Individual Report (Data Science for Developers)

BSc. (Hons.) Computing, Softwarica College of IT and E-commerce, University of Coventry

ST5014CEM (Data Science for Developers)

Rohan Chaulagain

19th August, 2024

GitHub link: https://github.com/0to1e/DS-Assignment.git

Table of Contents

Table of Figures

Introduction

This project involves analyzing datasets sourced from the UK government to generate data-driven recommendations for property investments in Bristol and Cornwall. Key factors such as housing prices, internet connectivity, and crime rates are examined. The datasets used include house price data from the housing and communities' section of the UK government website, crime statistics from Avon and Somerset Constabulary and Devon & Cornwall Police, school performance data from the UK government's "Compare school and college performance in England" section, and broadband data from OFCOM. Additionally, postcode-to-LSOA and population data based on postcodes were utilized, with the census data from 2011 being updated to reflect the 2023 population.

These datasets were deemed suitable as they were all publicly available and obtained from reliable sources, including official UK government websites, police websites, and OFCOM. This project provides an opportunity to enhance skills in data cleaning, normalization, exploratory data analysis, and statistical investigation. By developing a recommendation system that integrates various data attributes, practical experience in deriving actionable insights is gained. This process contributes to a deeper understanding of the data science life-cycle and improves the ability to make informed decisions based on complex datasets.

Data Cleaning

Before performing analytics and building the recommendation system, cleaning the data obtained from different sources is the foremost crucial process which involves identifying and correcting errors, inconsistencies, and missing values within a dataset. Cleaning the data supports ensuring data accuracy, consistency, and completeness, cleaning significantly enhances the reliability of subsequent analyses. This foundational step prevents misleading results, improves model performance, and ultimately empowers data-driven decision-making.

Talking about this task, datasets related to four entities, i.e. House Pricing data, Broadband Connection, Crime and Schools of UK has been provided to us to work on throughout the analysis process. Further discussion involves the cleaning process of each raw dataset obtained.

1. **House Pricing Data:**

Raw Dataset:

The house pricing data that was obtained wasn't usable at all when first received for following reasons:

i. No header name, i.e. unable to identify composition of datasets.
ii. Data separated into four different files yearwise, making it difficult to access the dataset.
iii. Presence of Null values.
iv. Presence of unwanted columns in the dataset.
v. Improper date format.

Cleaning Process:

i. Assigned standardized column names.
ii. Selected columns only which are necessary to perform the analysis.
I.e.(Price, Date_of_Transfer, Postal.Code, Property_Type, Town.City, District, County)
iii. Mutated date format appropriately.
iv. Removed null values present.

Final Outcome:

i. Only necessary columns in the cleaned dataset.

ii. No null values

iii. Appropriate datatypes handled

For detailed code implement, please refer to Appendix.

2. Broadband Data:

Raw Datasets:

The initial datasets obtained included:

i. Broadband Coverage Data: Contains information on broadband availability.

ii. Broadband Performance Data: Includes data on broadband performance metrics.

Issues Identified:

i. No Standardized Column Names: Column names were inconsistent or unclear.

ii. Irrelevant Columns: Some columns were not necessary for analysis.

iii. Inconsistent Data: Postcodes had inconsistent formatting and were not aligned with other datasets.

iv. Missing Values: Various columns had missing values that needed to be handled.

v. Lack of City and County Information: Needed to enrich data with additional geographic information.

Cleaning Process:

i. Load and Clean Broadband Coverage Data:

1. Columns Selected: Focused on postcode, SFBB availability (% premises), UFBB availability (% premises), and FTTP availability (% premises).
2. Postcode Formatting: Removed spaces from postcodes for consistency.
3. Renaming: Standardized the column name for postcode to PostalCode.

ii. Load and Clean Broadband Performance Data:

1. Columns Selected: Focused on postcode, Maximum download speed (Mbit/s), and Average download speed (Mbit/s).
2. Renaming: Standardized the column name for postcode to PostalCode.

iii. Load and Clean Postcode Data:

1. Postcode Data: Used to merge and enrich broadband datasets with city and county information.

iv. Merge Data with Postcode Information:

1. Broadband Coverage Data: Joined with postcode data to add city and county information. Assigned "NA" for unmatched entries.
2. Broadband Performance Data: Similarly joined with postcode data, handled missing values by imputing the median for download speeds.

Final Outcome:

i. Cleaned Datasets: Only relevant columns retained, and unnecessary ones removed.

ii. Consistent Formatting: Postcodes standardized and merged with geographic information.

iii. Handled Missing Values: Missing values addressed with median or mean imputation as appropriate.

For detailed code implementation, please refer to the Appendix.

3. Crime Data:

Raw Dataset:

i. Crime Data Files: Multiple CSV files containing crime data.

Issues Identified:

i. Multiple Files: Data split across several CSV files.

ii. Irrelevant Columns: Not all columns were necessary.

iii. Formatting Issues: Inconsistent date formats and missing county information.

Cleaning Process:

i. Combine CSV Files: Merged all CSV files into a single dataset.

ii. Select and Clean Data:

1. Retained relevant columns and formatted Month to date.
2. Filtered for crimes in Bristol or Cornwall.
3. Renamed columns and assigned County based on LSOA name.

Final Outcome:

iii. Consolidated Data: Combined multiple files into a single dataset.

iv. Relevant Columns: Only necessary columns retained.

v. Data Consistency: Dates formatted and county information assigned.

4. School Data:

Raw Datasets:

i. School Data Files: Key Stage 4 final data from Bristol and Cornwall for two academic years.

Issues Identified:

i. Multiple Files: Data spread across multiple files for different years and regions.

ii. Irrelevant Columns: Not all columns were necessary.

iii. Missing Values: Missing school names and values in key columns.

iv. Inconsistent Formatting: Postcode formatting needed standardization.

Cleaning Process:

i. Load and Prepare Data:
    1. Loaded datasets for Bristol and Cornwall for the academic years 2021-2022 and 2022-2023.
    2. Selected relevant columns and added a Year column to each dataset.

ii. Combine and Clean Data:
    1. Merged all datasets into one.
    2. Filtered out rows with missing school names.
    3. Converted and imputed missing values in ATT8SCR.
    4. Standardized PCODE, joined with postcode data, and handled missing city and county values.

Final Outcome:

i. Consolidated Data: Combined multiple datasets into a single, comprehensive dataset.

ii. Relevant Columns: Focused on necessary columns and handled missing values.

iii. Data Consistency: Postcode formatting standardized, city and county information added.

EDA

1. EDA of House Pricing:

i. Boxplot of House Prices in 2023 for CITY OF BRISTOL and CORNWALL

- Graph: Boxplot
- Datasets Used: Cleaned House Pricing Data
- X-axis: County – Categorical
- Y-axis: House Price – Quantitative
- Summary: This boxplot compares the distribution of house prices in Bristol and Cornwall for the year 2023, showcasing the variation and central tendency of house prices in these areas.
- Result:

ii. Total House Prices by Town within County (2023)

- Graph: Stacked Bar Chart:
- Datasets Used: Cleaned House Pricing Data
- X-axis: County – Categorical
- Y-axis: Total Price – Quantitative
- Summary: This stacked bar chart illustrates the total house prices by town within each county for the year 2023, allowing comparison of house price contributions from different towns within Bristol and Cornwall.

2. EDA of Broad band

i. Average Download Speed by County2020-2023

- Graph: Boxplot
- Datasets Used: Cleaned Broadband Performance Data
- X-axis: County - Categorical
- Y-axis: Average Download Speed (Mbit/s) – Quantitative
- Summary: This boxplot visualizes the distribution of average download speeds across different counties, providing insights into broadband performance variability within each county.

ii. Bristol: Maximum vs Average Download Speed by City

- Graph: Side-by-Side Bar Chart
- Datasets Used: Cleaned Broadband Performance Data
- X-axis: City – Categorical
- Y-axis: Download Speed (Mbit/s) – Quantitative
- Summary: This side-by-side bar chart compares maximum and average download speeds for different cities within Bristol, highlighting variations in broadband performance across these cities.

iii. Cornwall: Maximum vs Average Download Speed by City

- Graph: Side-by-Side Bar Chart
- Datasets Used: Cleaned Broadband Performance Data
- X-axis: City - Categorical
- Y-axis: Download Speed (Mbit/s) - Quantitative
- Summary: This side-by-side bar chart compares maximum and average download speeds for different cities within Cornwall, illustrating the variation in broadband performance across these cities.

