DATA621 Final Project Proposal

Andrew Bowen, Glen Davis, Josh Forster, Shoshana Farber, Charles Ugiagbe

2023-10-14

Introduction:

Our dataset comes from OpenData's - School Quality Report for NYC high schools between 2013 and 2014. Our final project will potentially include more up-to-date educational information and geographic data to augment our analysis.

First, let's read in our source CSV file. This is posted in our GitHub repository in the interest of reproducibility.

```r
df <- read.csv("https://raw.githubusercontent.com/andrewbowen19/businessAnalyticsDataMiningDATA621/main/
```

There are several predictor variables of interest to us. The NYC School Quality Report from the Department of Education included ratings for schools in the following categories. We would like to see how effective these ratings are at predicting student success:

- *Quality Review Rating*
- *Achievement Rating*
- *Environment Rating*
- *College and Career Readiness Rating*

In addition, it would be interesting to see whether these ratings could be replaced by proxy variables. Not having to assign ratings, and instead being able to ascertain school quality from other information the Department of Education (DOE) already knows about schools could save the DOE time.

Our response variable will be the average student's SAT score at a given school. SAT scores are an imperfect metric given their correlation with other socioeconomic factors, but for our purposes, they can serve as an imperfect benchmark to measure academic performance.

Main research question: Do these DOE ratings accurately predict whether a school will foster high academic performance in students, and are there other proxy variables that can be used to more accurately predict academic performance (measured in SAT scores)?

Data Cleaning:

```r
# Renaming some dataframe variables
df$math_score_8 <- df$Average.Grade.8.Math.Proficiency
df$english_score_8 <- df$Average.Grade.8.English.Proficiency
df$avg_sat_score <- df$Average.SAT.Score

# Make predictor rating variables factors
```

```r
levels_ach_rat <- c("N/A", "Not Meeting Target", "Approaching Target",
                    "Meeting Target", "Exceeding Target")
levels_qual_rev_rat <- c("N/A", "Underdeveloped", "Developing", "Proficient",
                         "Well Developed")
df$Achievement.Rating <- factor(df$Achievement.Rating,
                                levels=levels_ach_rat)
df$Quality.Review.Rating <- factor(df$Quality.Review.Rating,
                                   levels=levels_qual_rev_rat)
df$Environment.Rating <- factor(df$Environment.Rating,
                                levels=levels_ach_rat)
df$College.and.Career.Readiness.Rating <- factor(
    df$College.and.Career.Readiness.Rating, levels=levels_ach_rat)
```

Exploratory Data Analysis:

First, let's plot the counts of schools receiving all possible different rating combinations. This should give us a sense whether schools with a rating for one category tend to have a similar rating for other categories. That is, schools with high *Quality Review Rating* values might be expected to have high *Achievement Ratings*, and this plot will highlight whether expectations for these variables match reality for all possible buckets.

