DS250 - Final Project

Amy Werner-Allen

November 15, 2015

A recently completed study of colleges in the US collected information regarding school and student demographics, degrees awarded by subject area as well as financial information including average degree cost by income bracket, student tuition and scholarships, repayment rates, and post-college average earnings.

As the Data Scientist on this project, you have broad discretion in the selection of source data, correlation of meaningful attributes, predictive model selection, and presentation of findings from this analysis.

Retrieve the Data

Here we pull in the most recent data that we have, from 2014-2015. (Note: I chose to evaluate only the most recent year for two reasons: 1) processing time and 2) in hopes that the most recent data will be best representative of the colleges. Looking back five years would probably be the best method).

```r
# setwd("~/Miscellaneous/UW_Final")
library(plyr)
library(ggplot2)
library(reshape2)
library(gdata)

# Load in all of the data
data = read.csv("MERGED2014_15_PP.csv", header=TRUE)
```

Exploratory Data Analysis

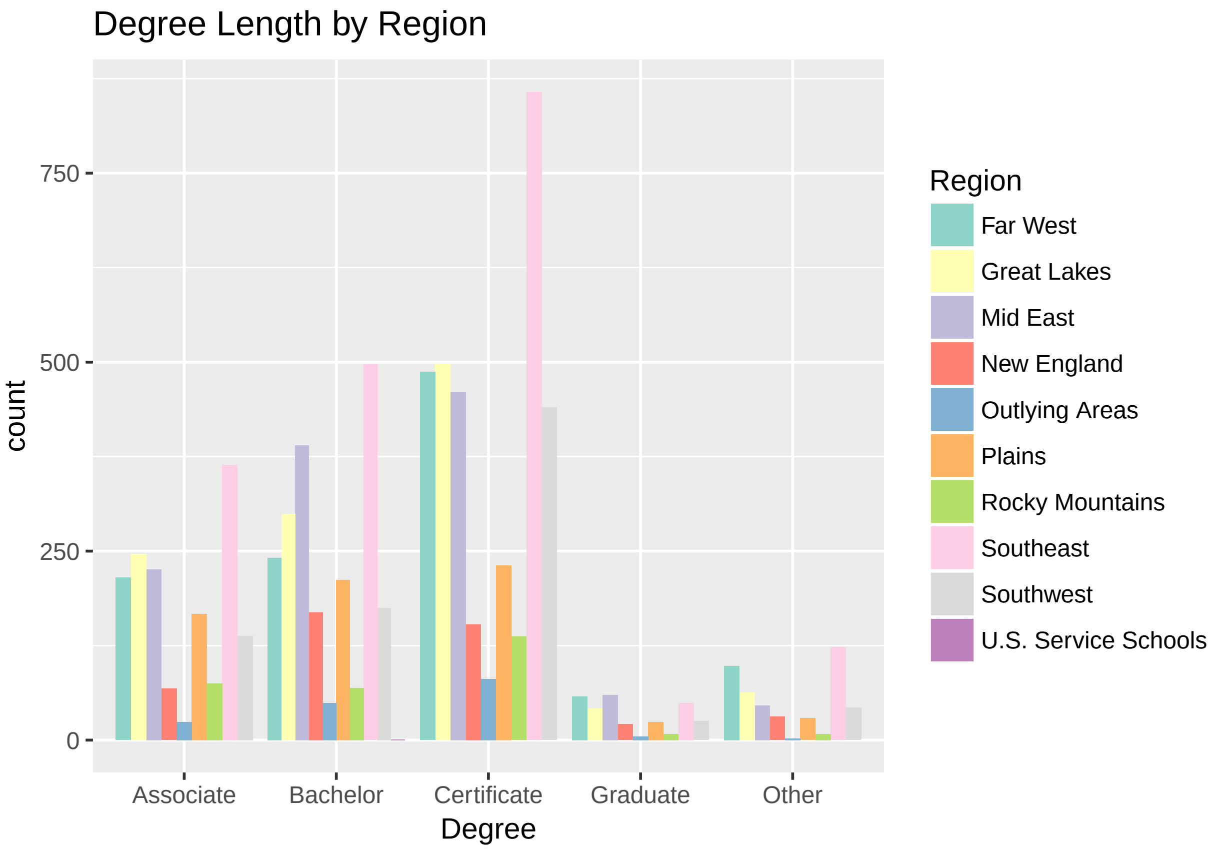
Count of colleges by degree length and region: for this analysis, we need to hone in on region and degree type. Using the data dictionary, we can map the region and degree IDs to more helpful concepts. Although degree length isn't defined in the data set, we can use degree type as an estimate: certificate (1 year), associate (2 years), bachelor (4 years), graduate (3-5 years).

```r
# Create a subset with relevant columns
data1 = data[,c(4,5,6,7,15,16,19)]

data1$RegionDesc[data1$REGION==0] = "U.S. Service Schools"
data1$RegionDesc[data1$REGION==1] = "New England"
data1$RegionDesc[data1$REGION==2] = "Mid East"
data1$RegionDesc[data1$REGION==3] = "Great Lakes"
data1$RegionDesc[data1$REGION==4] = "Plains"
data1$RegionDesc[data1$REGION==5] = "Southeast"
data1$RegionDesc[data1$REGION==6] = "Southwest"
data1$RegionDesc[data1$REGION==7] = "Rocky Mountains"
data1$RegionDesc[data1$REGION==8] = "Far West"
data1$RegionDesc[data1$REGION==9] = "Outlying Areas"

data1$Degree[data1$PREDDEG==0] = "Other"
data1$Degree[data1$PREDDEG==1] = "Certificate"
data1$Degree[data1$PREDDEG==2] = "Associate"
data1$Degree[data1$PREDDEG==3] = "Bachelor"
```

```
data1$Degree[data1$PREDDEG==4] = "Graduate"

data1$RegionDesc = as.factor(data1$RegionDesc)
data1$Degree = as.factor(data1$Degree)

d1 = count(data1, c('RegionDesc', 'Degree'))
colnames(d1) = c("Region", "Degree", "Count")
```

