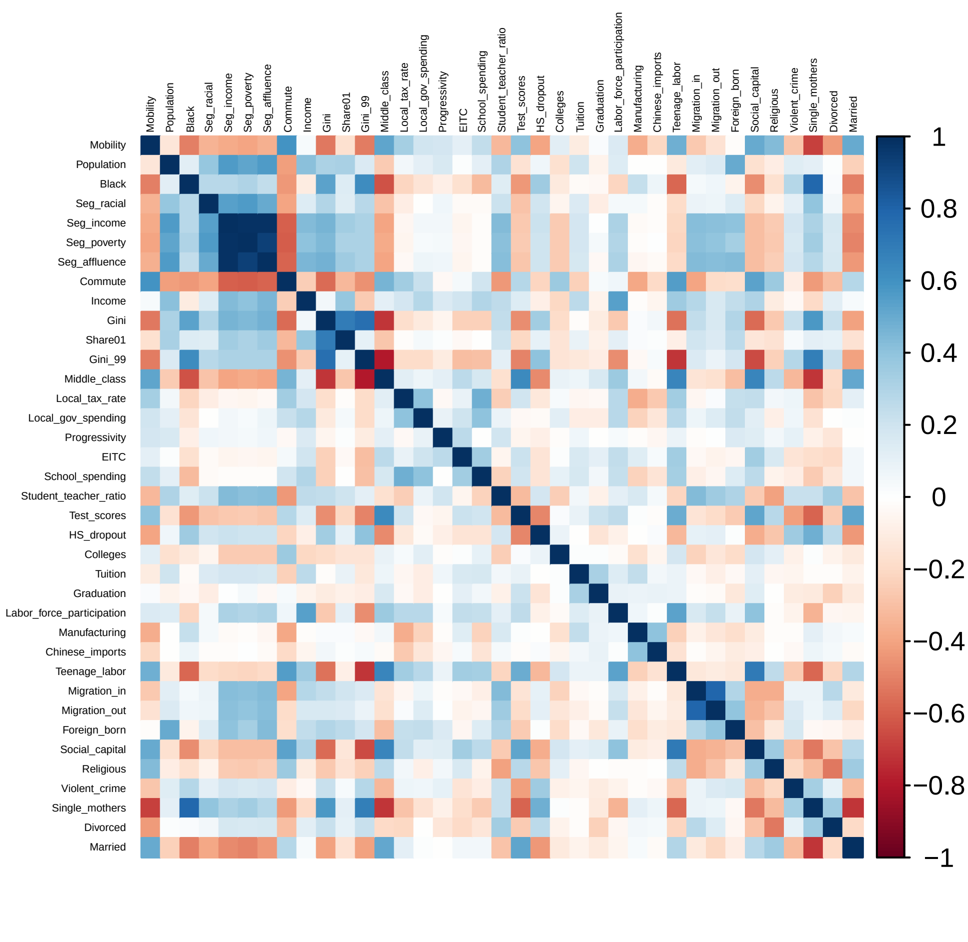
DASC32103Project1-WIlliamBuckey

2025-02-05

```r
# Step 1: Remove non-informative numeric columns (ID, Longitude, Latitude)
cleaned_data <- mobility_data %>%
  select(-c(ID, Longitude, Latitude))

# Step 2: Remove columns with ANY missing values
cleaned_data <- mobility_data %>%
  select(where(is.numeric)) %>%  # Keeps only numeric variables
  select(-c(ID, Longitude, Latitude))  # Explicitly remove ID & coordinates

# Step 3: Compute correlation matrix
cor_matrix <- cor(cleaned_data, use = "pairwise.complete.obs")

# Step 4: Create the full heatmap (WITHOUT numbers)
corrplot(cor_matrix,
         method = "color",  # Color-coded correlation plot
         tl.col = "black",  # Black text labels
         tl.cex = 0.3)      # Adjust text size for readability
```



```r
cor_df <- as.data.frame(as.table(cor_matrix))

# Step 3: Remove self-correlations (where variable == variable)
cor_df <- cor_df %>%
  filter(Var1 != Var2)  # Exclude diagonal (self-correlation)

# Step 4: Sort by absolute correlation strength (highest to lowest)
top_corr <- cor_df %>%
  arrange(desc(abs(Freq))) %>%  # Sort by absolute correlation
  head(50)  # Select top 30

# Step 5: Print top 50 correlated variable pairs
print(top_corr)
```

```
##              Var1          Var2       Freq
## 1   Seg_affluence    Seg_income  0.9857398
## 2      Seg_income Seg_affluence  0.9857398
## 3     Seg_poverty    Seg_income  0.9806223
## 4      Seg_income   Seg_poverty  0.9806223
## 5   Seg_affluence   Seg_poverty  0.9387360
## 6     Seg_poverty Seg_affluence  0.9387360
## 7    Middle_class       Gini_99 -0.7951413
## 8         Gini_99  Middle_class -0.7951413
## 9   Migration_out  Migration_in  0.7929604
## 10   Migration_in Migration_out  0.7929604
## 11 Single_mothers         Black  0.7810011
## 12          Black Single_mothers  0.7810011
## 13        Gini_99          Gini  0.7532210
## 14           Gini       Gini_99  0.7532210
## 15        Married Single_mothers -0.7158522
## 16 Single_mothers       Married -0.7158522
## 17   Middle_class          Gini -0.7149591
## 18           Gini  Middle_class -0.7149591
## 19  Teenage_labor       Gini_99 -0.7146509
## 20        Gini_99  Teenage_labor -0.7146509
## 21 Single_mothers  Middle_class -0.7112846
## 22   Middle_class Single_mothers -0.7112846
## 23 Social_capital  Teenage_labor  0.7081949
## 24  Teenage_labor Social_capital  0.7081949
## 25        Share01          Gini  0.6974718
## 26           Gini       Share01  0.6974718
## 27 Single_mothers      Mobility -0.6858853
## 28       Mobility Single_mothers -0.6858853
## 29 Single_mothers       Gini_99  0.6831614
## 30        Gini_99 Single_mothers  0.6831614
## 31  Teenage_labor  Middle_class  0.6584454
## 32   Middle_class  Teenage_labor  0.6584454
## 33 Social_capital       Gini_99 -0.6561782
## 34        Gini_99 Social_capital -0.6561782
## 35 Social_capital  Middle_class  0.6516978
## 36   Middle_class Social_capital  0.6516978
## 37   Middle_class         Black -0.6379982
## 38          Black  Middle_class -0.6379982
## 39    Test_scores  Middle_class  0.6378213
```

```
## 40   Middle_class    Test_scores  0.6378213
## 41        Gini_99          Black  0.6288560
## 42          Black        Gini_99  0.6288560
## 43        Commute    Seg_poverty -0.6026864
## 44    Seg_poverty        Commute -0.6026864
## 45        Commute     Seg_income -0.5992370
## 46     Seg_income        Commute -0.5992370
## 47        Commute       Mobility  0.5906339
## 48       Mobility        Commute  0.5906339
## 49        Commute  Seg_affluence -0.5801970
## 50  Seg_affluence        Commute -0.5801970
```

```r
# Load required libraries
library(dplyr)

# Step 1: Define policy-driven variables
policy_vars <- c("Local_tax_rate", "Local_gov_spending", "Progressivity", "EITC",
                 "School_spending", "Student_teacher_ratio", "Test_scores",
                 "HS_dropout", "Labor_force_participation", "Social_capital",
                 "Colleges", "Tuition", "Single_mothers")

# Step 2: Compute correlation matrix
cor_matrix <- cor(cleaned_data, use = "pairwise.complete.obs")

# Step 3: Convert matrix into a dataframe
cor_df <- as.data.frame(as.table(cor_matrix))

# Step 4: Remove self-correlations (diagonal)
cor_df <- cor_df %>%
  filter(Var1 != Var2)

# Step 5: Standardize Var1 & Var2 order to remove duplicates
cor_df <- cor_df %>%
  rowwise() %>%
  mutate(pair = paste(sort(c(Var1, Var2)), collapse = "_")) %>%  # Create a unique pair ID
  distinct(pair, .keep_all = TRUE) %>%  # Remove duplicate pairs
  select(-pair)  # Drop helper column

# Step 6: Find top 5 correlated variables for each policy predictor
top_correlations <- list()

for (var in policy_vars) {
  top_5 <- cor_df %>%
    filter(Var1 == var | Var2 == var) %>%  # Select rows where var appears
    arrange(desc(abs(Freq))) %>%  # Sort by absolute correlation
    head(5)  # Select top 5
  top_correlations[[var]] <- top_5
}

# Step 7: Display results
print(top_correlations)
```

```
## $Local_tax_rate
## # A tibble: 5 x 3
## # Rowwise:
```