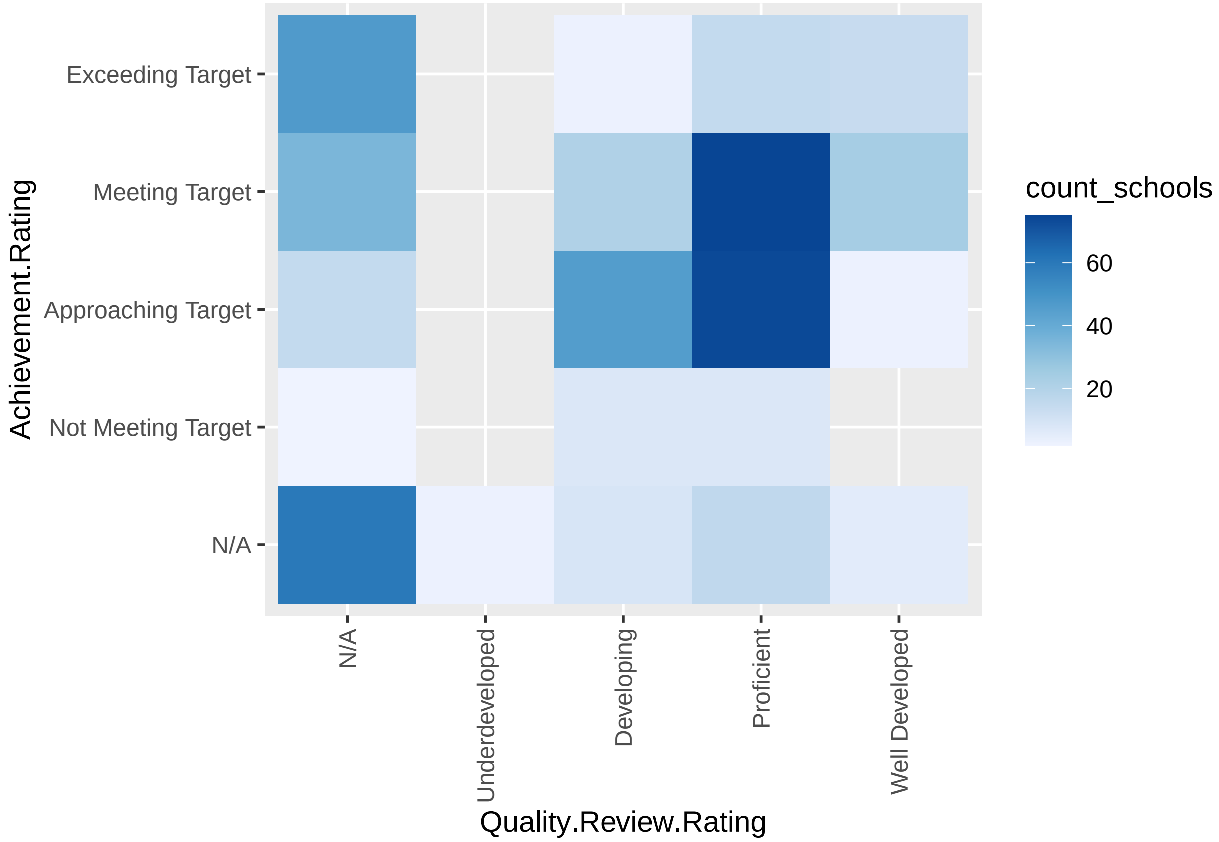
```r
# Group schools by ratings
ratings <- df %>% group_by(Achievement.Rating, Quality.Review.Rating) %>% summarise(count_schools=n())
```

```
## ‘summarise()‘ has grouped output by ’Achievement.Rating’. You can override
## using the ‘.groups‘ argument.
```

```r
head(ratings, 5)
```

```
## # A tibble: 5 x 3
## # Groups:   Achievement.Rating [1]
##   Achievement.Rating Quality.Review.Rating count_schools
##   <fct>              <fct>                         <int>
## 1 N/A                N/A                              60
## 2 N/A                Underdeveloped                    3
## 3 N/A                Developing                        9
## 4 N/A                Proficient                       16
## 5 N/A                Well Developed                    6
```

```r
ggplot(ratings, aes(x=Quality.Review.Rating,
                    y=Achievement.Rating,
                    fill=count_schools)) +
    geom_tile() +
    scale_fill_distiller(palette = "Blues", direction = 1) +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



Modeling:

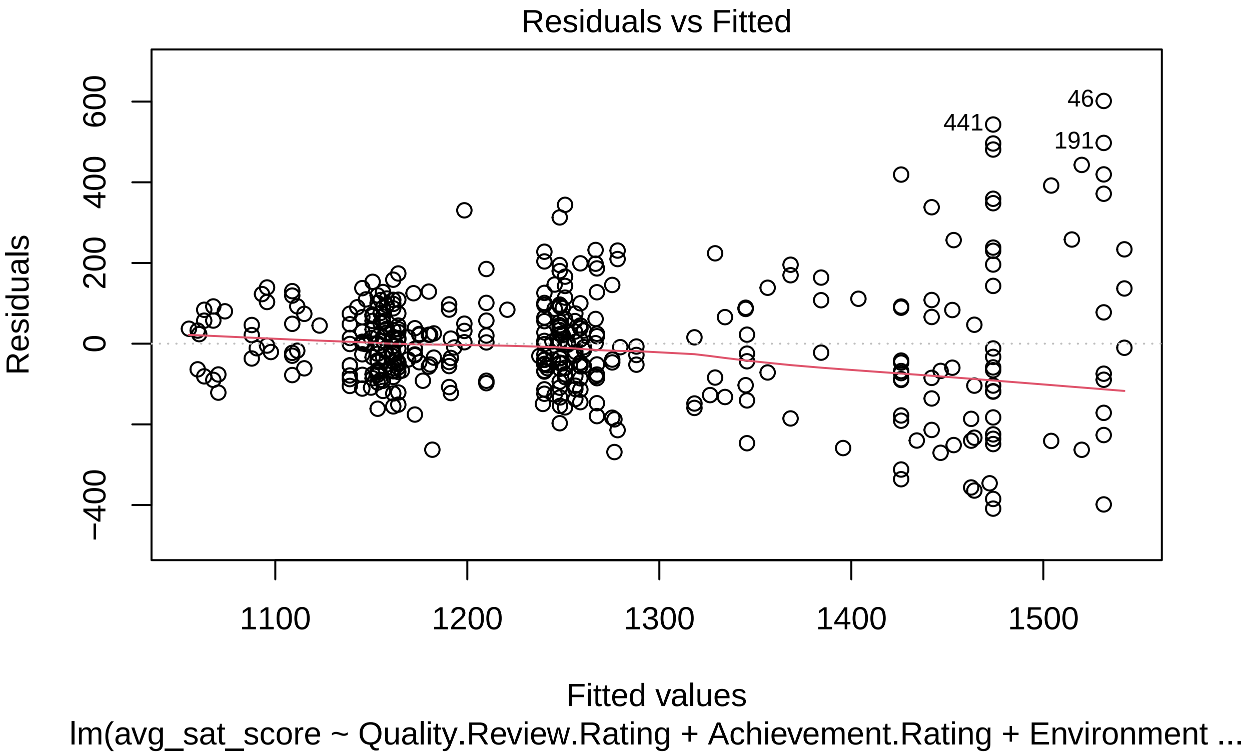
First, let's create a basic linear model between our rating variables, and our dependent variable (Average SAT Score of a school)

