Softwarica College of IT & E-commerce

ST5014CEM Data Science for Developers

August 15, 2024

Siddhartha Neupane

Submitted By:

Aabrity Dhungana

202270

Drive Link

https://drive.google.com/file/d/1Gp8zyzTGFemscOINlhBzf5n-tORmrILe/view?usp=drive_link

Table of Contents

Table of Figures

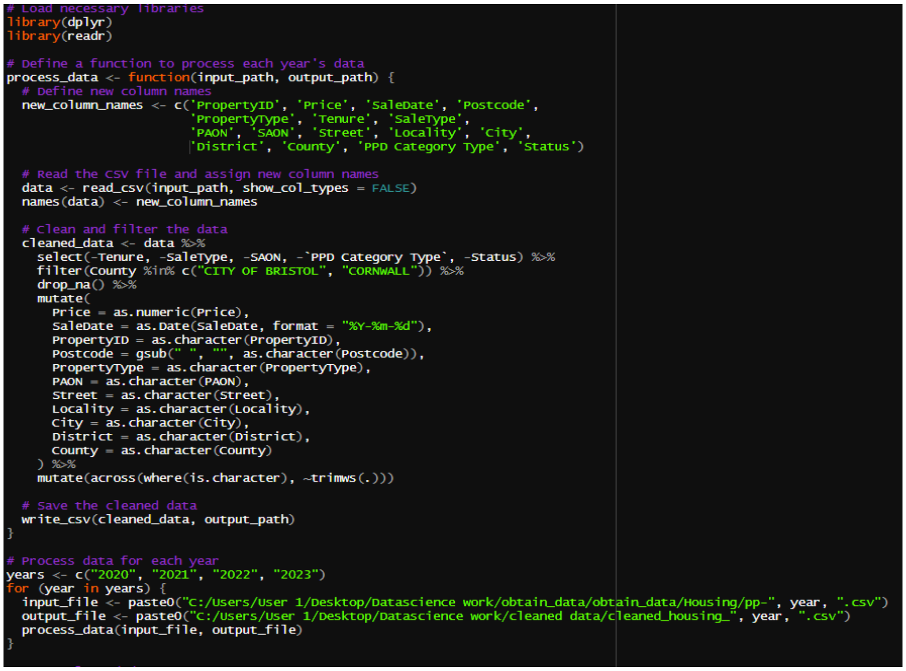
Introduction

The report is about a setup and review that were enacted in an instance when relatives of data analyst seeks residence and there is need for a recommendation involving the expertise of the analyst to identify the best towns where one can invest in property within Bristol and Cornwall. A successful investor in real estate takes into account many aspects such as housing prices, availability of fast internet speed, crime rates, and school ranking statistics. These factors are very useful when trying to identify the desirability and long-term worthiness of some property by revealing various economic opportunities or lifestyles existing within different areas. As such, incorporating these elements into its recommendations will enable the user to have an idea of all towns so that it select an investment option which will be both profitable and sustainable, matching with current market patterns.

Cleaning Data

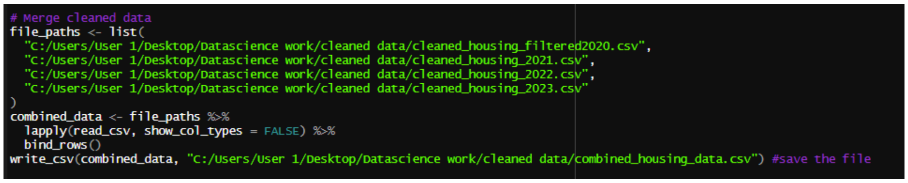
The project started by collecting data from several different places so it could have an all-inclusive database. The next stage according to the data science life cycle was cleaning the data. To do this, the datasets were loaded in the working environment. Some cleaning steps were taken, which included dealing with problems such as missing values and inconsistencies and formatting errors and thus made sure that data was both accurate and prepared for analysis.

House pricing dataset cleaning



```r
# Load necessary libraries
library(dplyr)
library(readr)

# Define a function to process each year's data
process_data <- function(input_path, output_path) {
  # Define new column names
  new_column_names <- c('PropertyID', 'Price', 'SaleDate', 'Postcode',
                        'PropertyType', 'Tenure', 'SaleType',
                        'PAON', 'SAON', 'Street', 'Locality', 'City',
                        'District', 'County', 'PPD Category Type', 'Status')

  # Read the CSV file and assign new column names
  data <- read_csv(input_path, show_col_types = FALSE)
  names(data) <- new_column_names

  # Clean and filter the data
  cleaned_data <- data %>%
    select(-Tenure, -SaleType, -SAON, -`PPD Category Type`, -Status) %>%
    filter(County %in% c("CITY OF BRISTOL", "CORNWALL")) %>%
    drop_na() %>%
    mutate(
      Price = as.numeric(Price),
      SaleDate = as.Date(SaleDate, format = "%Y-%m-%d"),
      PropertyID = as.character(PropertyID),
      Postcode = gsub(" ", "", as.character(Postcode)),
      PropertyType = as.character(PropertyType),
      PAON = as.character(PAON),
      Street = as.character(Street),
      Locality = as.character(Locality),
      City = as.character(City),
      District = as.character(District),
      County = as.character(County)
    ) %>%
    mutate(across(where(is.character), ~trimws(.)))

  # Save the cleaned data
  write_csv(cleaned_data, output_path)
}

# Process data for each year
years <- c("2020", "2021", "2022", "2023")
for (year in years) {
  input_file <- paste0("C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/Housing/pp-", year, ".csv")
  output_file <- paste0("C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_", year, ".csv")
  process_data(input_file, output_file)
}
```

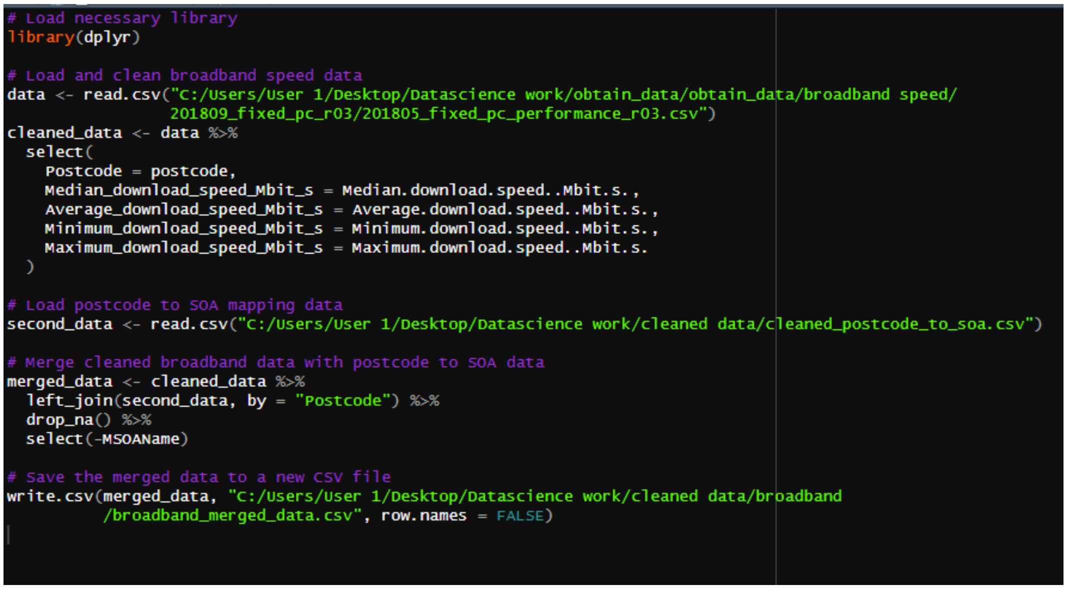
Figure 1:Housing Cleaning Code



```r
# Merge cleaned data
file_paths <- list(
  "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_filtered2020.csv",
  "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_2021.csv",
  "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_2022.csv",
  "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_2023.csv"
)
combined_data <- file_paths %>%
  lapply(read_csv, show_col_types = FALSE) %>%
  bind_rows()
write_csv(combined_data, "C:/Users/User 1/Desktop/Datascience work/cleaned data/combined_housing_data.csv") #save the file
```

This R script cleans and combines housing data from multiple years. It starts by loading necessary libraries and reading CSV files. The core function, `process_data`, renames columns, removes unnecessary ones, selects specific counties, and adjusts data types. It also ensures proper formatting of postcodes and removes excess spaces in text fields. The function is applied to housing data files for 2020-2023, with each cleaned dataset saved to a new file.

Broadband speed dataset cleaning

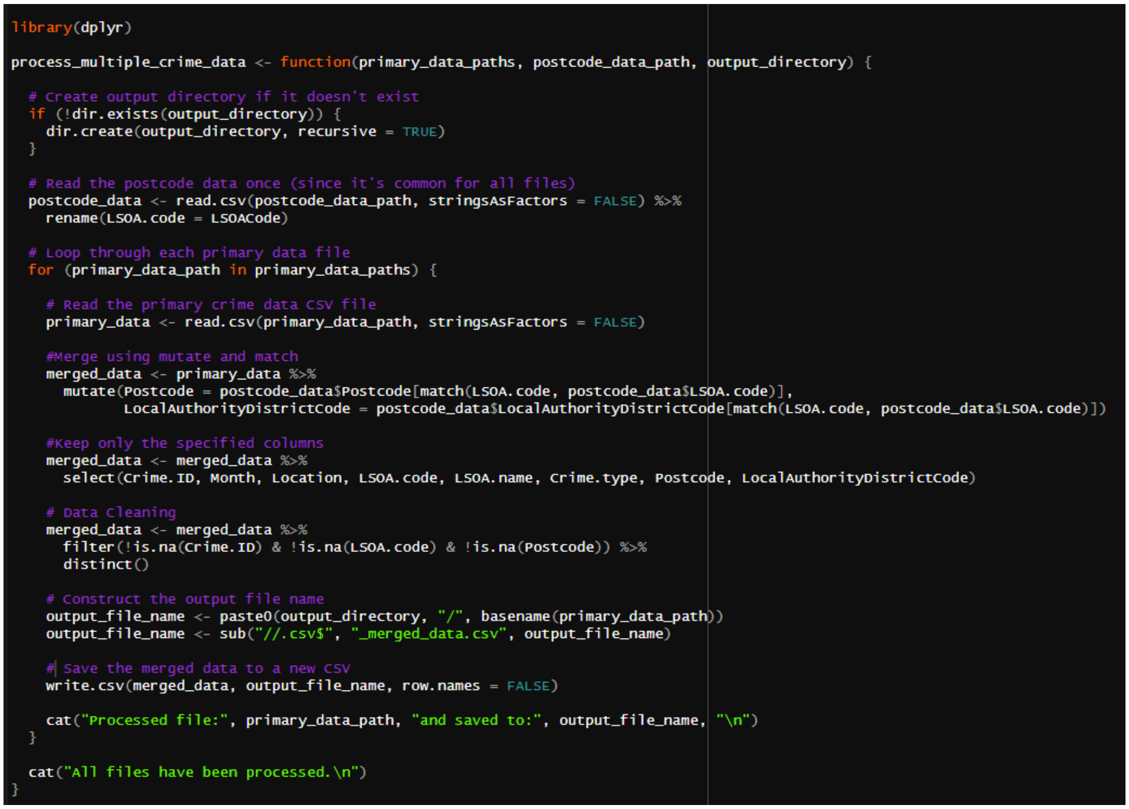


```r
# Load necessary library
library(dplyr)

# Load and clean broadband speed data
data <- read.csv("C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/broadband speed/
                 201809_fixed_pc_r03/201805_fixed_pc_performance_r03.csv")
cleaned_data <- data %>%
  select(
    Postcode = postcode,
    Median_download_speed_Mbit_s = Median.download.speed..Mbit.s.,
    Average_download_speed_Mbit_s = Average.download.speed..Mbit.s.,
    Minimum_download_speed_Mbit_s = Minimum.download.speed..Mbit.s.,
    Maximum_download_speed_Mbit_s = Maximum.download.speed..Mbit.s.
  )

# Load postcode to SOA mapping data
second_data <- read.csv("C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_postcode_to_soa.csv")

# Merge cleaned broadband data with postcode to SOA data
merged_data <- cleaned_data %>%
  left_join(second_data, by = "Postcode") %>%
  drop_na() %>%
  select(-MSOAName)

# Save the merged data to a new CSV file
write.csv(merged_data, "C:/Users/User 1/Desktop/Datascience work/cleaned data/broadband
          /broadband_merged_data.csv", row.names = FALSE)
```

Figure 2:Broadband speed cleaning code

This R script cleans and consolidates postcode-to-LSOA mapping for broadband speed data. It begins by loading the `dplyr` library and reading broadband speed data from a CSV file, selecting and renaming relevant columns. It then loads another CSV with postcode-to-LSOA codes, merges it with the broadband data, removes rows with missing values, and drops an irrelevant column.

Crime dataset cleaning

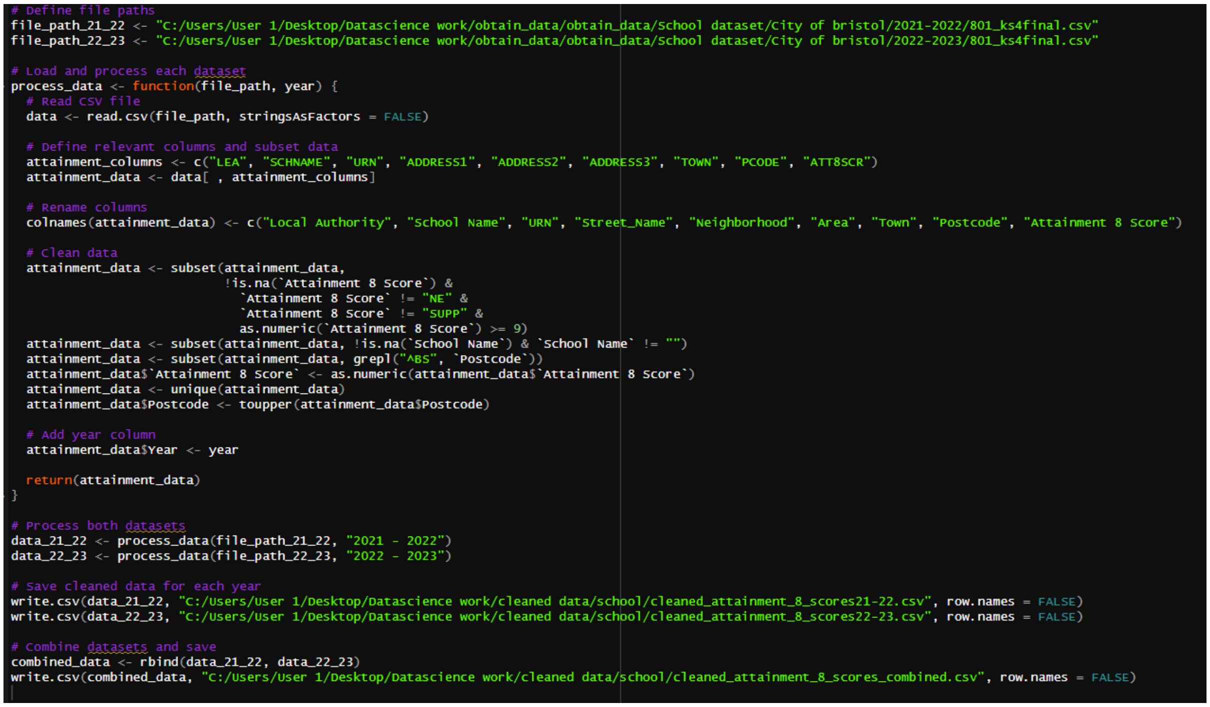


```r
library(dplyr)

process_multiple_crime_data <- function(primary_data_paths, postcode_data_path, output_directory) {

  # Create output directory if it doesn't exist
  if (!dir.exists(output_directory)) {
    dir.create(output_directory, recursive = TRUE)
  }

  # Read the postcode data once (since it's common for all files)
  postcode_data <- read.csv(postcode_data_path, stringsAsFactors = FALSE) %>%
    rename(LSOA.code = LSOACode)

  # Loop through each primary data file
  for (primary_data_path in primary_data_paths) {

    # Read the primary crime data CSV file
    primary_data <- read.csv(primary_data_path, stringsAsFactors = FALSE)

    #Merge using mutate and match
    merged_data <- primary_data %>%
      mutate(Postcode = postcode_data$Postcode[match(LSOA.code, postcode_data$LSOA.code)],
             LocalAuthorityDistrictCode = postcode_data$LocalAuthorityDistrictCode[match(LSOA.code, postcode_data$LSOA.code)])

    #Keep only the specified columns
    merged_data <- merged_data %>%
      select(Crime.ID, Month, Location, LSOA.code, LSOA.name, Crime.type, Postcode, LocalAuthorityDistrictCode)

    # Data Cleaning
    merged_data <- merged_data %>%
      filter(!is.na(Crime.ID) & !is.na(LSOA.code) & !is.na(Postcode)) %>%
      distinct()

    # Construct the output file name
    output_file_name <- paste0(output_directory, "/", basename(primary_data_path))
    output_file_name <- sub("//.csv$", "_merged_data.csv", output_file_name)

    # Save the merged data to a new CSV
    write.csv(merged_data, output_file_name, row.names = FALSE)

    cat("Processed file:", primary_data_path, "and saved to:", output_file_name, "\n")
  }

  cat("All files have been processed.\n")
}
```

Figure 3:Crime cleaning Code

Given the large number of crime files, a function to load and process them is optimal. The function first ensures the output directory exists or creates it if necessary. It reads the postcode data once and merges each crime file with this data based on a common `LSOA.code`, selecting specific columns. It then removes rows with missing key values and eliminates duplicates. The cleaned and merged datasets are saved as new CSV files in the output folder, with a confirmation message printed for each file processed and a final completion message after all files are handled.

School dataset cleaning

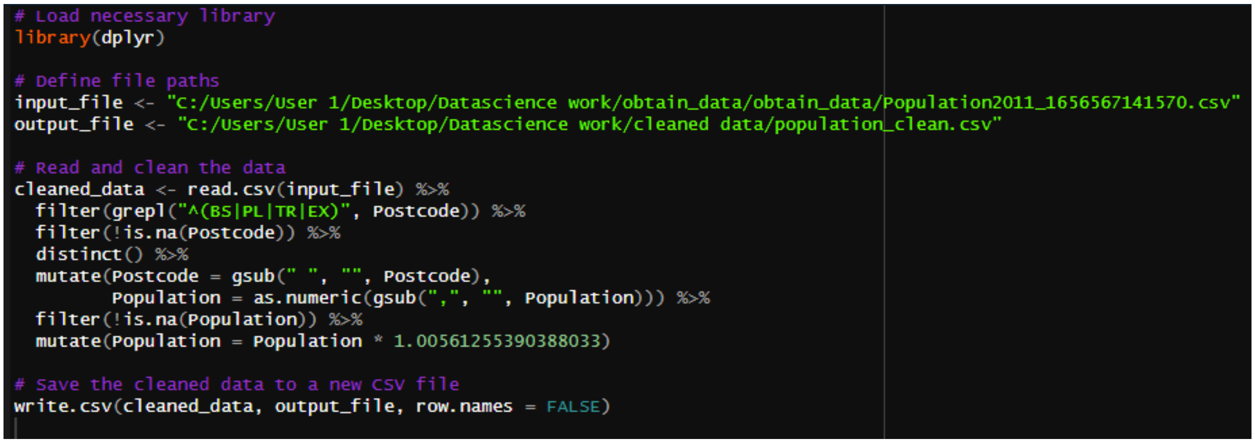


```r
# Define file paths
file_path_21_22 <- "C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/School dataset/City of bristol/2021-2022/801_ks4final.csv"
file_path_22_23 <- "C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/School dataset/City of bristol/2022-2023/801_ks4final.csv"

# Load and process each dataset
process_data <- function(file_path, year) {
  # Read CSV file
  data <- read.csv(file_path, stringsAsFactors = FALSE)

  # Define relevant columns and subset data
  attainment_columns <- c("LEA", "SCHNAME", "URN", "ADDRESS1", "ADDRESS2", "ADDRESS3", "TOWN", "PCODE", "ATT8SCR")
  attainment_data <- data[ , attainment_columns]

  # Rename columns
  colnames(attainment_data) <- c("Local Authority", "School Name", "URN", "Street_Name", "Neighborhood", "Area", "Town", "Postcode", "Attainment 8 Score")

  # Clean data
  attainment_data <- subset(attainment_data,
                            !is.na(`Attainment 8 Score`) &
                              `Attainment 8 Score` != "NE" &
                              `Attainment 8 Score` != "SUPP" &
                              as.numeric(`Attainment 8 Score`) >= 9)
  attainment_data <- subset(attainment_data, !is.na(`School Name`) & `School Name` != "")
  attainment_data <- subset(attainment_data, grepl("^BS", `Postcode`))
  attainment_data$`Attainment 8 Score` <- as.numeric(attainment_data$`Attainment 8 Score`)
  attainment_data <- unique(attainment_data)
  attainment_data$Postcode <- toupper(attainment_data$Postcode)

  # Add year column
  attainment_data$Year <- year

  return(attainment_data)
}

# Process both datasets
data_21_22 <- process_data(file_path_21_22, "2021 - 2022")
data_22_23 <- process_data(file_path_22_23, "2022 - 2023")

# Save cleaned data for each year
write.csv(data_21_22, "C:/Users/User 1/Desktop/Datascience work/cleaned data/school/cleaned_attainment_8_scores21-22.csv", row.names = FALSE)
write.csv(data_22_23, "C:/Users/User 1/Desktop/Datascience work/cleaned data/school/cleaned_attainment_8_scores22-23.csv", row.names = FALSE)

# Combine datasets and save
combined_data <- rbind(data_21_22, data_22_23)
write.csv(combined_data, "C:/Users/User 1/Desktop/Datascience work/cleaned data/school/cleaned_attainment_8_scores_combined.csv", row.names = FALSE)
```

Figure 4: School data cleaning code

The R script processes and cleans Attainment 8 score data for the academic years 2021-2022 and 2022-2023. It specifies file paths, loads the data, and uses a `process_data` function to subset relevant columns, rename them, and clean the data by removing invalid scores, rows without school names, and standardizing postcodes. The cleaned datasets are then merged and saved as a new CSV file, ensuring a unified output for further analysis. This modular approach enhances code readability and maintenance.

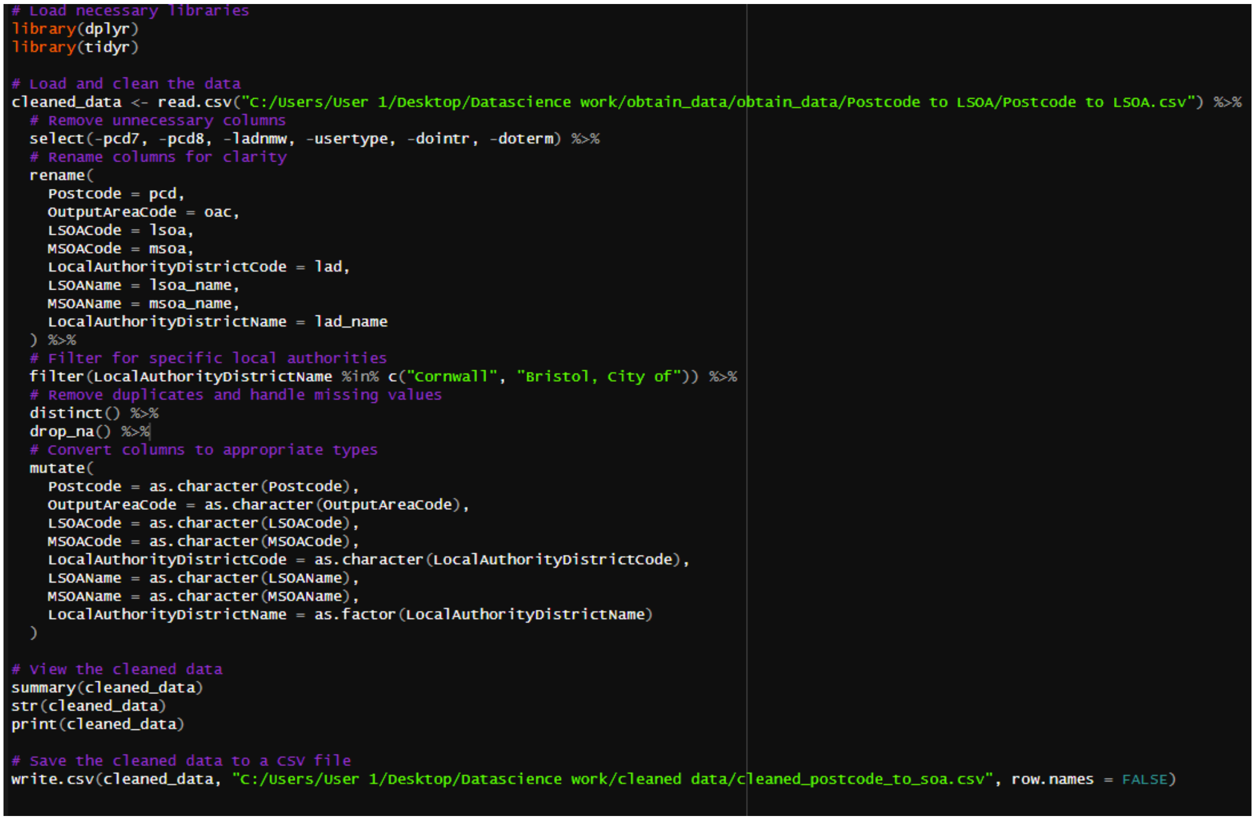
Population and LSOA dataset cleaning



```
# Load necessary library
library(dplyr)

# Define file paths
input_file <- "C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/Population2011_1656567141570.csv"
output_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/population_clean.csv"

# Read and clean the data
cleaned_data <- read.csv(input_file) %>%
  filter(grepl("^(BS|PL|TR|EX)", Postcode)) %>%
  filter(!is.na(Postcode)) %>%
  distinct() %>%
  mutate(Postcode = gsub(" ", "", Postcode),
         Population = as.numeric(gsub(",", "", Population))) %>%
  filter(!is.na(Population)) %>%
  mutate(Population = Population * 1.00561255390388033)

# Save the cleaned data to a new CSV file
write.csv(cleaned_data, output_file, row.names = FALSE)
```

Figure 5: Population Code cleaning

The process begins by importing datasets and filtering rows to include only those with postcodes starting with 'BS', 'PL', 'TR', or 'EX', while dropping rows with missing columns. Duplicates are removed, and spaces in the postcode field are trimmed. Population data is cleaned by removing commas and converting values to numeric form, with non-convertible rows removed. Population values are then updated to 2023 using a growth factor, and the cleaned data is saved as a new CSV file. This method ensures the dataset is properly prepared for further analysis.



```r
# Load necessary libraries
library(dplyr)
library(tidyr)

# Load and clean the data
cleaned_data <- read.csv("C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/Postcode to LSOA/Postcode to LSOA.csv") %>%
  # Remove unnecessary columns
  select(-pcd7, -pcd8, -ladnmw, -usertype, -dointr, -doterm) %>%
  # Rename columns for clarity
  rename(
    Postcode = pcd,
    OutputAreaCode = oac,
    LSOACode = lsoa,
    MSOACode = msoa,
    LocalAuthorityDistrictCode = lad,
    LSOAName = lsoa_name,
    MSOAName = msoa_name,
    LocalAuthorityDistrictName = lad_name
  ) %>%
  # Filter for specific local authorities
  filter(LocalAuthorityDistrictName %in% c("Cornwall", "Bristol, City of")) %>%
  # Remove duplicates and handle missing values
  distinct() %>%
  drop_na() %>%
  # Convert columns to appropriate types
  mutate(
    Postcode = as.character(Postcode),
    OutputAreaCode = as.character(OutputAreaCode),
    LSOACode = as.character(LSOACode),
    MSOACode = as.character(MSOACode),
    LocalAuthorityDistrictCode = as.character(LocalAuthorityDistrictCode),
    LSOAName = as.character(LSOAName),
    MSOAName = as.character(MSOAName),
    LocalAuthorityDistrictName = as.factor(LocalAuthorityDistrictName)
  )

# View the cleaned data
summary(cleaned_data)
str(cleaned_data)
print(cleaned_data)

# Save the cleaned data to a CSV file
write.csv(cleaned_data, "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_postcode_to_soa.csv", row.names = FALSE)
```

Figure 6: Lsoa cleaning code

The script processes postcode-to-SOA mapping data from a CSV file by cleaning and transforming it. This includes removing unwanted columns, renaming remaining ones, filtering records to Cornwall and Bristol, identifying and deleting duplicates, handling missing values, and adjusting column types if needed. The cleaned data is then verified, summarized, printed, and saved as a new CSV file.