3. EDA of Crime

i. Yearly Drug Offence Rate per 10000 Population (2020-2023)

- Graph: Line Graph
- Datasets Used: Cleaned Crime Data
- Related Datasets: Cleaned Population Data
- X-axis: Year – Quantitative
- Y-axis: Offence Rate per 10,000 Population - Quantitative
- Summary: This line graph visualizes the yearly drug offence rates per 10,000 population for Bristol and Cornwall from 2020 to 2023, highlighting trends and comparisons between these regions.

ii. Drug Offence Rate (2023)

- Graph: Boxplot
- Datasets Used: Cleaned Crime Data
- Related Datasets: Cleaned Population Data
- X-axis: County – Categorical
- Y-axis: Offence Rate - Quantitative
- Summary: This boxplot displays the distribution of drug offence rates in 2023 across different counties, highlighting variations in the rate of drug-related crimes.

iii. Robbery Offences Distribution (October 2022)

- Graph: Pie Chart
- Datasets Used: Cleaned Crime Data
- Related Datasets: Cleaned Population Data
- X-axis: N/A (Pie chart does not have an X-axis)
- Y-axis: Percentage - Quantitative
- Summary: This pie chart visualizes the distribution of robbery offences across different counties for October 2022, showing the proportion of total offences represented by each county.

4. EDA of School:

i. Boxplot of Attainment 8 score by County

- Graph: Boxplot
- Datasets Used: Cleaned Schools Data
- X-axis: County - Categorical
- Y-axis: ATT8SCR - Quantitative
- Summary: This boxplot displays the distribution of Attainment 8 Scores (ATT8SCR) across various counties for the year 2023, providing insight into the educational performance variations between counties.

ii. Average Attainment 8 score by City in Cornwall

- Graph: Line Graph
- Datasets Used: Cleaned Schools Data
- X-axis: Year - Quantitative
- Y-axis: Average Attainment 8 score - Quantitative
- Summary: This line graph visualizes the trend of average Attainment 8 scores by city within Cornwall over the years, highlighting how educational performance has changed over time in different cities.

iii. Average Attainment 8 score by City in Bristol

- Graph: Line Graph
- Datasets Used: Cleaned Schools Data
- X-axis: Year - Quantitative
- Y-axis: Average Attainment 8 score - Quantitative
- Summary: This line graph shows the trend of average Attainment 8 Scores by city within Bristol over the years, providing insights into the changes in educational performance across Bristol's cities.

Linear Modeling:

i. School ATT8 Scores and House Prices

- Independent variable: Price
- Dependent variable: ATT8SCR
- Equation: ATT8SCR = 37.12 + 0.000003557 * Price
- P-value : 0.257
- R squared: 0.0095
- RSD: 16.41
- Summary: The model's fit is poor as the R-squared value is very low (0.0095), indicating that Price explains only a small fraction of the variability in ATT8SCR. The residual standard error (16.41) is relatively high compared to the range of ATT8SCR values, suggesting substantial unexplained variation.

ii. Average Download Speed and Attainment 8 score

- Independent variable: Average download speed (Mbit/s)
- Dependent variable: ATT8SCR
- Equation: ATT8SCR = 43.10 - 0.0471 * Average download speed (Mbit/s)
- P-value : <2e-16
- R squared: 0.0104
- RSD: 12.6
- Summary: The model's fit is poor as the R-squared value is very low (0.0104), indicating that Average download speed explains only a minimal fraction of the variability in ATT8SCR. Despite the statistically significant coefficient, the residual standard error (12.6) is relatively high, suggesting considerable unexplained variation in ATT8SCR.

iii. Average Download Speed and Drug Offence Rates per 10000 people

- Independent variable: Average download speed (Mbit/s)
- Dependent variable: offence_rate
- Equation: offence_rate = 0.0345 - 0.00001973 * Average download speed (Mbit/s)
- R squared variable: 0.0000672
- RSD: 0.06649
- P-value: 0.065
- Summary: The model's fit is very poor as the R-squared value is extremely low (0.0000672), indicating that Average download speed explains only an almost negligible fraction of the variability in offence_rate. The residual standard error (0.06649) suggests significant unexplained variation, and the p-value (0.065) indicates that the coefficient is marginally significant.

iv. House Price vs Average Download Speed

- Independent variable: Average download speed (Mbit/s)
- Dependent variable: Price
- Equation: Price = 347015.29 + 489.17 * Average download speed (Mbit/s)
- R squared: 0.0005203
- RSD: 576300
- P-value: 5.2e-09
- Summary: The model's fit is very poor as the R-squared value is extremely low (0.0005203), indicating that Average download speed explains only a negligible fraction of the variability in Price. Although the coefficient is statistically significant (p-value < 2e-16), the residual standard error (576300) is quite large, suggesting substantial unexplained variation in Price.

v. House Price vs Drug Offence Rate (2023)

- Independent variable: Offence rate (per 10,000 population)
- Dependent variable: House Price
- Equation: Price = 274,700 - 43,240,000,000 * Offence rate (per 10,000 population)
- P-value: < 2e-16
- R squared: 0.0135
- RSD: 1,309,000
- Summary: The R-squared value of 0.0135 indicates that the offence rate explains only a small fraction (1.35%) of the variability in house prices. Despite the coefficient for the offence rate being statistically significant (with a p-value less than 0.001), the model's fit is quite limited. The residual standard deviation (RSD) of 1,309,000 suggests considerable unexplained variation in house prices, highlighting that the offence rate may not be a strong predictor of house prices.

In comparing the models, none demonstrate a strong fit to their respective data. The models for School ATT8 Scores and Average Download Speed have low R-squared values, indicating minimal explanatory power for Price and ATT8SCR, with substantial unexplained variation. The model relating Average Download Speed to Drug Offence Rates shows an almost negligible R-squared, suggesting that average download speed has little impact on offence rates. The model for House Price versus Average Download Speed and Drug Offence Rate both shows very low R-squared values, with the latter having a high residual standard deviation, reflecting significant unexplained variability. Overall, these results highlight that neither average download speed nor drug offence rates are strong predictors of house prices or educational attainment, emphasizing the need for alternative explanatory variables or models.

Ranking:

Overview

The process used to rank cities within each county based on four criteria: school performance, house pricing, broadband availability, and crime rates. The process involves several steps, including data loading, ranking within individual categories, and combining the results to produce a final ranking of cities.

1. Data Loading
1.1. School Rankings: The school_dataset was loaded from the file Cleaned_datasets/Schools/Schools_Clean.csv. This dataset contains information on school performance metrics.

House Pricing

1.2. House Pricing: The house_pricing_dataset was loaded from Cleaned_datasets/House_Pricing_Data/clean_house_pricing_data.csv. It includes data on house prices across different towns and cities.

Broad band

1.3. Broadband Coverage and Performance:

Broadband data was sourced from two files:

Cleaned_datasets/Broadband/clean_broadband_coverage.csv (coverage data)
Cleaned_datasets/Broadband/clean_broadband_performance.csv (performance data)

Crime

1.4. Crime Data:The crime data was loaded from Cleaned_datasets/Crime/Crime_Data_Combined.csv.

This dataset includes information on various crime types and their frequencies.

2. Ranking Cities
2.1. Ranking Schools

Objective: Rank schools based on their ATT8SCR (Attainment 8 score).

Method:

- Grouped data by County.
- Arranged schools in descending order of ATT8SCR.
- Assigned ranks to schools using row_number().
- Filtered to include only the top 10 schools per county.

Results

2.2. Ranking House Pricing

Objective: Rank towns and cities based on a calculated score from house pricing data.

Method:

- Calculated percentiles (p25, p50, p75) and skewness for house prices.
- Computed a score using a weighted formula combining these percentiles and skewness.
- Ranked cities within each county based on the score.
- Filtered to include only the top 10 cities per county.

2.3. Ranking Broadband

Objective: Rank cities based on the weighted average score of various broadband metrics.

Method:

- Merged broadband coverage and performance datasets.
- Normalized the data for several broadband metrics.
- Calculated a weighted score for each city based on normalized metrics.
- Aggregated the score at the city level.
- Ranked cities within each county based on the aggregated score.
- Filtered to include only the top 10 cities per county.

2.4. Ranking Crime

Objective: Rank cities based on the total number of crime offences.

Method:

- Aggregated total offences for each crime type.
- Calculated the total number of offences for each city.
- Ranked cities within each county based on total crime offences.
- Filtered to include only the top 10 cities with the fewest total crime offences.