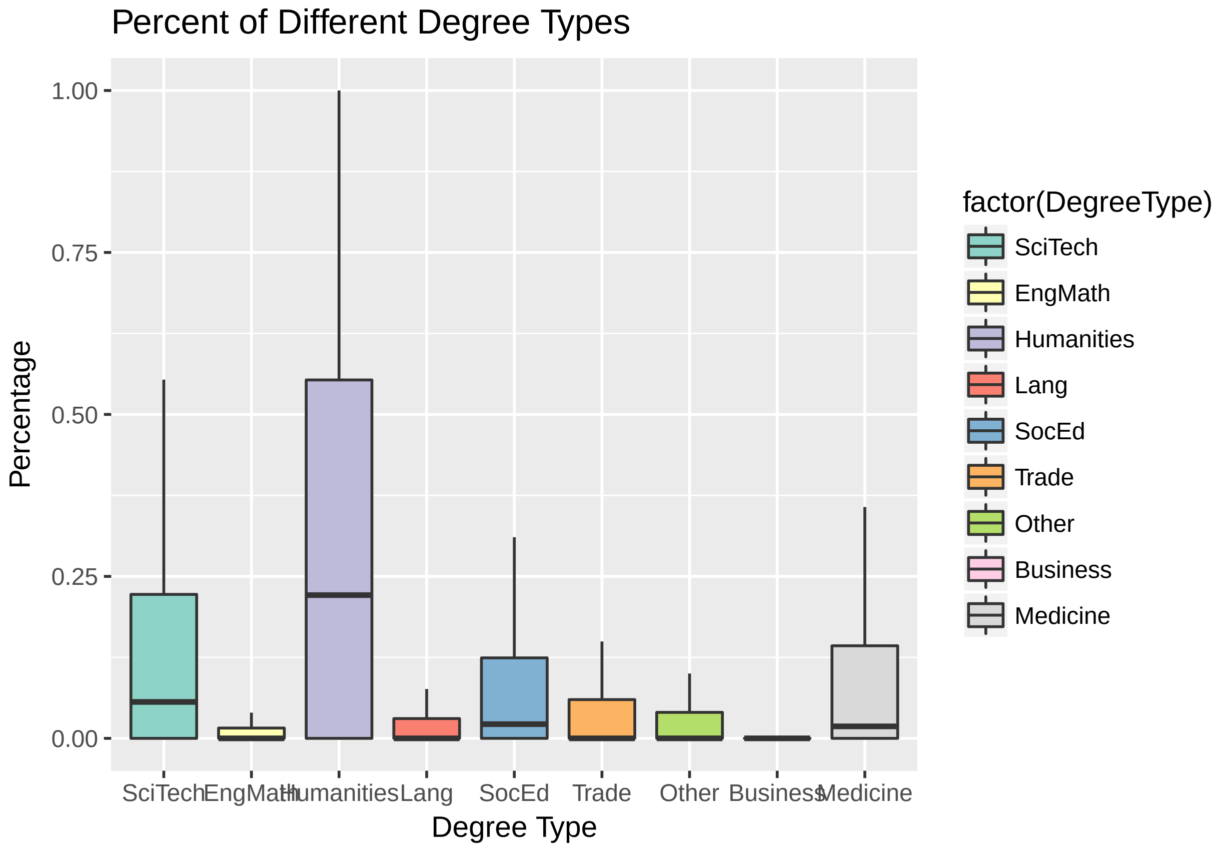
Now we can plot the data by degree and region:

```
p = ggplot(data = d1, aes(x=Degree))+labs(x="Degree")+ggtitle("Degree Length by Region")
p = p + geom_bar(position="dodge", aes(weights=Count, fill=Region))
p = p + scale_fill_brewer(palette="Set3")
p
```