EDA (Exploratory Data Analysis)

House Pricing

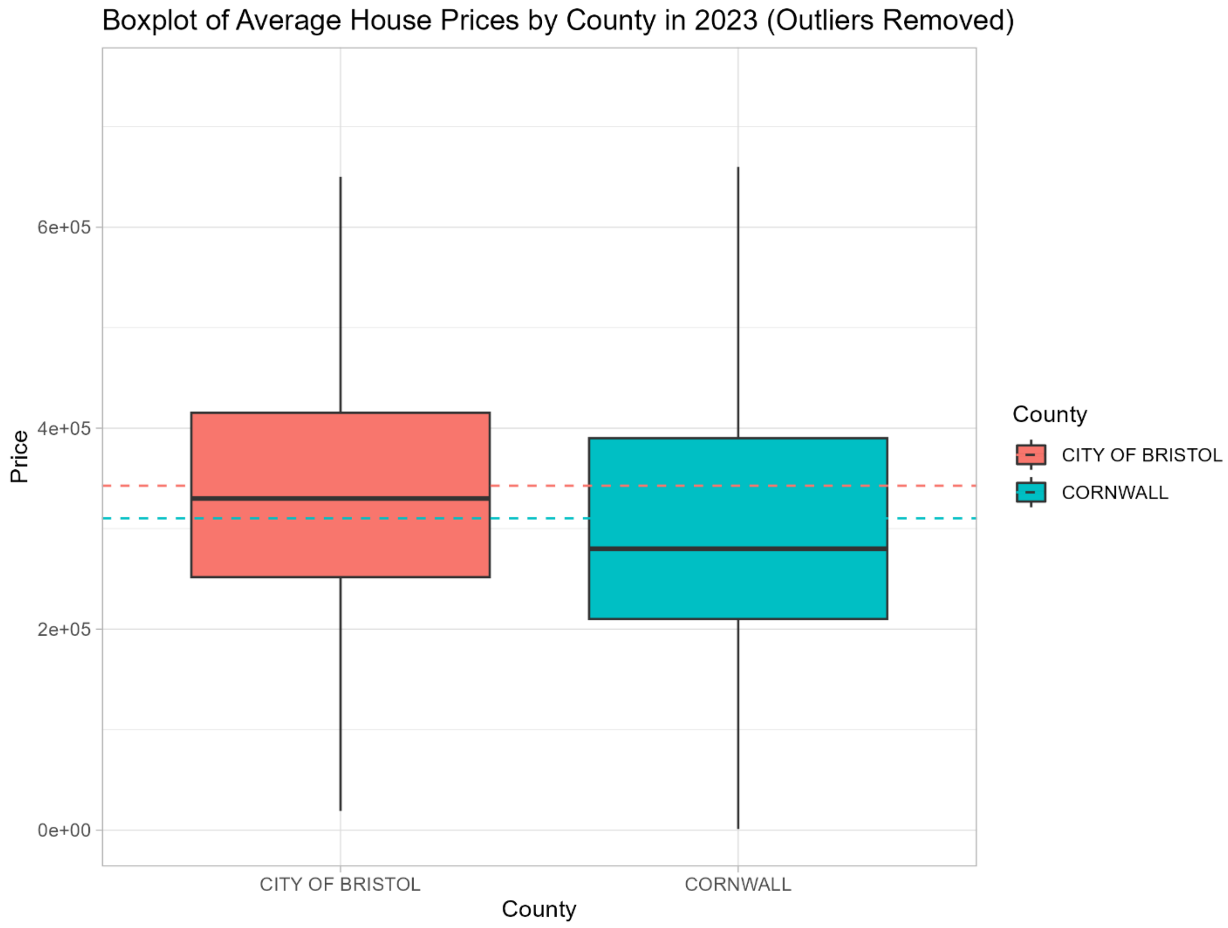


Figure 7: Boxplot Average house price

In 2023, Bristol's housing market shows a higher median price and wider interquartile range (IQR), indicating greater variability. Cornwall has a lower median price and narrower IQR, reflecting a more stable and uniform market.

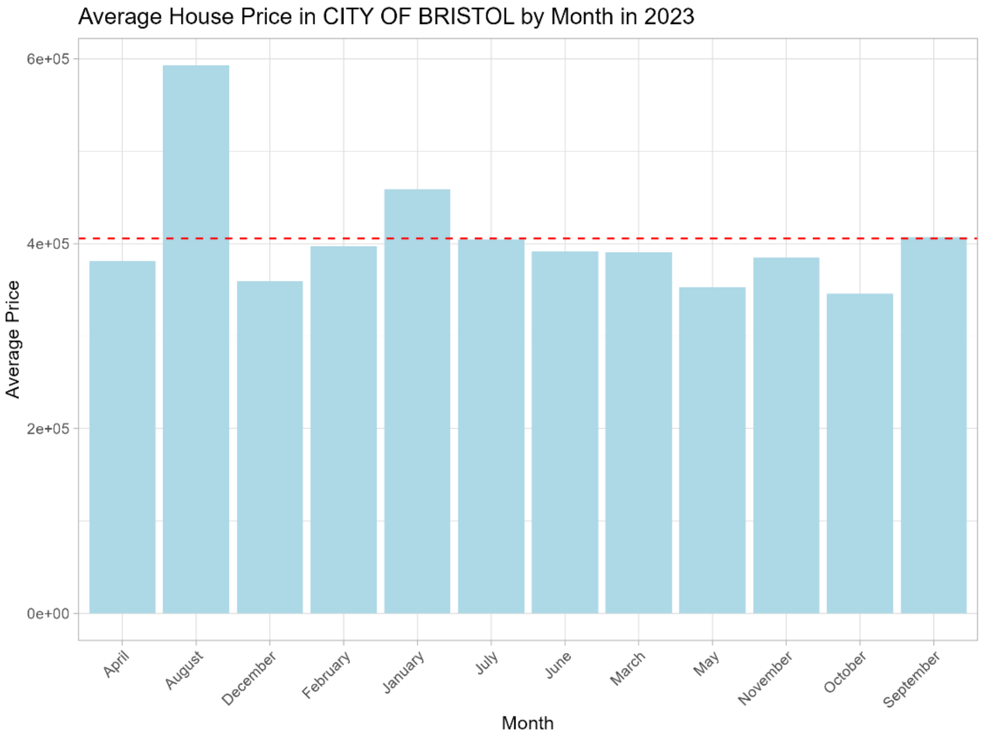


Figure 8 : Average house price bar chart Bristol

In 2023, Bristol’s house prices peak above £600,000 in August due to increased demand, while February and December see lows below £400,000, likely from seasonal slowdowns. Prices generally stay around £400,000, reflecting seasonal market trends.

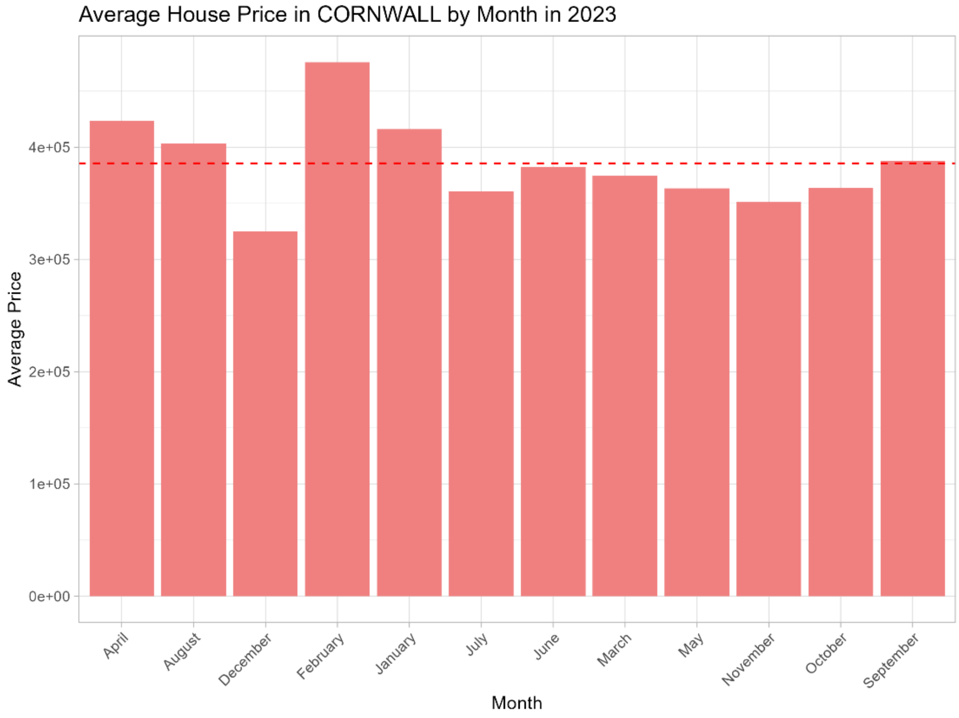


Figure 9 Average house price bar chart Cornwall

In 2023, Cornwall’s housing market shows seasonal fluctuations with a peak in February and a trough in June, remaining generally stable and recovering towards year-end. Timing transactions according to these trends could be beneficial for better deals or higher prices.

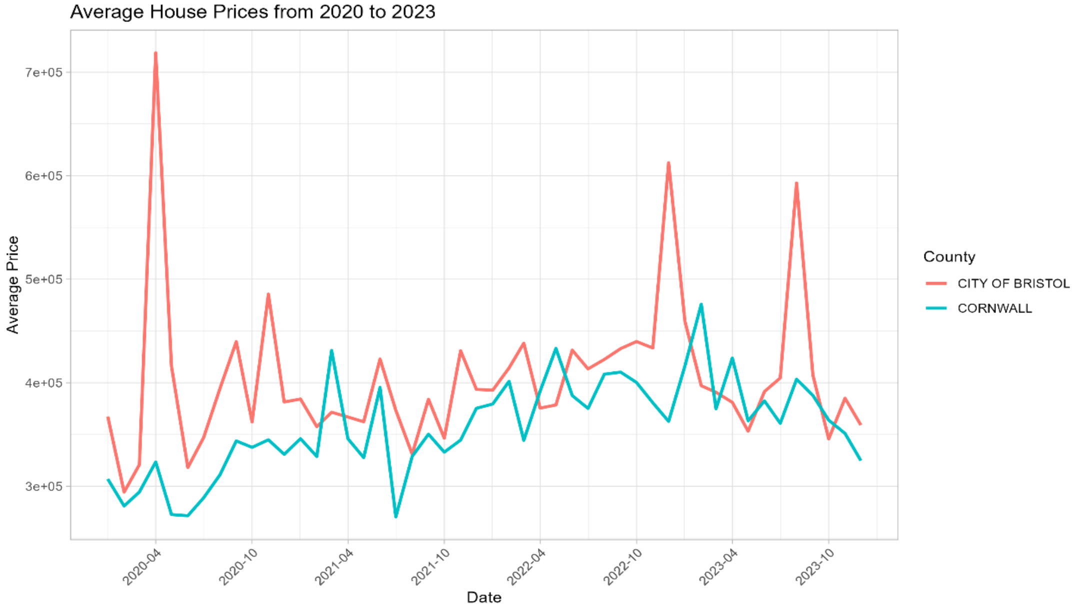


Figure 10: Average House Price Line graph both counties

From 2020 to 2023, Bristol's housing market is volatile with sharp price spikes over £700,000, while Cornwall remains stable between £300,000 and £450,000. Post-2021, both markets stabilize, but Bristol continues to show greater fluctuations compared to Cornwall.

Broad Band

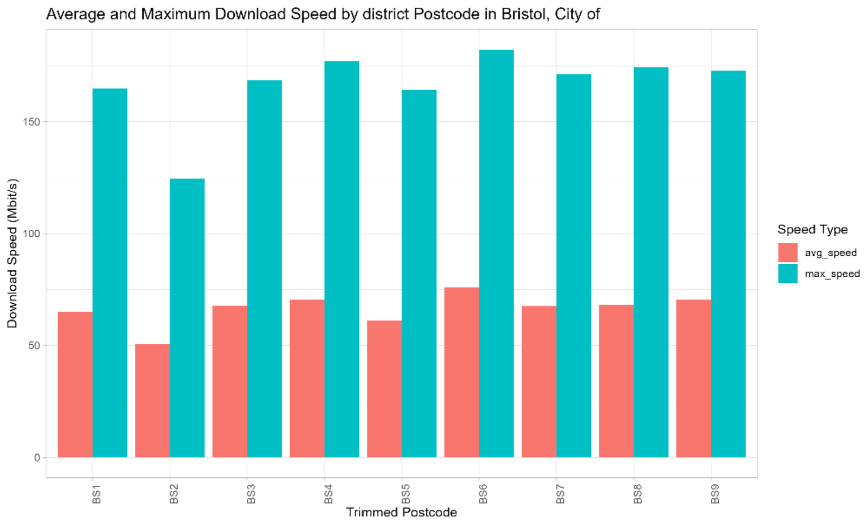


Figure 11 Average and maximum download speed for Bristol Barchart

The bar chart shows a significant gap between maximum (150-175 Mbps) and average (50-75 Mbps) download speeds across Bristol's postcode districts. BS1 and BS6 have the highest average speeds, while BS2 and BS5 have the lowest, indicating that network congestion or distance from exchanges may be impacting consistency.

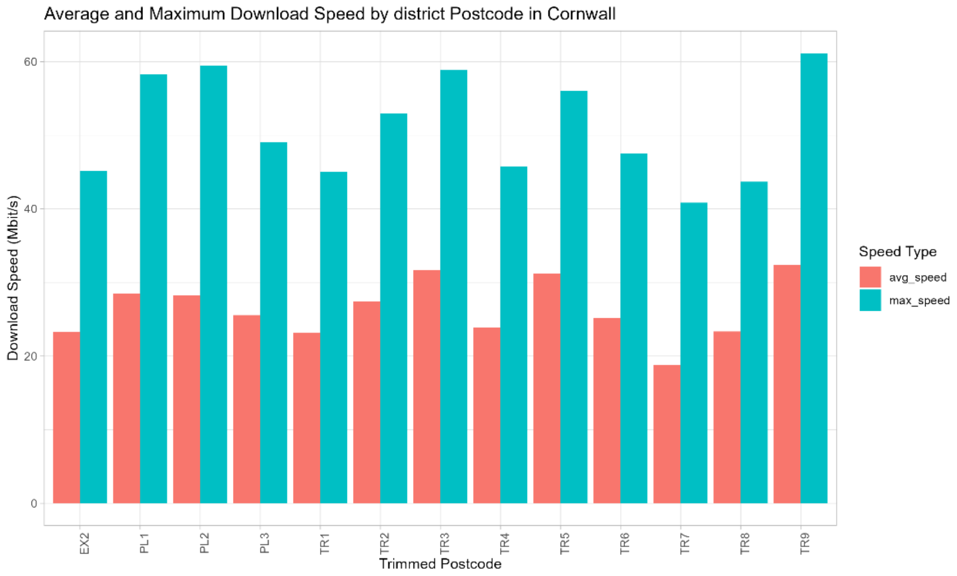


Figure 12: Average and maximum download speed for Cornwall barchat

The bar chart shows that "TR" postcodes in Cornwall, especially around Truro, have higher maximum download speeds, indicating superior infrastructure. In contrast, "PL" and "EX" postcodes have more uniform speeds, suggesting standardized service. A consistent gap between average and maximum speeds across all postcodes highlights underutilized infrastructure, with

TR9 boasting the highest maximum speed due to likely recent upgrades, revealing regional disparities in broadband investment.

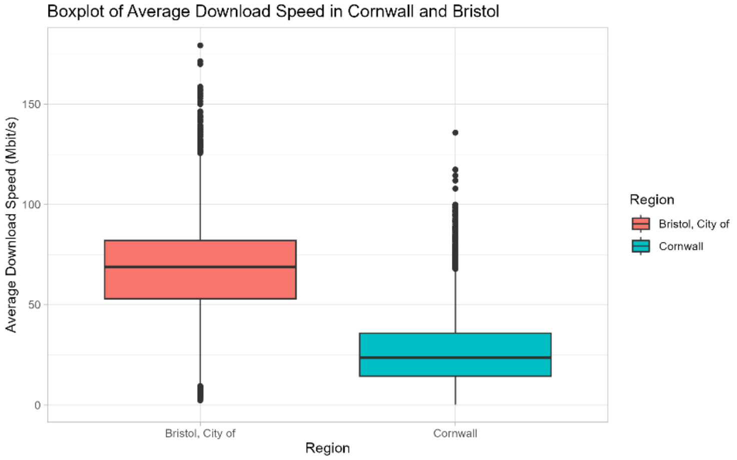


Figure 13: Average download speed in Cornwall and Bristoll Boxplot

The boxplot reveals Bristol's more consistent and higher download speeds, while Cornwall shows greater variability and lower median speeds, underscoring the need for more uniform broadband infrastructure in Cornwall.

Crime

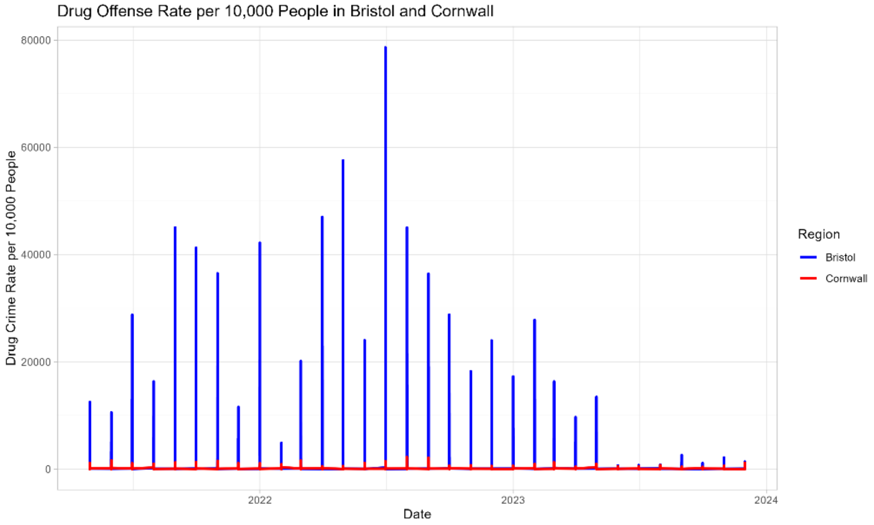


Figure 14: Line chart for Drug offense rate per 10000 people in Bristoll and Cornwall

The analysis shows Bristol with higher drug offense rates and noticeable peaks, particularly in mid-2022, possibly due to seasonal factors or spikes in activity. Rates decline afterward, suggesting successful interventions. Cornwall's rates remain low and stable, indicating fewer disturbances. By the end of the period, both regions stabilize, with Bristol's rates leveling off and Cornwall maintaining its steady trend, reflecting Bristol's fluctuating challenges and Cornwall's stability.

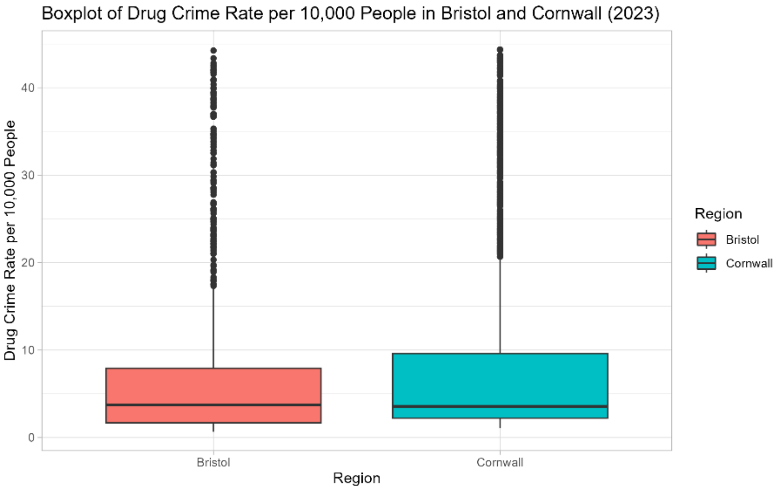


Figure 15: Boxplot Drug Crime Rate per 10000 in Both Counties

The 2023 boxplot shows right-skewed drug crime rate distributions in both Bristol and Cornwall, with most rates low but occasional significant spikes. Bristol's higher median indicates more consistently elevated drug activity, while Cornwall has a lower median but more extreme outliers, reflecting sharp, infrequent surges. Both regions share similar interquartile ranges, suggesting comparable core variability. Bristol's drug crime is generally stable, while Cornwall's pronounced peaks may require targeted attention.

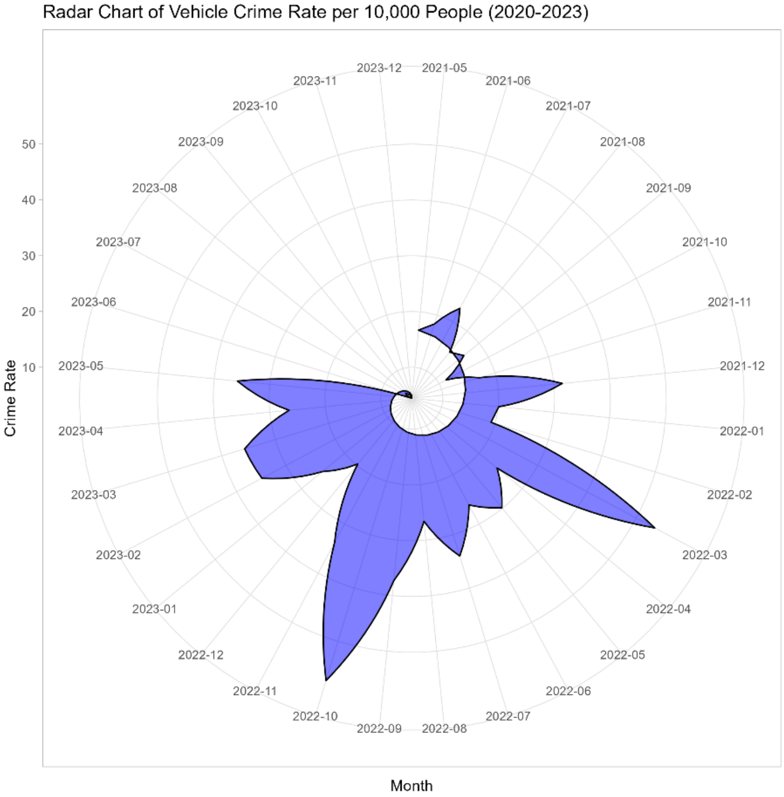


Figure 16: Radar chart Vehicle Crime Rate per 10000 people (20-23)

The radar chart shows that vehicle crime rates peak during warmer months (May to August) and drop in winter (December to February), likely due to seasonal activity changes.

Over the years, there is a gradual decline in overall crime, with less pronounced peaks from 2022 onward, suggesting improved prevention, law enforcement, or vehicle security.

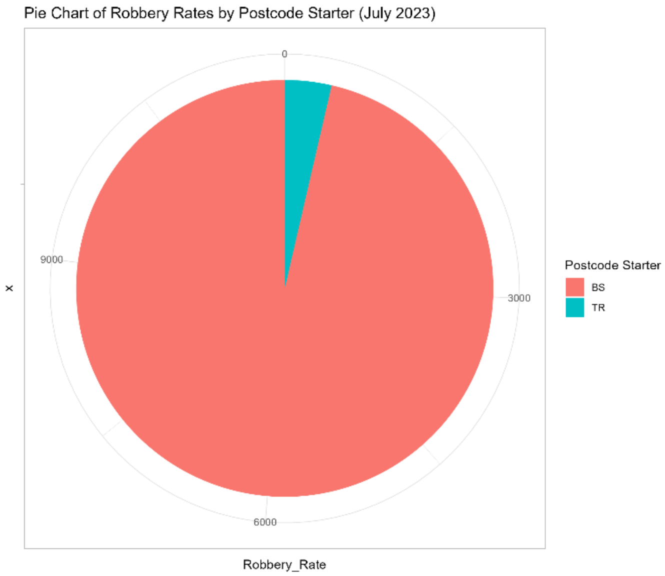


Figure 17: Pie chart of Robbery july 2023

The pie chart reveals a significant concentration of robbery incidents in the "BS" postcode area, indicating much higher rates compared to the rural "TR" area. This urban-rural divide suggests that factors like population density and socio-economic conditions contribute to the imbalance, highlighting the need for targeted crime prevention in the "BS" region.

School

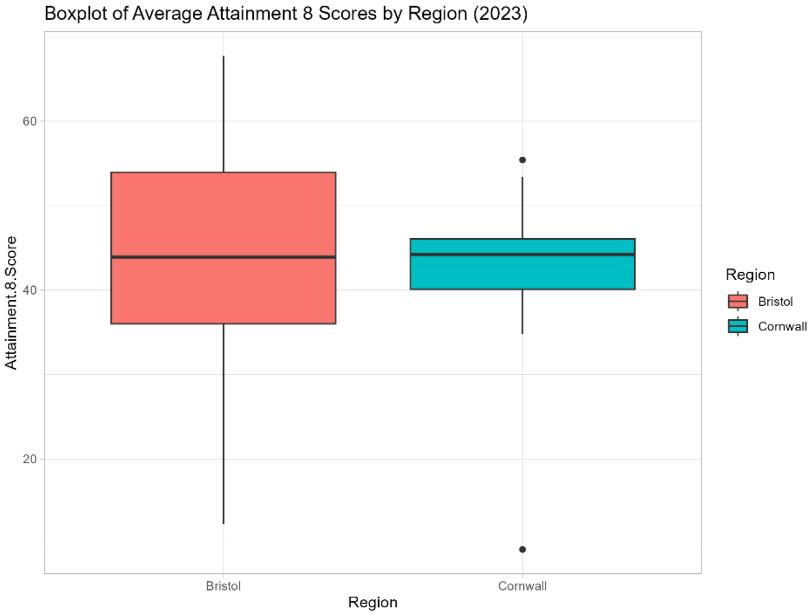


Figure 18: Boxplot of Average Attainment 8 Scores (2023)

The boxplot shows that Bristol's Average Attainment 8 Scores have greater variability and a wider range, indicating diverse performance. Cornwall's scores are more consistent, though there's a notable outlier with lower scores. Bristol's median is slightly higher, reflecting diverse outcomes, while Cornwall's schools show more uniform results.

