Softwarica College of IT & E-commerce

ST5014CEM Data Science for Developers

August 15, 2024

Siddhartha Neupane

Submitted By:

Aabrity Dhungana

202270

Table of Contents

Table of Figures

Introduction

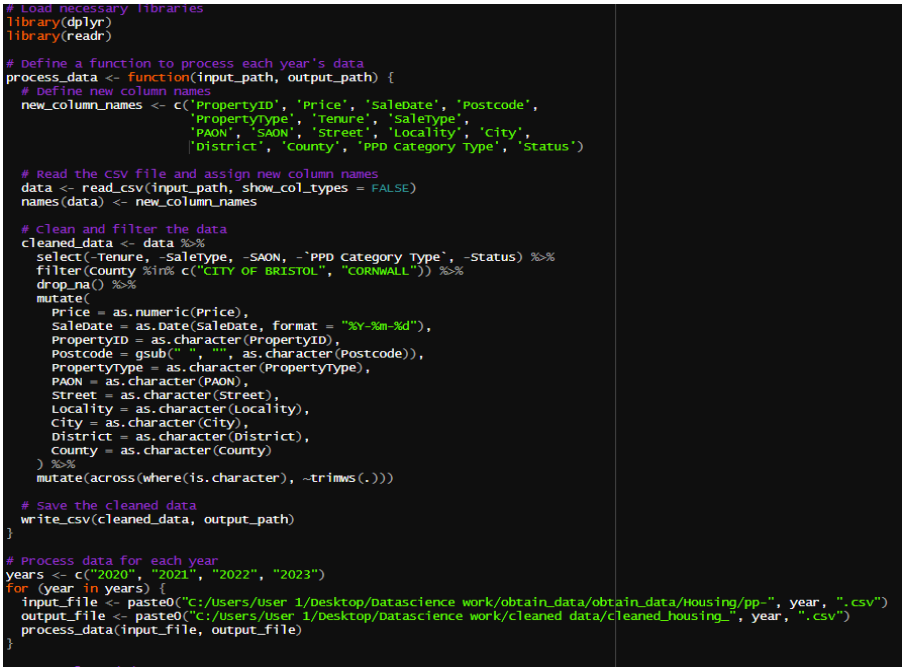
The report is about a setup and review that were enacted in an instance when relatives of data analyst seeks residence and there is need for a recommendation involving the expertise of the analyst to identify the best towns where one can invest in property within Bristol and Cornwall. A successful investor 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.

Thus, this report aims at identifying three topmost townships in Bristol and Cornwall that may count among potential best places to invest in properties. It also goes beyond the traditional measures used in investment decisions like dwelling prices only, emphasizing on broader indicators such as digital connectivity and educational qualifications which are becoming increasingly relevant as far as today’s property markets are concerned.

Cleaning Data

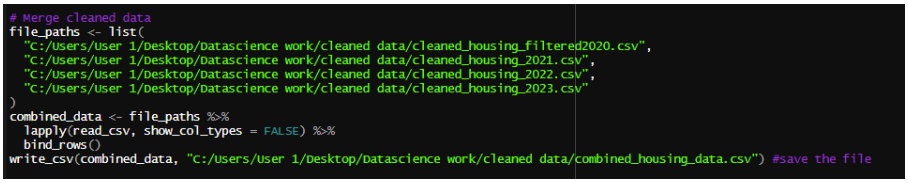
The project started by collecting data from several different places so it could have an all-inclusive database. The next stage according to the data science life cycle was cleaning the data. To do this, the datasets were loaded in the working environment. Some cleaning steps were taken, which included 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 provides a streamlined means by which housing data from multiple years can be cleaned and combined. It begins by loading libraries that are necessary for manipulating data as well as reading CSV files. The central part of the script is a function called process_data, which does activities like changing column names to ones that are more understandable, cleaning up unneeded columns in the dataset, selecting specific counties, or probably altering some data types. Besides making sure postcodes are properly formatted and text fields do not have excess spaces in them, this function also ensures that the postal code is properly formatted and that the leading or trailing spaces are removed from character strings in text variables. This script applies this function to housing data files for 2020-2023 saving each cleaned dataset into a new file. Finally, all these clean datasets are merged into one main file so that it can be easily analyzed holistically.

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

To clean and consolidate postcode-to-LSOA mapping information, this R script is intended for broadband speed data. For the purposes of data manipulation, it starts by loading the dplyr library. Initially, it reads in broadband speed data from a CSV file, chooses and changes names of columns linked to downloading speeds before tidying up the dataset. After that, another CSV file containing postcodes to LSOA codes is loaded. In the script, broadband merge with postcode-to-LSOA takes place whereby all rows with missing values are removed and one irrelevant column is dropped off. Lastly, it saves the cleaned and merged dataset in a new CSV file that can be used for further analysis.

Crime dataset cleaning

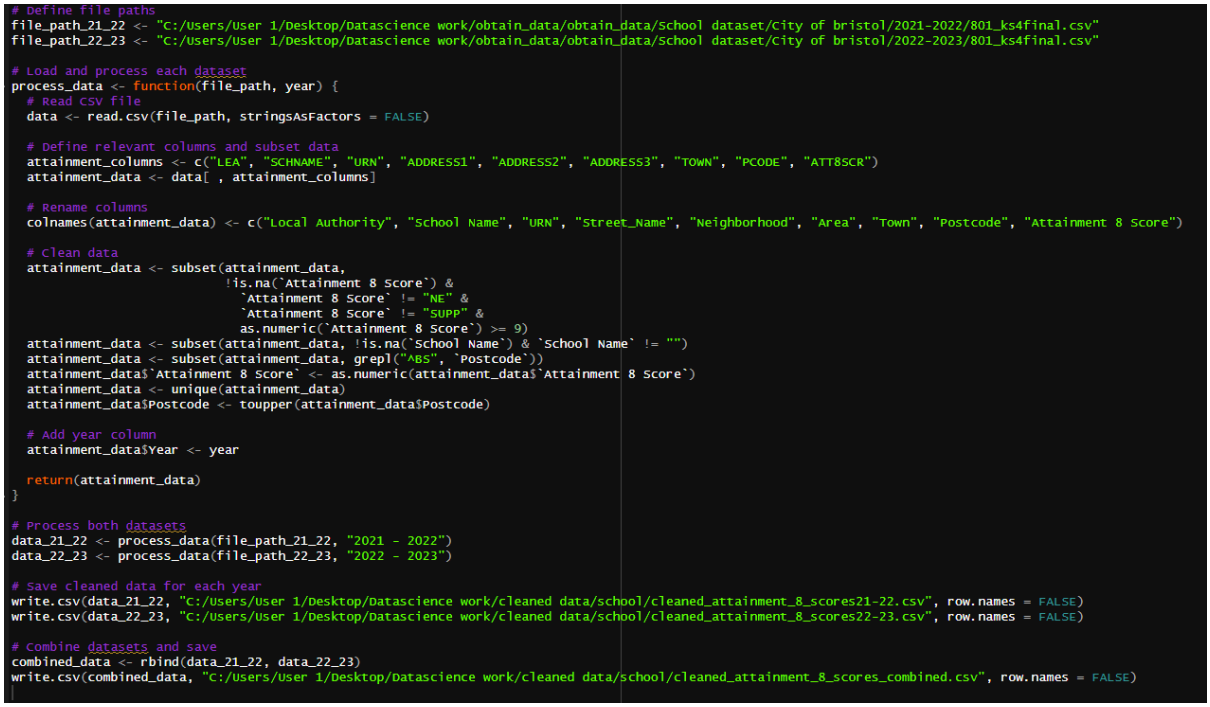
```r
library(dplyr)

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

Due to the large number of files present in the obtained dataset of Crime, making a function to load and process the file was an optimal approach. Cleaning data used in various crimes requires merging multiple files of crime data with one file containing postcodes. This involves two steps including first ensuring the existence of the output directory specified or creating it when it is not there. The second step begins by reading postcode data once because all crime files have the same postcode details and merge each crime data file with this dataset based on a common LSOA.code that allows retrieval of related postal code details, and then select specific columns only. Finally, rows in which key columns have missing values are removed from merged data while duplicates are eliminated. Lastly, within this process the cleaned and merged datasets are saved into new

CSVs within an output folder as well as a message being printed to confirm each file's processing is complete. It then prints a completion message after handling all the files.

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 will process and clean Attainment 8 score data for the academic years 2021-2022 and 2022-2023. It begins by specifying file paths for each dataset and then loads the data. There is a process_data function that takes care of such common tasks as sub setting to relevant columns, renaming them descriptively, and cleaning up data through deleting invalid scores, removing rows without school names, and standardizing postcodes. These datasets are worked on individually before being stored in different CSV files. Later, these cleaned datasets are appended into one dataset which is then saved as a new CSV file to have unified consolidated final output ready for

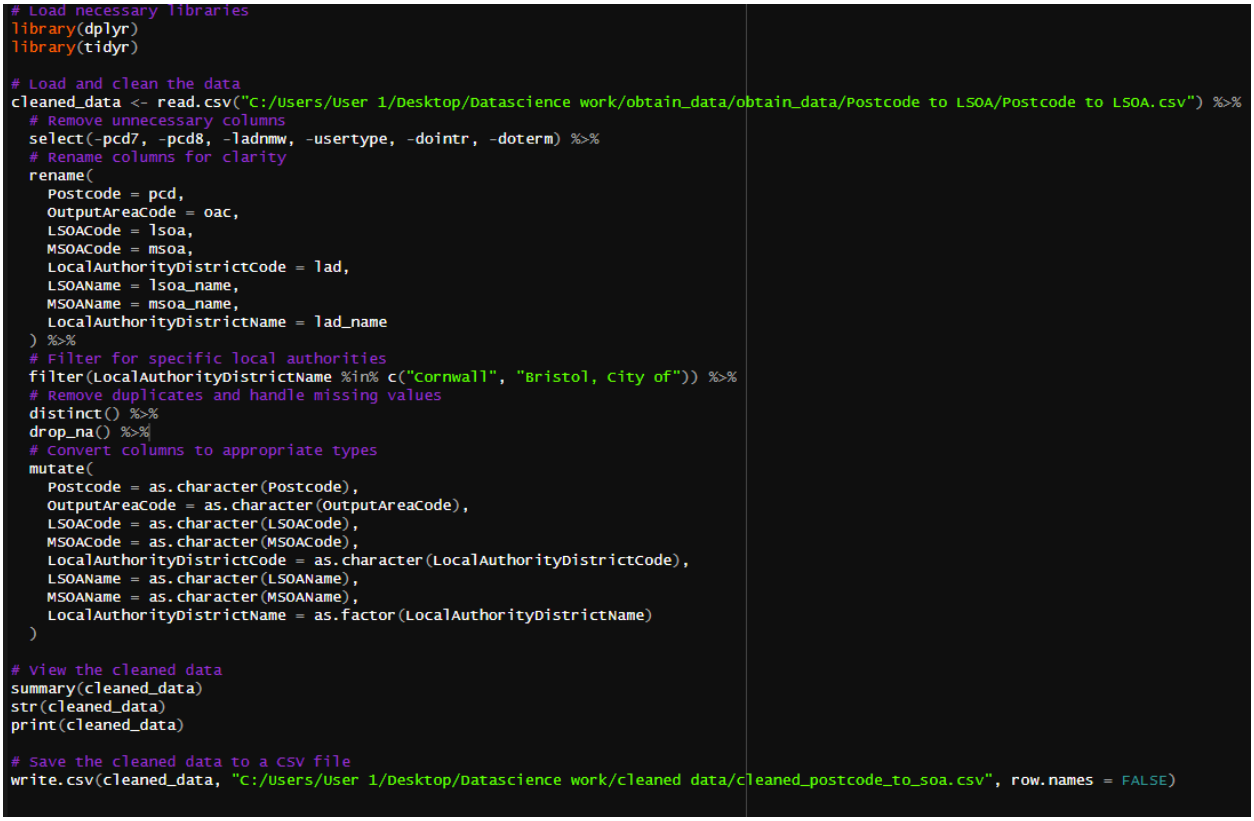
further analysis purposes. This approach in modularization promotes code readability and make it easy to maintain the code.

Population and LSOA dataset cleaning

