Analyzing Socioeconomic and Housing Factors to Predict Arrest Trends in New York State

Joyce Yan (qy249), Lydia Lin (dl2253), Victoria Xiao (sx287)

2024-11-26

Set up

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(readr)
library(ggplot2)
library(reshape2)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v stringr   1.5.1
v lubridate 1.9.3     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.1
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```
library(tidymodels)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.2.0 --
v broom        1.0.7     v rsample      1.2.1
v dials        1.3.0     v tune         1.2.1
v infer        1.0.7     v workflows    1.1.4
v modeldata    1.4.0     v workflowsets 1.1.0
v parsnip      1.2.1     v yardstick    1.3.1
v recipes      1.1.0
-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Search for functions across packages at https://www.tidymodels.org/find/
```

Objective

Our project aims to predict **the total number of adult arrests in New York State counties based on key social, economic, and housing factors from 2015 to 2022**. By analyzing historical data and identifying patterns, we want to understand how these factors influence arrest counts and use this knowledge to create a reliable predictive model.

This model can help policymakers and law enforcement anticipate trends in crime and allocate resources more effectively. Our goal is to address two key problems:

1. Identifying which socioeconomic and housing factors are most strongly associated with arrest counts.
2. Developing a predictive tool to estimate future arrest counts, which can guide decisions on crime prevention and community support.

By achieving these objectives, we hope to contribute to data-driven strategies for building safer, more equitable communities.

Data description

Our analysis-ready dataset combines arrest data with socioeconomic and housing characteristics for counties in New York State from 2015 to 2022. The dataset is organized at the county-year level and includes both arrest counts by offense type and demographic, economic, and housing indicators as rates. This structure makes it suitable for exploring relationships and building predictive models.

Variables in the dataset

General Information

- **County**: The name of the county in New York State.
- **Year**: The year of the data, ranging from 2015 to 2022.

Arrest Data

- **Total**: Total adult arrest counts in the county for the respective year.
- **Felony Total**: Total felony arrests.
  - **Drug Felony**: Arrests for drug-related felonies.
  - **Violent Felony**: Arrests for violent felonies.
  - **DWI Felony**: Arrests for driving while intoxicated (DWI) felonies.
  - **Other Felony**: Arrests for other felony offenses.
- **Misdemeanor Total**: Total misdemeanor arrests.
  - **Drug Misdemeanor**: Arrests for drug-related misdemeanors.
  - **DWI Misdemeanor**: Arrests for DWI misdemeanors.
  - **Property Misdemeanor**: Arrests for property-related misdemeanors.
  - **Other Misdemeanor**: Arrests for other misdemeanor offenses.

Social Characteristics

- **Less_than_9th_grade**: Percentage of the population aged 25 and over with less than a 9th-grade education.
- **With_disability**: Percentage of the civilian noninstitutionalized population with a disability.
- **Civilian_veterans**: Percentage of the civilian population aged 18 years and over who are veterans.
- **Limited_English**: Percentage of the population aged 5 and over who speak a language other than English at home and speak English less than "very well."

Housing Data

- **Total_housing_units**: Total housing units in terms of occupancy as a percentage of the population.
- **With_mortgage**: Percentage of owner-occupied housing units that have a mortgage.
- **Without_mortgage**: Percentage of owner-occupied housing units without a mortgage.

Economic Characteristics

- **Unemployment_rate**: The percentage of the civilian labor force that is unemployed.
- **Median_household_income**: Median income of all households in the county, adjusted for 2022 inflation.
- **Below_poverty_level**: Percentage of individuals aged 18 and over whose income is below the poverty threshold.
- **No_health_insurance**: Percentage of the civilian noninstitutionalized population without health insurance coverage.

Key Characteristics

- **Arrest Data**: Includes comprehensive breakdowns by felony and misdemeanor types, enabling analysis of specific crime trends.
- **Socioeconomic Indicators**: Rates of unemployment, poverty, disability, and limited English proficiency provide context on social challenges within counties.
- **Housing Data**: Percentages of housing units with and without mortgages offer insight into housing stability in each county.

The dataset ensures comparability across counties by using rates instead of raw counts for demographic, economic, and housing variables. This approach accounts for differences in county population sizes, making the data suitable for cross-county analysis.

EDA

Based on our team's previous exploration, we observed a significant variation in the population sizes of counties in New York State. Some counties have substantially larger populations compared to others. To minimize bias, it would be more appropriate to use percentages for our feature columns instead of raw counts. Therefore, we decided to revisit the data cleaning process and transform the count-based features into percentage-based features. We will then perform exploratory data analysis using these newly transformed feature columns.

You can find our detailed data cleaning process in the **explore.qmd** file. I will export a cleaned **merged_data.csv** file with the percentage features added to facilitate further analysis.

If we were to display all 62 counties in New York State in our EDA, the plots would become overly cluttered and difficult to interpret. Therefore, we decided to focus on the top 10 counties with the highest arrests to initially explore relationships between variables. Later, we will apply feature selection methods to gain a deeper understanding of the key features.

```
df = read_csv("data/merged_data.csv")
```

```
Rows: 552 Columns: 24
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): County
dbl (23): Year, Total, Felony Total, Drug Felony, Violent Felony, DWI Felony...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

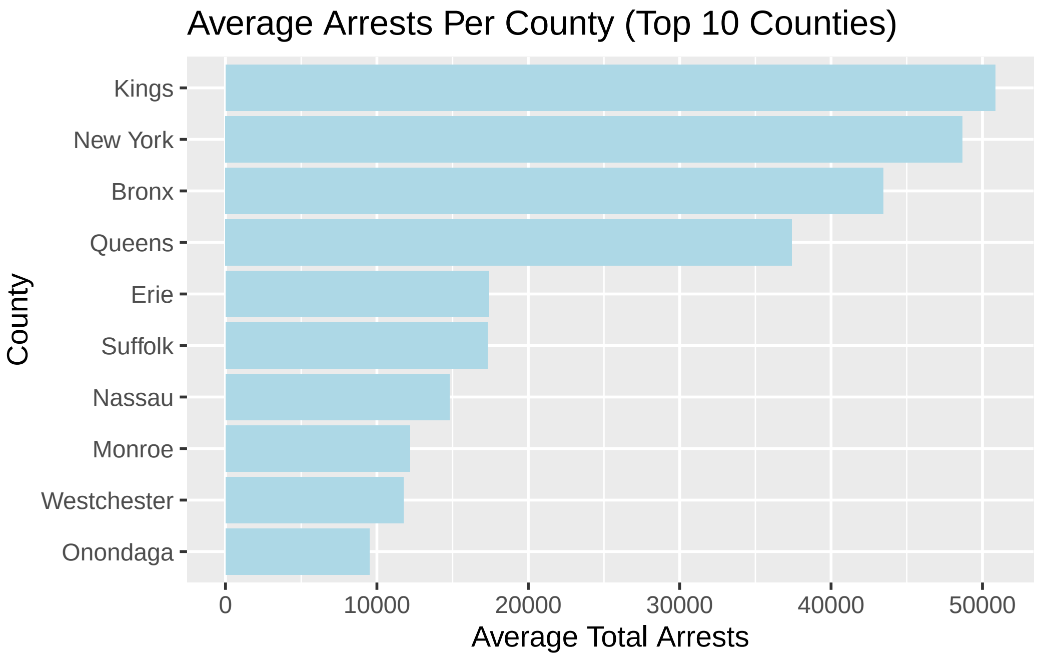
```
head(df)
```

```
# A tibble: 6 x 24
  County   Year Total `Felony Total` `Drug Felony` `Violent Felony` `DWI Felony`
  <chr>   <dbl> <dbl>          <dbl>         <dbl>            <dbl>        <dbl>
1 Albany   2015  7412           2590           509              533          161
2 Allega~  2015   792            209            33               39           38
3 Bronx    2015 59576          16121          4333             5623          156
4 Broome   2015  4968           1532           216              389           91
5 Cattar~  2015  1516            439           103               86           47
6 Cayuga   2015  1327            431            68               57           27
# i 17 more variables: `Other Felony` <dbl>, `Misdemeanor Total` <dbl>,
#   `Drug Misdemeanor` <dbl>, `DWI Misdemeanor` <dbl>,
#   `Property Misdemeanor` <dbl>, `Other Misdemeanor` <dbl>,
#   Less_than_9th_grade <dbl>, With_disability <dbl>, Civilian_veterans <dbl>,
#   Limited_English <dbl>, Total_housing_units <dbl>, With_mortgage <dbl>,
#   Without_mortgage <dbl>, Unemployment_rate <dbl>,
#   Median_household_income <dbl>, Below_poverty_level <dbl>, ...
```

