NYC GT data

Pengruijie Zhou

2025-03-08

Load libraries and read files

```
# Load necessary libraries
library(readxl)
library(dplyr)

# Read the Excel files
setwd("C:/Users/78789/Desktop/McDaniel/ANA515/DATA")
df1 <- read_excel("G&T Results 2017-18 Responses.xlsx")
df2 <- read_excel("G&T Results 2018-19 Responses.xlsx")
```

Standardizing column names

The first 12 columns are identical in both datasets, while the 2018-19 dataset has two extra columns: 'Unnamed: 12' and 'Unnamed: 13'. These seem unnecessary. So, I removed the extra columns from the 2018-19 dataset and ensured that both datasets have the exact same column names.

```
# Keep only columns present in df1
df2 <- df2[, names(df1)]

# Print column names to verify
print(names(df1))
```

```
##  [1] "Timestamp"                  "Entering Grade Level"
##  [3] "District"                   "Birth Month"
##  [5] "OLSAT Verbal Score"         "OLSAT Verbal Percentile"
##  [7] "NNAT Non Verbal Raw Score"  "NNAT Non Verbal Percentile"
##  [9] "Overall Score"              "School Preferences"
## [11] "School Assigned"            "Will you enroll there?"
```

```
print(names(df2))
```

```
##  [1] "Timestamp"                  "Entering Grade Level"
##  [3] "District"                   "Birth Month"
##  [5] "OLSAT Verbal Score"         "OLSAT Verbal Percentile"
##  [7] "NNAT Non Verbal Raw Score"  "NNAT Non Verbal Percentile"
##  [9] "Overall Score"              "School Preferences"
## [11] "School Assigned"            "Will you enroll there?"
```

Handling data types

Now that both datasets have standardized column names, the next step is ensuring that each column has the correct data type. There are some issues with the data types:

1. Timestamp Column is stored as text (character), and should be converted to Date-Time format;
2. District Column has inconsistent data types in two files:it is stored as character (text)in df1 and stored as numeric in df2. Should be converted both to numeric;
3. Some scores of OLSAT and NNAT Score Columns are stored as text (e.g., "28/30") and should be converted to numeric.

```
# Convert Timestamp to Date-Time format
df1$Timestamp <- as.POSIXct(df1$Timestamp, format="%Y-%m-%d %H:%M:%S")
df2$Timestamp <- as.POSIXct(df2$Timestamp, format="%Y-%m-%d %H:%M:%S")

# Convert District to numeric
df1$District <- as.numeric(df1$District)
df2$District <- as.numeric(df2$District)

# Extract and convert scores from "28/30" format to numeric
# Function to extract numeric value from "28/30" format
extract_score <- function(score) {
  return(as.numeric(sub("/.*", "", score))) # Extract first number before "/"
}

# Apply function to relevant columns
df1$`OLSAT Verbal Score` <- sapply(df1$`OLSAT Verbal Score`, extract_score)
df1$`NNAT Non Verbal Raw Score` <- sapply(df1$`NNAT Non Verbal Raw Score`, extract_score)

df2$`OLSAT Verbal Score` <- sapply(df2$`OLSAT Verbal Score`, extract_score)
df2$`NNAT Non Verbal Raw Score` <- sapply(df2$`NNAT Non Verbal Raw Score`, extract_score)

# Check data structure after conversion
str(df1)
```

```
## tibble [117 x 12] (S3: tbl_df/tbl/data.frame)
##  $ Timestamp                 : POSIXct[1:117], format: "2017-04-08 06:44:01" "2017-04-07 10:40:45" .
##  $ Entering Grade Level      : chr [1:117] "1" "K" "1" "K" ...
##  $ District                  : num [1:117] 6 NA NA NA 22 NA NA NA NA NA ...
##  $ Birth Month               : chr [1:117] "September" "August" "March" "September" ...
##  $ OLSAT Verbal Score        : Named num [1:117] 28 25 27 23 2 24 26 24 23 29 ...
##   ..- attr(*, "names")= chr [1:117] "28/30" "25" "27" "23" ...
##  $ OLSAT Verbal Percentile   : chr [1:117] "99" "99" "96" "97" ...
##  $ NNAT Non Verbal Raw Score : Named num [1:117] 45 39 42 40 38 36 42 42 42 44 ...
##   ..- attr(*, "names")= chr [1:117] "45/50" "39" "42" "40" ...
##  $ NNAT Non Verbal Percentile: chr [1:117] "99" "99" "99" "99" ...
##  $ Overall Score             : num [1:117] 99 99 98 98 99 0 99 99 95 99 ...
##  $ School Preferences        : chr [1:117] "NEST+m, TAG, Anderson, Q300" "Anderson, NEST+m" NA NA ..
##  $ School Assigned           : chr [1:117] "NEST" NA NA NA ...
##  $ Will you enroll there?    : chr [1:117] "YES" "Maybe" "Maybe" NA ...
```

```r
str(df2)
```

```
## tibble [100 x 12] (S3: tbl_df/tbl/data.frame)
##  $ Timestamp                 : POSIXct[1:100], format: "2017-03-27" "2018-03-27" ...
##  $ Entering Grade Level      : chr [1:100] "1" "1" "1" "1" ...
##  $ District                  : num [1:100] 13 1 2 2 3 2 2 2 2 2 ...
##  $ Birth Month               : chr [1:100] "August" "March" "February" "January" ...
##  $ OLSAT Verbal Score        : Named num [1:100] 30 26 26 25 28 25 27 25 25 23 ...
##   ..- attr(*, "names")= chr [1:100] "30" "26" "26" "25" ...
##  $ OLSAT Verbal Percentile   : num [1:100] 99 99 99 99 98 99 99 99 99 98 ...
##  $ NNAT Non Verbal Raw Score : Named num [1:100] 46 47 47 45 45 NA 43 42 40 39 ...
##   ..- attr(*, "names")= chr [1:100] "46" "47" "47" "45" ...
##  $ NNAT Non Verbal Percentile: num [1:100] 99 99 99 99 0 99 99 99 99 99 ...
##  $ Overall Score             : num [1:100] 99 99 99 99 99 99 99 99 99 99 ...
##  $ School Preferences        : chr [1:100] "Anderson?" "1111" NA "NEST+M" ...
##  $ School Assigned           : chr [1:100] NA NA NA NA ...
##  $ Will you enroll there?    : chr [1:100] "at LL now, has a sib ent K w/ 99" NA NA "Yes" ...
```

Handling critical missing values

In this dataset, I encountered missing values in several columns, including District, School Preferences, School Assigned, and Will you enroll there?.

For critical variables such as Timestamp, OLSAT Verbal Score, NNAT Non-Verbal Raw Score, and Overall Score, I chose to remove rows with missing values to ensure data accuracy and completeness for analysis.

However, for non-critical variables like District, I retained the missing values because they were originally absent in the 2017-18 dataset, and imputing them without reliable external data could introduce bias.