3. Combining Rankings

Objective: Create an overall ranking by combining rankings from all categories.

Method:

- Merged all individual rankings using full_join to include all cities, even if they are missing from some rankings.
- Replaced NA values with a high rank to ensure they are ranked lower.
- Calculated the average rank across all categories for each city.
- Sorted cities within each county based on the average rank.
- Selected the top 3 cities per county based on the overall average rank.

Conclusion: This methodology allows for a comprehensive ranking of cities within each county based on multiple factors, providing a holistic view of city performance across schools, housing, broadband, and crime.

Reflection

This project underscored the importance of a structured approach to data analysis. Starting with thorough data cleaning ensured that the analysis was based on reliable data. EDA offered critical insights and guided the modeling process. Linear modeling provided a quantitative understanding of factors influencing city rankings. Finally, combining and ranking the results offered a holistic view of city performance.

The iterative nature of the project—from cleaning to ranking—demonstrates the complexity and interconnectedness of data analysis tasks. Each phase was crucial in building towards a robust final product. The experience highlighted the value of an integrated approach to data science, where each step builds upon the previous one, culminating in actionable insights and informed decision-making.

Overall, this project was a valuable exercise in applying data science methodologies to real-world problems, reinforcing the importance of careful planning, rigorous analysis, and thoughtful interpretation.

Overall Score:

Based on all the calculations below, here is the table of final ranking of the cities countywise.

(row)	County	City
1	Bristol	BRISTOL
2	Bristol	Bristol 040
3	Bristol	Bristol 060E
4	Cornwall	Cornwall 005
5	Cornwall	Cornwall 040F
6	Cornwall	Cornwall 068

Figure 1: Final Ranking

The data used in this project was sourced entirely from open-access and legally compliant channels, ensuring adherence to all ethical and legal standards. The datasets were obtained from reputable and official sources, including the UK government's housing and community's portal, Avon and Somerset Constabulary, Devon & Cornwall Police, the UK government's school performance database, and OFCOM for broadband information. Additionally, postcode-to-LSOA and census data were leveraged, with updates made to reflect current population estimates. All sources are publicly available and are provided under transparent data-sharing agreements, eliminating any concerns regarding data privacy or unauthorized use. By relying on these legitimate and ethical data sources, the project not only adheres to legal requirements but also upholds high standards of integrity and transparency in data handling and analysis. This approach reinforces the credibility of the analysis and ensures that the insights generated are both reliable and compliant with ethical norms.

Conclusion:

This project vividly illustrated the significance of a methodical approach to data analysis. Commencing with meticulous data cleaning ensured that our analysis was grounded in accurate and reliable data. Exploratory Data Analysis (EDA) played a pivotal role in uncovering critical insights and steering the subsequent modeling efforts. Linear modeling then offered a quantitative perspective on the factors impacting city rankings, enhancing our understanding of their relative importance.

The iterative process—from cleaning through to final ranking—highlighted the complexity and interdependency inherent in data analysis tasks. Each phase, from initial data preparation to the synthesis of final rankings, was integral to achieving a comprehensive and insightful final product. This project underscored the value of an integrated data science approach, where each step builds upon the last, leading to actionable insights and well-informed decision-making.

In conclusion, this project served as a practical application of data science methodologies to real-world issues, reaffirming the importance of careful planning, rigorous analysis, and thoughtful interpretation.

References

1. Dataquest. (2020, July 24). Loading and cleaning data with R and the tidyverse. Retrieved from https://www.dataquest.io/blog/load-clean-data-r-tidyverse/
2. Lamethods. (n.d.). An R approach to data cleaning and wrangling for education. Retrieved from https://lamethods.org/chapters/ch04-data-cleaning/ch4-datacleaning.html
3. Wickham, H., & Grolemund, G. (n.d.). Exploratory data analysis. In R for data science. Retrieved from https://r4ds.had.co.nz/exploratory-data-analysis.html
4. Wickham, H. (n.d.). Tidy data. In tidyr. Retrieved from https://tidyr.tidyverse.org/articles/tidy-data.html
5. Chang, W. (n.d.). R graphics cookbook. Retrieved from https://www.cookbook-r.com/Graphs/

Appendix

Figure 2: cleaning code for house pricing

```r
# Defining column names as orignal data does'nt contain any
column_names <- c("Transaction_ID", "Price", "Date_of_Transfer", "Postal Code", "Property_Type", "Old/New", "Duration", "PAON", "SAON", "Street", "Locality", "Town/City", "District", "County", "PPD_Category_Type", "Record_Status")

postcode_data <- read_csv("Cleaned_datasets/Postcode_clean.csv") %>%
  select(pcd7, Street)
# Read CSV files for house price data from 2020 to 2023 , each file containing a year's data
data0 <- read.csv("Obtained_Data/House_Pricing/pp-2020.csv", header = FALSE, col.names = column_names)
data1 <- read.csv("Obtained_Data/House_Pricing/pp-2021.csv", header = FALSE, col.names = column_names)
data2 <- read.csv("Obtained_Data/House_Pricing/pp-2022.csv", header = FALSE, col.names = column_names)
data3 <- read.csv("Obtained_Data/House_Pricing/pp-2023.csv", header = FALSE, col.names = column_names)

# Combine the datasets into one data frame
combined_data <- bind_rows(data0, data1, data2, data3)

cleaned_house_data <- combined_data %>%
  # Select only the relevant columns
  select(Price, Date_of_Transfer, Postal.Code, Town.City, County) %>%
  mutate(County = ifelse(County == "CITY OF BRISTOL", "Bristol", County)) %>%
  mutate(County = ifelse(County == "CORNWALL", "Cornwall", County)) %>%
  rename("PostalCode" = "Postal.Code", "Date" = "Date_of_Transfer") %>%
  mutate(PostalCode = gsub(" ", "", PostalCode)) %>%
  # Filter for Bristol and Cornwall
  filter((County %in% c("Bristol", "Cornwall"))) %>%
  # Convert Date_of_Transfer to Date type
  mutate(Date = as.Date(Date))

#join post_code datase with combined house dataset to add street location data
final_house_data <- merge(
  cleaned_house_data,
  postcode_data,
  by.x = "PostalCode",
  by.y = "pcd7",
  all.x = TRUE,  # Keeps all rows from cleaned_house_data
  all.y = FALSE  # Excludes rows from postcode_data that do not match
) %>%
  na.omit()


write_csv(final_house_data, file = paste0(getwd(), "/Cleaned_datasets/House_Pricing_Data/clean_house_pricing_data.csv"))
```

Figure 3: Cleaning code for broadband

```r
# Load and clean the broadband coverage data
broadband_coverage <- read_csv("Obtained_Data/broadband/201809_fixed_pc_coverage_r01.csv") %>%
  # Select only relevant columns related to broadband coverage
  select(postcode, `SFBB availability (% premises)`, `UFBB availability (% premises)`, `FTTP availability (% premises)`) %>%
  # Remove spaces from postcodes to ensure consistency
  mutate(postcode = gsub(" ", "", postcode)) %>%
  # Rename the postcode column to match other datasets
  rename("PostalCode" = postcode)

# Load and clean the broadband performance data
broadband_perfomance <- read_csv("Obtained_Data/broadband/201805_fixed_pc_performance_r03.csv") %>%
  # Select only relevant columns related to broadband performance
  select(postcode, `Maximum download speed (Mbit/s)`, `Average download speed (Mbit/s)`) %>%
  # Rename the postcode column to match other datasets
  rename("PostalCode" = postcode)

#Load postcode data to merge with street and County locations based on the postcode data available
postcode_clean <- read_csv("Cleaned_datasets/Postcode_clean.csv")

# Step 3: Join the broadband coverage data with postcode data to add city and county information
clean_coverage <- broadband_coverage %>%
  inner_join(postcode_clean, by = c("PostalCode" = "pcd7")) %>%
  # Handle any unmatched postcodes by assigning "NA" to City and County
  mutate(City = ifelse(is.na(City), "NA", City), County = ifelse(is.na(County), "NA", County))

# Join the broadband performance data with postcode data to add city and county information
clean_performance <- broadband_perfomance %>%
  inner_join(postcode_clean, by = c("PostalCode" = "pcd7")) %>%
  mutate(City = ifelse(is.na(City), "NA", City), County = ifelse(is.na(County), "NA", County)) %>%
  # Handle missing values in the Maximum download speed column by imputing the median
  mutate(`Maximum download speed (Mbit/s)` = ifelse(is.na(`Maximum download speed (Mbit/s)`), median(`Maximum download speed (Mbit/s)`, na.rm = TRUE), `Maximum download speed (Mbit/s)`)) %>%
  # Handle missing values in the Average download speed column by imputing the mean
  mutate(`Average download speed (Mbit/s)` = ifelse(is.na(`Average download speed (Mbit/s)`), mean(`Average download speed (Mbit/s)`, na.rm = TRUE), `Average download speed (Mbit/s)`))

# Save the cleaned broadband coverage data to a CSV file
write_csv(clean_coverage, file = paste0(getwd(), "/Cleaned_datasets/Broadband/clean_broadband_coverage.csv"))

# Save the cleaned broadband performance data to a CSV file
write_csv(clean_performance, file = paste0(getwd(), "/Cleaned_datasets/Broadband/clean_broadband_performance.csv"))
```

