# STAT 4610 FCQ Project

Andrew Jossi and Brady Schiff

May 1, 2024

## Contents

## Introduction

This project aims to conduct a comprehensive examination of the FCQ dataset sourced from CU's Boulder, Colorado Springs, and Denver campuses, which can be accessed at www.colorado.edu/fcq/fcq-results. The primary purpose of this project is to explore the factors that contribute to exceptional teaching quality. Employing a variety of models, ranging from easily interpretable to more complex predictive frameworks, our goal is to pinpoint the key predictors of outstanding instruction. Our analysis will encompass four distinct types of predictive models, each offering different predictive capabilities and interpretability levels: a stepwise linear regression model, a lasso model, logistic model and diverse tree models. Ultimately, our goal is to utilize these models to determine if a new instructor will thrive at the University of Colorado. Specifically, our analysis will focus on the 2010-2019 dataset due to its size and more defined response variable, particularly the 'Instr' column, compared to other datasets available.

## Data

Our dataset includes the FCQ (faculty course questionnaire) results spanning from 2010 to 2019 from the University of Colorado. We have opted to exclude the more recent dataset for the reasons previously mentioned. The 2010-2019 dataset contains a total of 28 columns, including two columns representing the standard deviations of the 'Instr overall' and 'Course overall' ratings. We have chosen not to utilize these columns as our focus is on identifying predictors that contribute to the 'Instr overall' rating, and these standard deviation values lack interpretability. For instance, stating that a low 'Instr SD' indicates a good instructor is not particularly informative. Instead, our analysis will concentrate on the other predictors with most of the models using only numeric predictors. A description of the data set can be seen below where the mean scores are all measured on the scale: 1=lowest…6=highest.

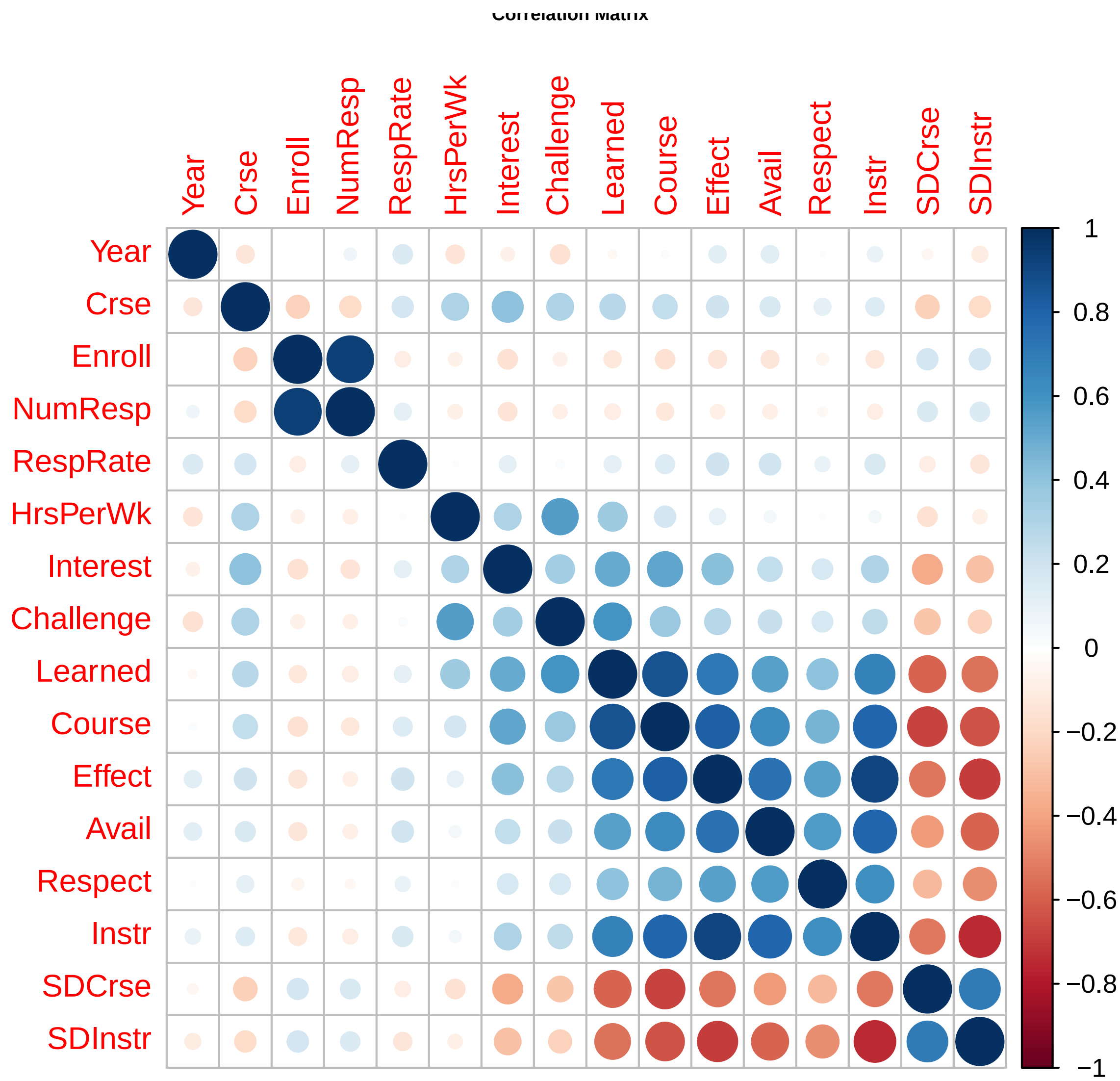
| Column Header | Full Description |
|---|---|
| Term | Term |
| Year | Year |
| Campus | Campus |
| College | College |
| Dept | Department |
| Sbjct | Subject |
| Crse | Course |
| Sect | Course Section |
| Crse Title | Course Title |
| Instructor Name | Instructor Name |
| Instr Grp | Instructor Group |
| Crse Type | Course Type |
| Crse Lvl | Course Level |
| Onlin | Online Administration |
| Enroll | Course Enrollment # |
| # Resp | # of Responses |
| Resp Rate | Response Rate |
| HrsPerWk | the average number of hours students spent on this course per weeek. |
| Interest | Mean Score of personal interest in this course before enrolling |
| Challenge | Mean Score of intellectual challenge of this course |
| Learned | Mean Score of how much students learned in this course |
| Course | Mean Score of how students rated the course overall |
| Effect | Mean Score of the instructor's effectiveness in encouraging interest in this subject. |

| Column Header | Full Description |
| --- | --- |
| Avail | Mean Score of the instructor's availability |
| Respect | Mean Score of the instructor's respect of students |
| Instr | Mean Score of the instructor's overall rating |

## Exploratory Analysis

```
## Rows: 112249 Columns: 28
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (13): Term, Campus, College, Dept, Sbjct, Sect, Crse Title, Instructor N...
## dbl (15): Year, Crse, Enroll, # Resp, HrsPerWk, Interest, Challenge, Learned...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

**Plots**




```
## $corr
##                   Year      Crse       Enroll     NumResp    RespRate
## Year       1.000000000 -0.1306290  0.006414238  0.06201083  0.15597607
## Crse      -0.130629009  1.0000000 -0.222069595 -0.18924538  0.18584950
## Enroll     0.006414238 -0.2220696  1.000000000  0.93826910 -0.09820069
## NumResp    0.062010832 -0.1892454  0.938269104  1.00000000  0.11837814
## RespRate   0.155976069  0.1858495 -0.098200687  0.11837814  1.00000000
## HrsPerWk  -0.140797849  0.3059006 -0.074552876 -0.08570153 -0.01256967
## Interest  -0.074529632  0.4060418 -0.157735534 -0.14009224  0.11641968
## Challenge -0.154318703  0.3028322 -0.077523179 -0.08189399  0.02889099
## Learned   -0.030082393  0.2719692 -0.120385199 -0.09992365  0.11772001
## Course     0.021054829  0.2466902 -0.153035187 -0.12264138  0.14812414
## Effect     0.122170483  0.2032579 -0.131614831 -0.08327244  0.20929969
## Avail      0.125912900  0.1637915 -0.130741500 -0.08567093  0.19091072
## Respect    0.010317477  0.1171668 -0.057471607 -0.03777497  0.09203308
```