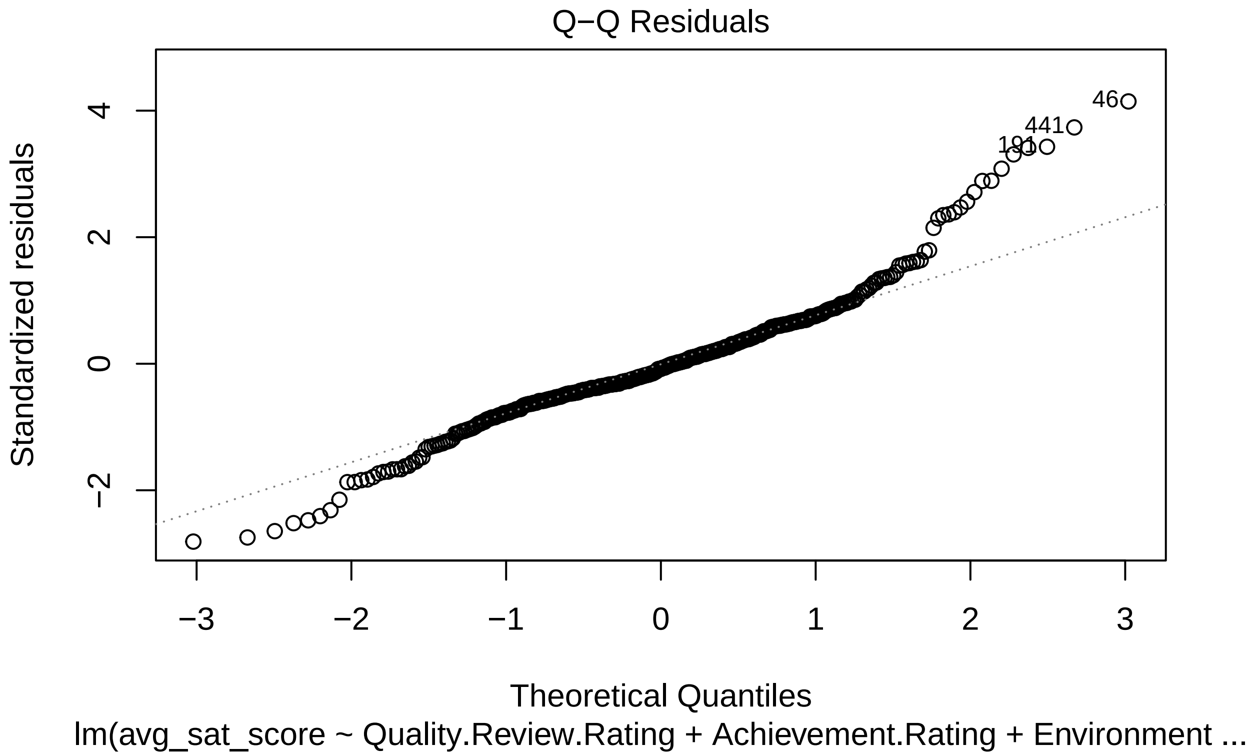
```
lm_ratings <- lm(avg_sat_score ~ Quality.Review.Rating +
  Achievement.Rating +
  Environment.Rating +
  College.and.Career.Readiness.Rating, df)
summary(lm_ratings)
```

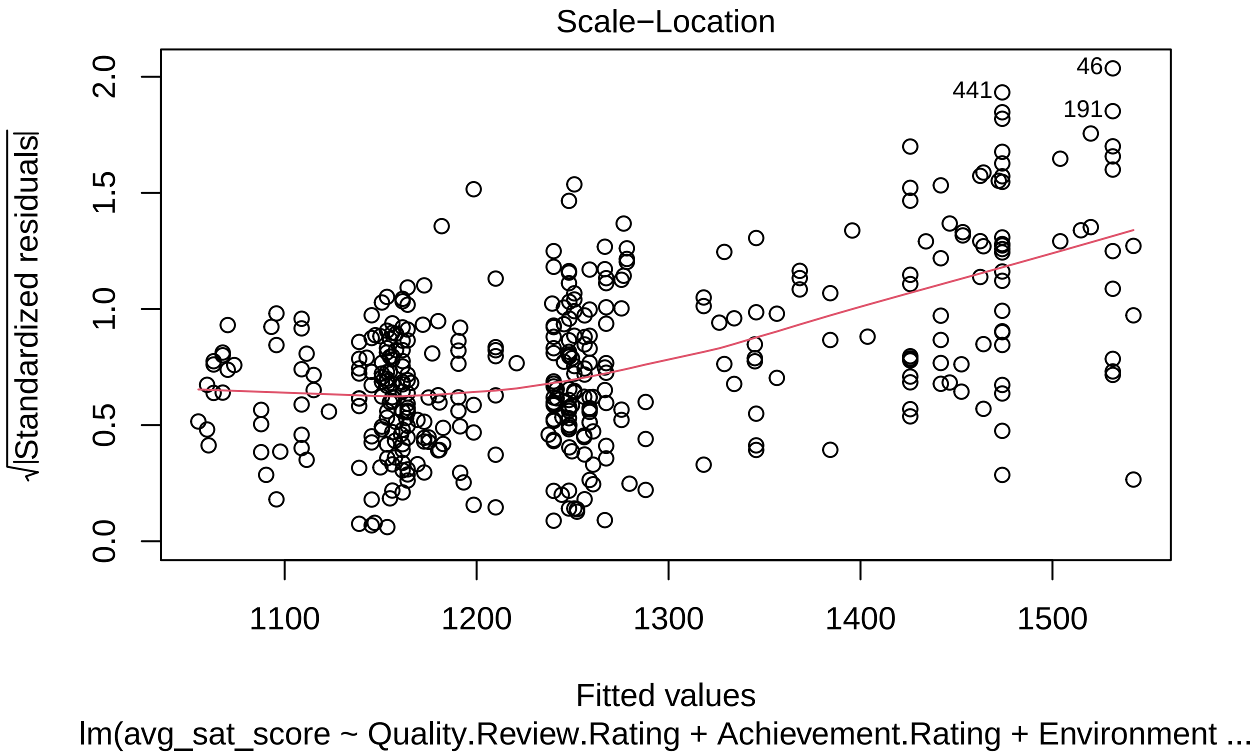
```
##
## Call:
## lm(formula = avg_sat_score ~ Quality.Review.Rating + Achievement.Rating +
##     Environment.Rating + College.and.Career.Readiness.Rating,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -408.85  -76.50  -11.34   74.20  601.57
##
## Coefficients: (2 not defined because of singularities)
##                                                          Estimate Std. Error
## (Intercept)                                               1138.80      37.23
## Quality.Review.RatingUnderdeveloped                      36.20      92.72
## Quality.Review.RatingDeveloping                         -29.98      25.60
## Quality.Review.RatingProficient                         -27.36      20.76
## Quality.Review.RatingWell Developed                     -11.46      27.65
## Achievement.RatingNot Meeting Target                    261.46      63.20
## Achievement.RatingApproaching Target                    264.88      47.15
## Achievement.RatingMeeting