```r
# 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

To process and clean a population data set, this script utilizes dplyr as one of the major packages for performing data manipulation in R. It begins by importation of datasets from a particular file following which data is filtered to only incorporate rows having postcodes that start with ‘BS’, ‘PL’, ‘TR’, or ‘EX’ where rows missing from any column are dropped. It then removes duplicate rows and trims spaces in the postal code field. The next step is cleaning population data where it removes commas and converts values into numeric form with any row that can not be converted being taken away. Lastly, it updates population values to 2023 using the growth factor and saves the cleaned data as a new CSV file. This method ensures that we have properly cleaned our dataset for further analysis purposes.



```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 use of this script is to accept postcode to SOA mapping data in a way so that it would be loaded from a CSV file and then cleansed several actions will be performed. It entails giving up unwanted columns, renaming the remaining ones for convenience and applying the data filters to limit the records just to Cornwall and Bristol. Correspondingly, the duplicated rows are collected for deleting and the ones with the missing values are finally made incomplete, and perhaps changes in the column type if it were necessary. The verified data is provided at the end of the process before actual production, then, the cleaned one is summarized before being printed out to show how it looks like; ultimately saved as a new CSV file.

EDA (Exploratory Data Analysis)

House Pricing

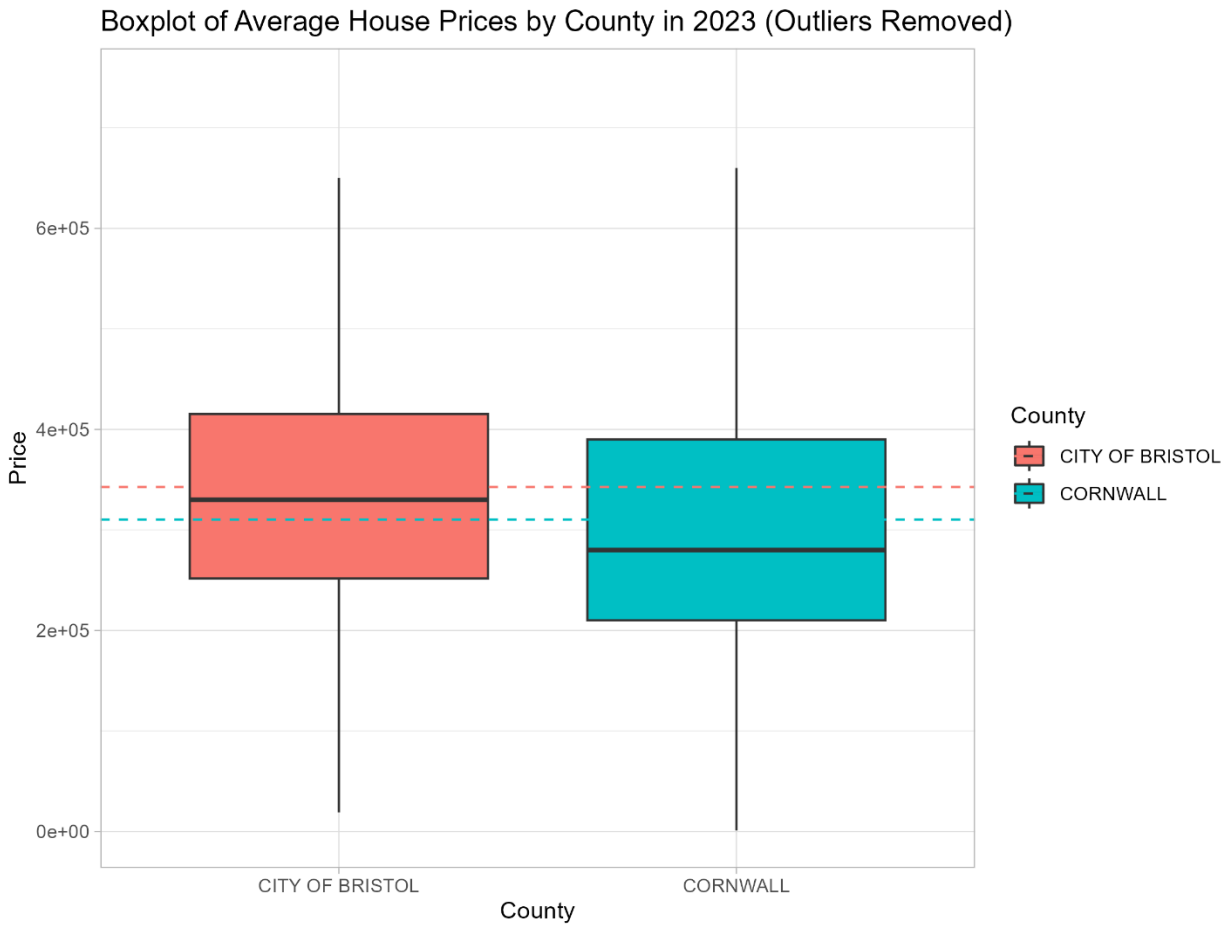


Figure 7: Boxplot Average house price

In 2023, Bristol's housing market features a higher median price and wider interquartile range (IQR), indicating diverse property values and greater variability. Cornwall, on the other hand, has a lower median price with a narrower IQR, reflecting a more stable and uniform market with fewer extremes in pricing.

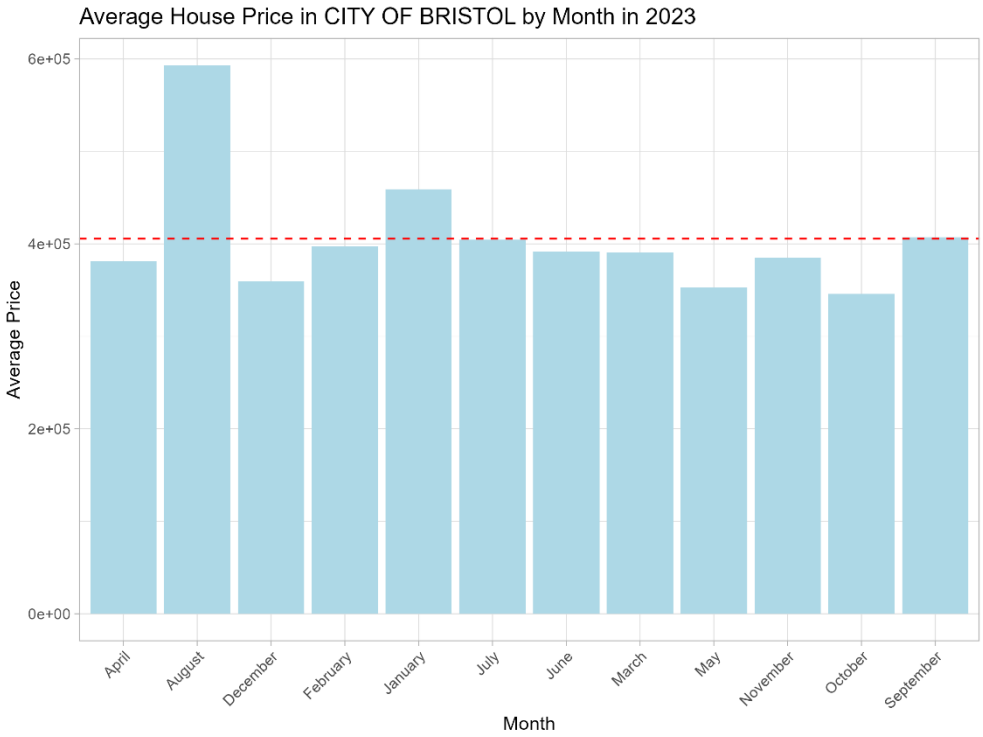


Figure 8 : Average house price bar chart Bristol

In 2023, Bristol's house prices peak in August above 600,000, likely due to increased demand, while February and December see lows below 400,000, possibly from seasonal slowdowns. Despite these fluctuations, prices generally remain stable around 400,000 throughout the year, highlighting the impact of seasonal factors on housing market trends.

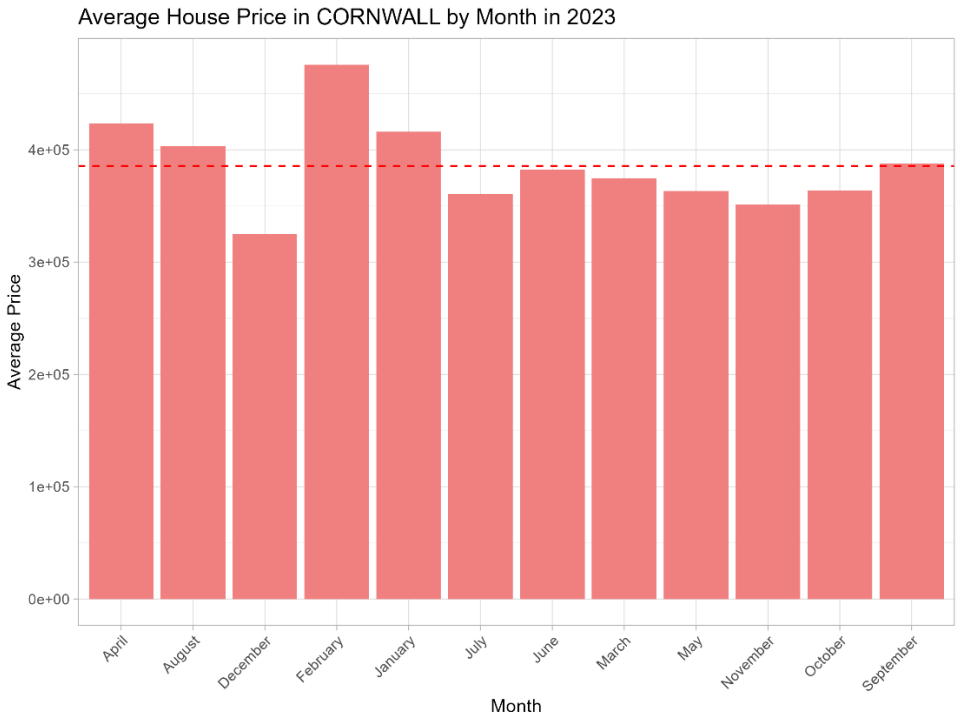


Figure 9 Average house price bar chart Cornwall

In 2023, Cornwall's housing market exhibits seasonal fluctuations with a peak in February and a trough in June. The market shows overall stability with most months around a benchmark price and a recovery towards the year's end. These patterns suggest a moderately consistent market, where timing transactions according to these trends could be advantageous for better deals or higher prices.