```
## Instr      0.095075958  0.1402181 -0.128690400 -0.09104464  0.16784861
## SDCrse    -0.043625552 -0.2317299  0.184407111  0.16147757 -0.09752733
## SDInstr   -0.103569054 -0.1876492  0.188096739  0.15504339 -0.13753209
##              HrsPerWk    Interest   Challenge     Learned      Course
## Year      -0.14079785 -0.07452963 -0.15431870 -0.03008239  0.02105483
## Crse       0.30590060  0.40604185  0.30283223  0.27196921  0.24669019
## Enroll    -0.07455288 -0.15773553 -0.07752318 -0.12038520 -0.15303519
## NumResp   -0.08570153 -0.14009224 -0.08189399 -0.09992365 -0.12264138
## RespRate  -0.01256967  0.11641968  0.02889099  0.11772001  0.14812414
## HrsPerWk   1.00000000  0.30313385  0.55905804  0.35621894  0.18841839
## Interest   0.30313385  1.00000000  0.34728288  0.50062681  0.52430057
## Challenge  0.55905804  0.34728288  1.00000000  0.59656233  0.37617024
## Learned    0.35621894  0.50062681  0.59656233  1.00000000  0.86103341
## Course     0.18841839  0.52430057  0.37617024  0.86103341  1.00000000
## Effect     0.10627687  0.41049885  0.28107704  0.71171219  0.81461837
## Avail      0.05570728  0.24891200  0.22877828  0.54460901  0.63172780
## Respect    0.01613597  0.17538233  0.17790587  0.40906850  0.46602056
## Instr      0.05798114  0.30163140  0.25021638  0.67860127  0.79531869
## SDCrse    -0.15710065 -0.37736567 -0.27752155 -0.58016313 -0.67084535
## SDInstr   -0.08419907 -0.29924559 -0.22332572 -0.54295229 -0.63604003
##                Effect       Avail     Respect       Instr      SDCrse
## Year       0.12217048  0.12591290  0.01031748  0.09507596 -0.04362555
## Crse       0.20325792  0.16379149  0.11716678  0.14021808 -0.23172988
## Enroll    -0.13161483 -0.13074150 -0.05747161 -0.12869040  0.18440711
## NumResp   -0.08327244 -0.08567093 -0.03777497 -0.09104464  0.16147757
## RespRate   0.20929969  0.19091072  0.09203308  0.16784861 -0.09752733
## HrsPerWk   0.10627687  0.05570728  0.01613597  0.05798114 -0.15710065
## Interest   0.41049885  0.24891200  0.17538233  0.30163140 -0.37736567
## Challenge  0.28107704  0.22877828  0.17790587  0.25021638 -0.27752155
## Learned    0.71171219  0.54460901  0.40906850  0.67860127 -0.58016313
## Course     0.81461837  0.63172780  0.46602056  0.79531869 -0.67084535
## Effect     1.00000000  0.74962431  0.54301328  0.91551761 -0.53910523
## Avail      0.74962431  1.00000000  0.56293383  0.79414051 -0.42091243
## Respect    0.54301328  0.56293383  1.00000000  0.61125782 -0.32565349
## Instr      0.91551761  0.79414051  0.61125782  1.00000000 -0.52942575
## SDCrse    -0.53910523 -0.42091243 -0.32565349 -0.52942575  1.00000000
## SDInstr   -0.69229347 -0.58937798 -0.46462520 -0.74392965  0.70936897
##               SDInstr
## Year      -0.10356905
## Crse      -0.18764923
## Enroll     0.18809674
## NumResp    0.15504339
## RespRate  -0.13753209
## HrsPerWk  -0.08419907
## Interest  -0.29924559
## Challenge -0.22332572
## Learned   -0.54295229
## Course    -0.63604003
## Effect    -0.69229347
## Avail     -0.58937798
## Respect   -0.46462520
## Instr     -0.74392965
## SDCrse     0.70936897
## SDInstr    1.00000000
```

```
##
## $corrPos
##         xName     yName  x  y         corr
## 1        Year      Year  1 16  1.000000000
## 2        Year      Crse  1 15 -0.130629009
## 3        Year    Enroll  1 14  0.006414238
## 4        Year   NumResp  1 13  0.062010832
## 5        Year  RespRate  1 12  0.155976069
## 6        Year  HrsPerWk  1 11 -0.140797849
## 7        Year  Interest  1 10 -0.074529632
## 8        Year Challenge  1  9 -0.154318703
## 9        Year   Learned  1  8 -0.030082393
## 10       Year    Course  1  7  0.021054829
## 11       Year    Effect  1  6  0.122170483
## 12       Year     Avail  1  5  0.125912900
## 13       Year   Respect  1  4  0.010317477
## 14       Year     Instr  1  3  0.095075958
## 15       Year    SDCrse  1  2 -0.043625552
## 16       Year   SDInstr  1  1 -0.103569054
## 17       Crse      Year  2 16 -0.130629009
## 18       Crse      Crse  2 15  1.000000000
## 19       Crse    Enroll  2 14 -0.222069595
## 20       Crse   NumResp  2 13 -0.189245385
## 21       Crse  RespRate  2 12  0.185849500
## 22       Crse  HrsPerWk  2 11  0.305900598
## 23       Crse  Interest  2 10  0.406041846
## 24       Crse Challenge  2  9  0.302832231
## 25       Crse   Learned  2  8  0.271969209
## 26       Crse    Course  2  7  0.246690186
## 27       Crse    Effect  2  6  0.203257917
## 28       Crse     Avail  2  5  0.163791486
## 29       Crse   Respect  2  4  0.117166775
## 30       Crse     Instr  2  3  0.140218082
## 31       Crse    SDCrse  2  2 -0.231729883
## 32       Crse   SDInstr  2  1 -0.187649230
## 33     Enroll      Year  3 16  0.006414238
## 34     Enroll      Crse  3 15 -0.222069595
## 35     Enroll    Enroll  3 14  1.000000000
## 36     Enroll   NumResp  3 13  0.938269104
## 37     Enroll  RespRate  3 12 -0.098200687
## 38     Enroll  HrsPerWk  3 11 -0.074552876
## 39     Enroll  Interest  3 10 -0.157735534
## 40     Enroll Challenge  3  9 -0.077523179
## 41     Enroll   Learned  3  8 -0.120385199
## 42     Enroll    Course  3  7 -0.153035187
## 43     Enroll    Effect  3  6 -0.131614831
## 44     Enroll     Avail  3  5 -0.130741500
## 45     Enroll   Respect  3  4 -0.057471607
## 46     Enroll     Instr  3  3 -0.128690400
## 47     Enroll    SDCrse  3  2  0.184407111
## 48     Enroll   SDInstr  3  1  0.188096739
## 49    NumResp      Year  4 16  0.062010832
## 50    NumResp      Crse  4 15 -0.189245385
## 51    NumResp    Enroll  4 14  0.938269104
```

