EDA

Cindy Sun, Cathy Liu, Jenny Zhu

2025-10-31

```r
library(tidyverse)
library(readr)
library(readxl)
library(snakecase)
```

```r

# reading the csv:
SchoolQualityReport <- read_csv("2017-2018_School_Quality_Report_-_High_School_20240807.csv")
RegentsExams <- read_excel("2014-15-to-2018-19-nyc-regents-overall-and-by-category---public.xls

colnames(SchoolQualityReport) <- to_snake_case(colnames(SchoolQualityReport))
colnames(RegentsExams) <- to_snake_case(colnames(RegentsExams))


# selecting the relevant variables from this data set.
SchoolQuality_Selected <- SchoolQualityReport |>
  select(dbn, school_name, percent_of_teachers_with_3_or_more_years_of_experience, percent_of_s

# changing one column name to make it easier to left join later
colnames(SchoolQuality_Selected)[1] <- c("school_dbn")


# filtering the RegentsExams data set
RegentsEdited <- RegentsExams |>
  select(2:10) |>
  filter(school_level == "High school", year == "2018", school_type == "General Academic")

# filtering for only the two tests we are interested in
RegentsEdited <- RegentsEdited |>
  filter(regents_exam == "Common Core Algebra2" | regents_exam == "Common Core English")

# creating two new rows, one for the english scores one for the math scores
RegentsEdited <- RegentsEdited |>
  pivot_wider(names_from = regents_exam,
              values_from = mean_score)
```

```r
# changing the names to work for programming:
colnames(RegentsEdited)[8:9] <- c("mean_algebra2", "mean_english")


# creating a data frame that only contains Algebra2 observations
Algebra2 <- RegentsEdited |>
  select(!9) |>
  filter(mean_algebra2 != "s" & !is.na(mean_algebra2))

English <- RegentsEdited |>
  select(school_dbn, mean_english) |>
  filter(mean_english != "s" & !is.na(mean_english))

# joining the math and english exam data sets into final one
FinalRegents <- left_join(x = Algebra2, y = English, by = "school_dbn")


# selecting only the minimum columns we are interested in
Scores <- FinalRegents |>
  select(school_dbn, mean_algebra2, mean_english)

# left joining by School.DBN to get the final data frame we want to work with:
joint_data <- left_join(x = Scores, y = SchoolQuality_Selected, by = "school_dbn")

# The test score columns were character vectors so we needed to coerce them to numeric vectors
joint_data$mean_algebra2 <- as.numeric(joint_data$mean_algebra2)
joint_data$mean_english <- as.numeric(joint_data$mean_english)


# final data set is called joint_data
```

```r

# Check missing data
sum(is.na(joint_data))
```

DATA IMPORT/WRANGLING

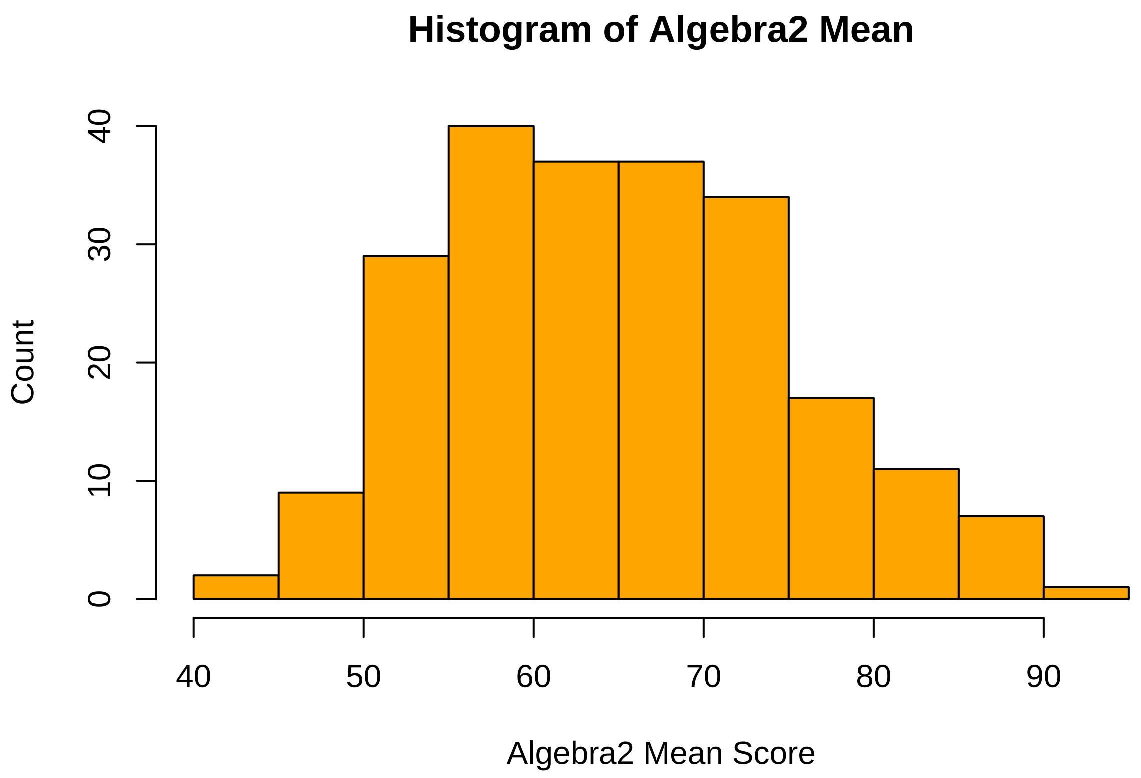
```
## [1] 17
```

```r
colSums(is.na(joint_data))
```

```
##                                              school_dbn
##                                                       0
##                                           mean_algebra2
##                                                       0
##                                            mean_english
##                                                       0
##                                             school_name
##                                                       0
## percent_of_teachers_with_3_or_more_years_of_experience
##                                                      16
##                 percent_of_students_chronically_absent
##                                                       1
##                  rigorous_instruction_percent_positive
##                                                       0
##                supportive_environment_percent_positive
##                                                       0
##                                    economic_need_index
##                                                       0
```

```
# Filter the data to fit predictions for math and english
cleaned_data <- joint_data |>
  dplyr::filter(
    !is.na(percent_of_teachers_with_3_or_more_years_of_experience),
    !is.na(percent_of_students_chronically_absent)
  )
```

