Case Study 2 - Alumni Giving

Jonathan Ratschat / Franziska Bülck

22.10.2019

Preparing dataset

```
Data <- read.csv("Data-Alumni-Giving.csv")
str(Data)
```

```
## 'data.frame':    123 obs. of  8 variables:
##  $ ID    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ School: Factor w/ 123 levels "Arizona State University",..: 1 2 3 57 58 61 64 65 4 5 ...
##  $ SFR   : int  24 19 18 8 8 20 20 18 18 14 ...
##  $ LT20  : Factor w/ 49 levels "14%","18%","19%",..: 24 31 7 47 49 21 17 10 16 31 ...
##  $ GT50  : Factor w/ 27 levels "0.10%","0%","1%",..: 10 22 11 1 2 24 11 12 6 27 ...
##  $ GRAD  : Factor w/ 55 levels "26%","27%","30%",..: 25 5 31 43 47 4 26 24 23 36 ...
##  $ FRR   : Factor w/ 31 levels "58%","67%","68%",..: 15 4 21 22 26 4 13 17 12 19 ...
##  $ GIVE  : Factor w/ 31 levels "10%","11%","12%",..: 30 2 20 2 16 6 27 14 2 5 ...
```

Description of variables

SFR (Student-to-faculty ratio): number of students who attend a university divided by the number of teachers; the lower the ratio, the better

LT20 (Percentage of classes with fewer than 20 students): the higher the percentage, the better

GT50 (Percentage of classes with greater than 50 students): the lower the percentage, the better

GRAD (Average six-year graduation rate): the higher the percentage, the better

FRR (Freshman retention rate): number of freshmen in a college or university who return for their sophomore year; the higher the rate, the better

GIVE (Average alumni giving rate): the higher the rate, the better

Transformation of variables

```
#Transform ID from integer to factor
Data$ID <- as.character(Data$ID)

#Vectors in Data containing percentages are saved as factors. Transformation to numeric data
Data$LT20 <- as.numeric(sub("%", "",Data$LT20,fixed=TRUE))/100
Data$GT50 <- as.numeric(sub("%", "",Data$GT50,fixed=TRUE))/100
Data$GRAD <- as.numeric(sub("%", "",Data$GRAD,fixed=TRUE))/100
Data$FRR <- as.numeric(sub("%", "",Data$FRR,fixed=TRUE))/100
Data$GIVE <- as.numeric(sub("%", "",Data$GIVE,fixed=TRUE))/100

#Control data structures
str(Data)
```

```
## 'data.frame':    123 obs. of  8 variables:
##  $ ID    : chr  "1" "2" "3" "4" ...
##  $ School: Factor w/ 123 levels "Arizona State University",..: 1 2 3 57 58 61 64 65 4 5 ...
##  $ SFR   : int  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 0.001 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 ...
```

```
summary(Data)
```

```
##       ID                                               School
##  Length:123         Arizona State University              :  1
##  Class :character   Arkansas State University\x97Jonesboro:  1
##  Mode  :character   Auburn University                     :  1
##                     Ball State University                 :  1
##                     Baylor University                     :  1
##                     Boise State University                :  1
##                     (Other)                               :117
##       SFR             LT20             GT50             GRAD
##  Min.   : 6.00   Min.   :0.1400   Min.   :0.0000   Min.   :0.2600
##  1st Qu.:16.00   1st Qu.:0.3200   1st Qu.:0.0950   1st Qu.:0.5050
##  Median :18.00   Median :0.3800   Median :0.1300   Median :0.6400
##  Mean   :17.77   Mean   :0.4037   Mean   :0.1363   Mean   :0.6452
##  3rd Qu.:20.00   3rd Qu.:0.4600   3rd Qu.:0.1800   3rd Qu.:0.7850
##  Max.   :31.00   Max.   :0.9500   Max.   :0.3100   Max.   :0.9600
##
##       FRR              GIVE
##  Min.   :0.5800   Min.   :0.0200
##  1st Qu.:0.7800   1st Qu.:0.0800
##  Median :0.8400   Median :0.1300
##  Mean   :0.8411   Mean   :0.1418
##  3rd Qu.:0.9100   3rd Qu.:0.1700
##  Max.   :0.9800   Max.   :0.4100
##
```

