



An Empirically Based Method to Set State Test Performance Targets

Methodology Brief

Clint Sattler, Supervisor of Research and Evaluation

Overview

The federal government requires each public school in Tennessee to have performance targets related to the percentage of students “passing” state tests. TDOE sets these targets using an annual measurable objective (AMO) methodology (TDOE, 2018). Mathematically, the AMO, which is reset each year, is calculated as follows;

$$AMO_{current\ year} = \frac{(100\% - \% Proficient_{previous\ year})}{2} * \frac{1}{8} + \% Proficient_{previous\ year}$$

The Knox County Schools (KCS) Department of Research, Evaluation, and Assessment (REA) completed a study on AMO attainment in March 2020 (Sattler, 2020). REA’s findings demonstrated that no school met AMOs in consecutive years (when considering the entire population of tested students). Accountability staff at the Tennessee Department of Education confirmed that it was exceedingly rare for a Tennessee school to meet consecutive AMO targets. The results suggest, among other issues:

- The changes in student outcomes required to meet an AMO target are unsustainable year-over-year.
- The methodology, though based on a high-level policy goal, leads to targets that are not empirically realistic.
- AMO targets provide a precise performance goal rounded to one decimal place. Schools may perform near AMO targets without meeting them.

REA commissioned this study to determine an empirically-based methodology for school-level goal setting and progress monitoring.

Methodology

REA’s methodology relies on historic state test data from Tennessee. REA uses the publicly available data to generate a list of schools with similar demographics and prior state test performance. We then use the test history from the matched schools to create a window of plausible performance targets for the school. We finally report the window and then use contextual information to determine which value in the window would be appropriate for the school to use as a target.

REA coded the methodology using R version 4.1.0 running on RStudio version 2023.09.1 build 494. The REA methodology uses all available state test data from the 2016-2017 school year (SY1617) onward. We did not consider data collected before SY1617 because prior tests used different state standards. Readers should also note that data is unavailable from SY1920 because the COVID-19 pandemic closed the school system before the second-semester test window.

The logic for the R code follows the sequence below:

- Create a grade band-specific list of schools in the state. We want to ensure we compare schools that enroll students in grades 3-5, grades 6-8, or grades 9-12 with other schools that serve the same grade bands.
- Use the test records to determine the demographics of the tested students at each school in Tennessee. The code calculates the total number of students in the school, as well as the percentage of students in each of the state-identified subgroups (Black/Hispanic/Native American (BHN), Economically Disadvantaged (ED), English Language Learners (ELL), and Students with Disabilities (SWD)). We determine the demographics in each school in each year.
- Use the test records to determine the content-specific proficiency rates at each school in Tennessee. We divide content in math (grades 3-8 math, Algebra I, Geometry, and Algebra II) and English/Language Arts (grades 3-8 ELA, English I, and English II). We determine the percentage of students testing proficient (on-track, mastered, met expectations, or exceeded expectations) in each content area in each school in each year.
- Create a data frame of how schools change proficiencies across two years in each content area. REA calculated two-year changes in proficiency by matching records from the same school number and the same system (district) number in consecutive years. REA determined the change as $\text{proficiency}_{\text{year two}} - \text{proficiency}_{\text{year one}}$. The code then maps the two-year change in proficiency to the demographics of each school in the baseline year (year one).
- Create a data frame of how schools change proficiencies across three years in each content area. REA calculated three-year changes in proficiency by matching records from the same school number and the same system (district) number after a gap year. REA determined the change as $\text{proficiency}_{\text{year three}} - \text{proficiency}_{\text{year one}}$. The code then maps the three-year change in proficiency to the demographics of each school in the baseline year (year one).
- We then loop through the following procedure for each KCS school:
 - Select a KCS school (with the most recently available demographic and proficiency data) and remove that school from the list of possible comparison schools.
 - Search the comparison schools for the closest matches in two-year changes in proficiencies and demographics. The algorithm calculates the Euclidean distance (in proficiency and the percentage of students in the four demographic groups) between the chosen KCS school i and each school j in the comparison pool.

$$\begin{aligned}
 \text{Distance}_i^2 &= (\% \text{Prof}_i - \% \text{Prof}_j)^2 + (\% \text{BHN}_i - \% \text{BHN}_j)^2 \\
 &\quad + (\% \text{ED}_i - \% \text{ED}_j)^2 + (\% \text{ELL}_i - \% \text{ELL}_j)^2 \\
 &\quad + (\% \text{SWD}_i - \% \text{SWD}_j)^2 \\
 \text{Distance}_i &= \sqrt{\text{Distance}_i^2}
 \end{aligned}$$

REA chose to return 50 matches with the smallest distance between the current-year KCS school and the other schools in Tennessee. These 50 matches form the KCS school's two-year proficiency comparison pool.