```
## 52    NumResp   NumResp  4 13  1.000000000
## 53    NumResp  RespRate  4 12  0.118378136
## 54    NumResp  HrsPerWk  4 11 -0.085701526
## 55    NumResp  Interest  4 10 -0.140092244
## 56    NumResp Challenge  4  9 -0.081893990
## 57    NumResp   Learned  4  8 -0.099923654
## 58    NumResp    Course  4  7 -0.122641385
## 59    NumResp    Effect  4  6 -0.083272443
## 60    NumResp     Avail  4  5 -0.085670933
## 61    NumResp   Respect  4  4 -0.037774972
## 62    NumResp     Instr  4  3 -0.091044638
## 63    NumResp    SDCrse  4  2  0.161477567
## 64    NumResp   SDInstr  4  1  0.155043386
## 65   RespRate      Year  5 16  0.155976069
## 66   RespRate      Crse  5 15  0.185849500
## 67   RespRate    Enroll  5 14 -0.098200687
## 68   RespRate   NumResp  5 13  0.118378136
## 69   RespRate  RespRate  5 12  1.000000000
## 70   RespRate  HrsPerWk  5 11 -0.012569671
## 71   RespRate  Interest  5 10  0.116419676
## 72   RespRate Challenge  5  9  0.028890987
## 73   RespRate   Learned  5  8  0.117720015
## 74   RespRate    Course  5  7  0.148124137
## 75   RespRate    Effect  5  6  0.209299692
## 76   RespRate     Avail  5  5  0.190910720
## 77   RespRate   Respect  5  4  0.092033077
## 78   RespRate     Instr  5  3  0.167848611
## 79   RespRate    SDCrse  5  2 -0.097527326
## 80   RespRate   SDInstr  5  1 -0.137532089
## 81   HrsPerWk      Year  6 16 -0.140797849
## 82   HrsPerWk      Crse  6 15  0.305900598
## 83   HrsPerWk    Enroll  6 14 -0.074552876
## 84   HrsPerWk   NumResp  6 13 -0.085701526
## 85   HrsPerWk  RespRate  6 12 -0.012569671
## 86   HrsPerWk  HrsPerWk  6 11  1.000000000
## 87   HrsPerWk  Interest  6 10  0.303133850
## 88   HrsPerWk Challenge  6  9  0.559058043
## 89   HrsPerWk   Learned  6  8  0.356218941
## 90   HrsPerWk    Course  6  7  0.188418387
## 91   HrsPerWk    Effect  6  6  0.106276873
## 92   HrsPerWk     Avail  6  5  0.055707283
## 93   HrsPerWk   Respect  6  4  0.016135969
## 94   HrsPerWk     Instr  6  3  0.057981137
## 95   HrsPerWk    SDCrse  6  2 -0.157100655
## 96   HrsPerWk   SDInstr  6  1 -0.084199071
## 97   Interest      Year  7 16 -0.074529632
## 98   Interest      Crse  7 15  0.406041846
## 99   Interest    Enroll  7 14 -0.157735534
## 100  Interest   NumResp  7 13 -0.140092244
## 101  Interest  RespRate  7 12  0.116419676
## 102  Interest  HrsPerWk  7 11  0.303133850
## 103  Interest  Interest  7 10  1.000000000
## 104  Interest Challenge  7  9  0.347282883
## 105  Interest   Learned  7  8  0.500626809
```

```
## 106  Interest    Course  7  7  0.524300575
## 107  Interest    Effect  7  6  0.410498846
## 108  Interest     Avail  7  5  0.248911999
## 109  Interest   Respect  7  4  0.175382333
## 110  Interest     Instr  7  3  0.301631405
## 111  Interest    SDCrse  7  2 -0.377365670
## 112  Interest   SDInstr  7  1 -0.299245595
## 113 Challenge      Year  8 16 -0.154318703
## 114 Challenge      Crse  8 15  0.302832231
## 115 Challenge    Enroll  8 14 -0.077523179
## 116 Challenge   NumResp  8 13 -0.081893990
## 117 Challenge  RespRate  8 12  0.028890987
## 118 Challenge  HrsPerWk  8 11  0.559058043
## 119 Challenge  Interest  8 10  0.347282883
## 120 Challenge Challenge  8  9  1.000000000
## 121 Challenge   Learned  8  8  0.596562333
## 122 Challenge    Course  8  7  0.376170242
## 123 Challenge    Effect  8  6  0.281077045
## 124 Challenge     Avail  8  5  0.228778281
## 125 Challenge   Respect  8  4  0.177905872
## 126 Challenge     Instr  8  3  0.250216383
## 127 Challenge    SDCrse  8  2 -0.277521548
## 128 Challenge   SDInstr  8  1 -0.223325716
## 129   Learned      Year  9 16 -0.030082393
## 130   Learned      Crse  9 15  0.271969209
## 131   Learned    Enroll  9 14 -0.120385199
## 132   Learned   NumResp  9 13 -0.099923654
## 133   Learned  RespRate  9 12  0.117720015
## 134   Learned  HrsPerWk  9 11  0.356218941
## 135   Learned  Interest  9 10  0.500626809
## 136   Learned Challenge  9  9  0.596562333
## 137   Learned   Learned  9  8  1.000000000
## 138   Learned    Course  9  7  0.861033412
## 139   Learned    Effect  9  6  0.711712191
## 140   Learned     Avail  9  5  0.544609011
## 141   Learned   Respect  9  4  0.409068496
## 142   Learned     Instr  9  3  0.678601268
## 143   Learned    SDCrse  9  2 -0.580163128
## 144   Learned   SDInstr  9  1 -0.542952290
## 145    Course      Year 10 16  0.021054829
## 146    Course      Crse 10 15  0.246690186
## 147    Course    Enroll 10 14 -0.153035187
## 148    Course   NumResp 10 13 -0.122641385
## 149    Course  RespRate 10 12  0.148124137
## 150    Course  HrsPerWk 10 11  0.188418387
## 151    Course  Interest 10 10  0.524300575
## 152    Course Challenge 10  9  0.376170242
## 153    Course   Learned 10  8  0.861033412
## 154    Course    Course 10  7  1.000000000
## 155    Course    Effect 10  6  0.814618369
## 156    Course     Avail 10  5  0.631727798
## 157    Course   Respect 10  4  0.466020563
## 158    Course     Instr 10  3  0.795318690
## 159    Course    SDCrse 10  2 -0.670845351
```

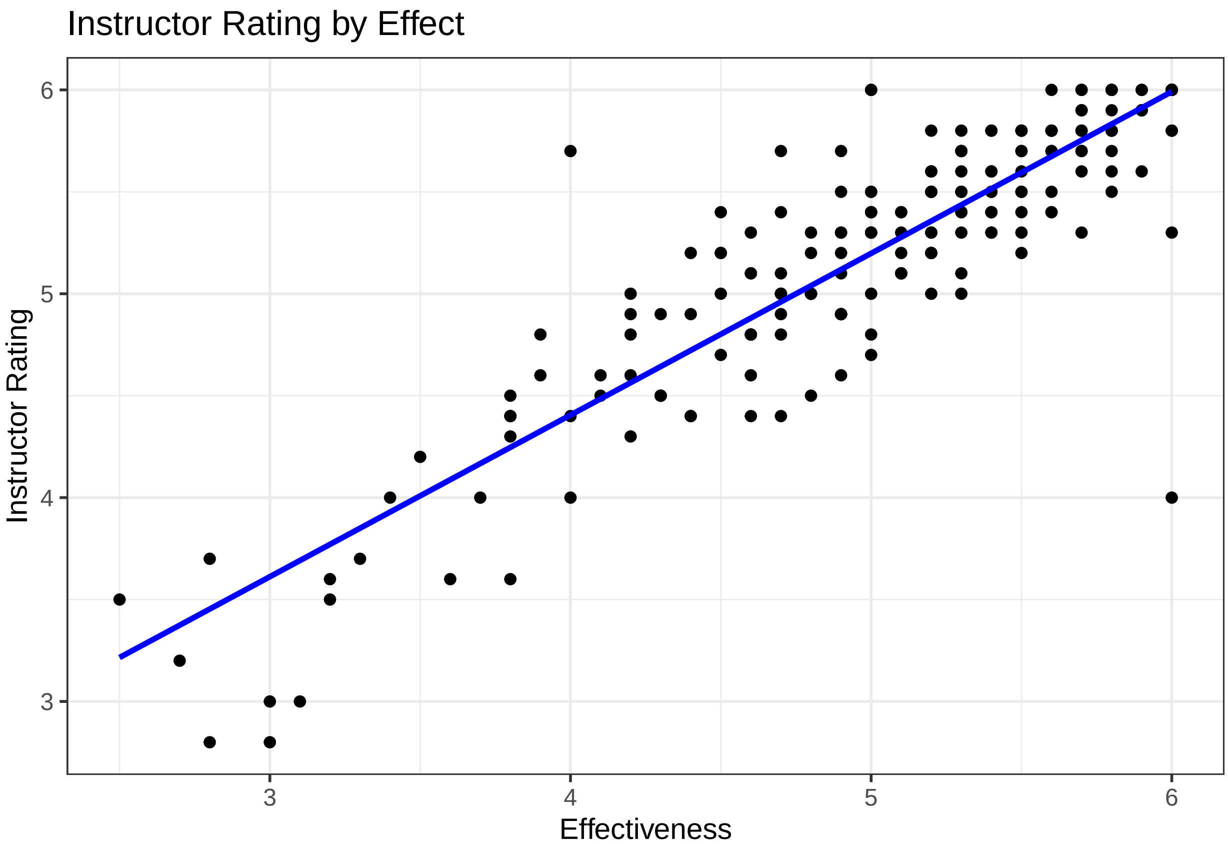
```
## 160    Course   SDInstr 10  1 -0.636040034
## 161    Effect      Year 11 16  0.122170483
## 162    Effect      Crse 11 15  0.203257917
## 163    Effect    Enroll 11 14 -0.131614831
## 164    Effect   NumResp 11 13 -0.083272443
## 165    Effect  RespRate 11 12  0.209299692
## 166    Effect  HrsPerWk 11 11  0.106276873
## 167    Effect  Interest 11 10  0.410498846
## 168    Effect Challenge 11  9  0.281077045
## 169    Effect   Learned 11  8  0.711712191
## 170    Effect    Course 11  7  0.814618369
## 171    Effect    Effect 11  6  1.000000000
## 172    Effect     Avail 11  5  0.749624310
## 173    Effect   Respect 11  4  0.543013278
## 174    Effect     Instr 11  3  0.915517611
## 175    Effect    SDCrse 11  2 -0.539105232
## 176    Effect   SDInstr 11  1 -0.692293468
## 177     Avail      Year 12 16  0.125912900
## 178     Avail      Crse 12 15  0.163791486
## 179     Avail    Enroll 12 14 -0.130741500
## 180     Avail   NumResp 12 13 -0.085670933
## 181     Avail  RespRate 12 12  0.190910720
## 182     Avail  HrsPerWk 12 11  0.055707283
## 183     Avail  Interest 12 10  0.248911999
## 184     Avail Challenge 12  9  0.228778281
## 185     Avail   Learned 12  8  0.544609011
## 186     Avail    Course 12  7  0.631727798
## 187     Avail    Effect 12  6  0.749624310
## 188     Avail     Avail 12  5  1.000000000
## 189     Avail   Respect 12  4  0.562933830
## 190     Avail     Instr 12  3  0.794140511
## 191     Avail    SDCrse 12  2 -0.420912428
## 192     Avail   SDInstr 12  1 -0.589377976
## 193   Respect      Year 13 16  0.010317477
## 194   Respect      Crse 13 15  0.117166775
## 195   Respect    Enroll 13 14 -0.057471607
## 196   Respect   NumResp 13 13 -0.037774972
## 197   Respect  RespRate 13 12  0.092033077
## 198   Respect  HrsPerWk 13 11  0.016135969
## 199   Respect  Interest 13 10  0.175382333
## 200   Respect Challenge 13  9  0.177905872
## 201   Respect   Learned 13  8  0.409068496
## 202   Respect    Course 13  7  0.466020563
## 203   Respect    Effect 13  6  0.543013278
## 204   Respect     Avail 13  5  0.562933830
## 205   Respect   Respect 13  4  1.000000000
## 206   Respect     Instr 13  3  0.611257815
## 207   Respect    SDCrse 13  2 -0.325653490
## 208   Respect   SDInstr 13  1 -0.464625199
## 209     Instr      Year 14 16  0.095075958
## 210     Instr      Crse 14 15  0.140218082
## 211     Instr    Enroll 14 14 -0.128690400
## 212     Instr   NumResp 14 13 -0.091044638
## 213     Instr  RespRate 14 12  0.167848611
```