Similarly, I kept missing values in School Preferences, School Assigned, and Will you enroll there?, as these were optional responses that do not impact the core analysis.

```r
# Count missing values in each dataset
missing_df1 <- colSums(is.na(df1))
missing_df2 <- colSums(is.na(df2))

# Print missing value counts
print(missing_df1)
```

```
##                  Timestamp       Entering Grade Level
##                          0                          0
##                   District                Birth Month
##                         16                          0
##         OLSAT Verbal Score    OLSAT Verbal Percentile
##                          5                          0
##  NNAT Non Verbal Raw Score NNAT Non Verbal Percentile
##                          6                          0
##              Overall Score         School Preferences
##                          0                         42
##            School Assigned     Will you enroll there?
##                         87                         46
```

```r
print(missing_df2)
```

```
##                   Timestamp       Entering Grade Level
##                           4                          0
##                    District                Birth Month
##                           0                          1
##         OLSAT Verbal Score    OLSAT Verbal Percentile
##                           6                          1
##  NNAT Non Verbal Raw Score NNAT Non Verbal Percentile
##                           8                          0
##               Overall Score         School Preferences
##                           0                         26
##             School Assigned     Will you enroll there?
##                          78                         56
```

```r
# Remove rows where Timestamp is missing
df1 <- df1[!is.na(df1$Timestamp), ]
df2 <- df2[!is.na(df2$Timestamp), ]

# Remove rows where OLSAT Verbal Score, NNAT Non-Verbal Raw Score are missing
df1 <- df1[!is.na(df1$`OLSAT Verbal Score`) & !is.na(df1$`NNAT Non Verbal Raw Score`), ]
df2 <- df2[!is.na(df2$`OLSAT Verbal Score`) & !is.na(df2$`NNAT Non Verbal Raw Score`), ]

# Print missing values again to verify
print(colSums(is.na(df1)))
```

```
##                   Timestamp       Entering Grade Level
##                           0                          0
##                    District                Birth Month
##                          16                          0
##         OLSAT Verbal Score    OLSAT Verbal Percentile
##                           0                          0
##  NNAT Non Verbal Raw Score NNAT Non Verbal Percentile
##                           0                          0
##               Overall Score         School Preferences
##                           0                         40
##             School Assigned     Will you enroll there?
##                          84                         44
```

```r
print(colSums(is.na(df2)))
```

```
##                   Timestamp       Entering Grade Level
##                           0                          0
##                    District                Birth Month
##                           0                          0
##         OLSAT Verbal Score    OLSAT Verbal Percentile
##                           0                          0
##  NNAT Non Verbal Raw Score NNAT Non Verbal Percentile
##                           0                          0
##               Overall Score         School Preferences
##                           0                         22
##             School Assigned     Will you enroll there?
##                          71                         47
```

Merging the two datasets

Now that all critical missing values have been removed, I then merged the 2017-18 and 2018-19 datasets into a single dataset.

```
# Add a new column "Year" to distinguish the datasets
df1$Year <- "2017-18"
df2$Year <- "2018-19"

# Combine the datasets
final_df <- rbind(df1, df2)

# Print the structure to verify merge success
str(final_df)
```

```
## tibble [199 x 13] (S3: tbl_df/tbl/data.frame)
##  $ Timestamp                 : POSIXct[1:199], format: "2017-04-08 06:44:01" "2017-04-07 10:40:45" .
##  $ Entering Grade Level      : chr [1:199] "1" "K" "1" "K" ...
##  $ District                  : num [1:199] 6 NA NA NA 22 NA NA NA NA NA ...
##  $ Birth Month               : chr [1:199] "September" "August" "March" "September" ...
##  $ OLSAT Verbal Score        : Named num [1:199] 28 25 27 23 2 24 26 24 23 29 ...
##   ..- attr(*, "names")= chr [1:199] "28/30" "25" "27" "23" ...
##  $ OLSAT Verbal Percentile   : chr [1:199] "99" "99" "96" "97" ...
##  $ NNAT Non Verbal Raw Score : Named num [1:199] 45 39 42 40 38 36 42 42 42 44 ...
##   ..- attr(*, "names")= chr [1:199] "45/50" "39" "42" "40" ...
##  $ NNAT Non Verbal Percentile: chr [1:199] "99" "99" "99" "99" ...
##  $ Overall Score             : num [1:199] 99 99 98 98 99 0 99 99 95 99 ...
##  $ School Preferences        : chr [1:199] "NEST+m, TAG, Anderson, Q300" "Anderson, NEST+m" NA NA ..
##  $ School Assigned           : chr [1:199] "NEST" NA NA NA ...
##  $ Will you enroll there?    : chr [1:199] "YES" "Maybe" "Maybe" NA ...
##  $ Year                      : chr [1:199] "2017-18" "2017-18" "2017-18" "2017-18" ...
```

Data visualization

I chose two visualizations to effectively highlight key aspects of the dataset.