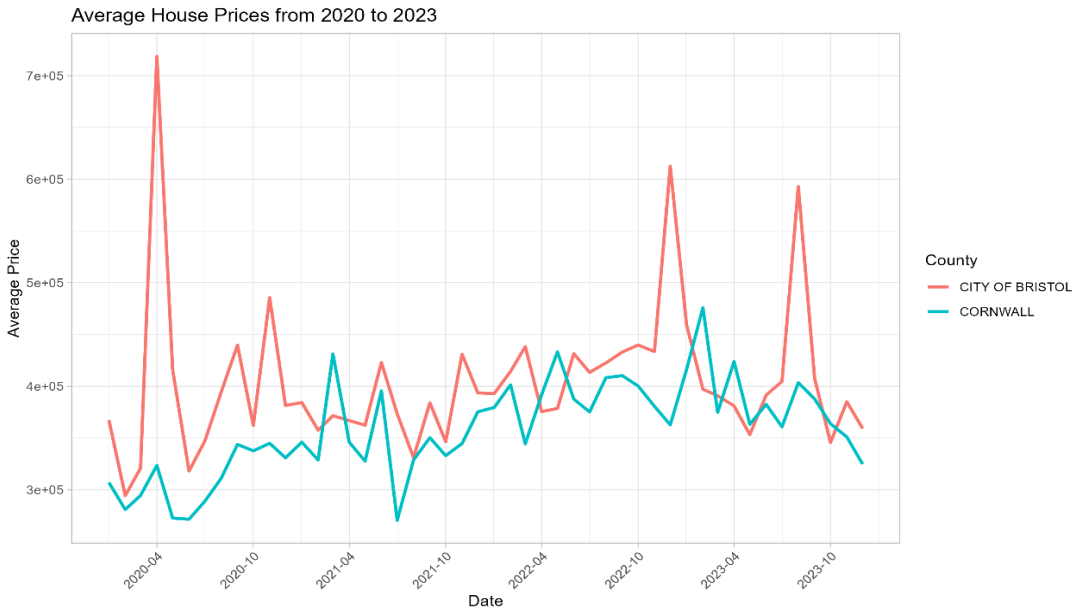


Figure 10: Average House Price Line graph both counties

From 2020 to 2023, Bristol's housing market shows significant volatility with sharp price spikes, notably over £700,000 in early 2020, indicating sensitivity to external factors. In contrast, Cornwall's market remains stable, with prices between £300,000 and £450,000. Post-2021, both markets converge, reflecting broader stabilization, though Bristol continues to experience higher price fluctuations compared to the steadier Cornwall market.

Broad Band

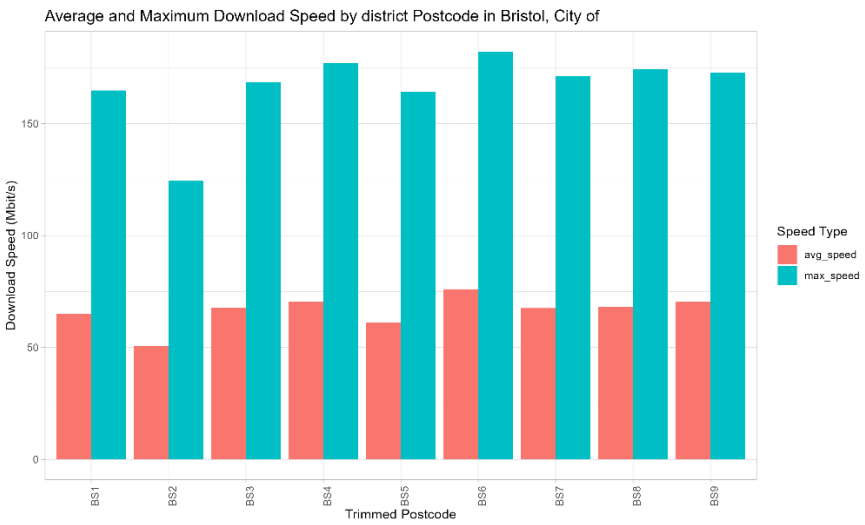


Figure 11 Average and maximum download speed for Bristol Barchart

The bar chart reveals a significant gap between maximum and average download speeds across Bristol's postcode districts. Maximum speeds range from 150 to 175 Mbps, while average speeds are lower, between 50 and 75 Mbps. BS1 and BS6 show the highest average speeds, whereas BS2 and BS5 have the lowest. This disparity suggests factors like network congestion or distance from exchanges affect average speeds, highlighting a need for improved broadband consistency.

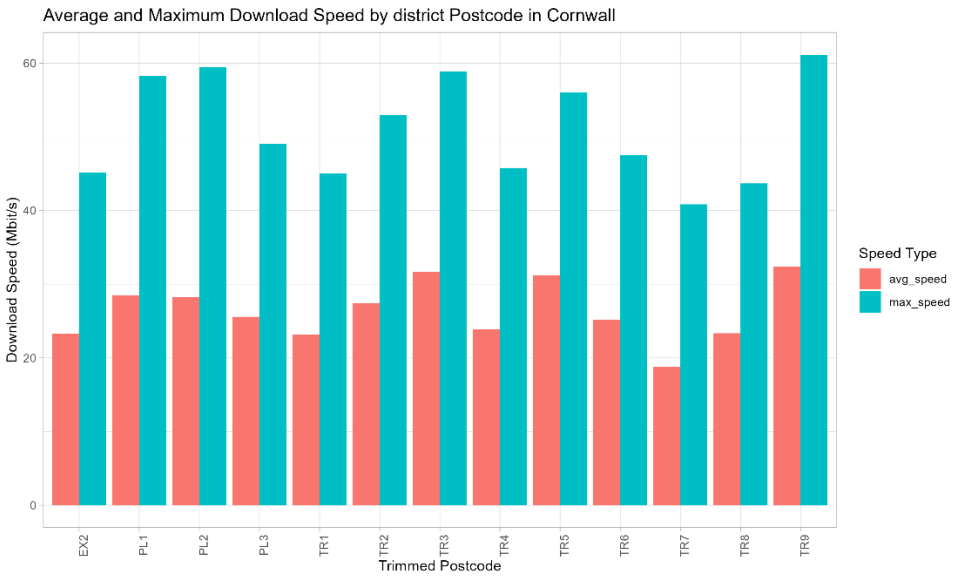


Figure 12: Average and maximum download speed for Cornwall barchat

The bar chart reveals that "TR" postcodes in Cornwall, particularly around Truro, have higher maximum download speeds, indicating superior broadband infrastructure. In contrast, "PL" and "EX" postcodes show more uniform speeds, suggesting standardized service. A consistent gap between average and maximum speeds across all postcodes points to underutilized infrastructure potential. TR6 and TR7 exhibit lower speeds, while TR9 boasts the highest maximum speed, likely due to recent upgrades. Regional disparities in broadband investment are also evident.

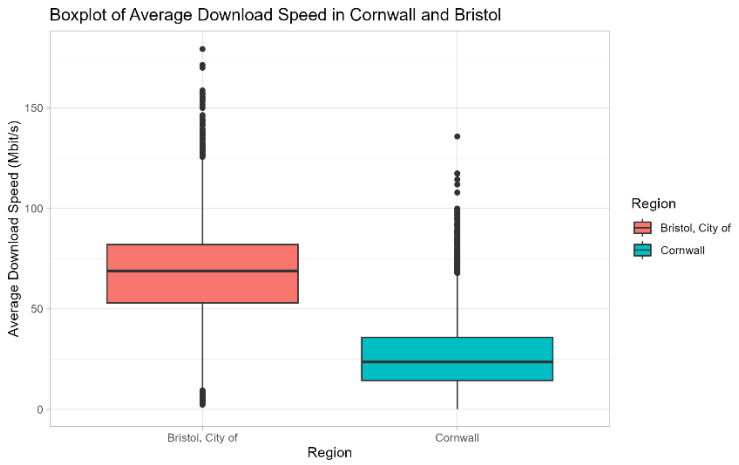


Figure 13: Average download speed in Cornwall and Bristoll Boxplot

The boxplot shows Bristol with higher and more consistent download speeds, mostly between 60 and 100 Mbps, and fewer outliers, indicating reliable broadband performance. In contrast, Cornwall has greater variability, with a wider interquartile range and a lower median speed around 50 Mbps. Numerous high-speed outliers suggest uneven infrastructure, highlighting the need for more consistent broadband across Cornwall to match Bristol's higher standards.

Crime

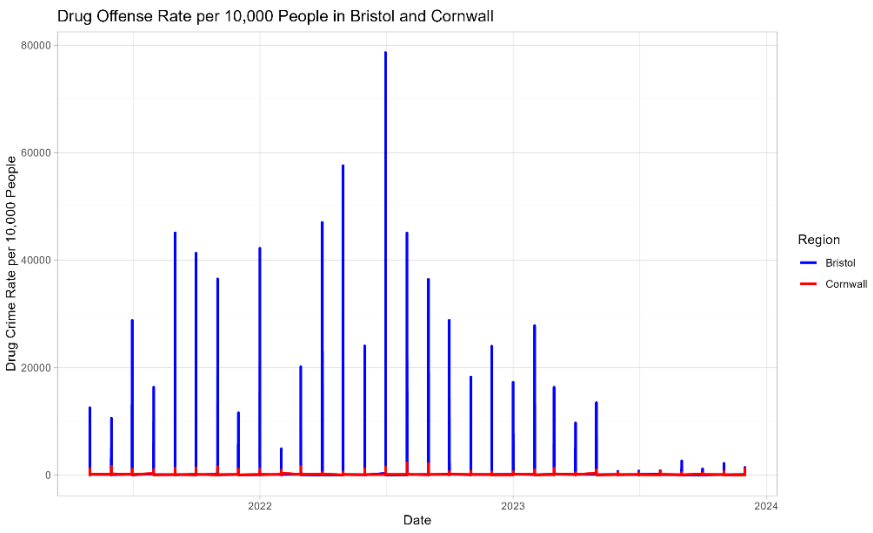


Figure 14: Line chart for Drug offense rate per 10000 people in Bristoll and Cornwall

The analysis shows Bristol with significantly higher drug offense rates and noticeable peaks, especially in mid-2022, possibly due to seasonal factors or spikes in activity. Rates then decline, suggesting successful interventions or behavioral changes. Cornwall's rates remain low and stable throughout, indicating fewer disturbances. Both regions stabilize by the end of the period, with Bristol's rates leveling off and Cornwall maintaining its steady trend. This indicates Bristol faces more fluctuating challenges, while Cornwall experiences a more stable situation.