Figure 4: Cleaning code for crime

```r
# Define the directory containing the crime data files
crime_data_dir <- "Obtained_Data/Crime"
# List all CSV files in the directory recursively
all_files <- list.files(crime_data_dir, recursive = TRUE, full.names = TRUE, pattern = "\\.csv$")

# Function to read and combine multiple CSV files into one data frame
combine_csv_files <- function(files) {
  map_df(files, read_csv, .id = "file_id")
}

# Combine all crime data files into one dataset
combined_data <- combine_csv_files(all_files) %>%
  # Select relevant columns for crime analysis
  select(`Crime ID`, `Month`, `Falls within`, `LSOA code`, `LSOA name`, `Crime type`)

# Clean the crime data
clean_data <- combined_data %>%
  # Replace Crime ID with a unique identifier
  mutate(`Crime ID` = row_number(), Month = ymd(paste0(Month, "-01"))) %>%
  # Filter data to include only crimes in Bristol or Cornwall
  filter(str_detect(`LSOA name`, "Bristol|Cornwall")) %>%
  # Rename the Month column to date and assign County based on LSOA name
  rename(date = Month) %>%
  mutate(
    `Falls within` = ifelse(
      startsWith(`LSOA name`, "Bristol"),
      "Bristol",
      "Cornwall"
    )
  ) %>%
  rename("County" = "Falls within", lsoa_code = `LSOA code`, City = `LSOA name`, Crime = `Crime type`)

# Save the cleaned crime data to a CSV file
write_csv(clean_data, file = paste0(getwd(), "/Cleaned_datasets/Crime/Crime_Data_Combined.csv"))
```

Figure 5: Cleaning code for school

```r
# Load postcode dataset to enable joining tables and fetching street location information
postcode_dataset <- read_csv("Cleaned_datasets/Postcode_clean.csv")

# Load the 2021-2022 Bristol Key Stage 4 final data
bristol_key_stage_4_final_2021_2022 <- read_csv("Obtained_Data/schools/Bristol/2021-2022/801_ks4final.csv") %>%
  # Select relevant columns
  select(SCHNAME, PCODE, TOWN, ATT8SCR) %>%
  # Add a Year column indicating the year of the data
  mutate(Year = 2022)

# Load the 2022-2023 Bristol Key Stage 4 final data
bristol_key_stage_4_final_2022_2023 <- read_csv("Obtained_Data/schools/Bristol/2022-2023/801_ks4final.csv") %>%
  # Select relevant columns
  select(SCHNAME, PCODE, TOWN, ATT8SCR) %>%
  # Add a Year column indicating the year of the data
  mutate(Year = 2023)

# Load the 2021-2022 Cornwall Key Stage 4 final data
cornwall_key_stage_4_final_2021_2022 <- read_csv("Obtained_Data/schools/Cornwall/2021-2022/908_ks4final.csv") %>%
  # Select relevant columns
  select(SCHNAME, PCODE, TOWN, ATT8SCR) %>%
  # Add a Year column indicating the year of the data
  mutate(Year = 2022)

# Load the 2022-2023 Cornwall Key Stage 4 final data
cornwall_key_stage_4_final_2022_2023 <- read_csv("Obtained_Data/schools/Cornwall/2022-2023/908_ks4final.csv") %>%
  # Select relevant columns
  select(SCHNAME, PCODE, TOWN, ATT8SCR) %>%
  # Add a Year column indicating the year of the data
  mutate(Year = 2023)

# Combine all the school data into one dataset
combined_ks4_data <- bind_rows(
  bristol_key_stage_4_final_2021_2022,
  bristol_key_stage_4_final_2022_2023,
  cornwall_key_stage_4_final_2021_2022,
  cornwall_key_stage_4_final_2022_2023
)

# Filter out rows with missing school names (NA values)
clean_school_data <- combined_ks4_data %>%
  filter(!is.na(SCHNAME)) %>% # Filter for rows where SCHNAME is not NA
  # Convert ATT8SCR column to numeric (assuming it's a character vector)
  mutate(ATT8SCR = as.numeric(ATT8SCR)) %>% # Convert ATT8SCR to numeric
  # Impute missing values in ATT8SCR with the mean (excluding NAs)
  mutate(ATT8SCR = ifelse(is.na(ATT8SCR), mean(ATT8SCR, na.rm = TRUE), ATT8SCR)) %>% # Replace NA with mean
  # Remove spaces from PCODE column
  mutate(PCODE = gsub(" ", "", PCODE)) %>% # Remove spaces in PCODE using gsub
  # Inner join with postcode_clean data based on PCODE and pcd7
  inner_join(postcode_dataset, by = c("PCODE" = "pcd7")) %>% # Join with postcode data
  # Rename PCODE column to PostalCode
  rename("PostalCode" = "PCODE") %>% # Rename PCODE to PostalCode
  # Impute missing City and County with "NA"
  mutate(City = ifelse(is.na(City), "NA", City), County = ifelse(is.na(County), "NA", County)) %>% # Replace NA with "NA"
  # Select all columns except TOWN
  select(-TOWN)

# Save the cleaned school data to a CSV file
write_csv(clean_school_data, file = paste0(getwd(), "/Cleaned_datasets/Schools/school_cleaned_data.csv"))

```

Figure 6: cleaning post code

```r
#Read the CSV file containing postcode to LSOA mapping and select relevant columns
postcode_clean <- read_csv("Obtained_Data/Postcode to LSOA.csv") %>%
  # Select only the necessary columns from the dataset
  select(pcd7, oa11cd, lsoa11nm, msoa11nm, ladnm) %>%
  # Rename the columns for clarity
  rename(County = ladnm, City = msoa11nm, Street = lsoa11nm) %>%
  # Remove spaces from the 'pcd7' column values for consistency
  mutate(pcd7 = gsub(" ", "", pcd7)) %>%
  # Standardize the 'County' column to ensure uniform naming
  mutate(County = ifelse(County == "Bristol, City of", "Bristol", County)) %>%
  # Filter the data to include only rows where the 'County' is either 'Bristol' or 'Cornwall'
  filter((County %in% c("Bristol", "Cornwall")))

# Write the cleaned data to a new CSV file in the specified directory
write_csv(postcode_clean, file = paste0(getwd(), "/Cleaned_datasets/Postcode_clean.csv"))
```

Figure 7: Cleaning Population dataset

```r
# Read the population data from the CSV file
population_dataset <- read_csv("Obtained_Data/Population2011_1656567141570.csv") %>%

  # Extract the first two characters from the 'Postcode' column to match the format in the postcode dataset
  mutate(Postcode = substr(Postcode, start = 1, stop = 2))

# Read the cleaned postcode data and select relevant columns
postcode_dataset <- read_csv("Cleaned_datasets/Postcode_clean.csv") %>%

  # Select only the columns 'pcd7' (postcode) and 'County'
  select(pcd7, County) %>%

  # Extract the first two characters from the 'pcd7' column to match the format in the population dataset
  mutate(pcd7 = substr(pcd7, start = 1, stop = 2)) %>%

  # Remove duplicate rows based on 'pcd7' and 'County'
  distinct()

# Merge the population data with the postcode data and calculate the total population by county
population_2011 <- population_dataset %>%

  # Perform a left join to combine the population data with postcode data based on matching 'Postcode' and 'pcd7'
  left_join(postcode_dataset, by = c("Postcode" = "pcd7")) %>%

  # Group the data by 'County'
  group_by(County) %>%

  # Summarize the data by calculating the total population for each county
  summarize(total_population = sum(Population, na.rm = TRUE))

# Write the cleaned and summarized population data to a new CSV file
write_csv(population_2011, file = paste0(getwd(), "/Cleaned_datasets/Population_clean.csv"))
```

