

# Relationship between Test Scores and Population: Report

Jacob Blankenship

12/4/2025

General idea: First include a graph on distribution of good AP score's and maybe compare to to distribution of test takers, then make a graph comparing good AP score vs County population/Density to see if rural counties produce worse AP scorers. Finally create a linear model so see if county population/Density is a good predictor for AP exams.

## Report Section 1 - Introduction:

In general terms I wanted to explore how academic performance differs across communities in North Carolina. The state provides clean publicly available data on AP standardized test scores and population estimates, which allow for geographic comparison of educational outcomes. Because AP exams are an optional indicator of advanced coursework, counties with more resources or larger student populations may perform differently.

The specific variables I was looking for were: 1. County Names 2. Number of AP Exams taken by County 3. Percent of passed AP Exams by County (3/5 or higher) 4. Longitude and Latitude coordinates of counties to create map plots

My Research question is: "Does population size of a North Carolina county predict AP exam success rates?"

Links to source are given below.

<https://www.osbm.nc.gov/facts-figures/population-demographics/state-demographer/county-population-estimates/certified-county-population-estimates>

<https://www.dpi.nc.gov/2025-ap-test-results>

```
# initial packages install
packages <- c("ggplot2", "dplyr", "readxl", "maps", "scales", "sf", "ggrepel", "tidymodels")
# only install packages I haven't installed
missing <- packages[!(packages %in% installed.packages()[, "Package"])]
if (length(missing) > 0) install.packages(missing)

# load datasets + add NC map data (for polygons)
invisible(lapply(packages, library, character.only = TRUE))
#https://www.dpi.nc.gov/2025-ap-test-results
NCScoreData <- read_excel("Import2025NC.AP.RESULTS.xlsx")
NCPopData <- read_excel("NCPopulationCensusData.xlsx")
NCMapData <- map_data( "county",
                       region = "north carolina" )
```

## Initial Imports

## Report Section 2 - Data Analysis:

I wanted my explanatory variable to be the population and my predictor variable to be the AP exam proficient percent.

Before any modeling or visualization, the datasets required extensive cleaning to ensure valid comparisons between maps and score data. This included removing duplicate regions, converting character fields to numeric values, standardizing county names, and computing geometric centroids for map labeling. It should also be noted that I edited the .xlsx files that I directly downloaded as some small formatting changes made the file much more workable in r.

```
# Replace 3rd column with correct name, filter out non schools, filter out '*'s in dataset, convert table
NCScoreData <- NCScoreData %>% rename_at(vars(3), ~ "County") %>% filter(is.na(`School System & School`))
  select(
    `2025`,
    County,
    `# of Exams Taken5`,
    `# of Exams with Scores of 3 or Higher6`,
    `% of Exams with Scores of 3 or Higher7`
  ) %>% mutate(
    Exams = case_when(`# of Exams Taken5` == "*" ~ "", TRUE ~ `# of Exams Taken5`),
    GoodScore = case_when(
      `# of Exams with Scores of 3 or Higher6` == "*" ~ "",
      TRUE ~ `# of Exams with Scores of 3 or Higher6`
    ),
    Percentage = case_when(
      `% of Exams with Scores of 3 or Higher7` == "*" ~ "",
      TRUE ~ `% of Exams with Scores of 3 or Higher7`
    )
  ) %>% rename_at(vars(1), ~ "ID") %>% select(ID,County,Exams,GoodScore,Percentage) %>% filter(ID != "7")
# ^ SPECIFICALLY TAKING OUT A DUPLICATE POINT. EMAILED GOV TO CLARIFY IN MEANTIME

# Setting chars to lower, and taking out County from fields to
# make values match with map data
NCScoreData$County <- gsub(" County", "", NCScoreData$County)
NCScoreData$County <- tolower(NCScoreData$County)

# converting nums to num types (for graph)
NCScoreData$Percentage <- as.numeric(as.character(NCScoreData$Percentage))
NCScoreData$Exams <- as.numeric(as.character(NCScoreData$Exams))
NCScoreData$GoodScore <- as.numeric(as.character(NCScoreData$GoodScore))

#setequal(unique(NCMapData$subregion), unique(NCScoreDataClean$County))

# Left join to get points added to main data set
NCScoreDataClean <- left_join(NCScoreData, NCMapData, by = c('County' = 'subregion'))

# Manually put in centroid formula for accurate centered points (package wasn't working)
CountyCenters <- NCMapData %>%
  group_by(subregion, group) %>%
  summarise(
```

```

A = 0.5 * sum(
  long * lead(lat, default = first(lat)) - lead(long, default = first(long)) * lat
),
long_c = sum((long + lead(
  long, default = first(long)
)) *
(
  long * lead(lat, default = first(lat)) - lead(long, default = first(long)) * lat
) / (6 * A),
lat_c = sum((lat + lead(lat, default = first(
  lat
)))) *
(
  long * lead(lat, default = first(lat)) - lead(long, default = first(long)) * lat
) / (6 * A)
) %>%
group_by(subregion) %>%
summarise(long = mean(long_c), lat = mean(lat_c)) %>%
ungroup()

# Adding in extra values to make points more informative (add filtering capability)
CountyCenters <- CountyCenters %>%
  left_join(NCScoreData %>%
              select(County, Percentage, Exams, GoodScore),
              by = c("subregion" = "County"))
# Reconvert to have first letter uppercase to make it look nice
CountyCenters$subregion <- paste0(
  toupper(substr(CountyCenters$subregion, 1, 1)),
  substr(CountyCenters$subregion, 2, nchar(CountyCenters$subregion)))
)
# taking out non-county points + left_join to add population data to main dataset
NCPopData <- NCPopData %>% filter(County != "State of North Carolina")
CountyCenters <- CountyCenters %>%
  left_join(NCPopData %>% select(County, Population), by = c("subregion" = "County"))
# label best and worst counties
CountyCenters <- CountyCenters %>%
  mutate(
    Highlight = case_when(
      min_rank(Percentage) <= 5 ~ "Bottom 5", # 5 lowest percentages
      min_rank(desc(Percentage)) <= 5 ~ "Top 5", # 5 highest percentages
      TRUE ~ "Other"
    )
  )