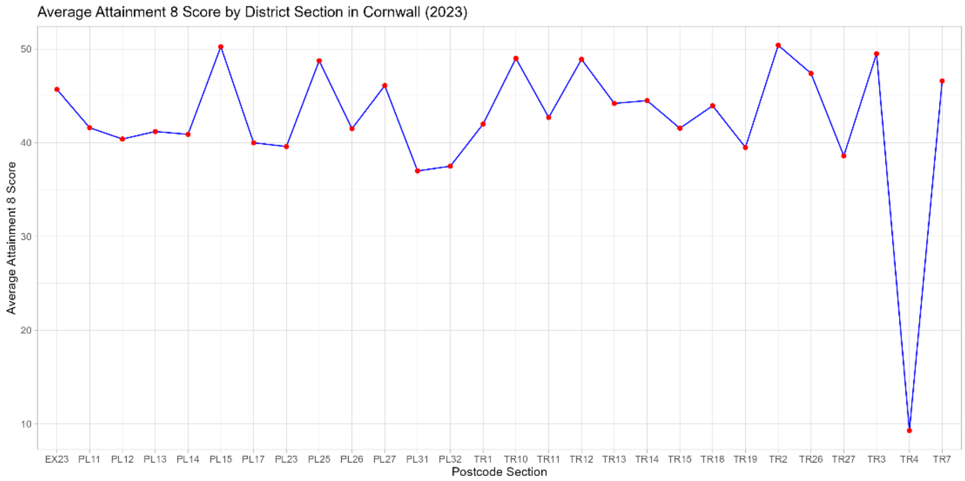
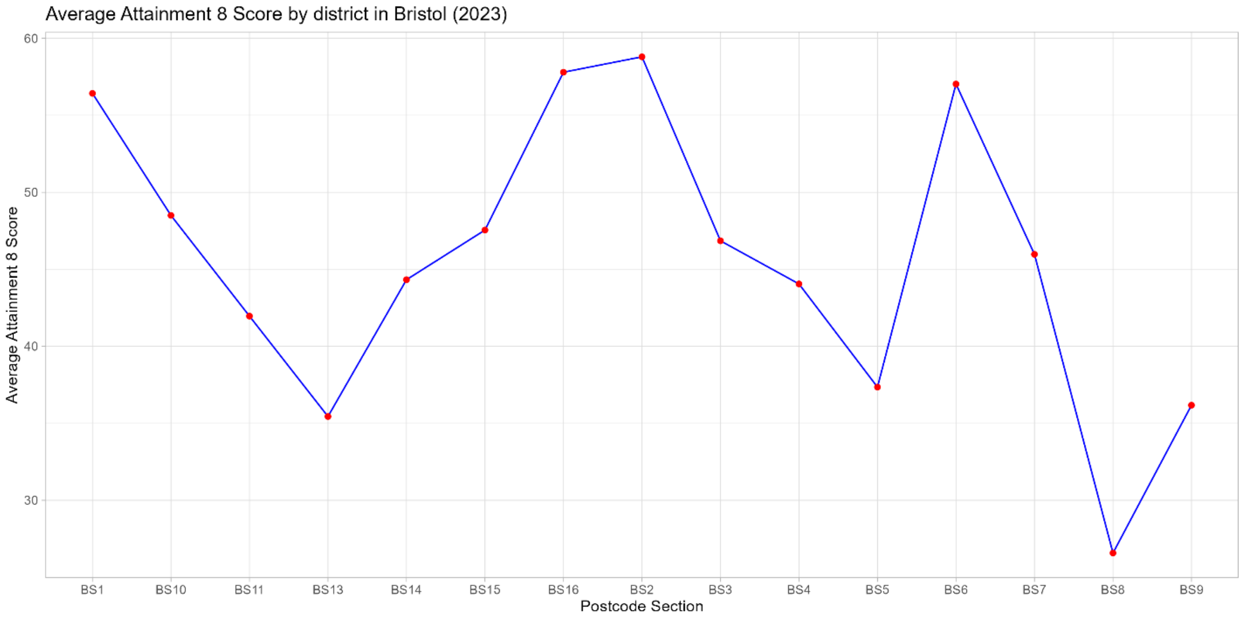


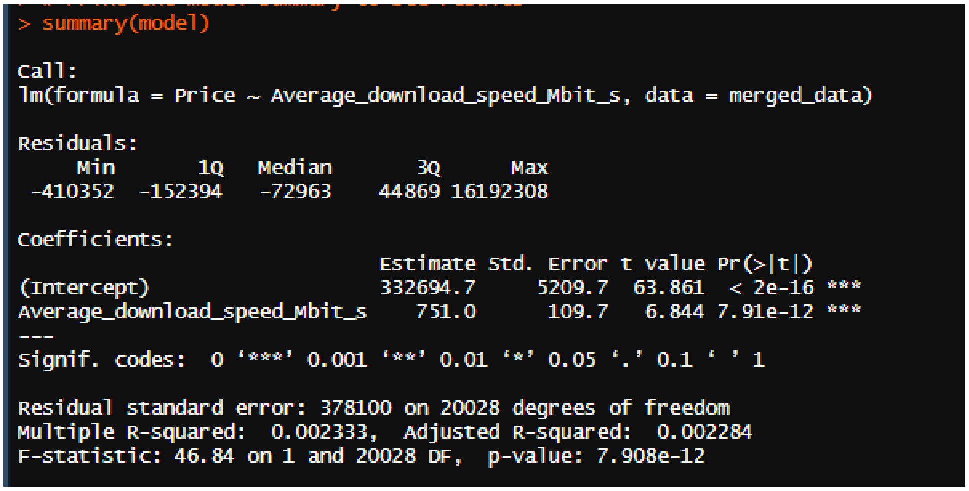
Figure 19: Line Graph for Attainment 8 score by district Cornwall(2023)

The 2023 line graph for Cornwall shows significant variability in Average Attainment 8 Scores. PL12, PL25, and TR1 score above 50, while TR4 sees sharp declines. PL11 and TR12 show mid-range consistency, highlighting a mix of high and low performers and the need to explore localized factors behind these disparities.



The graph reveals significant variability in Average Attainment 8 Scores across Bristol districts in 2023, with BS1 and BS16 scoring around 60, while BS13 and BS7 drop to around 30, highlighting disparities in educational performance.

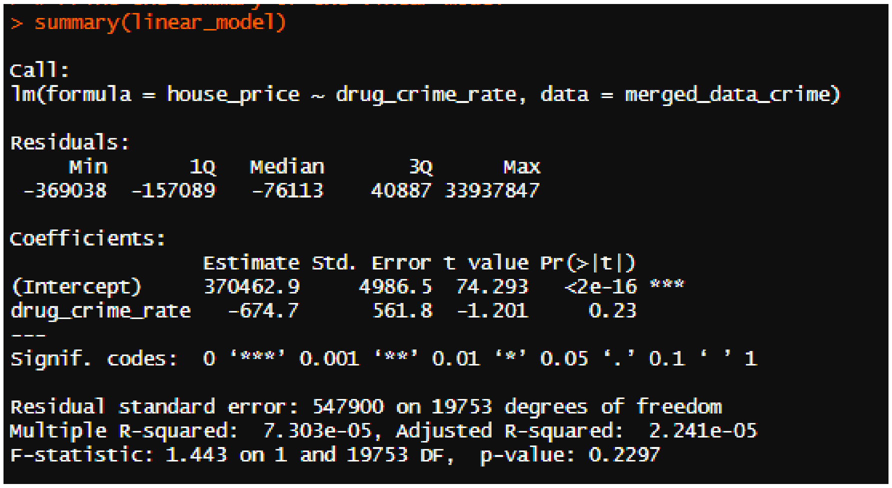
Linear modeling



```
> summary(model)

Call:
lm(formula = Price ~ Average_download_speed_Mbit_s, data = merged_data)

Residuals:
     Min       1Q   Median       3Q      Max 
 -410352  -152394   -72963    44869 16192308 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                   332694.7     5209.7  63.861  < 2e-16 ***
Average_download_speed_Mbit_s    751.0      109.7   6.844 7.91e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 378100 on 20028 degrees of freedom
Multiple R-squared:  0.002333,	Adjusted R-squared:  0.002284 
F-statistic: 46.84 on 1 and 20028 DF,  p-value: 7.908e-12
```

Figure 20: Price vs Average_download_speed

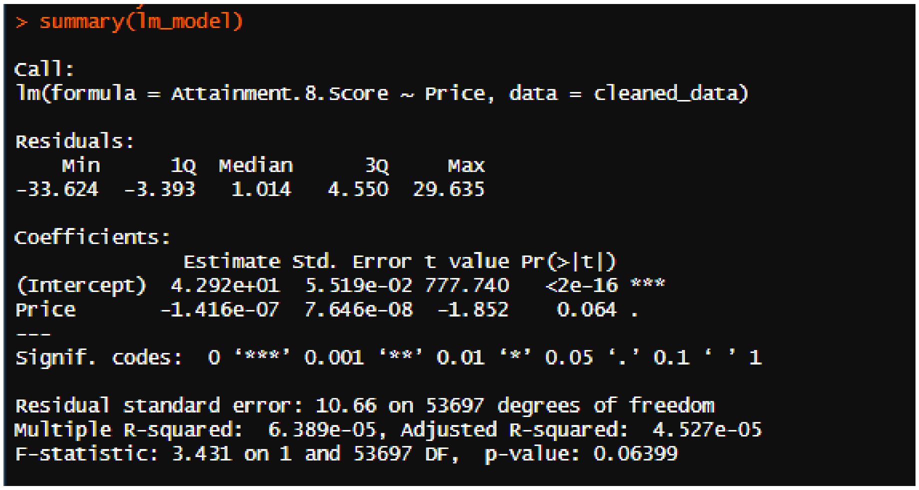
This linear regression shows a positive, statistically significant relationship between Price and Average_download_speed, with prices rising by 751 units for each unit increase in download speed. However, the low R-squared value of 0.0023 indicates that download speed explains only a small fraction of price variability. Despite a significant F-statistic (46.84) and p-value (7.91e-12), large residual ranges suggest other factors are likely influencing the price.



```
> summary(linear_model)

Call:
lm(formula = house_price ~ drug_crime_rate, data = merged_data_crime)

Residuals:
     Min       1Q   Median       3Q      Max 
 -369038  -157089   -76113    40887 33937847 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     370462.9     4986.5  74.293   <2e-16 ***
drug_crime_rate   -674.7      561.8  -1.201     0.23    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 547900 on 19753 degrees of freedom
Multiple R-squared:  7.303e-05,	Adjusted R-squared:  2.241e-05 
F-statistic: 1.443 on 1 and 19753 DF,  p-value: 0.2297
```

Figure 21: house price vs drug crime rate

In this linear regression model, the house price and drug_crime_rate intercept significantly (p < 2e-16), providing a baseline price when the crime rate is zero. The coefficient for drug_crime_rate is -674.7, indicating a decrease in house price with higher crime rates, but this effect is not statistically significant (p = 0.23). The very low R-squared value (7.303e-05) and an F-statistic of 1.443 (p = 0.2297) show that the model does not significantly improve prediction accuracy. Overall, the drug crime rate has minimal effect on house prices, suggesting the need to explore other factors.

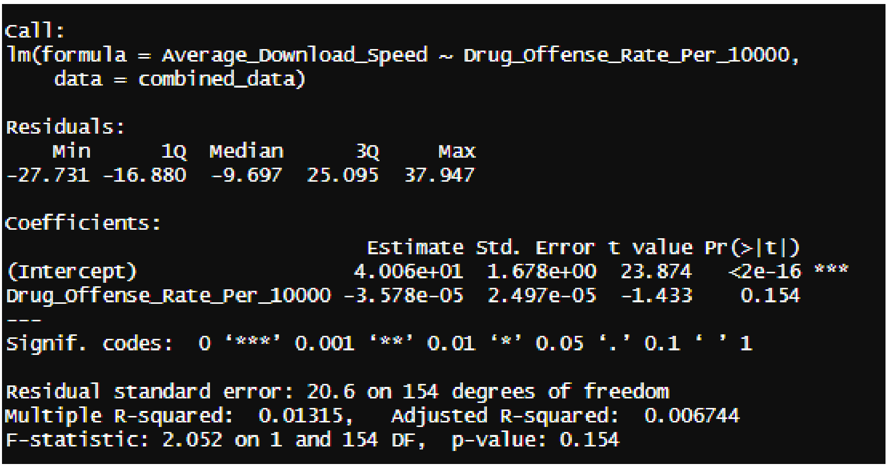


```
> summary(lm_model)

Call:
lm(formula = Attainment.8.Score ~ Price, data = cleaned_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-33.624  -3.393   1.014   4.550  29.635 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.292e+01  5.519e-02 777.740   <2e-16 ***
Price       -1.416e-07  7.646e-08  -1.852    0.064 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.66 on 53697 degrees of freedom
Multiple R-squared:  6.389e-05,	Adjusted R-squared:  4.527e-05 
F-statistic: 3.431 on 1 and 53697 DF,  p-value: 0.06399
```

Figure 22: Attainment 8 score vs price

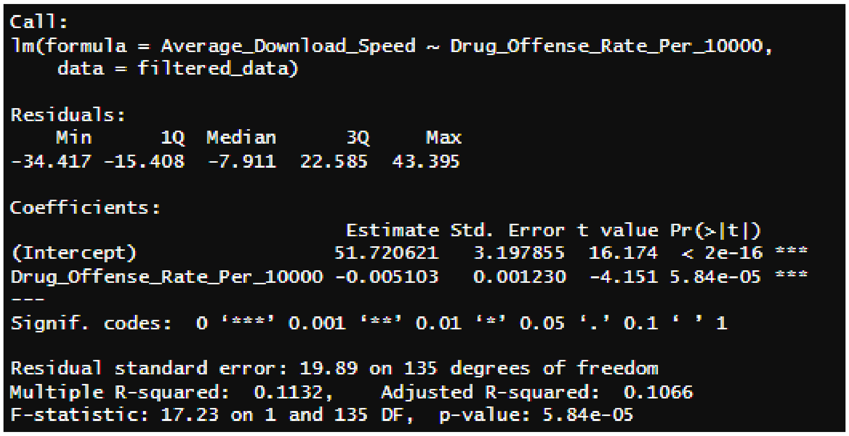
The model testing the impact of Price on Attainment.8.Score shows a baseline score of 42.92 when Price is zero (p < 2e-16). The coefficient for Price is -1.416e-07, suggesting a slight decrease in the score with higher Price, though the p-value of 0.064 is just above the 0.05 threshold, indicating a marginal relationship that could become significant with further study. The very low R-squared value (6.389e-05) means Price explains very little of the variation in

Attainment.8.Score. Despite an F-statistic of 3.431, the small p-value suggests other factors likely play a larger role



```
Call:
lm(formula = Average_Download_Speed ~ Drug_Offense_Rate_Per_10000, 
    data = combined_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-27.731 -16.880  -9.697  25.095  37.947 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  4.006e+01  1.678e+00  23.874   <2e-16 ***
Drug_Offense_Rate_Per_10000 -3.578e-05  2.497e-05  -1.433    0.154    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.6 on 154 degrees of freedom
Multiple R-squared:  0.01315,	Adjusted R-squared:  0.006744 
F-statistic: 2.052 on 1 and 154 DF,  p-value: 0.154
```

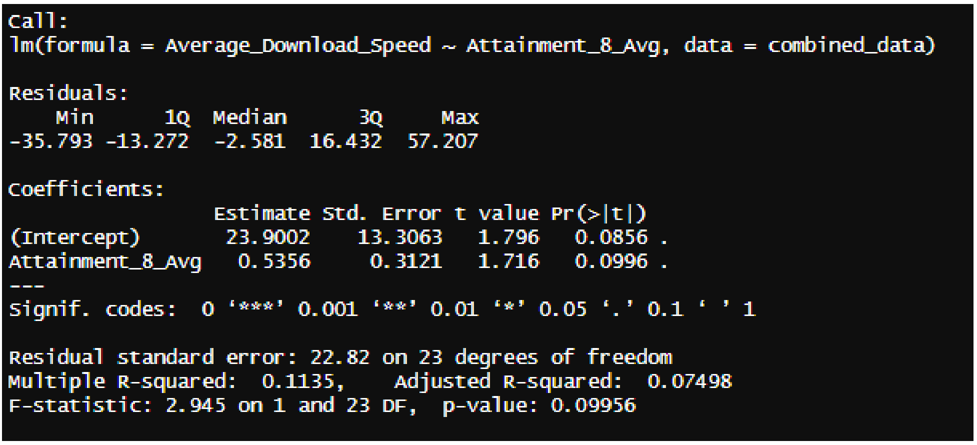
Figure 23: Average download speed vs drug offense rate per 1000



```
Call:
lm(formula = Average_Download_Speed ~ Drug_Offense_Rate_Per_10000, 
    data = filtered_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-34.417 -15.408  -7.911  22.585  43.395 

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 51.720621   3.197855  16.174  < 2e-16 ***
Drug_Offense_Rate_Per_10000 -0.005103   0.001230  -4.151 5.84e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.89 on 135 degrees of freedom
Multiple R-squared:  0.1132,	Adjusted R-squared:  0.1066 
F-statistic: 17.23 on 1 and 135 DF,  p-value: 5.84e-05
```

The linear regression analysis examines the effect of drug offence rates on download speed with and without outliers. Including outliers results in a negligible coefficient of -0.00003578 (p = 0.154) and a low R-squared of 0.01315, explaining only 1.3% of the variance. Excluding outliers improves the model significantly, with a coefficient of -0.005103 (p < 0.001) and an R-squared of 0.1132, explaining 11.3% of the variance. This demonstrates that outliers distorted the initial weak relationship, highlighting the importance of managing outliers in regression analysis

.



```
Call:
lm(formula = Average_Download_Speed ~ Attainment_8_Avg, data = combined_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-35.793 -13.272  -2.581  16.432  57.207 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)  
(Intercept)       23.9002    13.3063   1.796   0.0856 .
Attainment_8_Avg   0.5356     0.3121   1.716   0.0996 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22.82 on 23 degrees of freedom
Multiple R-squared:  0.1135,	Adjusted R-squared:  0.07498 
F-statistic: 2.945 on 1 and 23 DF,  p-value: 0.09956
```

Figure 24: Average download speed vs attainment 8

The linear regression analysis of average Attainment 8 scores and average download speed shows residuals ranging from -35.793 to 57.207 Mbps, with a median of -2.581 Mbps. The intercept is 23.9002 (p = 0.0856), and the coefficient for Attainment_8_Avg is 0.5356 (p = 0.0996), indicating a weak positive association. The model explains only 11.35% of the variance in download speed (R-squared = 0.1135) and has a high residual standard error of 22.82, suggesting that additional factors are needed for a more accurate model.

Recommendation system

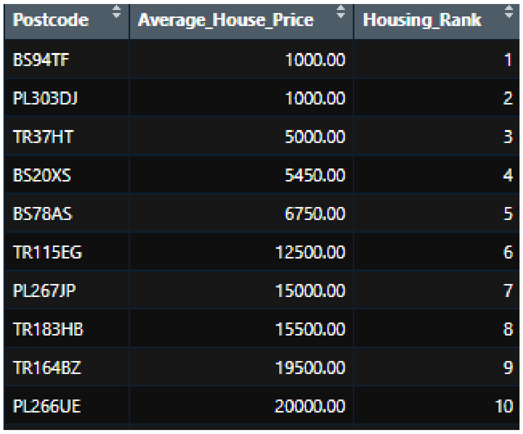
General overview

The ranking process is the bottom line of a clear picture about how different areas fare on various key factors. First, information is garnered from diverse sources: crime reports, broadband speeds, housing prices, school performance. Each of these factors is then transformed into some kind of simple metric, for instance; total number crimes on average broadband speeds typical house price or average school score.

These measures make up individual rankings for each area within each category. Following that, these rankings are pooled together.. This results in an overall ranking by averaging across all ranking categories hence giving a balanced view of regional performance with respect to areas where they outperformed others and those that need improvement. So to this end it would be well to go about it holistically since it allows for comparative views against which such factors can be measured overall.

Results

House price

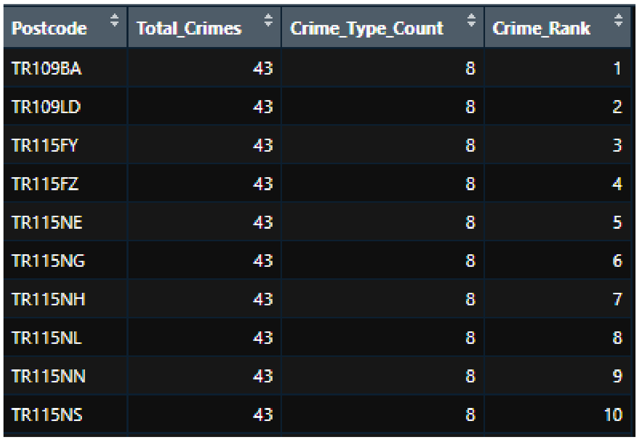


Postcode	Average_House_Price	Housing_Rank
BS94TF	1000.00	1
PL303DJ	1000.00	2
TR37HT	5000.00	3
BS20XS	5450.00	4
BS78AS	6750.00	5
TR115EG	12500.00	6
PL267JP	15000.00	7
TR183HB	15500.00	8
TR164BZ	19500.00	9
PL266UE	20000.00	10

Figure 25: House ranking

Using the average house price in a particular postcode area, the area was ranked with more affordable the house price, higher the rank.

Crime

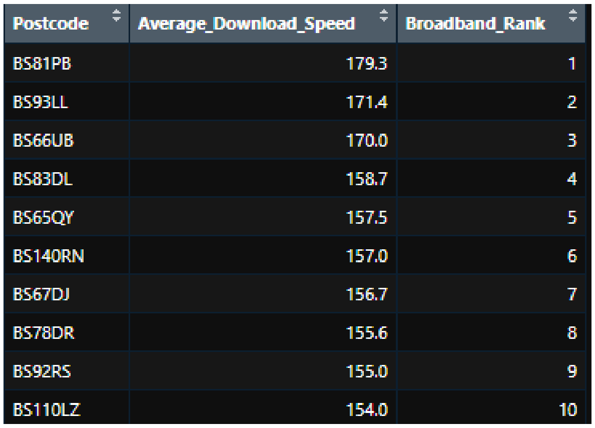


Postcode	Total_Crimes	Crime_Type_Count	Crime_Rank
TR109BA	43	8	1
TR109LD	43	8	2
TR115FY	43	8	3
TR115FZ	43	8	4
TR115NE	43	8	5
TR115NG	43	8	6
TR115NH	43	8	7
TR115NL	43	8	8
TR115NN	43	8	9
TR115NS	43	8	10

Figure 26: Crime ranking

Using the total crimes and total crime type count, the area postcodes were ranked with less the amount of crimes and crime types more the ranking.

Broadband

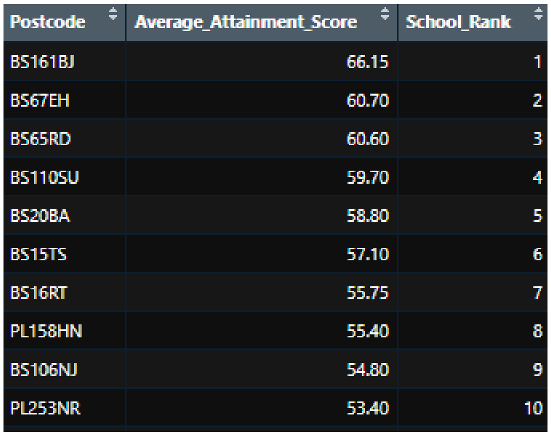


Postcode	Average_Download_Speed	Broadband_Rank
BS81PB	179.3	1
BS93LL	171.4	2
BS66UB	170.0	3
BS83DL	158.7	4
BS65QY	157.5	5
BS140RN	157.0	6
BS67DJ	156.7	7
BS78DR	155.6	8
BS92RS	155.0	9
BS110LZ	154.0	10

Figure 27: Broadband ranking

Using the average download speed, the area postcodes were ranked on the basis of higher the speed, higher the rank.

School



Postcode	Average_Attainment_Score	School_Rank
BS161BJ	66.15	1
BS67EH	60.70	2
BS65RD	60.60	3
BS110SU	59.70	4
BS20BA	58.80	5
BS15TS	57.10	6
BS16RT	55.75	7
PL158HN	55.40	8
BS106NJ	54.80	9
PL253NR	53.40	10

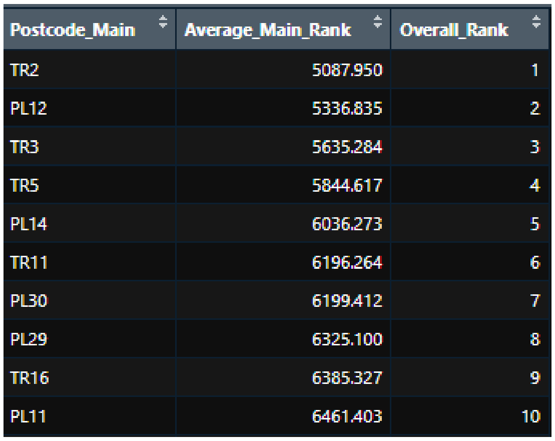
Figure 28: School ranking

Using the average attainment 8 score, the area postcodes were ranked based on higher the score higher the rank.

Reflection on the results

With the ranking based on certain characteristics, the observation can be made that if one makes affordable house price as an important factor then, even though Bristol has the top position, it only shows three times on the top 10 rank, making Cornwall a better option. Now, for crime the lowest rates Cornwall can be seen taking all the top 10 ranks. Making it safer than Bristol. Bristol on the other hand, dominates the top 10 for the highest average download speed. Considering the Attainment 8 score, Cornwall only has two candidates in the top 10 on the 8th and 9th positions. So, it is seen that, each character gives different places more importance when ranking. Making preference a valuable factor while making any kind of recommendation.

Overall score



Postcode_Main	Average_Main_Rank	Overall_Rank
TR2	5087.950	1
PL12	5336.835	2
TR3	5635.284	3
TR5	5844.617	4
PL14	6036.273	5
TR11	6196.264	6
PL30	6199.412	7
PL29	6325.100	8
TR16	6385.327	9
PL11	6461.403	10

Figure 29: Overall ranking

For a uniform wellbeing and all characteristics-based ranking**, I recommend Tregony** from Truro, Cornwall.