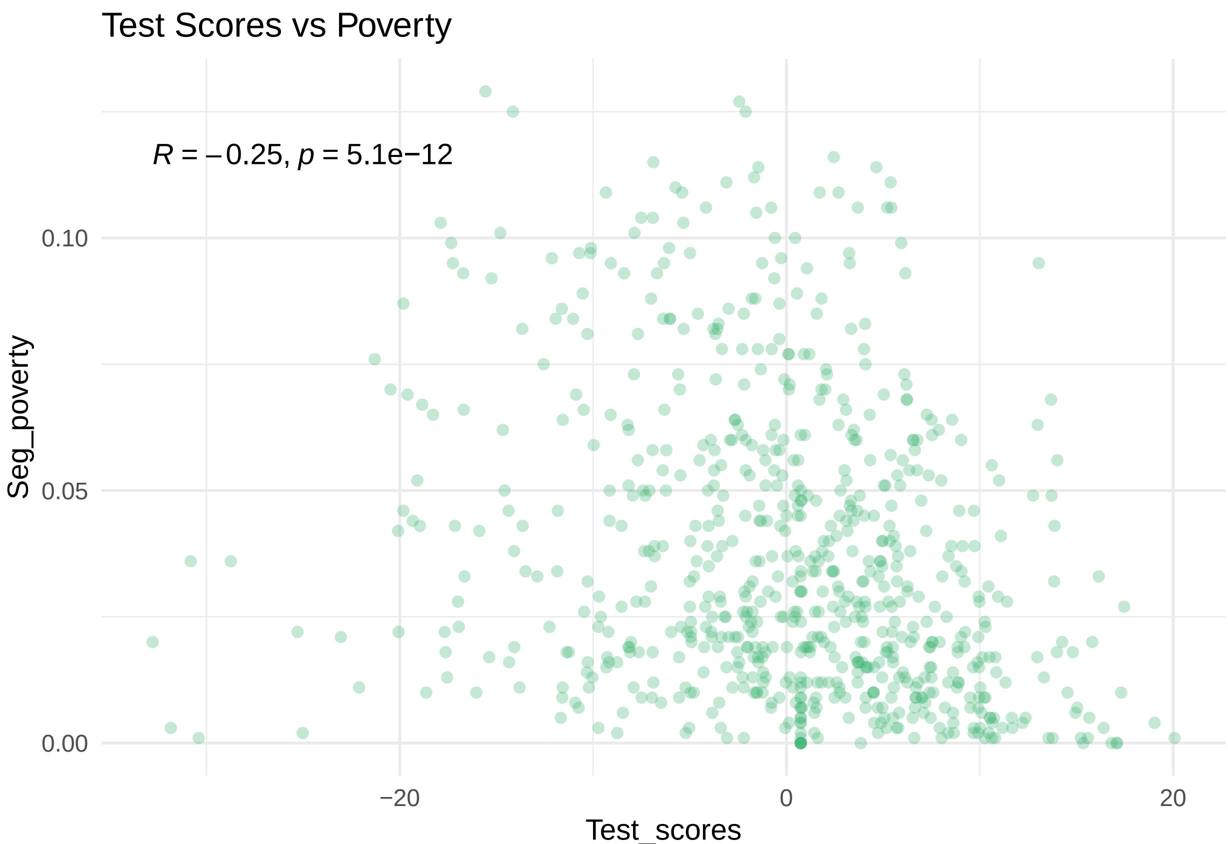
```
##   Var1               Var2            Freq
##   <fct>              <fct>          <dbl>
## 1 School_spending    Local_tax_rate  0.486
## 2 Local_gov_spending Local_tax_rate  0.406
## 3 Manufacturing      Local_tax_rate -0.362
## 4 Local_tax_rate     Commute         0.350
## 5 Teenage_labor      Local_tax_rate  0.349
##
## $Local_gov_spending
## # A tibble: 5 x 3
## # Rowwise:
##   Var1                     Var2                Freq
##   <fct>                    <fct>              <dbl>
## 1 Local_gov_spending        Local_tax_rate     0.406
## 2 School_spending           Local_gov_spending 0.403
## 3 Local_gov_spending        Income             0.285
## 4 Teenage_labor             Local_gov_spending 0.275
## 5 Labor_force_participation Local_gov_spending 0.271
##
## $Progressivity
## # A tibble: 5 x 3
## # Rowwise:
##   Var1                  Var2           Freq
##   <fct>                 <fct>         <dbl>
## 1 EITC                  Progressivity 0.262
## 2 Student_teacher_ratio Progressivity 0.197
## 3 Progressivity         Mobility      0.190
## 4 Progressivity         Population    0.160
## 5 Foreign_born          Progressivity 0.154
##
## $EITC
## # A tibble: 5 x 3
## # Rowwise:
##   Var1            Var2           Freq
##   <fct>           <fct>         <dbl>
## 1 Teenage_labor   EITC           0.350
## 2 School_spending EITC           0.349
## 3 Social_capital  EITC           0.345
## 4 EITC            Gini_99       -0.305
## 5 EITC            Middle_class   0.268
##
## $School_spending
## # A tibble: 5 x 3
## # Rowwise:
##   Var1            Var2                 Freq
##   <fct>           <fct>               <dbl>
## 1 School_spending Local_tax_rate      0.486
## 2 School_spending Local_gov_spending  0.403
## 3 School_spending EITC                0.349
## 4 Teenage_labor   School_spending     0.335
## 5 School_spending Black              -0.311
##
## $Student_teacher_ratio
## # A tibble: 5 x 3
```

```
## # Rowwise:
##   Var1                  Var2                    Freq
##   <fct>                 <fct>                  <dbl>
## 1 Migration_in          Student_teacher_ratio  0.435
## 2 Student_teacher_ratio Seg_income             0.432
## 3 Student_teacher_ratio Commute               -0.431
## 4 Student_teacher_ratio Seg_affluence          0.428
## 5 Student_teacher_ratio Seg_poverty            0.417
##
## $Test_scores
## # A tibble: 5 x 3
## # Rowwise:
##   Var1           Var2           Freq
##   <fct>          <fct>         <dbl>
## 1 Test_scores    Middle_class  0.638
## 2 Single_mothers Test_scores  -0.580
## 3 Social_capital Test_scores   0.523
## 4 Married        Test_scores   0.521
## 5 Test_scores    Gini_99      -0.496
##
## $HS_dropout
## # A tibble: 5 x 3
## # Rowwise:
##   Var1           Var2           Freq
##   <fct>          <fct>         <dbl>
## 1 HS_dropout     Test_scores  -0.487
## 2 Single_mothers HS_dropout    0.482
## 3 HS_dropout     Middle_class -0.474
## 4 Married        HS_dropout   -0.432
## 5 HS_dropout     Gini_99       0.402
##
## $Labor_force_participation
## # A tibble: 5 x 3
## # Rowwise:
##   Var1                      Var2                       Freq
##   <fct>                     <fct>                     <dbl>
## 1 Labor_force_participation Income                     0.544
## 2 Teenage_labor             Labor_force_participation  0.534
## 3 Labor_force_participation Gini_99                   -0.465
## 4 Social_capital            Labor_force_participation  0.403
## 5 Labor_force_participation Middle_class               0.361
##
## $Social_capital
## # A tibble: 5 x 3
## # Rowwise:
##   Var1           Var2            Freq
##   <fct>          <fct>          <dbl>
## 1 Social_capital Teenage_labor  0.708
## 2 Social_capital Gini_99       -0.656
## 3 Social_capital Middle_class   0.652
## 4 Social_capital Gini          -0.569
## 5 Social_capital Commute        0.531
##
## $Colleges
```