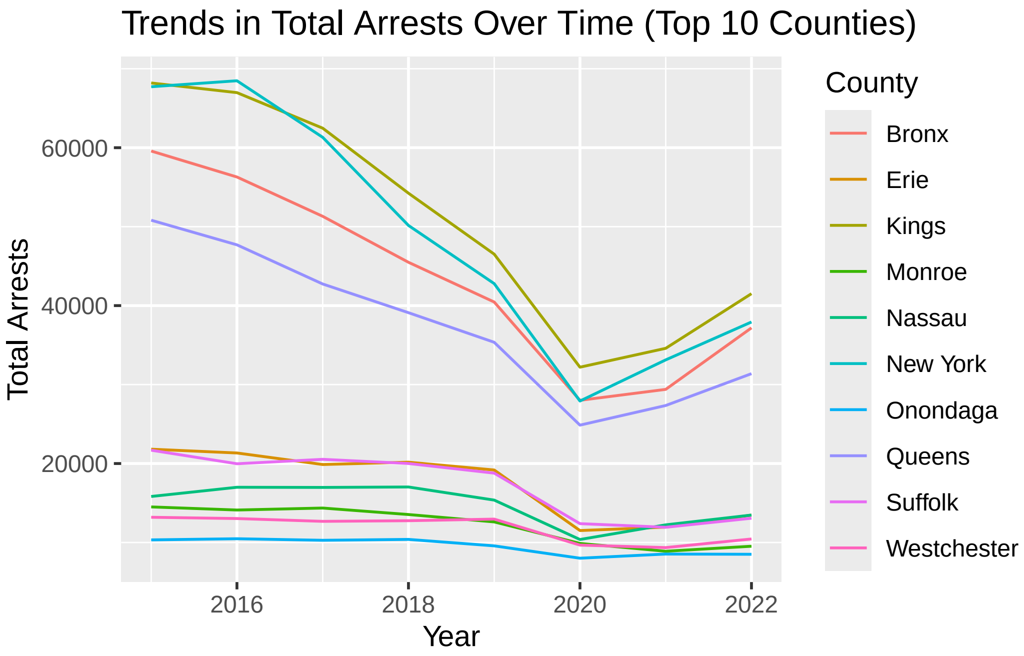
EDA on arrests

```
# Determine top 10 counties with the most arrests
top_counties <- df |>
```

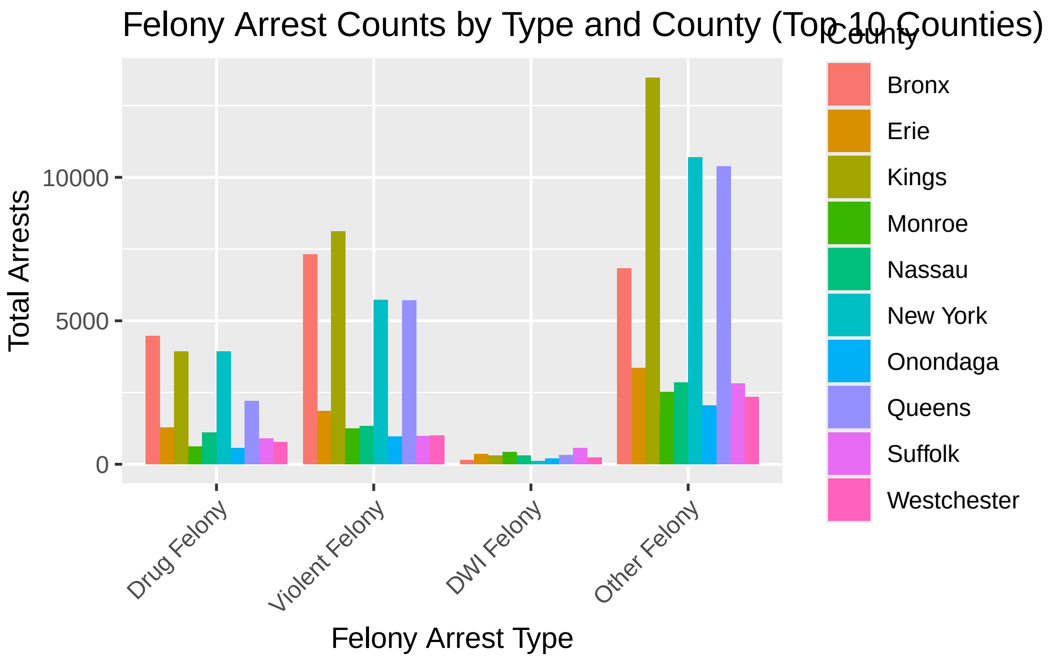
```r
  group_by(County) |>
  summarize(Total_Arrests = sum(Total, na.rm = TRUE)) |>
  arrange(desc(Total_Arrests)) |>
  slice(1:10) |>
  pull(County)

# Filter the dataset to include only the top 10 counties
df_top10 <- df |>
  filter(County %in% top_counties)

# Horizontal bar plot showing the average number of total arrests for each of the top 10 cou
county_summary_top10 <- df_top10 |>
  group_by(County) |>
  summarize(Average_Total_Arrests = mean(Total, na.rm = TRUE))

ggplot(county_summary_top10, aes(x = reorder(County, Average_Total_Arrests),
                                 y = Average_Total_Arrests)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  coord_flip() +
  labs(
    title = "Average Arrests Per County (Top 10 Counties)",
    x = "County",
    y = "Average Total Arrests"
  )
```