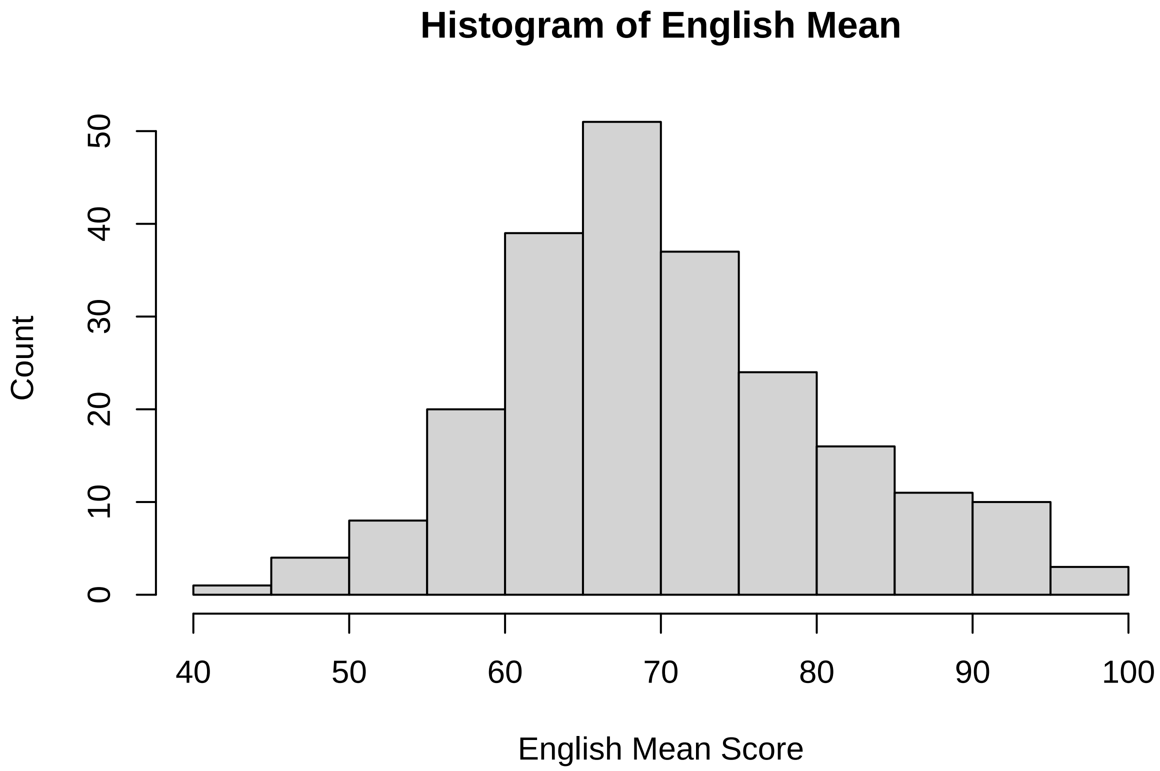
```
# Summary mean_algebra2
# Graphical
hist(cleaned_data$mean_algebra2,
main = "Histogram of Algebra2 Mean",
xlab = "Algebra2 Mean Score",
ylab = "Count",
col = "orange")
```



```
# Numerical
summary(cleaned_data$mean_algebra2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   42.71   56.62   63.92   64.79   72.01   90.52
```

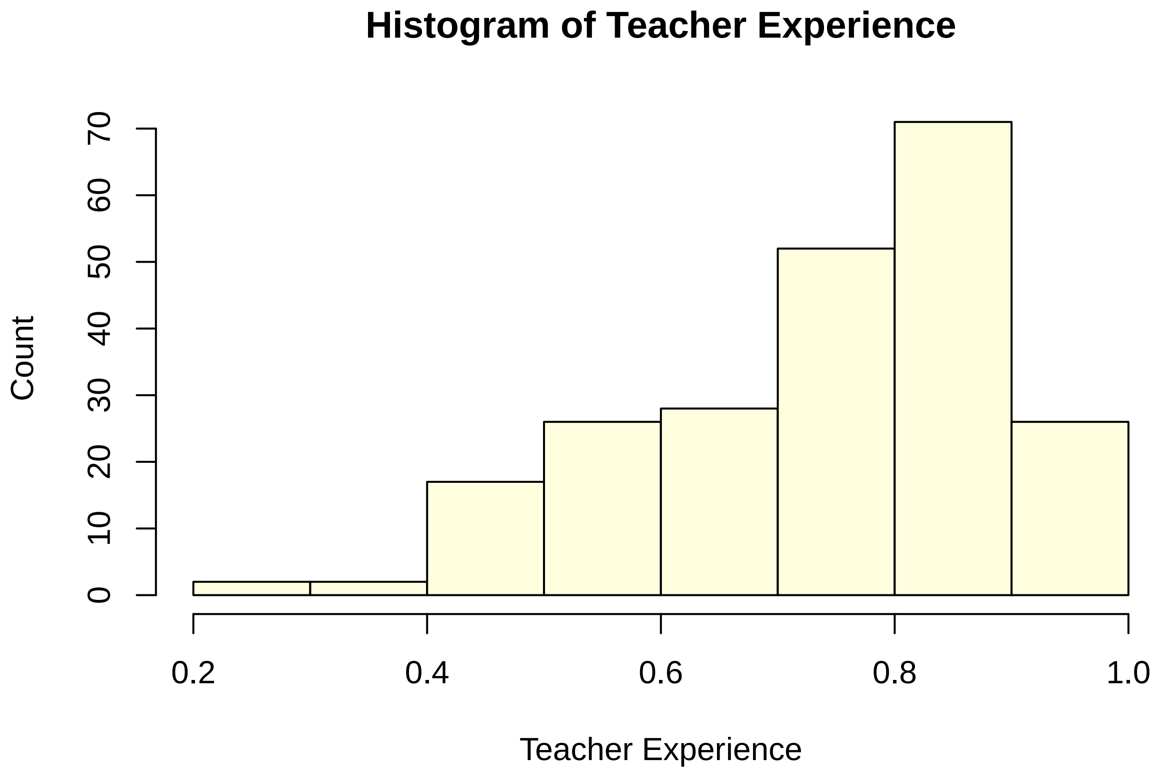
```
# Summary mean_english
# Graphical
hist(cleaned_data$mean_english,
main = "Histogram of English Mean",
xlab = "English Mean Score",
ylab = "Count",
col = "lightgrey")
```



```
# Numerical
summary(cleaned_data$mean_english)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   44.40   62.71   69.11   70.14   76.26   95.45
```