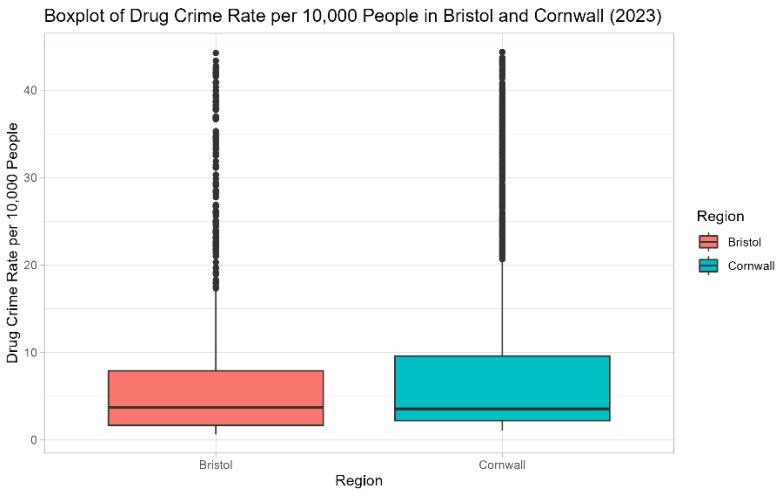


Figure 15: Boxplot Drug Crime Rate per 10000 in Both Counties

The boxplot for 2023 shows that both Bristol and Cornwall have right-skewed drug crime rate distributions, with most rates clustered at lower levels but occasional significant spikes. Bristol's median rate is higher, indicating more consistently elevated drug activity. Cornwall's median is lower, but it has more extreme outliers, reflecting infrequent but sharp surges in drug incidents. Both regions have similar interquartile ranges, suggesting comparable core data variability. Bristol's drug crime is generally stable, while Cornwall experiences more pronounced peaks, which may need targeted attention.

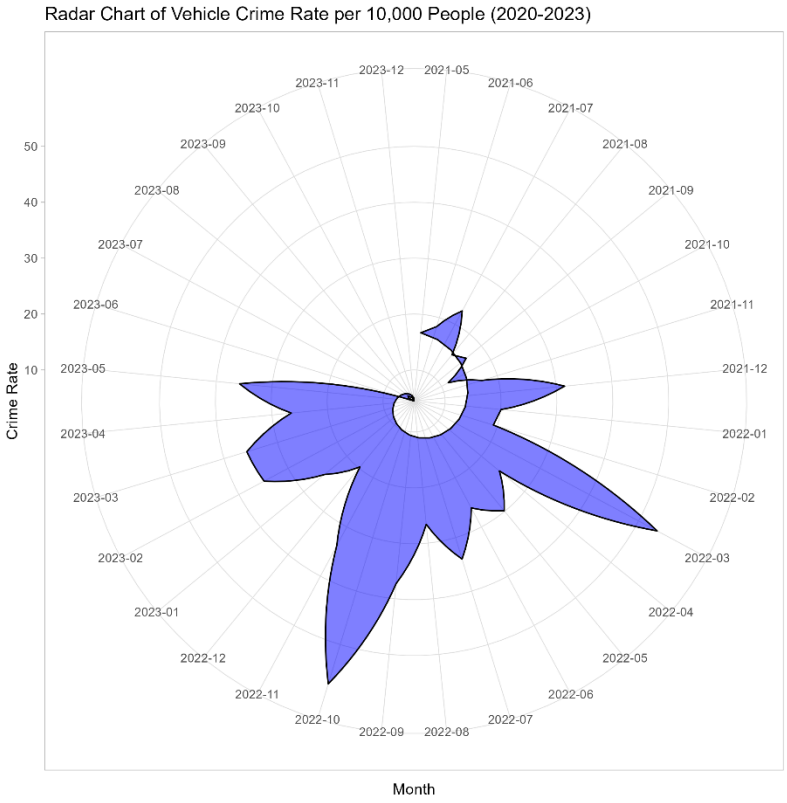


Figure 16: Radar chart Vehicle Crime Rate per 10000 people (20-23)

The radar chart shows that vehicle crime rates peak during the warmer months (May to August) due to increased outdoor activity, while winter months (December to February) have lower rates, likely due to colder conditions. Over the years, there is a gradual decline in overall crime levels, with peaks becoming less pronounced from 2022 onward. This indicates that, despite consistent seasonal patterns, vehicle crime rates are decreasing, possibly due to improved prevention, law enforcement, or vehicle security.

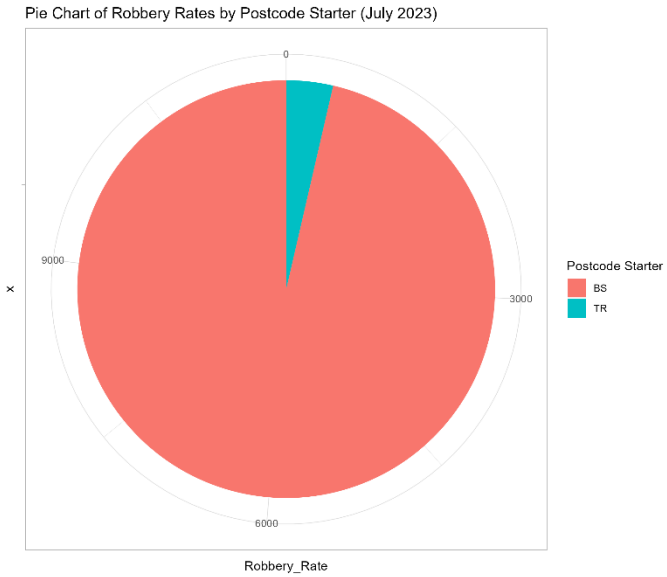


Figure 17: Pie chart of Robbery july 2023

The pie chart shows a significant concentration of robbery incidents in the "BS" postcode area, highlighting a stark imbalance compared to the "TR" area. This suggests that the urban "BS" region experiences much higher robbery rates, likely due to factors such as higher population density and socio-economic conditions. In contrast, the more rural "TR" area shows minimal incidents. This urban-rural divide underscores the need for targeted crime prevention and resource allocation in the "BS" region.

School

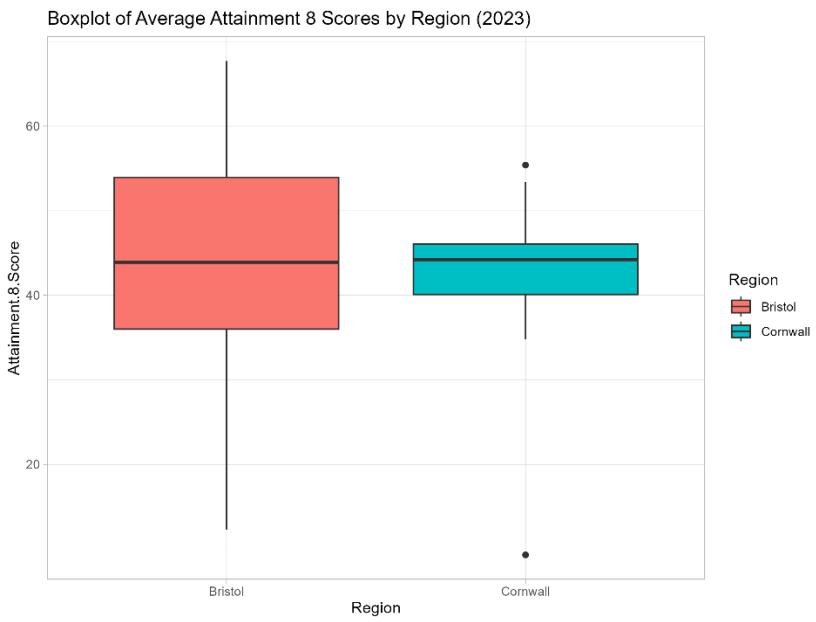


Figure 18: Boxplot of Average Attainment 8 Scores (2023)

The boxplot shows that Bristol's Average Attainment 8 Scores have a wider range and greater variability, indicating a mix of high and low-performing schools. Cornwall, on the other hand, has more consistent performance with a narrower range, although there is a notable outlier with significantly lower scores. Bristol's median score is slightly higher, reflecting diverse performance due to varying socio-economic factors, while Cornwall's schools exhibit more uniform outcomes.

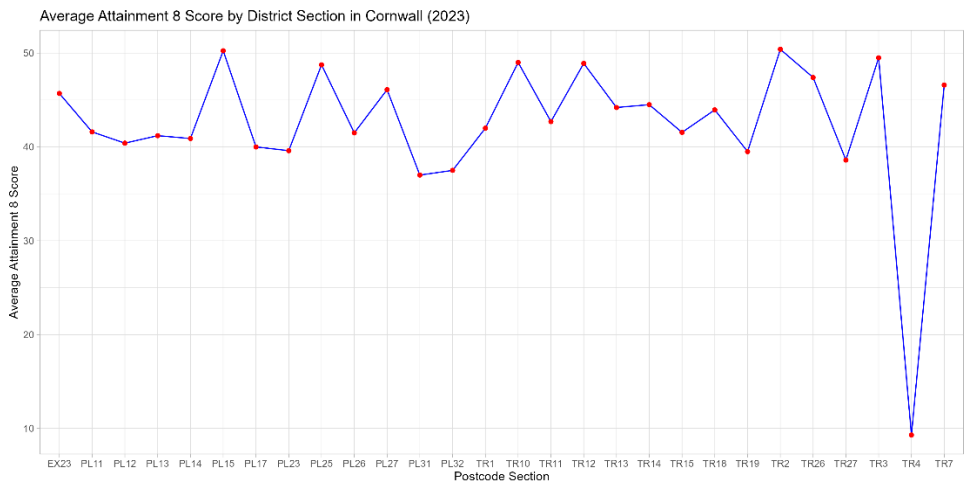
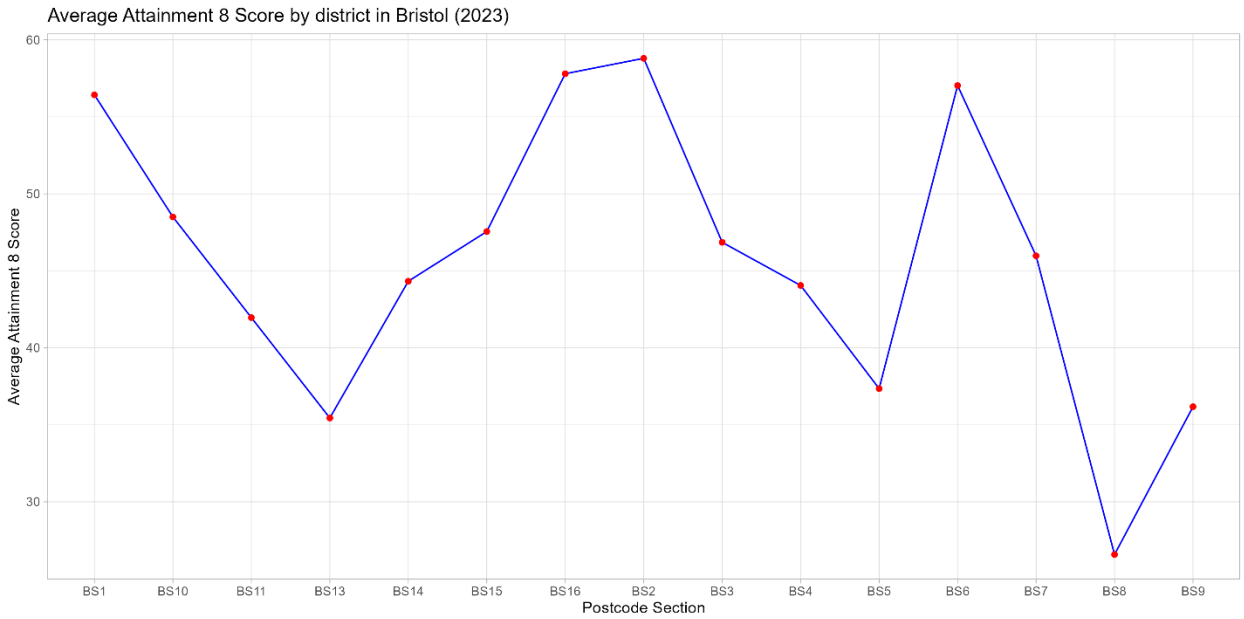


Figure 19: Line Graph for Attainment 8 score by district Cornwall(2023)