Target                        256.91      44.62
## Achievement.RatingExceeding Target                      335.05      42.54
## Environment.RatingNot Meeting Target                     61.77      41.60
## Environment.RatingApproaching Target                     68.35      23.76
## Environment.RatingMeeting Target                         57.58      21.92
## Environment.RatingExceeding Target                          NA         NA
## College.and.Career.Readiness.RatingNot Meeting Target  -370.92      50.26
## College.and.Career.Readiness.RatingApproaching Target  -280.61      26.14
## College.and.Career.Readiness.RatingMeeting Target      -185.79      22.40
## College.and.Career.Readiness.RatingExceeding Target        NA         NA
##                                                       t value Pr(>|t|)
## (Intercept)                                             30.585  < 2e-16 ***
## Quality.Review.RatingUnderdeveloped                      0.390  0.69646
## Quality.Review.RatingDeveloping                         -1.171  0.24231
## Quality.Review.RatingProficient                         -1.318  0.18820
## Quality.Review.RatingWell Developed                     -0.414  0.67877
## Achievement.RatingNot Meeting Target                     4.137 4.33e-05 ***
## Achievement.RatingApproaching Target                     5.617 3.74e-08 ***
## Achievement.RatingMeeting Target                         5.757 1.76e-08 ***
## Achievement.RatingExceeding Target                       7.875 3.56e-14 ***
## Environment.RatingNot Meeting Target                     1.485  0.13844
## Environment.RatingApproaching Target                     2.876  0.00425 **
## Environment.RatingMeeting Target                         2.628  0.00895 **
## Environment.RatingExceeding Target                          NA       NA
## College.and.Career.Readiness.RatingNot Meeting Target  -7.380 9.96e-13 ***
## College.and.Career.Readiness.RatingApproaching Target -10.735  < 2e-16 ***
## College.and.Career.Readiness.RatingMeeting Target      -8.294 1.89e-15 ***
## College.and.Career.Readiness.RatingExceeding Target        NA       NA
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 147.1 on 382 degrees of freedom
##   (87 observations deleted due to missingness)
## Multiple R-squared:  0.4206, Adjusted R-squared:  0.3993
## F-statistic:  19.8 on 14 and 382 DF,  p-value: < 2.2e-16
```

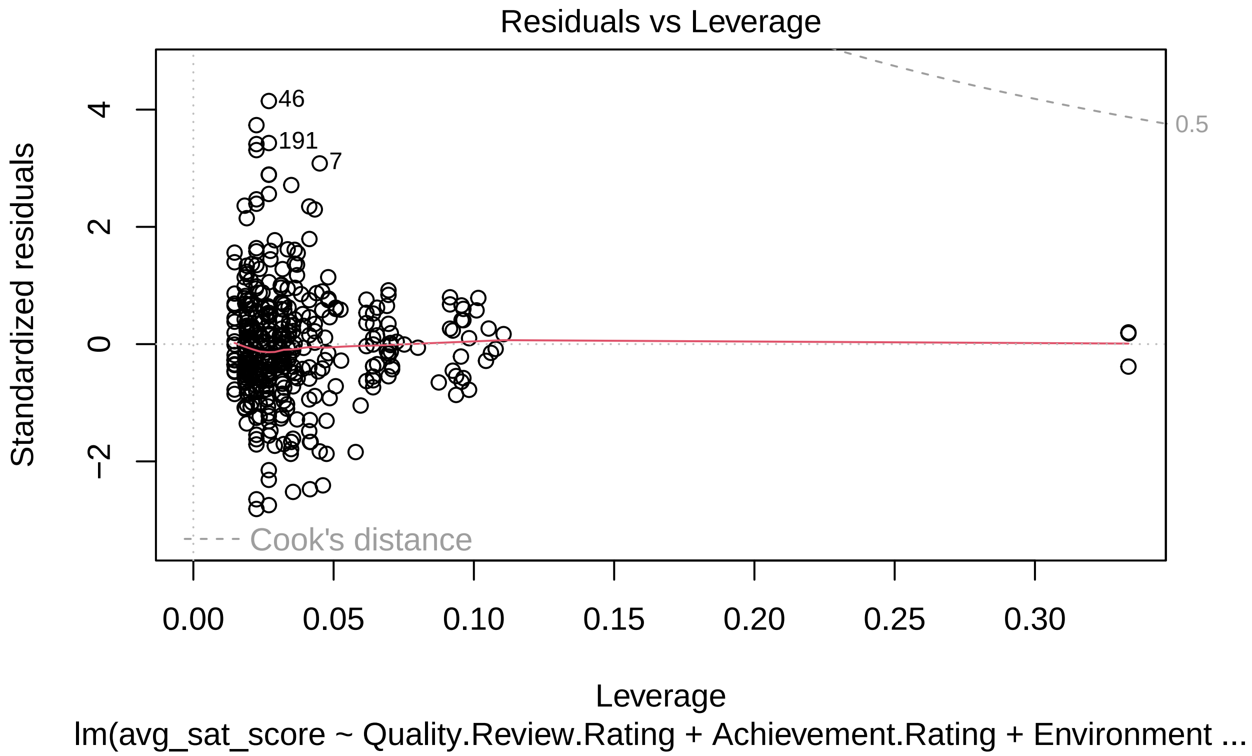
Let’s plot our ratings-only model