The top 3 postcodes are TR2, PL12 and TR3. Thus, making (Tregony, Truro, Cornwall) the top choice, (Liskeard, Plymouth, Cornwall) the second and (Falmouth/Penryn, Truro, Cornwall) the third.

As for the overall score, the received ranks of all character were taken and average for taken out. For the selection of a larger area the postcode was maintained accordingly. Then grouped and averaged again for overall rank. On the list, all the spots were occupied by Cornwall. Making Cornwall automatically preferable.

Legals and ethical issues

The current task was conducted in line with the highest standards of law and ethics. It was ensured that datasets used were sourced from open public UK government datasets in terms of open data licenses, not infringing on rights to intellectual property. Since this data is subject to the GDPR, it was treated under the tenet of confidentiality and non-disclosure. No identifiable person or any other sensitive information has been used for analysis.

On the ethical side, it was aimed at safeguarding the Bristol and Cornwall community residents from any negative impact. Therefore, the recommendation was carried out carefully to eliminate biases and make sure that similar evaluative standards for each city were applied to prevent gentrification or displacement of individuals. Where making them morally right was going to render them legally correct, this effort focused on transparency, the integrity of information, and social responsibility to have its findings provide results that are considered fair and acceptable to all involved stakeholders.

Conclusion

The project developed a recommendation for cities based on essentialities such as house pricing, broadband speed, crime rates and another attribute that is supposed to be relevant. The project employed datasets from data.gov.uk, which were subjected to a thorough cleaning exercise and normalization in order to ensure accuracy and consistency of the results obtained.

Exploratory Data Analysis (EDA) uncovered significant associations between attributes that helped with the design of a scoring system. Each attribute was turned into ranks and their scores summed up across features so as to rank towns. It gave the top three towns through their cumulative scores which can be used as an instrument to rate towns. The whole project has been well documented, showing how the methodologies were done justifying why these particular attributes were included. As such, it is an operational recommendation algorithm that offers useful insights in making decisions on residential options/ investments.

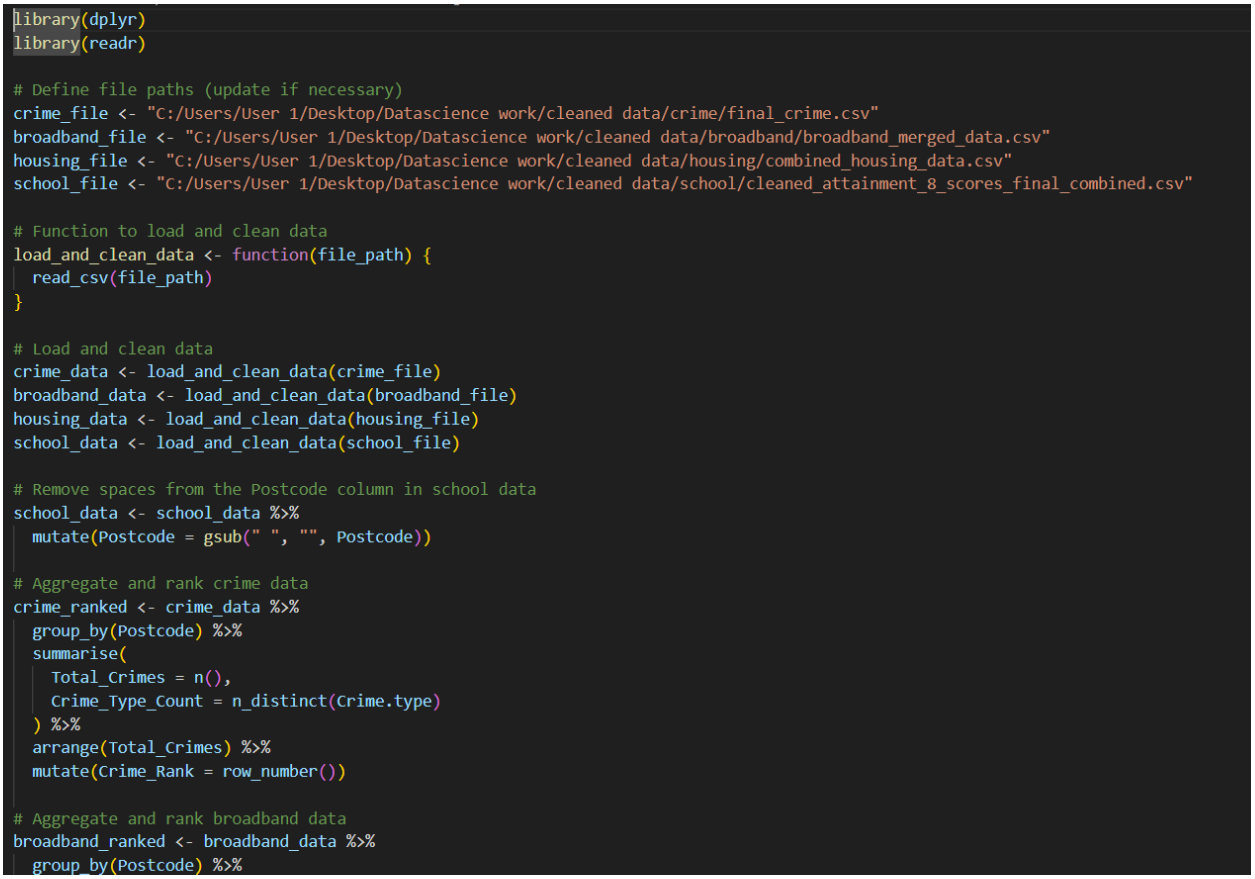
References

1. HM Land Registry. (n.d.). Price paid data downloads (2020-2023). HM Land Registry. https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads
2. Ofcom. (2018). Connected nations: Data downloads (Broadband speed). Ofcom. https://www.ofcom.org.uk/research-and-data/multi-sector-research/infrastructure-research/connected-nations-2018/data-downloads
3. UK Home Office. (n.d.). Police force areas: Data download (Crime dataset). UK Home Office. https://data.police.uk/data/
4. [Classroom reference]. (n.d.). Population dataset (2011). [Unpublished dataset]. Provided by CR via Microsoft Teams.
5. Department for Education. (2019). School performance data downloads (2018-2019). Department for Education. https://www.compare-school-performance.service.gov.uk/download-data?currentstep=datatypes&regiontype=all&la=0&downloadYear=2018-2019&datatypes=ks5
6. Provost, F., & Fawcett, T. (2013). Data science for business: What you need to know about data mining and data-analytic thinking. O'Reilly Media.
7. Tukey, J. W. (1977). Exploratory data analysis. Addison-Wesley.
8. McKinney, W. (2017). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.
9. Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches (5th ed.). SAGE Publications.

10. RGraph. (n.d.). RGraph: Free, open-source JavaScript charts. RGraph. https://www.rgraph.net/

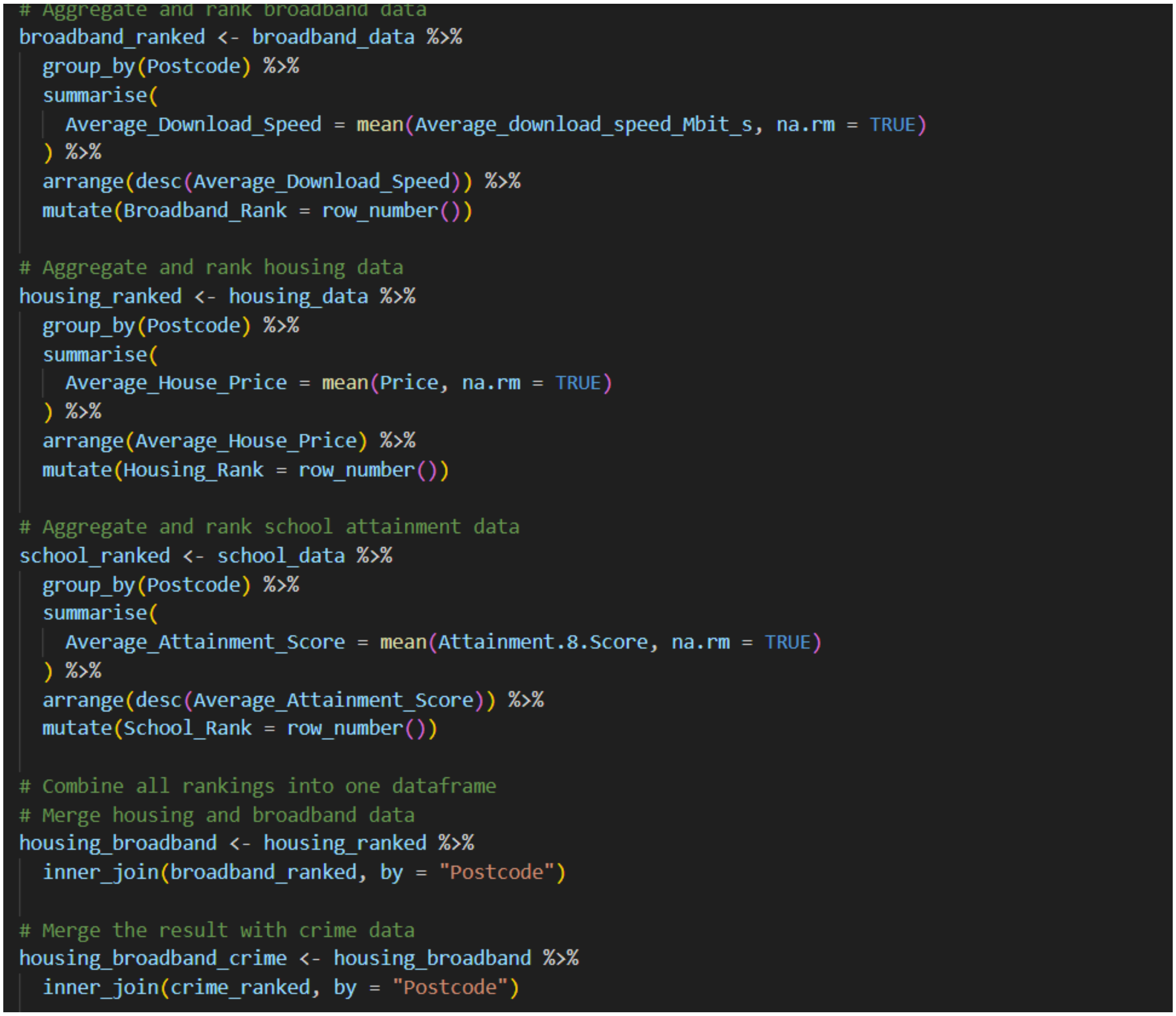
11. Setti, M. (2020, May 27). The analysis lifecycle. Towards Data Science. https://towardsdatascience.com/the-analysis-lifecycle-448e6b36931c

Appendix



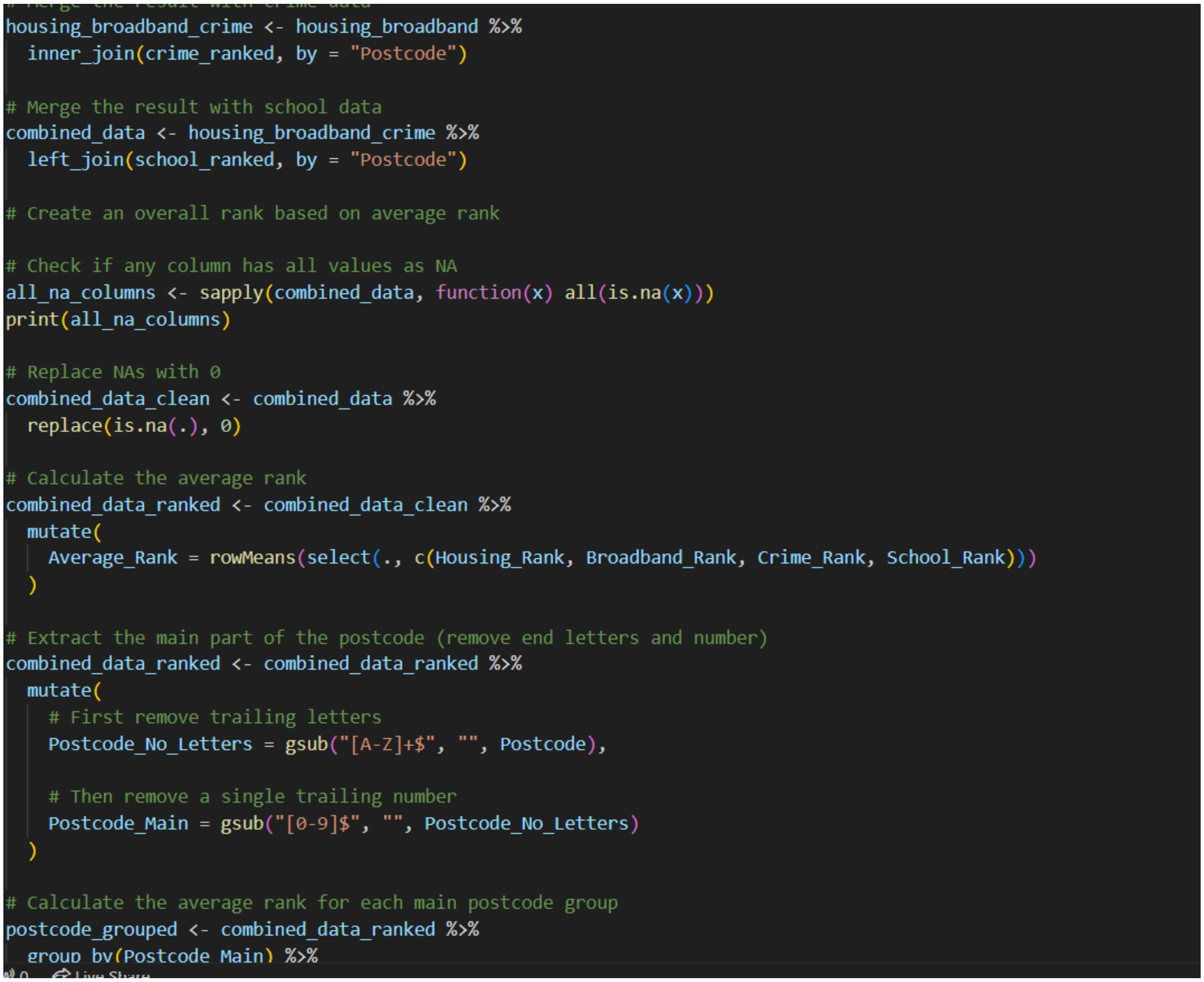
```
library(dplyr)
library(readr)

# Define file paths (update if necessary)
crime_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/crime/final_crime.csv"
broadband_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/broadband/broadband_merged_data.csv"
housing_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/housing/combined_housing_data.csv"
school_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/school/cleaned_attainment_8_scores_final_combined.csv"

# Function to load and clean data
load_and_clean_data <- function(file_path) {
  read_csv(file_path)
}

# Load and clean data
crime_data <- load_and_clean_data(crime_file)
broadband_data <- load_and_clean_data(broadband_file)
housing_data <- load_and_clean_data(housing_file)
school_data <- load_and_clean_data(school_file)

# Remove spaces from the Postcode column in school data
school_data <- school_data %>%
  mutate(Postcode = gsub(" ", "", Postcode))

# Aggregate and rank crime data
crime_ranked <- crime_data %>%
  group_by(Postcode) %>%
  summarise(
    Total_Crimes = n(),
    Crime_Type_Count = n_distinct(Crime.type)
  ) %>%
  arrange(Total_Crimes) %>%
  mutate(Crime_Rank = row_number())

# Aggregate and rank broadband data
broadband_ranked <- broadband_data %>%
  group_by(Postcode) %>%
```

Figure 30: Ranking code



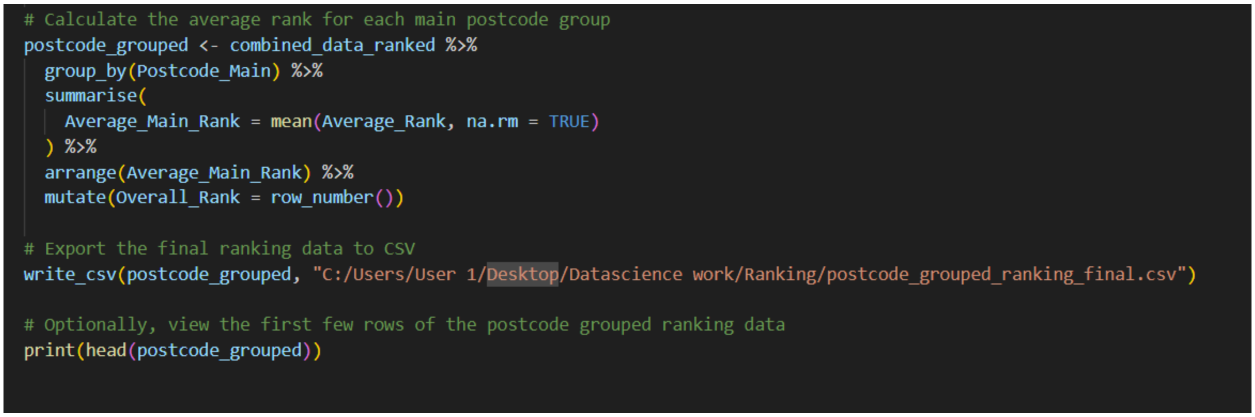
```
# Aggregate and rank broadband data
broadband_ranked <- broadband_data %>%
  group_by(Postcode) %>%
  summarise(
    Average_Download_Speed = mean(Average_download_speed_Mbit_s, na.rm = TRUE)
  ) %>%
  arrange(desc(Average_Download_Speed)) %>%
  mutate(Broadband_Rank = row_number())

# Aggregate and rank housing data
housing_ranked <- housing_data %>%
  group_by(Postcode) %>%
  summarise(
    Average_House_Price = mean(Price, na.rm = TRUE)
  ) %>%
  arrange(Average_House_Price) %>%
  mutate(Housing_Rank = row_number())

# Aggregate and rank school attainment data
school_ranked <- school_data %>%
  group_by(Postcode) %>%
  summarise(
    Average_Attainment_Score = mean(Attainment.8.Score, na.rm = TRUE)
  ) %>%
  arrange(desc(Average_Attainment_Score)) %>%
  mutate(School_Rank = row_number())

# Combine all rankings into one dataframe
# Merge housing and broadband data
housing_broadband <- housing_ranked %>%
  inner_join(broadband_ranked, by = "Postcode")

# Merge the result with crime data
housing_broadband_crime <- housing_broadband %>%
  inner_join(crime_ranked, by = "Postcode")
```

Figure 31: Ranking code



```
# Merge the result with crime data
housing_broadband_crime <- housing_broadband %>%
  inner_join(crime_ranked, by = "Postcode")

# Merge the result with school data
combined_data <- housing_broadband_crime %>%
  left_join(school_ranked, by = "Postcode")

# Create an overall rank based on average rank

# Check if any column has all values as NA
all_na_columns <- sapply(combined_data, function(x) all(is.na(x)))
print(all_na_columns)

# Replace NAs with 0
combined_data_clean <- combined_data %>%
  replace(is.na(.), 0)

# Calculate the average rank
combined_data_ranked <- combined_data_clean %>%
  mutate(
    Average_Rank = rowMeans(select(., c(Housing_Rank, Broadband_Rank, Crime_Rank, School_Rank)))
  )

# Extract the main part of the postcode (remove end letters and number)
combined_data_ranked <- combined_data_ranked %>%
  mutate(
    # First remove trailing letters
    Postcode_No_Letters = gsub("[A-Z]+$", "", Postcode),

    # Then remove a single trailing number
    Postcode_Main = gsub("[0-9]$", "", Postcode_No_Letters)
  )

# Calculate the average rank for each main postcode group
postcode_grouped <- combined_data_ranked %>%
  group_by(Postcode_Main) %>%
```

Figure 32: Ranking Code



```
# Calculate the average rank for each main postcode group
postcode_grouped <- combined_data_ranked %>%
  group_by(Postcode_Main) %>%
  summarise(
    Average_Main_Rank = mean(Average_Rank, na.rm = TRUE)
  ) %>%
  arrange(Average_Main_Rank) %>%
  mutate(Overall_Rank = row_number())

# Export the final ranking data to CSV
write_csv(postcode_grouped, "C:/Users/User 1/Desktop/Datascience work/Ranking/postcode_grouped_ranking_final.csv")

# Optionally, view the first few rows of the postcode grouped ranking data
print(head(postcode_grouped))
```

