Summer Programs' Effect on Poverty-Adjusted Performance

Kyle Joecken

March 21, 2016

Abstract

We look at the effect of summer and food programs on the performance of Oregon schools after controlling for the effects of poverty.

Controlling for Poverty

First, we need to load the data into data frames and clean what we have. We'll read poverty data in first, dropping rows of schools on which we don't have complete data (many of these are Jails/Juvenile facilities or EI/ECSE programs):

```
# Read into a variable from csv, keeping only columns we care about
poverty <- read.csv("../data/FreeReducedLunch.csv", stringsAsFactors = FALSE)
poverty <- poverty[, c(1, 5, 8, 11, 14, 17)]
# Eliminate incomplete cases
poverty <- poverty[complete.cases(poverty), ]
head(poverty)
```

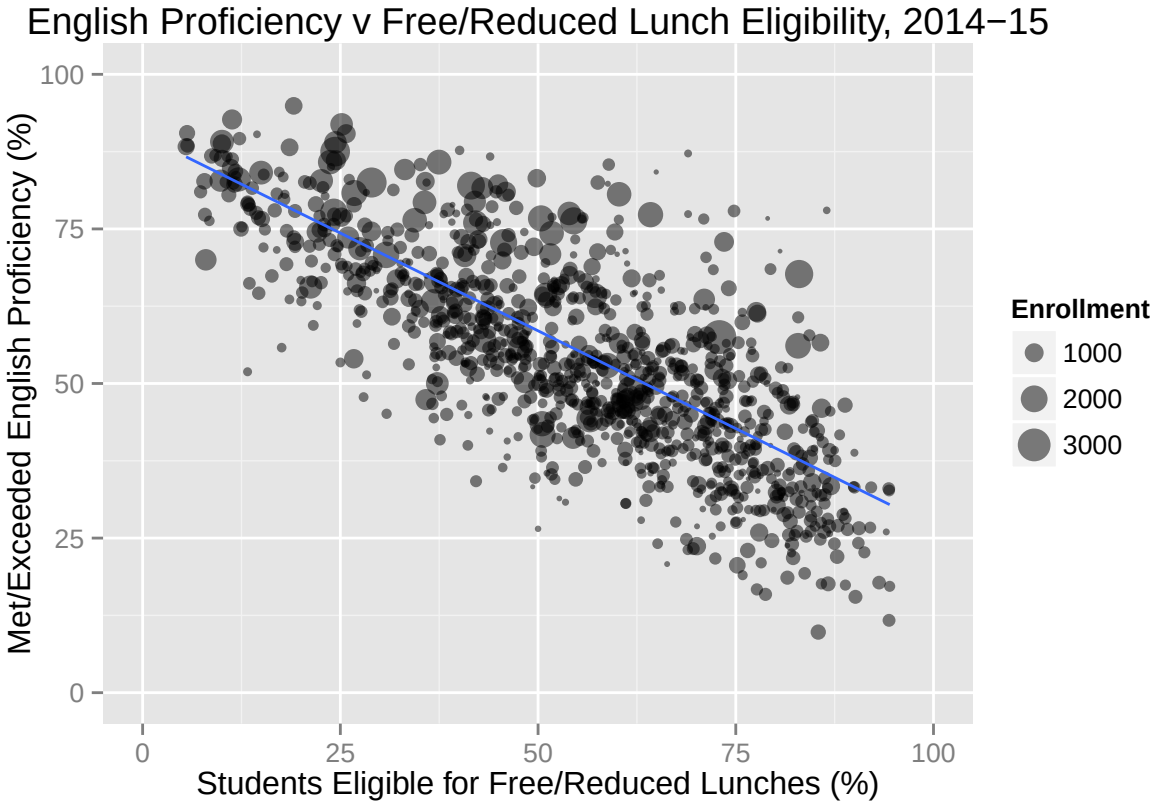
```
##   SchoolID Enrollment FreeLunches ReducedLunches EligibleStudents
## 3      707        186         100             25              125
## 4       17         80          39              7               46
## 5       16         92          65             10               75
## 7     1210        291         104             27              131
## 8     1208        353         160             24              184
## 9     1209        198          92             21              113
##   PercentEligible
## 3           67.20
## 4           57.50
## 5           81.52
## 7           45.02
## 8           52.12
## 9           57.07
```

