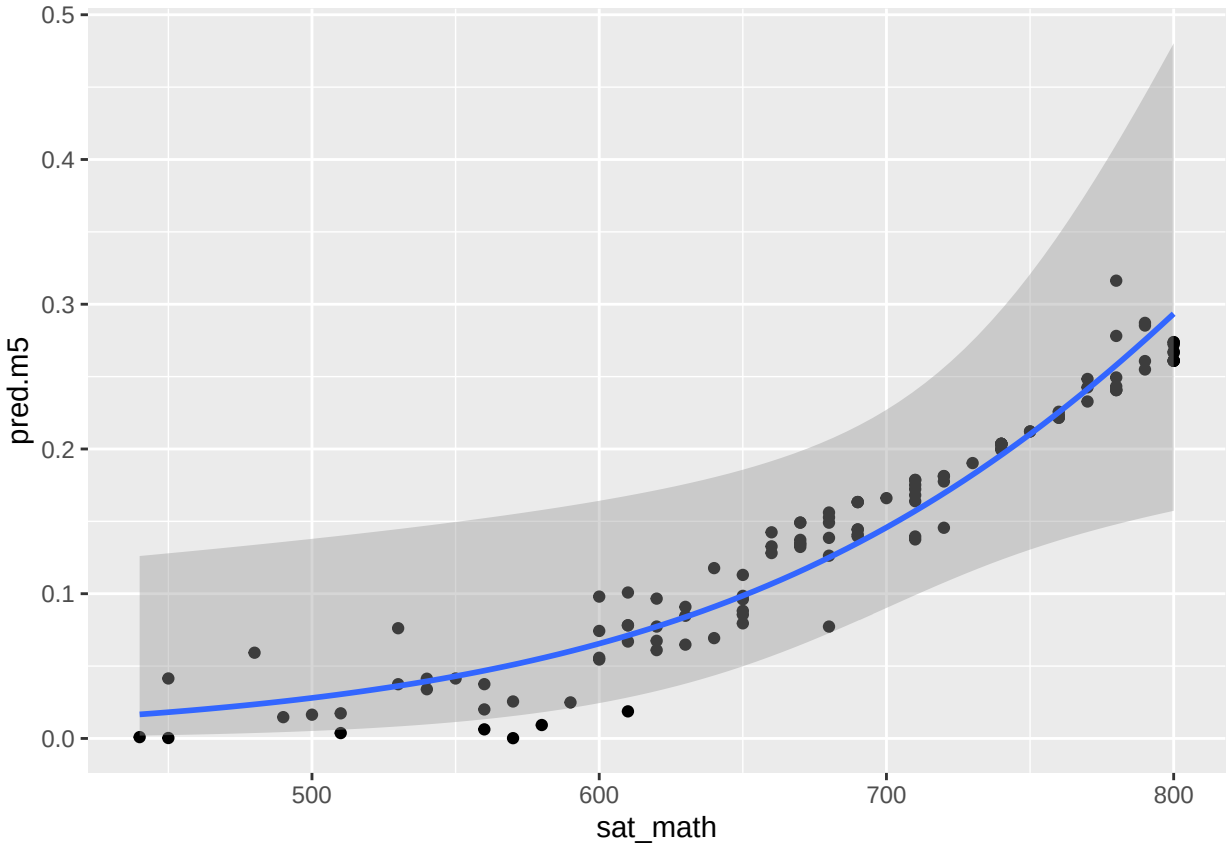
```
##        prediction
## actual  FALSE TRUE
##   FALSE    35   43
##   TRUE      7   34
```

```
# Accuracy for the training set, sort of
((checkmodel5[1,1]+checkmodel5[2,2])/length(profile.test[,1]))
```

```
## [1] 69
```

```
# Let's plot this logistic regression curve:

# For the SAT cutoff:
ggplot(profile.test, aes(x=sat_math, y=pred.m5)) + geom_point() +
  stat_smooth(method="glm",  method.args=list(family=binomial()))
```

```
## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 72 rows containing non-finite values (stat_smooth).
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning: Removed 72 rows containing missing values (geom_point).
```