Figure 8: Code to generate House Price Boxplot

```r
# Load the cleaned house price dataset
house_dataset <- read_csv("Cleaned_datasets/House_Pricing_Data/clean_house_pricing_data.csv")

# Filter the dataset for transactions that occurred in 2023
house_data_2023 <- subset(house_dataset, format(as.Date(Date_of_Transfer), "%Y") == "2023")

# Generate a boxplot comparing house prices in Bristol and Cornwall for 2023
ggplot(house_data_2023, aes(x = County, y = log(Price), fill = County)) +
  geom_boxplot() +
  labs(
    title = "Boxplot of House Prices in 2023 for CITY OF BRISTOL and CORNWALL",
    x = "County",
    y = "House Price"
  )

# Save the boxplot as a PNG file
ggsave(paste0(getwd(), "/Graphs/House_Pricing/BoxP_BL_CL_2023.png"))
```

Figure 9: House Price BoxPlot

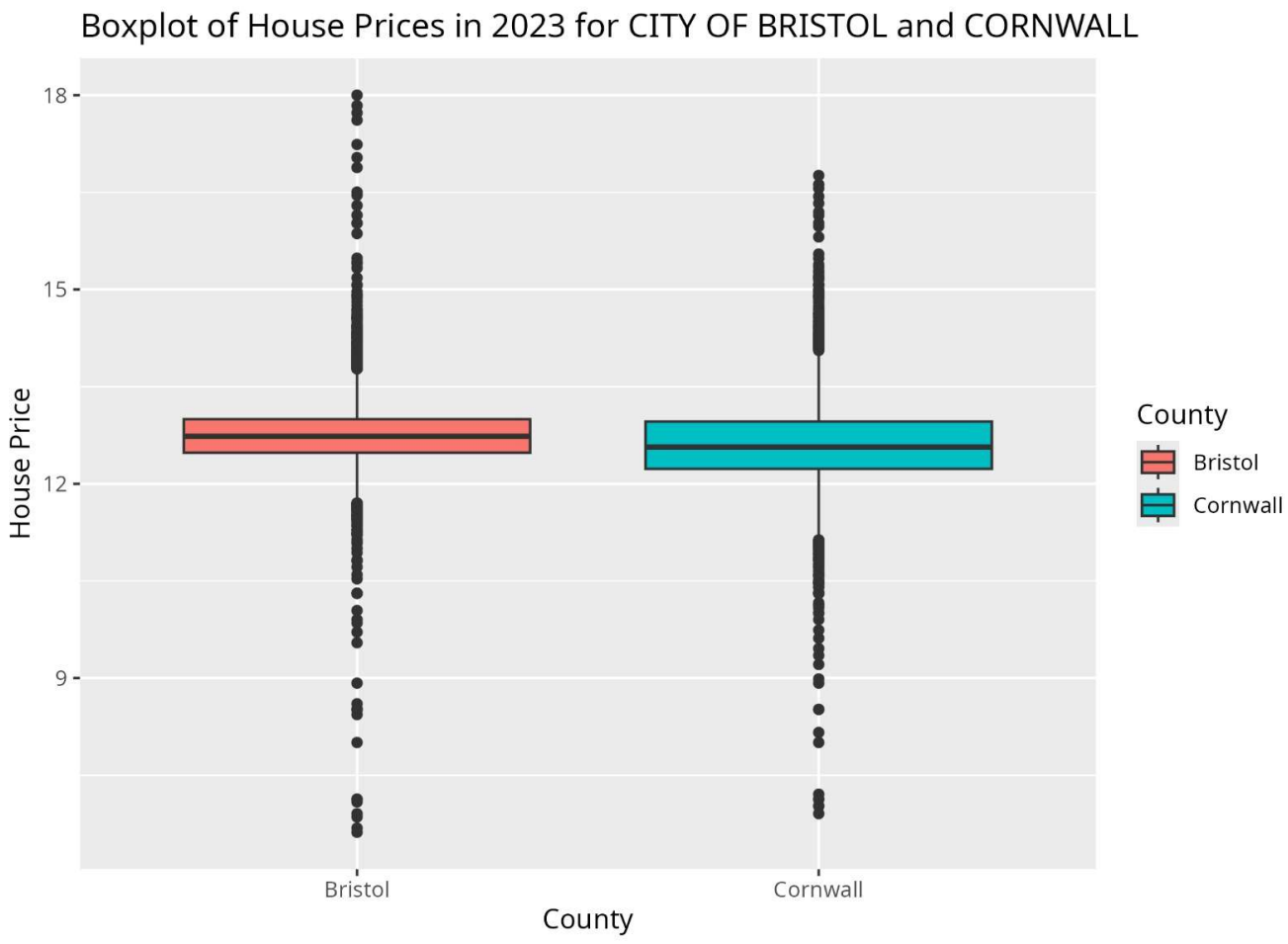


Figure 10: Code to generate Stacked BarChart for House Prices 2023

```r
# Calculate the total house price for each town within each county in 2023
town_price_summary <- house_data_2023 %>%
  group_by(County, Town.City) %>%
  summarise(Total_Price = sum(Price))

# Create a stacked bar graph showing the total house prices by town within each county
ggplot(town_price_summary, aes(x = County, y = Total_Price, fill = Town.City)) +
  geom_bar(stat = "identity") + # Stack bars based on the sum
  labs(
    title = "Total House Prices by Town within County (2023)",
    x = "County",
    y = "Total Price"
  ) +
  theme_classic() + # Optional: adjust plot aesthetics
  guides(fill = guide_legend(title = "Town")) # Label the legend as "Town"

# Save the stacked bar graph as a PNG file
ggsave(paste0(getwd(), "/Graphs/House_Pricing/BarC_Stack_Town_BL_CL_2023.png"))
```

Figure 11: Stacked Barchart House Price (2023)

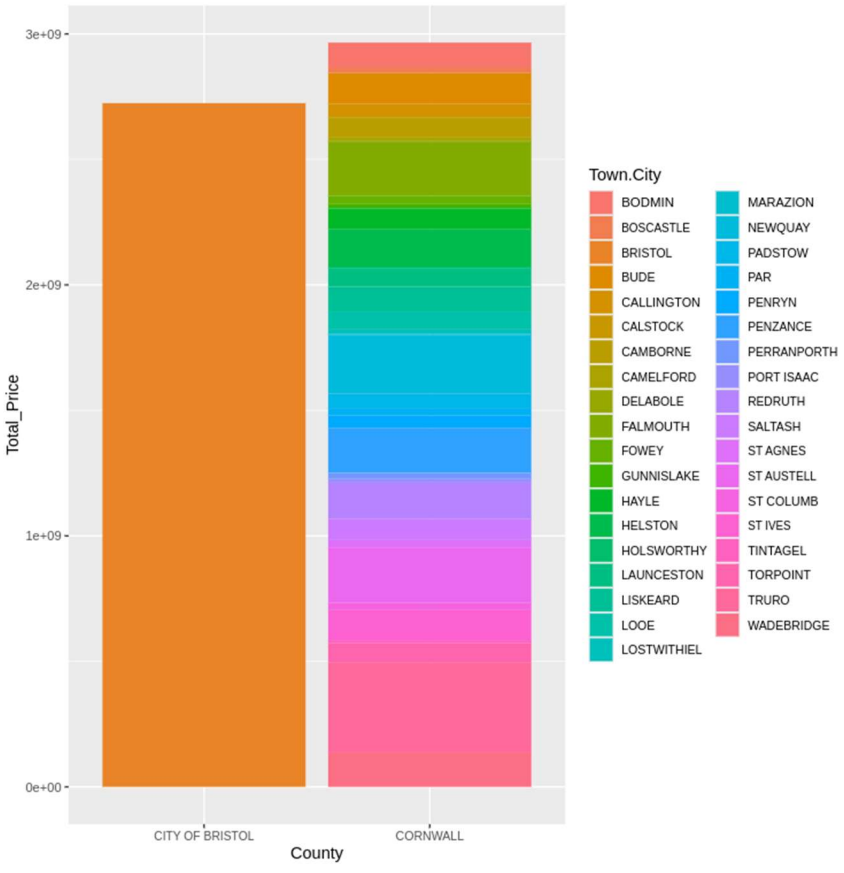


Figure 12:Code to generate Line graph for house price by town

```r
avg_prices <- house_dataset %>%
  group_by(Town.City) %>%
  summarize(avg_price = mean(Price, na.rm = TRUE)) %>%
  arrange(desc(avg_price))

# Create a line graph showing the average house price by town
ggplot(avg_prices, aes(x = reorder(Town.City, -avg_price), y = avg_price)) +
  geom_line(group = 1) +
  geom_point() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    title = "Average House Prices by Town",
    x = "Town/City",
    y = "Average Price"
  )
```