The line graph for Cornwall in 2023 shows significant variability in Average Attainment 8 Scores. Some districts, like PL12, PL25, and TR1, have strong scores above 50, indicating better outcomes. In contrast, districts such as TR4 experience sharp declines, possibly due to resource limitations or socio-economic challenges. Areas like PL11 and TR12 show mid-range consistency. Overall, the graph highlights a mix of high and low performers, underscoring the need to investigate localized factors influencing these disparities.



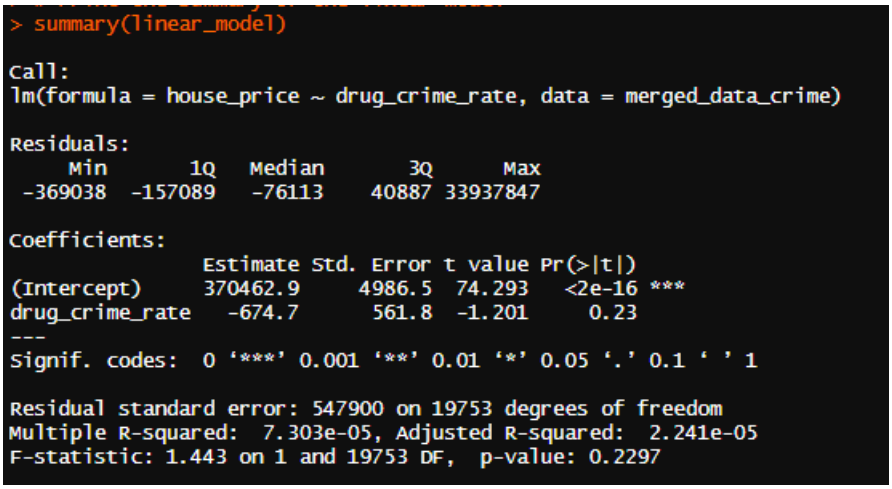
The graph shows variability in the Average Attainment 8 Scores across Bristol districts in 2023. BS1 and BS16 have the highest scores around 60, while BS13 and BS7 show the lowest, dipping to around 30. The pattern indicates significant disparities in educational attainment across districts, with noticeable peaks and troughs in 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 models the relationship between the variables Price and Average_download_speed. The model returns a positive, statistically significant relationship: for every 1 unit of download speed, the price would rise by 751 units. However, the extremely low R-squared value of 0.0023 indicates that download speed alone explains only a small fraction of variability in price. The model has an F-statistic = 46.84 and a p-value = 7.91e-12, so it is significant. However, the large ranges for the residuals indicate big prediction errors, so it is likely that other factors not included in this model are really driving 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 model of linear regression, the house price and drug_crime_rate are intercepting statistically significantly with p < 2e-16, which gives a baseline house price for when the rate of crime is zero. The coefficient for drug_crime_rate is -674.7, suggesting a decrease in house price with an increase in the rate of crime, though the effect is not statistically significant, p = 0.23. This is proved by the very low R-squared value of 7.303e-05, which conveys that drug crime rate explains close to none of the variation in house prices. Also, the F-statistic 1.443, p = 0.2297 confirms that the model does not improve the accuracy of prediction significantly. In general, drug crime rate seems to have very minimal effects on house prices, and further exploration into other factors was necessary.

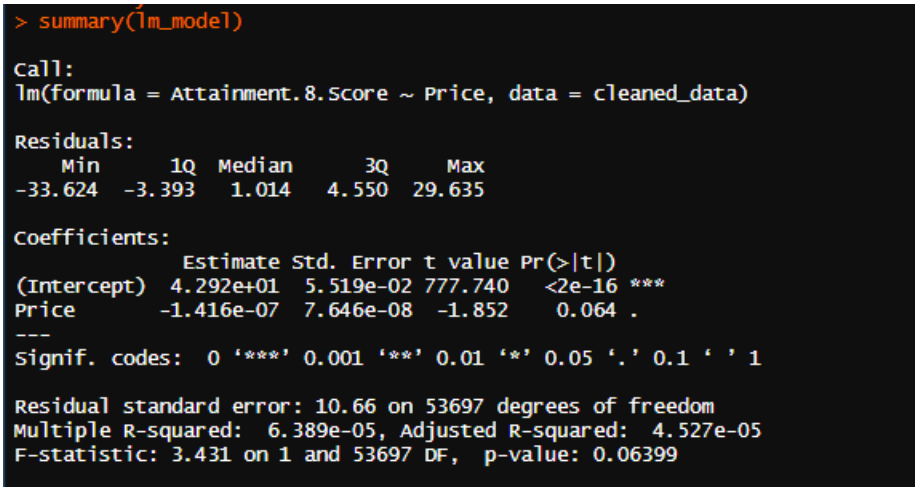


Figure 22: Attainment 8 score vs price

Such that the impact of Price on Attainment.8.Score was tested. The intercept is 42.92 ($p < 2e-16$), suggesting a baseline for Attainment.8.Score when Price is zero. The coefficient of Price of -1.416e-07 would suggest a very slight lowering of the score with increased Price. Though the p-value of 0.064 is marginally higher than the standard cut-point of 0.05, it still explains a marginal relationship and remains likely to be significant after further expansion in the study. The R-squared value of this model is very low at 6.389e−05: this means that Price explains Attainment.8. Score very poorly. The F-statistic is 3.431, which sounds good; however, the fact p is so small makes one think that probably other factors play a bigger role in affecting Attainment.8.