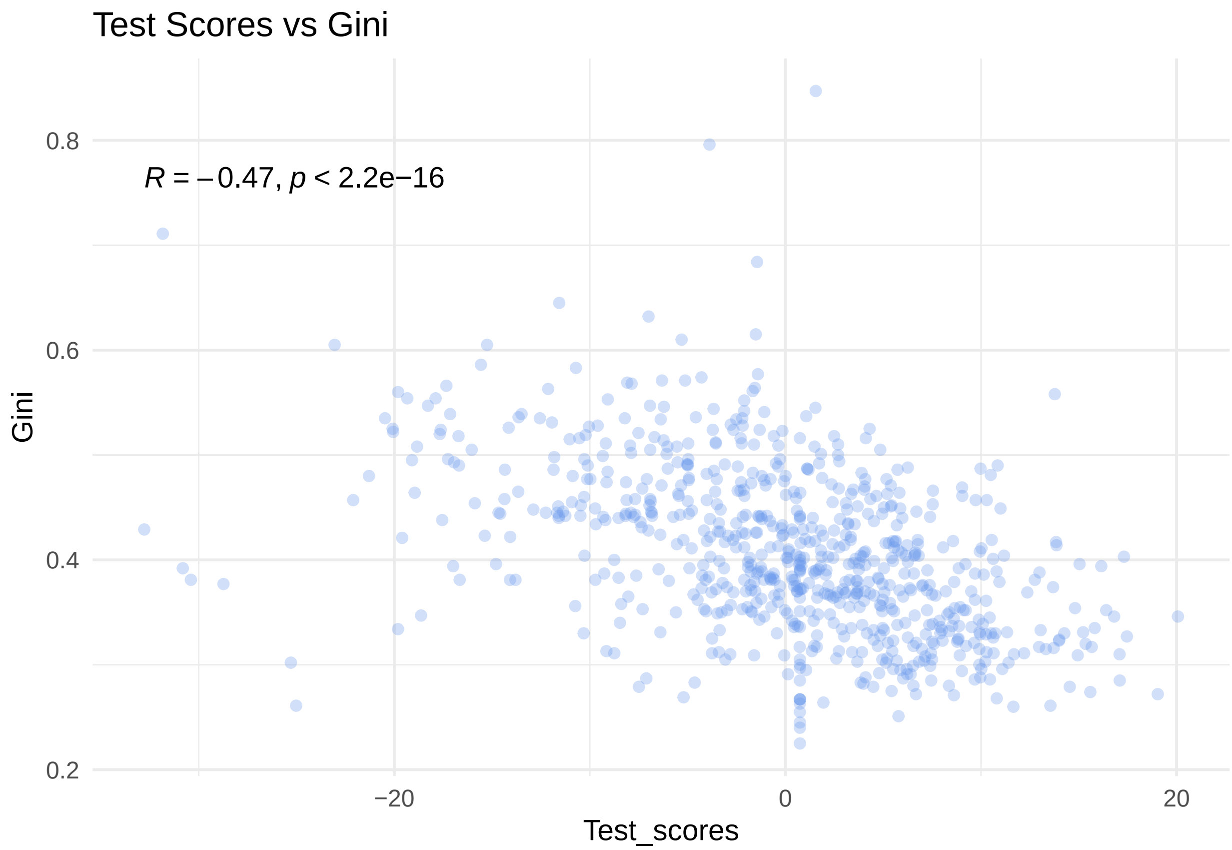
```
## # A tibble: 5 x 3
## # Rowwise:
##   Var1     Var2                   Freq
##   <fct>    <fct>                 <dbl>
## 1 Colleges Commute               0.360
## 2 Colleges Seg_affluence        -0.260
## 3 Colleges Seg_income           -0.257
## 4 Colleges Seg_poverty          -0.251
## 5 Colleges Student_teacher_ratio -0.242
##
## $Tuition
## # A tibble: 5 x 3
## # Rowwise:
##   Var1          Var2         Freq
##   <fct>         <fct>       <dbl>
## 1 Graduation    Tuition     0.325
## 2 Tuition       Income      0.260
## 3 Manufacturing Tuition     0.244
## 4 Tuition       Commute    -0.231
## 5 Tuition       Population  0.203
##
## $Single_mothers
## # A tibble: 5 x 3
## # Rowwise:
##   Var1           Var2             Freq
##   <fct>          <fct>           <dbl>
## 1 Single_mothers Black           0.781
## 2 Married        Single_mothers -0.716
## 3 Single_mothers Middle_class   -0.711
## 4 Single_mothers Mobility       -0.686
## 5 Single_mothers Gini_99         0.683
```

```r
# Define base dataset
data <- cleaned_data  # Use cleaned dataset without missing values

# Function to create individual scatter plots (Fixed for ggplot2 3.0+)
plot_scatter <- function(x_var, y_var, color, title) {
  ggplot(data, aes(.data[[x_var]], .data[[y_var]])) +  # Updated for tidy evaluation
    geom_point(color = color, alpha = .3) +
    stat_cor(label.x = min(data[[x_var]], na.rm = TRUE),
             label.y = max(data[[y_var]], na.rm = TRUE) * 0.9) +
    ggtitle(title) +
    theme_minimal()
}

# Generate and display individual plots
p1 <- plot_scatter("Test_scores", "Seg_poverty", "mediumseagreen", "Test Scores vs Poverty")
p2 <- plot_scatter("Test_scores", "Gini", "cornflowerblue", "Test Scores vs Gini")
p3 <- plot_scatter("Test_scores", "Gini_99", "skyblue", "Test Scores vs Gini (99%)")
p4 <- plot_scatter("Test_scores", "Middle_class", "darkorange", "Test Scores vs Middle Class")
p5 <- plot_scatter("Test_scores", "Single_mothers", "red", "Test Scores vs Single Mothers")
p6 <- plot_scatter("Test_scores", "School_spending", "pink", "Test Scores vs School Spending")

# Print plots one by one
```