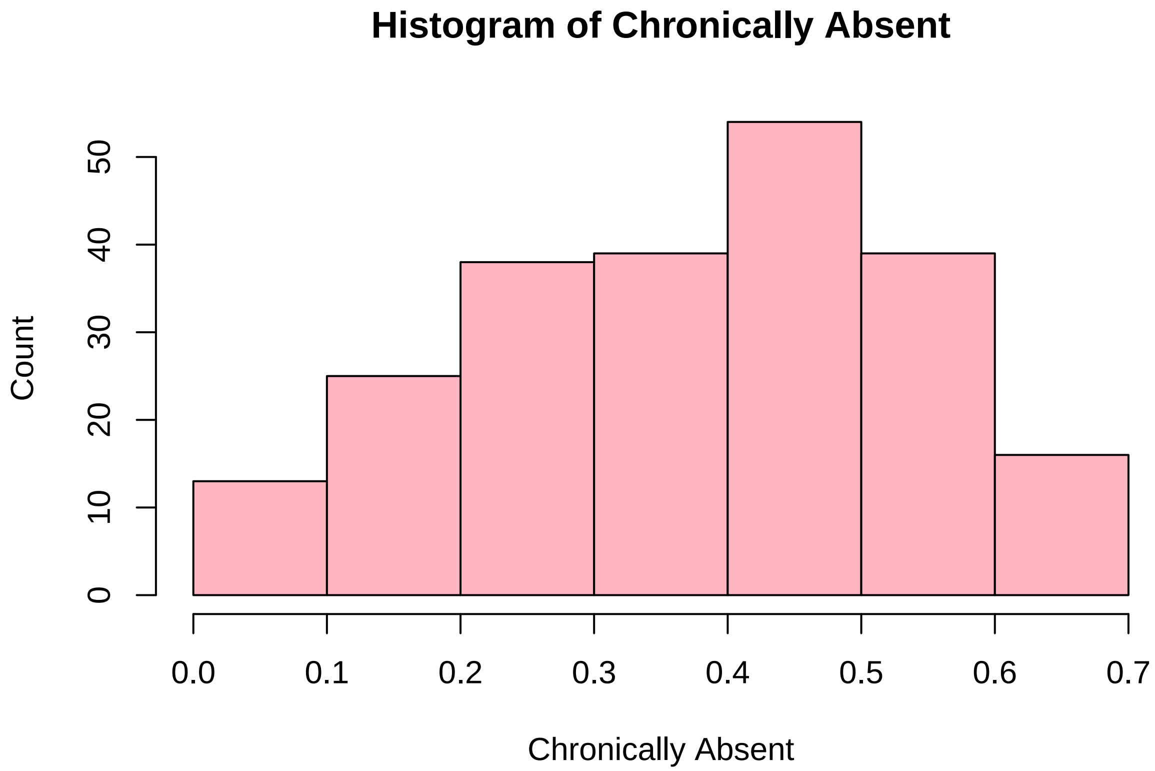
```
# Summary percent_of_teachers_with_3_or_more_years_of_experience
# Graphical
hist(cleaned_data$percent_of_teachers_with_3_or_more_years_of_experience,
main = "Histogram of Teacher Experience",
xlab = "Teacher Experience",
ylab = "Count",
col = "lightyellow")
```



```
# Numerical
summary(cleaned_data$percent_of_teachers_with_3_or_more_years_of_experience)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.2690  0.6512  0.7785  0.7419  0.8525  1.0000
```

