---

title: "datathon"

author: "Monique Sierra"

date: "10/28/2023"

output:

pdf\_document: default

---

```{r setup, include=FALSE}

library(dplyr)

library(readxl)

library(ggplot2)

leaCharCopy <- LEA.Characteristics

edgeCopy <- EDGE\_GEOCODE\_PUBLICLEA\_1718

advMathCopy <- Advanced.Mathematics

ussd17Copy <- ussd17

tractCopy <- grf17\_lea\_tract

addressBlockCountList <- tract\_kfr\_rP\_gP\_pall

```

```{r, include=FALSE}

difference1 <- setdiff(leaCharCopy$LEAID, edgeCopy$LEAID)

completeDifferenecs <- (leaCharCopy$LEAID == difference1)

head(completeDifferenecs)

```

# minor data improvements

```{r}

# adding first two numbers to match on IDs

ussd17Copy$completeAID <- paste(ussd17Copy$`State FIPS Code`, ussd17Copy$`District ID`, sep = "")

```

```{r}

# 3. Plot the number of students per school district vs. the number of students

# in poverty for all school districts in the US. Explore how all subject matters,

# which you have now imported into PostgreSQL, are related to poverty. Do

# data aggregation and query in PostgreSQL. Plot in your preferred

# programming language.

# the number of students/school district =

```

## joining to solve Q3

```{r}

# joining together the poverty and ID from edge and lea. doing these first bc their longer IDs match. edge has abbrev one added for future join

joiningOnNameForPoverty <- inner\_join(edgeCopy, leaCharCopy, by = "LEAID")

# joining together on shortened ID with other two tables

threeTables <- left\_join(ussd17Copy, joiningOnNameForPoverty, by = c("completeAID"= "LEAID"))

print(head(threeTables))

```

# grouping by mf something idk yet

```{r}

# look at everything as one eeks

print(head(joiningOnNameForPoverty))

#change some ugly names (do earlier if possible)

colnames(threeTables)[6] <- "5\_17pop"

colnames(threeTables)[7] <- "5\_17povertyPop"

# subsetting for main things i want

selected\_columns <- threeTables[c("completeAID", "5\_17pop", "5\_17povertyPop", "STATE")]

print(head(selected\_columns))

```

```{r}

# Create a scatter plot of the ratio

selected\_columns$poverty\_to\_population\_ratio <- selected\_columns$`5\_17povertyPop` / selected\_columns$`5\_17pop`

# we changed it to a percentage to account for the population to account for the relative poverty to population

ggplot(selected\_columns, aes(x = `5\_17pop`, y = `5\_17povertyPop`, label = "poverty\_to\_population\_ratio")) +

geom\_point() +

labs(x = "Population Size", y = "Amount of Poverty", size = "Poverty/Population Ratio")

```

```{r}

ggplot(selected\_columns, aes(x = `5\_17pop`, y = `5\_17povertyPop`)) +

geom\_point(size = .5) +

geom\_smooth(method = "lm", se = TRUE) + # Add a linear regression line without confidence intervals

labs(

x = "Population Size",

y = "Amount of Poverty",

title = "Scatterplot with Linear Regression Line"

)

lm\_model <- lm(`5\_17povertyPop` ~ `5\_17pop`, data = selected\_columns)

summary(lm\_model)

# Get the correlation coefficient

correlation <- cor(selected\_columns$`5\_17pop`, selected\_columns$`5\_17povertyPop`)

# Get the residuals

residuals <- resid(lm\_model)

# Print the correlation coefficient

cat("Correlation coefficient:", correlation, "\n")

```

# combining tract download and the district file

```{r}

# changed the location one to show the tract as a character instead of number

addressBlockCountList$tract <- as.character(addressBlockCountList$tract)

# joined the school district with income data by tract

tractAndSchoolDistrict <- inner\_join(tractCopy, addressBlockCountList, by = c("TRACT"= "tract"))

print(head(tractAndSchoolDistrict))

# combing the tract and lat/long file

```

Select a location you are familiar with. Go to the Opportunity Atlas. Down-

load the data for that location. It will have the census tract as a geographicID.