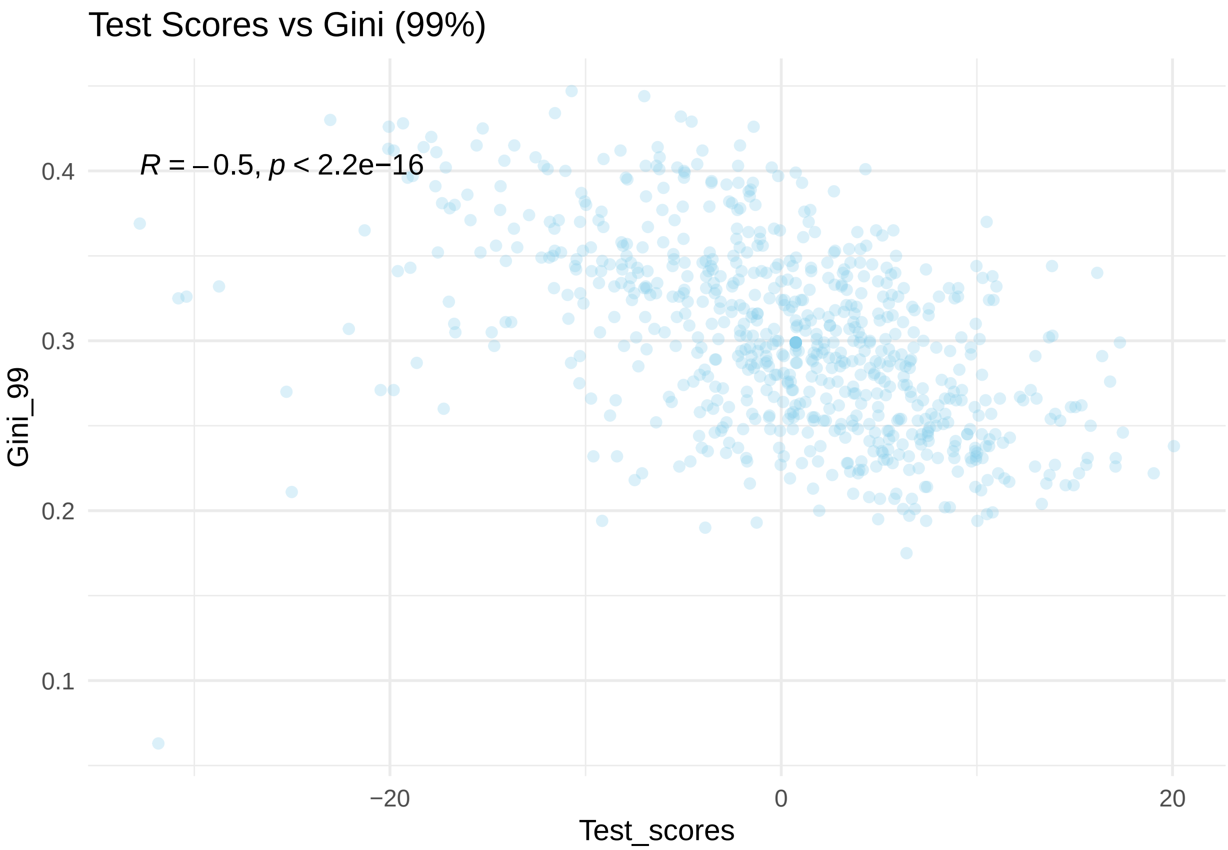
```r
print(p1)
```



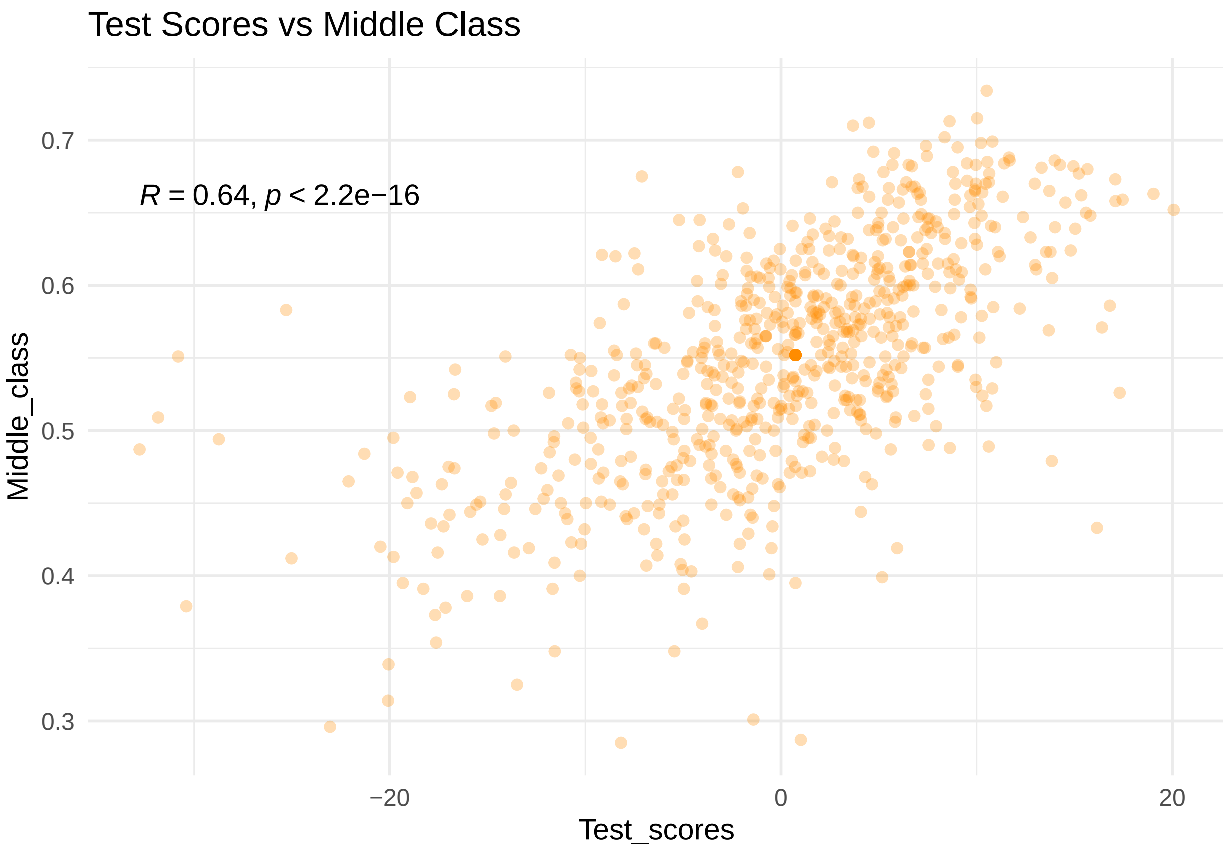

```r
print(p2)
```


Test Scores vs Gini
R = −0.47, p < 2.2e−16
0.8
0.6
0.4
0.2
Gini
−20
0
20
Test_scores