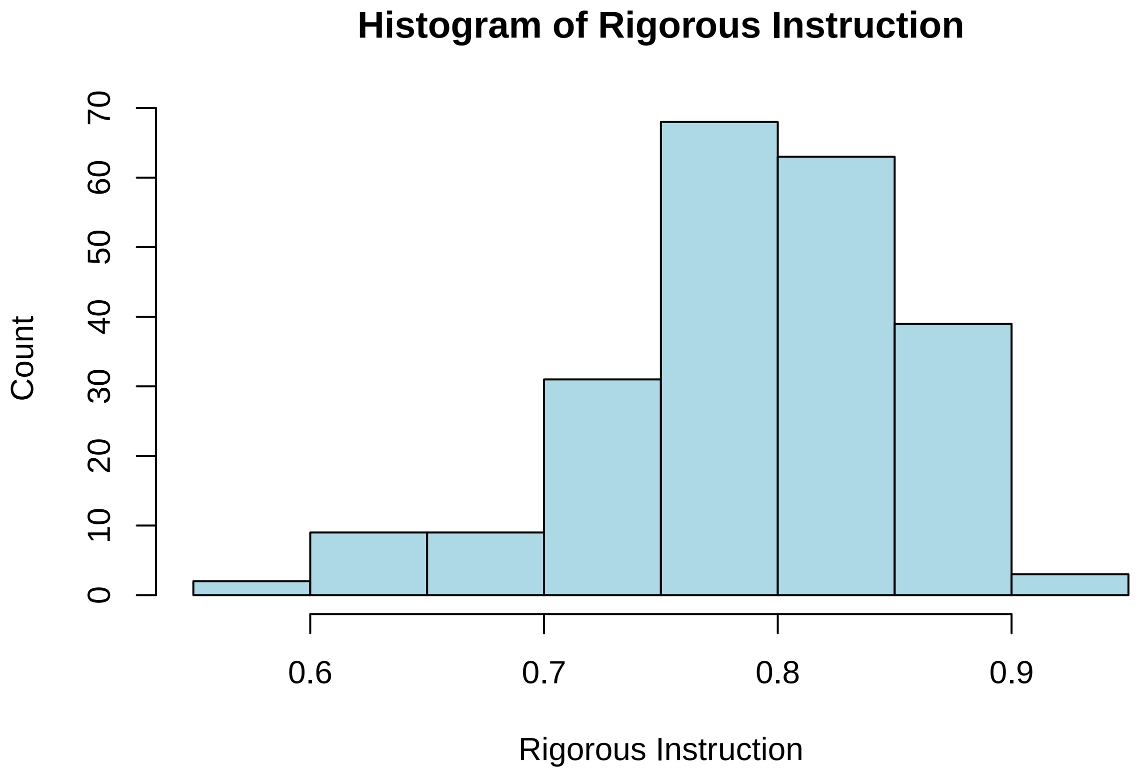
```
# Summary percent_of_students_chronically_absent
# Graphical
hist(cleaned_data$percent_of_students_chronically_absent,
main = "Histogram of Chronically Absent",
xlab = "Chronically Absent",
ylab = "Count",
col = "lightpink")
```



```r
# Numerical
summary(cleaned_data$percent_of_students_chronically_absent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0230  0.2520  0.3915  0.3746  0.4983  0.6890
```

```r
# Summary rigorous_instruction_percent_positive
# Graphical
hist(cleaned_data$rigorous_instruction_percent_positive,
main = "Histogram of Rigorous Instruction",
xlab = "Rigorous Instruction",
ylab = "Count",
col = "lightblue")
```



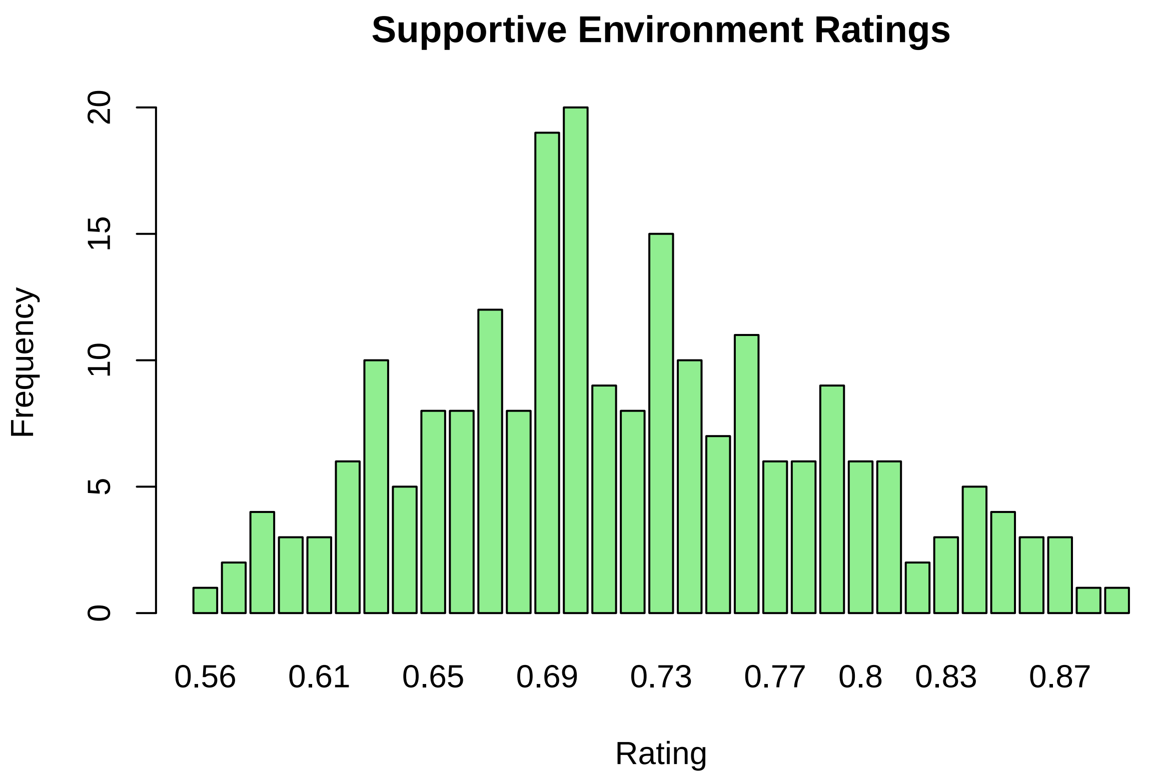
```r
# Numerical
summary(cleaned_data$rigorous_instruction_percent_positive)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5700  0.7600  0.8000  0.7949  0.8400  0.9500
```