Categorize student graduation counts by degree name: to tackle this question, I decided instead to look at the percent of the graduating class that majored in a particular degree. This way we could evaulate which degree types are most common. By using percent instead of counts, this normalizes across institution size. Because there are many degrees listed, I decided to bucket the degree types into larger categories: Science and Technology, Engineering and Mathematics, Humanities, Language, Social and Education Studies, Trade Studies, Business, Medicine, and all others. Then I visualized the data using a boxplot to show the variability and average in each group.

```
data2 = data[,c(seq(from=63, to=99, by=1),387,291)]
data2$C150_4 = suppressWarnings(as.numeric(as.character(data2$C150_4)))
data2$UGDS = suppressWarnings(as.numeric(as.character(data2$UGDS)))
data2$TotalGrad = floor(data2$C150_4*data2$UGDS/4)

colnames(data2)[1:37] =
c('Agricultures','Natural Resources','Architecture','Area, Ethnic, Cultural, Gender, And Group Studies'

### Convert all data to numeric
cols = seq(from=1, to=37, by=1)
data2[,cols] = suppressWarnings(apply(data2[,cols], 2, function(x) as.numeric(as.character(x))))

### Create sub-categories
# Science and Tech
cols = c(1,2,7,18,20,25,26)
data2$SciTech = apply(data2[,cols], 1, function(x) sum(x))

# Engineering and Math
cols = c(3,10,11,19)
data2$EngMath = apply(data2[,cols], 1, function(x) sum(x))

# Humanities
cols = c(16,17,23,24,27,35)
data2$Humanities = apply(data2[,cols], 1, function(x) sum(x))

# Languages
cols = c(12,15)
data2$Lang = apply(data2[,cols], 1, function(x) sum(x))

# Social and Education Studies
cols = c(4,5,6,9,13,29,30)
data2$SocEd = apply(data2[,cols], 1, function(x) sum(x))

# Trade Studies
cols = c(31:34)
data2$Trade = apply(data2[,cols], 1, function(x) sum(x))

# Other
cols = c(8,21,22,28)
data2$Other = apply(data2[,cols], 1, function(x) sum(x))

data2s = data2[,c(41:47,37,36)]
data2s = melt(data2s)
colnames(data2s) = c("DegreeType", "Percentage")

p = ggplot(data2s, aes(factor(DegreeType), Percentage))+labs(x="Degree Type")+ggtitle("Percent of Diffe
p = p + scale_fill_brewer(palette="Set3")
p = p + geom_boxplot(aes(fill = factor(DegreeType)), outlier.shape = NA)
p
```



Study Evaluation

- Study type and approach
- Bias analysis: populations measured, access to scholarships, etc.
- Hypothesis analysis

The **college scorecard** data is collected from educational institutions across the United States to allow access and insight into higher education costs and outputs.

There are inherent biases in this data. As stated in the documentation:

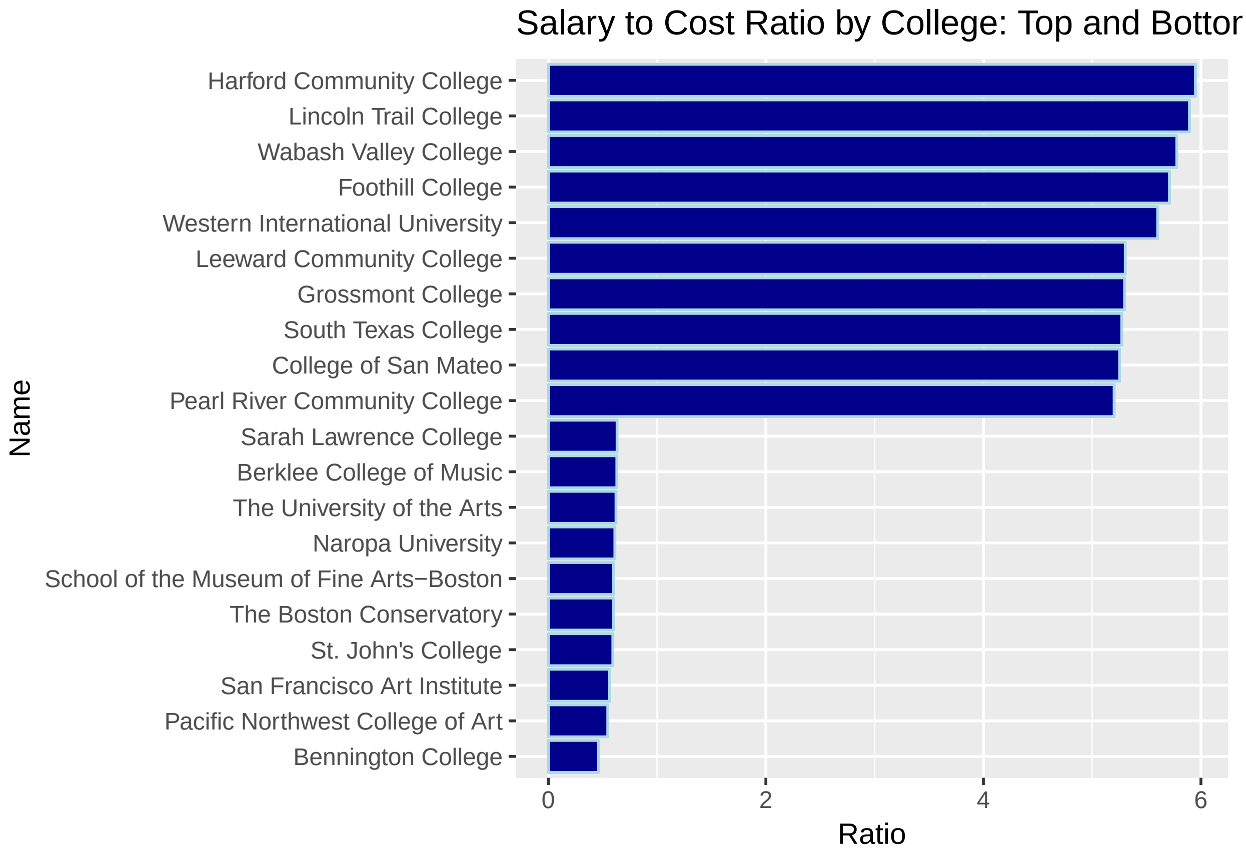
Many elements are available only for Title IV recipients, or students who receive federal grants and loans. These data are reported at the individual level to the National Student Loan Data System (NSLDS), which is used to distribute federal aid, and published at the aggregate institutional level.