Figure 13: Line Graph for house prices by town

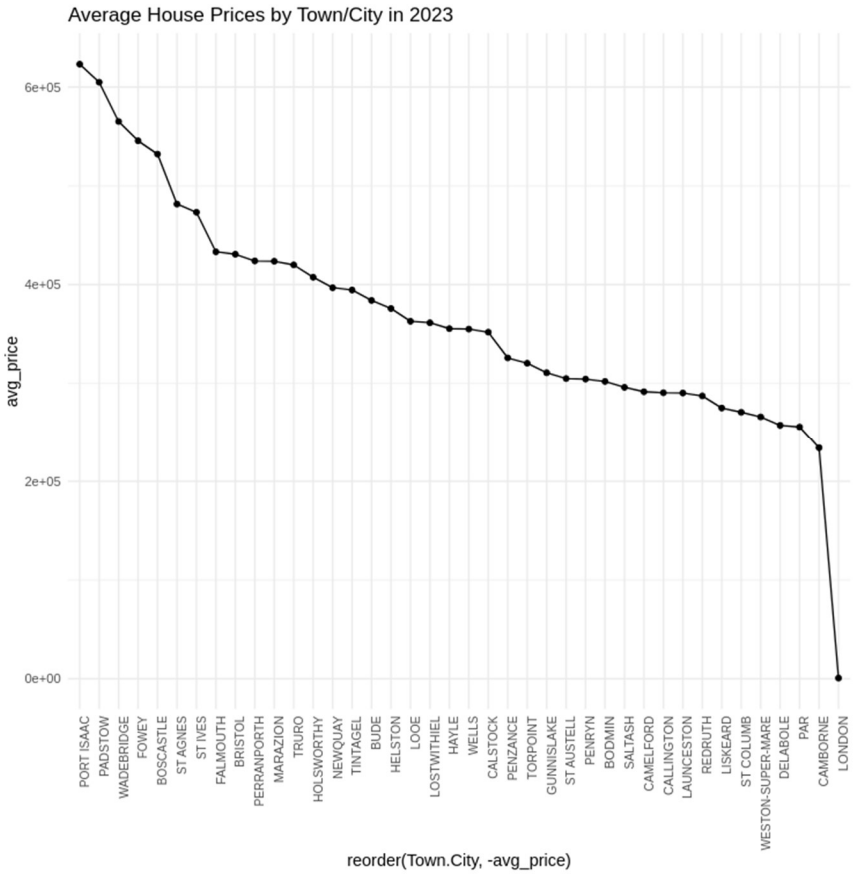


Figure 14: Code to generate Boxplot for Average download speed by county

```r
# Load the cleaned broadband performance dataset
broadband_performance_dataset <- read_csv("Cleaned_datasets/Broadband/clean_broadband_performance.csv")

# Generate a boxplot of average download speeds by county
ggplot(broadband_performance_dataset, aes(x = county, y = `Average download speed (Mbit/s)`, fill = county)) +
  geom_boxplot() + # Create a boxplot to show distribution of average download speeds
  labs(
    title = "Average Download Speed by County", # Title of the plot
    x = "County", # Label for the x-axis
    y = "Average Download Speed (Mbit/s)" # Label for the y-axis
  )

# Save the boxplot as a PNG file in the specified directory
ggsave(paste0(getwd(), "/Graphs/Broadband/average_down_speed_boxplots.png"))

```

Figure 15: Box plot for average internet speed by County

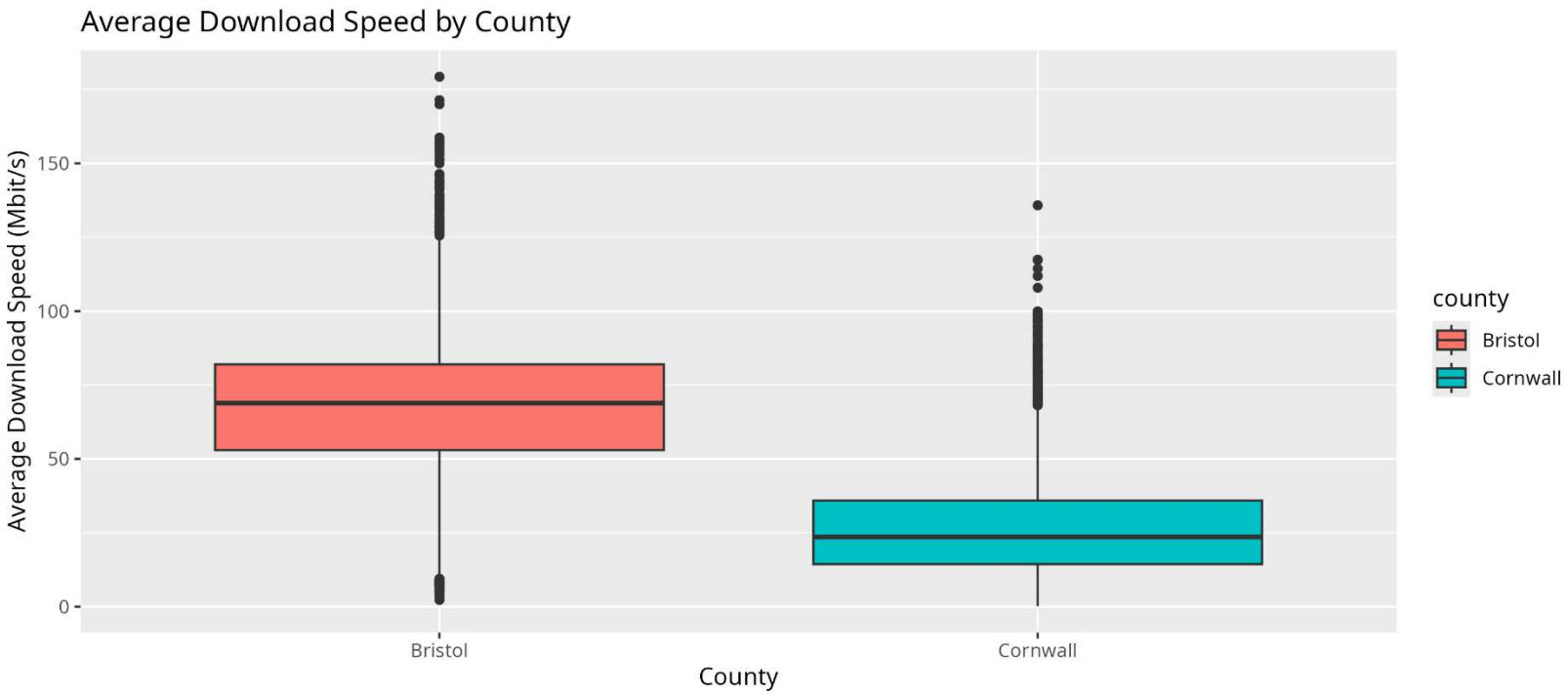


Figure 17:Code to generate dodge bar chart of maximum and average download speed by City of Bristol

```r
# Filter the dataset to include only data for Bristol
bristol_data <- broadband_performance_dataset %>%
  filter(county == "Bristol")

# Filter the dataset to include only data for Cornwall
cornwall_data <- broadband_performance_dataset %>%
  filter(county == "Cornwall")

# Reshape the Bristol data into a long format for easier plotting
long_bristol <- tidyr::gather(bristol_data, key = "speed_type", value = "speed", `Maximum download speed (Mbit/s)`, `Average download speed (Mbit/s)`)

# Reshape the Cornwall data into a long format for easier plotting
long_cornwall <- tidyr::gather(cornwall_data, key = "speed_type", value = "speed", `Maximum download speed (Mbit/s)`, `Average download speed (Mbit/s)`)

# Create a side-by-side bar chart for Bristol showing maximum vs average download speeds by city
ggplot(long_bristol, aes(x = city, y = speed, fill = speed_type)) +
  geom_bar(stat = "identity", position = "dodge") + # Use bars to show the values and dodge to place bars side by side
  labs(
    title = "Bristol: Maximum vs Average Download Speed by City", # Title of the plot
    x = "City", # Label for the x-axis
    y = "Download Speed (Mbit/s)" # Label for the y-axis
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability

# Save the Bristol bar chart as a PNG file in the specified directory
ggsave(paste0(getwd(), "/Graphs/Broadband/bristol_down_median_bar_chart.png"))
```