```

**Data Cleaning** Now that my data is clean I can do general summary statistics by County.

```

# compute summary stats
CountyCenters %>%
  summarise(
    count = n(),
    # Population

```

```

mean_population = mean(Population, na.rm = TRUE),
sd_population = sd(Population, na.rm = TRUE),
min_population = min(Population, na.rm = TRUE),
max_population = max(Population, na.rm = TRUE),
# Percentage of good AP scores
mean_percentage = mean(Percentage, na.rm = TRUE),
sd_percentage = sd(Percentage, na.rm = TRUE),
min_percentage = min(Percentage, na.rm = TRUE),
max_percentage = max(Percentage, na.rm = TRUE),
# Number of exams
mean_exams = mean(Exams, na.rm = TRUE),
sd_exams = sd(Exams, na.rm = TRUE),
min_exams = min(Exams, na.rm = TRUE),
max_exams = max(Exams, na.rm = TRUE)
)

```

## Summary Statistics

```

## # A tibble: 1 x 13
##   count mean_population sd_population min_population max_population
##   <int>      <dbl>          <dbl>          <dbl>          <dbl>
## 1    100     109831.       187200.        3477     1235748
## # i 8 more variables: mean_percentage <dbl>, sd_percentage <dbl>,
## #   min_percentage <dbl>, max_percentage <dbl>, mean_exams <dbl>,
## #   sd_exams <dbl>, min_exams <dbl>, max_exams <dbl>

```

I needed a better way of showing the viewer what this data should look like. To combat this (especially to people that are unfamiliar with the AP exam or North Carolina), I wanted to create a weighted mapped plot of exam scores. This way people can get a representation of population density and overall performance in exams across the county before my model is created.

```

# Labs + Theme
JacobLabs1 <- labs(title = "North Carolina's Percent of Passed AP Exams by County",
                     subtitle = "2025 AP Exam Results",
                     caption = "NA value are gray, Data comes from www.dpi.nc.gov")
JacobTheme1 <- theme_bw() + theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    text = element_text(face = "bold"))
)

# Making initial map plot
NCScoreDataClean %>%
  ggplot(aes(
    x = long,
    y = lat,