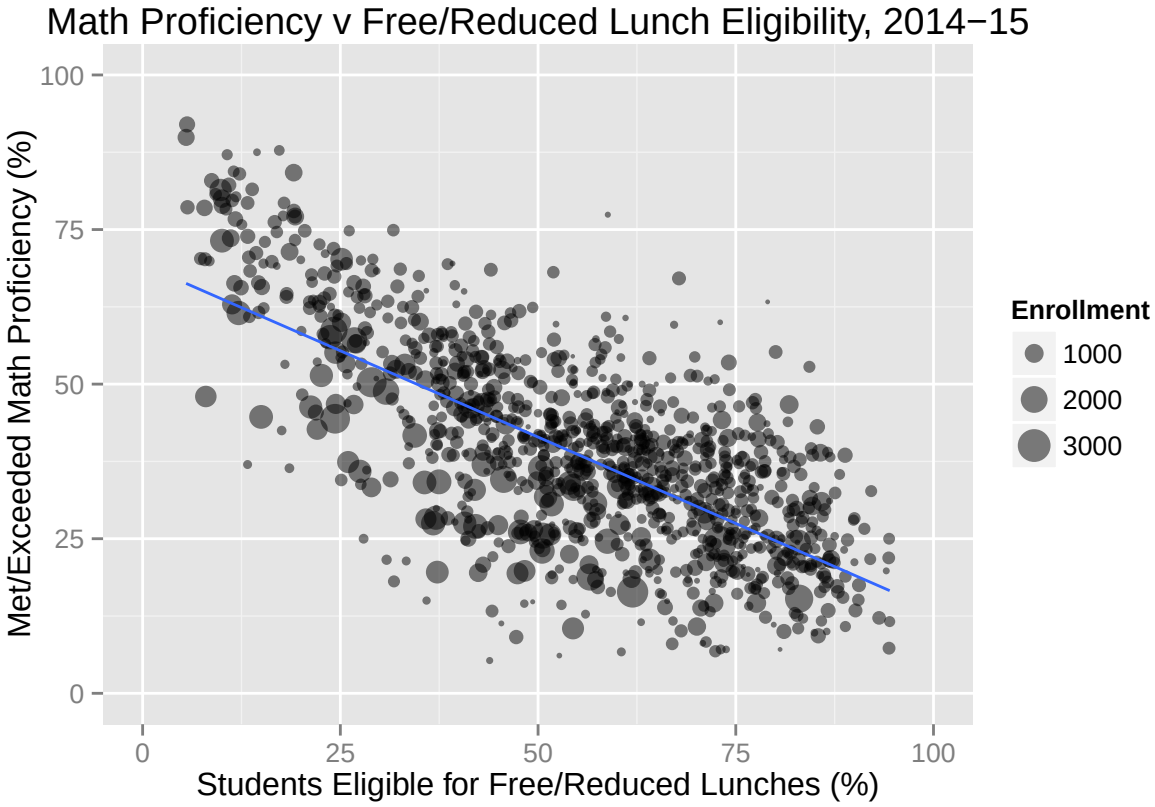
We next load performance data on English and Mathematics. These data will need to be heavily massaged to get them into a format we can proceed with.

```
# Read into a variable from csv, keeping only columns we care about
performance <- read.csv("../data/Performance.csv", stringsAsFactors = FALSE)
performance <- performance[, c(3, 6, 7, 9, 10)]
# Shorten terms, column names
performance$Subject[performance$Subject == "English Language Arts"] <- "Eng"
performance$Subject[performance$Subject == "Mathematics"] <- "Math"
performance$Subgroup[performance$Subgroup == "American Indian/Alaskan Native"] <- "NaAmer"
performance$Subgroup[performance$Subgroup == "Black/African American"] <- "Black"
performance$Subgroup[performance$Subgroup == "Econo. Disadvantaged"] <- "EcoDis"
performance$Subgroup[performance$Subgroup == "Extended Assessment"] <- "ExAsmt"
performance$Subgroup[performance$Subgroup == "Hispanic/Latino"] <- "HisLat"
performance$Subgroup[performance$Subgroup == "Indian Education"] <- "IndEd"
performance$Subgroup[performance$Subgroup == "Limited English Proficient (LEP)"] <- "LimEng"
performance$Subgroup[performance$Subgroup == "Migrant Education"] <- "MigEdu"
performance$Subgroup[performance$Subgroup == "Multi-Racial"] <- "Multi"
performance$Subgroup[performance$Subgroup == "Pacific Islander"] <- "PacIsl"
performance$Subgroup[performance$Subgroup == "Students with Disabilities (SWD)"] <- "SWD"
performance$Subgroup[performance$Subgroup == "SWD with Accommodations"] <- "SWDAcc"
performance$Subgroup[performance$Subgroup == "Talented and Gifted (TAG)"] <- "TAG"
performance$Subgroup[performance$Subgroup == "Total Population"] <- "Total"
names(performance)[4:5] <- c("Part", "Met")
# Clean up data, cast as numeric
performance$Part <- sub("[<>] ([0-9]{1,2}.[0-9])%", "\\1", performance$Part)
performance$Part[performance$Part %in% c("*", "-")] <- NA
performance$Part <- as.numeric(performance$Part)
performance$Met <- sub("[<>] ([0-9]{1,2}.[0-9])%", "\\1", performance$Met)
performance$Met[performance$Met %in% c("*", "-")] <- NA
performance$Met <- as.numeric(performance$Met)
# Reshape into one row per school, one column per Group_Subject_Metric combo
performance <- recast(performance, SchoolID ~ Subgroup + Subject + variable, id.var = 1:3)
# subset to overall data, schools that have compete data
overall <- performance[, c(1, 66:69)]
overall <- overall[complete.cases(overall), ]
# merge in poverty data via percent eligible for free/reduced lunches
overall <- merge(poverty[, c(1:2, 6)], overall)
names(overall) <- sub("Total_", "", names(overall))
head(overall)
```

```
##   SchoolID Enrollment PercentEligible Eng_Part Eng_Met Math_Part Math_Met
## 1        1        245           53.88     98.4    51.5      98.4     36.4
## 2        2        465           70.11     99.2    48.3     100.0     51.3
## 3        4        113           61.06     98.2    71.4      98.2     60.7
## 4        7        314           67.52     98.7    40.1      98.7     25.6
## 5        8        499           44.69     96.2    81.0      95.4     44.8
## 6       15        184           58.15     99.0    62.6     100.0     51.0
```