Create regression model

```
mod1 <- lm(GIVE ~ SFR + LT20 + GT50 + GRAD + FRR, data = Data)
summary(mod1)
```

```
##
## Call:
## lm(formula = GIVE ~ SFR + LT20 + GT50 + GRAD + FRR, data = Data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.128940 -0.036513 -0.009093  0.031618  0.186341
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.194952   0.114016  -1.710   0.0899 .
## SFR         -0.001099   0.001781  -0.617   0.5383
## LT20         0.150652   0.064400   2.339   0.0210 *
## GT50        -0.032219   0.119220  -0.270   0.7874
## GRAD         0.129830   0.098096   1.324   0.1882
## FRR          0.256903   0.176920   1.452   0.1492
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05632 on 117 degrees of freedom
## Multiple R-squared:  0.5325, Adjusted R-squared:  0.5125
## F-statistic: 26.66 on 5 and 117 DF,  p-value: < 2.2e-16
```

Use of a stepwise regression model to find more convincing model. The function stepAIC chooses the best model by Akaike information criterion (AIC). AIC is an estimator of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection.

```r
#install.packages("MASS")
library(MASS)

#Fit the full model
full.model <- lm(GIVE ~ SFR + LT20 + GT50 + GRAD, data = Data)

#Stepwise regression model
step.model <- stepAIC(full.model, direction = "both", trace = FALSE)
summary(step.model)
```

```
##
## Call:
## lm(formula = GIVE ~ LT20 + GRAD, data = Data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.130092 -0.032907 -0.006848  0.026963  0.191287
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.09281    0.02113  -4.392 2.43e-05 ***
## LT20         0.16457    0.04357   3.777 0.000248 ***
## GRAD         0.26064    0.03436   7.587 7.75e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05627 on 120 degrees of freedom
## Multiple R-squared:  0.5216, Adjusted R-squared:  0.5136
## F-statistic: 65.41 on 2 and 120 DF,  p-value: < 2.2e-16
```

LT20 and GRAD are highly significant. Therefore, the 0 Hypothesis can be rejected. F-statistic is highly significant as well.

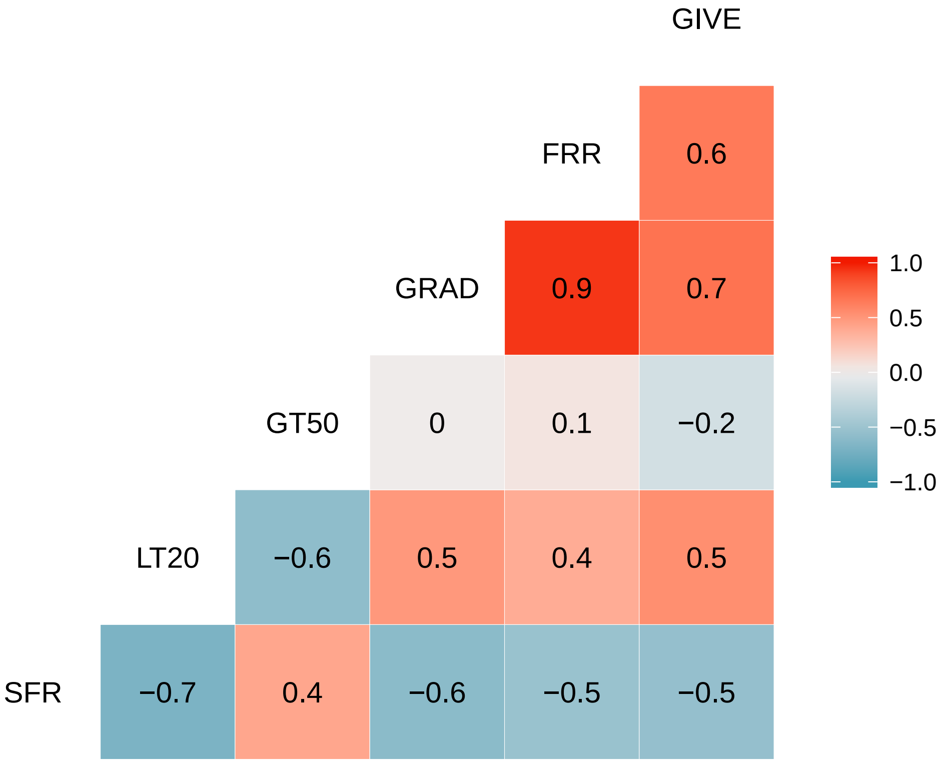
Test for Multicollinearity

Multicollinearity is when there is correlation between predictors (i.e. independent variables) in a model. Its presence can adversely affect our regression results.

```
#install.packages("GGally")
library(GGally)

#Create correlation matrix
ggcorr(Data, label = TRUE)
```

```
## Warning in ggcorr(Data, label = TRUE): data in column(s) 'ID', 'School' are
## not numeric and were ignored
```



GRAD highly correlates with GIVE, while SFR, LT20, and FRR moderately correlate with GIVE. GT50 has little correlation with GIVE.