```

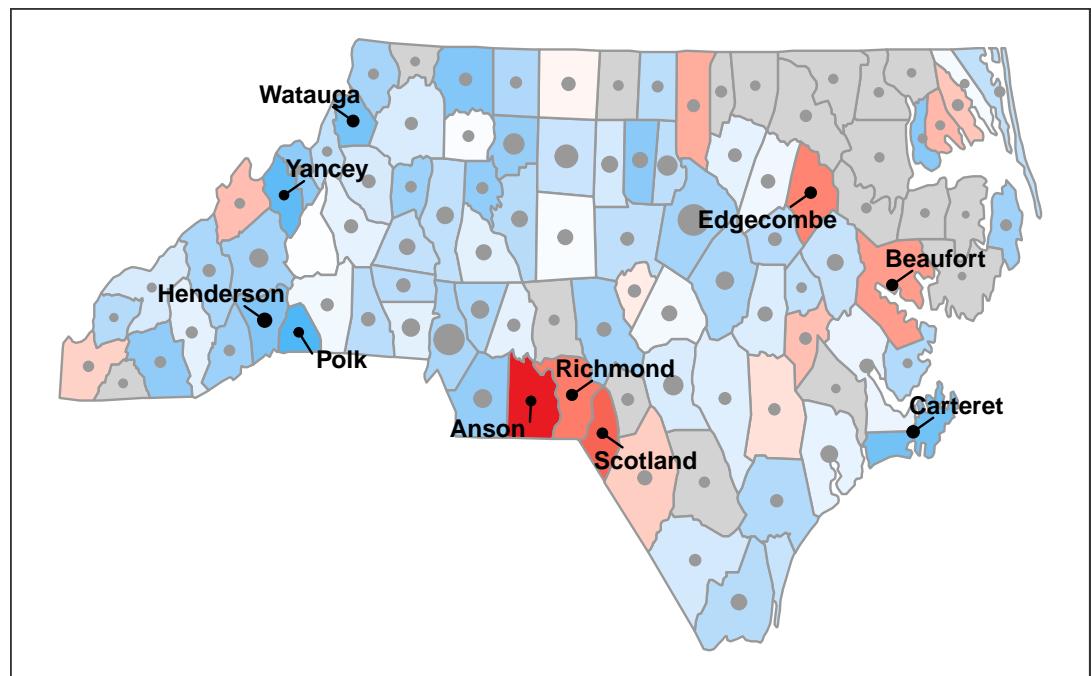
```

group = group,
  fill = Percentage
)) + # add polygons to create NC state map
geom_polygon(color = "#999999") +
scale_fill_gradient2(
  low = "#E81B23",
  mid = "white",
  high = "#00AEF3",
  midpoint = 50,
  na.value = "lightgray"
) + geom_point(
  data = CountyCenters,
  aes(x = long, y = lat, size = Population, color = Highlight),
  inherit.aes = FALSE
) + # label focuses points with correct colors
scale_color_manual(values = c("Top 5" = "black", "Bottom 5" = "black", "Other" = "#999999")) + guides
geom_text_repel( # add names to focuses counties
  data = CountyCenters %>% filter(Highlight != "Other"),
  aes(label = subregion, x = long, y = lat),
  color = "black",
  size = 4,
  inherit.aes = FALSE,
  fontface = "bold",
  max.overlaps = Inf,
  box.padding = 0.5,
  point.padding = 0.1
) +
# Points for all counties, color by highlight
JacobLabs1 + JacobTheme1

```

## North Carolina's Percent of Passed AP Exams by County

2025 AP Exam Results



Visualization #1

NA value are gray, Data comes from [www.dpi.nc.gov](http://www.dpi.nc.gov)

This visualization highlights both the spatial variation in AP performance and the disparity in the number of test takers among counties, which helps motivate whether population should reasonably act as a predictor.