Figure 16: Dodge Barchart for Average vs Maximum Download Speed in Bristol

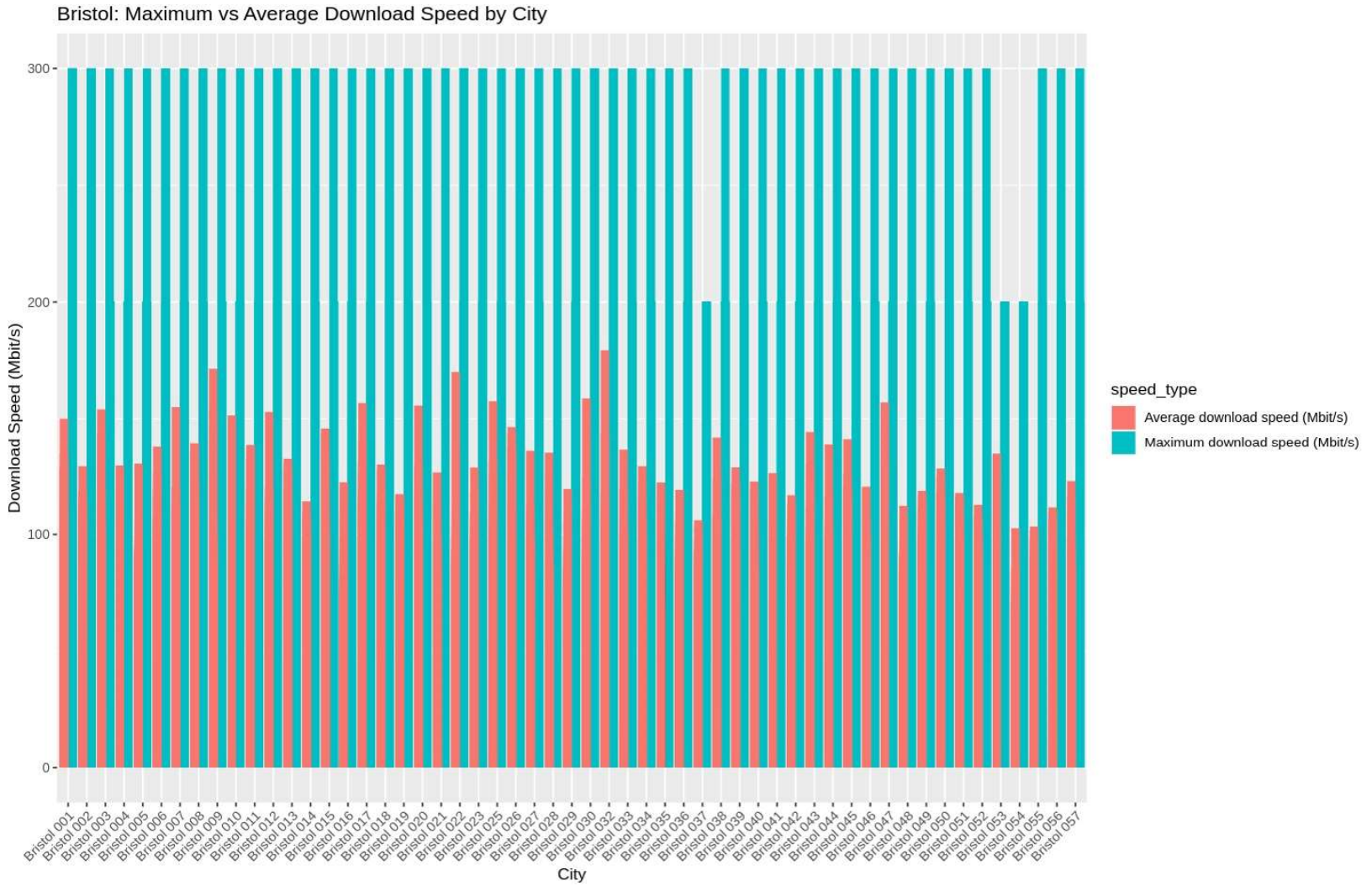


Figure 18: Figure 16: Code to generate Dodge Barchart for Average vs Maximum Download Speed in Cornwall

```r
# Create a side-by-side bar chart for Cornwall showing maximum vs average download speeds by city
ggplot(long_cornwall, aes(x = city, y = speed, fill = speed_type)) +
  geom_bar(stat = "identity", position = "dodge") + # Use bars to show the values and dodge to place bars side by side
  labs(
    title = "Cornwall: Maximum vs Average Download Speed by City", # Title of the plot
    x = "City", # Label for the x-axis
    y = "Download Speed (Mbit/s)" # Label for the y-axis
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability

# Save the Cornwall bar chart as a PNG file in the specified directory
ggsave(paste0(getwd(), "/Graphs/Broadband/cornwall_down_median_bar_chart.png"))