The implication here is that data is collected or available only for students on federal grants or loans, which does restrict full collection of data. The subcollection of data also may not be fully representative of the full student population, and my skew results. For example, while class is in no way indicative of a student's natural academic ability, students from poorer backgrounds may not have access to academically rigorous school systems; they may be unprepared for post-secondary education, and may have higher rates of dropping out. This sort of bias may skew the completion rate, for example.

Summary Analysis - Value-add

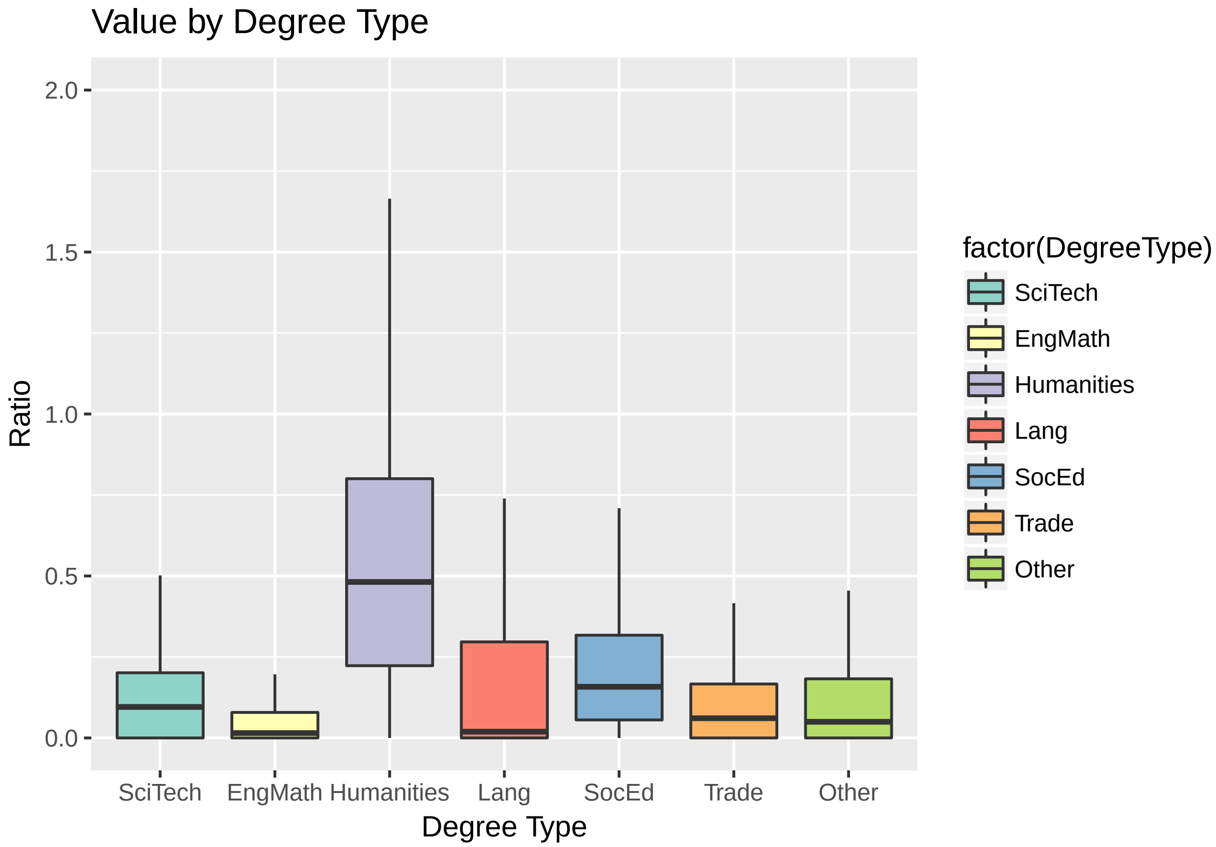
Identify highest "value" in terms of (cost - salary) per college: here we pull out the salary and costs per college. We create a ratio - of salaray to cost - in order to effectively rank which colleges give the most bang for their buck. The higher the ratio, the better the value.

```r
# Compile necessary data elements
data3 = read.xls("earnings.xlsx", sep=",", header=TRUE)
data3$MN_EARN_WNE_P10 = as.numeric(as.character(data3$MN_EARN_WNE_P10))
data3$MN_EARN_WNE_MALE0_P10 = as.numeric(as.character(data3$MN_EARN_WNE_MALE0_P10))
data3$MN_EARN_WNE_MALE1_P10 = as.numeric(as.character(data3$MN_EARN_WNE_MALE1_P10))

# Make new dataframe with relevant data
data4 = suppressWarnings(as.data.frame(cbind(as.character(data$INSTNM), as.numeric(as.character(data$CO
colnames(data4) = c("Name", "Cost", "MeanSalary", "SalaryF", "SalaryM")

# Ratio factor
data4$Ratio = suppressWarnings(as.numeric(as.character(data4$MeanSalary))/as.numeric(as.character(data4
data4 = data4[order(-data4$Ratio),]
data4 = na.omit(data4)

top10 = data4[1:10,]
bottom10 = data4[3257:3266,]
value = rbind(top10, bottom10)
value$Name = factor(value$Name, levels = value$Name[order(value$Ratio)])

ggplot(value, aes(x=Name, y=Ratio)) + geom_bar(stat='identity', fill="dark blue", color="light blue") +
```