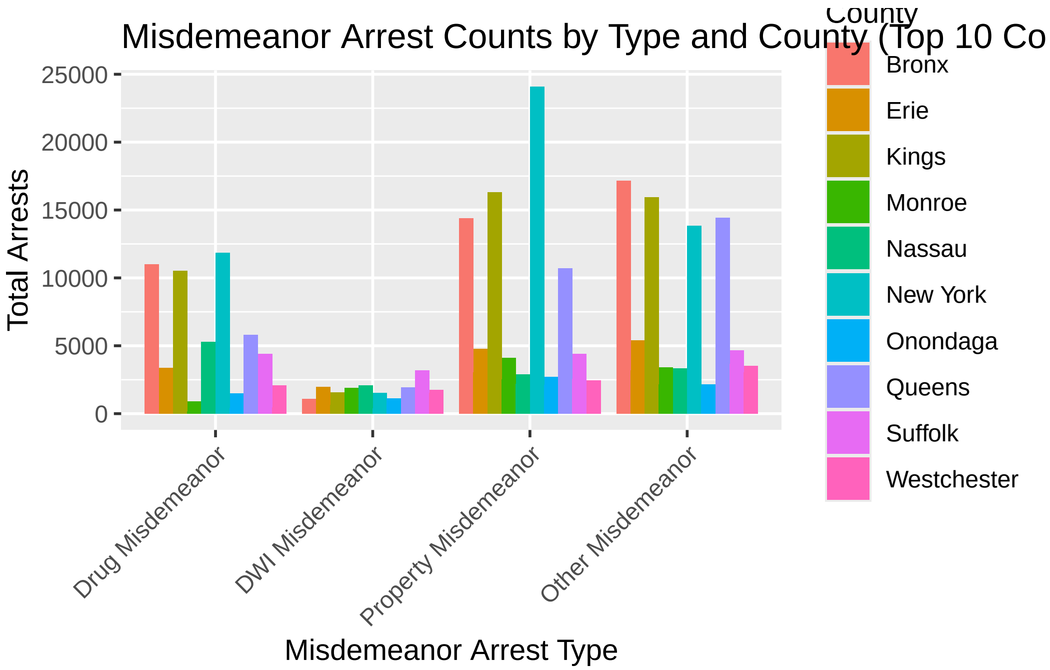
```
# Line plot showing the total number of arrests over time for the top 10 counties
ggplot(df_top10, aes(x = Year, y = Total, color = County)) +
  geom_line() +
  labs(
    title = "Trends in Total Arrests Over Time (Top 10 Counties)",
    x = "Year",
    y = "Total Arrests"
  )
```



```
# bar plot showing the number of arrests by type for each of the top 10 counties
arrests_stacked_top10 <- melt(
  df_top10,
  id.vars = c("County", "Year"),
  measure.vars = c("Drug Felony", "Violent Felony", "DWI Felony", "Other Felony",
                   "Drug Misdemeanor", "DWI Misdemeanor", "Property Misdemeanor",
                   "Other Misdemeanor"),
  variable.name = "Arrest_Type",
  value.name = "Count"
)

arrests_felony <- arrests_stacked_top10 |>
  filter(grepl("Felony", Arrest_Type))

arrests_misdemeanor <- arrests_stacked_top10 |>
  filter(grepl("Misdemeanor", Arrest_Type))

# Felony
ggplot(arrests_felony, aes(x = Arrest_Type, y = Count, fill = County)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Felony Arrest Counts by Type and County (Top 10 Counties)",
    x = "Felony Arrest Type",
    y = "Total Arrests"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



```
# Misdemeanor
ggplot(arrests_misdemeanor, aes(x = Arrest_Type, y = Count, fill = County)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Misdemeanor Arrest Counts by Type and County (Top 10 Counties)",
    x = "Misdemeanor Arrest Type",
    y = "Total Arrests"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

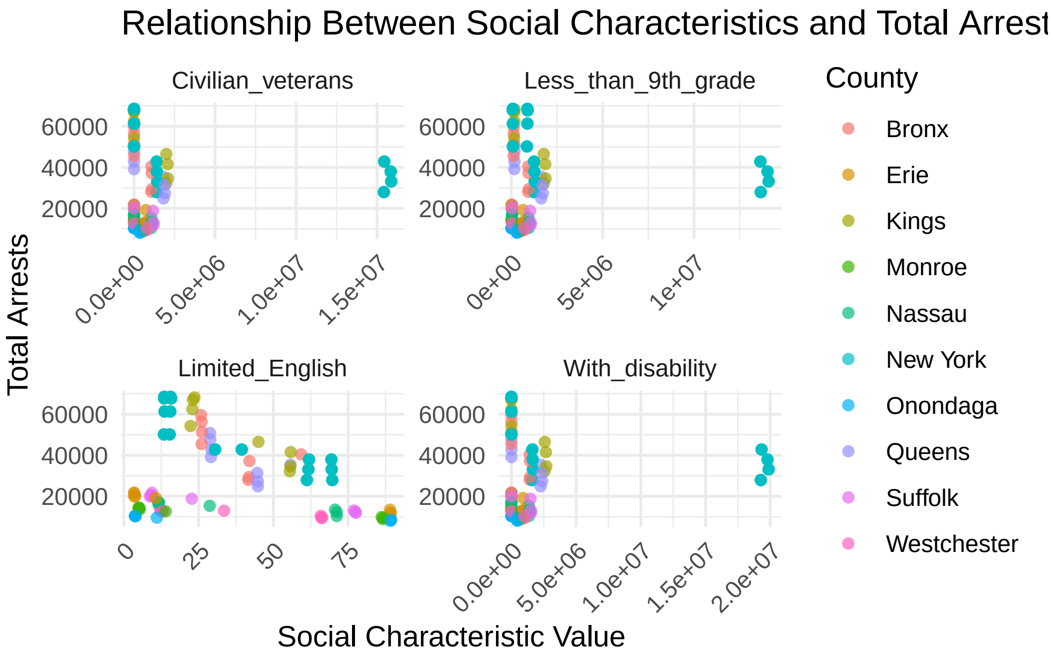


Based on our analysis of arrest data in New York State from 2015 to 2022, several key insights emerge. First, urban counties such as Kings, New York, and Bronx consistently lead in total arrests, with significantly higher average arrests compared to suburban and rural counties. This disparity highlights the influence of population density and socioeconomic challenges on arrest rates. Among felony arrests, violent crimes contribute the largest proportion after "Other Felonies," while property-related offenses account for the majority of misdemeanor arrests after "Other Misdemeanors." These patterns suggest strong correlations with socioeconomic factors such as income, unemployment, and housing instability. Additionally, all counties exhibit a similar trend in total arrests over time, indicating that year is a significant factor influencing arrest rates.

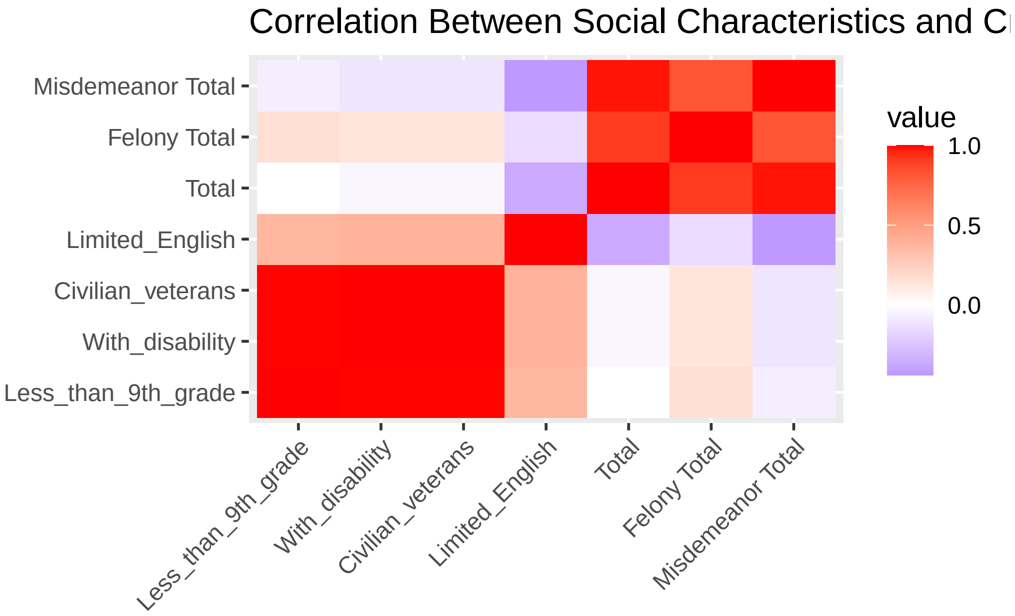
EDA on social characteristics

```
# Reshape social and crime data for scatter plots
social_crime_data <- df_top10 |>
  select(County, Year, Total, Less_than_9th_grade, With_disability,
         Civilian_veterans, Limited_English) |>
  pivot_longer(cols = c(Less_than_9th_grade, With_disability,
                        Civilian_veterans, Limited_English),
               names_to = "Social_Variable", values_to = "Social_Value")

# Scatter plot of social variables vs. total crimes
```

```
ggplot(social_crime_data, aes(x = Social_Value, y = Total, color = County)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ Social_Variable, scales = "free") +
  labs(
    title = "Relationship Between Social Characteristics and Total Arrests",
    x = "Social Characteristic Value",
    y = "Total Arrests"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