Let's have a look at the data as collected and see if a pattern emerges.





In order to remove the effect that our measure of poverty has on proficiency, we simply subtract the expected proficiency rates from the reported ones to get a *residual* proficiency rate (or a performance rate relative to expectation).

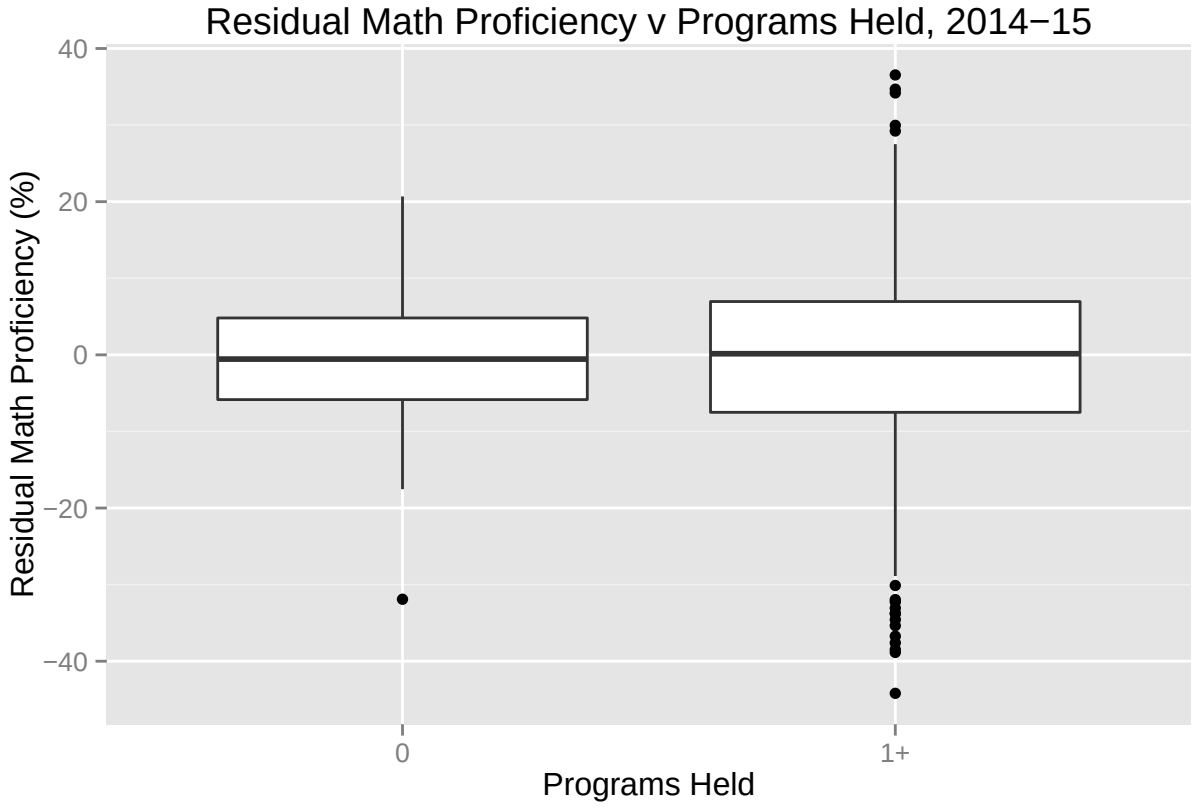
```
fit_eng <- lm(Eng_Met ~ Enrollment + PercentEligible, data = overall, weights = Enrollment)
overall$ResEng <- overall$Eng_Met - predict(fit_eng, overall)
fit_math <- lm(Math_Met ~ Enrollment + PercentEligible, data = overall, weights = Enrollment)
overall$ResMath <- overall$Math_Met - predict(fit_math, overall)
```