```
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 considers the effect of the drug offence rates with and without outliers on the download speed. The case, including the outliers, finds the model to have a small and close to zero relationship coefficient of -0.00003578; it is not statistically significant since p = 0.154. It shows a low R-squared value of 0.01315, meaning that only 1.3% of the variance in the download speed is explained. The model makes a drastic improvement when excluding any outliers: the coefficient is -0.005103, and the relationship is highly significant at the $p < 0.001$ level. The R-squared also increases to 0.1132, showing that 11.3% of the variance is explained.

Prove that an incorrect picture was given by outliers for the weak but statistically significant negative relationship between the drug offense rate and the download speed. Management of outliers becomes then an extremely important task 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

This analysis explains the relationship between the average Attainment 8 score and the average download speed using linear regression. So, there are some deviations from predictions as the residuals range from -35.793 to 57.207 Mbps with a median of -2.581 Mbps. The intercept is 23.9002, p = 0.0856; while that for Attainment_8_Avg is given as 0.5356, with p = 0.0996 which implies a weak positive association with broadband speed. However, even if these values are taken into account, it can be concluded that the variance of download speed explained by this model is only 11.35 % (R-squared value=0.1135) and its residual standard error is high at 22.82 suggesting that more factors need to be included.

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. To have everything as accurate as possible taking care of data gaps or inconsistencies is important. 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 in order 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.

Drive Link

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

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

```r
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

```r
# 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

```r
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