Identify highest "value" in terms of (cost - salary) per degree: since we do not have any direct information about salary related to degree, we will need to extrapolate that information. We will take the degree type buckets from an earlier portion of this exploration and apply that percentage to the salary outputs by school.

```
# Multiply each degree type by the ratio
data5$SciTech = data5$SciTech*data5$Ratio
data5$EngMath = data5$EngMath*data5$Ratio
data5$Humanities = data5$Humanities*data5$Ratio
data5$Lang = data5$Lang*data5$Ratio
data5$SocEd = data5$SocEd*data5$Ratio
data5$Trade = data5$Trade*data5$Ratio
data5$Other = data5$Other*data5$Ratio

data5s = data5[,1:7]
data5s = melt(data5s)
colnames(data5s) = c("DegreeType", "Ratio")
data5s = na.omit(data5s)

p = ggplot(data5s, aes(factor(DegreeType), Ratio))+labs(x="Degree Type")+ggtitle("Value by Degree Type"
p = p + scale_fill_brewer(palette="Set3")
p = p + geom_boxplot(aes(fill = factor(DegreeType)), outlier.shape = NA)
p = p + ylim(0, 2)
p
```

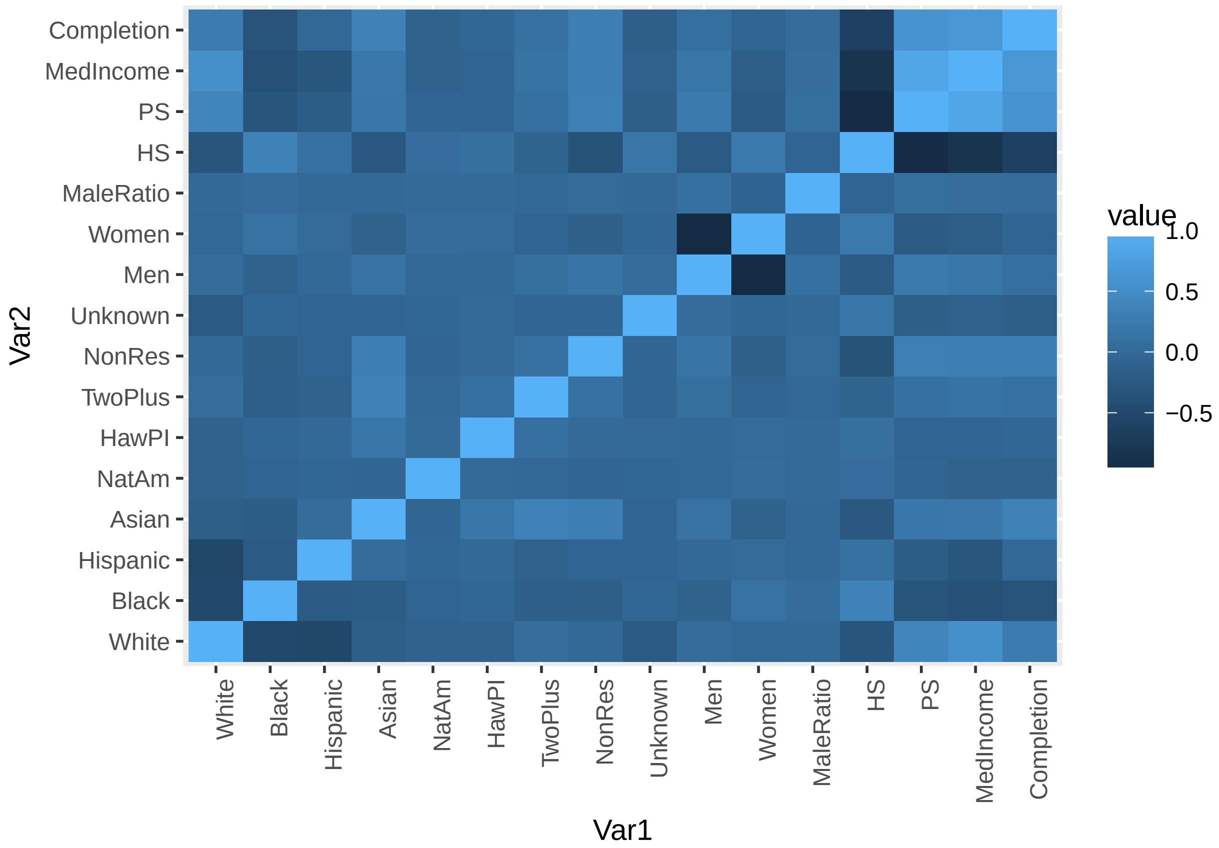


Notes: this is not a terribly helpful plot. One would expect that, perhaps if engineering and mathematics-related jobs paid higher salaries, then schools with a higher percetange of engineering and mathematics degrees would have higher average salaries. But this doesn't seem to be quite apparent.