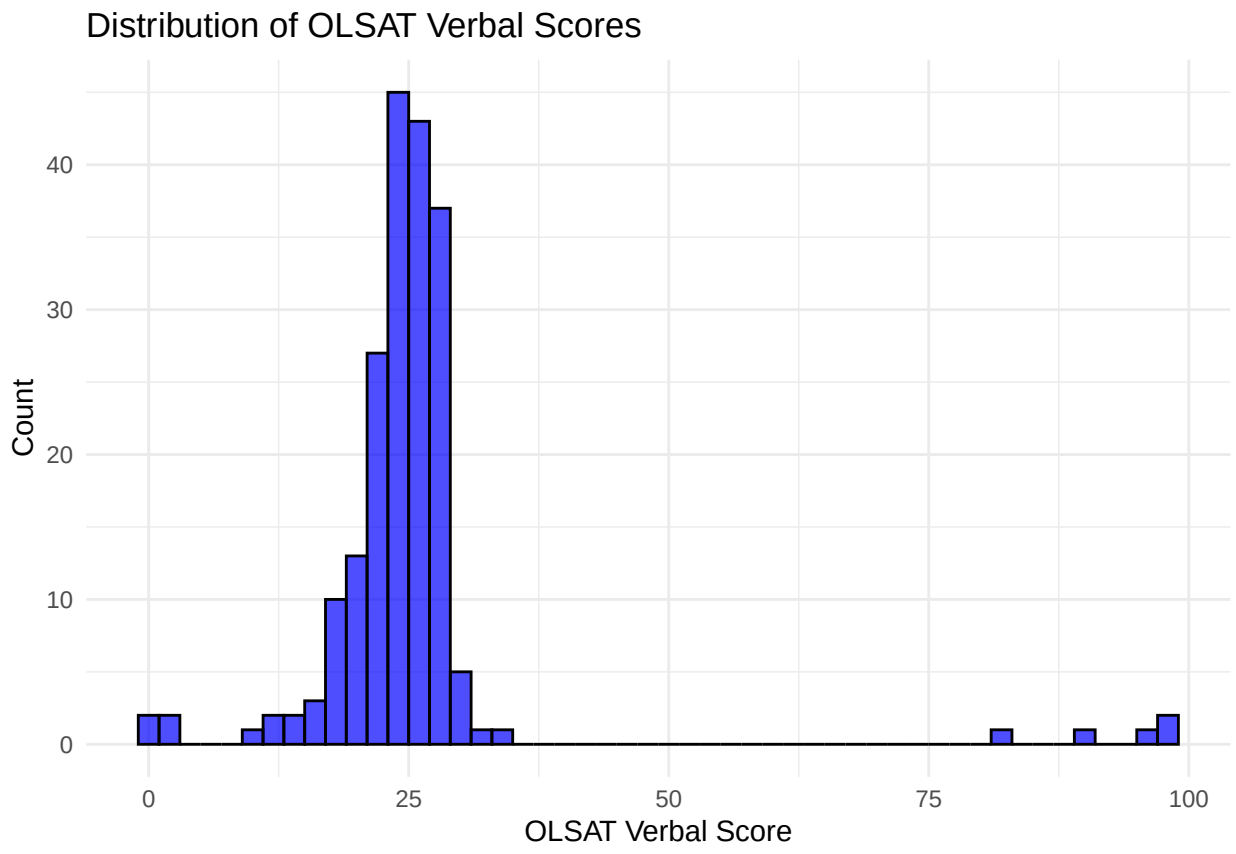
The histogram of OLSAT Verbal Scores allows me to visualize the distribution of student scores, helping to identify patterns such as normality, skewness, or clustering within specific ranges. This provides insight into overall performance trends.

I also chose the boxplot of Overall Scores by Year to compare the 2017-18 and 2018-19 datasets, as it clearly shows differences in median scores, quartiles, and potential outliers. This helps me determine whether student performance varied across years and if there were any significant fluctuations.