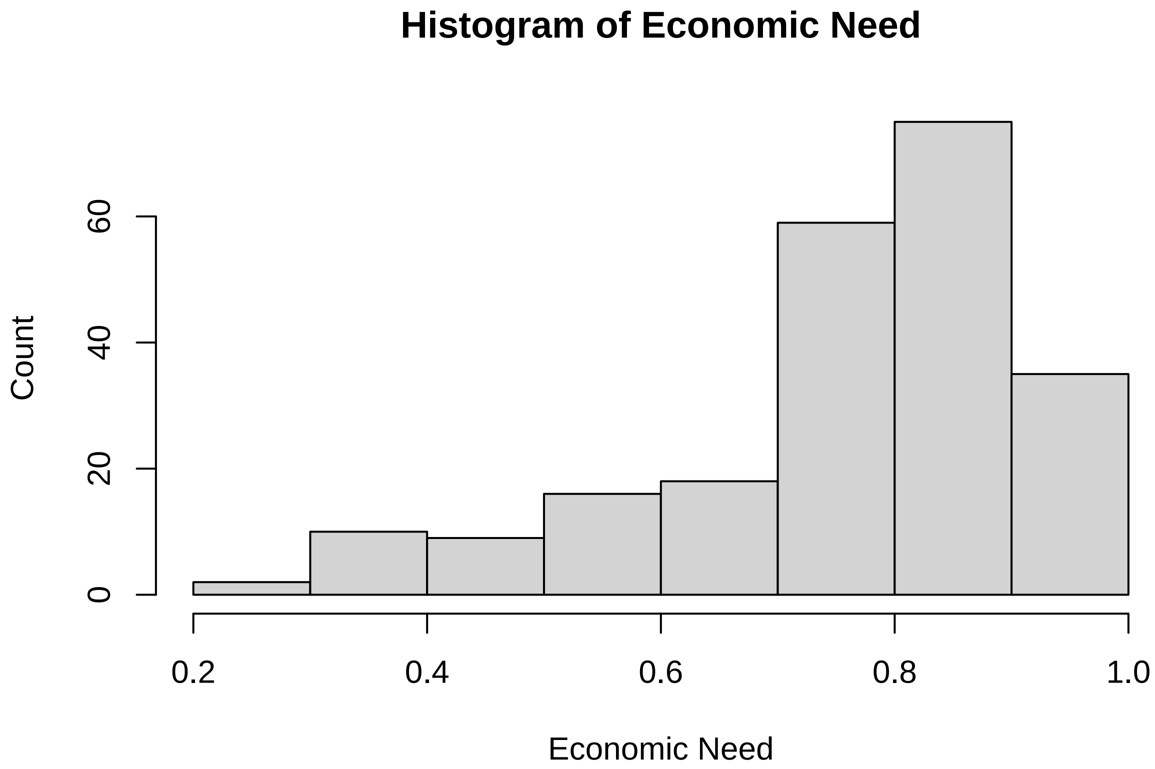
```r
# Summary supportive_environment_percent_positive
table(cleaned_data$supportive_environment_percent_positive)
```

```
##
## 0.56 0.57 0.58 0.59 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69  0.7 0.71 0.72
##    1    2    4    3    3    6   10    5    8    8   12    8   19   20    9    8
## 0.73 0.74 0.75 0.76 0.77 0.78 0.79  0.8 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.89
##   15   10    7   11    6    6    9    6    6    2    3    5    4    3    3    1
## 0.92
##    1
```

```r
barplot(table(cleaned_data$supportive_environment_percent_positive),
        main = "Supportive Environment Ratings",
        xlab = "Rating",
        ylab = "Frequency",
        col = "lightgreen")
```



```
# Summary economic_need_index
# Graphical
hist(cleaned_data$economic_need_index,
main = "Histogram of Economic Need", xlab = "Economic Need",
ylab = "Count")
```



```
# Numerical
summary(cleaned_data$economic_need_index)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.2380  0.7017  0.7975  0.7579  0.8675  0.9860
```

start

```
# --- Quick schema + completeness snapshot on your joined data frame (joint_data) ---
skimr::skim(joint_data)
```