Figure 33: Ranking Code

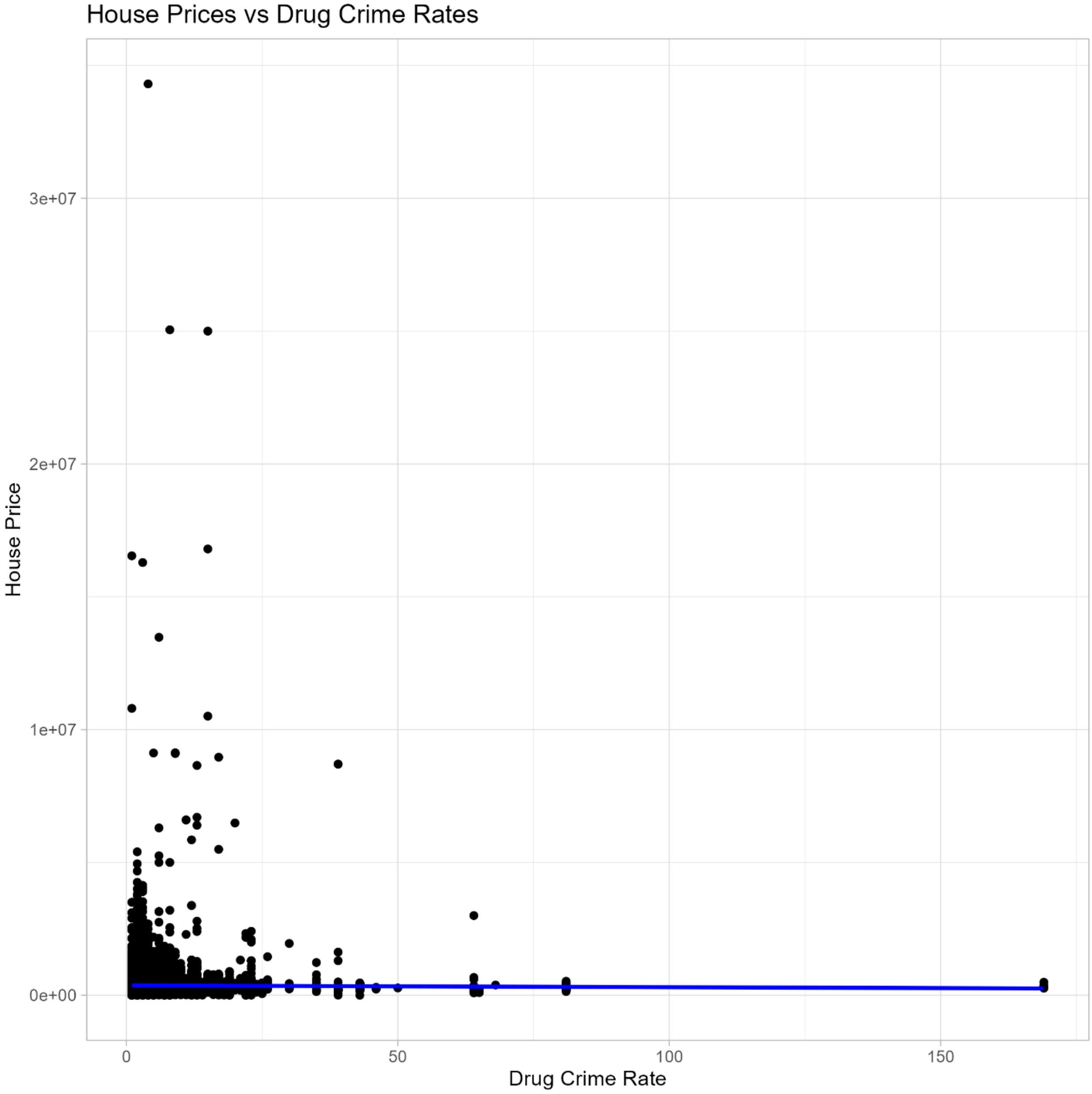


Figure 34: House price vs Drug crime rate scatterplot

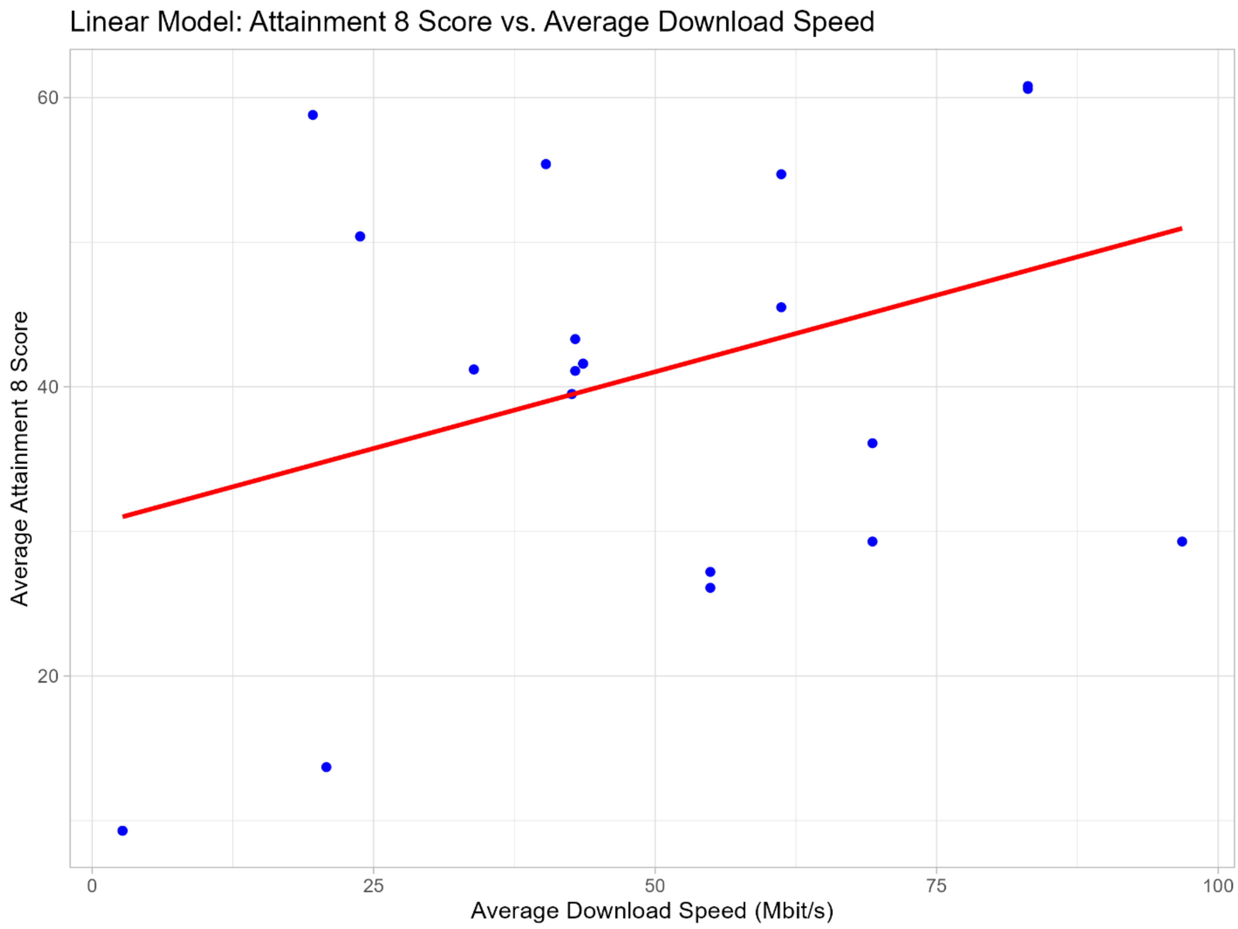


Figure 35: Average download speed vs average attainment 8 score scatterplot

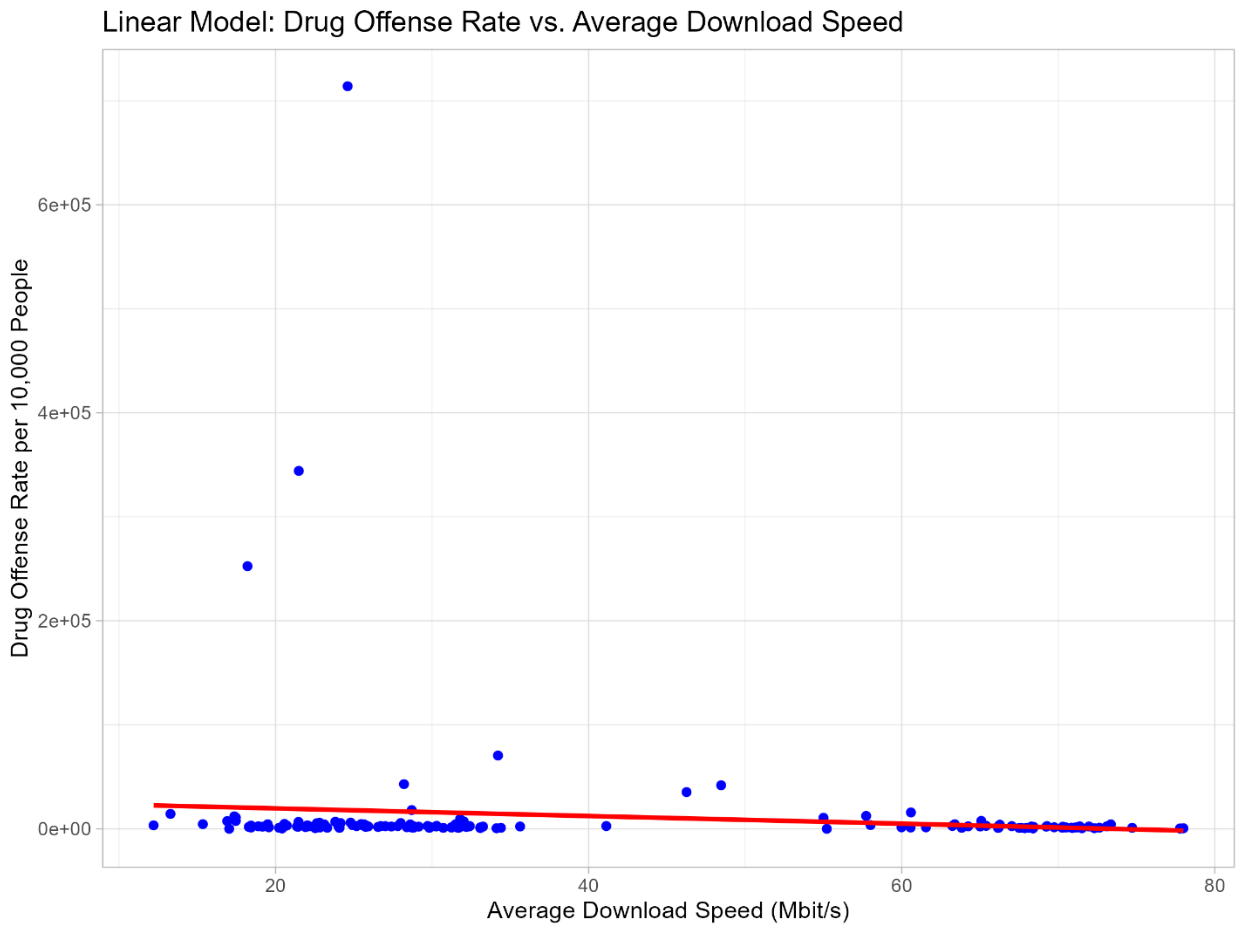


Figure 36: Average Download Speed vs Drug offense scatterplot

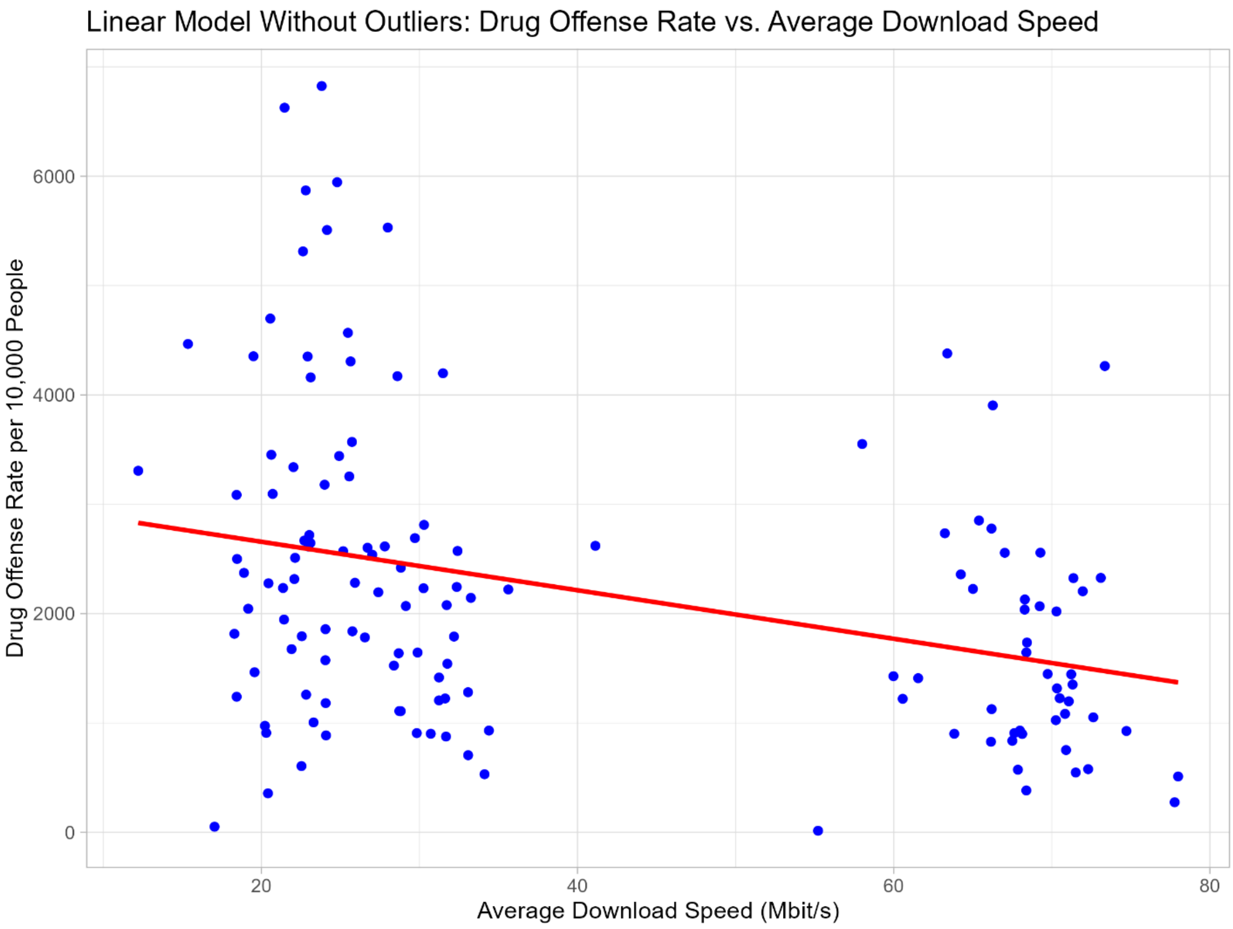


Figure 37: Average Download Speed vs Drug offense rate per 10000 people Scatterplot

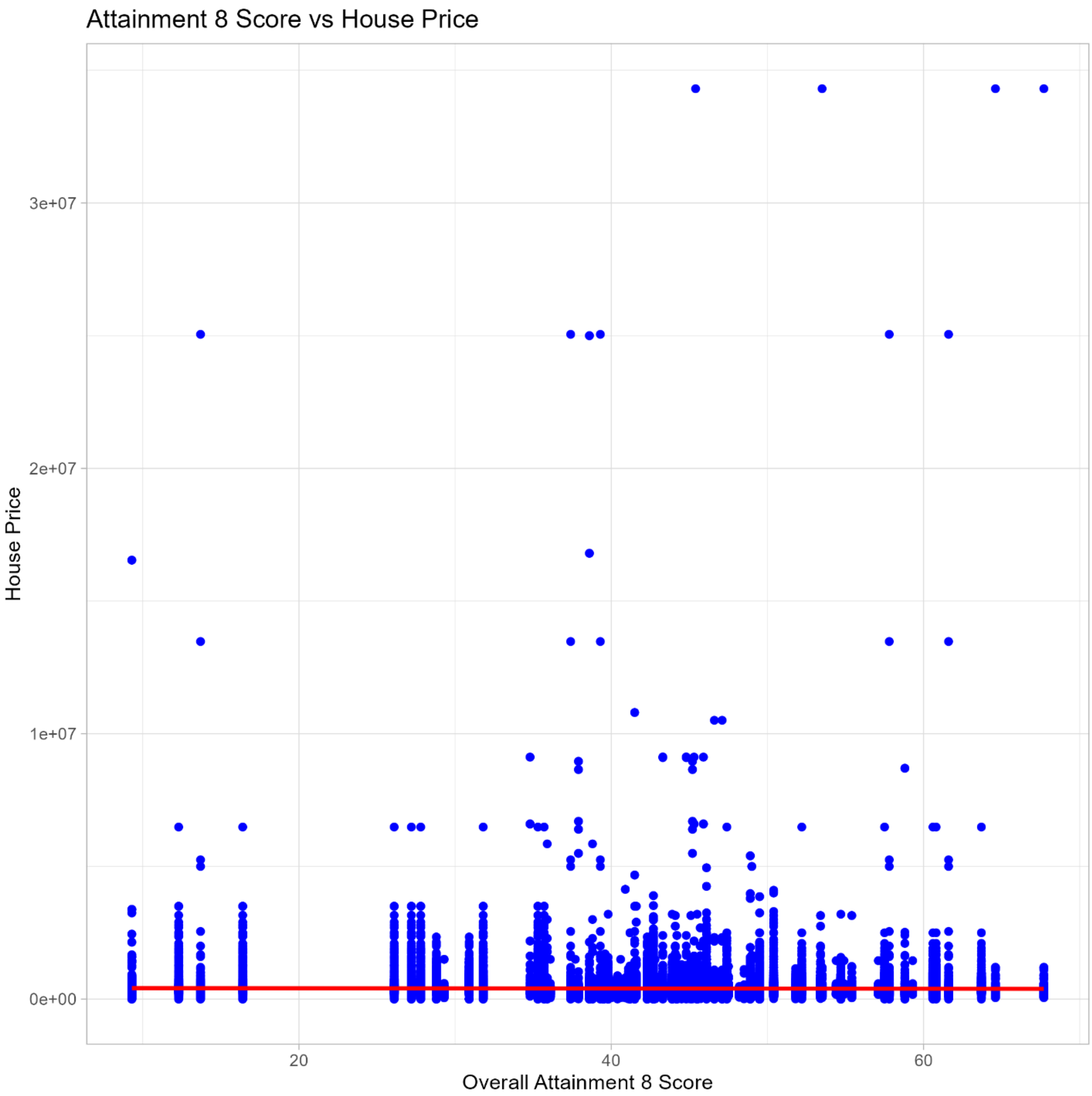


Figure 38: Attainment 8 vs house price scatterplot

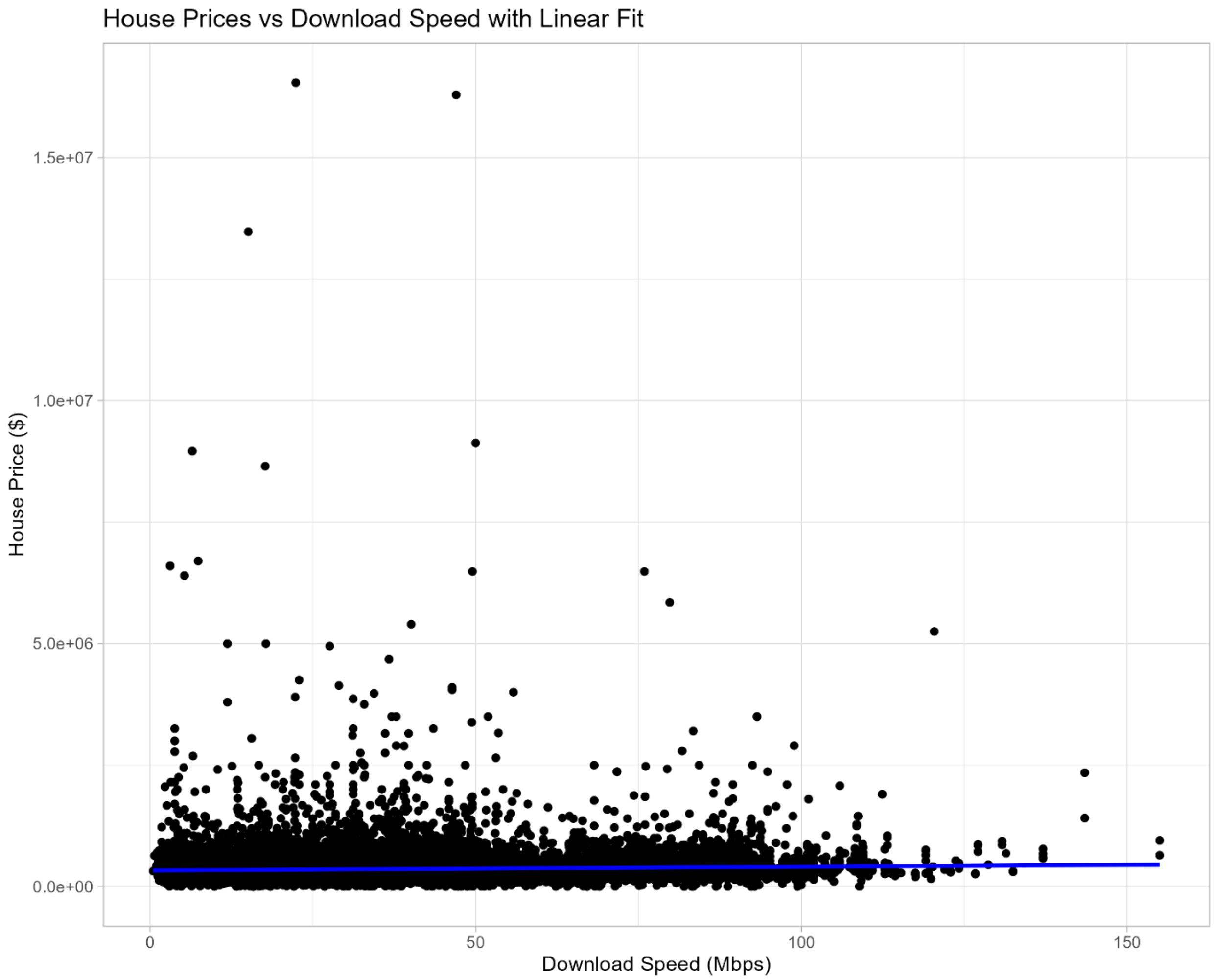
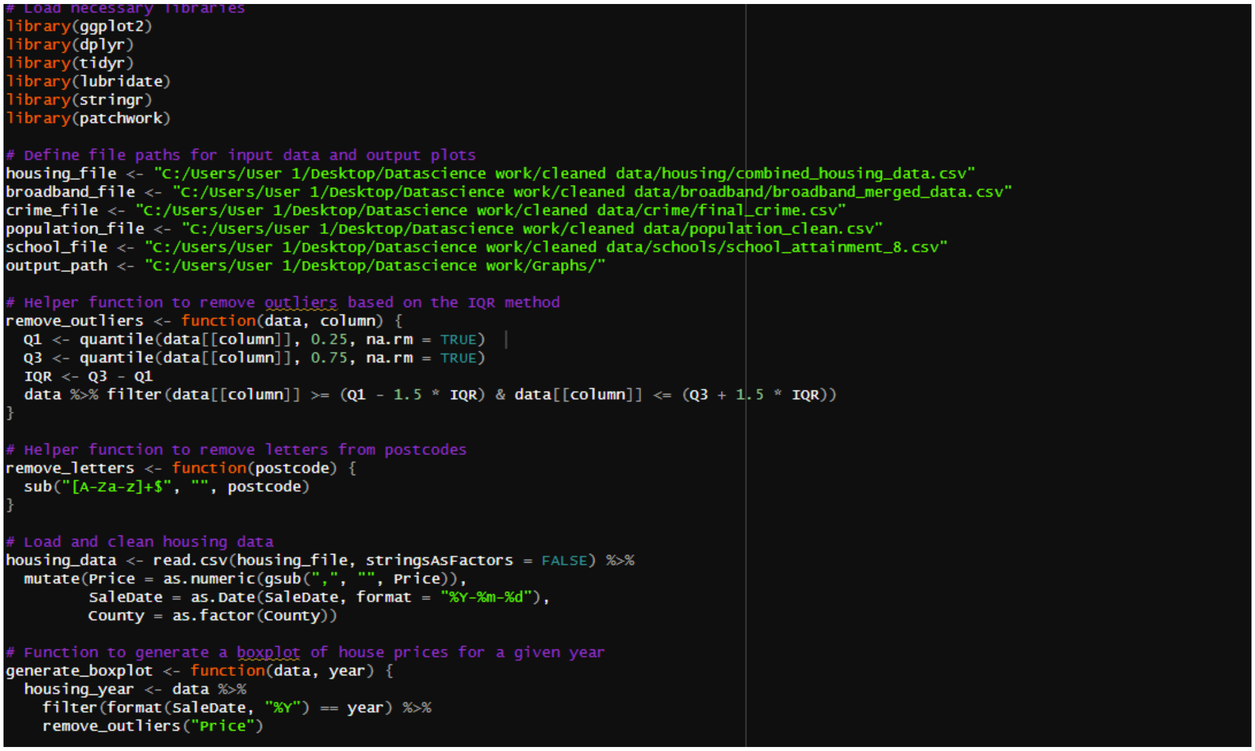
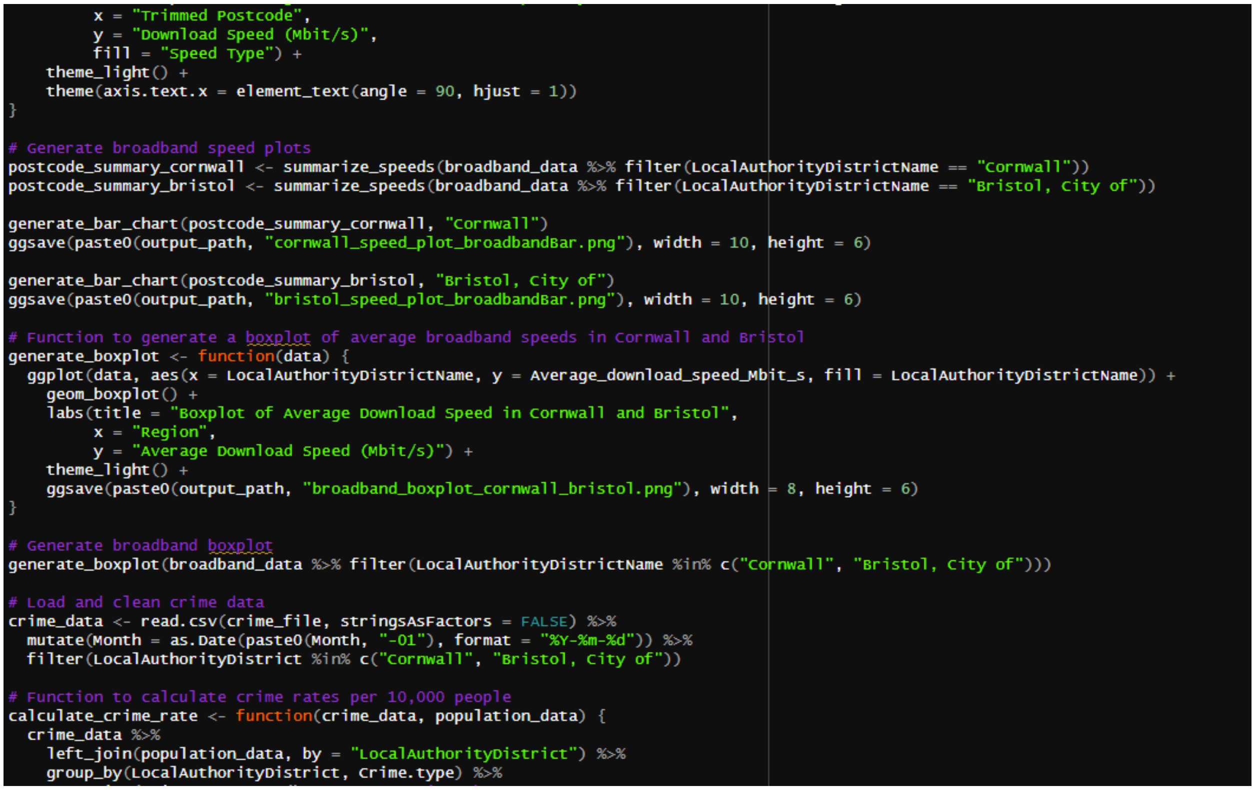