Use the SDGR database to map school districts to census tracts and

determine whether the income reported in the Opportunity Atlas has correlation with academic performance in high school.

```{r}

install.packages("plotly")

library(plotly)

# Create a scatter plot using plotly

scatter\_plot <- plot\_ly(data = selected\_columns, x = ~`5\_17pop`, y = ~`5\_17povertyPop`, text = ~completeAID,

mode = "markers", marker = list(size = 8, opacity = 0.6))

# Customize the layout

scatter\_plot <- scatter\_plot %>% layout(

xaxis = list(title = "Population Size"),

yaxis = list(title = "Amount of Poverty"),

title = "Scatter Plot with Tooltips"

)

# Show the plot

scatter\_plot

```

# colors of scatterolot by poverty quart

```{r}

## just tx

selected\_columns1 <- selected\_columns %>%

filter(STATE == "TX")

# graphing with scaterplot

smoothedLineTX<- ggplot(selected\_columns, aes(x = `5\_17pop`, y = `5\_17povertyPop`)) +

geom\_point(size = .5) +

geom\_smooth(method = "lm", se = TRUE) + # Add a linear regression line without confidence intervals

labs(

x = "District Population",

y = "Number of People in Poverty",

title = "Total Population v Poverty Correlation"

)

# getting linear model stats

lm\_model <- lm(`5\_17povertyPop` ~ `5\_17pop`, data = selected\_columns1)

summary(lm\_model)

# Define colors for income levels

income\_colors <- c("green", "yellow", "orange", "red")

# Create a scatter plot

scatter\_plot <- ggplot(selected\_columns, aes(x = `5\_17pop`, y = `5\_17povertyPop`, color = income\_level)) +

geom\_point(size = .5) +

scale\_color\_manual(values = income\_colors) +

labs(

x = "District Population",

y = "Number of People in Poverty",

title = "Total Population v Poverty by Income Level"

)

# Specify custom limits for the x and y axes (adjust as needed)

custom\_x\_limits <- c(0, 100000) # Set the x-axis limits

custom\_y\_limits <- c(0, 20000) # Set the y-axis limits

# Apply custom axis limits using coord\_cartesian

smoothedLineTX

scatter\_plot + coord\_cartesian(xlim = custom\_x\_limits, ylim = custom\_y\_limits)

```

```{r}

#getting the total number of people in advanced math male and female

Advanced\_Place$totalAdvancedEnrollment <- (Advanced\_Place$TOT\_APEXAM\_ONEORMORE\_M + Advanced\_Place$TOT\_APEXAM\_ONEORMORE\_F)

print(head(Advanced\_math$totalAdvancedEnrollment))

advancedPlacementClassesAndIncome <- inner\_join(Advanced\_Place, threeTables, by = c("LEAID"= "completeAID"))

advancedPlacementClassesAndIncome$percentageOfAdvClasses <- (advancedPlacementClassesAndIncome$totalAdvancedEnrollment / advancedPlacementClassesAndIncome$"5\_17pop" )

print(head(advancedPlacementClassesAndIncome$percentageOfAdvClasses))

```

```{r}

advancedPlacementClassesAndIncome$percentInPoverty <- (advancedPlacementClassesAndIncome$)

# Assuming you have the two columns in your data frame

percentAdvScatter <- ggplot(advancedPlacementClassesAndIncome, aes(x = percentageOfAdvClasses, y = totalAdvancedEnrollment)) +

geom\_point() +

labs(

x = "Percentage of Advanced Classes",

y = "Poverty to Population Ratio",

title = "Scatter Plot of Percentage of Advanced Classes vs. Poverty to Population Ratio"

)

# Specify custom limits for the x and y axes (adjust as needed)

custom\_x\_limits <- c(0, 1000) # Set the x-axis limits

custom\_y\_limits <- c(0, 10000) # Set the y-axis limits

# Apply custom axis limits using coord\_cartesian

percentAdvScatter + coord\_cartesian(xlim = custom\_x\_limits, ylim = custom\_y\_limits)

```A graph showing a number of colored dots

Description automatically generatedA graph showing a line of poverty

Description automatically generated with medium confidence