```
plot(lm_ratings)
```


Residuals vs Fitted
Residuals
Fitted values
lm(avg_sat_score ~ Quality.Review.Rating + Achievement.Rating + Environment ...
600
400
200
0
−400
1100
1200
1300
1400
1500
46
441
191


Q−Q Residuals
Standardized residuals
4
2
0
−2
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(avg_sat_score ~ Quality.Review.Rating + Achievement.Rating + Environment ...
46
441
191


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
46
441
191
1100
1200
1300
1400
1500
Fitted values
lm(avg_sat_score ~ Quality.Review.Rating + Achievement.Rating + Environment ...



There seems to be some pattern in our residuals vs fitted, as well as some tail behavior in our QQ plot indicating this may not be an ideal fit.

Outside of the DOE ratings, we can create some very basic linear models to help us identify potential predictor variables. Two variables seem like they could potentially be useful, as past academic performance seems like it would correlate tightly with future success:

- `Average.Grade.8.Math.Proficiency`
- `Average.Grade.8.English.Proficiency`

```
lm_english_math <- lm(Average.SAT.Score~ math_score_8 + english_score_8, df)
summary(lm_english_math)
```

```
##
## Call:
## lm(formula = Average.SAT.Score ~ math_score_8 + english_score_8,
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -144.146  -35.601   -1.924   33.716  187.616
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         141.15      17.12   8.246 2.52e-15 ***