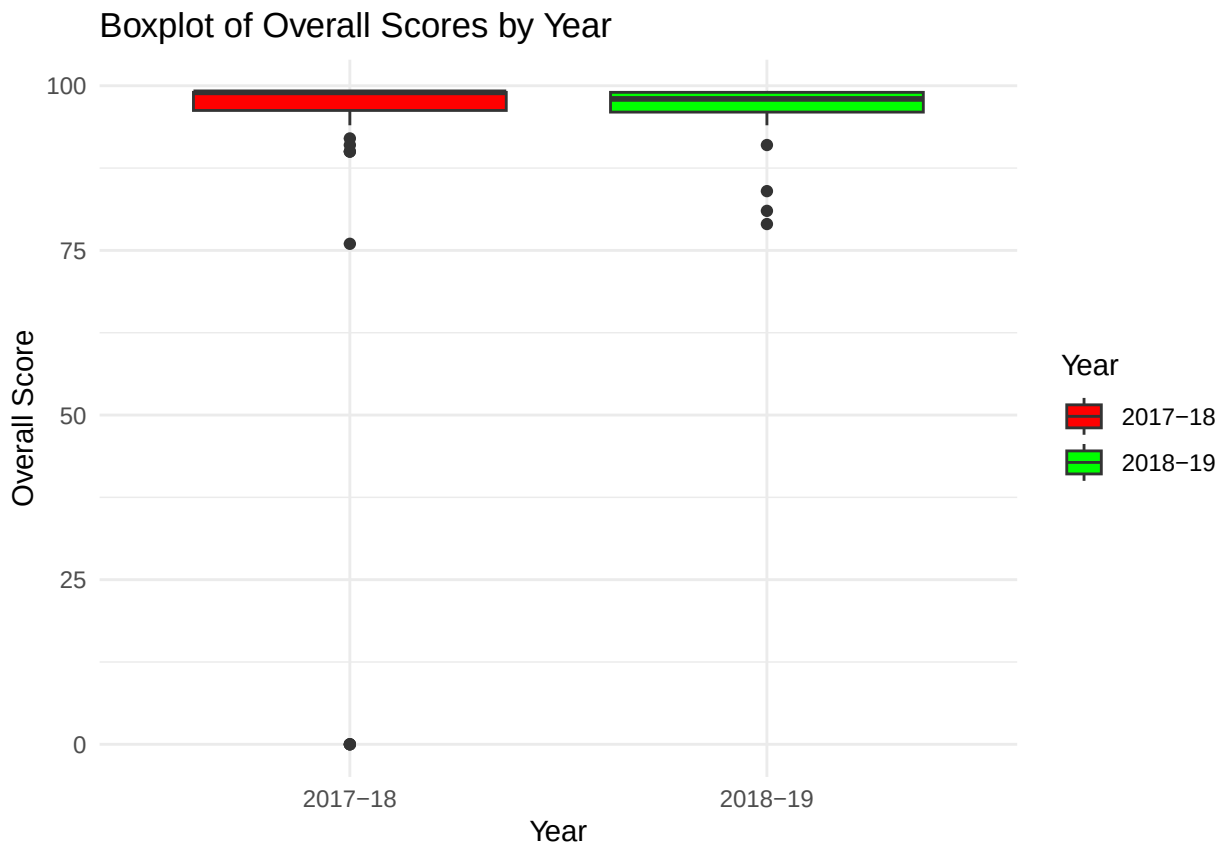
Together, these visualizations offer a comprehensive view of score distributions and year-over-year trends, making them essential for analyzing the dataset effectively.

```
# Load necessary visualization libraries
library(ggplot2)

# Histogram of OLSAT Verbal Scores
ggplot(final_df, aes(x = `OLSAT Verbal Score`)) +
  geom_histogram(binwidth = 2, fill = "blue", color = "black", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Distribution of OLSAT Verbal Scores", x = "OLSAT Verbal Score", y = "Count")
```




```r
# Boxplot of Overall Scores by Year
ggplot(final_df, aes(x = Year, y = `Overall Score`, fill = Year)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Boxplot of Overall Scores by Year", x = "Year", y = "Overall Score") +
  scale_fill_manual(values = c("2017-18" = "red", "2018-19" = "green"))
```



Save cleaned data

```
# Save the cleaned dataset as a CSV file
write.csv(final_df, "final_df.csv", row.names = FALSE)
```