SFR and LT20 highly correlate. If FRR would be included in regression model, it could indicate multicollinearity.

FRR and GRAD highly correlate. If FRR would be included in regression model, it could indicate multicollinearity.

GRAD and LT20 moderately correlate, therefore, we assume that multicollinearity is not considered as a problem in step.model.

In the next step, we test the variance inflation factor (VIF). The VIF estimates how much the variance of a regression coefficient is inflated due to multicollinearity in the model. It quantifies the severity of multicollinearity in an OLS regression analysis.

```
#install.packages("HH")
library(HH)

#Calculate VIF per independent variable in model
vif(step.model)
```

```
##     LT20     GRAD
## 1.311322 1.311322
```

```r
#Calculate mean of VIFs
mean(vif(step.model))
```

```
## [1] 1.311322
```

An average mean between 1 and 5 indicates a moderate correlated result. In general, a VIF above 10 indicates high correlation and is cause for concern. Some authors suggest a more conservative level of 2.5 or above. Since our average VIF is under 2.5, we assume that multicollinearity is not considered as a problem in step.model.

Test for Autocorrelation

Autocorrelation means that the residuals correlate with each other. Such correlation is most common in time series analyses, when the independent variables do not adequately cover the cyclical fluctuations in the time series.

We do not excpect autocorrelation to be a problem, but still perform a Durbin-Watson-Test.

```r
#install.packages("lmtest")
library(lmtest)

dwtest(step.model, data = Data)
```

```
##
##  Durbin-Watson test
##
## data:  step.model
## DW = 2.28, p-value = 0.9402
## alternative hypothesis: true autocorrelation is greater than 0
```

Autocorrelation between variables is not an issue.

Test for Homoscedasticity

Homoscedasticity means that the residuals do not have a constant variance. We need to check if residuals are normally distributed.

```r
shapiro.test(rstandard(step.model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(step.model)
## W = 0.95307, p-value = 0.0003028
```

Shapiro-Wilk-Test of standardized residuals is significant. Residuals differ significantly from normal distribution; assumption of homoscedasticity is violated.

The absence of a normal distribution signifies only that the F- and t-tests are not meaningfully applicable. The estimated regression coefficients are still unbiased.

Expected value of zero for the residuals $E(e_i) = 0$

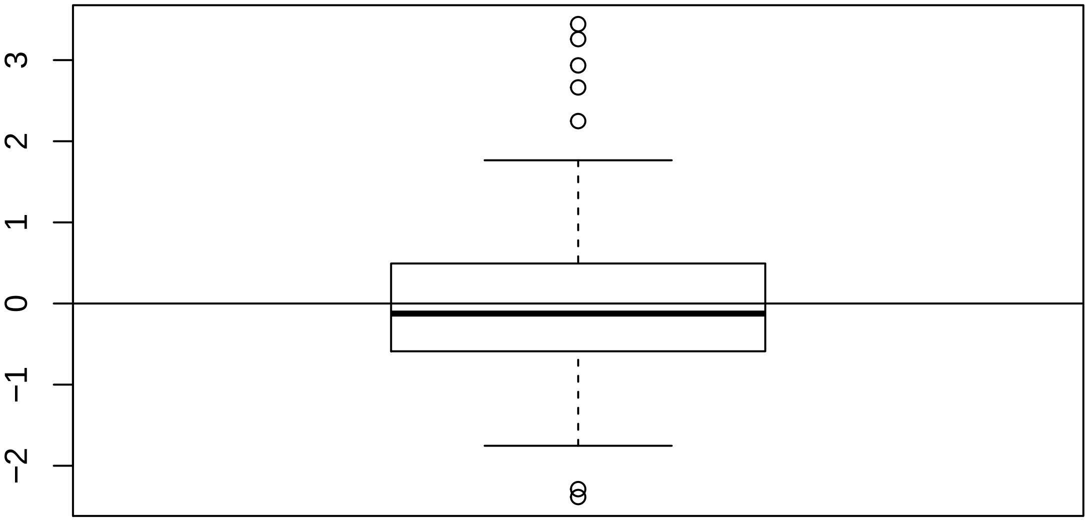
```
wilcox.test(rstandard(step.model))
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  rstandard(step.model)
## V = 3466, p-value = 0.3818
## alternative hypothesis: true location is not equal to 0
```

Zero hypothesis cannot be rejected.