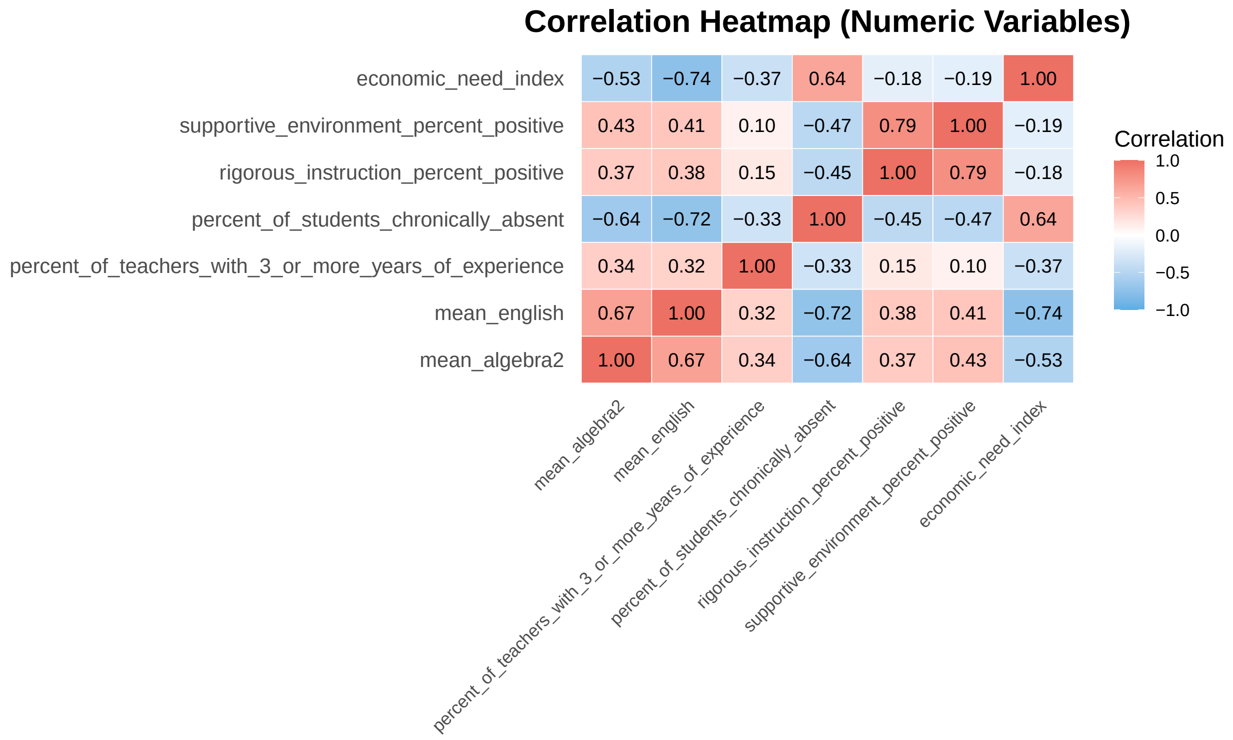
```
# --- Correlation matrix + heatmap (numeric only; pairwise-complete) ---

num_cols <- names(joint_data)[sapply(joint_data, is.numeric)]
num_df <- joint_data[, num_cols, drop = FALSE]
num_df <- num_df[, sapply(num_df, function(x) !all(is.na(x))), drop = FALSE]

if (ncol(num_df) >= 2) {
  cor_mat <- cor(num_df, use = "pairwise.complete.obs")

  cor_long <- as.data.frame(as.table(cor_mat))
  names(cor_long) <- c("x", "y", "r")
  library(ggplot2)

  ggplot(cor_long, aes(x, y, fill = r)) +
    geom_tile(color = "white") +
    geom_text(aes(label = sprintf("%.2f", r)),
              color = "black", size = 4) +
    scale_fill_gradient2(
      low = "#5DADE2", mid = "white", high = "#EC7063",
      midpoint = 0, limit = c(-1, 1), space = "Lab",
      name = "Correlation"
    ) +
    theme_minimal(base_size = 13) +
    theme(
      axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
      axis.text.y = element_text(size = 12),
      panel.grid = element_blank(),
      axis.title = element_blank(),
      plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
      legend.position = "right"
    ) +
    labs(title = "Correlation Heatmap (Numeric Variables)")
}
```



```
# --- Outlier scan (|z| > 3) across numeric columns ---
if (ncol(num_df) > 0) {
  z_df <- as.data.frame(scale(num_df))
  outlier_count <- sapply(z_df, function(x) sum(abs(x) > 3, na.rm = TRUE))
  outlier_tbl <- tibble::tibble(variable = names(outlier_count),
                                n_outliers = as.integer(outlier_count)) |>
    dplyr::arrange(dplyr::desc(n_outliers))
  print(head(outlier_tbl, 15))
}
```

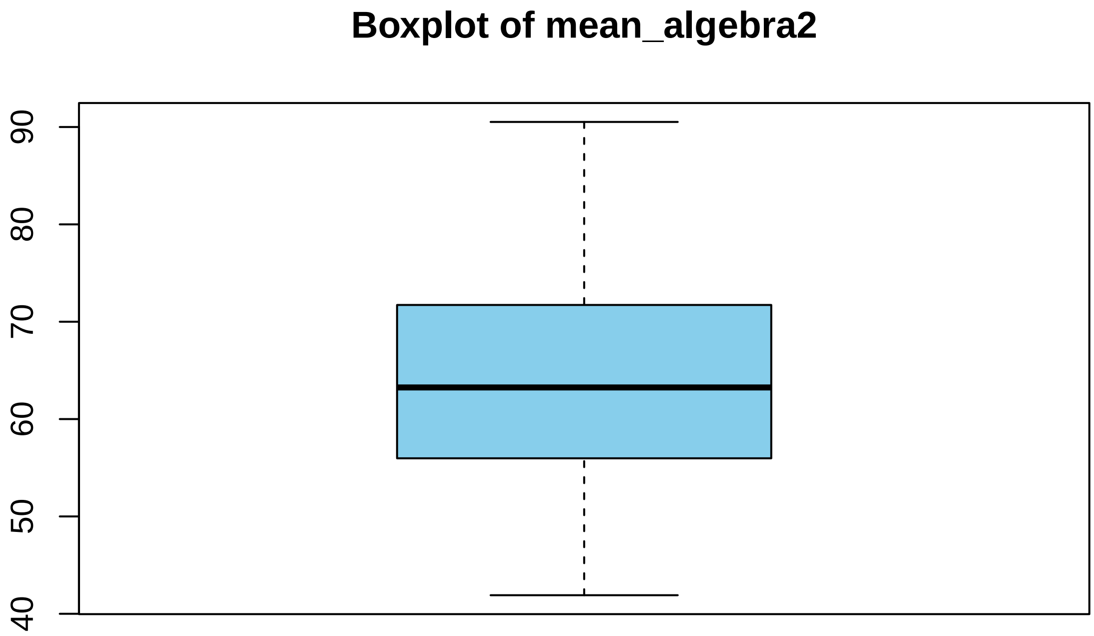
```
## # A tibble: 7 x 2
##   variable                                               n_outliers
##   <chr>                                                       <int>
## 1 percent_of_teachers_with_3_or_more_years_of_experience          2
## 2 economic_need_index                                             2
## 3 rigorous_instruction_percent_positive                           1
## 4 mean_algebra2                                                   0
## 5 mean_english                                                    0
## 6 percent_of_students_chronically_absent                          0
## 7 supportive_environment_percent_positive                         0
```