```r
# 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

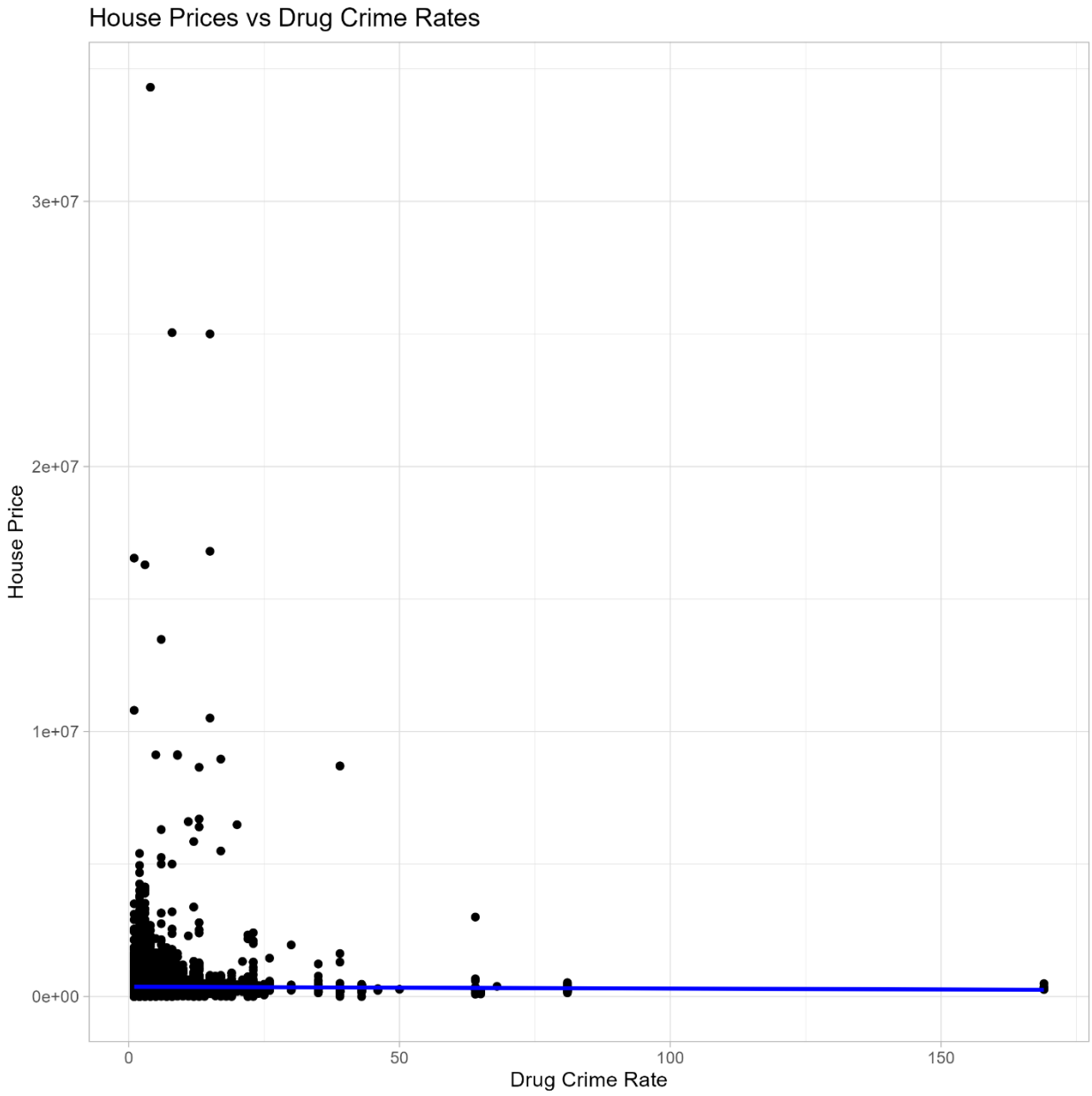


Figure 34: House price vs Drug crime rate scatterplot

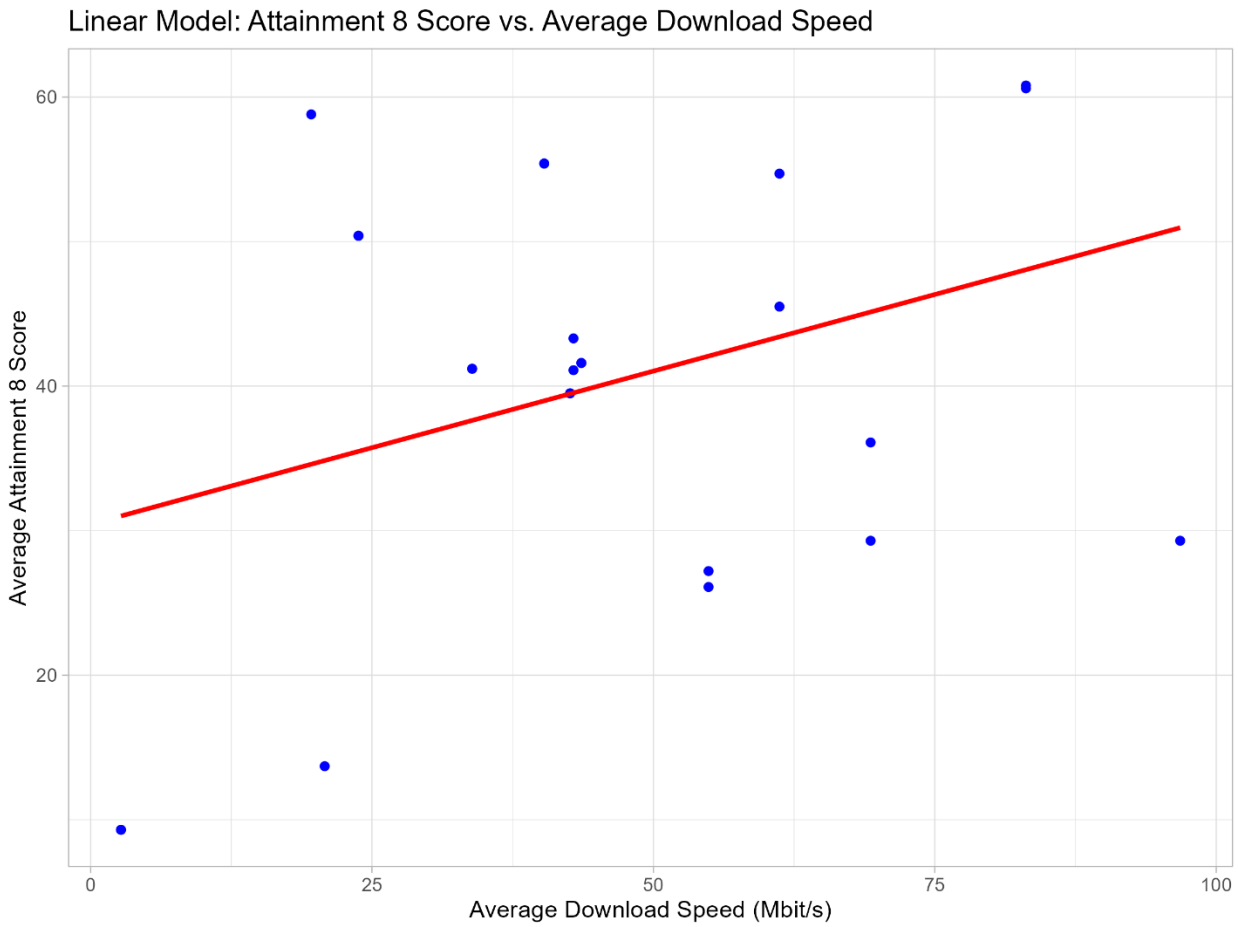


Figure 35: Average download speed vs average attainment 8 score scatterplot

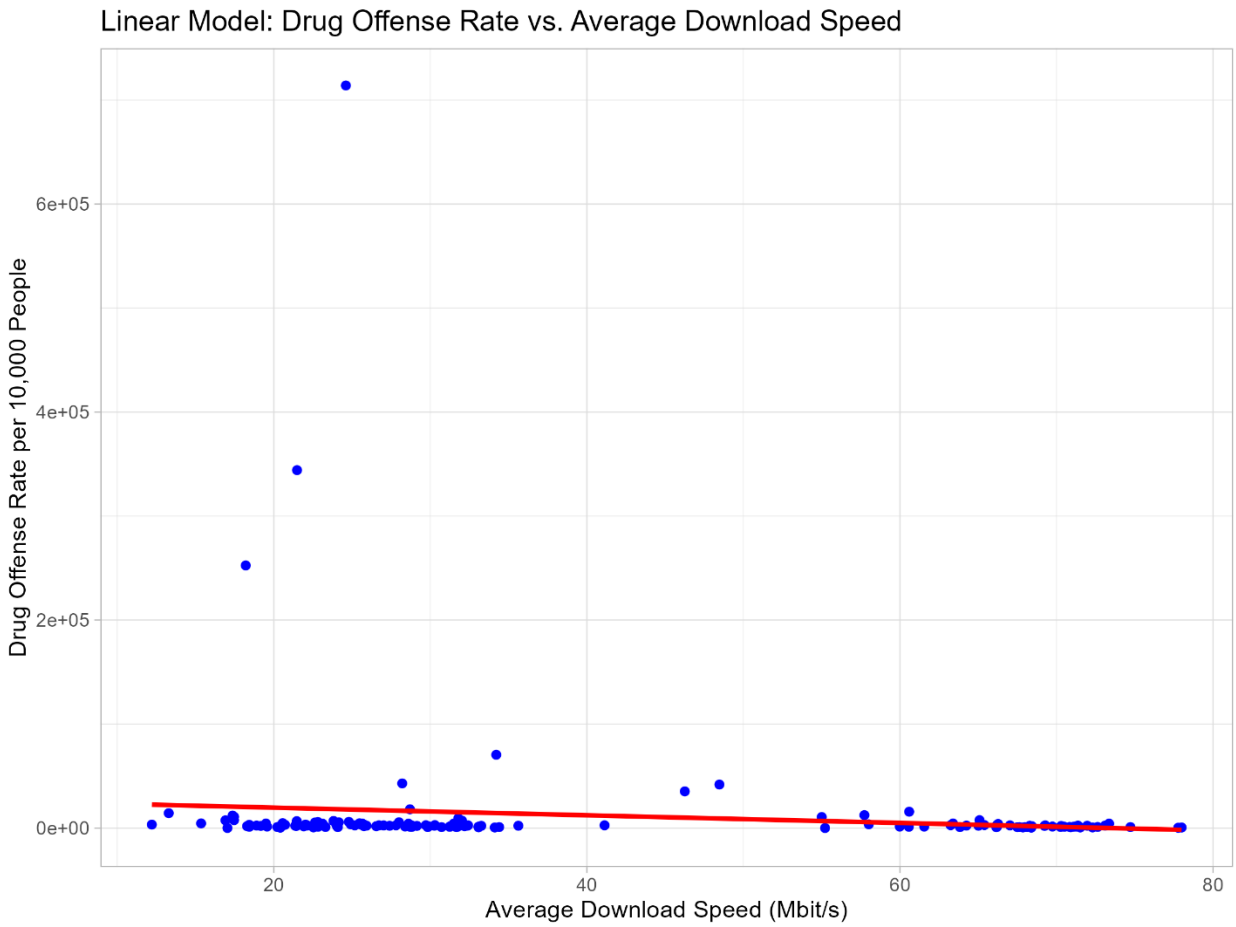