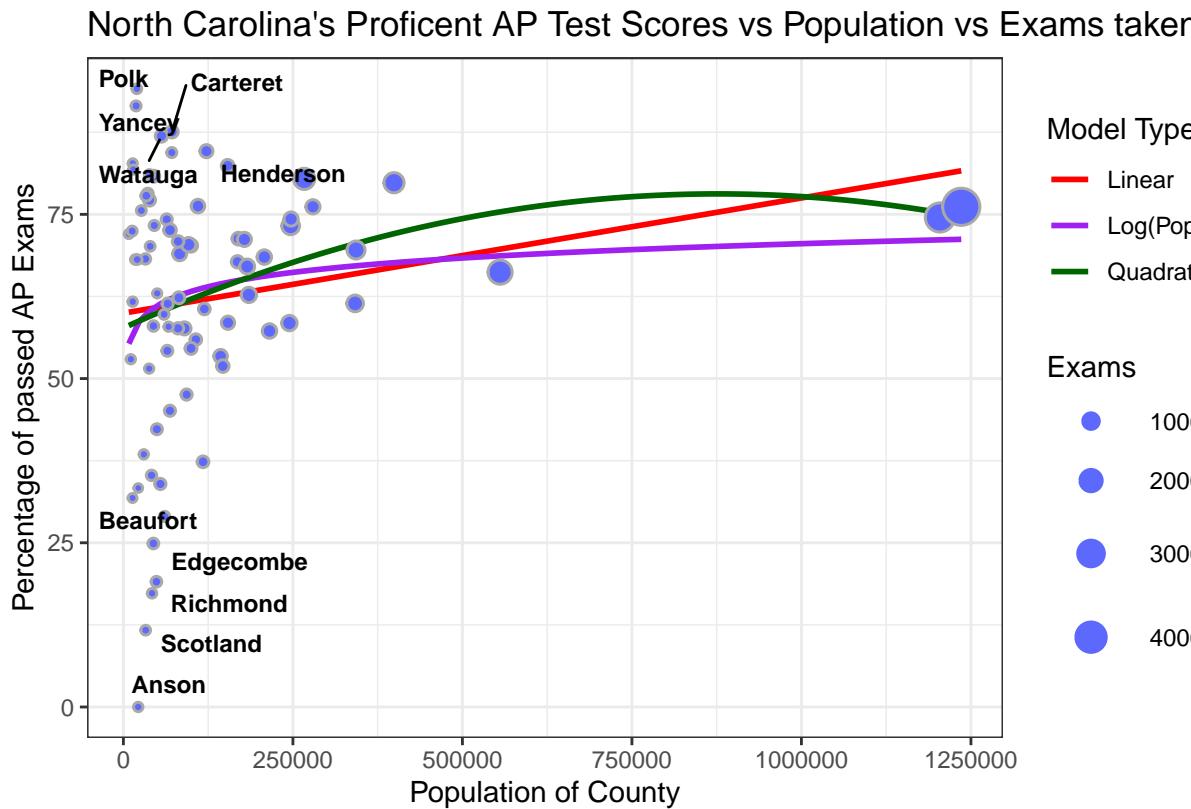
I also wanted to understand whether AP performance changes with community size, I used population as an explanatory variable and AP score pass rate as the response variable. I then fit linear, log-linear, and polynomial models to compare whether the relationship is linear or non-linear, and if the correlational coefficient was better for said mappings.

```
# Labs + Theme
JacobLabs2 <- labs(title = "North Carolina's Proficient AP Test Scores vs Population vs Exams taken",
  JacobTheme2 <- theme_bw()
  CountyCenters %>% # taking out n/As
  filter(!is.na(Population), !is.na(Percentage), !is.na(Exams)) %>%
  ggplot(aes(x = Population, y = Percentage, size = Exams)) +
  
  # Linear
  geom_smooth(method = "lm", se = FALSE, aes(color = "Linear"), linetype = "solid") +
  
  # Log/Linear
  geom_smooth(method = "lm", formula = y ~ log(x), se = FALSE, aes(color = "Log(Pop)"), linetype = "solid") +
  
  # Quadratic
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = FALSE, aes(color = "Quadratic"), linetype = "solid") +
  
  # Points
  geom_point(shape = 21, color = "darkgray", fill = "#5d68fc", stroke = 0.8) +
```

```

# Top/bottom 5 labels
geom_text_repel(
  data = CountyCenters %>%
    filter(!is.na(Percentage)) %>%
    arrange(Percentage) %>%
    slice(c(1:5, (n() - 4):n())),
  aes(label = subregion, x = Population, y = Percentage),
  color = "black",
  size = 3,
  inherit.aes = FALSE,
  fontface = "bold",
  max.overlaps = Inf,
  box.padding = 0.3,
  point.padding = 0.1
) +
  # Labels & theme application (also adding in colors for model lines)
  JacobLabs2 + JacobTheme2 +
  scale_color_manual(name = "Model Type", values = c("Linear" = "red", "Log(Pop)" = "purple", "Quadratic" = "green"),
                     size = guide_legend(),
                     override.aes = list(shape = 21, fill = "#5d68fc", color = "white", stroke = 0.8))
)
)