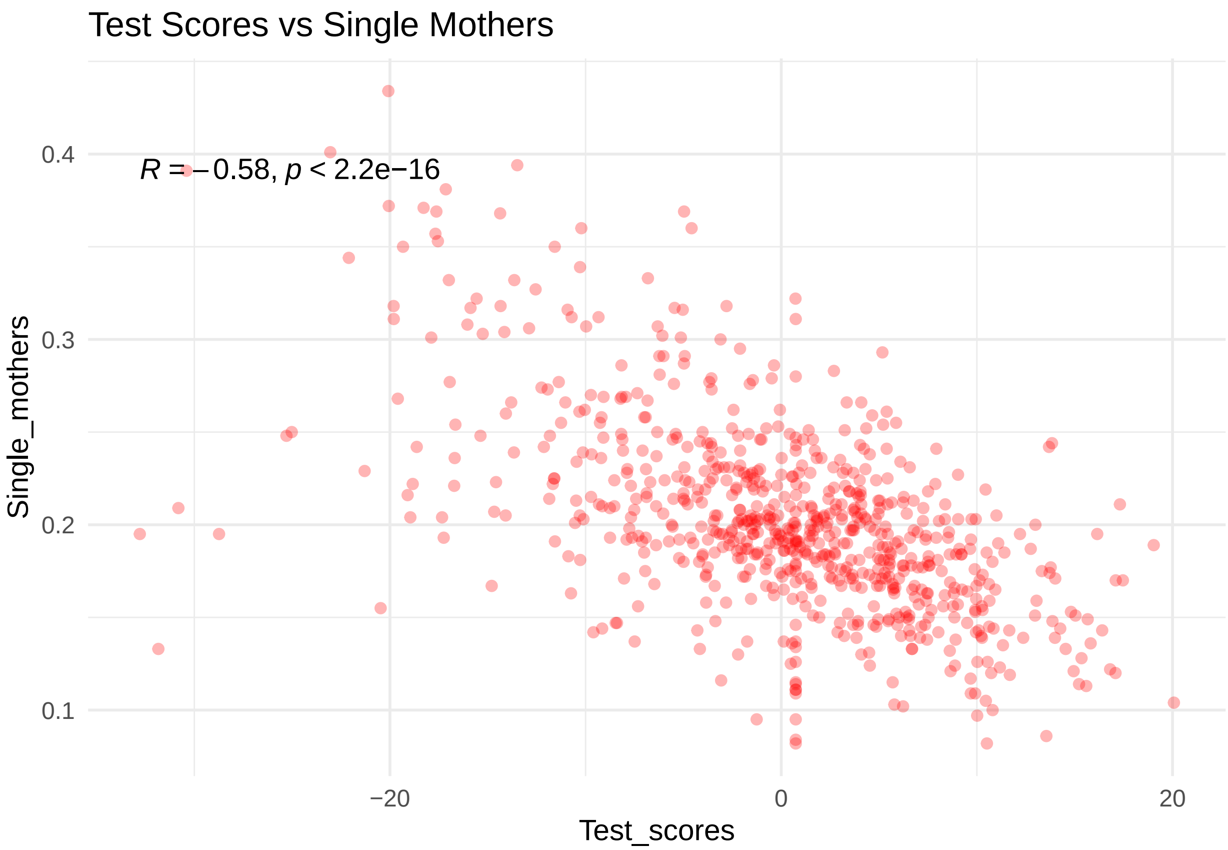
```r
print(p3)
```


Test Scores vs Gini (99%)
R = −0.5, p < 2.2e−16
0.4
0.3
0.2
0.1
Gini_99
−20
0
20
Test_scores

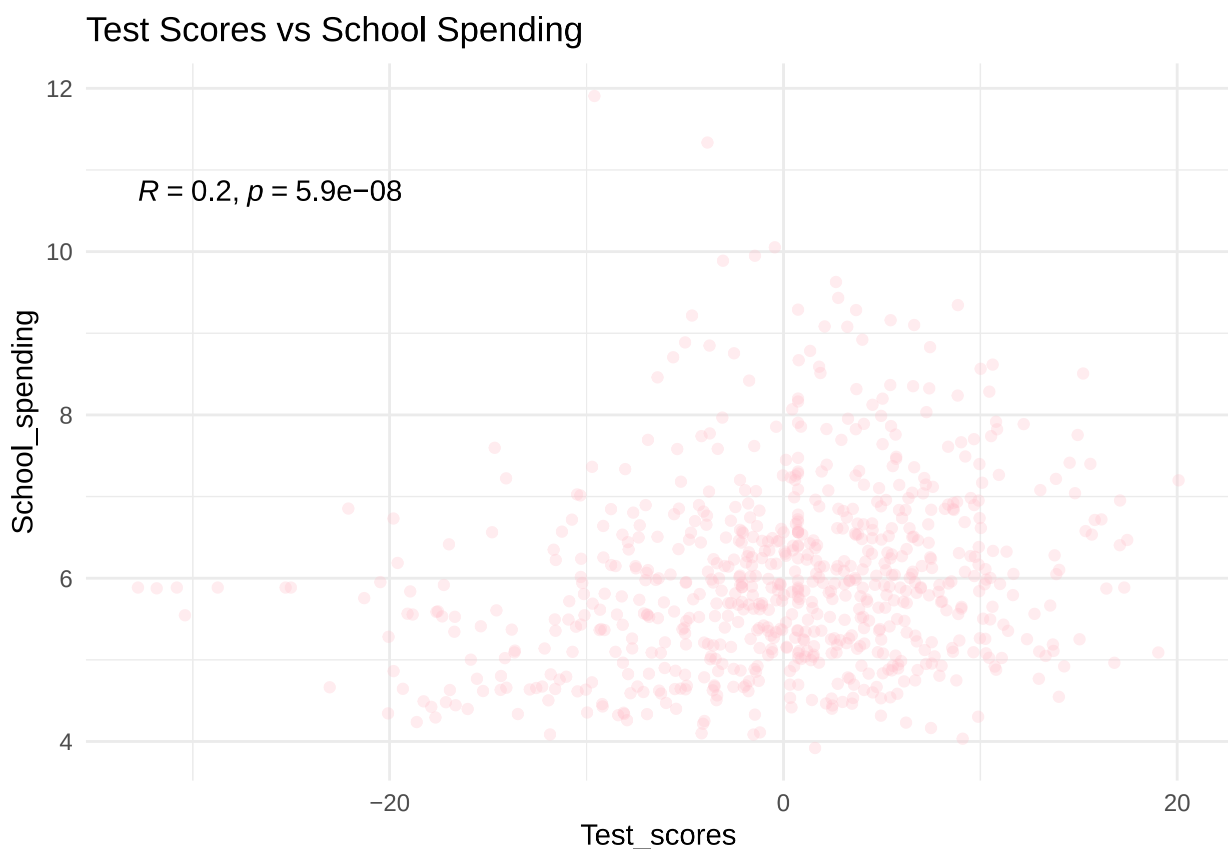
```
print(p4)
```


Test Scores vs Middle Class
R = 0.64, p < 2.2e−16
Middle_class
0.7
0.6
0.5
0.4
0.3
−20
0
20
Test_scores

```r
print(p5)
```