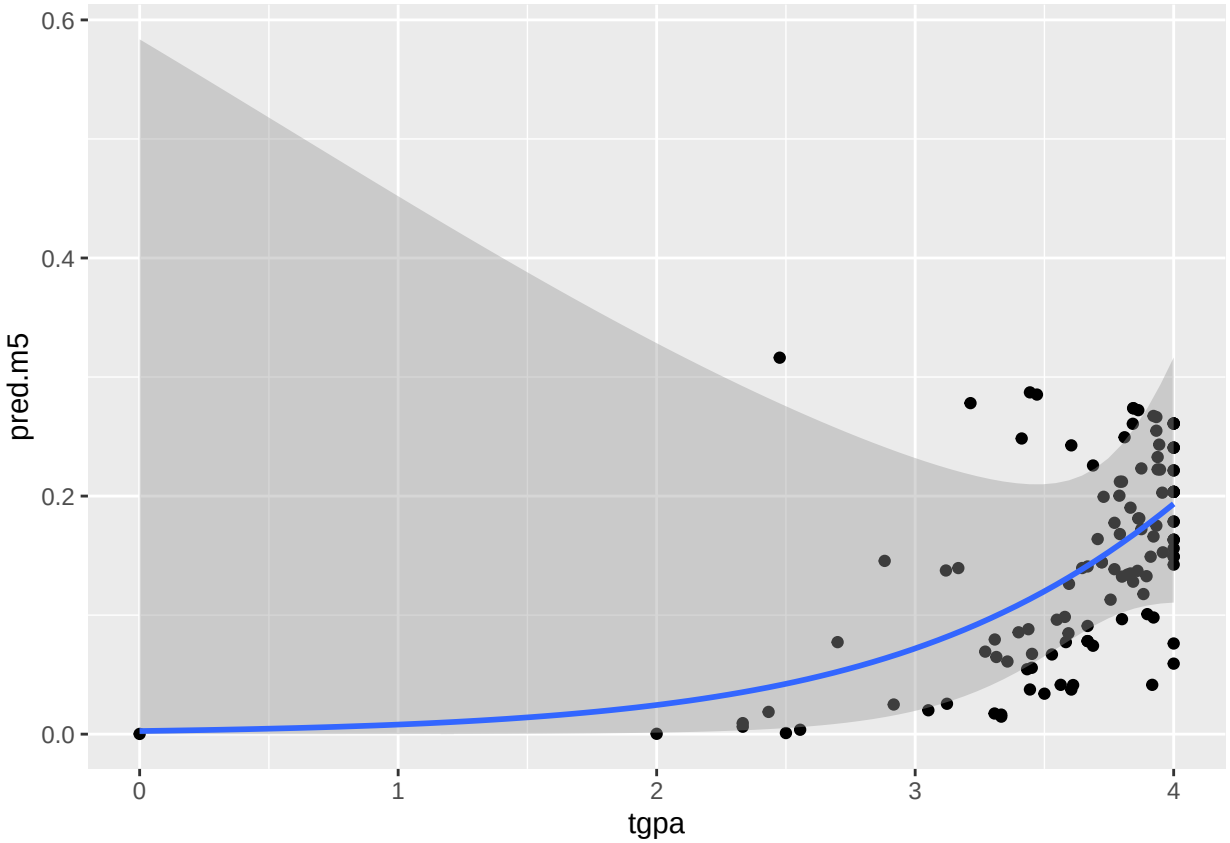

```
# For the first-term GPA cutoff:
ggplot(profile.test, aes(x=tgpa, y=pred.m5)) + geom_point() +
  stat_smooth(method="glm",  method.args=list(family=binomial()))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 72 rows containing non-finite values (stat_smooth).
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning: Removed 72 rows containing missing values (geom_point).
```