```



Visualization #1

```

# create model data with non-n/a values
model_data <- CountyCenters %>%
  filter(!is.na(Population), !is.na(Percentage), Population > 0, Percentage > 0)

# Linear model
LinReg <- linear_reg() %>%
  set_engine("lm") %>%
  fit(Percentage ~ Population, data = model_data)

# Log/linear model
LogPopReg <- linear_reg() %>%
  set_engine("lm") %>%
  fit(Percentage ~ log(Population), data = model_data)

# Polynomial model
PolyReg <- linear_reg() %>%
  set_engine("lm") %>%
  fit(Percentage ~ Population + I(Population^2), data = model_data)

# Show coefficients
LinReg %>% tidy()

```

```

## # A tibble: 2 x 5
##   term      estimate std.error statistic p.value
##   <chr>     <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 61.0      2.37      25.7  1.62e-39
## 2 Population  0.0000155 0.00000987    1.57  1.21e- 1

```

```
LogPopReg %>% tidy()
```

```

## # A tibble: 2 x 5
##   term      estimate std.error statistic p.value
##   <chr>     <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 37.1      21.8      1.70  0.0927
## 2 log(Population) 2.32      1.94      1.20  0.234

```

```
PolyReg %>% tidy()
```

```

## # A tibble: 3 x 5
##   term      estimate std.error statistic p.value
##   <chr>     <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 5.93e+ 1  3.20e+ 0    18.5  6.15e-30
## 2 Population  3.67e- 5  2.90e- 5    1.26  2.10e- 1
## 3 I(Population^2) -1.94e-11 2.49e-11   -0.777 4.40e- 1

```

```
# Show R-squared
glance(LinReg)$r.squared
```

```
## [1] 0.03097748
```

```
glance(LogPopReg)$r.squared
```

```
## [1] 0.0183111
```

```
glance(PolyReg)$r.squared
```

```
## [1] 0.03861632
```

The map above shows that while high population counties like Wake and Mecklenburg do have above average pass rates, much smaller counties such as Polk or Carteret outperform them significantly, suggesting that population may not drive quality.

**Linear:**

County Exam Percentage =  $60.99 + 0.0000155 \times \text{Population}$

$R^2 = 0.03097748$

**Logarithmic:**

County Exam Percentage =  $37.05 + 2.32 \times \log(\text{Population})$

$R^2 = 0.0183111$

**Exponential/Polynomial:**

County Exam Percentage =  $59.33 + (3.67 \times 10^{-5}) \times \text{Population} - (1.94 \times 10^{-11}) \times \text{Population}^2$

$R^2 = 0.03861632$

Overall there is little to no linear, logarithmic, or polynomial( $n^2$ ) correlational relationship between Population of County and County AP Exam Percentage. The best model, a polynomial regression, only slightly improved fit, suggesting no meaningful predictive relationship with our current dataset.

Critiques:

1. Several counties reported insufficient AP testing data, leading to missing values for roughly 20% of regions. This may bias estimates toward larger school systems with more complete reporting.
2. Census estimates are from 2024, while AP results are from 2025; demographic changes within a year may reduce precision in population-based modeling.
3. Things such as population density within counties is not taken into account. So if a very large school is shown in a very population low county, it would still flag as a small data point. A more accurate marker to test may be average or median population in highschools of each county.