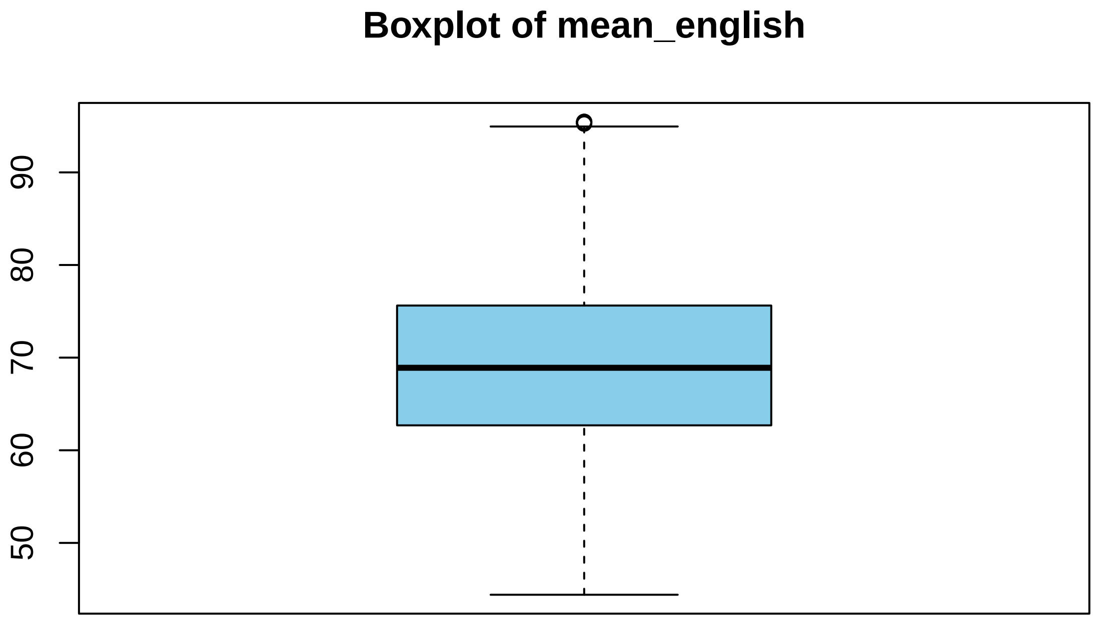
```
# --- Categorical profiling: top levels & shares (first 5 categorical cols) ---
cat_cols <- names(joint_data)[sapply(joint_data, function(x) is.character(x) || is.factor(x))]
for (c in head(cat_cols, 5)) {
  tab <- joint_data |>
    dplyr::count(.data[[c]], sort = TRUE, name = "n") |>
    dplyr::mutate(pct = scales::percent(n / sum(n)))
  print(glue::glue("Top levels for {c}:"))
  print(head(tab, 15))
}
```

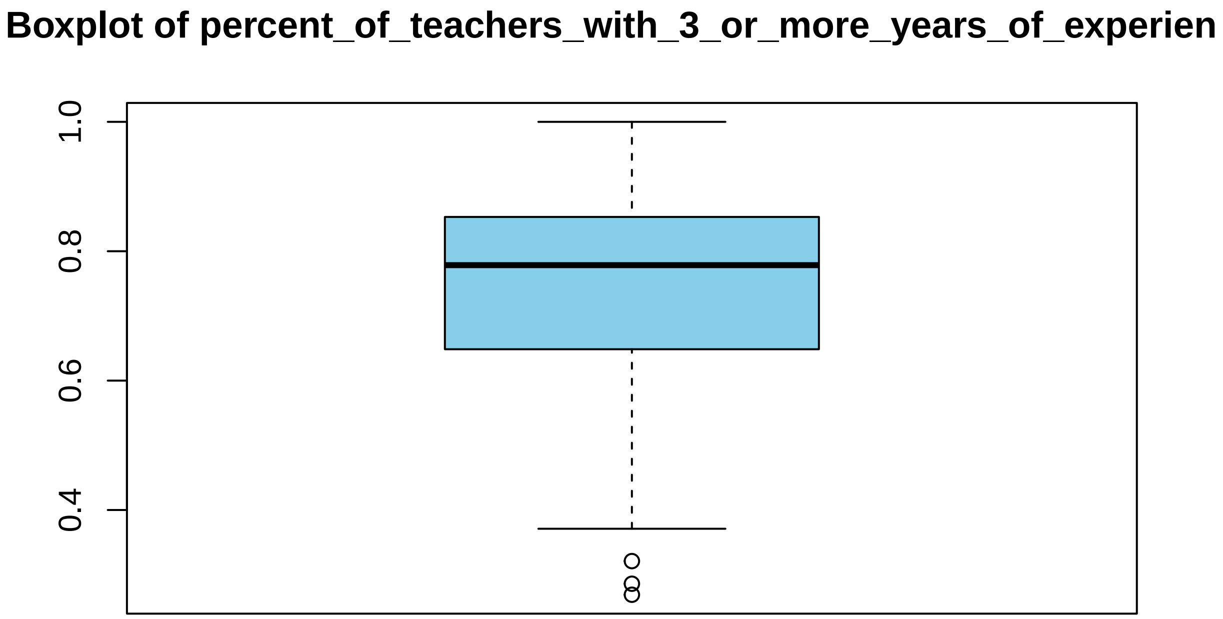
```
## Top levels for school_dbn:
## # A tibble: 15 x 3
##    school_dbn     n pct
##    <chr>      <int> <chr>
##  1 01M448         1 0%
##  2 01M509         1 0%
##  3 02M296         1 0%
##  4 02M298         1 0%
##  5 02M300         1 0%
##  6 02M305         1 0%
##  7 02M308         1 0%
##  8 02M316         1 0%
##  9 02M374         1 0%
## 10 02M392         1 0%
## 11 02M399         1 0%
## 12 02M400         1 0%
## 13 02M411         1 0%
## 14 02M412         1 0%
## 15 02M414         1 0%
## Top levels for school_name:
```