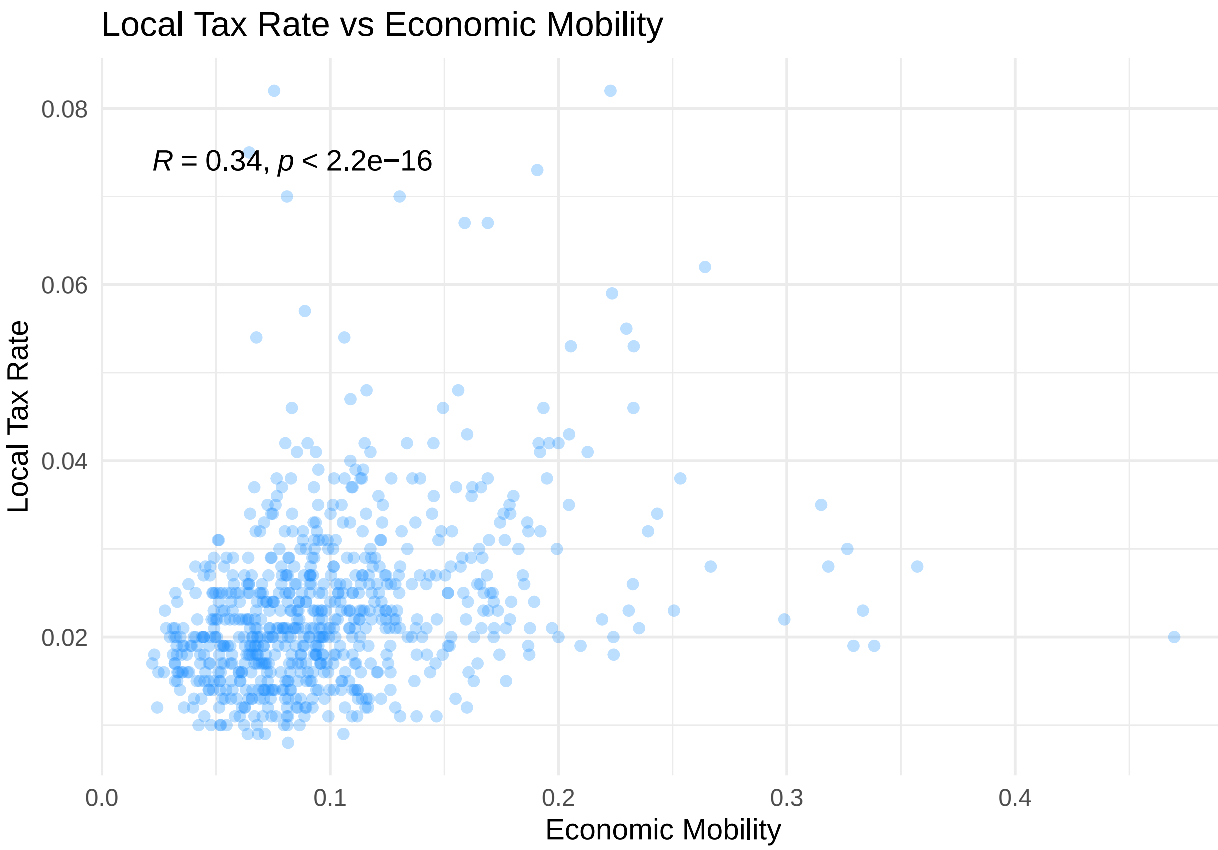

Test Scores vs Single Mothers
R = −0.58, p < 2.2e−16
0.4
0.3
0.2
0.1
Single_mothers
−20
0
20
Test_scores

```
print(p6)
```

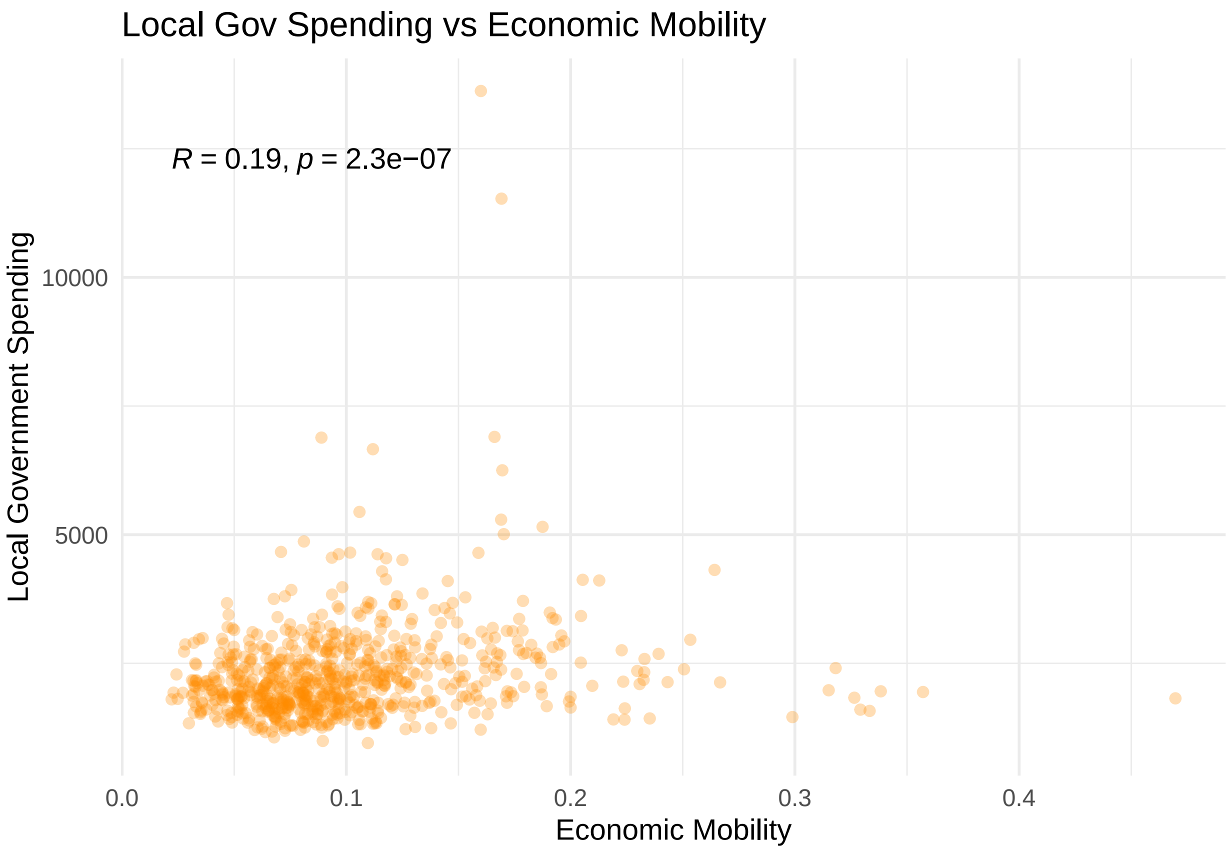



```
data_filtered <- cleaned_data %>%
  filter(
    !is.na(Test_scores) & !is.na(Mobility) &
    is.finite(Test_scores) & is.finite(Mobility)
  )
# Create scatter plot
p <- ggplot(data_filtered, aes(x = Test_scores, y = Mobility)) +
  geom_point(color = "mediumseagreen", alpha = .3) +
  stat_cor(label.x = min(data_filtered$Test_scores, na.rm = TRUE),
           label.y = max(data_filtered$Mobility, na.rm = TRUE) * 0.9) +
  ggtitle("Test Scores vs Economic Mobility") +
  xlab("Test Scores") +
  ylab("Economic Mobility") +
  theme_minimal()

print(p)
```