```
## 214      Instr  HrsPerWk 14 11  0.057981137
## 215      Instr  Interest 14 10  0.301631405
## 216      Instr Challenge 14  9  0.250216383
## 217      Instr   Learned 14  8  0.678601268
## 218      Instr    Course 14  7  0.795318690
## 219      Instr    Effect 14  6  0.915517611
## 220      Instr     Avail 14  5  0.794140511
## 221      Instr   Respect 14  4  0.611257815
## 222      Instr     Instr 14  3  1.000000000
## 223      Instr    SDCrse 14  2 -0.529425751
## 224      Instr   SDInstr 14  1 -0.743929650
## 225     SDCrse      Year 15 16 -0.043625552
## 226     SDCrse      Crse 15 15 -0.231729883
## 227     SDCrse    Enroll 15 14  0.184407111
## 228     SDCrse   NumResp 15 13  0.161477567
## 229     SDCrse  RespRate 15 12 -0.097527326
## 230     SDCrse  HrsPerWk 15 11 -0.157100655
## 231     SDCrse  Interest 15 10 -0.377365670
## 232     SDCrse Challenge 15  9 -0.277521548
## 233     SDCrse   Learned 15  8 -0.580163128
## 234     SDCrse    Course 15  7 -0.670845351
## 235     SDCrse    Effect 15  6 -0.539105232
## 236     SDCrse     Avail 15  5 -0.420912428
## 237     SDCrse   Respect 15  4 -0.325653490
## 238     SDCrse     Instr 15  3 -0.529425751
## 239     SDCrse    SDCrse 15  2  1.000000000
## 240     SDCrse   SDInstr 15  1  0.709368973
## 241    SDInstr      Year 16 16 -0.103569054
## 242    SDInstr      Crse 16 15 -0.187649230
## 243    SDInstr    Enroll 16 14  0.188096739
## 244    SDInstr   NumResp 16 13  0.155043386
## 245    SDInstr  RespRate 16 12 -0.137532089
## 246    SDInstr  HrsPerWk 16 11 -0.084199071
## 247    SDInstr  Interest 16 10 -0.299245595
## 248    SDInstr Challenge 16  9 -0.223325716
## 249    SDInstr   Learned 16  8 -0.542952290
## 250    SDInstr    Course 16  7 -0.636040034
## 251    SDInstr    Effect 16  6 -0.692293468
## 252    SDInstr     Avail 16  5 -0.589377976
## 253    SDInstr   Respect 16  4 -0.464625199
## 254    SDInstr     Instr 16  3 -0.743929650
## 255    SDInstr    SDCrse 16  2  0.709368973
## 256    SDInstr   SDInstr 16  1  1.000000000
##
## $arg
## $arg$type
## [1] "full"
```

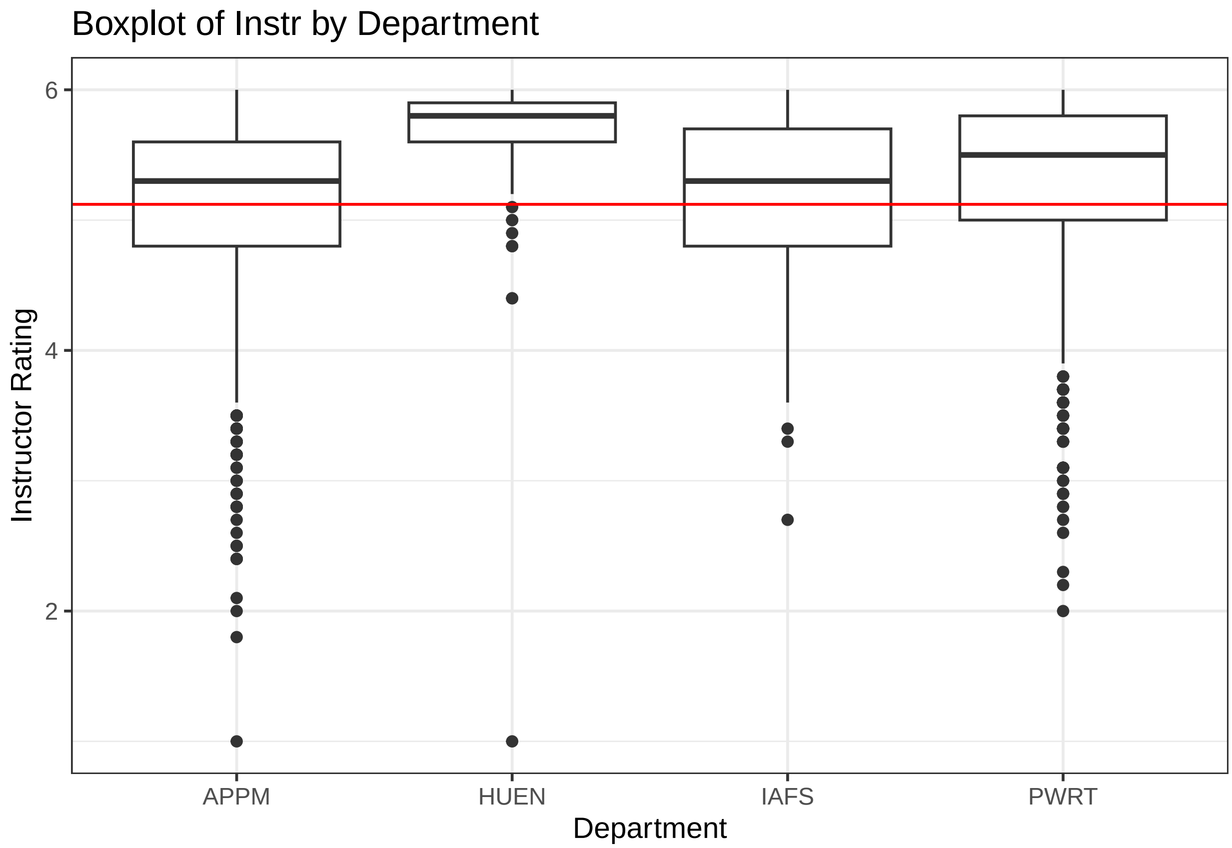
```
## `geom_smooth()` using formula = 'y ~ x'
```


As we can see within this plot, there is very little correlation within response rate by course rating. It would be logical to predict that there would be a greater response rate when there is a higher course overall rating. It is important to acknowledge this trend is slightly truer within lower division undergraduate courses. Additionally, within upper level undergraduate courses, we can see that there is mainly course ratings above 4, with most people completing the FCQ and the courses having higher (above 50%) response rate.

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




As seen within this graph, there is an extreme positive linear correlation between the rating of an instructor and their effectiveness as a professor. This is seen across all course levels. Intuitively, this should be the case, since someone who rates their professor as very effective is much more likely to rate their professor a good rating.



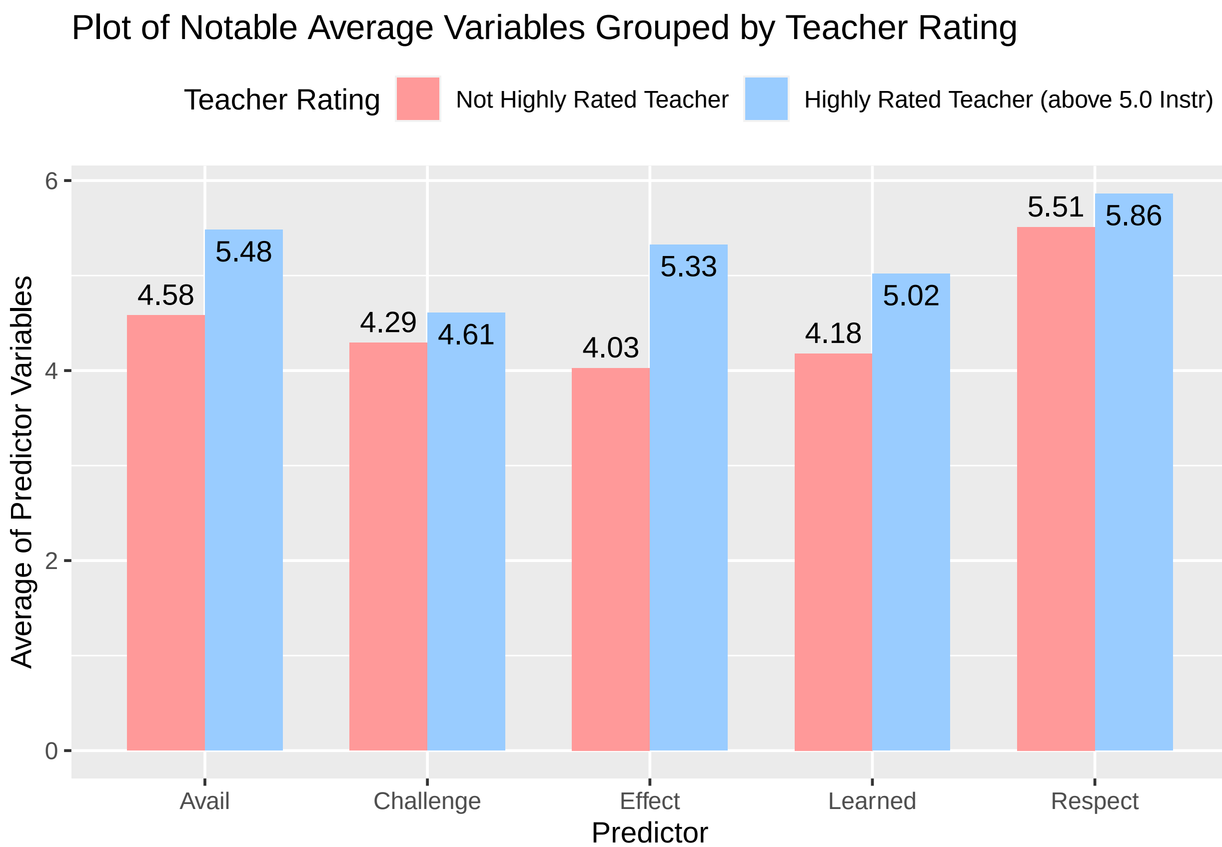

To make this graph, I took a random sample of four departments to analyze if there is a wide sway between departments as far as instructor rating. From this graph, we can see that there is very little sway in this data set, and that the vast majority of the departments are within the range of 5-6. This is of note since random departments will likely have ratings above a 5. This occurs likely since students filling out an FCQ are more likely to enjoy a professor and rate them higher overall.