## math_score_8        237.08      19.27  12.303  < 2e-16 ***
## english_score_8   228.96      19.28  11.873  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 54.44 on 391 degrees of freedom
##   (90 observations deleted due to missingness)
## Multiple R-squared:  0.9182, Adjusted R-squared:  0.9178
## F-statistic:  2195 on 2 and 391 DF,  p-value: < 2.2e-16
```

On the surface, our adjusted R^2 for the model based solely on past performance is significantly better than our model using the school's DOE ratings (`lm_ratings`). While it may seem like comparing apples to oranges

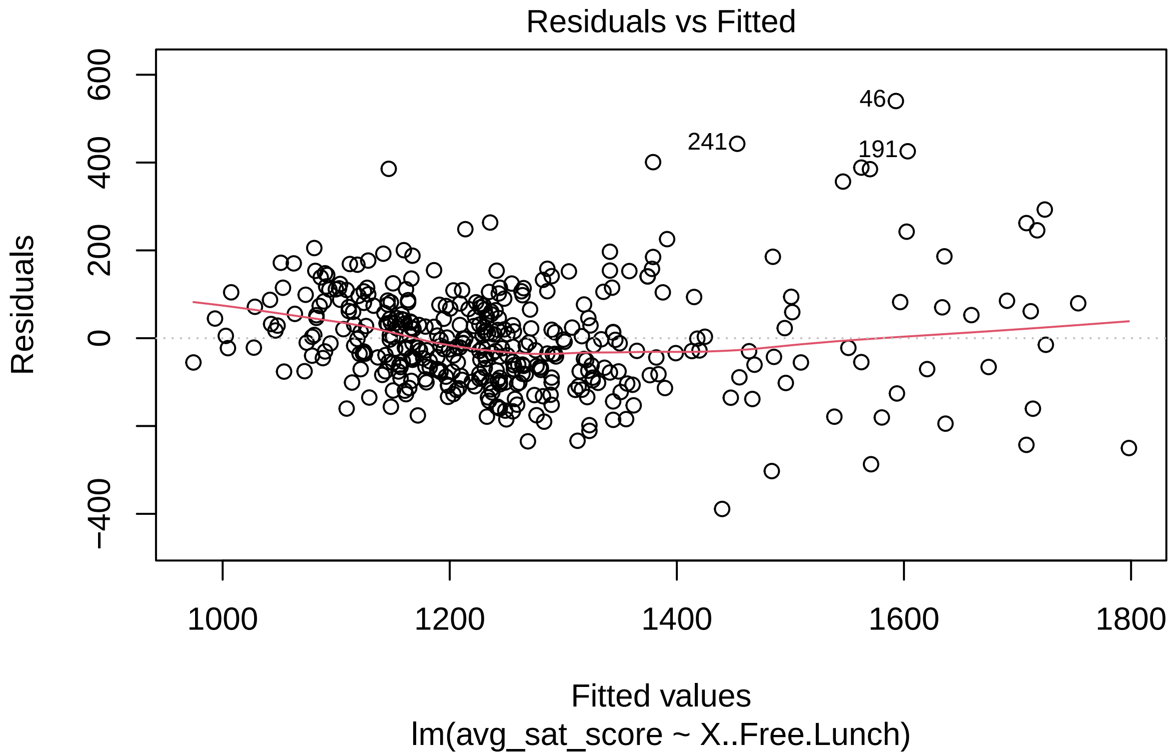
For instance, we see a jump in R^2 as well by using a model solely fit to predict average SAT score from the percentage of students who receive free lunch.

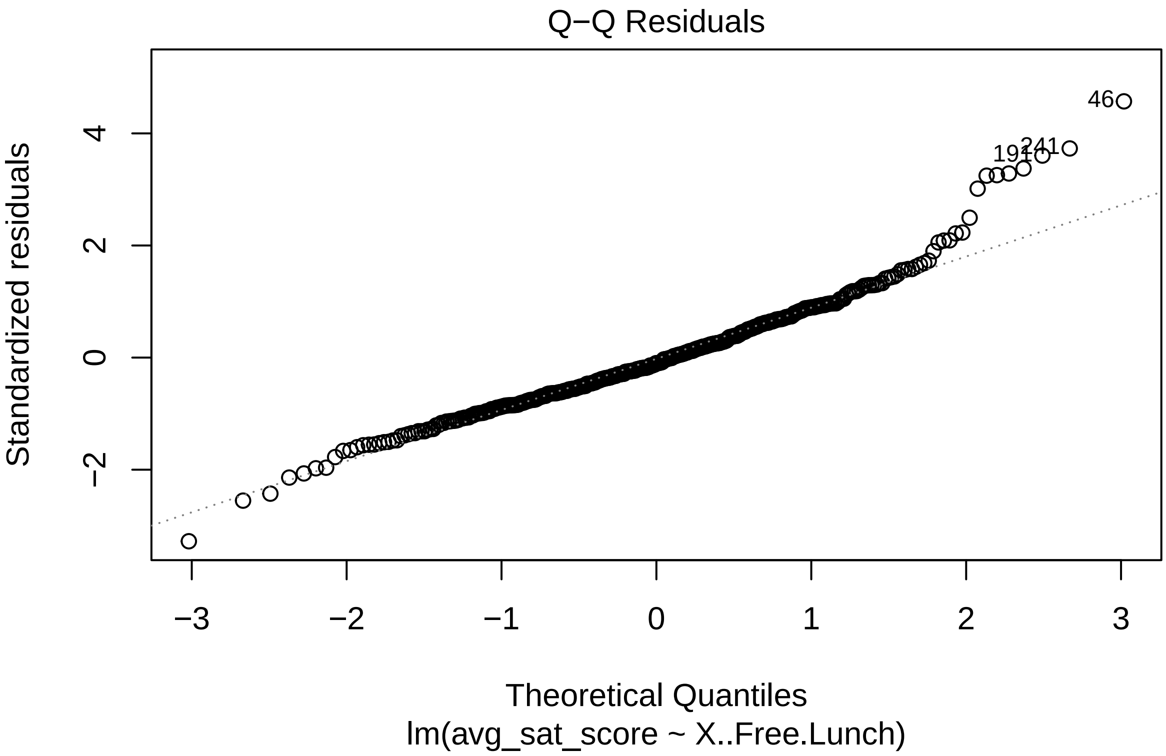
```
lm_free_lunch <- lm(avg_sat_score ~ X..Free.Lunch, df)
summary(lm_free_lunch)
```

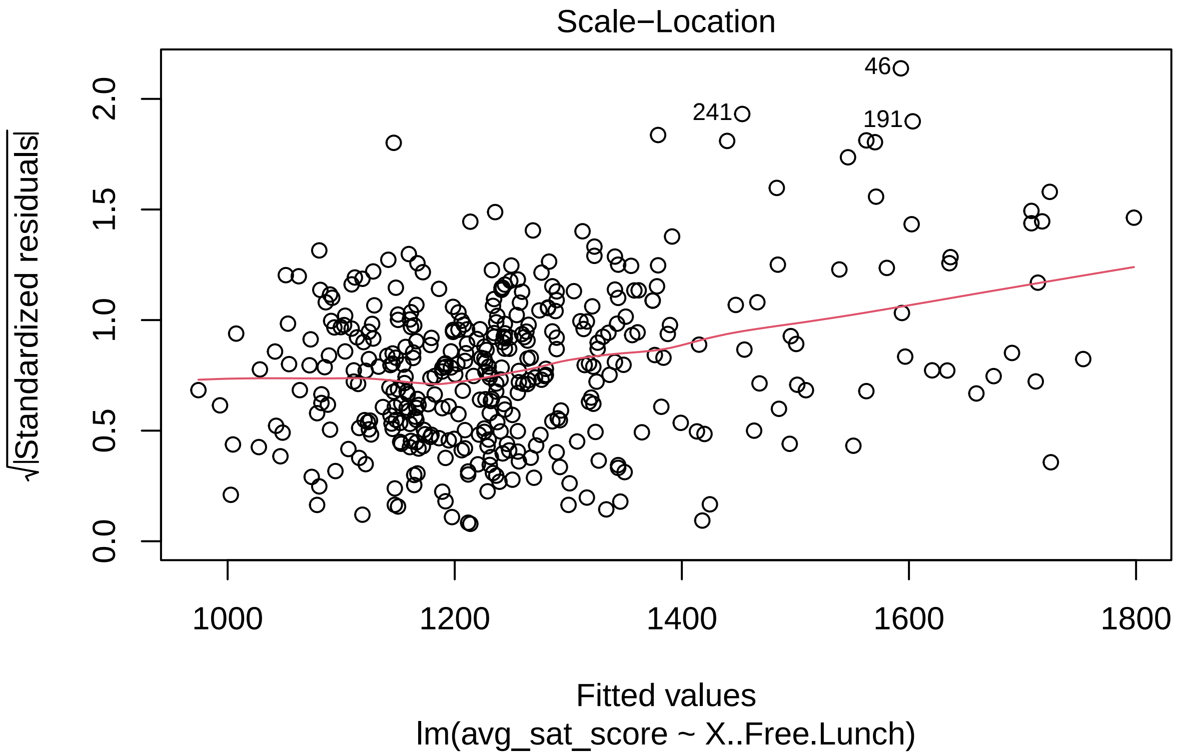
```
##
## Call:
## lm(formula = avg_sat_score ~ X..Free.Lunch, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -388.89  -75.78  -11.90   70.18  540.12
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1916.92      27.54   69.60   <2e-16 ***
## X..Free.Lunch  -950.25      38.57  -24.64   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 119.1 on 392 degrees of freedom
##   (90 observations deleted due to missingness)
## Multiple R-squared:  0.6076, Adjusted R-squared:  0.6066
## F-statistic:   607 on 1 and 392 DF,  p-value: < 2.2e-16
```

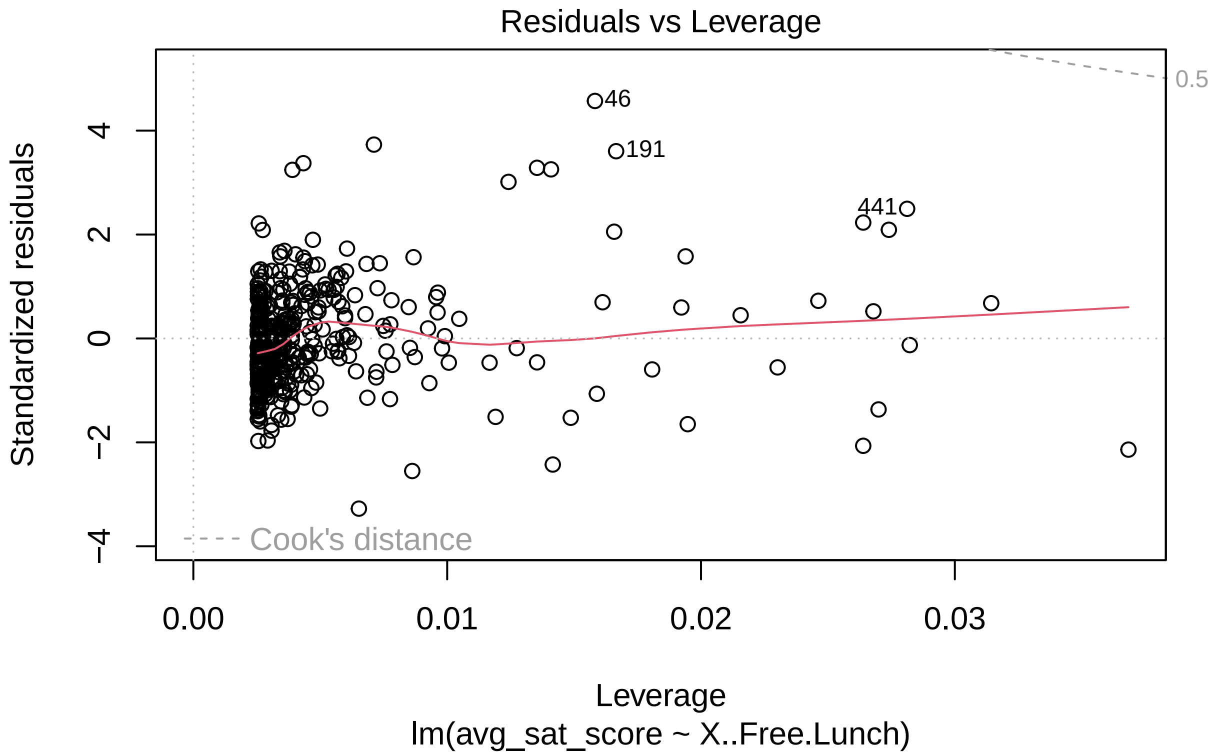
We can plot this free lunch model as well:

```
plot(lm_free_lunch)
```


Residuals vs Fitted
Residuals
600
400
200
0
−400
1000
1200
1400
1600
1800
Fitted values
lm(avg_sat_score ~ X..Free.Lunch)
46
241
191


Q−Q Residuals
Standardized residuals
4
2
0
−2
46
191
241
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(avg_sat_score ~ X..Free.Lunch)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
46
241
191
1000
1200
1400
1600
1800
Fitted values
lm(avg_sat_score ~ X..Free.Lunch)



This Free Lunch plot isn't ideal, but better behaved than our Rating model plot from above.

Going further in model evaluation, we can use other metrics of interest when evaluating a model: Root mean squared error (RMSE - calculated via the `modelr` package) or AIC, which measures model performance with an added penalty for overly complex models (more input params).

```r
# Calculate RMSE and AIC for both models
rmse_fl <- rmse(lm_free_lunch, df)
rmse_ratings <- rmse(lm_ratings, df)
print(glue("Free Lunch Model RMSE: {rmse_fl}"))
```

```
## Free Lunch Model RMSE: 118.795653099892
```

```r
print(glue("Ratings Model RMSE: {rmse_ratings}"))
```

```
## Ratings Model RMSE: 144.276661079985
```

```r
aic_fl <- AIC(lm_free_lunch)
aic_ratings <- AIC(lm_ratings)
print(glue("Free Lunch Model AIC: {aic_fl}"))
```

```
## Free Lunch Model AIC: 4888.71855932368
```

```
print(glue("Ratings Model AIC: {aic_ratings}"))
```

```
## Ratings Model AIC: 5106.19297013857
```

From these simple evaluation metrics, we can see the free-lunch only model performs a bit better, however, we can likely find a more optimized fit across our variable space in the source dataset.

Further Work

Below are some additional modeling points to consider for the final project, as well as to augment our dataset for our final project:

- SAT Scores are not a perfect indicator of future academic []. Using other response variables could paint a more complete picture on the educational variables that most impact outcomes
- Include other educational outcome metrics (job placement rates on graduation, income by school) joined to our data
- Fortunately, NYC's DBN (*District-Borough Number*) system allows for easier joining to other education datasets posted on NYC Open Data
- Use a more recent dataset than 2013-2014. NYC Open Data is an excellent tool for this