```
# Correlation heatmap for social and crime variables
correlation_matrix_social <- df_top10 |>
  select(Less_than_9th_grade, With_disability, Civilian_veterans,
         Limited_English,
         Total, `Felony Total`, `Misdemeanor Total`) |>
  cor()

ggplot(melt(correlation_matrix_social), aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0)+
  labs(
    title = "Correlation Between Social Characteristics and Crime Data",
    x = "",
    y = ""
  ) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
    axis.text.y = element_text()
  )
```

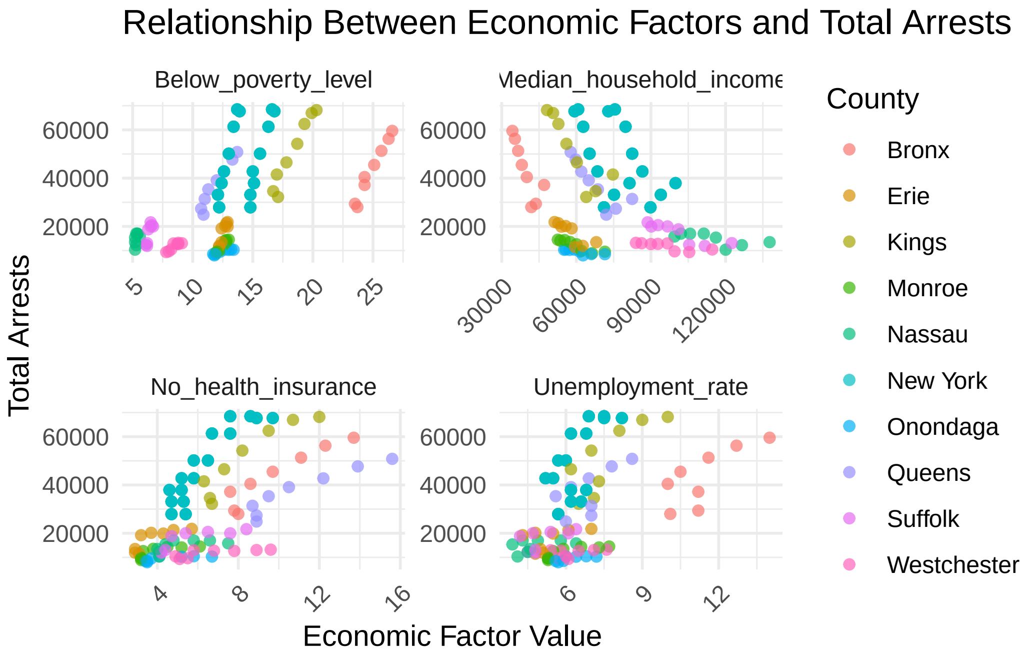


The scatter plots reveal a potential positive correlation between social variables and total arrests, particularly in counties with higher arrest rates. For instance, counties such as New York and Bronx exhibit both elevated arrest totals and significant socioeconomic challenges, including lower education levels and higher rates of disabilities. The correlation heatmap supports these findings, showing moderate to strong relationships between social characteristics and crime variables, such as misdemeanor and felony arrests. These insights suggest that socioeconomic factors—including education, disability status, and language barriers—play a critical role in shaping arrest rates.

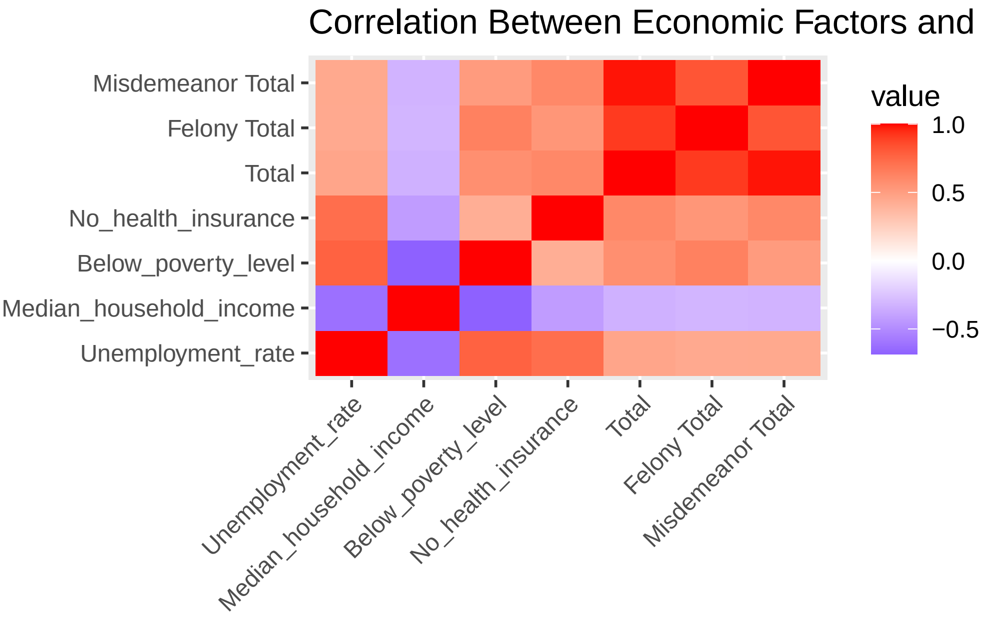
EDA on economic characteristics

```
# Reshape economic and crime data for scatter plots
economic_crime_data <- df_top10 |>
```

```
  select(County, Year, Total, Unemployment_rate, Median_household_income,
         Below_poverty_level, No_health_insurance) |>
  pivot_longer(cols = c(Unemployment_rate, Median_household_income,
                        Below_poverty_level, No_health_insurance),
               names_to = "Economic_Variable", values_to = "Economic_Value")

# Scatter plot of economic variables vs. total crimes
ggplot(economic_crime_data, aes(x = Economic_Value, y = Total, color = County)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ Economic_Variable, scales = "free") +
  labs(
    title = "Relationship Between Economic Factors and Total Arrests",
    x = "Economic Factor Value",
    y = "Total Arrests"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



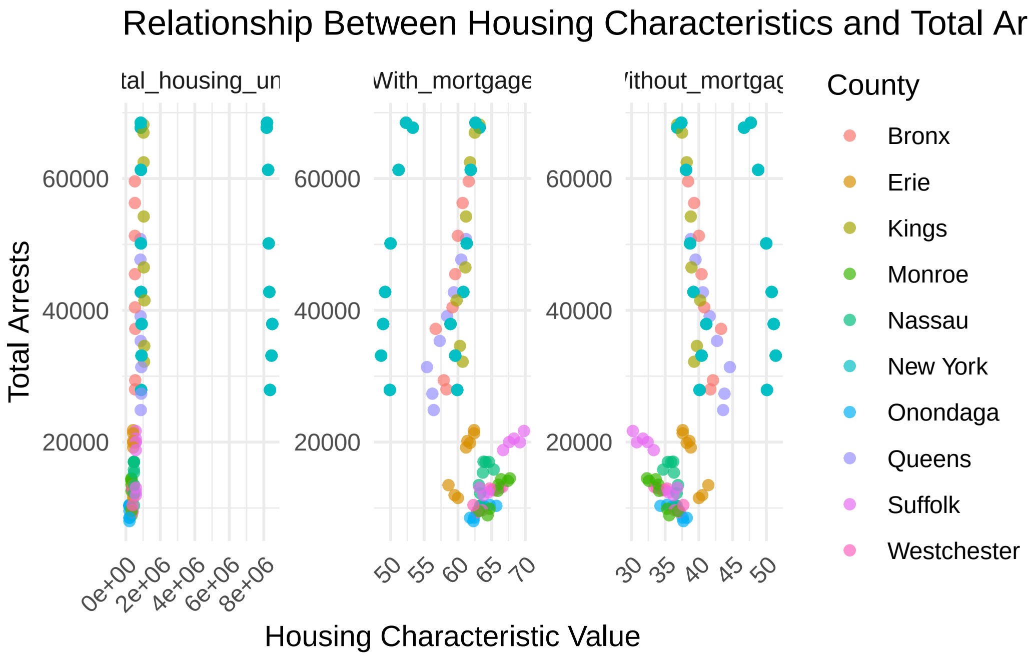
```
# Correlation heatmap for economic and crime variables
correlation_matrix_economic <- df_top10 |>
  select(Unemployment_rate, Median_household_income, Below_poverty_level,
         No_health_insurance,
         Total, `Felony Total`, `Misdemeanor Total`) |>
  cor()

ggplot(melt(correlation_matrix_economic),
       aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +
  labs(
    title = "Correlation Between Economic Factors and Crime Data",
    x = "",
    y = ""
  ) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
    axis.text.y = element_text()
  )
```