```
# Group the data based on whether 'Instr' is above or below 4.0
grouped_data <- fcq %>%
  filter(!is.na(Instr)) %>%
  group_by(Instr_group = ifelse(Instr >= 5.0, "Instr >= 5.0", "Instr < 5.0")) %>%
  summarise(Learned = mean(Learned, na.rm = TRUE),
            Challenge = mean(Challenge, na.rm = TRUE),
            Effect = mean(Effect, na.rm = TRUE),
            Avail = mean(Avail, na.rm = TRUE),
            Respect = mean(Respect, na.rm = TRUE))

# Reshape the data into long format for plotting
data_long <- grouped_data %>%
  pivot_longer(cols = Learned:Respect, names_to = "Predictor", values_to = "Average")

# Plot the grouped bar graph of each average of predictor variable
ggplot(data_long, aes(x = Predictor, y = Average, fill = Instr_group)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.7) +
  geom_text(aes(label = ifelse(Instr_group == "Instr < 5.0", round(Average, 2), "")),
            position = position_dodge(width = 0.7), vjust = -0.5) +
  geom_text(aes(label = ifelse(Instr_group == "Instr >= 5.0", round(Average, 2), "")),
            position = position_dodge(width = 0.7), vjust = 1.5) +
```

```r
labs(x = "Predictor", y = "Average of Predictor Variables") +
ggtitle("Plot of Notable Average Variables Grouped by Teacher Rating") +
scale_fill_manual(name = "Teacher Rating",
                  values = c("Instr < 5.0" = "#FF9999", "Instr >= 5.0" = "#99CCFF"),
                  labels = c("Instr < 5.0" = "Not Highly Rated Teacher",
                             "Instr >= 5.0" = "Highly Rated Teacher (above 5.0 Instr)")) +
theme(legend.position = "top", legend.justification = "right")
```




This graph shows the average value of several key predictor variables, and groups them by highly rated and not highly rated teachers. The metric to determine what a highly rated teacher is was a Instr value of 5 or above. This makes up about 68% of the data, and is important to note. It ends up showing a bar graph with a rather simple explanation, that the higher rated teachers usually have higher scores for availability, effectiveness, etc.

## Modeling

### Linear Regression

The first model is an ordinary linear regression, which is represented by the equation below.

$$\hat{Y}_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik}$$

$\hat{Y}_i$ is the predicted response
$\beta_0$ represents the intercept
$\beta_k$ represents the coefficient
$X_{ik}$ represents a feature

$$

$$
\begin{aligned}
\hat{Y}_i = 9.317 &- 0.004842 \times \text{Year} + 0.00032 \times \text{Enroll} - 0.00086 \times \text{NumResp} \\
&- 0.0005121 \times \text{RespRate} - 0.01473 \times \text{HrsPerWk} - 0.09526 \times \text{Interest} \\
&- 0.007383 \times \text{Challenge} - 0.01362 \times \text{Learned} + 0.2034 \times \text{Course} \\
&+ 0.5735 \times \text{Effect} + 0.211 \times \text{Avail} + 0.2195 \times \text{Respect} + \epsilon
\end{aligned}
$$

$$

The RMSE of the liner regression model is 0.25.

```r
predictions <- predict(stepwise.mod, newdata = fcqnum)
rmse <- sqrt(mean((predictions - fcqnum$Instr)^2))
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: 0.248560190505702"
```

Overall, within the linear regression models, we were able to predict a teachers instructor rating to reliably within a quarter of a point. This is significant seeing as the scale is on a 0-6 rating, and a professor with a 0 could be extremely bad at teaching. It is important to get as accurate as possible within these models. Furthermore, this model is based on the fundamental process of training and testing sets. Since it is crucial to test your model to determine a root mean squared error on data the model has never seen before, we found it most useful to train the set using 75% of the available data and test it on the other 25%. Had we made any of our models with 100% training data, we would be testing on compromised data, and the model would likely over perform. It is extremely important to use training and testing sets any time we are analyzing data in depth. As well, our models were built using all numeric variables, and using step wise.

Step wise uses computer trials to find what are deemed the most important variables and most crucial in predicting any given variable. In our trials, we found no explicit difference in accuracy when using step wise or all numeric variables, indicating that all variables are needed to some extent, and that they are not overfitting by being in the model.

### Lasso Model

We can see from our step wise selection that we lost a variable, so we will use a lasso regression to see if we can further simplify the model. Lasso represents regression, but the coefficients also have a penalty term applied to them that makes non-relevant coefficients to go to 0.

Lasso follows the same $\hat{Y}_i$ formula as OLS, but the way the predictors are found is different the OLS equation is changed and instead we are minimizing the function with an added penalty term

$$(Y - X\beta)^T(Y - X\beta) + \lambda\|\beta\|_1 = (Y - X\beta)^T(Y - X\beta) + \lambda\sum_{i=1}^{p}|\beta|$$