Figure 39: Download speed vs House price Scatterplot

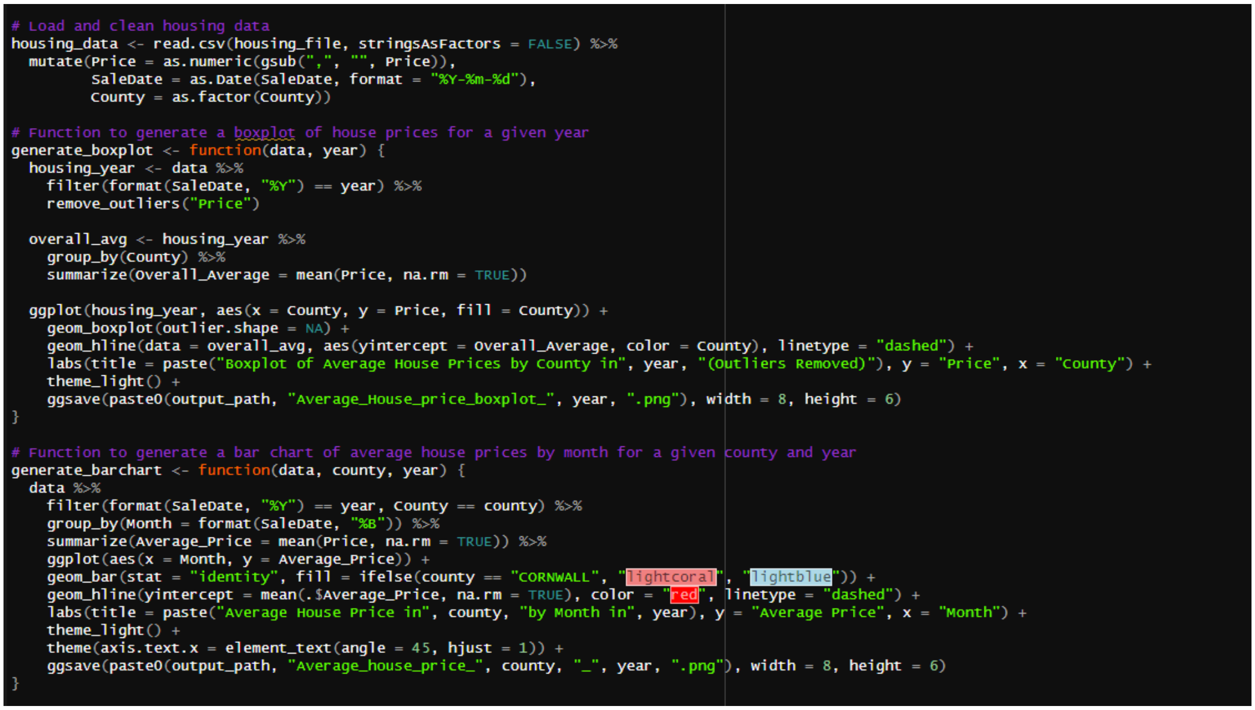
EDA Code



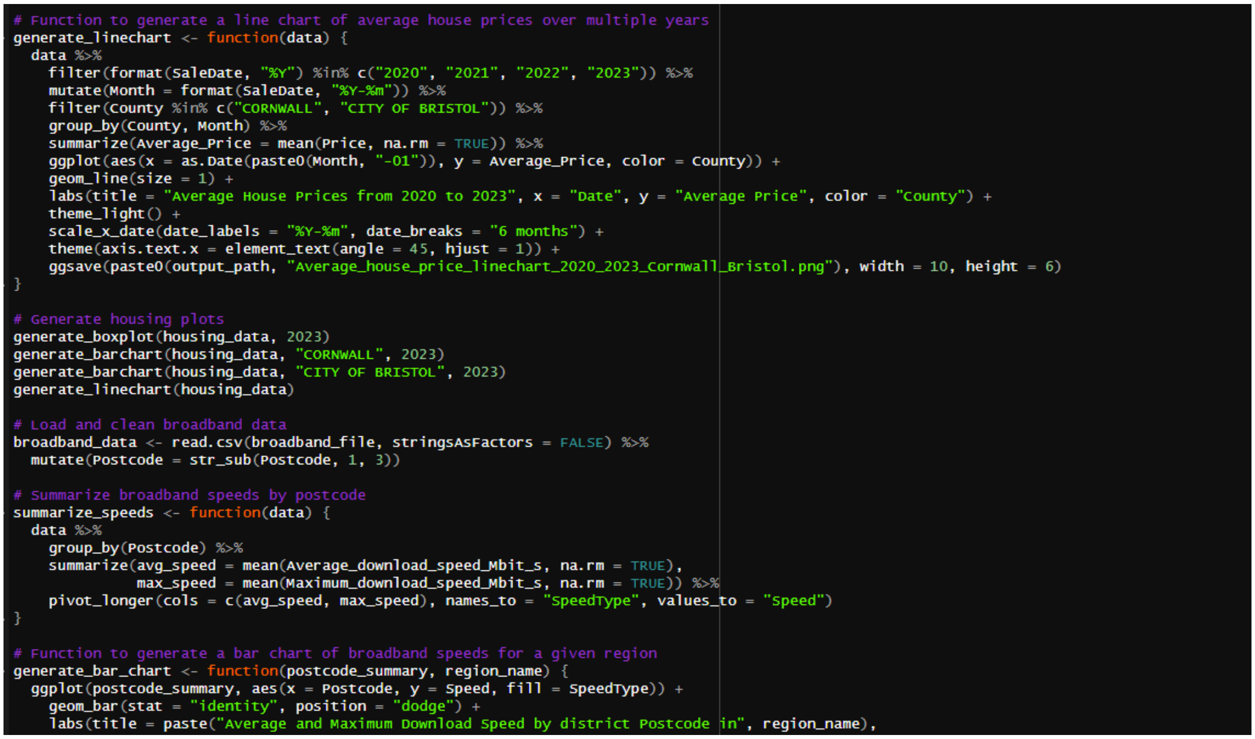
```
# Load necessary libraries
library(ggplot2)
library(dplyr)
library(tidyr)
library(lubridate)
library(stringr)
library(patchwork)

# Define file paths for input data and output plots
housing_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/housing/combined_housing_data.csv"
broadband_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/broadband/broadband_merged_data.csv"
crime_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/crime/final_crime.csv"
population_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/population_clean.csv"
school_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/schools/school_attainment_8.csv"
output_path <- "C:/Users/User 1/Desktop/Datascience work/Graphs/"

# Helper function to remove outliers based on the IQR method
remove_outliers <- function(data, column) {
  Q1 <- quantile(data[[column]], 0.25, na.rm = TRUE)
  Q3 <- quantile(data[[column]], 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  data %>% filter(data[[column]] >= (Q1 - 1.5 * IQR) & data[[column]] <= (Q3 + 1.5 * IQR))
}

# Helper function to remove letters from postcodes
remove_letters <- function(postcode) {
  sub("[A-Za-z]+$", "", postcode)
}

# Load and clean housing data
housing_data <- read.csv(housing_file, stringsAsFactors = FALSE) %>%
  mutate(Price = as.numeric(gsub(",", "", Price)),
         SaleDate = as.Date(SaleDate, format = "%Y-%m-%d"),
         County = as.factor(County))

# Function to generate a boxplot of house prices for a given year
generate_boxplot <- function(data, year) {
  housing_year <- data %>%
    filter(format(SaleDate, "%Y") == year) %>%
    remove_outliers("Price")
```



```
         x = "Trimmed Postcode",
         y = "Download Speed (Mbit/s)",
         fill = "Speed Type") +
    theme_light() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
}

# Generate broadband speed plots
postcode_summary_cornwall <- summarize_speeds(broadband_data %>% filter(LocalAuthorityDistrictName == "Cornwall"))
postcode_summary_bristol <- summarize_speeds(broadband_data %>% filter(LocalAuthorityDistrictName == "Bristol, City of"))

generate_bar_chart(postcode_summary_cornwall, "Cornwall")
ggsave(paste0(output_path, "cornwall_speed_plot_broadbandBar.png"), width = 10, height = 6)

generate_bar_chart(postcode_summary_bristol, "Bristol, City of")
ggsave(paste0(output_path, "bristol_speed_plot_broadbandBar.png"), width = 10, height = 6)

# Function to generate a boxplot of average broadband speeds in Cornwall and Bristol
generate_boxplot <- function(data) {
  ggplot(data, aes(x = LocalAuthorityDistrictName, y = Average_download_speed_Mbit_s, fill = LocalAuthorityDistrictName)) +
    geom_boxplot() +
    labs(title = "Boxplot of Average Download Speed in Cornwall and Bristol",
         x = "Region",
         y = "Average Download Speed (Mbit/s)") +
    theme_light() +
    ggsave(paste0(output_path, "broadband_boxplot_cornwall_bristol.png"), width = 8, height = 6)
}

# Generate broadband boxplot
generate_boxplot(broadband_data %>% filter(LocalAuthorityDistrictName %in% c("Cornwall", "Bristol, City of")))

# Load and clean crime data
crime_data <- read.csv(crime_file, stringsAsFactors = FALSE) %>%
  mutate(Month = as.Date(paste0(Month, "-01"), format = "%Y-%m-%d")) %>%
  filter(LocalAuthorityDistrict %in% c("Cornwall", "Bristol, City of"))

# Function to calculate crime rates per 10,000 people
calculate_crime_rate <- function(crime_data, population_data) {
  crime_data %>%
    left_join(population_data, by = "LocalAuthorityDistrict") %>%
    group_by(LocalAuthorityDistrict, Crime.type) %>%
```

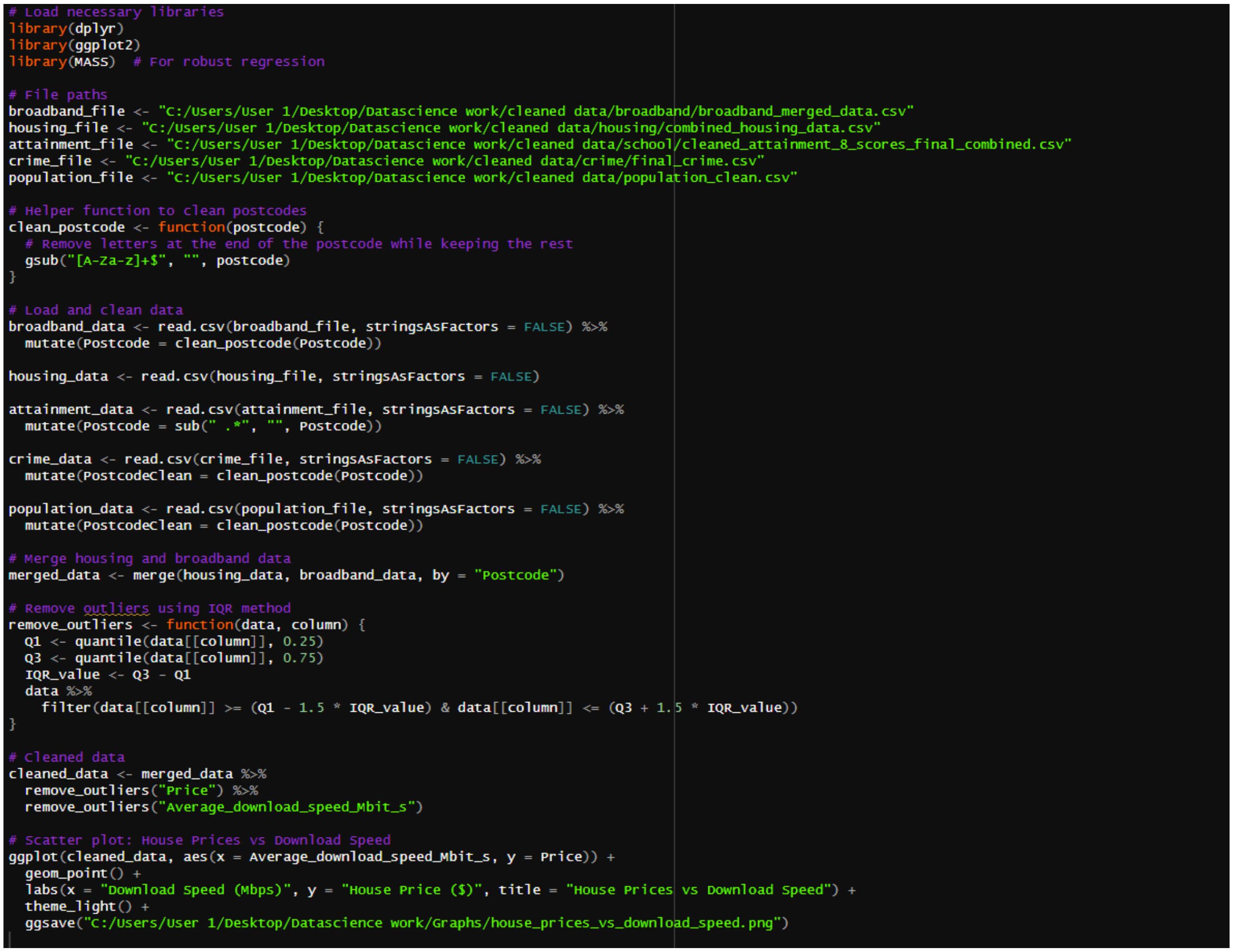


```r
# Load and clean housing data
housing_data <- read.csv(housing_file, stringsAsFactors = FALSE) %>%
  mutate(Price = as.numeric(gsub(",", "", Price)),
         SaleDate = as.Date(SaleDate, format = "%Y-%m-%d"),
         County = as.factor(County))

# Function to generate a boxplot of house prices for a given year
generate_boxplot <- function(data, year) {
  housing_year <- data %>%
    filter(format(SaleDate, "%Y") == year) %>%
    remove_outliers("Price")

  overall_avg <- housing_year %>%
    group_by(County) %>%
    summarize(Overall_Average = mean(Price, na.rm = TRUE))

  ggplot(housing_year, aes(x = County, y = Price, fill = County)) +
    geom_boxplot(outlier.shape = NA) +
    geom_hline(data = overall_avg, aes(yintercept = Overall_Average, color = County), linetype = "dashed") +
    labs(title = paste("Boxplot of Average House Prices by County in", year, "(Outliers Removed)"), y = "Price", x = "County") +
    theme_light() +
    ggsave(paste0(output_path, "Average_House_price_boxplot_", year, ".png"), width = 8, height = 6)
}

# Function to generate a bar chart of average house prices by month for a given county and year
generate_barchart <- function(data, county, year) {
  data %>%
    filter(format(SaleDate, "%Y") == year, County == county) %>%
    group_by(Month = format(SaleDate, "%B")) %>%
    summarize(Average_Price = mean(Price, na.rm = TRUE)) %>%
    ggplot(aes(x = Month, y = Average_Price)) +
    geom_bar(stat = "identity", fill = ifelse(county == "CORNWALL", "lightcoral", "lightblue")) +
    geom_hline(yintercept = mean(.$Average_Price, na.rm = TRUE), color = "red", linetype = "dashed") +
    labs(title = paste("Average House Price in", county, "by Month in", year), y = "Average Price", x = "Month") +
    theme_light() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    ggsave(paste0(output_path, "Average_house_price_", county, "_", year, ".png"), width = 8, height = 6)
}
```

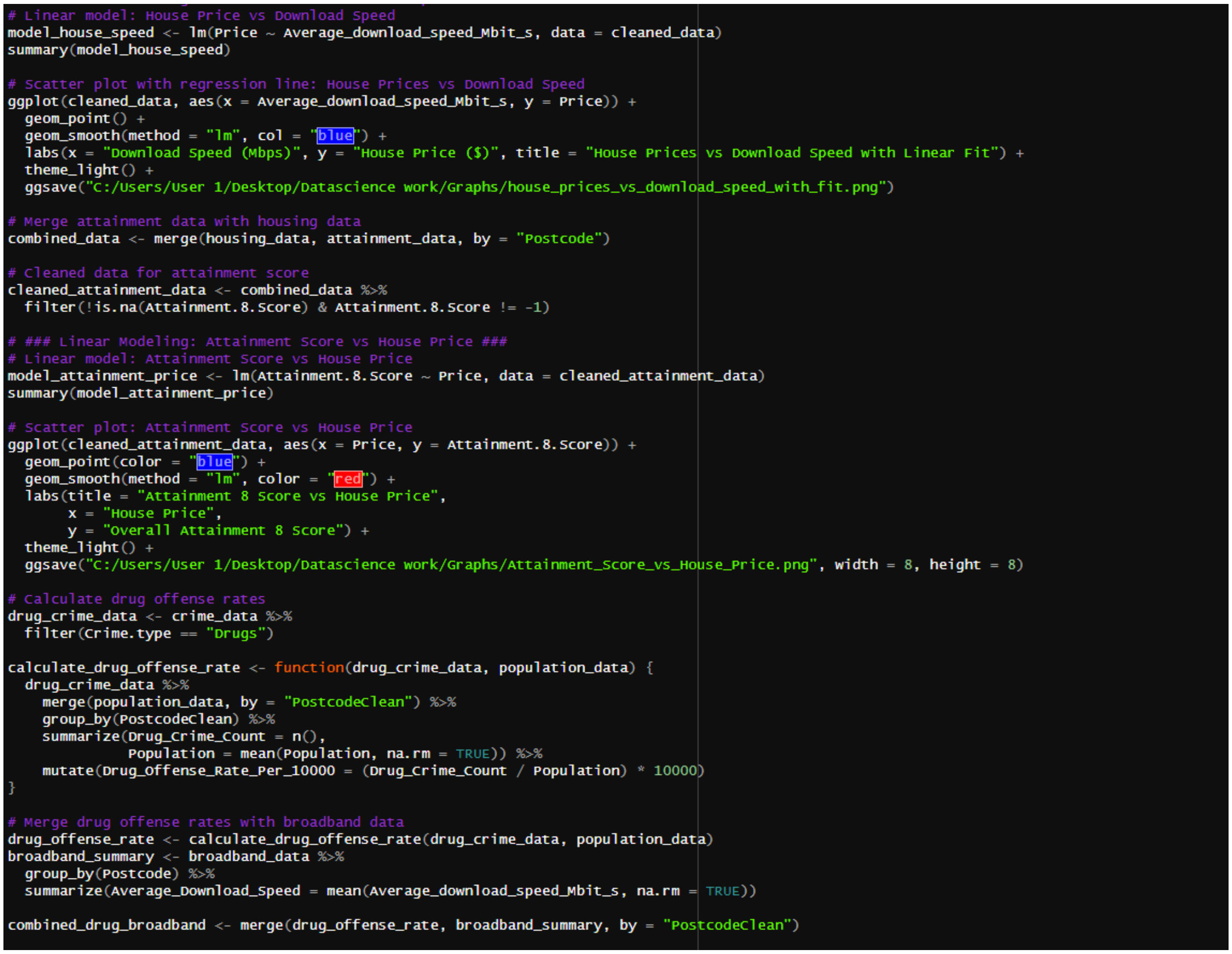


```r
# Function to generate a line chart of average house prices over multiple years
generate_linechart <- function(data) {
  data %>%
    filter(format(SaleDate, "%Y") %in% c("2020", "2021", "2022", "2023")) %>%
    mutate(Month = format(SaleDate, "%Y-%m")) %>%
    filter(County %in% c("CORNWALL", "CITY OF BRISTOL")) %>%
    group_by(County, Month) %>%
    summarize(Average_Price = mean(Price, na.rm = TRUE)) %>%
    ggplot(aes(x = as.Date(paste0(Month, "-01")), y = Average_Price, color = County)) +
    geom_line(size = 1) +
    labs(title = "Average House Prices from 2020 to 2023", x = "Date", y = "Average Price", color = "County") +
    theme_light() +
    scale_x_date(date_labels = "%Y-%m", date_breaks = "6 months") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    ggsave(paste0(output_path, "Average_house_price_linechart_2020_2023_Cornwall_Bristol.png"), width = 10, height = 6)
}

# Generate housing plots
generate_boxplot(housing_data, 2023)
generate_barchart(housing_data, "CORNWALL", 2023)
generate_barchart(housing_data, "CITY OF BRISTOL", 2023)
generate_linechart(housing_data)

# Load and clean broadband data
broadband_data <- read.csv(broadband_file, stringsAsFactors = FALSE) %>%
  mutate(Postcode = str_sub(Postcode, 1, 3))

# Summarize broadband speeds by postcode
summarize_speeds <- function(data) {
  data %>%
    group_by(Postcode) %>%
    summarize(avg_speed = mean(Average_download_speed_Mbit_s, na.rm = TRUE),
              max_speed = mean(Maximum_download_speed_Mbit_s, na.rm = TRUE)) %>%
    pivot_longer(cols = c(avg_speed, max_speed), names_to = "SpeedType", values_to = "Speed")
}

# Function to generate a bar chart of broadband speeds for a given region
generate_bar_chart <- function(postcode_summary, region_name) {
  ggplot(postcode_summary, aes(x = Postcode, y = Speed, fill = SpeedType)) +
    geom_bar(stat = "identity", position = "dodge") +
    labs(title = paste("Average and Maximum Download Speed by district Postcode in", region_name),
```

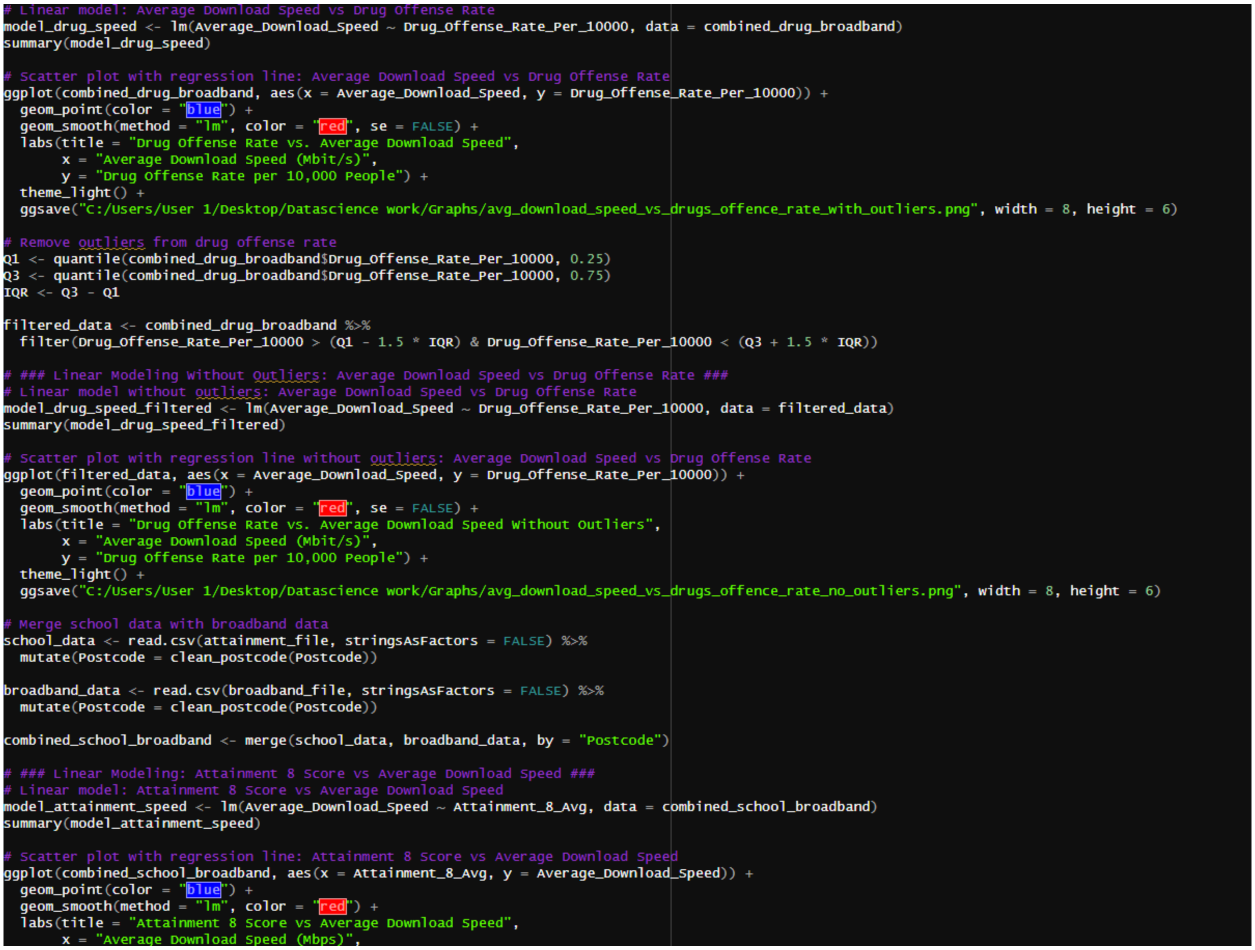
Linear Code



```r
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(MASS)  # For robust regression

# File paths
broadband_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/broadband/broadband_merged_data.csv"
housing_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/housing/combined_housing_data.csv"
attainment_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/school/cleaned_attainment_8_scores_final_combined.csv"
crime_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/crime/final_crime.csv"
population_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/population_clean.csv"

# Helper function to clean postcodes
clean_postcode <- function(postcode) {
  # Remove letters at the end of the postcode while keeping the rest
  gsub("[A-Za-z]+$", "", postcode)
}

# Load and clean data
broadband_data <- read.csv(broadband_file, stringsAsFactors = FALSE) %>%
  mutate(Postcode = clean_postcode(Postcode))

housing_data <- read.csv(housing_file, stringsAsFactors = FALSE)

attainment_data <- read.csv(attainment_file, stringsAsFactors = FALSE) %>%
  mutate(Postcode = sub(" .*", "", Postcode))

crime_data <- read.csv(crime_file, stringsAsFactors = FALSE) %>%
  mutate(PostcodeClean = clean_postcode(Postcode))

population_data <- read.csv(population_file, stringsAsFactors = FALSE) %>%
  mutate(PostcodeClean = clean_postcode(Postcode))

# Merge housing and broadband data
merged_data <- merge(housing_data, broadband_data, by = "Postcode")

# Remove outliers using IQR method
remove_outliers <- function(data, column) {
  Q1 <- quantile(data[[column]], 0.25)
  Q3 <- quantile(data[[column]], 0.75)
  IQR_value <- Q3 - Q1
  data %>%
    filter(data[[column]] >= (Q1 - 1.5 * IQR_value) & data[[column]] <= (Q3 + 1.5 * IQR_value))
}

# Cleaned data
cleaned_data <- merged_data %>%
  remove_outliers("Price") %>%
  remove_outliers("Average_download_speed_Mbit_s")

# Scatter plot: House Prices vs Download Speed
ggplot(cleaned_data, aes(x = Average_download_speed_Mbit_s, y = Price)) +
  geom_point() +
  labs(x = "Download Speed (Mbps)", y = "House Price ($)", title = "House Prices vs Download Speed") +
  theme_light() +
  ggsave("C:/Users/User 1/Desktop/Datascience work/Graphs/house_prices_vs_download_speed.png")
```

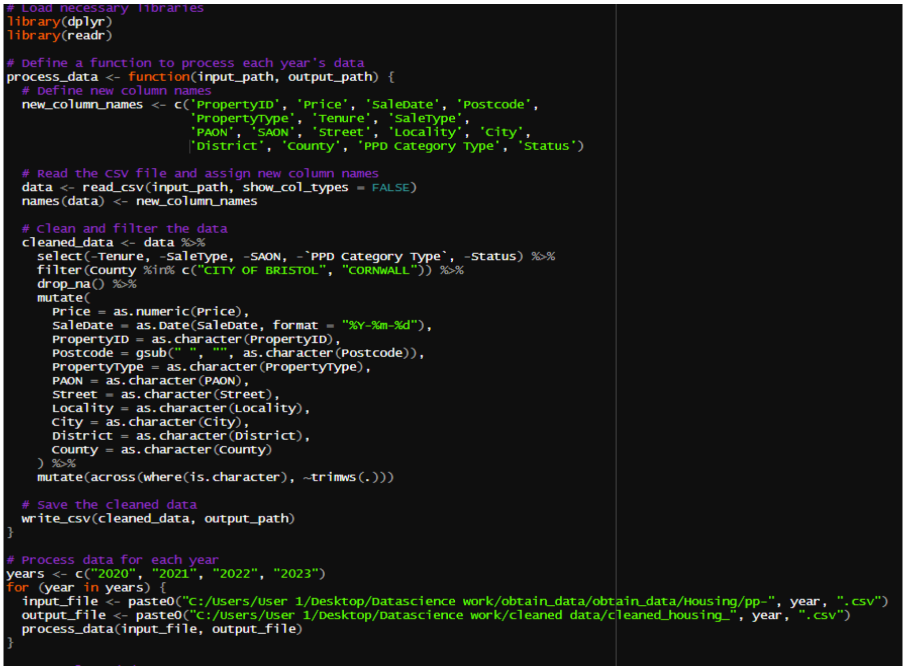


```r
# Linear model: House Price vs Download Speed
model_house_speed <- lm(Price ~ Average_download_speed_Mbit_s, data = cleaned_data)
summary(model_house_speed)

# Scatter plot with regression line: House Prices vs Download Speed
ggplot(cleaned_data, aes(x = Average_download_speed_Mbit_s, y = Price)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(x = "Download Speed (Mbps)", y = "House Price ($)", title = "House Prices vs Download Speed with Linear Fit") +
  theme_light() +
  ggsave("C:/Users/User 1/Desktop/Datascience work/Graphs/house_prices_vs_download_speed_with_fit.png")

# Merge attainment data with housing data
combined_data <- merge(housing_data, attainment_data, by = "Postcode")

# Cleaned data for attainment score
cleaned_attainment_data <- combined_data %>%
  filter(!is.na(Attainment.8.Score) & Attainment.8.Score != -1)

# ### Linear Modeling: Attainment Score vs House Price ###
# Linear model: Attainment Score vs House Price
model_attainment_price <- lm(Attainment.8.Score ~ Price, data = cleaned_attainment_data)
summary(model_attainment_price)

# Scatter plot: Attainment Score vs House Price
ggplot(cleaned_attainment_data, aes(x = Price, y = Attainment.8.Score)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Attainment 8 Score vs House Price",
       x = "House Price",
       y = "Overall Attainment 8 Score") +
  theme_light() +
  ggsave("C:/Users/User 1/Desktop/Datascience work/Graphs/Attainment_Score_vs_House_Price.png", width = 8, height = 8)

# Calculate drug offense rates
drug_crime_data <- crime_data %>%
  filter(Crime.type == "Drugs")

calculate_drug_offense_rate <- function(drug_crime_data, population_data) {
  drug_crime_data %>%
    merge(population_data, by = "PostcodeClean") %>%
    group_by(PostcodeClean) %>%
    summarize(Drug_Crime_Count = n(),
              Population = mean(Population, na.rm = TRUE)) %>%
    mutate(Drug_Offense_Rate_Per_10000 = (Drug_Crime_Count / Population) * 10000)
}

# Merge drug offense rates with broadband data
drug_offense_rate <- calculate_drug_offense_rate(drug_crime_data, population_data)
broadband_summary <- broadband_data %>%
  group_by(Postcode) %>%
  summarize(Average_Download_Speed = mean(Average_download_speed_Mbit_s, na.rm = TRUE))

combined_drug_broadband <- merge(drug_offense_rate, broadband_summary, by = "PostcodeClean")
```