So- so far we are seeing something promising with math SAT scores above 650 and possibly also the first term GPA above 3.3
Still, let's do some miscellaneous tests, just in case we missed something

```
with (profile.train, table(subj_code,success))
```

```
##          success
## subj_code FALSE TRUE
##     FALSE   252   31
##     TRUE     90   18
```

```
with (profile.train, mean(subj_code==success))
```

```
## [1] 0.6905371
```

```
# Interesting- about 69% success, but not enough maybe to be useful.

with (profile.train, table(round(gpa),success))
```

```
##      success
##       FALSE TRUE
##   67      0    1
##   76      1    0
##   77      1    0
```

```
##   79      1    0
##   80      2    0
##   81      2    0
##   82      5    1
##   83      6    1
##   84      5    1
##   85      5    1
##   86     14    1
##   87     10    0
##   88     15    0
##   89     14    2
##   90     22    3
##   91     25    3
##   92     24    5
##   93     31    4
##   94     28    3
##   95     25    3
##   96     27    9
##   97     29    3
##   98     18    3
##   99     10    4
##   100     6    0
##   101     1    0
##   102     1    0
##   103     1    0
```

```
# Not clearly useful

with (profile.train, table(act_math,success))
```

```
##         success
## act_math FALSE TRUE
##       16     2    0
##       20     2    0
##       21     3    0
##       23     3    1
##       24     7    0
##       25     5    0
##       26     9    0
##       27     6    3
##       28     6    1
##       29     6    0
##       30     4    0
##       31     6    0
##       32    11    0
##       33     5    0
##       34     8    1
##       35     3    2
##       36     2    0
```

```
# so not enough people took ACT


# Let's see if the combination of sat_math and first term gpa can work better
model.4 <- glm(success ~ sat_math + round(tgpa,1), data=profile.train,
```

```
              family=binomial())
summary(model.4)
```

```
##
## Call:
## glm(formula = success ~ sat_math + round(tgpa, 1), family = binomial(),
##     data = profile.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8499  -0.6346  -0.4598  -0.3214   2.6495
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -9.404355   2.367439  -3.972 7.12e-05 ***
## sat_math        0.007564   0.002748   2.752  0.00592 **
## round(tgpa, 1)  0.630170   0.614305   1.026  0.30497
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 252.94  on 308  degrees of freedom
## Residual deviance: 237.16  on 306  degrees of freedom
##   (82 observations deleted due to missingness)
## AIC: 243.16
##
## Number of Fisher Scoring iterations: 5
```

```
# So that wasn't any better ostensibly than the sat math alone.
```

This doesn't necessarily give us much, but maybe a different approach will be better.

```
# seems like we need to eliminate the NA records.
p2.train <- profile.train[!is.na(profile.train$sat_math) & !is.na(profile.train$tgpa),
                          c("sat_math","tgpa","success")]

p2.test <- profile.test[!is.na(profile.test$sat_math) & !is.na(profile.test$tgpa),
                        c("sat_math","tgpa","success")]


p.pred <- knn(p2.train[1:2]  ,p2.test[1:2]  , k=1, cl=p2.train$success)
(acctable <- table(p.pred, p2.test$success))
```

Let's try a cluster analysis

```
##
## p.pred  FALSE TRUE
##   FALSE    66   36
##   TRUE     12    5
```

```
mean(p.pred==p2.test$success)
```

```
## [1] 0.5966387
```

```r
# looks like about 60% accuracy so far. Let's try larger k
p.pred <- knn(p2.train[1:2]  ,p2.test[1:2]  , k=4, cl=p2.train$success)
(acctable <- table(p.pred, p2.test$success))
```

```
##
## p.pred  FALSE TRUE
##   FALSE    75   41
##   TRUE      3    0
```

```r
mean(p.pred==p2.test$success)
```

```
## [1] 0.6302521
```

Although this looks like an improvement, it's not, since we predicted only 5 people would be successful, and of those only one was; while of all the others whom we predicted would not be successful- about a third were. So at this point our best model revealed that students who achieve 650 or above on their math SAT, and also possibly achieve 3.3 or above on their first term on campus are most likely to succeed in their computer science program but we wouldn't be strongly influenced by that.

Limitations

This study did reveal some interesting non-correlations, such as HS gpa did not matter much, and it didn't seem to matter whether the student took the math or computer AP.
Several areas of weakness in this project include: perhaps there ought to be a better or different measure of success, whether utilizing the data available or not; perhaps the specific math gpa on their HS transcript would be a better indicator than the overall HS gpa, but that wasn't available for analysis; perhaps the population ought to have been reduced further to students who indicated an interest in the major and ignore the students who took a computer course early, since perhaps that throws off the analysis; however, students don't necessarily declare a major or revise their earlier declaration until they get ready to graduate, so we were somewhat compelled to infer their intentions indirectly.

Concluding Remarks

This project seemed to have great promise for yielding valuable information on a matter that might have been of some interest to school professionals, but using my parameters, it was a bit of a letdown. Although it might still have some benefit, probably success in this field may come down to an entirely different set of characteristics, such as grit, work ethic, school support systems, etc.
The programming aspects of this project did compel me to go over many of the earlier work we did to pull off the analyses, data manipulations, plotting, and so on. Additionally, some of the data wrangling tasks were entirely new to me, which was great, while difficult. As all others have pointed out, I can confirm that at least for this project the time involved in prepping the data was much, much greater than the time involved in analyzing it.
Perhaps there is still room for reanalyzing the data, with better input from students and experienced advisors and deans, from surveys or other methods.

Thank you, Professor Williams, for all of your dedication to your students and the program- it is entirely manifest and I can speak for the whole class insaying that we do appreciate it tremendously!