```
## [1] "RMSE: 0.247564870047942"
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  8.0909629720
## Year        -0.0042239270
## Enroll       .
## NumResp     -0.0002638326
## RespRate    -0.0005152610
## HrsPerWk    -0.0154176502
## Interest    -0.0913283687
## Challenge   -0.0078022112
## Learned      .
## Course       0.1934552856
## Effect       0.5688930930
## Avail        0.2128799804
## Respect      0.2129136183
```

The RMSE of the lasso model improved slightly but it is still 0.25 when rounded to two decimal places. Based on the complexity of the model, the original linear regression model is still the easiest to interpret and has just about the same RMSE. By looking at the coefficients of the lasso model, we can see that `Enroll` and `Learned` were deemed irrelevant to the model and therefore not used. It is interesting that `Learned` is not used in the lasso model based ont the context of the problem. One could assume that how much a student felt they learned in the course would greatly impact the instructor rating, however we determined that it is not necessary for the model.

**Logistic**

According to this data, there are 39,828 professors that have a rating of a "Highly Rated Professor". There are 18,313 professors that are not "Highly Rated". This means that there are about 68.5% highly rated professors and 31.5% not highly rated professors. If someone were to guess at random, they would be right nearly 68% of the time if they purely guessed "Highly Rated".

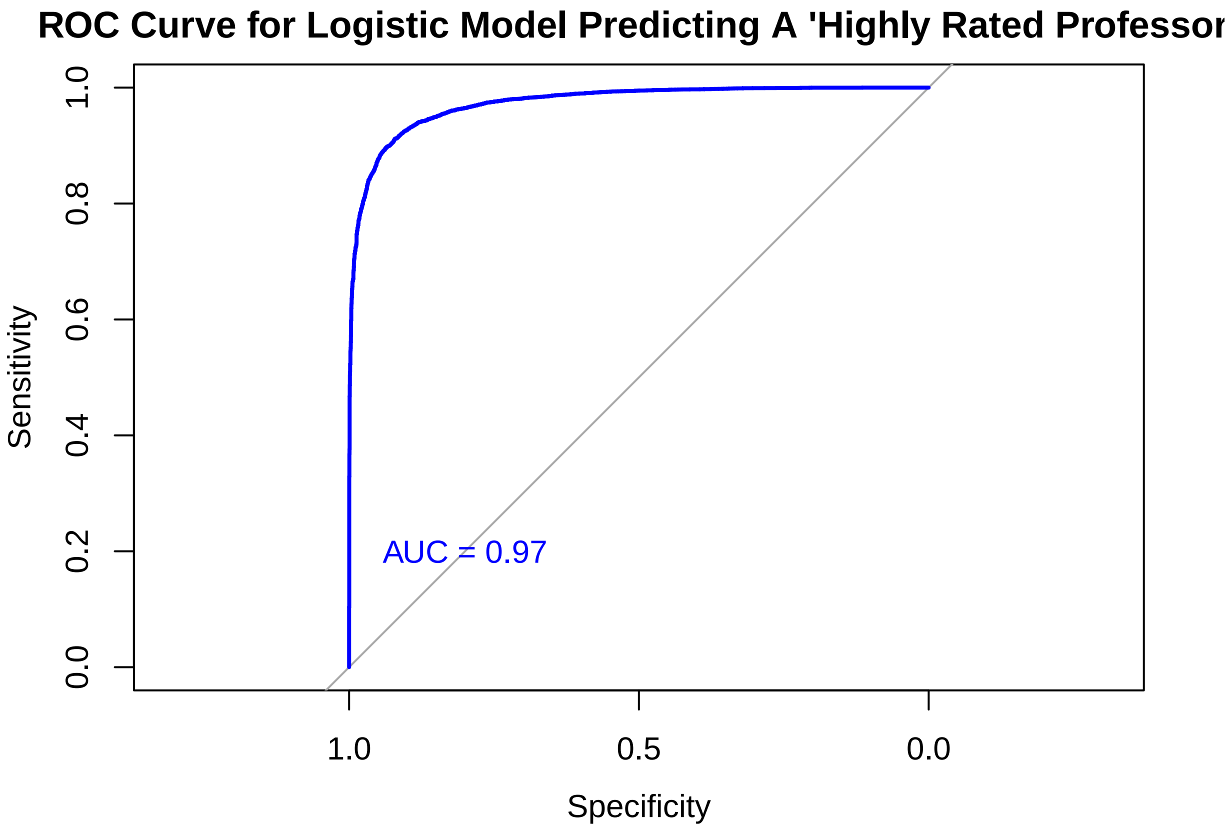
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

| Actual Values | Not Highly Rated | Highly Rated |
|---|---|---|
| Predicted | | |
| Not Highly Rated | 4078 | 654 |
| Highly Rated | 508 | 9295 |

| Actual Values | Not Highly Rated | Highly Rated |
|---|---|---|

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```




From this logistic regression model, we were able to make a model that is 92.15% accurate in determining if a professor is highly rated (above or equal Instr of 5). This model is made up of 11 predictor variables, including most notably RespRate, Course Level, and Effect of professor. These variables are able to identify very efficiently if a professor will clear that 5.0 rating, and earn the "Highly Rated" title. This model has a cut-off at .6, meaning if the model gives a value above .6, the professor is classified as "Highly Rated", and vice versa. This led to the following results:

Accuracy: 92.16% Proportion of Correct Predictions: 92.16% Error Rate: 7.84% True Positive Rate: 93.82% False Positive Rate: 11.51%

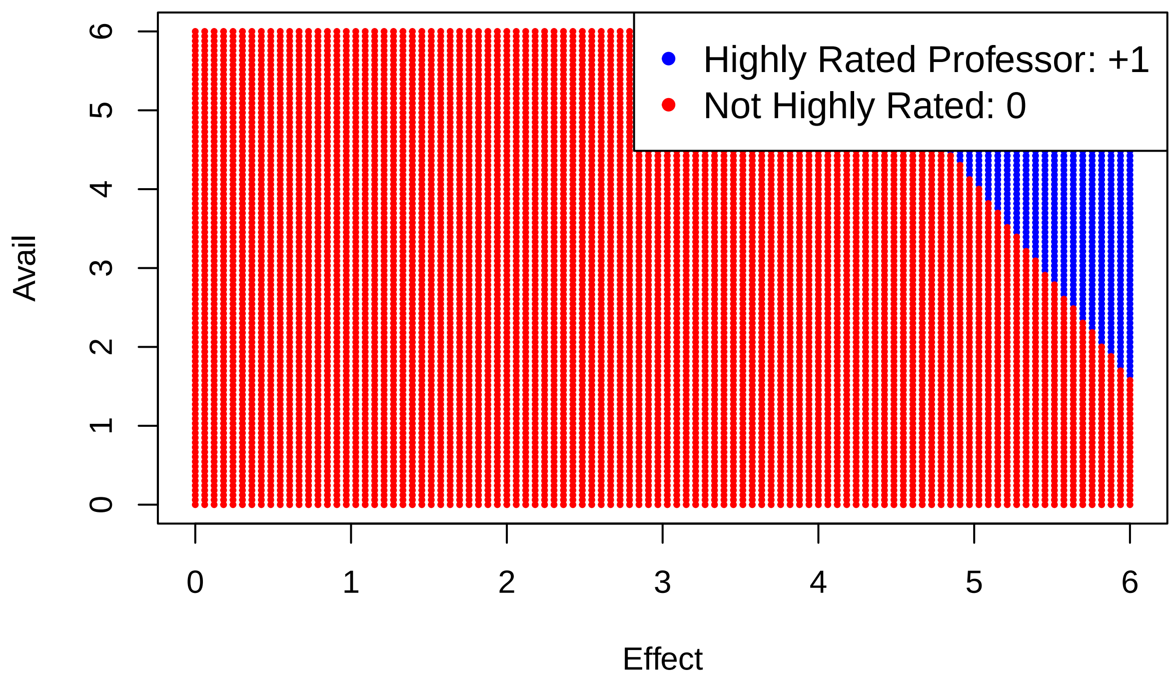
**SVM**

| Actual Values | Not Highly Rated | Highly Rated |
|---|---|---|
| Predicted | | |
| Not Highly Rated | 3822 | 559 |
| Highly Rated | 730 | 9425 |

Accuracy: 91.13% Proportion of Correct Predictions: 91.13% Error Rate: 8.87% True Positive Rate: 94.44% False Positive Rate: 16.04%

Within the support vector machine, we are able to get an extremely accurate and simple model. By using only two variables, the machine is able to generate a cutoff line that predicts where we should assume the professor is highly rated. Since they share a linear relationship and are closely related, it made more sense to have this support vector machine act linearly. With that, using a tuning parameter, I found the best cost for the model to be 0.1. This was the most efficient and accurate for the model. In the end, it ended up being slightly less accurate than the logisitic model. What is interesting is that it only takes in two predictor variables as opposed to 11 within the logistic model.

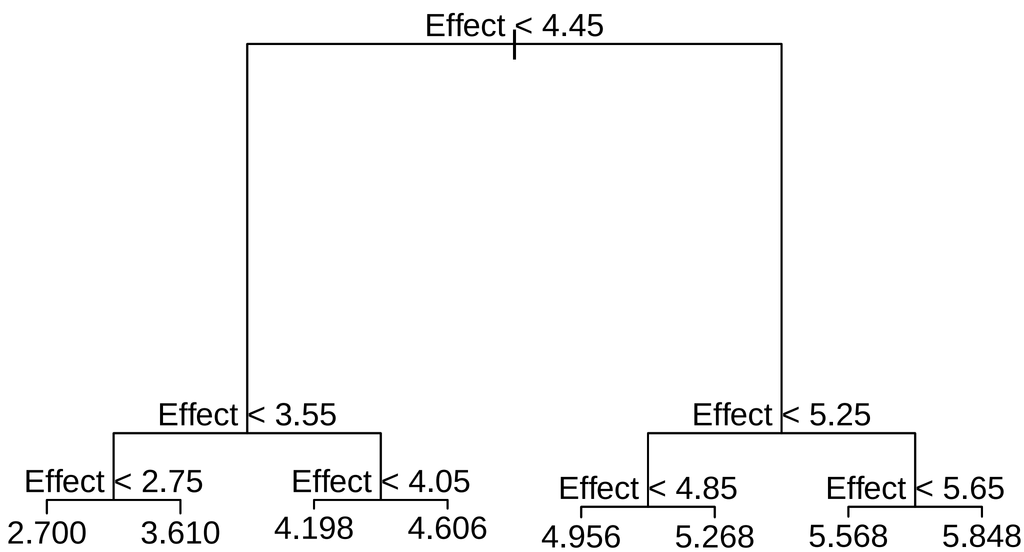
To save space within the machine, I would recommend using the support vector machine since it only requires two very simple predictor variables (Avail and Effect) to predict accurately if a teacher will be "Highly Rated".