Predictive model - Student graduation

Evaluate the primary features that affect college program completion: here I decided to start by cherry-picking a few features that I thought would be relevant to this question. In particular, I looked at race, gender, and family information (level of parental education, family income). I started with this subset of features and then did some feature analysis to figure out which data elements would be the best predictors.

```
library(mlbench)
library(caret)

# Pull out Race data
data_race = as.data.frame(data[,293:301])
cols = 1:9
data_race[,cols] = suppressWarnings(apply(data_race[,cols], 2, function(x) as.numeric(as.character(x)))
colnames(data_race) = c("White", "Black", "Hispanic", "Asian", "NatAm", "HawPI", "TwoPlus", "NonRes", "

# Pull out Gender data
data_gender = cbind(data$UGDS_MEN, data$UGDS_WOMEN)
data_gender = apply(data_gender, 2, function(x) as.numeric(as.character(x)))
colnames(data_gender) = c("Men", "Women")
data_gender = as.data.frame(data_gender)
data_gender$MaleRatio = data_gender$Men/data_gender$Women

# Pull out Family Info
data_par = data[, 1426:1427]
data_par = cbind(data_par, data$FAMINC)
data_par = suppressWarnings(apply(data_par, 2, function(x) as.numeric(as.character(x))))
colnames(data_par) = c("HS", "PS", "MedIncome")
data_par = as.data.frame(data_par)

# Combine everything together
data_cor = cbind(data_race, data_gender, data_par)
data_cor = cbind(data_cor, data$C150_4)
colnames(data_cor)[16] = "Completion"
data_cor$Completion = suppressWarnings(as.numeric(as.character(data_cor$Completion)))

data_cor = na.omit(data_cor)
cor = cor(data_cor)
h = qplot(x=Var1, y=Var2, data=melt(cor), fill=value, geom="tile")
h + theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
high_cor = melt(cor)
high_cor = high_cor[abs(high_cor$value)>0.5,]
high_cor = high_cor[high_cor$value!=1,]
high_cor = high_cor[high_cor$Var2=="Completion",]
print(high_cor)
```

```
##          Var1       Var2      value
## 253        HS Completion -0.6270467
## 254        PS Completion  0.5918200
## 255 MedIncome Completion  0.6477116
```

We can see from above that the highest correlated factors to completion rates are parental high school highest degree (negative), parental post-secondary highest degree (positive), and median income (positive).

We will now use these three factors to generate a regression on completion rates.

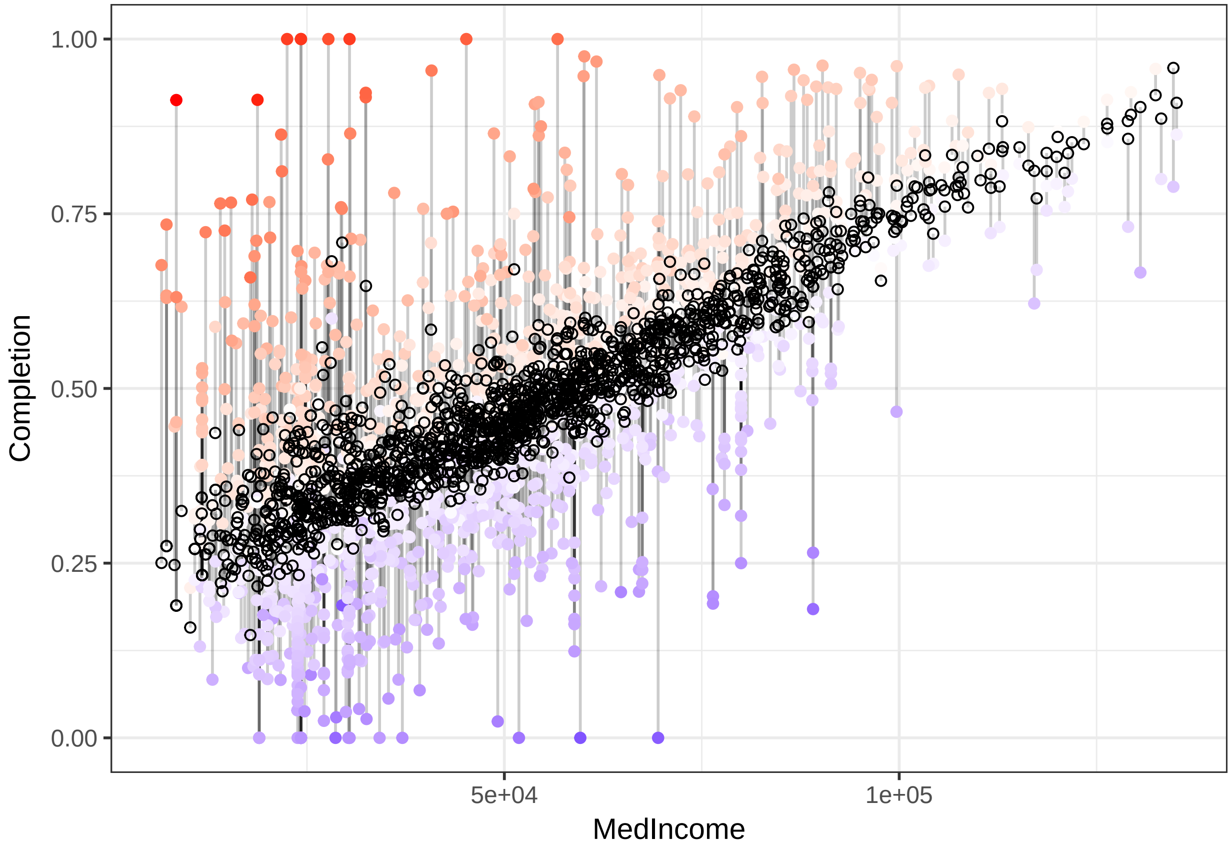
Develop a model to predict student graduation