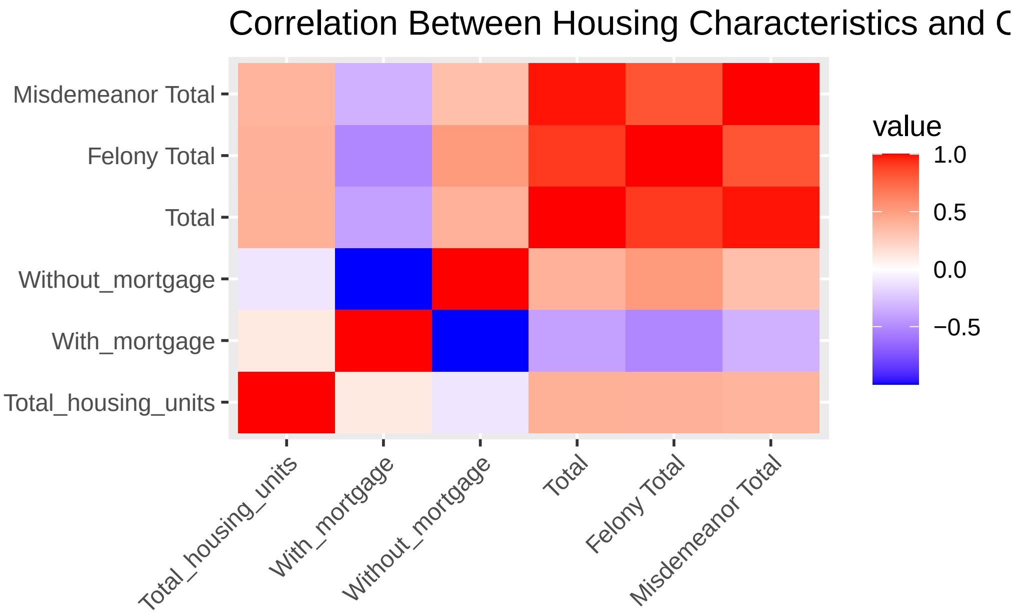
The scatter plots highlight notable trends: higher unemployment rates and greater percentages of the population below the poverty level are associated with increased total arrests, particularly in counties like Bronx and New York. Conversely, higher median household incomes show a negative correlation with total arrests, indicating that wealthier areas experience fewer arrests. Additionally, the percentage of individuals without health insurance exhibits a positive correlation with arrest totals, further emphasizing the role of economic vulnerability in influencing crime rates. The correlation heatmap reinforces these findings, with strong positive correlations observed between unemployment, poverty, and crime variables such as misdemeanor and felony arrests. These results suggest that economic instability, including unemployment, poverty, and lack of access to healthcare, are significant drivers of crime.

EDA on housing characteristics

```r
# Reshape housing and crime data for scatter plots
housing_crime_data <- df_top10 |>
  select(County, Year, Total, Total_housing_units, With_mortgage,
         Without_mortgage) |>
  pivot_longer(cols = c(Total_housing_units, With_mortgage, Without_mortgage),
               names_to = "Housing_Variable", values_to = "Housing_Value")

# Scatter plot of housing variables vs. total crimes
ggplot(housing_crime_data, aes(x = Housing_Value, y = Total, color = County)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ Housing_Variable, scales = "free") +
  labs(
    title = "Relationship Between Housing Characteristics and Total Arrests",
    x = "Housing Characteristic Value",
    y = "Total Arrests"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



```r
# Correlation heatmap for housing and crime variables
correlation_matrix_housing <- df_top10 |>
  select(Total_housing_units, With_mortgage, Without_mortgage,
         Total, `Felony Total`, `Misdemeanor Total`) |>
  cor()

ggplot(melt(correlation_matrix_housing),
       aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0)+
  labs(
    title = "Correlation Between Housing Characteristics and Crime Data",
    x = "",
    y = ""
  ) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
    axis.text.y = element_text()
  )
```

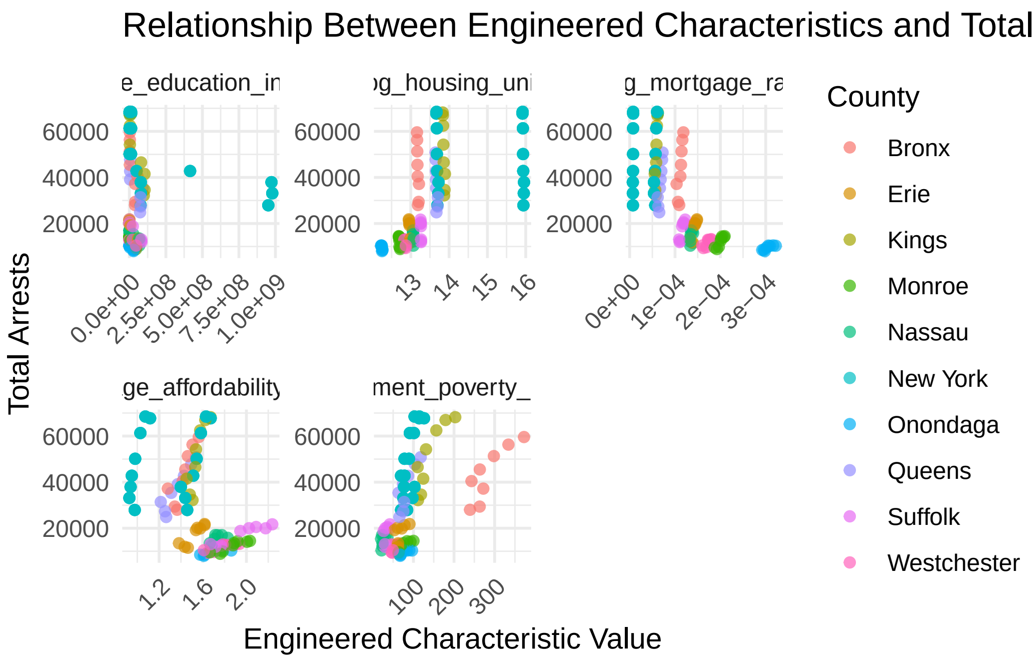


The scatter plots show no strong trends between housing variables and total arrests across counties. Similarly, the correlation heatmap indicates that while total housing units and homes with mortgages have slight positive correlations with crime metrics, these relationships are far weaker than those observed with economic or social factors. The "total housing units" variable forms a straight line, making it difficult to determine any meaningful trend. Both "with mortgage" and "without mortgage" variables show slight trends, but it is challenging to determine whether they are positive or negative, as the scatter points are widely dispersed. Overall, housing characteristics appear to have a limited influence on crime patterns, suggesting that other factors play a more significant role in shaping arrest rates.

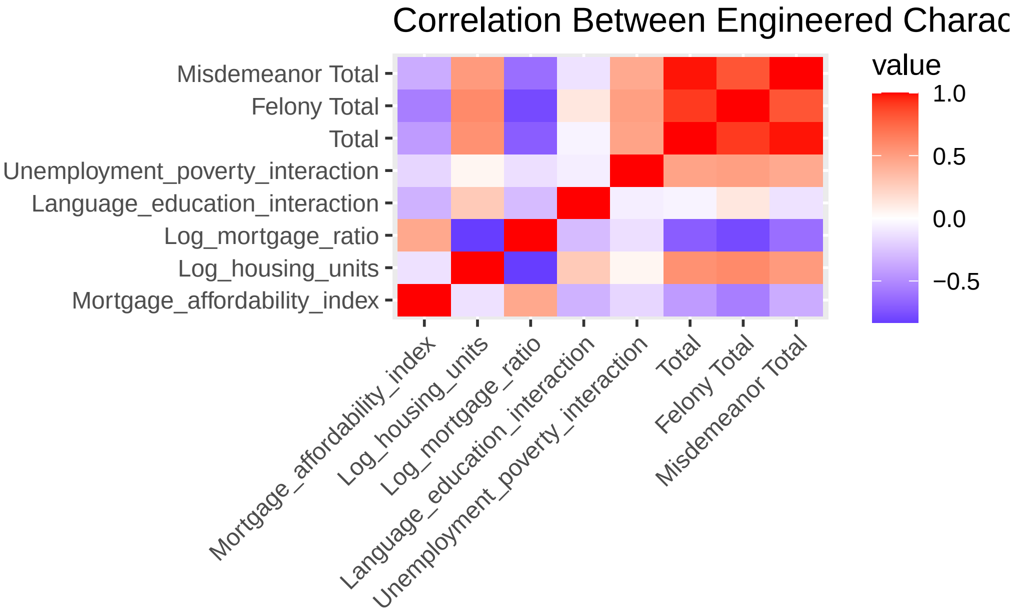
Decisions based on EDA

According to our EDA process, we determined that four economic variables—"Below Poverty Level," "Median Household Income," "Unemployment Rate," and "No Health Insurance"—show a noticeable correlation with arrests, suggesting that these are critical predictors. In contrast, the social and housing factors exhibit weaker relationships with total arrests, with no obvious recognizable trends and low correlation values in the heatmaps. Instead of outright exclusion, we decided to perform feature engineering to transform these variables into potentially more meaningful representations.