**Decision Tree**

A regression tree just splits the predictor space into regions, and uses the average response within each region as the predictor. The regression tree will choose the best predictor variables for the tree and determine the best number of nodes for the model. As we are focus on enhancing predictive ability, we are primarily focused on reducing the RMSE for our model.

$$Y = f(\mathbf{X}) + \epsilon$$




After plotting our regression tree, we can see that the model chose `Effect` as the most important and needed variable with 8 terminal nodes for the tree. This makes sense because if an instructor is effective in teaching for the class then they will have a higher rating. The tree is interesting as it does not take in any other variables which are deemed unimportant by the model. Based on the graph , we can see `Effect` is split into different regions which will give us our `Instr` rating. Based on the thresholds, we can identify that `Effect` has a positive relationship `Instr` which shows lower ratings for one will give a lower rating for the other and vice versa. The RMSE of the regression tree is 0.31 which is not too bad. We can try to reduce the RMSE by pruning the tree but first we can perform a cross-validation to see if our regression tree is already the best.

**Bagged Tree**

$$\hat{f}(\mathbf{X}) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{b}(\mathbf{x})$$

After 500 iterations, the RMSE of the bagged regression tree went down to 0.31, which is not a significant improvement.

## Conclusion

From this project, we can conclude that, based on this data, it is possible to predict whether a new professor will be a “good” professor or not. When analyzing it initially, we quickly found that these predictor variables exemplified the concept of multicolinearity, the occurence of variables sharing trends to the point where they can confuse models. This was observed within many of our plots and we were able to identify general patterns moving forward.

Within the regression model, we were able to achieve a root mean squared error of .25, a value that means the instructor can be predicted reliably to within less than a quarter of a point. This linear regression model was backed up by a lasso model, in which we find variables that can be removed through a “lasso” process that sends some variables effectively to 0. In making this lasso model, we were able to achieve the exact same RMSE, equal to .25. To put into perspective, we interpret a “highly rated professor” as one above a 5.0 on a 6.0 scale. That means that a root mean squared error of .25 is very likely to be accurate enough to determine this.

Furthermore, we developed a logistic regression model to predict if a professor would be classified as “Highly Rated”. They were classified as “Highly Rated” when they had a 5.0 or above on this 6.0 scale. This model developed an accuracy rate of 92%. Checked with a testing and training set of data (.75 split), this model held up and accurately found when a teacher was going to be highly rated.

To add on, we used a support vector machine with only two predictor variables (Effect and Avail) to get a model that tested over 90% accurately. This accurate support vector machine further added to our confidence, indicating we can accurately predict whether a professor will be effective and highly rated with only these two predictor variables.

Finally, we developed a regular and bagged regression tree. This acted as a final check that led to a root mean squared error of approximately .3 on both of them. Our least accurate regression, it was still trustworthy and efficient in its prediction of a teachers instructor rating.

Overall, this project has given us confidence we can accurately predict a professor’s rating and their reception from a random student. In our least accurate regression, we were still relatively accurate, and in our least complex model, we were nearly 92% effective in predicting if a professor was going to be highly rated or not. To sum, we are confident that if we were able to gain some basic data on a professor, such as their availability and effectiveness, we would be able to predict their teacher “rating” and if they would be highly rated.

## Appendix

