02 - Alumni Giving

Nils Gandlau

23 10 2019

Data Preparation

```
str(data)
```

```
## Classes 'data.table' and 'data.frame':   123 obs. of  8 variables:
##  $ ID    : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ School: chr  "Arizona State University" "Arkansas State University<U+0097>Jonesboro" "Auburn Unive
##  $ SFR   : num  24 19 18 8 8 20 20 18 18 14 ...
##  $ LT20  : chr  "42%" "49%" "24%" "74%" ...
##  $ GT50  : chr  "16%" "4%" "17%" "0.10%" ...
##  $ GRAD  : chr  "59%" "37%" "66%" "81%" ...
##  $ FRR   : chr  "81%" "69%" "87%" "88%" ...
##  $ GIVE  : chr  "8%" "11%" "31%" "11%" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ID = col_double(),
##   ..   School = col_character(),
##   ..   SFR = col_double(),
##   ..   LT20 = col_character(),
##   ..   GT50 = col_character(),
##   ..   GRAD = col_character(),
##   ..   FRR = col_character(),
##   ..   GIVE = col_character()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
```

Data Types

Percentages are saved as characters. Reformat those to make them numeric.

```
data <- data %>%
  mutate(
    SFR = SFR,
    LT20 = as.numeric(str_extract(LT20, "[0-9]*")) / 100,
    GT50 = as.numeric(str_extract(GT50, "[0-9]*")) / 100,
    GRAD = as.numeric(str_extract(GRAD, "[0-9]*")) / 100,
    FRR = as.numeric(str_extract(FRR, "[0-9]*")) / 100,
    GIVE = as.numeric(str_extract(GIVE, "[0-9]*")) / 100,
  ) %>%
  setDT()

str(data)
```

```
## Classes 'data.table' and 'data.frame':   123 obs. of  8 variables:
##  $ ID    : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ School: chr  "Arizona State University" "Arkansas State University<U+0097>Jonesboro" "Auburn Unive
##  $ SFR   : num  24 19 18 8 8 20 20 18 18 14 ...
##  $ LT20  : num  0.42 0.49 0.24 0.74 0.95 0.39 0.35 0.28 0.34 0.49 ...
```

```
##  $ GT50  : num  0.16 0.04 0.17 0 0 0.06 0.17 0.18 0.12 0.09 ...
##  $ GRAD  : num  0.59 0.37 0.66 0.81 0.86 0.35 0.6 0.58 0.57 0.71 ...
##  $ FRR   : num  0.81 0.69 0.87 0.88 0.92 0.69 0.79 0.83 0.78 0.85 ...
##  $ GIVE  : num  0.08 0.11 0.31 0.11 0.28 0.15 0.05 0.23 0.11 0.14 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Missing Values

There are no missing values.

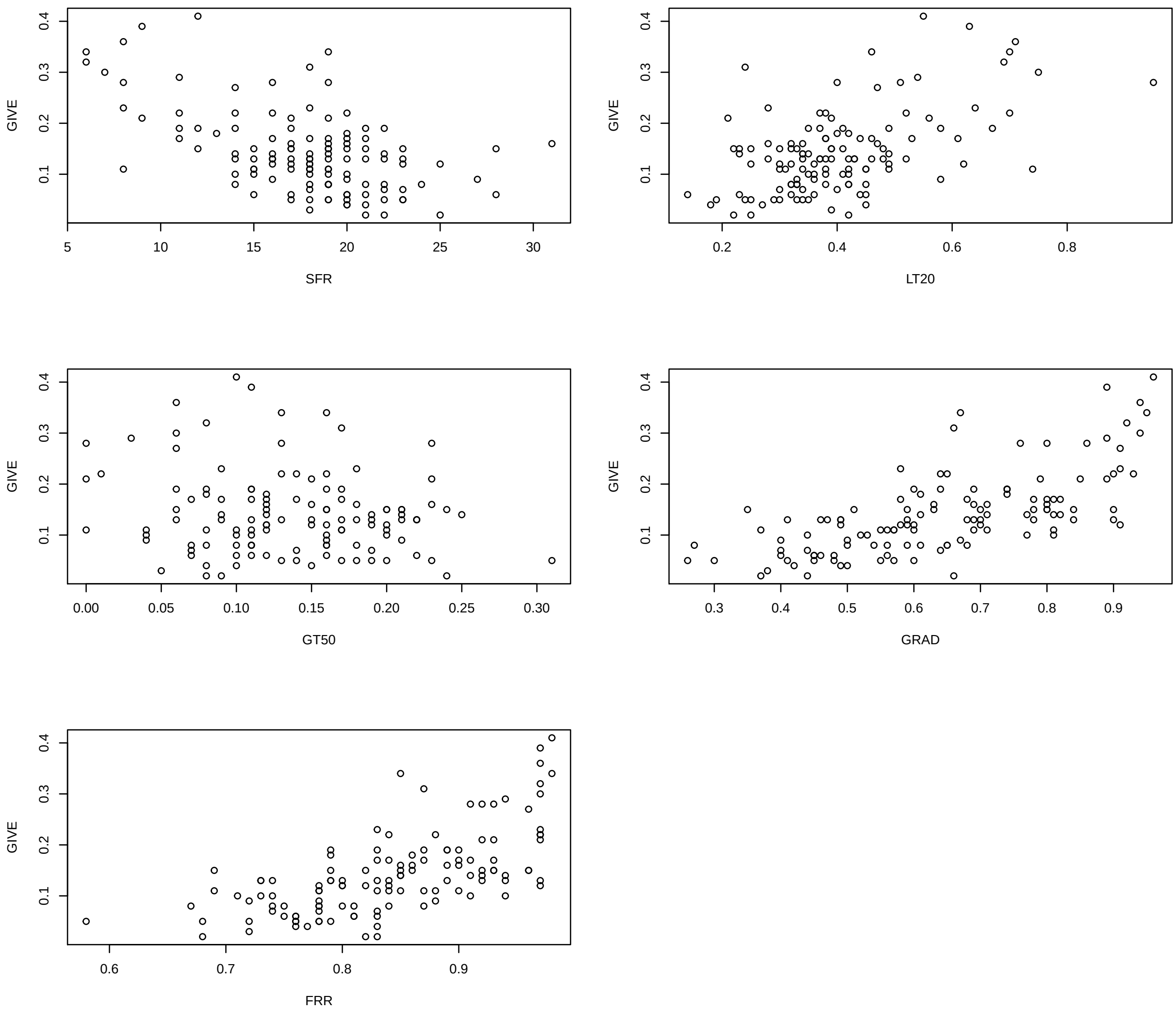
```
nas <- rbindlist(lapply(names(data), function(colname) {
  nMissingValues <- sum(is.na(data[, get(colname)]))
  return(data.table(feature = colname, number_of_nas = nMissingValues))
}))