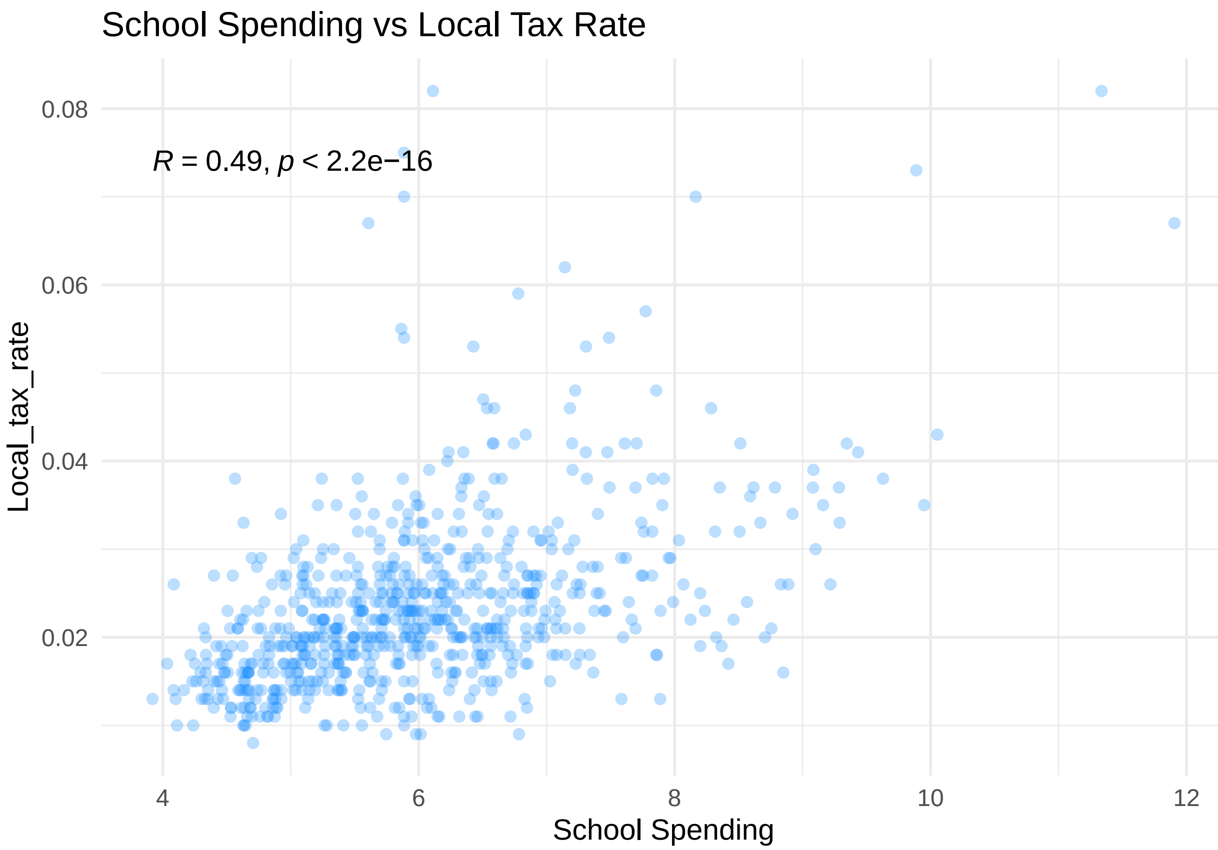

```r
# Define base dataset
data <- cleaned_data  # Use cleaned dataset without missing values

# Plot 1: Mobility vs Local Tax Rate
p1 <- ggplot(data, aes(x = Mobility, y = Local_tax_rate)) +
  geom_point(color = "dodgerblue", alpha = .3) +
  stat_cor(label.x = min(data$Mobility, na.rm = TRUE),
           label.y = max(data$Local_tax_rate, na.rm = TRUE) * 0.9) +
  ggtitle("Local Tax Rate vs Economic Mobility") +
  xlab("Economic Mobility") +
  ylab("Local Tax Rate") +
  theme_minimal()

# Plot 2: Mobility vs Local Government Spending
p2 <- ggplot(data, aes(x = Mobility, y = Local_gov_spending)) +
  geom_point(color = "darkorange", alpha = .3) +
  stat_cor(label.x = min(data$Mobility, na.rm = TRUE),
           label.y = max(data$Local_gov_spending, na.rm = TRUE) * 0.9) +
  ggtitle("Local Gov Spending vs Economic Mobility") +
  xlab("Economic Mobility") +
  ylab("Local Government Spending") +
  theme_minimal()

# Print each plot separately
print(p1)
```


Local Tax Rate vs Economic Mobility
R = 0.34, p < 2.2e−16
0.08
0.06
0.04
0.02
Local Tax Rate
0.0
0.1
0.2
0.3
0.4
Economic Mobility