```
fcq <- read_csv("fcqdata3.csv")

fcq <- fcq %>% rename_all(~gsub(" ", "", .))
rows_to_remove <- c(105946, 105985, 111690)
fcq <- fcq[-rows_to_remove, ]

names(fcq)[names(fcq) == "#Resp"] <- "NumResp"

#view(fcq)
```

```
fcqdata <- na.omit(fcq)

#summary(fcqdata)
```

```
# Filter out non-numeric columns
numeric_columns <- sapply(fcqdata, is.numeric)
numeric_data <- fcqdata[, numeric_columns]

# Calculate correlation matrix
correlation_matrix <- cor(numeric_data)

# Plot correlation matrix
plot_correlation <- corrplot(correlation_matrix, method = "circle", title = "Correlation Matrix")

# Display the plot
#print(plot_correlation)
```

```
fcqnum <- select_if(fcqdata, is.numeric)

fcqnum <- fcqnum %>%
  dplyr::select(-Crse, -SDCrse, -SDInstr)
```

```
set.seed(303)
rows <- sample(1:nrow(fcqnum),size=floor(nrow(fcqnum)*0.75))
train <- fcqnum[rows,]
test <- fcqnum[-rows,]
```

```
linear.mod <- lm(Instr ~ ., data = fcqnum)
summary(linear.mod)
```

```
predictions <- predict(linear.mod, newdata = fcqnum)
rmse <- sqrt(mean((predictions - fcqnum$Instr)^2))
print(paste("RMSE:", rmse))
```

```
# Fit a Lasso regression model
lasso_model <- glmnet(x = x_train, y = y_train, family = "gaussian", alpha = 1)

# Use cross-validation to select the optimal lambda (regularization parameter)
```

```r
cv_fit <- cv.glmnet(x = x_train, y = y_train, family = "gaussian", alpha = 1)

# Extract the optimal lambda
optimal_lambda <- cv_fit$lambda.min

# Refit the Lasso model with the optimal lambda
lasso_model_optimal <- glmnet(x = x_train, y = y_train, family = "gaussian", alpha = 1, lambda = optimal

predictions <- predict(lasso_model_optimal, newx = as.matrix(test[, -ncol(test)]))

rmse <- sqrt(mean((predictions - test$Instr)^2))
print(paste("RMSE:", rmse))
```

```r
fcq.log <- fcq %>%
  mutate(Good = ifelse(Instr >= 5, 1, 0))
```

```r
sum(fcq.log$Good)
```

```r
set.seed(303)
rows <- sample(1:nrow(fcq.log), size = floor(nrow(fcq.log)*.75))
training <- fcq.log[rows,]
testing <- fcq.log[-rows,]
```

```r
logmod <- glm(Good ~ RespRate + Year + Enroll + HrsPerWk + Interest + CrseLvl + Learned + Course + Effe
summary(logmod)
```

```r
predicted <- predict(logmod, newdata = testing, type = "response")

predicted_class <- ifelse(predicted >= 0.6, "Good Professor Rating", "Not Good Professor Rating")

# Create the confusion matrix
conf_matrix <- table(predicted_class, testing$Good)

# Print the confusion matrix
print(conf_matrix)
```

```r
# Predict probabilities on the testing data
predicted_probs <- predict(logmod, newdata = testing, type = "response")

# Create ROC curve
roc_curve <- roc(testing$Good, predicted_probs)

# Plot the ROC curve
plot(roc_curve, main = "ROC Curve for Logistic Model Predicting A 'Highly Rated Professor'", col = "blu

# Add AUC to the plot
auc_value <- round(auc(roc_curve), 2)
text(0.8, 0.2, paste("AUC =", auc_value), col = "blue")
```

```r
predicted_probs <- predict(logmod, newdata = testing, type = "response")
accuracy <- sum(ifelse(predicted_probs >= 0.5, 1, 0) == testing$Good) / length(testing$Good)
```

```r
testing$class_pred <- ifelse(predicted_probs >= 0.6, 1, 0)
```

```r
mean(testing$class_pred != testing$Good)
```

```r
fcqSVM <- data.frame(Effect = fcq.log$Effect, Avail = fcq.log$Avail, Good = fcq.log$Good)
nrows <- sample(1:nrow(fcqSVM),size=floor(nrow(fcqSVM)*0.75))
fcqSVM$Good <- as.factor(fcqSVM$Good)
training2 <- fcqSVM[nrows,]
testing2 <- fcqSVM[-nrows,]
```

```r
svmPoly <- svm(Good~., kernel = "linear", degree = 2, cost = .1, data = training2)
predsPoly <- predict(svmPoly, newdata = testing2)
confusionMatrix(predsPoly, testing2$Good)
```

```r
x1_values <- seq(0, 6, length.out = 100)
x2_values <- seq(0, 6, length.out = 100)
grid <- expand.grid(Effect = x1_values, Avail = x2_values)

# Predict using SVM model
predictions <- predict(svmPoly, newdata = grid)

# Plot
plot(grid$Effect, grid$Avail, type = "n", xlab = "Effect", ylab = "Avail")
points(grid$Effect[predictions == "1"], grid$Avail[predictions == "1"], col = "blue", pch = 20, cex = 0
points(grid$Effect[predictions == "0"], grid$Avail[predictions == "0"], col = "red", pch = 20, cex = 0.

# Add legend
legend("topright", legend = c("Highly Rated Professor: +1", "Not Highly Rated: 0"),
       col = c("blue", "red"), pch = 20, cex = 1.2, bg = "white")
```

```r
model_tree <- tree(Instr ~., data = train)
summary(model_tree)
```

```r
plot(model_tree)
text(model_tree, pretty = 0)
```

```r
preds <- predict(model_tree,newdata = test)
RMSE_tree <- sqrt(mean((test$Instr - preds)^2))
cat(paste("RMSE of Regression Tree:", round(RMSE_tree, 2)))
```

```r
out <- tree(Instr~.,data=train)
# predict on test data and check MSE
pred <- predict(out,newdata=test)
sqrt(mean( (test$Instr - pred)^2 )) # out of sample RMSE
sqrt(mean(summary(out)$resid^2)) # in sample RMSE
```

```r
N <- 500
PRED.boot <- matrix(nr=length(test$Instr),nc=N)

set.seed(303)
```

```r
for(i in 1:N){
  bag.indices <- sample(1:dim(train)[1],size=dim(train)[1],replace=TRUE)
  out <- tree(Instr~.,data=train[bag.indices,])
  PRED.boot[,i] <- predict(out,newdata=test)
}
# average the predictions from the bootstrap-resampled data tree fits
PRED.bagged <- apply(PRED.boot,1,mean)

sqrt(mean( (test$Instr - pred)^2 ))
sqrt(mean( (test$Instr - PRED.bagged)^2 ))
```