First, I started with a simple multiple linear regression, using the top three features identified above. The plot below shows the predicted values (in black) and the actual values (colored basd on residual size and direction) for the most-correlated variable, median income. We can see that there are a few outliers – including completion rates of 1 and 0, which seems suspicious – but otherwise most of the points fall in a linear-trending direction.

```
fit = lm(Completion ~ HS + PS + MedIncome, data=data_cor)
summary(fit)
```

```
##
## Call:
## lm(formula = Completion ~ HS + PS + MedIncome, data = data_cor)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53432 -0.08516 -0.00268  0.07385  0.72336
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.728e+00  1.444e-01  11.970   <2e-16 ***
## HS          -1.942e+00  1.685e-01 -11.523   <2e-16 ***
## PS          -1.275e+00  1.456e-01  -8.754   <2e-16 ***
## MedIncome    3.946e-06  2.706e-07  14.581   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1485 on 1824 degrees of freedom
## Multiple R-squared:  0.4642, Adjusted R-squared:  0.4634
## F-statistic: 526.8 on 3 and 1824 DF,  p-value: < 2.2e-16
```

```r
data_cor$predicted = predict(fit)
data_cor$residuals = residuals(fit)

p = ggplot(data_cor, aes(x = MedIncome, y = Completion)) +
  geom_segment(aes(xend = MedIncome, yend = predicted), alpha = .2) +  # Lines to connect points
  geom_point() +  # Points of actual values
  geom_point(aes(y = predicted), shape = 1)

ggplot(data_cor, aes(x = MedIncome, y = Completion)) +
  geom_segment(aes(xend = MedIncome, yend = predicted), alpha = .2) +
  geom_point(aes(color = residuals)) +
  scale_color_gradient2(low = "blue", mid = "white", high = "red") +
  guides(color = FALSE) +
  geom_point(aes(y = predicted), shape = 1) +
  theme_bw()
```

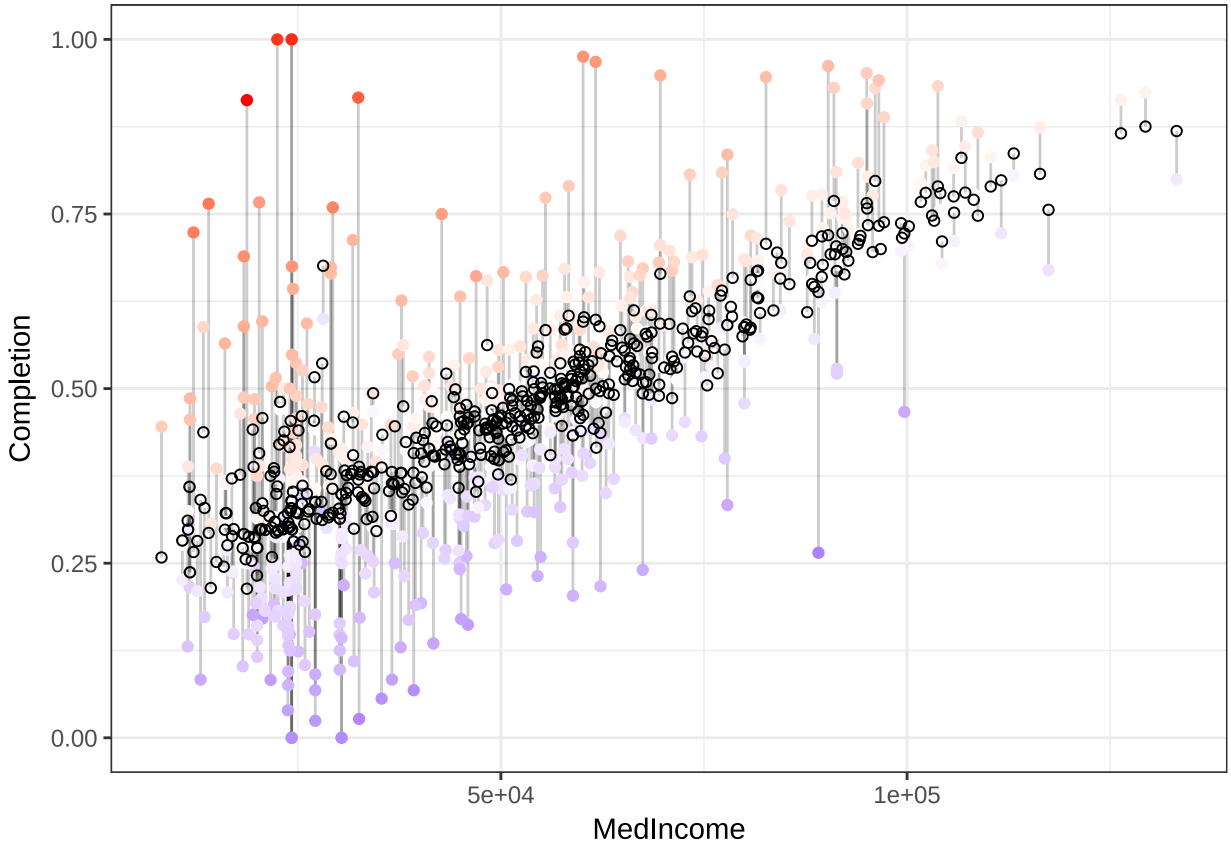


This model shows a good fit, with low p-values for each of the selected features. Let's now create a test dataset to create the model.

```
data_cor$ind = sample(2, nrow(data_cor), replace=TRUE, prob=c(0.67, 0.33))
data_cor.train = data_cor[data_cor$ind==1,1:16]
data_cor.test = data_cor[data_cor$ind==2,1:16]

set.seed(120)
myglm = glm(Completion ~ HS + PS + MedIncome, data=data_cor.train)
summary(myglm)
```

```
##
## Call:
## glm(formula = Completion ~ HS + PS + MedIncome, data = data_cor.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.53731  -0.08398  -0.00450   0.07490   0.71925
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.716e+00  1.801e-01   9.530  < 2e-16 ***
## HS          -1.942e+00  2.099e-01  -9.251  < 2e-16 ***
## PS          -1.223e+00  1.819e-01  -6.724  2.7e-11 ***
## MedIncome    3.585e-06  3.335e-07  10.749  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.02261663)
##
##     Null deviance: 49.738  on 1227  degrees of freedom
## Residual deviance: 27.683  on 1224  degrees of freedom
## AIC: -1162.1
##
## Number of Fisher Scoring iterations: 2
```