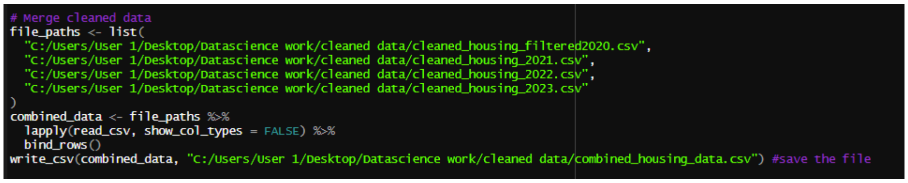


```
# Linear model: Average Download Speed vs Drug Offense Rate
model_drug_speed <- lm(Average_Download_Speed ~ Drug_Offense_Rate_Per_10000, data = combined_drug_broadband)
summary(model_drug_speed)

# Scatter plot with regression line: Average Download Speed vs Drug Offense Rate
ggplot(combined_drug_broadband, aes(x = Average_Download_Speed, y = Drug_Offense_Rate_Per_10000)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Drug Offense Rate vs. Average Download Speed",
       x = "Average Download Speed (Mbit/s)",
       y = "Drug Offense Rate per 10,000 People") +
  theme_light() +
  ggsave("C:/Users/User 1/Desktop/Datascience work/Graphs/avg_download_speed_vs_drugs_offence_rate_with_outliers.png", width = 8, height = 6)

# Remove outliers from drug offense rate
Q1 <- quantile(combined_drug_broadband$Drug_Offense_Rate_Per_10000, 0.25)
Q3 <- quantile(combined_drug_broadband$Drug_Offense_Rate_Per_10000, 0.75)
IQR <- Q3 - Q1

filtered_data <- combined_drug_broadband %>%
  filter(Drug_Offense_Rate_Per_10000 > (Q1 - 1.5 * IQR) & Drug_Offense_Rate_Per_10000 < (Q3 + 1.5 * IQR))

# ### Linear Modeling Without Outliers: Average Download Speed vs Drug Offense Rate ###
# Linear model without outliers: Average Download Speed vs Drug Offense Rate
model_drug_speed_filtered <- lm(Average_Download_Speed ~ Drug_Offense_Rate_Per_10000, data = filtered_data)
summary(model_drug_speed_filtered)

# Scatter plot with regression line without outliers: Average Download Speed vs Drug Offense Rate
ggplot(filtered_data, aes(x = Average_Download_Speed, y = Drug_Offense_Rate_Per_10000)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Drug Offense Rate vs. Average Download Speed Without Outliers",
       x = "Average Download Speed (Mbit/s)",
       y = "Drug Offense Rate per 10,000 People") +
  theme_light() +
  ggsave("C:/Users/User 1/Desktop/Datascience work/Graphs/avg_download_speed_vs_drugs_offence_rate_no_outliers.png", width = 8, height = 6)

# Merge school data with broadband data
school_data <- read.csv(attainment_file, stringsAsFactors = FALSE) %>%
  mutate(Postcode = clean_postcode(Postcode))

broadband_data <- read.csv(broadband_file, stringsAsFactors = FALSE) %>%
  mutate(Postcode = clean_postcode(Postcode))

combined_school_broadband <- merge(school_data, broadband_data, by = "Postcode")

# ### Linear Modeling: Attainment 8 Score vs Average Download Speed ###
# Linear model: Attainment 8 Score vs Average Download Speed
model_attainment_speed <- lm(Average_Download_Speed ~ Attainment_8_Avg, data = combined_school_broadband)
summary(model_attainment_speed)

# Scatter plot with regression line: Attainment 8 Score vs Average Download Speed
ggplot(combined_school_broadband, aes(x = Attainment_8_Avg, y = Average_Download_Speed)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Attainment 8 Score vs Average Download Speed",
       x = "Average Download Speed (Mbps)",
```

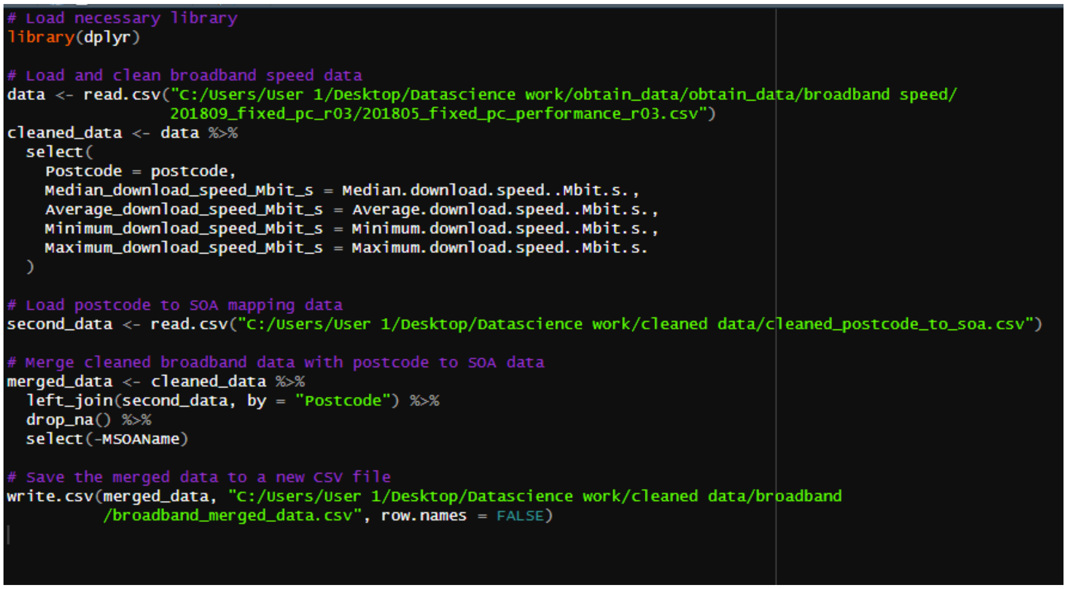
Cleaning code



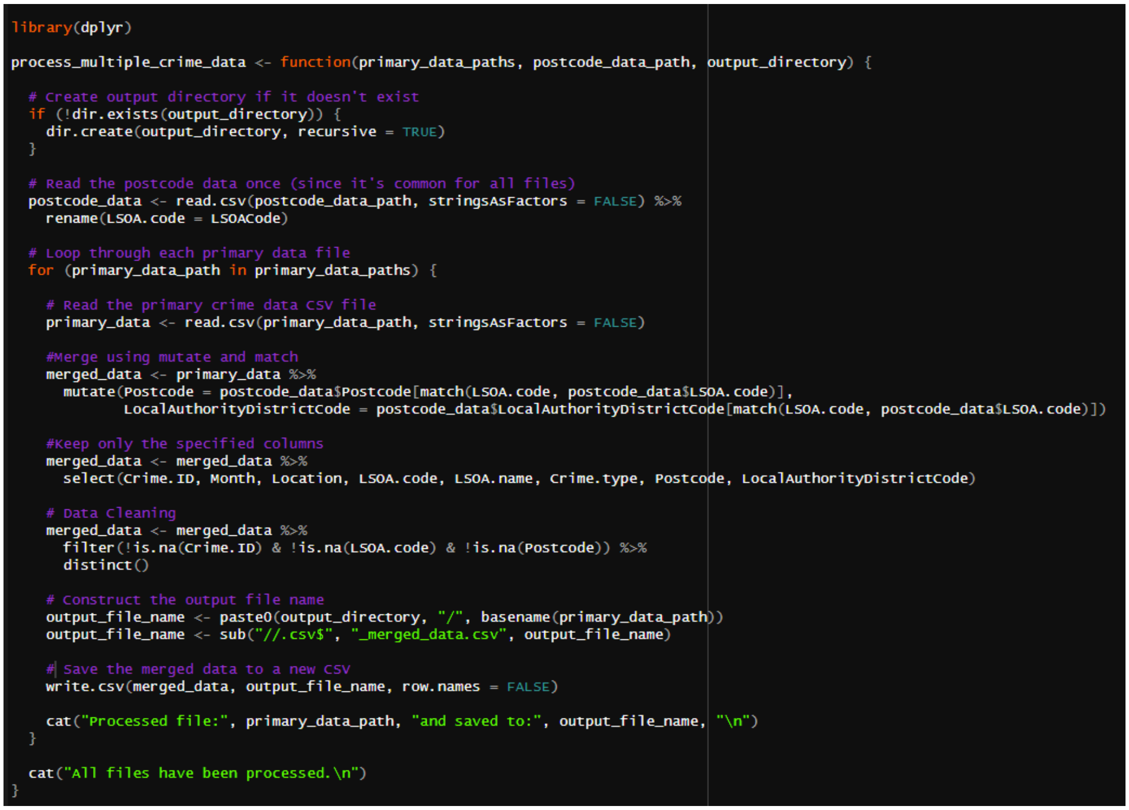
```
# Load necessary libraries
library(dplyr)
library(readr)

# Define a function to process each year's data
process_data <- function(input_path, output_path) {
  # Define new column names
  new_column_names <- c('PropertyID', 'Price', 'SaleDate', 'Postcode',
                        'PropertyType', 'Tenure', 'SaleType',
                        'PAON', 'SAON', 'Street', 'Locality', 'City',
                        'District', 'County', 'PPD Category Type', 'Status')

  # Read the CSV file and assign new column names
  data <- read_csv(input_path, show_col_types = FALSE)
  names(data) <- new_column_names

  # Clean and filter the data
  cleaned_data <- data %>%
    select(-Tenure, -SaleType, -SAON, -`PPD Category Type`, -Status) %>%
    filter(County %in% c("CITY OF BRISTOL", "CORNWALL")) %>%
    drop_na() %>%
    mutate(
      Price = as.numeric(Price),
      SaleDate = as.Date(SaleDate, format = "%Y-%m-%d"),
      PropertyID = as.character(PropertyID),
      Postcode = gsub(" ", "", as.character(Postcode)),
      PropertyType = as.character(PropertyType),
      PAON = as.character(PAON),
      Street = as.character(Street),
      Locality = as.character(Locality),
      City = as.character(City),
      District = as.character(District),
      County = as.character(County)
    ) %>%
    mutate(across(where(is.character), ~trimws(.)))

  # Save the cleaned data
  write_csv(cleaned_data, output_path)
}

# Process data for each year
years <- c("2020", "2021", "2022", "2023")
for (year in years) {
  input_file <- paste0("C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/Housing/pp-", year, ".csv")
  output_file <- paste0("C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_", year, ".csv")
  process_data(input_file, output_file)
}
```



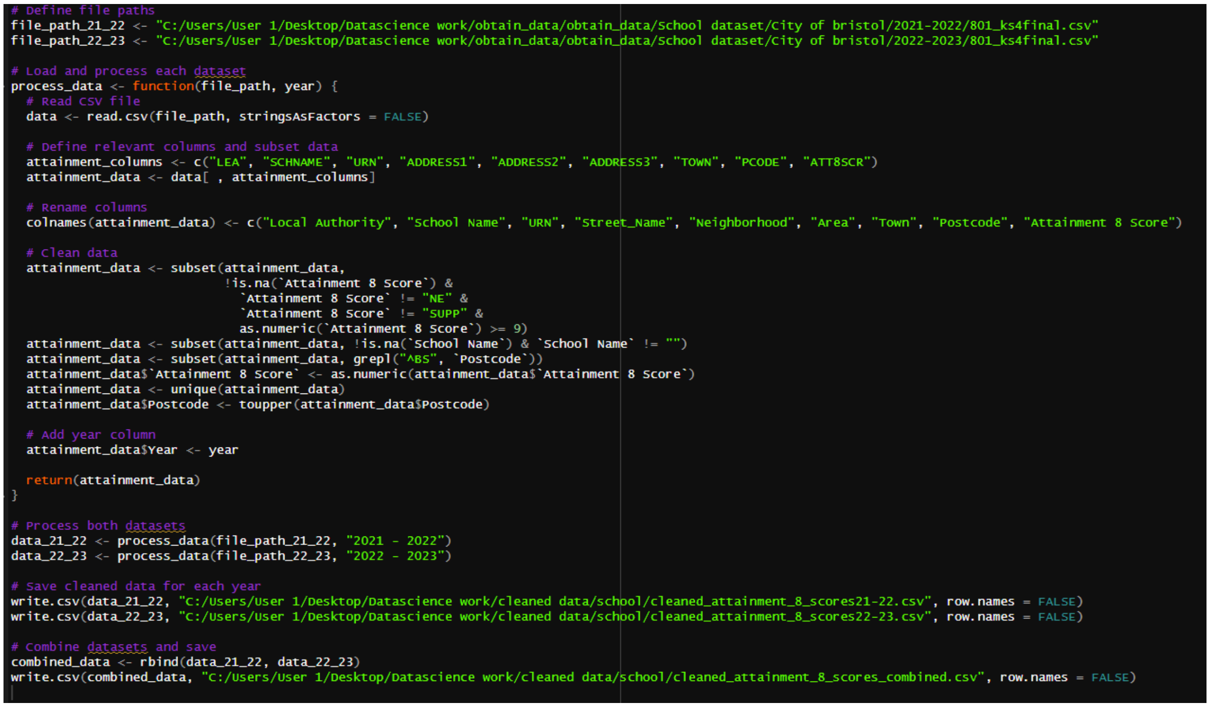
```
# Merge cleaned data
file_paths <- list(
  "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_filtered2020.csv",
  "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_2021.csv",
  "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_2022.csv",
  "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_housing_2023.csv"
)
combined_data <- file_paths %>%
  lapply(read_csv, show_col_types = FALSE) %>%
  bind_rows()
write_csv(combined_data, "C:/Users/User 1/Desktop/Datascience work/cleaned data/combined_housing_data.csv") #save the file
```



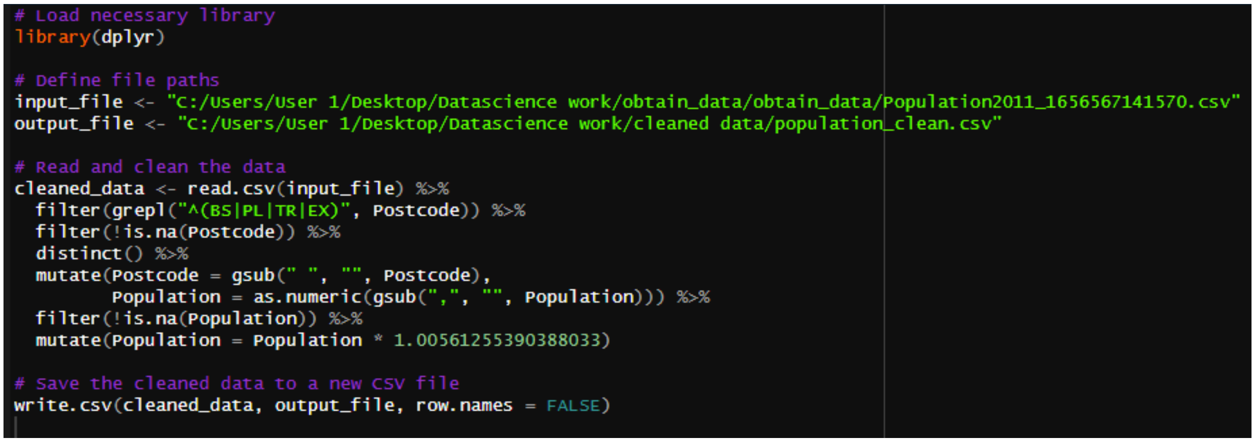
```
# Load necessary library
library(dplyr)

# Load and clean broadband speed data
data <- read.csv("C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/broadband speed/
                 201809_fixed_pc_r03/201805_fixed_pc_performance_r03.csv")
cleaned_data <- data %>%
  select(
    Postcode = postcode,
    Median_download_speed_Mbit_s = Median.download.speed..Mbit.s.,
    Average_download_speed_Mbit_s = Average.download.speed..Mbit.s.,
    Minimum_download_speed_Mbit_s = Minimum.download.speed..Mbit.s.,
    Maximum_download_speed_Mbit_s = Maximum.download.speed..Mbit.s.
  )

# Load postcode to SOA mapping data
second_data <- read.csv("C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_postcode_to_soa.csv")

# Merge cleaned broadband data with postcode to SOA data
merged_data <- cleaned_data %>%
  left_join(second_data, by = "Postcode") %>%
  drop_na() %>%
  select(-MSOAName)

# Save the merged data to a new CSV file
write.csv(merged_data, "C:/Users/User 1/Desktop/Datascience work/cleaned data/broadband
          /broadband_merged_data.csv", row.names = FALSE)
```



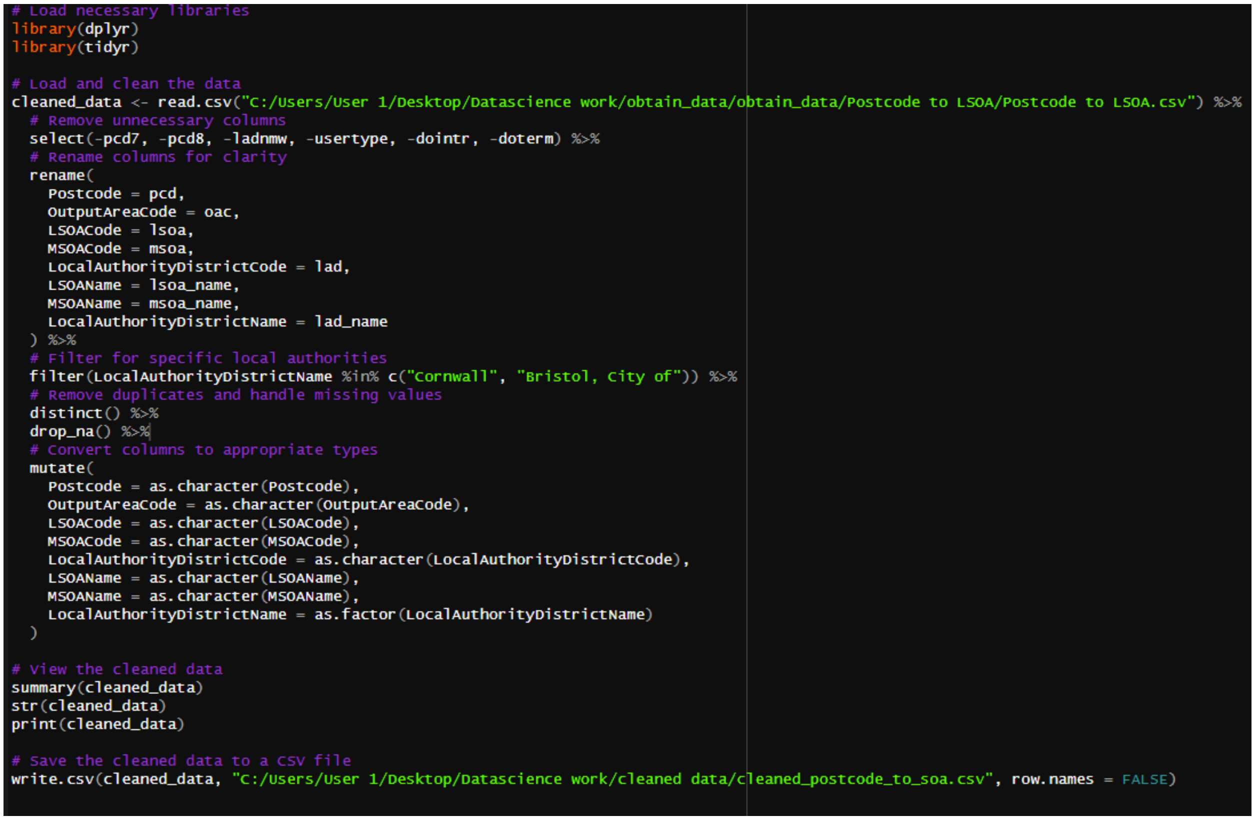
```r
library(dplyr)

process_multiple_crime_data <- function(primary_data_paths, postcode_data_path, output_directory) {

  # Create output directory if it doesn't exist
  if (!dir.exists(output_directory)) {
    dir.create(output_directory, recursive = TRUE)
  }

  # Read the postcode data once (since it's common for all files)
  postcode_data <- read.csv(postcode_data_path, stringsAsFactors = FALSE) %>%
    rename(LSOA.code = LSOACode)

  # Loop through each primary data file
  for (primary_data_path in primary_data_paths) {

    # Read the primary crime data CSV file
    primary_data <- read.csv(primary_data_path, stringsAsFactors = FALSE)

    #Merge using mutate and match
    merged_data <- primary_data %>%
      mutate(Postcode = postcode_data$Postcode[match(LSOA.code, postcode_data$LSOA.code)],
             LocalAuthorityDistrictCode = postcode_data$LocalAuthorityDistrictCode[match(LSOA.code, postcode_data$LSOA.code)])

    #Keep only the specified columns
    merged_data <- merged_data %>%
      select(Crime.ID, Month, Location, LSOA.code, LSOA.name, Crime.type, Postcode, LocalAuthorityDistrictCode)

    # Data Cleaning
    merged_data <- merged_data %>%
      filter(!is.na(Crime.ID) & !is.na(LSOA.code) & !is.na(Postcode)) %>%
      distinct()

    # Construct the output file name
    output_file_name <- paste0(output_directory, "/", basename(primary_data_path))
    output_file_name <- sub("//.csv$", "_merged_data.csv", output_file_name)

    # Save the merged data to a new CSV
    write.csv(merged_data, output_file_name, row.names = FALSE)

    cat("Processed file:", primary_data_path, "and saved to:", output_file_name, "\n")
  }

  cat("All files have been processed.\n")
}
```



```r
# Define file paths
file_path_21_22 <- "C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/School dataset/City of bristol/2021-2022/801_ks4final.csv"
file_path_22_23 <- "C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/School dataset/City of bristol/2022-2023/801_ks4final.csv"

# Load and process each dataset
process_data <- function(file_path, year) {
  # Read CSV file
  data <- read.csv(file_path, stringsAsFactors = FALSE)

  # Define relevant columns and subset data
  attainment_columns <- c("LEA", "SCHNAME", "URN", "ADDRESS1", "ADDRESS2", "ADDRESS3", "TOWN", "PCODE", "ATT8SCR")
  attainment_data <- data[ , attainment_columns]

  # Rename columns
  colnames(attainment_data) <- c("Local Authority", "School Name", "URN", "Street_Name", "Neighborhood", "Area", "Town", "Postcode", "Attainment 8 Score")

  # Clean data
  attainment_data <- subset(attainment_data,
                            !is.na(`Attainment 8 Score`) &
                              `Attainment 8 Score` != "NE" &
                              `Attainment 8 Score` != "SUPP" &
                              as.numeric(`Attainment 8 Score`) >= 9)
  attainment_data <- subset(attainment_data, !is.na(`School Name`) & `School Name` != "")
  attainment_data <- subset(attainment_data, grepl("^BS", `Postcode`))
  attainment_data$`Attainment 8 Score` <- as.numeric(attainment_data$`Attainment 8 Score`)
  attainment_data <- unique(attainment_data)
  attainment_data$Postcode <- toupper(attainment_data$Postcode)

  # Add year column
  attainment_data$Year <- year

  return(attainment_data)
}

# Process both datasets
data_21_22 <- process_data(file_path_21_22, "2021 - 2022")
data_22_23 <- process_data(file_path_22_23, "2022 - 2023")

# Save cleaned data for each year
write.csv(data_21_22, "C:/Users/User 1/Desktop/Datascience work/cleaned data/school/cleaned_attainment_8_scores21-22.csv", row.names = FALSE)
write.csv(data_22_23, "C:/Users/User 1/Desktop/Datascience work/cleaned data/school/cleaned_attainment_8_scores22-23.csv", row.names = FALSE)

# Combine datasets and save
combined_data <- rbind(data_21_22, data_22_23)
write.csv(combined_data, "C:/Users/User 1/Desktop/Datascience work/cleaned data/school/cleaned_attainment_8_scores_combined.csv", row.names = FALSE)
```

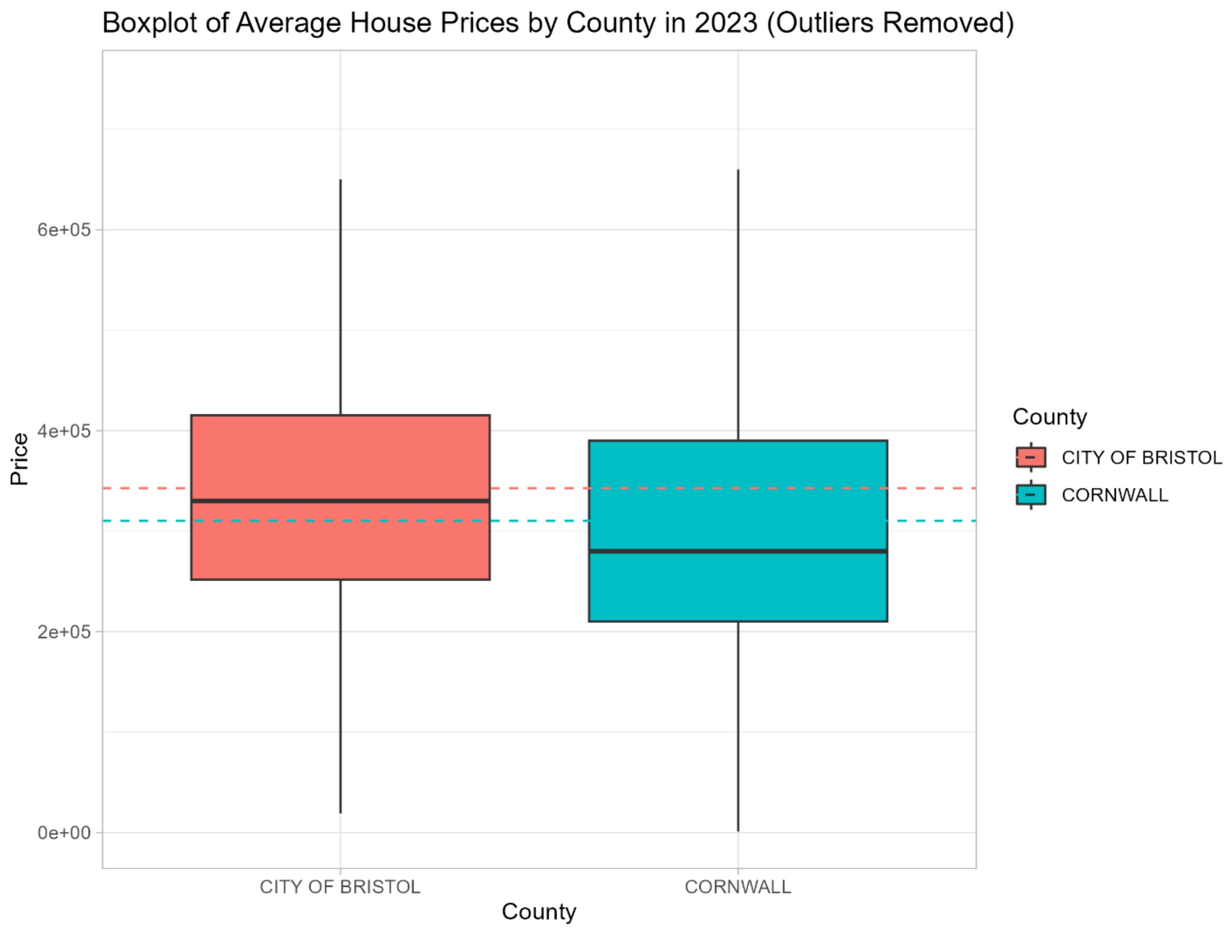


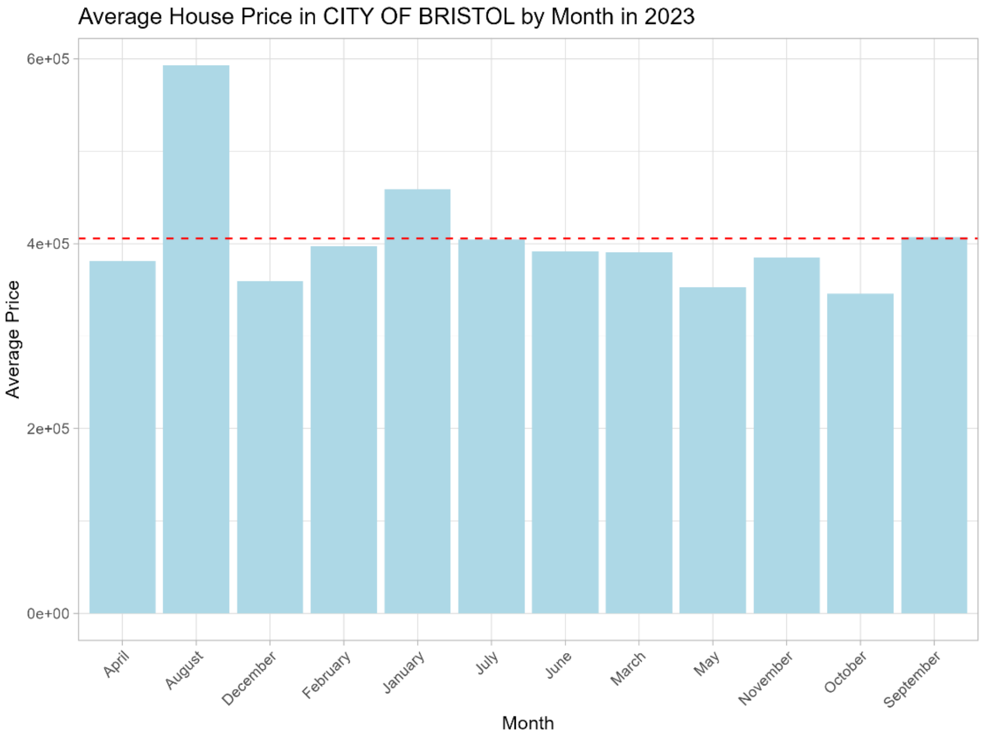
```
# Load necessary library
library(dplyr)

# Define file paths
input_file <- "C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/Population2011_1656567141570.csv"
output_file <- "C:/Users/User 1/Desktop/Datascience work/cleaned data/population_clean.csv"

# Read and clean the data
cleaned_data <- read.csv(input_file) %>%
  filter(grepl("^(BS|PL|TR|EX)", Postcode)) %>%
  filter(!is.na(Postcode)) %>%
  distinct() %>%
  mutate(Postcode = gsub(" ", "", Postcode),
         Population = as.numeric(gsub(",", "", Population))) %>%
  filter(!is.na(Population)) %>%
  mutate(Population = Population * 1.00561255390388033)

# Save the cleaned data to a new CSV file
write.csv(cleaned_data, output_file, row.names = FALSE)
```