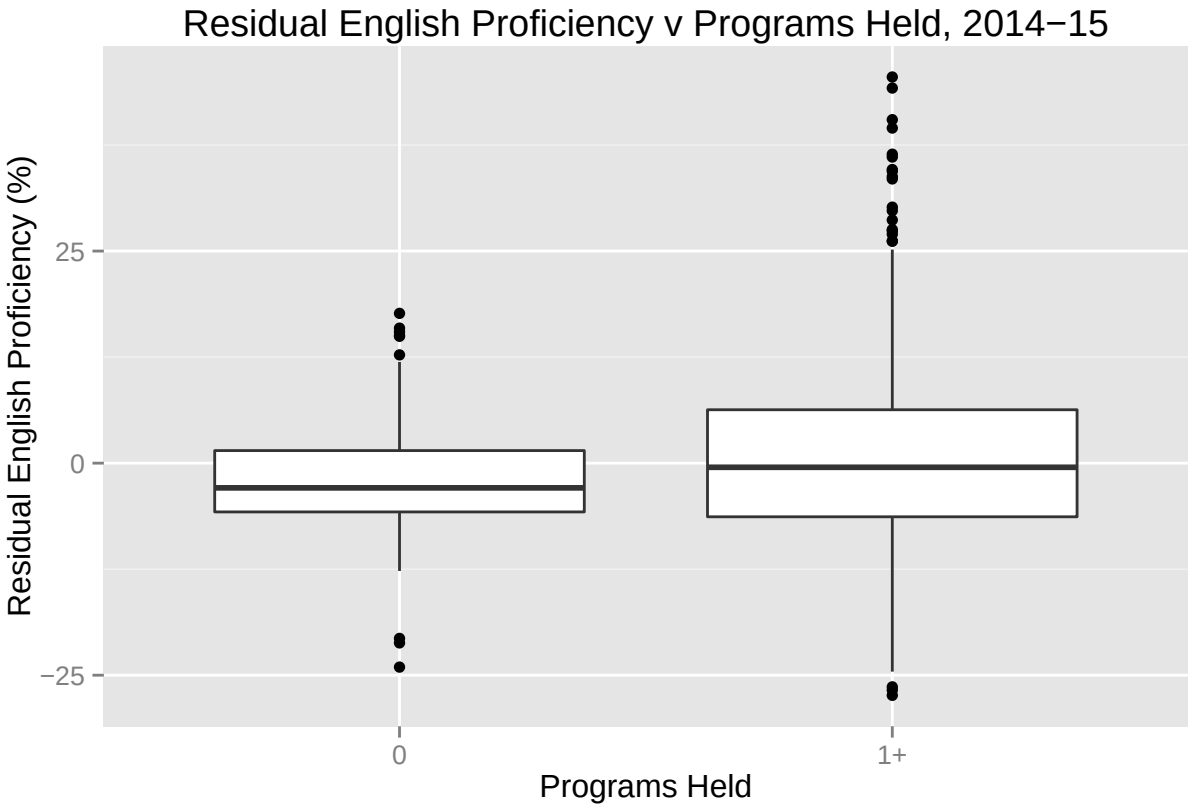
Now we collect information on which schools have programs and which do not.

```
programs <- read.csv("../data/OASK_DB.csv")
programs <- programs[!is.na(programs$SchoolID), 1]
programs <- as.data.frame(table(programs))
names(programs) <- c("SchoolID", "programs")
overall <- merge(overall, programs, all.x = TRUE)
overall$programs[is.na(overall$programs)] <- 0
overall$programs[overall$programs >= 1] <- "1+"
overall$programs <- as.factor(overall$programs)
table(overall$programs)
```

```
##
##    0   1+
##   59 1002
```

Finally, we plot residual school performance against the number of programs held at the school to see if a pattern emerges.




Residual English Proficiency v Programs Held, 2014−15
Residual English Proficiency (%)
25
0
−25
0
1+
Programs Held