- Calculate the bottom quartile (25th percentile), middle (50th percentile), and top quartile (75th percentile) of proficiency changes. The window between the top

and bottom quartile contains the change in proficiency for the middle 50 percent of comparison schools.

- Search the comparison schools for the closest matches in three-year changes in proficiencies and demographics. Determine the 50 schools with the smallest Euclidean distance in baseline proficiencies and demographics.
 - Calculate the bottom quartile, middle, and top quartile of proficiency changes.
- Report the final quartile cuts for each KCS school i.

The author included the R code for the algorithm in the appendix.

A Note on Sample Size

REA had to choose the number of comparison schools to match. Each KCS school would have similar targets if we used a large pool. This would negate our objective to set empirical goals that made sense considering the context of the school. Each KCS school would have meaningless goals if the comparison pool was too small.

REA investigated the impact of varying the number of matches in the comparisons group from 10 to 100 (by intervals of 10) and 100 to 1000 (by intervals of 100). The author nested the above algorithm inside a loop, allowing us to manipulate the number of matches. We then plotted the distribution of interquartile ranges of the goals. Inspections of the plots did not indicate any particular n count as particularly advantageous over another. REA chose 50 schools somewhat arbitrarily. Fifty matches provided a reasonable distribution of interquartile ranges and seemed to provide different but reasonable targets for each KCS school.

Conclusions

REA presented the new methodology to school leaders before the start of SY2324. REA staff plotted schools' SY2122-SY2223 change in proficiency versus their comparison schools' interquartile range. School leaders reported to REA staff that they appreciated having an empirical basis with which they could monitor their performance.

There are some potential issues with using this methodology. REA uses the state test data to generate the pool of comparison schools. The Tennessee Department of Education (TDOE) does not maintain a regular schedule for (public) data release. Therefore, REA may not be able to update the comparison data with the most recent test results. Additionally, some supervisors may find the empirical ranges politically or motivationally untenable since median changes can be negative. Although this information may help a school reflect on its prior performance, it may be counterproductive to setting goals for the coming school year. Finally, REA needs to study the stability of the algorithm. Specifically, how sensitive is the output when REA updates comparison data without changing input conditions? There may be impacts from the COVID-19 pandemic and lost learning that increases instability in the short term.

References

TDOE (2018). Every Student Succeeds Act: Building Success in Tennessee ESSA State Plan. Retrieved from https://www.tn.gov/content/dam/tn/education/documents/TN_ESSA_State_Plan_Approved.pdf.

Sattler, C. (2020). The Observed Conditional Probabilities of Annual Measurable Objectives. Knox County Schools Department of Research Evaluation and Assessment. Retrieved from <https://www.knoxschools.org/cms/lib/TN01917079/Centricity/domain/1044/research-evaluation/program%20evaluations/The%20Observed%20Conditional%20Probabilities%20of%20Annual%20Measurable%20Objectives.pdf>.