Figure 36: Average Download Speed vs Drug offense scatterplot

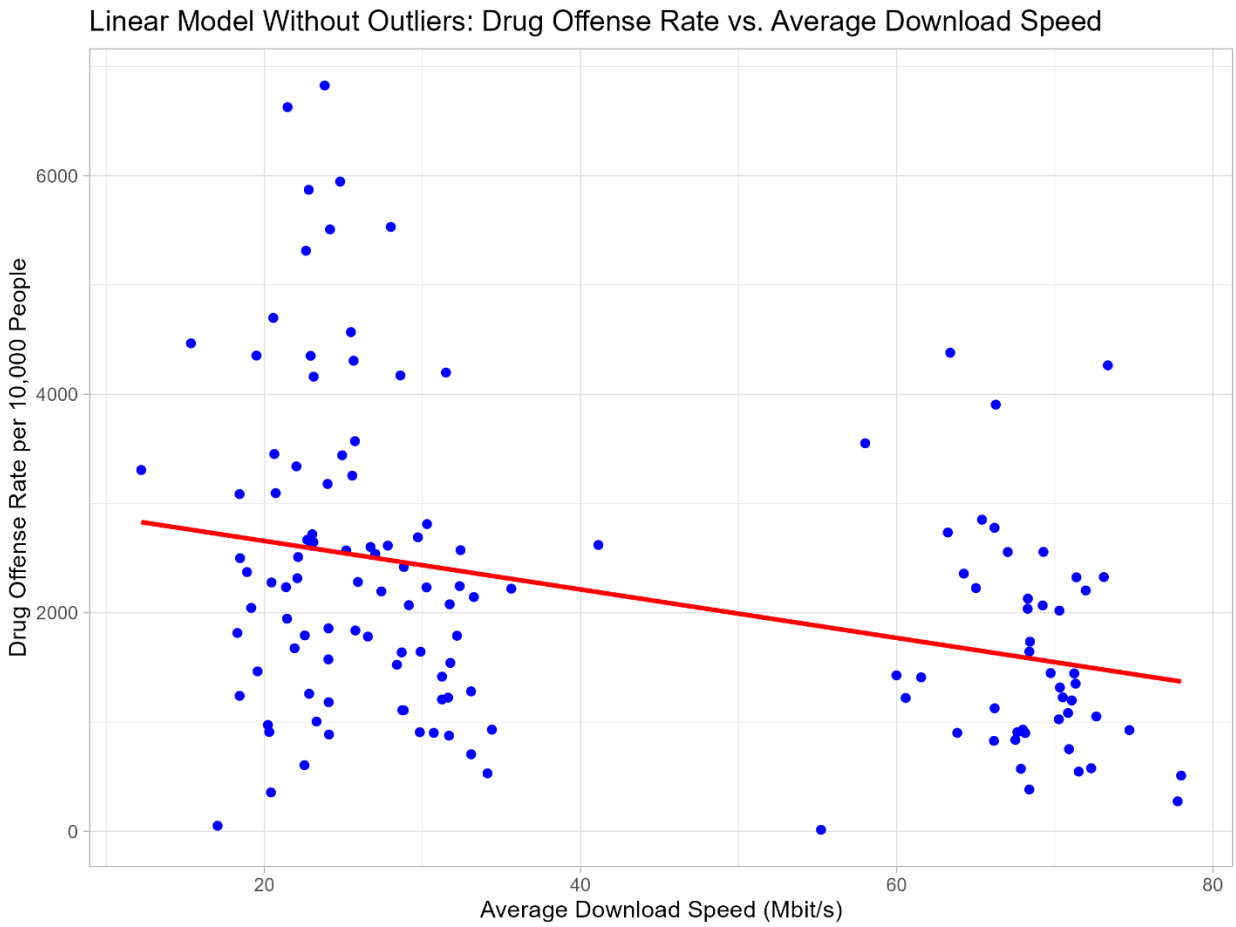


Figure 37: Average Download Speed vs Drug offense rate per 10000 people Scatterplot

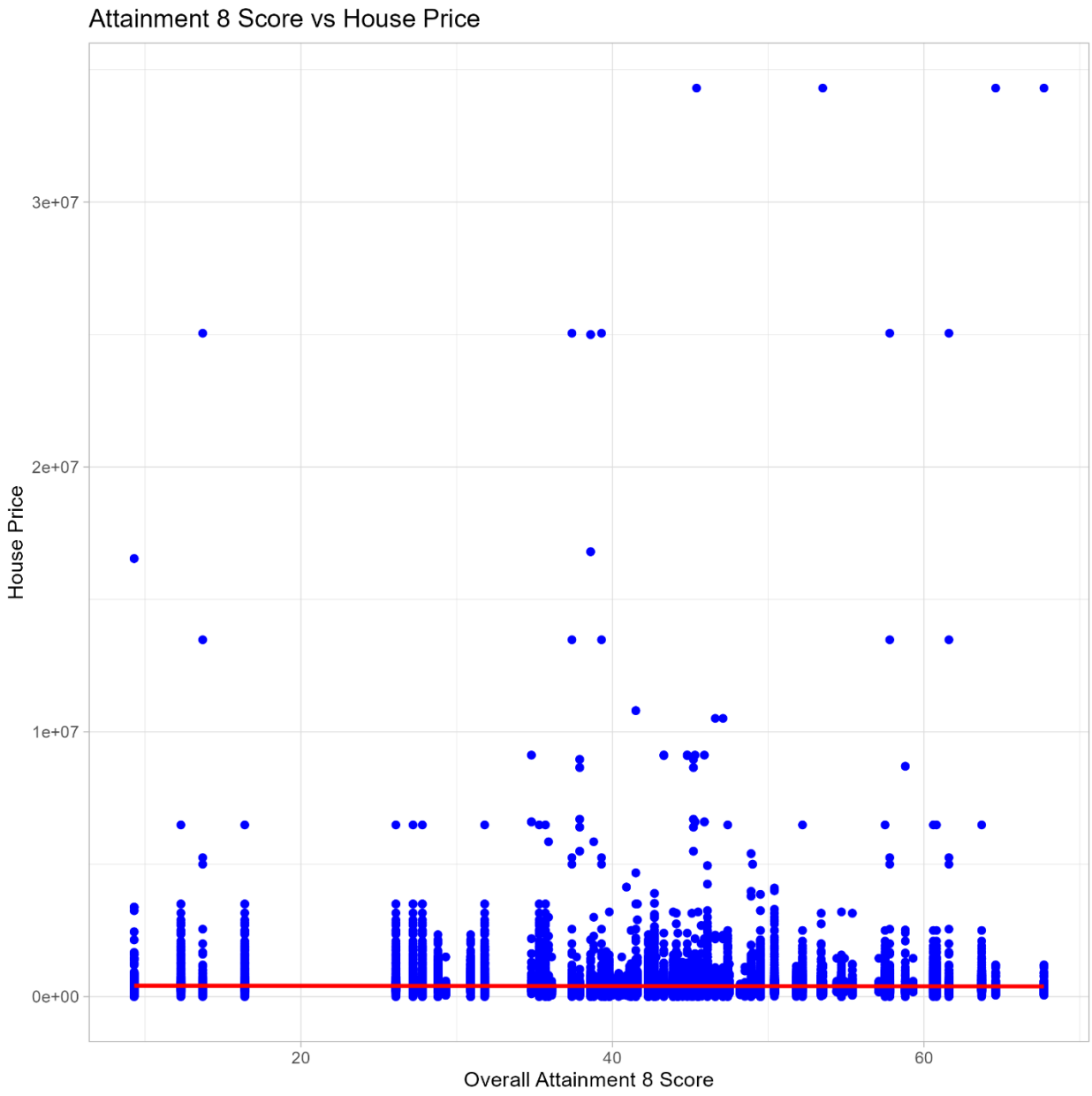


Figure 38: Attainment 8 vs house price scatterplot

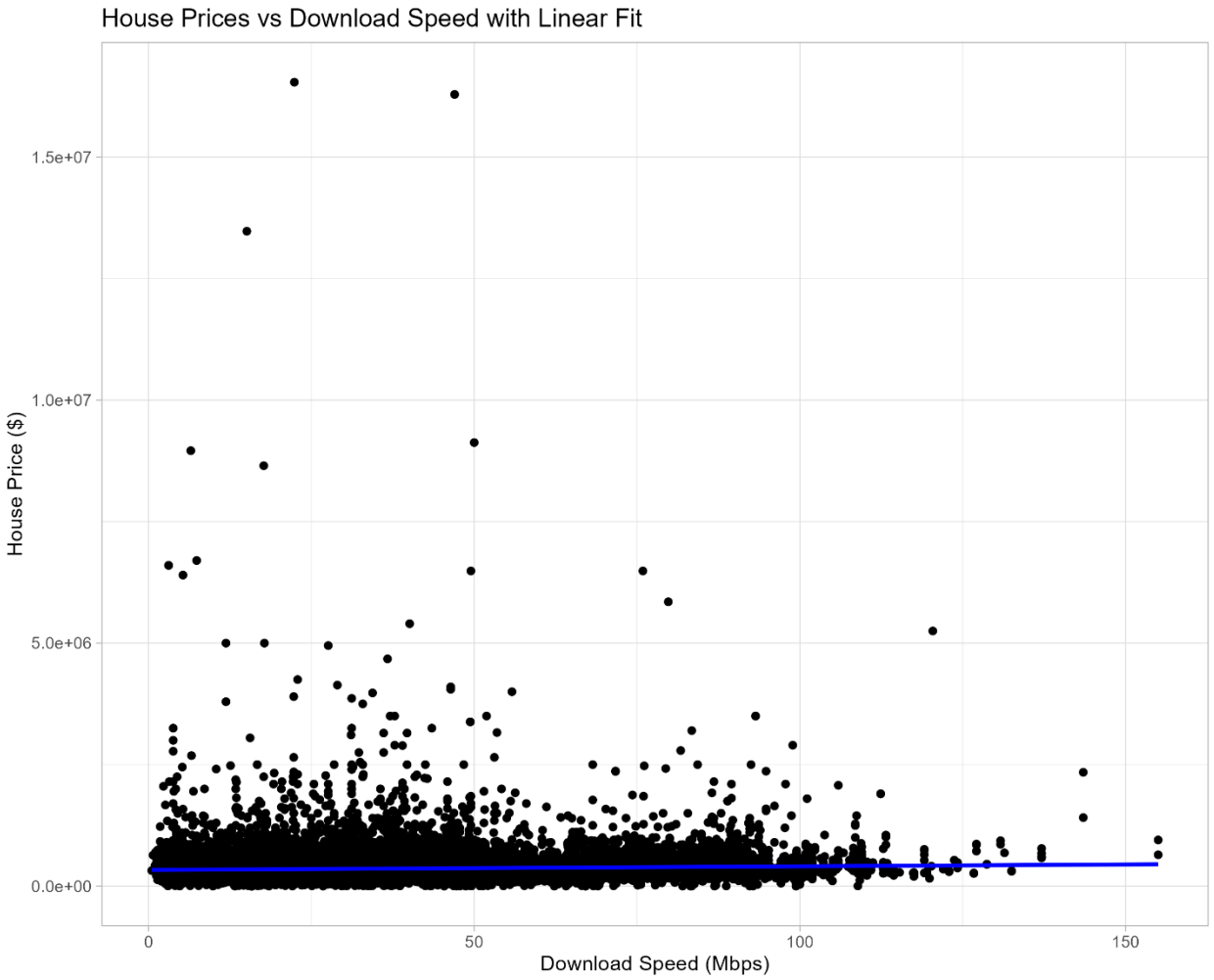


Figure 39: Download speed vs House price Scatterplot