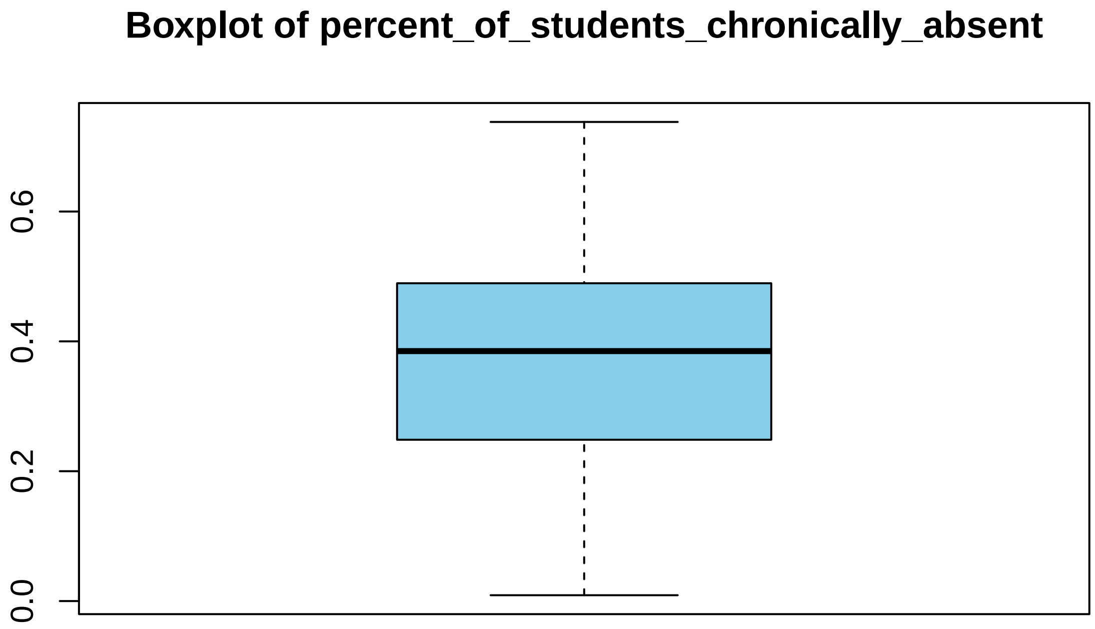
```
## # A tibble: 15 x 3
##    school_name                                            n pct
##    <chr>                                              <int> <chr>
##  1 New Visions Charter High School for Advanced Math      4 1.67%
##  2 New Visions Charter High School for the Humanities     3 1.25%
##  3 A. Philip Randolph Campus High School                  1 0.42%
##  4 ACORN Community High School                            1 0.42%
##  5 Abraham Lincoln High School                            1 0.42%
##  6 Academy for Conservation and the Environment           1 0.42%
##  7 Academy for Environmental Leadership                   1 0.42%
##  8 Academy for Language and Technology                    1 0.42%
##  9 Academy for Scholarship and Entrepreneurship: A Co     1 0.42%
## 10 Academy for Social Action                              1 0.42%
## 11 Academy of American Studies                            1 0.42%
## 12 Academy of Finance and Enterprise                      1 0.42%
## 13 Astor Collegiate Academy                               1 0.42%
## 14 Baruch College Campus High School                      1 0.42%
## 15 Bayside High School                                    1 0.42%
```

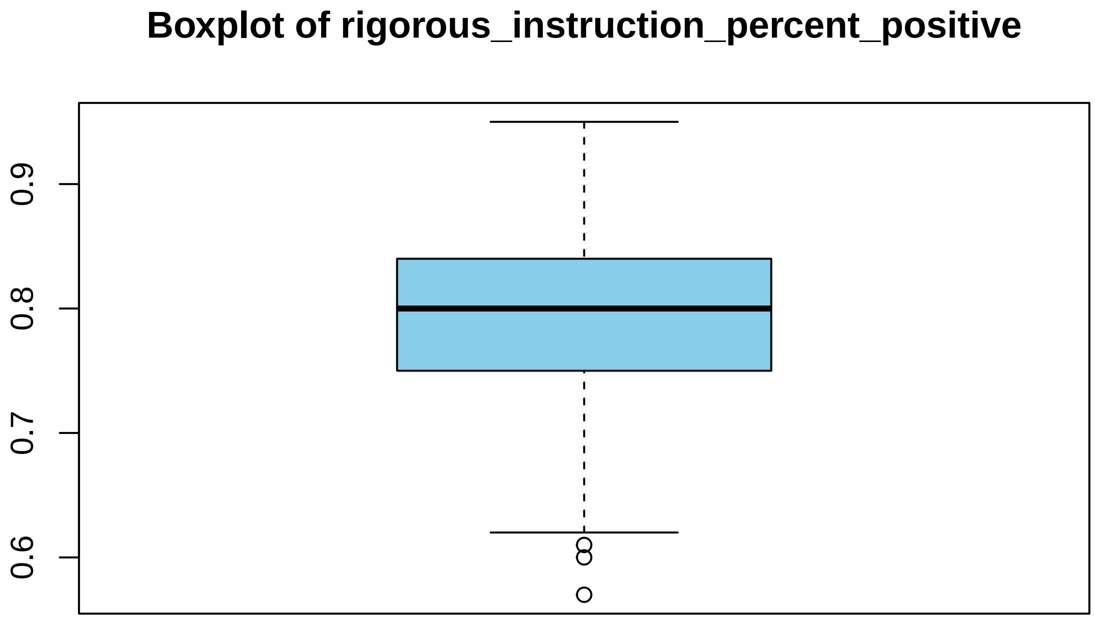
```
# Simple boxplots for numeric columns
numeric_cols <- names(joint_data)[sapply(joint_data, is.numeric)]
for (col in head(numeric_cols, 5)) {
  boxplot(joint_data[[col]], main = paste("Boxplot of", col), col = "skyblue")
}
```


Boxplot of mean_algebra2
90
80
70
60
50
40


Boxplot of mean_english
90
80
70
60
50


Boxplot of percent_of_teachers_with_3_or_more_years_of_experien
1.0
0.8
0.6
0.4


Boxplot of percent_of_students_chronically_absent
0.0
0.2
0.4
0.6


Boxplot of rigorous_instruction_percent_positive
0.9
0.8
0.7
0.6