Appendix

```
library(dplyr)
library(openxlsx)
library(tidyr)
library(ggplot2)
```

```
dat.17 <- read.csv("Z:/Sattler/Program Evaluation/SY2223/TNReady Goal
Setting/school_assessment_file_suppressed_17.csv")
dat.18 <- read.csv("Z:/Sattler/Program Evaluation/SY2223/TNReady Goal
Setting/school_assessment_file_suppressed_18.csv")
dat.19 <- read.csv("Z:/Sattler/Program Evaluation/SY2223/TNReady Goal
Setting/school_assessment_file_suppressed_19.csv")
dat.21 <- read.csv("Z:/Sattler/Program Evaluation/SY2223/TNReady Goal
Setting/school_assessment_file_suppressed_21.csv")
dat.22 <- read.xlsx("Z:/Sattler/Program Evaluation/SY2223/TNReady Goal
Setting/school_assessment_file_suppressed_22.xlsx")
```

```
district    <- 470
current.year <- 2022
n.matches   <- 50
current.df   <- dat.22
```

```
renamer<- c(pct_met_exceeded = "pct_on_mastered", student_group = "subgroup")
```

```
grade.bander <- function(df,grades){
  df <- df %>%
    rename(any_of(renamer)) %>%
    filter(valid_tests >= 10) %>%
    select(year, system, school, grade, subject) %>%
    filter(grade %in% grades) %>%
    mutate(short.key = paste(system,school,sep=".")) %>%
    mutate(key      = paste(year, system, school, subject, sep = "."))

  return(df)
}
```

```
elem <- rbind(grade.bander(dat.17,c("3","4","5")),
              grade.bander(dat.18,c("3","4","5")),
              grade.bander(dat.19,c("3","4","5")),
              grade.bander(dat.21,c("3","4","5")),
              grade.bander(dat.22,c("3","4","5")))
```

```
middle <- rbind(grade.bander(dat.17,c("6","7","8")),
```

```

grade.bander(dat.18,c("6","7","8")),
grade.bander(dat.19,c("6","7","8")),
grade.bander(dat.21,c("6","7","8")),
grade.bander(dat.22,c("6","7","8")))

high <- rbind(grade.bander(dat.17,c("9","10","11","12")),
  grade.bander(dat.18,c("9","10","11","12")),
  grade.bander(dat.19,c("9","10","11","12")),
  grade.bander(dat.21,c("9","10","11","12")),
  grade.bander(dat.22,c("9","10","11","12")))

bands <- rbind(elem, middle, high) %>%
  select(short.key, key) %>%
  mutate(band = ifelse(short.key %in% elem$short.key & !(short.key %in% middle$short.key | short.key
%in% high$short.key), "ES",
    ifelse(short.key %in% middle$short.key & !(short.key %in% elem$short.key | short.key
%in% high$short.key), "MS",
    ifelse(short.key %in% high$short.key & !(short.key %in% elem$short.key | short.key
%in% middle$short.key), "HS","MIX")))) %>%
  select(key, band) %>%
  unique()

demos <- function(df){
  df <- df %>%
    rename(any_of(renamer)) %>%
    filter(grade == "All Grades" &
      subject %in% c("Algebra I", "Algebra II", "Integrated Math I", "Integrated Math II", "English I",
"English II", "ELA", "Math") &
      student_group %in% c("All Students", "Black/Hispanic/Native American", "Economically
Disadvantaged", "English Learners", "Students with Disabilities")) %>%
    mutate(valid_tests = as.numeric(valid_tests)) %>%
    mutate(pct_met_exceeded = as.numeric(pct_met_exceeded)) %>%
    group_by(year, system, school, subject, grade, student_group) %>%
    summarise(valid_tests = sum(valid_tests)) %>%
    pivot_wider(names_from = student_group, values_from = valid_tests) %>%
    mutate_all(~replace_na(.,0)) %>%
    filter(`All Students` >= 10) %>%
    mutate(`Black/Hispanic/Native American` = `Black/Hispanic/Native American`/`All Students`*100)
%>%
    mutate(`Economically Disadvantaged` = `Economically Disadvantaged`/`All Students`*100) %>%
    mutate(`English Learners` = `English Learners`/`All Students`*100) %>%
    mutate(`Students with Disabilities` = `Students with Disabilities`/`All Students`*100) %>%
    ungroup()

```

```

return(df)
}

all.demos <- rbind(demos(dat.17), demos(dat.18), demos(dat.19), demos(dat.21), demos(dat.22)) %>%
  mutate(key = paste(year, system, school, subject, sep = ".")) %>%
  merge(y = bands,
        by = "key",
        all.x = TRUE)

prof <- function(df){
  df <- df %>%
    rename(any_of(renamer)) %>%
    filter(grade == "All Grades" &
           subject %in% c("Algebra I", "Algebra II", "Integrated Math I", "Integrated Math II", "English I",
"English II", "ELA", "Math") &
           student_group %in% c("All Students")) %>%
    mutate(valid_tests = as.numeric(valid_tests)) %>%
    mutate(pct_met_exceeded = as.numeric(pct_met_exceeded)) %>%
    na.omit() %>%
    mutate(n_met_exceeded = pct_met_exceeded/100*valid_tests) %>%
    group_by(year, system, school, subject, grade) %>%
    summarise(valid_tests = sum(valid_tests),
              n_met_exceeded = sum(n_met_exceeded)) %>%
    mutate(pct_met_exceeded = n_met_exceeded/valid_tests * 100) %>%
    mutate(pct_met_exceeded = round(pct_met_exceeded, 1)) %>%
    select(-n_met_exceeded, - valid_tests) %>%
    ungroup()

  return(df)
}

base.dat <- rbind(prof(dat.17), prof(dat.18), prof(dat.19), prof(dat.21), prof(dat.22)) %>%
  mutate(key = paste(year, system, school, subject, sep = ".")) %>%
  mutate(two.year.key = paste(year+1, system, school, subject, sep = ".")) %>%
  mutate(three.year.key = paste(year+2, system, school, subject, sep = ".")) %>%
  merge(y = all.demos %>%
        select(-year, -system, -school, -subject, - grade ),
        by = "key",
        all = FALSE)

one.year.comp <- base.dat %>%
  filter(year %in% c(2017, 2018, 2021)) %>%