nas
```

```
##    feature number_of_nas
## 1:      ID             0
## 2:  School             0
## 3:     SFR             0
## 4:    LT20             0
## 5:    GT50             0
## 6:    GRAD             0
## 7:     FRR             0
## 8:    GIVE             0
```

Data Exploration

```
par(mfrow = c(4, 2))
plot(GIVE ~ ., data %>% select(-ID, -School))
```



One may identify the following trends:

- The larger the student-to-faculty ratio, the lower the giving rate
- The larger the share of classes with fewer than 20 students, the larger the giving rate
- The larger the share of classes with more than 20 students, the slightly lower the giving rate
- The larger the average six-year graduation rate, the larger the giving rate
- The more freshmen stay, the larger the giving rate

Correlations of (numeric) features

```
numericFeatures <- c("GIVE", "SFR", "LT20", "GT50", "GRAD", "FRR")
dataNumericFeatures <- data[, c(numericFeatures), with = F]

corMatrix <- round(cor(dataNumericFeatures), 2)
corMatrix
```

```
##       GIVE   SFR  LT20  GT50  GRAD   FRR
## GIVE  1.00 -0.55  0.54 -0.18  0.68  0.65
## SFR  -0.55  1.00 -0.69  0.41 -0.60 -0.52
## LT20  0.54 -0.69  1.00 -0.58  0.49  0.38
## GT50 -0.18  0.41 -0.58  1.00  0.02  0.06
## GRAD  0.68 -0.60  0.49  0.02  1.00  0.93
## FRR   0.65 -0.52  0.38  0.06  0.93  1.00
```

- `GRAD` and `FRR` have high (positive) correlation 0.93. We may want to avoid using both in our models.
- `SFR` and `LT20` have medium-high (negative) correlation -0.69. We also note that the two variables have, when increased, an *opposing effect* on the target variable. Hence, when both were used in a regression model, their regression coefficients may be biased/untrustworthy due to multicolinearity.

Regression

We deploy a standard linear regression, but exclude `FRR` since it is highly correlated with `GRAD`.