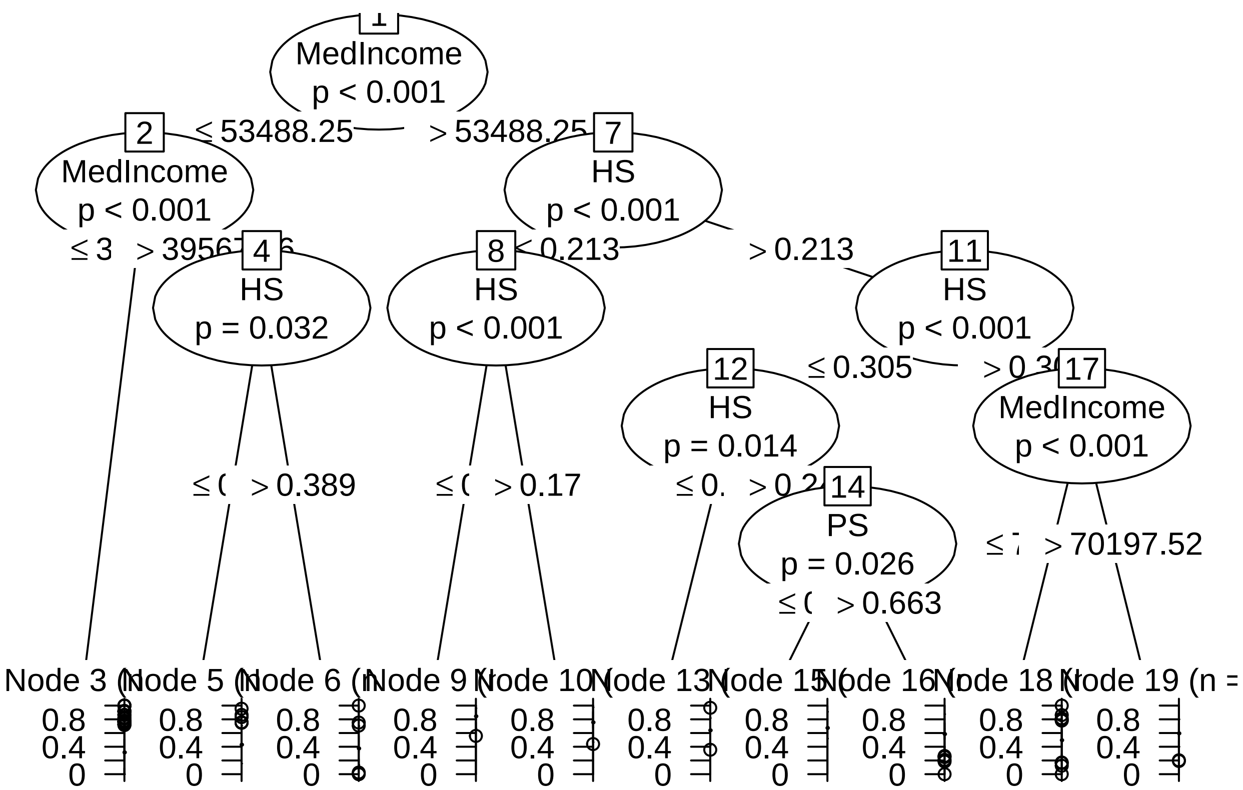
```
pred = predict(myglm, newdata = data_cor.test)
data_cor.test$pred = pred
data_cor.test$res = data_cor.test$Completion-data_cor.test$pred

ggplot(data_cor.test, aes(x = MedIncome, y = Completion)) +
  geom_segment(aes(xend = MedIncome, yend = pred), alpha = .2) +
  geom_point(aes(color = res)) +
  scale_color_gradient2(low = "blue", mid = "white", high = "red") +
  guides(color = FALSE) +
  geom_point(aes(y = pred), shape = 1) +
  theme_bw()
```



Similar to above graph, with subset of data.

Just for fun, here is a decision tree based on the three highest-correlated factors:

```r
library(party)
comp_tree <- ctree(Completion ~ HS + PS + MedIncome, data=data_cor)
plot(comp_tree)
```


MedIncome
p < 0.001
≤ 53488.25
> 53488.25
MedIncome
p < 0.001
≤ 3
> 39567.6
HS
p = 0.032
≤ 0
> 0.389
HS
p < 0.001
≤ 0.213
> 0.213
HS
p < 0.001
≤ 0
> 0.17
HS
p < 0.001
≤ 0.305
> 0.3
HS
p = 0.014
≤ 0.
> 0.2
PS
p = 0.026
≤ 0
> 0.663
MedIncome
p < 0.001
≤ 7
> 70197.52
Node 3 (
Node 5 (
Node 6 (n
Node 9 (
Node 10 (
Node 13 (
Node 15 (
Node 16 (
Node 18 (
Node 19 (n =