```
# Create the engineered features
df <- df |>
  mutate(
    # Mortgage Affordability Index
    Mortgage_affordability_index = With_mortgage / (Without_mortgage + 1),

    # Log Housing Units
    Log_housing_units = log(Total_housing_units + 1),

    # Log Mortgage Ratio
    mortgage_ratio = With_mortgage / Total_housing_units,
    Log_mortgage_ratio = log(mortgage_ratio + 1),

    # Language × Education Interaction
    Language_education_interaction = Limited_English * Less_than_9th_grade,

    # Unemployment × Poverty Interaction
    Unemployment_poverty_interaction = Unemployment_rate * Below_poverty_level
  )

# Filter the dataset to include only the top 10 counties
df_top10 <- df |>
  filter(County %in% top_counties)

# Reshape engineered data for scatter plots
engineered_data <- df_top10 |>
  select(County, Year, Total,
         Mortgage_affordability_index, Log_housing_units,
         Log_mortgage_ratio, Language_education_interaction,
         Unemployment_poverty_interaction) |>
  pivot_longer(cols = c(Mortgage_affordability_index, Log_housing_units,
                        Log_mortgage_ratio, Language_education_interaction,
                        Unemployment_poverty_interaction),
               names_to = "Engineered_Variable", values_to = "Engineered_Value")

# Scatter plot of engineered variables vs. total crimes
ggplot(engineered_data, aes(x = Engineered_Value, y = Total, color = County)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ Engineered_Variable, scales = "free") +
  labs(
    title = "Relationship Between Engineered Characteristics and Total Arrests",
    x = "Engineered Characteristic Value",
    y = "Total Arrests"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



```
# Correlation heatmap for engineered variables and crime data
correlation_matrix_engineered <- df_top10 |>
  select(
         Mortgage_affordability_index, Log_housing_units,
         Log_mortgage_ratio, Language_education_interaction,
         Unemployment_poverty_interaction,
         Total, `Felony Total`, `Misdemeanor Total`) |>
  cor()

ggplot(melt(correlation_matrix_engineered),
       aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0)+
  labs(
    title = "Correlation Between Engineered Characteristics and Crime Data",
    x = "",
    y = ""
  ) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
    axis.text.y = element_text()
  )
```



New variables after feature engineering

We performed feature engineering on the dataset to create new derived features based on domain knowledge and mathematical transformations.

- **Mortgage_affordability_index**: This feature reflects the relative affordability of housing by comparing the number of households with a mortgage to those without one. Adding +1 ensures no division by zero for areas with zero mortgage-free housing.
- **Log_housing_units**: This is a logarithmic transformation of the total housing units to normalize the data distribution and reduce the influence of extreme outliers.
- **Log_mortgage_ratio**: It applies a logarithmic transformation of mortgage ratio to reduce skewness.
- **Language_education_interaction**: This feature models the compounded effect of language barriers and low education levels. Combining these two social factors allows us to explore whether their combined impact has a stronger association with arrests than the individual factors alone.
- **Unemployment × Poverty Interaction**: Combined "Unemployment Rate" and "Below Poverty Level" to account for the compounded effect of economic stress on crime rates. This interaction models how the combination of unemployment and poverty amplifies arrest rates.

Features included in models

- **Mortgage_affordability_index**
- **Log_housing_units**
- **Log_mortgage_ratio**
- **Language_education_interaction**
- **Unemployment_poverty_interaction**
- **Unemployment_rate**
- **Median_household_income**
- **Below_poverty_level**
- **No_health_insurance**
- **Felony Total**
  - **Drug Felony**
  - **Violent Felony**
  - **DWI Felony**
  - **Other Felony**
- **Misdemeanor Total**
  - **Drug Misdemeanor**
  - **DWI Misdemeanor**
  - **Property Misdemeanor**
  - **Other Misdemeanor**
- **County**
- **Year**

Resampling strategy

We decided to first partition the dataset into an 80% training set and a 20% test set, stratified by the target variablemtotal arrest counts.We further partitioned the training set using 5-fold cross-validation. This approach was chosen because it balances the need for efficient use of limited data and provides a reliable estimate of model performance. The training/test split allows us to evaluate the model's ability to generalize to new data, while cross-validation reduces the risk of overfitting by providing multiple evaluations on subsets of the training data. Together, these strategies ensure a comprehensive assessment of our models before applying them to the test set.

```r
# Select relevant columns for modeling
model_features <- df |>
  select(
    # Socio-economic and housing features
    Mortgage_affordability_index,
    Log_housing_units,
    Log_mortgage_ratio,
    Language_education_interaction,
    Unemployment_poverty_interaction,
    Unemployment_rate,
    Median_household_income,
    Below_poverty_level,
    No_health_insurance,

    # Arrest data features
    `Felony Total`,
    `Drug Felony`,
    `Violent Felony`,
    `DWI Felony`,
    `Other Felony`,
    `Misdemeanor Total`,
    `Drug Misdemeanor`,
    `DWI Misdemeanor`,
    `Property Misdemeanor`,
    `Other Misdemeanor`,
    County,
    Year,

    # Target variable
    Total
  )

set.seed(123)

data_split <- initial_split(model_features, prop = 0.8, strata = Total)
train_data <- training(data_split)
test_data <- testing(data_split)

cv_folds <- vfold_cv(train_data, v = 5, strata = Total)

all_metrics <- metric_set(rmse, rsq, mae)
```

Overview of modeling strategies

Models to be Tested

1. **Poisson Regression:**

- We will test two variations:
  - Using `factor(Year)` to capture year-specific fixed effects.
  - Using `Year` as a continuous variable to capture linear temporal trends.
- Interaction terms (e.g., **Unemployment Rate × Below Poverty Level**) will be included to explore combined effects of key predictors.
- The model's coefficients will provide insights into the strength and direction of relationships between predictors and arrest counts.