```

```

select(year, system, school, subject, pct_met_exceeded, key, two.year.key) %>%
merge(y = base.dat %>%
  select(key, pct_met_exceeded),
  by.x = "two.year.key",
  by.y = "key",
  all.x = TRUE) %>%
na.omit() %>%
mutate(two.year.prof_change = pct_met_exceeded.y - pct_met_exceeded.x) %>%
select(-pct_met_exceeded.y, -pct_met_exceeded.x)

```

```

three.year.comp <- base.dat %>%
filter(year %in% c(2017)) %>%
select(year, system, school, subject, pct_met_exceeded, key, three.year.key) %>%
merge(y = base.dat %>%
  select(key, pct_met_exceeded),
  by.x = "three.year.key",
  by.y = "key",
  all.x = TRUE) %>%
na.omit() %>%
mutate(three.year.prof_change = pct_met_exceeded.y - pct_met_exceeded.x) %>%
select(-pct_met_exceeded.y, -pct_met_exceeded.x)

```

```

demos.and.prof <- all.demos %>%
merge(y = base.dat %>%
  select(key, pct_met_exceeded),
  by = "key",
  all.x = TRUE) %>%
na.omit()

```

```

#####
knox.demos <- subset(demos.and.prof,
  # year %in% c(2017, 2018, 2022) & system == district) %>%
  year %in% c(2022) & system == district) %>%
filter(!(band == "HS" & subject %in% c("ELA", "Math")))

```

```

demos.and.prof.more <- demos.and.prof %>%
merge(y = one.year.comp %>%
  select(key, two.year.prof_change),
  by = "key",
  all.x = TRUE) %>%
merge(y = three.year.comp %>%
  select(key, three.year.prof_change),

```



```

    by = "key",
    all.x = TRUE)

n.range<- n.matches

ind.matches.one.year    <- data.frame()

for(j in n.range){

  n.matches              <- j
  final.one.year.errors  <- data.frame()
  final.three.year.errors <- data.frame()

  for(i in 1:nrow(knox.demos)){

    band.i              <- knox.demos[i,]$band

    max.size.i <- demos.and.prof.more %>%
      filter(band == band.i) %>%
      summarise(max(`All Students`, na.rm = TRUE)) %>%
      pull()

    school.i          <- knox.demos[i,]$school
    subject.i          <- knox.demos[i,]$subject
    pct_met_exceeded.i <- knox.demos[i,]$pct_met_exceeded
    bhn.i              <- knox.demos[i,]$`Black/Hispanic/Native American`
    ed.i               <- knox.demos[i,]$`Economically Disadvantaged`
    swd.i              <- knox.demos[i,]$`Students with Disabilities`
    ell.i              <- knox.demos[i,]$`English Learners`
    count.i            <- knox.demos[i,]$`All Students`/max.size.i*100

    comp.one.i <- demos.and.prof.more %>%
      filter(!(system == district & school == school.i) & band == band.i) %>%
      filter(subject == subject.i) %>%
      drop_na(two.year.prof_change) %>%
      mutate(size.pct = `All Students`/max.size.i*100)

    comp.three.i <- demos.and.prof.more %>%
      filter(!(system == district & school == school.i) & band == band.i) %>%
      filter(subject == subject.i) %>%
      drop_na(three.year.prof_change) %>%
      mutate(size.pct = `All Students`/max.size.i*100)
  }
}