```
linReg1 <- lm(GIVE ~ SFR + LT20 + GT50 + GRAD, data = data)
summary(linReg1)
```

```
##
## Call:
## lm(formula = GIVE ~ SFR + LT20 + GT50 + GRAD, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.134447 -0.035869 -0.009671  0.025116  0.187546
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.052494   0.058295  -0.900   0.3697
## SFR         -0.001024   0.001789  -0.573   0.5681
## LT20         0.130407   0.063132   2.066   0.0411 *
## GT50        -0.046532   0.119255  -0.390   0.6971
## GRAD         0.257573   0.043415   5.933 3.05e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05659 on 118 degrees of freedom
## Multiple R-squared:  0.5241, Adjusted R-squared:  0.5079
## F-statistic: 32.49 on 4 and 118 DF,  p-value: < 2.2e-16
```

We check whether including `FRR` over `GRAD` is better:

```
linReg2 <- lm(GIVE ~ SFR + LT20 + GT50 + FRR, data = data)
summary(linReg2)
```

```
##
## Call:
```

```
## lm(formula = GIVE ~ SFR + LT20 + GT50 + FRR, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12488 -0.03985 -0.00619  0.03272  0.18563
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.296045   0.084897  -3.487 0.000687 ***
## SFR         -0.001661   0.001736  -0.957 0.340564
## LT20         0.180157   0.060654   2.970 0.003606 **
## GT50         0.013807   0.114480   0.121 0.904207
## FRR          0.466932   0.078185   5.972 2.53e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0565 on 118 degrees of freedom
## Multiple R-squared:  0.5255, Adjusted R-squared:  0.5094
## F-statistic: 32.67 on 4 and 118 DF,  p-value: < 2.2e-16
```

- We note that in the second regression, the R^2 is slightly lower, hence we will use the first one, where we included `GRAD` and not `FRR`

Question 1

If School A's graduation rate `GRAD` is 10 (percentage) points higher (note that 10 percentage points equals 0.1 in our case, since we converted the percentages to be numbers in $[0, 1]$) than School B's, then School A can expect to have an *alumni giving rate* that is $0.257573 * 0.1 * 100 \approx 2.6$ percentage points higher than School B's.

The calculation is based on the following idea:

$$\begin{aligned}
\Delta GIVE &= (w_0 + w_1 GRAD_2 + \dots) - (w_0 + w_1 GRAD_1 + \dots) \\
&= w_1 \cdot (GRAD_2 - GRAD_1) \\
&= w_1 \cdot 0.1
\end{aligned}$$

Question 2

The answer doesn't change compared to Question 1.

Since we have assumed for each predictor a linear relationship with the target variable, the benefit of a marginal increase in `GRAD` is not dependent on the (absolute) level of any of the predictors.

That is, regardless of the value of student-to-faculty ratio, our model says that increasing the graduation rate by 10 percentage points (or 0.1 in our case), always increases the expected giving rate by 2.6 percentage points.

Question 3

```r
maxGIVE <- max(data$GIVE)
minGIVE <- min(data$GIVE)

rbindlist(list(
  data[GIVE == maxGIVE, c("School", "GIVE")],
  data[GIVE == minGIVE, c("School", "GIVE")]
))
```

```
##                           School GIVE
## 1:    University of Notre Dame 0.41
## 2:  San Diego State University 0.02
## 3:   San Jose State University 0.02
## 4: University of South Alabama 0.02
```

- University of Notre Dame has the most impressive giving rate (41%)
- Three universities share the last place: San Diego State University, San Jose State University, University of South Alabama. Each with a giving rate of 2%.

Question 4

School's feature vector:

- `GRAD = 0.67`
- `SFR = 17`
- `LT20 = 0.34`
- `GT50 = 0.23`
- `FRR = 0.77`

Note that in our regressions before, we haven't included both `GRAD` and `FRR` at the same time, since they were highly correlated. Hence, we have to run a new regression:

```r
linReg <- lm(GIVE ~ SFR + LT20 + GT50 + GRAD + FRR, data = data)
summary(linReg)
```

```
##
## Call:
## lm(formula = GIVE ~ SFR + LT20 + GT50 + GRAD + FRR, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12894 -0.03651 -0.00910  0.03164  0.18632
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.195063   0.114015  -1.711   0.0898 .
## SFR         -0.001102   0.001782  -0.619   0.5374
## LT20         0.150867   0.064396   2.343   0.0208 *
## GT50        -0.031469   0.119150  -0.264   0.7922
## GRAD         0.129653   0.098093   1.322   0.1888
## FRR          0.256999   0.176922   1.453   0.1490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05633 on 117 degrees of freedom
## Multiple R-squared:  0.5325, Adjusted R-squared:  0.5125
## F-statistic: 26.65 on 5 and 117 DF,  p-value: < 2.2e-16
```

Given the school's feature vector, we can compute the expected giving rate that is predicted by our model:

$$\begin{aligned} GRAD &= w_0 + w_1 \cdot SFR + w_2 \cdot LT20 + w_3 \cdot GT50 + w_4 \cdot GRAD + w_5 \cdot FRR \\ &\approx -0.1951 - 0.0011 \cdot 17 + 0.1509 \cdot 0.34 - 0.0315 \cdot 0.23 + 0.1297 \cdot 0.67 + 0.2570 \cdot 0.77 \\ &= 0.11505 \end{aligned}$$

So our model predicts a graduation rate of 11.505 %, which is greater than 8%.