2. **Random Forest:**

- Train Random Forest models using the same set of predictors as Poisson regression:
  - Economic variables: `Unemployment Rate`, `Median Household Income`, `Below Poverty Level`, `No Health Insurance`.
  - Engineered features: `Mortgage Affordability Index`, `Log Housing Units`, `Log Mortgage Ratio`, `Language × Education Interaction`, and `Unemployment × Poverty Interaction`.
  - Metadata: `County` and `Year`.
- Tune hyperparameters to optimize performance:
  - Number of trees (`n_estimators`)
  - Maximum tree depth
  - Minimum samples per split

Evaluation Metrics

- **Prediction Performance:**
  - Root Mean Squared Error (RMSE): To evaluate the average magnitude of prediction errors.
  - R-squared (RSQ): To measure how well the model explains the variance in the data.
  - Mean Absolute Error (MAE): To assess the average absolute difference between predicted and observed counts.
- **Variable Importance:**

- Use Random Forest's feature importance scores to identify the most influential predictors.

Null model

```
train_data <- train_data |>
  mutate(County = as.factor(County))

# Set up the null model
null_model <- null_model(mode = "regression") |>
  set_engine("parsnip")

# Build the recipe
null_recipe <- recipe(Total ~ ., data = train_data) |>
  step_rm(County)

# Build the workflow
null_workflow <- workflow() |>
  add_model(null_model) |>
  add_recipe(null_recipe)

# Fit the null model to the training data
null_fit <- fit(null_workflow, data = train_data)

# Predict on the testing set
test_predictions <- predict(null_fit, new_data = test_data) |>
  bind_cols(test_data)

# Calculate test metrics
test_metrics <- test_predictions |>
  metrics(truth = Total, estimate = .pred)
```

```
Warning: A correlation computation is required, but `estimate` is constant and has 0
standard deviation, resulting in a divide by 0 error. `NA` will be returned.
```

```
test_metrics
```

```
# A tibble: 3 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard      17258.
2 rsq     standard         NA
3 mae     standard      12393.
```

The RMSE and MAE are significantly large, at 17,257.99 and 12,393.07, reflecting the simplicity of the null model. The R-squared value is NA, which is expected for a null model, as it does not use any predictors and cannot explain the variance in the data. This serves as a baseline model for comparison with the more complex models that we will create later.

```r
library(poissonreg)


# pre-processing recipe, ensure Year is numeric for one of the models
#convert categorical variables into dummies
base_recipe <- recipe(Total ~ ., data = train_data) |>
  step_mutate(
    Year = as.numeric(Year)
  ) |>
  step_dummy(all_nominal_predictors(), -all_outcomes())

# Variation 1: Poisson regression with factor(Year)
recipe_factor_year <- base_recipe |>
  step_mutate(Year = factor(Year))

model_factor_year <- poisson_reg() |>
  set_engine("glm")

workflow_factor_year <- workflow() |>
  add_recipe(recipe_factor_year) |>
  add_model(model_factor_year)

fit_factor_year <- workflow_factor_year |>
  fit(data = train_data)

# Variation 2: Poisson regression with Year as a continuous variable
recipe_continuous_year <- base_recipe  # numeric Year

model_continuous_year <- poisson_reg() |>
  set_engine("glm")

workflow_continuous_year <- workflow() |>
```

```
  add_recipe(recipe_continuous_year) |>
  add_model(model_continuous_year)

fit_continuous_year <- workflow_continuous_year |>
  fit(data = train_data)

# Evaluate the models on the test set
predictions_factor_year <- fit_factor_year |>
  predict(new_data = test_data) |>
  bind_cols(test_data)

predictions_continuous_year <- fit_continuous_year |>
  predict(new_data = test_data) |>
  bind_cols(test_data)

# Calculate metrics for comparison
metrics_factor_year <- predictions_factor_year |>
  metrics(truth = Total, estimate = .pred)

metrics_continuous_year <- predictions_continuous_year |>
  metrics(truth = Total, estimate = .pred)

# Metrics for Poisson Regression with factor(Year)
print(metrics_factor_year)
```

```
# A tibble: 3 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard     459.
2 rsq     standard       0.999
3 mae     standard     224.
```

```
# Metrics for Poisson Regression with Year as a Continuous Variable
print(metrics_continuous_year)
```

```
# A tibble: 3 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard     771.
2 rsq     standard       0.998
3 mae     standard     351.
```

The analysis shows that using factor(Year) in the Poisson regression model provides better predictive accuracy compared to treating Year as a continuous variable. The model with factor(Year) achieves lower RMSE and MAE values and a slightly higher R^2, indicating it captures year-specific variations and trends more effectively. In contrast, the model using Year as a continuous variable struggles to account for non-linear or year-specific effects, resulting in less accurate predictions. These results suggest that including factor(Year) is a better approach for modeling total arrests when using Posisson regression.

```r
library(tidymodels)
library(ranger)
library(parsnip)
library(vip)
```

```

Attaching package: 'vip'

The following object is masked from 'package:utils':

    vi
```

```r
# Convert County to character before data splitting
df <- df |> mutate(County = as.character(County))

# Split the data
set.seed(123)
data_split <- initial_split(df, prop = 0.8, strata = Total)
train_data <- training(data_split)
test_data <- testing(data_split)

# Feature engineering: focusing on transformations
base_recipe <- recipe(Total ~ ., data = train_data) |>
  step_mutate(
    Year = as.numeric(Year),  # Convert Year to numeric
    Mortgage_affordability_index = log(Mortgage_affordability_index + 1)
    # apply log transform to reduce skewness
  ) |>
  step_dummy(all_nominal_predictors(), -all_outcomes()) |>
  # Dummy encode categorical variables
  step_zv(all_predictors()) |>
  # Remove zero-variance predictors to avoid issues during modeling
  step_normalize(all_numeric_predictors(), -all_outcomes())
```