```

```

comp.one.i$dist.one.i <- sqrt((comp.one.i$size.pct - count.i)^2+
                             (comp.one.i$pct_met_exceeded - pct_met_exceeded.i)^2+
                             (comp.one.i$`Black/Hispanic/Native American` - bhn.i)^2+
                             (comp.one.i$`Economically Disadvantaged` - ed.i)^2+
                             (comp.one.i$`Students with Disabilities` - swd.i)^2+
                             (comp.one.i$`English Learners` - ell.i)^2)

comp.three.i$dist.three.i <- sqrt((comp.three.i$size.pct - count.i)^2+
                                   (comp.three.i$pct_met_exceeded - pct_met_exceeded.i)^2+
                                   (comp.three.i$`Black/Hispanic/Native American` - bhn.i)^2+
                                   (comp.three.i$`Economically Disadvantaged` - ed.i)^2+
                                   (comp.three.i$`Students with Disabilities` - swd.i)^2+
                                   (comp.three.i$`English Learners` - ell.i)^2)

comp.one.reduced.i <- comp.one.i %>%
  arrange(dist.one.i) %>%
  slice(1:j)

ind.matches.i          <- comp.one.reduced.i
ind.matches.i$kcs.school <- school.i
ind.matches.i$subject   <- subject.i
ind.matches.one.year    <- rbind(ind.matches.one.year, ind.matches.i)

comp.three.reduced.i <- comp.three.i %>%
  arrange(dist.three.i) %>%
  slice(1:j)

knox.demos[i,14] <- quantile(comp.one.reduced.i$two.year.prof_change, probs = c(.25))
knox.demos[i,15] <- quantile(comp.one.reduced.i$two.year.prof_change, probs = c(.5))
knox.demos[i,16] <- quantile(comp.one.reduced.i$two.year.prof_change, probs = c(.75))

knox.demos[i,17] <- quantile(comp.three.reduced.i$three.year.prof_change, probs = c(.25))
knox.demos[i,18] <- quantile(comp.three.reduced.i$three.year.prof_change, probs = c(.5))
knox.demos[i,19] <- quantile(comp.three.reduced.i$three.year.prof_change, probs = c(.75))

knox.demos[i,20] <- median(comp.one.reduced.i$dist.one.i)
knox.demos[i,21] <- median(comp.three.reduced.i$dist.three.i)
knox.demos[i,22] <- j

j.one.year.errors <- comp.one.reduced.i %>%
  select(band, dist.one.i)

```

```

j.three.year.errors <- comp.three.reduced.i %>%
  select(band, dist.three.i)

j.one.year.errors$n.matches <- j.three.year.errors$n.matches <- n.matches
j.one.year.errors$school <- j.three.year.errors$school <- school.i

final.one.year.errors <- rbind(final.one.year.errors, j.one.year.errors)
final.three.year.errors <- rbind(final.three.year.errors, j.one.year.errors)
}

colnames(knox.demos)[14] <- 'bottom.quartile.one.year.change'
colnames(knox.demos)[15] <- 'median.one.year.change'
colnames(knox.demos)[16] <- 'top.quartile.one.year.change'
colnames(knox.demos)[17] <- 'bottom.quartile.three.year.change'
colnames(knox.demos)[18] <- 'median.three.year.change'
colnames(knox.demos)[19] <- 'top.quartile.three.year.change'
colnames(knox.demos)[20] <- 'median.one.year.dist'
colnames(knox.demos)[21] <- 'median.three.year.dist'
colnames(knox.demos)[22] <- 'n.matches'

assign(paste("knox.demos",n.matches,sep = "."),
      knox.demos)

assign(paste("one.year.errors",n.matches,sep = "."),
      final.one.year.errors)

assign(paste("three.year.errors",n.matches,sep = "."),
      final.three.year.errors)
}

write.csv(ind.matches.one.year,
  paste0("Z:/Sattler/Program Evaluation/SY2223/TNReady Goal Setting/",
    format(Sys.Date(), "%m.%d.%y"),
    ".one.year.school.matches.csv"),
  row.names = FALSE)

write.csv(knox.demos.50,
  paste0("Z:/Sattler/Program Evaluation/SY2223/TNReady Goal Setting/",
    format(Sys.Date(), "%m.%d.%y"),
    ".final.school.goal.ranges.csv"),
  row.names = FALSE)

```