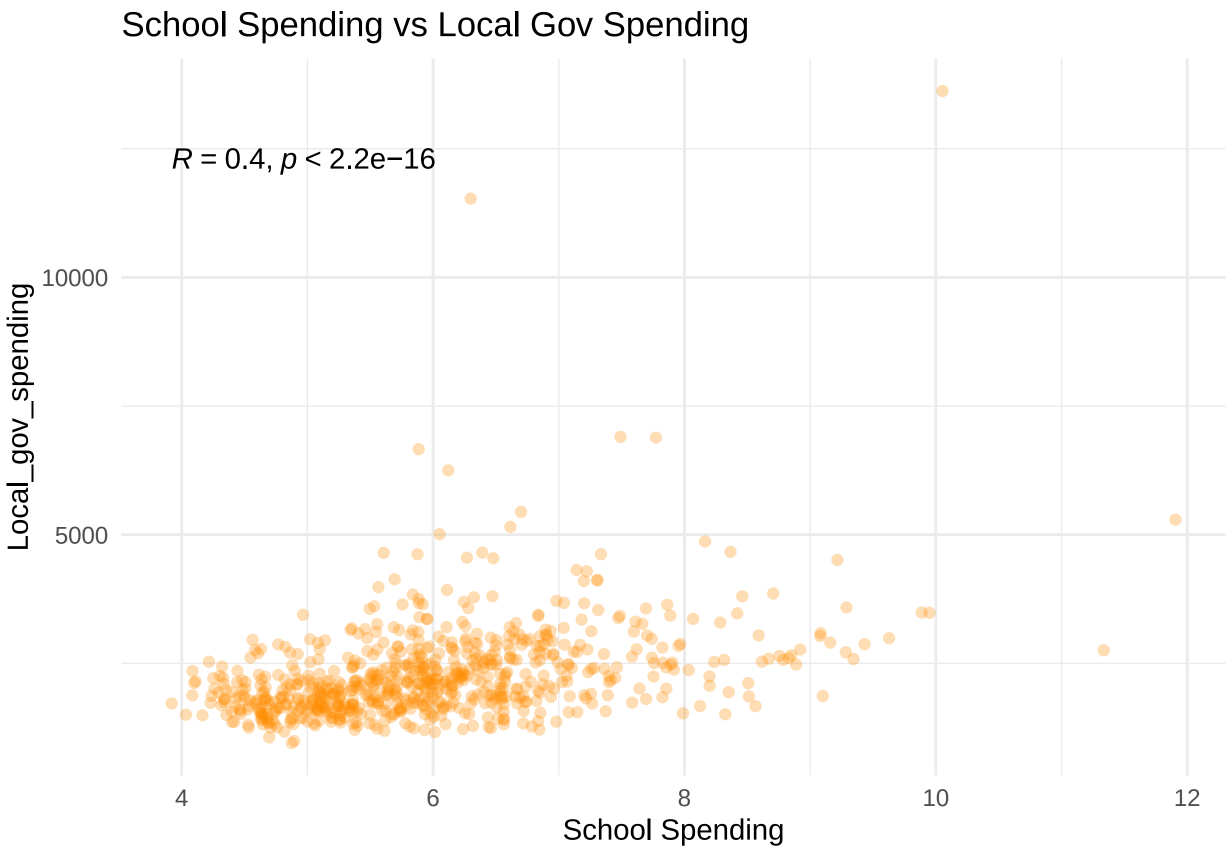
```r
print(p2)
```



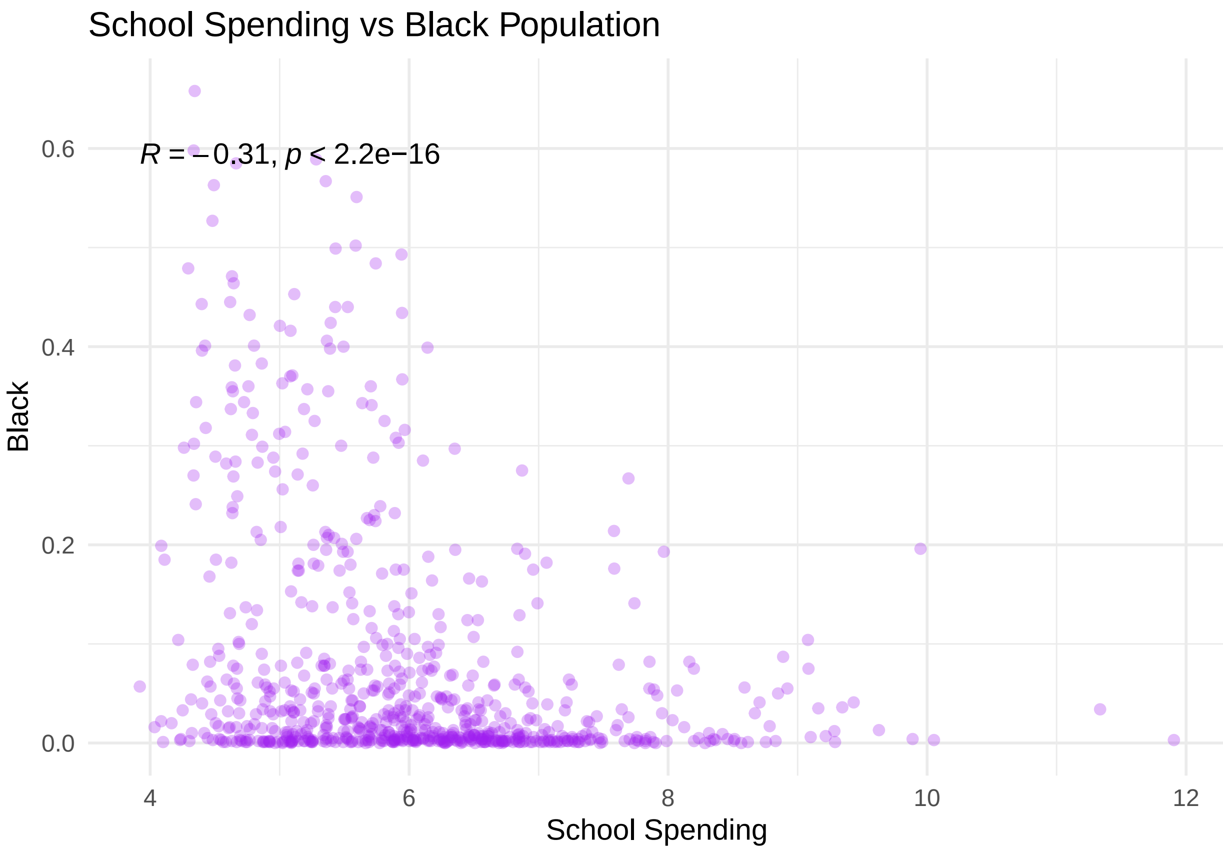

```
# Remove rows with missing or infinite values in relevant columns
data_filtered <- cleaned_data %>%
  filter(
    !is.na(School_spending) & !is.na(Local_tax_rate) & !is.na(Local_gov_spending) & !is.na(Black) &
    is.finite(School_spending) & is.finite(Local_tax_rate) & is.finite(Local_gov_spending) & is.finite(
  )

# Function to create scatter plots
plot_scatter <- function(x_var, color, title) {
  ggplot(data_filtered, aes(x = School_spending, y = .data[[x_var]])) +
    geom_point(color = color, alpha = .3) +
    stat_cor(label.x = min(data_filtered$School_spending, na.rm = TRUE),
             label.y = max(data_filtered[[x_var]], na.rm = TRUE) * 0.9) +
    ggtitle(title) +
    xlab("School Spending") +
    ylab(x_var) +
    theme_minimal()
}

# Generate and display each plot separately
print(plot_scatter("Local_tax_rate", "dodgerblue", "School Spending vs Local Tax Rate"))
```




```
print(plot_scatter("Local_gov_spending", "darkorange", "School Spending vs Local Gov Spending"))
```




```r
print(plot_scatter("Black", "purple", "School Spending vs Black Population"))
```


School Spending vs Black Population
R = −0.31, p < 2.2e−16
0.6
0.4
0.2
0.0
Black
4
6
8
10
12
School Spending