The standardized residual is the residual divided by its standard deviation. We use it to show the difference of mean of residuals from 0 graphically.

```
boxplot(rstandard(step.model))
abline(h=0)
```



Thoughts on Endogeneity

Endogeneity means that one or more of the independent variables correlate with the residuals and therefore influences the relationship between the dependent variable and the independent variable. Its existence leads to a systematic distortion of the estimated regression coefficient. It can occur for several reasons, such as an omitted independent variable, simultaneity in the variables, measurement error in an independent variable, autocorrelation with delayed dependent variable, or selfselection.

In our point of view, we should keep in mind that the independent variables of the model could not be the only explainatory variables of GIVE. Besides the variables LT20 and GRAD, we could also include variables like the commuting time, to which extent classes are taught by professors, how great the campus life is, or how strong the researchers of the university are.

Selection bias

The dataset contains the schools that fielded football teams in the Football Bowl Subdivision (the highest competitive division for U.S. college football). This could lead to problem in the dataset because the included schools were not randomly selected.

Questions

School A‘s graduation rate is 10 points higher than school B‘s. How much higher do we expect A‘s giving rate to be?

Regression formula: Predicted GIVE = Intercept + LT20 coefficient * LT20(i) + GRAD coefficient * GRAD(i)

If the other variables are kept constant, then with every increase of GRAD by 100 points, GIVE increases by 0.26064. Therefore, when GRAD is increased by 10 points, then GIVE increases by 0.026064 or 2.6064%.

```
0.26064*1/10
```

```
## [1] 0.026064
```

Since we do not know the other independent variables of school A and school B, we can only say that GRAD of school A leads to an increase in GIVE of 2.6064pps relative to school B’s GRAD rate. School B could still have a higher GIVE percentage in total.

How does the answer to question 1 change if we learn that A and B have identical student-to-faculty ratios?

Our answer does not change since the SFR is not included in our step.model.

Which of the 123 schools has the most (least) impressive giving rate?

In our opinion, an impresse giving rate is a giving rate that substantially exceeds the predicted giving rate from step.model. We chose a relative approach dividing the difference of predicted and actual GIVE with actual GIVE.

```
Data$PredictedGIVE <- predict(step.model, Data)
Data$RelDifGIVE <- (Data$GIVE - Data$PredictedGIVE)/Data$PredictedGIVE
```

Most impressive schools:

```
head(Data[order(-Data$RelDifGIVE) ,],n=3)
```

```
##      ID                             School SFR LT20 GT50 GRAD  FRR GIVE
## 110 110 University of Texas\x97San Antonio  23 0.24 0.23 0.26 0.58 0.05
## 3     3                  Auburn University  18 0.24 0.17 0.66 0.87 0.31
## 11   11             Boise State University  21 0.33 0.11 0.27 0.67 0.08
##     PredictedGIVE RelDifGIVE
## 110    0.01445553   2.458884
## 3      0.11871270   1.611347
## 11     0.03187287   1.509971
```

Least impressive schools:

```r
tail(Data[order(-Data$RelDifGIVE) ,],n=3)
```

```
##    ID                       School SFR LT20 GT50 GRAD  FRR GIVE
## 86 86   San Jose State University  25 0.25 0.09 0.44 0.82 0.02
## 90 90 University of South Alabama  22 0.42 0.08 0.37 0.68 0.02
## 85 85  San Diego State University  21 0.22 0.24 0.66 0.83 0.02
##    PredictedGIVE RelDifGIVE
## 86    0.06301691 -0.6826249
## 90    0.07274808 -0.7250787
## 85    0.11542138 -0.8267219
```

Consider a school similar to ours (i.e., one with the following characteristics): We have a 67% graduation rate and a student-faculty ratio of 1:17, 34% of the classes have fewer than 20 students, 23% of the classes have more than 50 students, and we have a freshman retention rate of 77%. Should this school's giving rate be greater than or less than 8%?

For our model, we only need GRAD and LT20 to predict GIVE.

Predicted Give = Intercept + LT20 coefficient * LT20(i) + GRAD coefficient * GRAD(i)

```r
#Upload data in the same format as Data
NewSchool <- read.csv("New school.csv",sep = ";")

#Show NewSchool
NewSchool
```

```
##    ID         School SFR LT20 GT50 GRAD  FRR GIVE
## 1 124 Similar school  17 0.34 0.23 0.67 0.77 0.08
```

```r
#Predict GIVE using step.model
NewSchool$PredictedGIVE <- predict(step.model, NewSchool)

#Inspect data
NewSchool
```

```
##    ID         School SFR LT20 GT50 GRAD  FRR GIVE PredictedGIVE
## 1 124 Similar school  17 0.34 0.23 0.67 0.77 0.08     0.1377757
```

The school's giving rate should be greater than 8% since step.models predict a GIVE of 13.78%. Therefore, our school underperforms as it in reality only receives 8%.