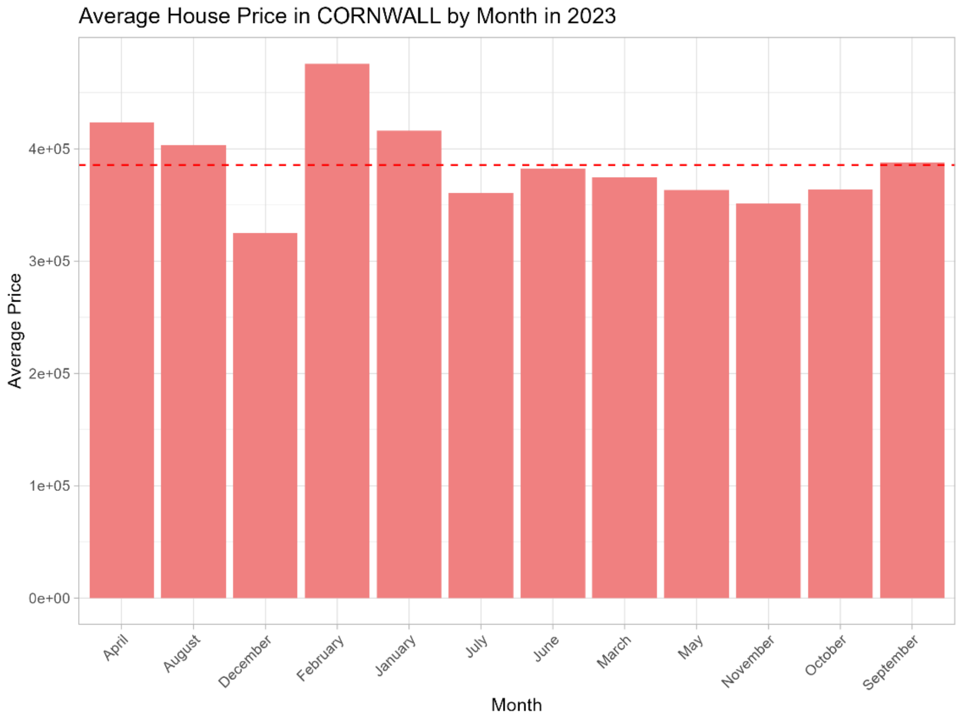


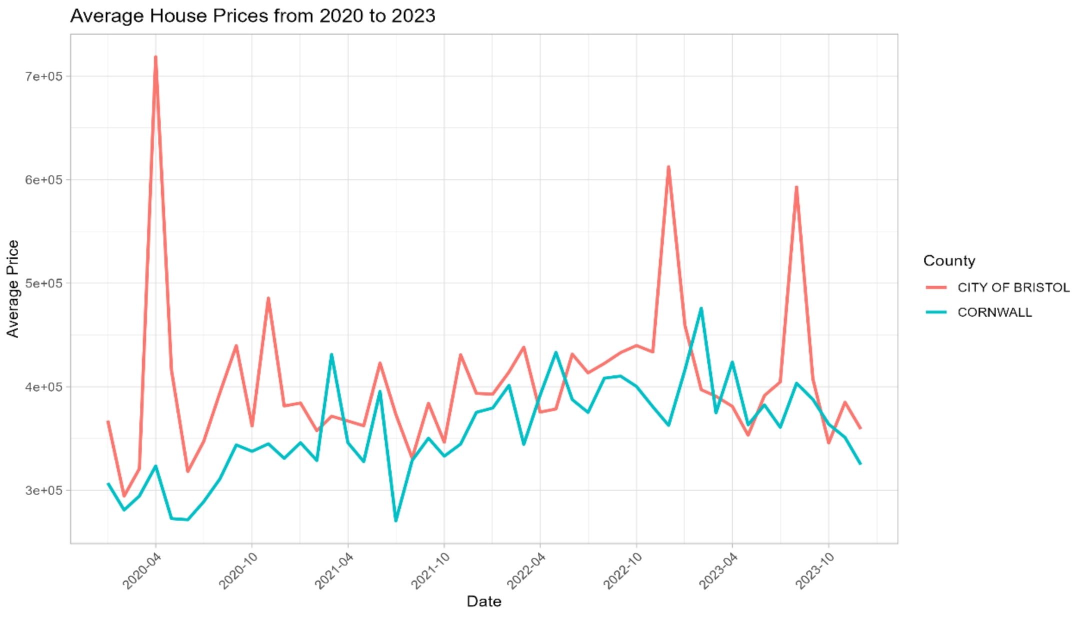
```
# Load necessary libraries
library(dplyr)
library(tidyr)

# Load and clean the data
cleaned_data <- read.csv("C:/Users/User 1/Desktop/Datascience work/obtain_data/obtain_data/Postcode to LSOA/Postcode to LSOA.csv") %>%
  # Remove unnecessary columns
  select(-pcd7, -pcd8, -ladnmw, -usertype, -dointr, -doterm) %>%
  # Rename columns for clarity
  rename(
    Postcode = pcd,
    OutputAreaCode = oac,
    LSOACode = lsoa,
    MSOACode = msoa,
    LocalAuthorityDistrictCode = lad,
    LSOAName = lsoa_name,
    MSOAName = msoa_name,
    LocalAuthorityDistrictName = lad_name
  ) %>%
  # Filter for specific local authorities
  filter(LocalAuthorityDistrictName %in% c("Cornwall", "Bristol, City of")) %>%
  # Remove duplicates and handle missing values
  distinct() %>%
  drop_na() %>%
  # Convert columns to appropriate types
  mutate(
    Postcode = as.character(Postcode),
    OutputAreaCode = as.character(OutputAreaCode),
    LSOACode = as.character(LSOACode),
    MSOACode = as.character(MSOACode),
    LocalAuthorityDistrictCode = as.character(LocalAuthorityDistrictCode),
    LSOAName = as.character(LSOAName),
    MSOAName = as.character(MSOAName),
    LocalAuthorityDistrictName = as.factor(LocalAuthorityDistrictName)
  )

# View the cleaned data
summary(cleaned_data)
str(cleaned_data)
print(cleaned_data)

# Save the cleaned data to a CSV file
write.csv(cleaned_data, "C:/Users/User 1/Desktop/Datascience work/cleaned data/cleaned_postcode_to_soa.csv", row.names = FALSE)
```

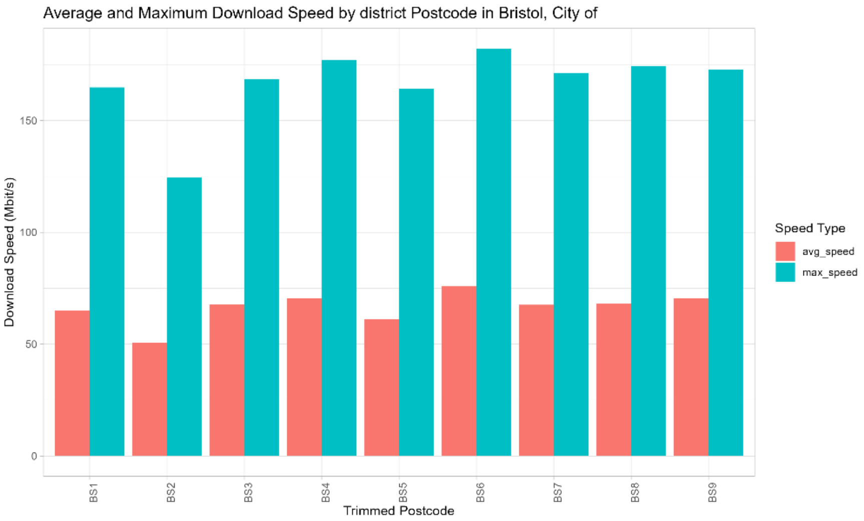
Graphs


Boxplot of Average House Prices by County in 2023 (Outliers Removed)
Price
6e+05
4e+05
2e+05
0e+00
CITY OF BRISTOL
CORNWALL
County
County
CITY OF BRISTOL
CORNWALL

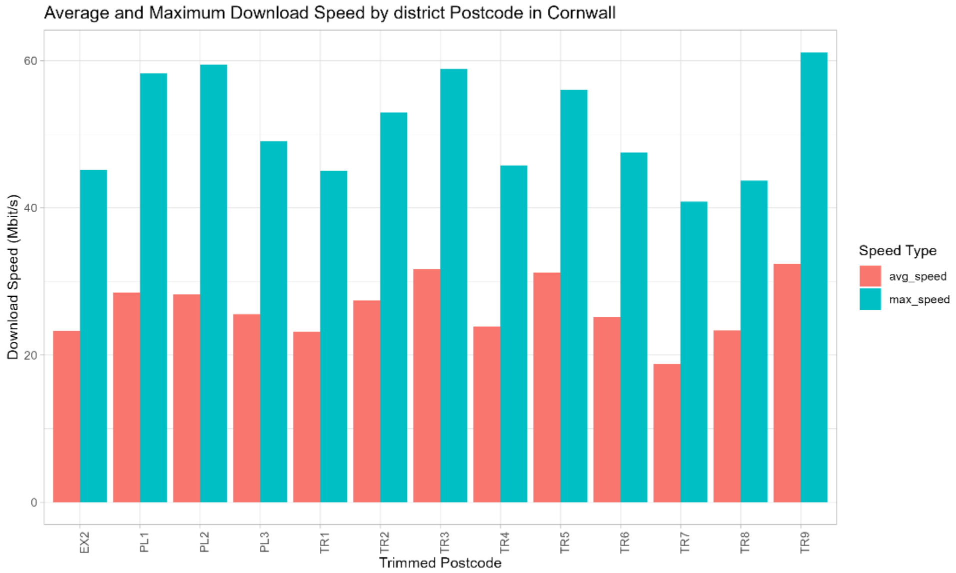

Average House Price in CITY OF BRISTOL by Month in 2023
6e+05
4e+05
2e+05
0e+00
Average Price
April
August
December
February
January
July
June
March
May
November
October
September
Month


Average House Price in CORNWALL by Month in 2023
Average Price
4e+05
3e+05
2e+05
1e+05
0e+00
April
August
December
February
January
July
June
March
May
November
October
September
Month

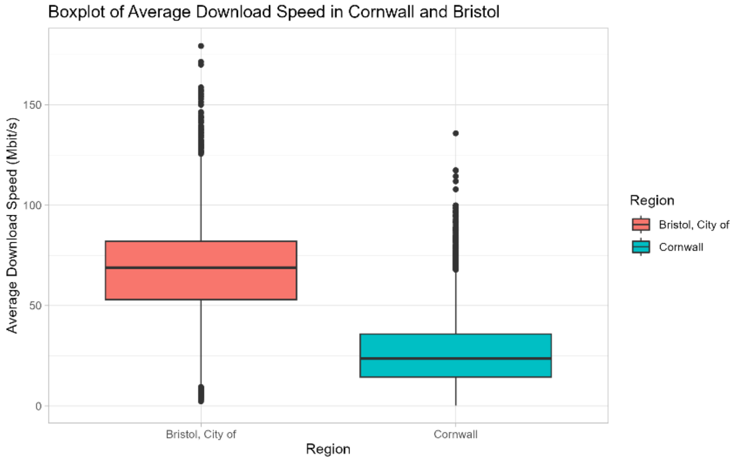

Average House Prices from 2020 to 2023
Average Price
7e+05
6e+05
5e+05
4e+05
3e+05
2020-04
2020-10
2021-04
2021-10
2022-04
2022-10
2023-04
2023-10
Date
County
CITY OF BRISTOL
CORNWALL



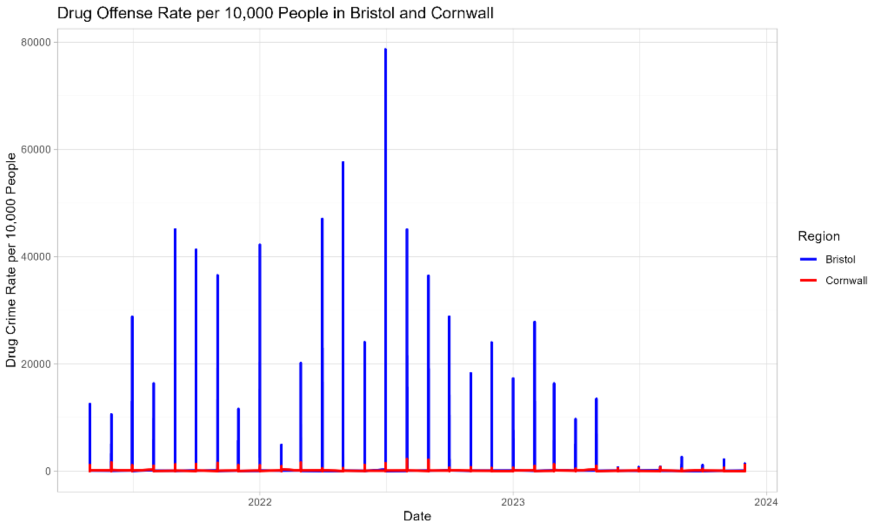
Average and Maximum Download Speed by district Postcode in Bristol, City of



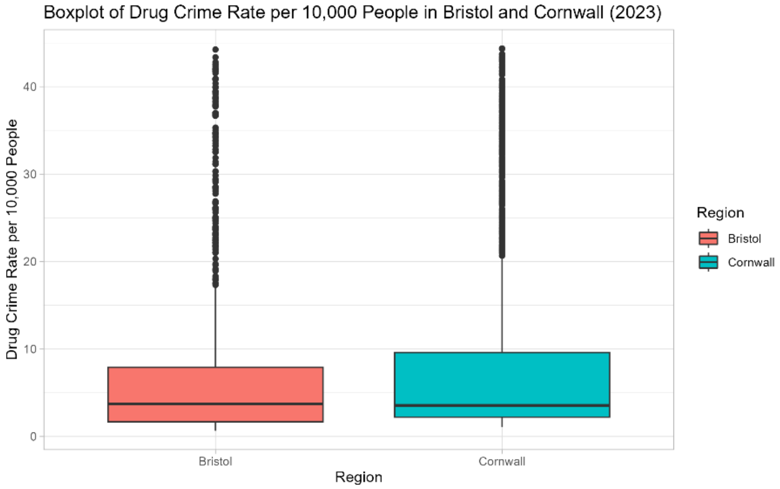
Average and Maximum Download Speed by district Postcode in Cornwall



Boxplot of Average Download Speed in Cornwall and Bristol

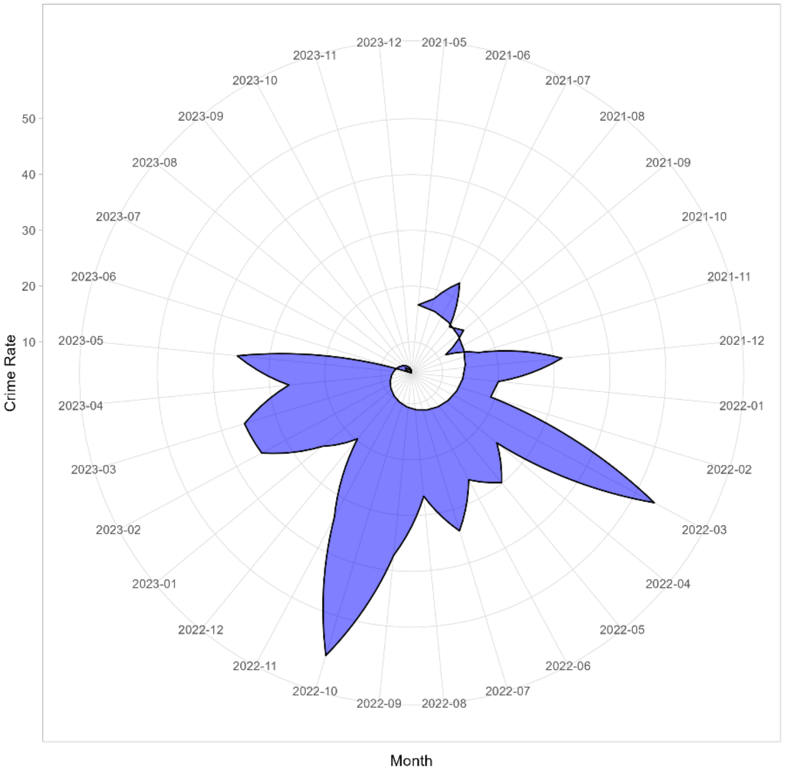


Drug Offense Rate per 10,000 People in Bristol and Cornwall

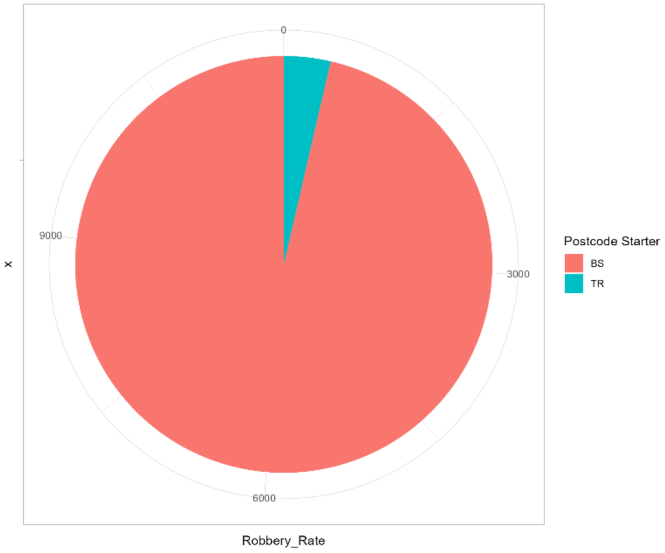


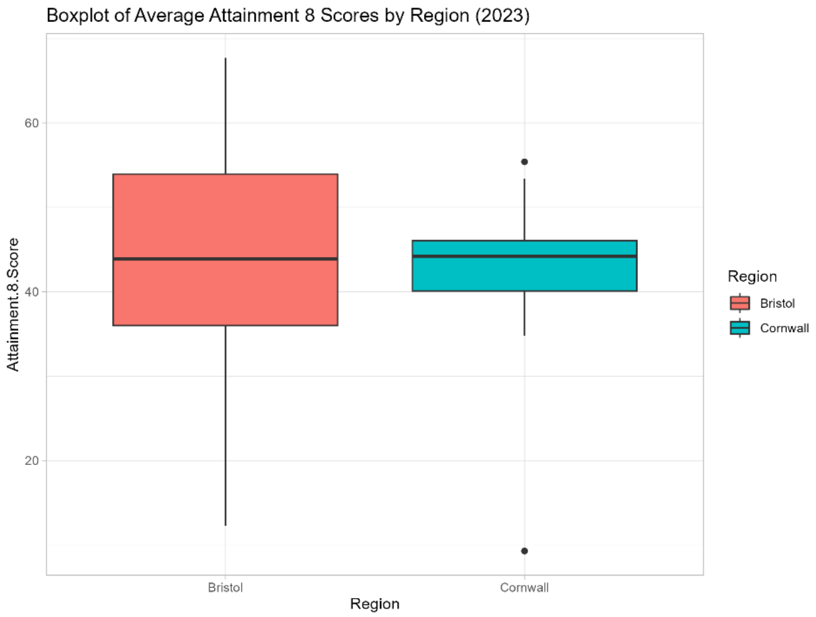
Boxplot of Drug Crime Rate per 10,000 People in Bristol and Cornwall (2023)

Radar Chart of Vehicle Crime Rate per 10,000 People (2020-2023)

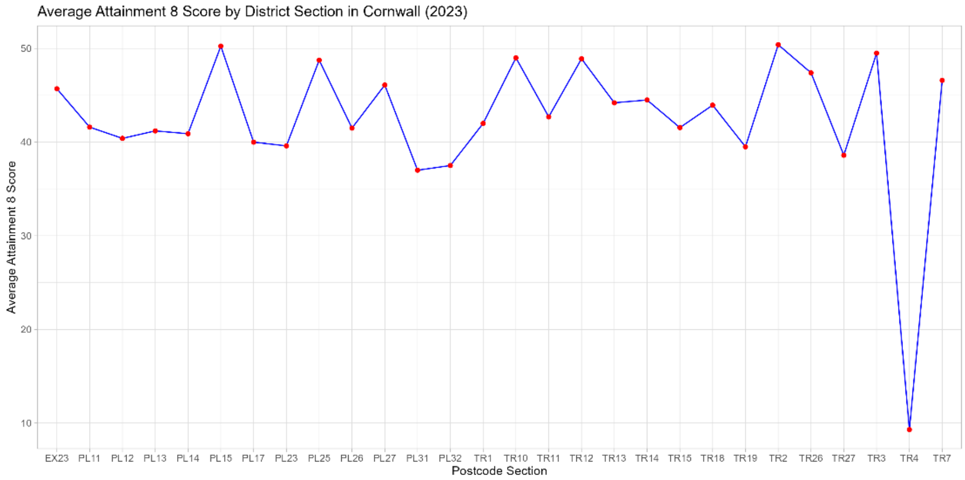


Pie Chart of Robbery Rates by Postcode Starter (July 2023)

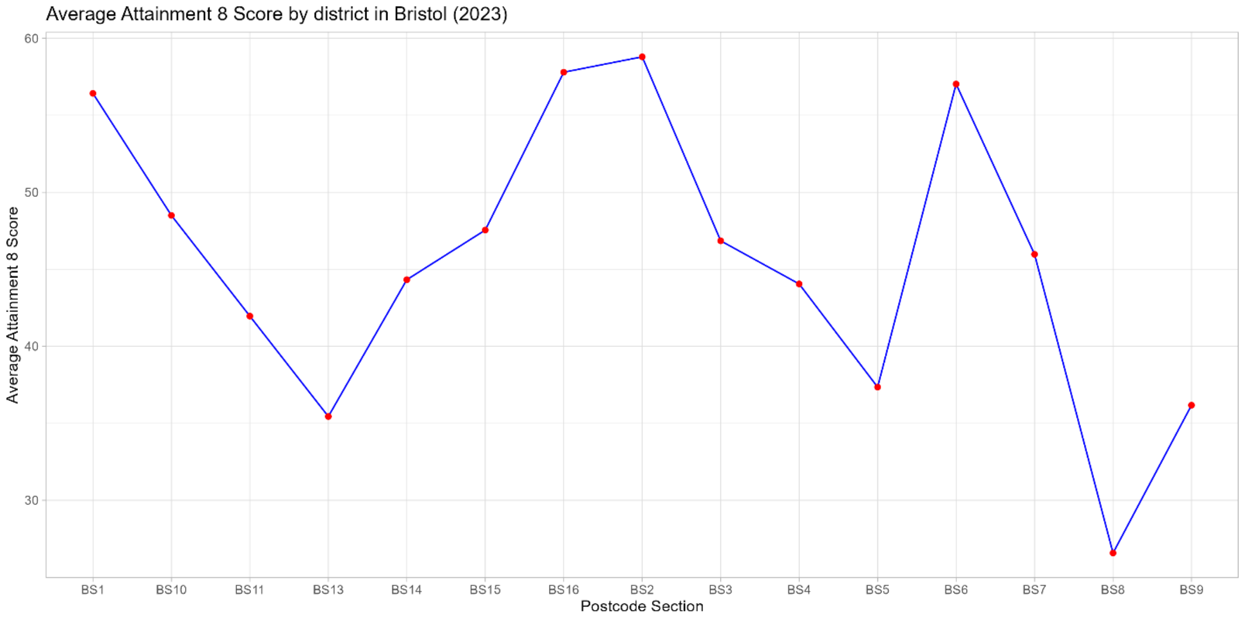




Boxplot of Average Attainment 8 Scores by Region (2023)



Average Attainment 8 Score by District Section in Cornwall (2023)


Average Attainment 8 Score by district in Bristol (2023)
Average Attainment 8 Score
60
50
40
30
BS1
BS10
BS11
BS13
BS14
BS15
BS16
BS2
BS3
BS4
BS5
BS6
BS7
BS8
BS9
Postcode Section