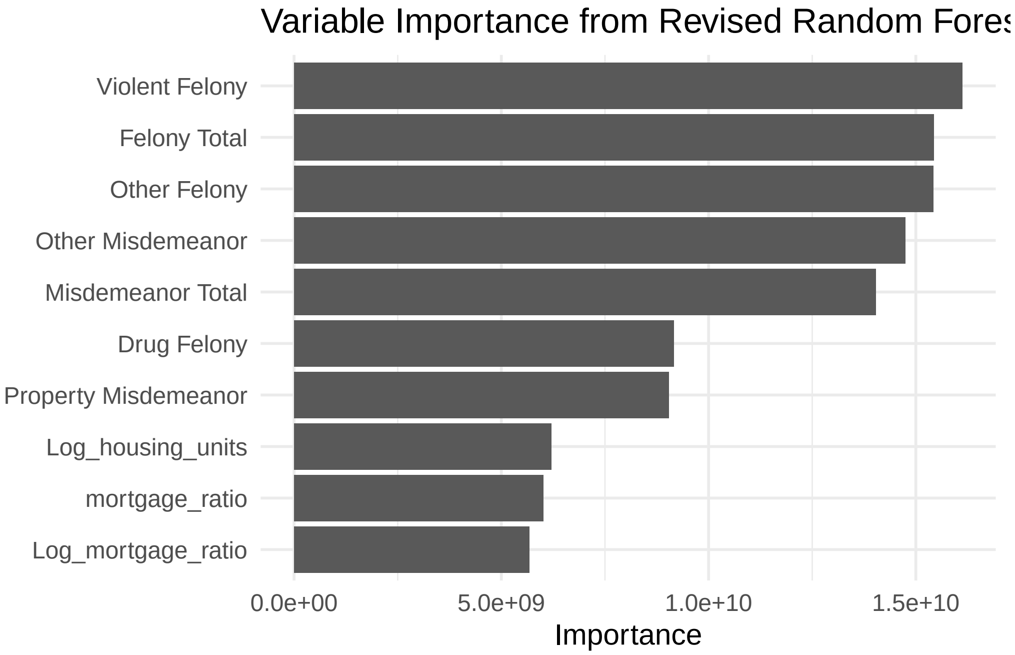
```r
# Normalize all numeric features

# Random Forest specification
rf_spec <- rand_forest(
  mode = "regression",
  trees = tune(),
  mtry = tune(),
  min_n = tune()
) |>
  set_engine("ranger", importance = "impurity")

# Random Forest workflow using the revised recipe
rf_workflow <- workflow() |>
  add_recipe(base_recipe) |>
  add_model(rf_spec)

# Define an expanded grid for hyperparameter tuning
rf_grid <- grid_regular(
  trees(range = c(500, 1500)),
  mtry(range = c(3, 12)),
  min_n(range = c(3, 10)),
  levels = 5
)

# Set up cross validation resamples
cv_folds <- vfold_cv(train_data, v = 10, strata = Total)

# Tune the Random Forest model
set.seed(123)
rf_res <- tune_grid(
  rf_workflow,
  resamples = cv_folds,
  grid = rf_grid,
  metrics = metric_set(rmse, rsq, mae)
)

# check notes if models fail
show_notes(rf_res)
```

```
Great job! No notes to show.
```

```r
# Select the best hyperparameters based on RMSE
best_rf <- select_best(rf_res, metric = "rmse")

# Finalize the workflow with the best parameters
final_rf_workflow <- finalize_workflow(rf_workflow, best_rf)

# Fit the final Random Forest model on training data
final_rf_fit <- final_rf_workflow |> fit(data = train_data)

# Predict and evaluate on the test set
rf_predictions <- final_rf_fit |>
  predict(new_data = test_data) |>
  bind_cols(test_data)

rf_metrics <- rf_predictions |>
  metrics(truth = Total, estimate = .pred)

# Print evaluation metrics
print(rf_metrics)
```

```
# A tibble: 3 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard     966.
2 rsq     standard       0.997
3 mae     standard     331.
```

```r
# Extract and visualize feature importance
rf_imp <- final_rf_fit |>
  extract_fit_parsnip() |>
  vi()

vip(rf_imp) +
  labs(title = "Variable Importance from Revised Random Forest Model") +
  theme_minimal()
```



```
#ggsave(
#  filename = "report_files/figure-typst/random_forest-1.png",
#  plot = last_plot(), width = 8, height = 6, dpi = 300
#)
```

The Random Forest Model has shown improvement, achieving an RMSE of 966.31 and an R^2 of 0.9969, showing very good model fit and predictive accuracy. The MAE value of 331.47 further highlights the model's improved capability to closely predict the actual arrest values. By exploring the feature importance, we determined that crime-specific features, such as violent felonies and total felonies, are the strongest drivers of arrest counts. While less impactful, housing and mortgage-related variables contribute additional context to the model. They may indirectly capture socioeconomic conditions that correlate with crime levels.

Initial results

The objective of this project was to predict the total number of adult arrests in New York State counties based on key social, economic, and housing factors from 2015 to 2022. By building predictive models, we aimed to understand how these factors influence arrest counts, ultimately helping policymakers make informed decisions regarding crime prevention and resource allocation.

Three models were evaluated: a **Null Model**, a **Poisson Regression Model**, and a **Random Forest Model**.

1. **Null Model:** The null model serves as a baseline for comparison. It does not include any predictors and only predicts the mean value of the target variable (total number of adult arrests).

Metric	**Estimator**	**Estimate**
RMSE	Standard	17,257.99
R^2	Standard	NA
MAE	Standard	12,393.07

Root Mean Squared Error (RMSE) is 17257.99, which is quite high. This reflects the poor predictive accuracy of the null model, as it doesn't account for any predictors.

R^2 is NA since the null model does not use any predictors. Mean Absolute Error (MAE) is 12393.07, which also highlights the high level of error in the model's predictions. The null model's high RMSE and MAE values demonstrate that using just the mean of the target variable is insufficient for accurately predicting the total number of adult arrests.

2. **Poisson Regression Model:** We applied a Poisson regression model with two variations:

Variation 1 used factor(Year) to capture year-specific fixed effects. Variation 2 used Year as a continuous variable to capture linear trends. We observed that Variation 1 provided better predictive accuracy, so we report those results here.

Metric	**Estimator**	**Estimate**
RMSE	Standard	771.42
R^2	Standard	0.9980
MAE	Standard	350.76

The Root Mean Squared Error (RMSE) of 771.42 shows a substantial improvement in model accuracy compared to the null model.

R^2 of 0.9980 shows that the model explains 99.8% of the variance in arrest counts. This suggests it has captured most of the relevant factors influencing arrest rates.

Mean Absolute Error (MAE) is 350.76, significantly lower than the null model, meaning better prediction of actual values. The Poisson regression model provided strong predictive performance, especially in capturing year-specific trends, which suggests the importance of temporal factors in arrest trends.

3. **Random Forest Model:** The random forest model was trained using the key predictors from the dataset. Hyperparameter tuning was performed to optimize the performance.

Metric	Estimator	Estimate
RMSE	Standard	966.31
R^2	Standard	0.9969
MAE	Standard	331.47

The Root Mean Squared Error (RMSE) of 966.31 for the Random Forest model, while higher than the 771.42 RMSE from the Poisson Regression model. This suggests that the Random Forest model is now much closer in performance to the Poisson regression, even though it does not outperform it.

The R^2 value of 0.9969 shows that the Random Forest model can explain 99.69% of the variance in arrest counts, which is very close to the 0.9980 R^2 value achieved by the Poisson regression. While the Random Forest model is slightly behind in terms of variance explained, it still effectively captures most of the important relationships in the data.

The Mean Absolute Error (MAE) of 331.47 for the Random Forest model is lower than the 350.76 from the Poisson Regression model, showing better accuracy in predicting actual arrest counts on average. This suggests that the Random Forest model offers more precise predictions for individual values, despite having a slightly higher RMSE.

Overall, the Random Forest model has closed the gap with the Poisson Regression model, with a lower MAE and a competitive R^2 value. While it has a marginally higher RMSE, the improved MAE suggests that the Random Forest model is highly competitive, particularly for accurately predicting specific arrest values.

##Summary and Insights##

The Null Model served as a baseline, providing predictions based solely on the mean of the arrest data. This model had high errors, indicating it was insufficient for accurately modeling arrest trends.

The Poisson Regression Model demonstrated the best overall performance. The inclusion of year-specific fixed effects allowed this model to effectively capture temporal patterns in arrest trends, suggesting the importance of accounting for year-to-year variability. These results indicate that the Poisson Regression Model captures nearly all of the variance in the arrest counts, providing the most reliable predictive model among those tested.

The Random Forest Model also achieved competitive results, which is slightly better than the Poisson Regression Model, showing that it provides more accurate predictions for individual arrest counts. However, it has a higher RMSE compared to the Poisson Regression model, which implies slightly larger errors for some of the predictions.

The results from both the Poisson Regression and Random Forest models suggest that a mix of crime-specific and socioeconomic features is crucial for understanding arrest patterns across different counties and years.

##Conclusion##

Based on the results, the Poisson Regression Model stands out as the best performing model overall. With an RMSE of 771.42 and an R^2 of 0.9980, this model effectively captures 99.8% of the variance in arrest counts and provides robust predictive capabilities. Its use of year-specific fixed effects highlights the importance of capturing temporal dynamics when predicting arrest trends. This model is particularly useful for understanding overarching trends and for making reliable policy decisions to allocate resources where needed.

The Random Forest Model, although slightly behind in RMSE, provides comparable performance with a lower MAE of 331.47, making it useful for accurate individual-level predictions. This model's variable importance analysis also provides valuable insights into the relative influence of different crime categories and socioeconomic factors on arrest rates, thereby supporting more granular policy decisions.