```

Figure 19:Dodge bar chart for Maximum vs Average Download Speed in Cornwall

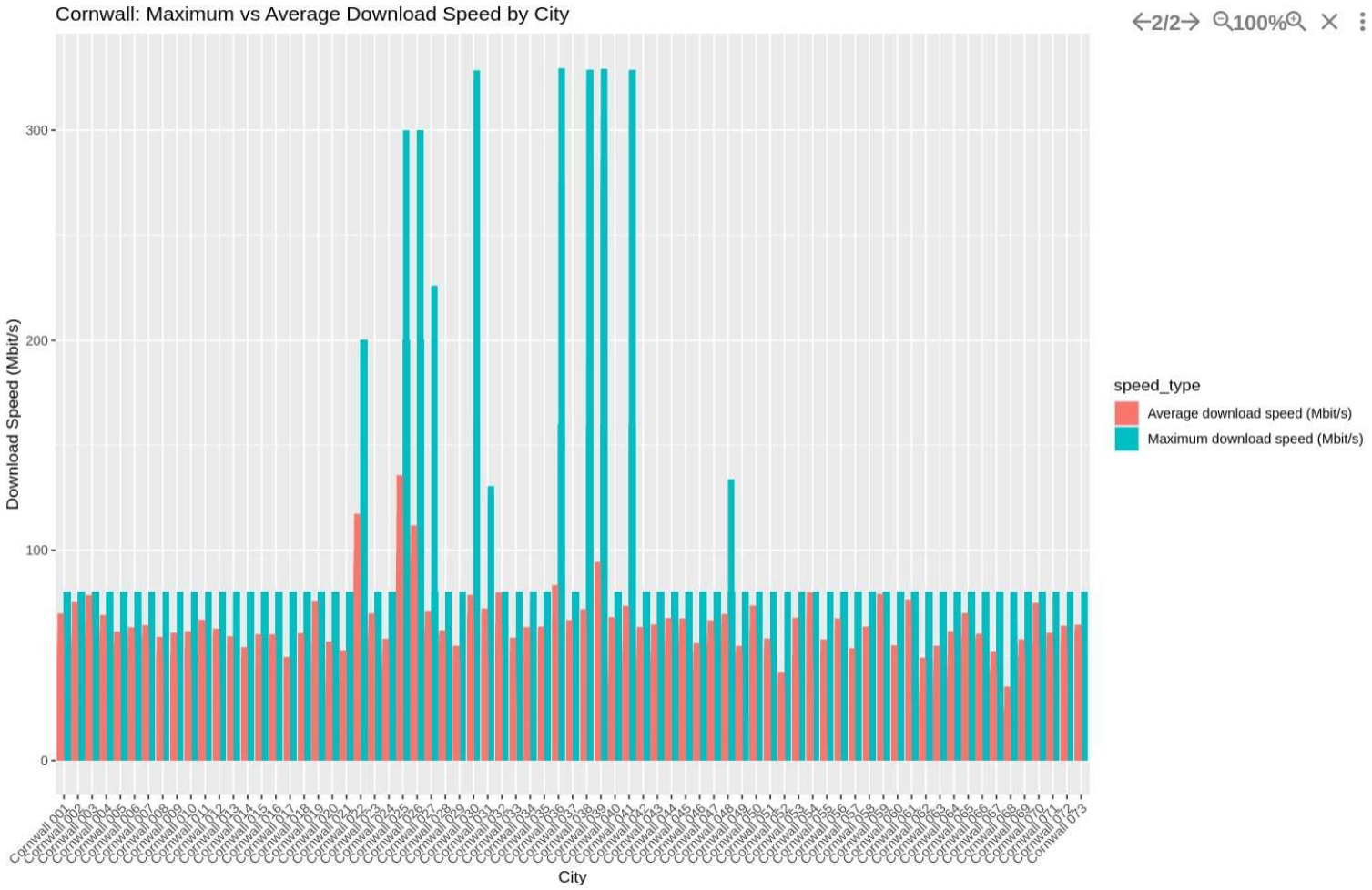


Figure 20: Code to generate boxplot of Drug Offence rate per 10000 people 2023

```r
# Filter drug offences data for the year 2023
drug_offences_2023 <- subset(drug_offences_2020_2023, format(as.Date(date), "%Y") == "2023")

# Calculate drug offence rates for 2023
drug_offences_rate_2023 <- drug_offences_2023 %>%
  filter(Crime == "Drugs") %>% # Ensure the crime type is "Drugs"
  mutate(month = floor_date(ymd(date), "month")) %>% # Extract month from the date
  group_by(month, County) %>% # Group by month and county
  summarize(
    total_offences = n(), # Count total drug offences
    offence_rate = (total_offences / population_2023[County]) * 10000 # Calculate offence rate per 10,000 population
  )

# Create a boxplot to visualize drug offence rates in 2023
ggplot(drug_offences_rate_2023, aes(x = County, y = offence_rate, fill = County)) +
  geom_boxplot() + # Draw boxplots
  labs(
    title = "Drug Offence Rate (2023)", # Title of the plot
    x = "Region", # Label for the x-axis
    y = "Offence Rate " # Label for the y-axis
  )

# Save the boxplot as a PNG file
ggsave(paste0(getwd(), "/Graphs/Crime/BoxP_Drug_BL_CL_2023.png"))

```

Figure 21: Box plot for drug offence (2023)

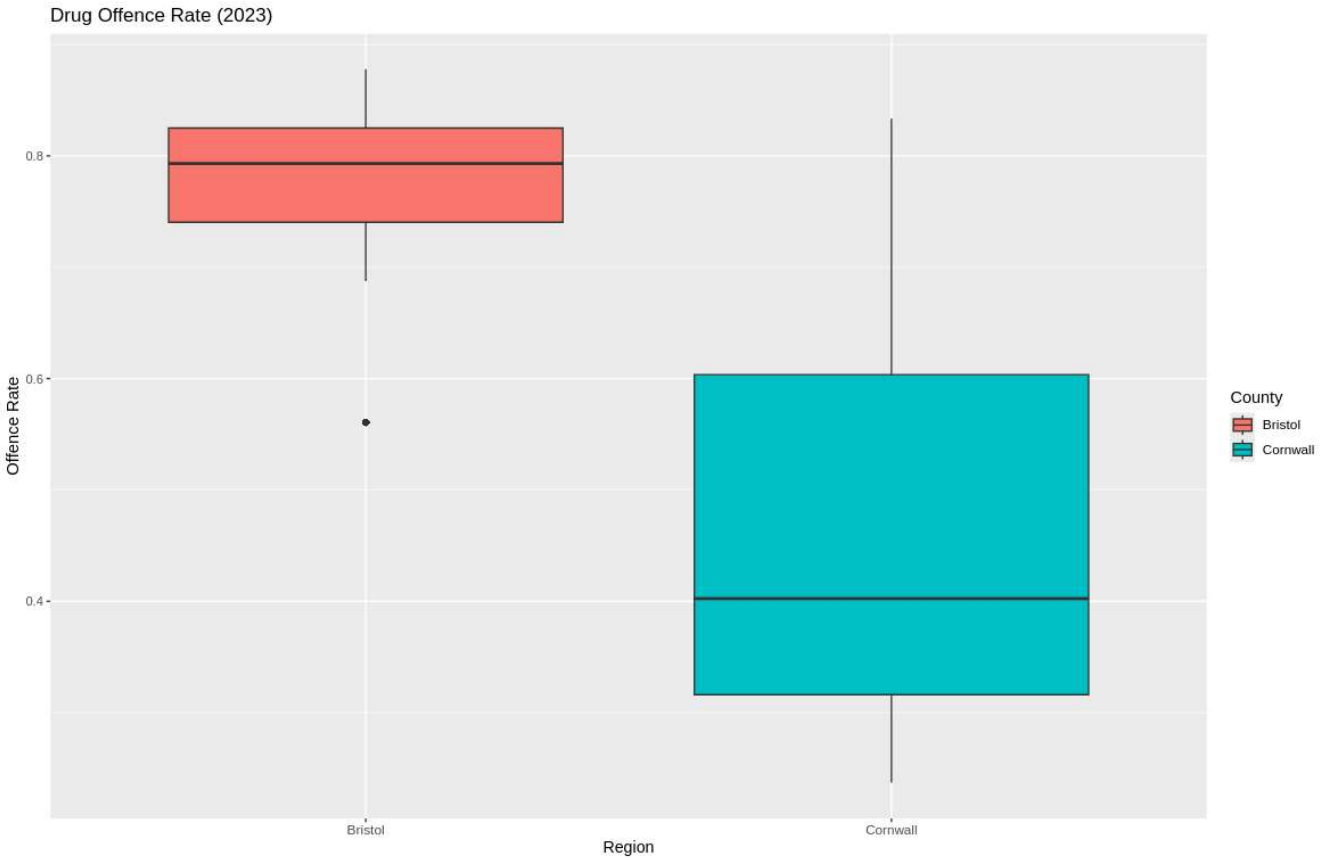


Figure 22: Code to generate Line graph for drug offence rate 2020-2023

```r
#Load population dataset to aceess data on population around Bristol and Cornall
population_2011 <- read_csv("Cleaned_datasets/Population_clean.csv") %>%
  # Convert the data frame to a named vector
  deframe()

# Calculate the total population for Bristol and Cornwall combined
total_population <- sum(population_2011[c("Bristol", "Cornwall")], na.rm = TRUE)

# Add the combined total to the named vector with the name "Total"
population_2011 <- c(population_2011, Total = total_population)

# Remove any NA entries from the named vector
population_2011 <- population_2011[!is.na(names(population_2011))]

# Estimate the population for 2023 using a growth factor
population_2023 <- floor((1.00561255390388033 * population_2011))

#########################################################################################################################################

# Load crime data
crime_dataset <- read_csv("Cleaned_datasets/Crime/Crime_Data_Combined.csv")

# Filter crime data for drug offences between 2020 and 2023
drug_offences_2020_2023 <- crime_dataset %>%
  filter(Crime == "Drugs", year(date) >= 2020 & year(date) <= 2023)

# Calculate the drug offence rate per 10,000 population
drug_offences_rate_2020_2023 <- drug_offences_2020_2023 %>%
  # Extract month and year from the date for grouping
  mutate(
    month = floor_date(ymd(date), "month"),
    year = year(date)
  ) %>%
  # Group by year and county, then count the total number of offences
  group_by(year, County) %>%
  summarize(
    total_offences = n(),
    .groups = "drop"
  ) %>%
  # Join with population data to get population numbers for each county
  left_join(
    tibble(
      County = names(population_2023),
      population = as.numeric(population_2023)
    ),
    by = "County"
  ) %>%
  # Calculate the offence rate per 10,000 population
  mutate(offence_rate = (total_offences / population) * 10000)

# Create a line graph to visualize drug offence rates by year
ggplot(drug_offences_rate_2020_2023, aes(x = year, y = offence_rate, color = County, group = County)) +
  geom_line() + # Add lines to connect points
  geom_point(size = 3) + # Add points to the line graph
  labs(
    title = "Yearly Drug Offence Rate per 10000 Population (2020-2023)", # Title of the plot
    x = "Year", # Label for the x-axis
    y = "Offence Rate per 10000 Population", # Label for the y-axis
    color = "Region" # Label for the color legend
  ) +
  theme_minimal() + # Use a minimal theme for the plot
  theme(
    legend.position = "bottom" # Position the legend at the bottom
  ) +
  scale_x_continuous(breaks = 2020:2023) # Set x-axis breaks to display years 2020 to 2023

# Save the line graph as a PNG file in the specified directory
ggsave(paste0(getwd(), "/Graphs/Crime/LineG_Drug_10000_2023.png"))
```

Figure 23:Line graph for Yearly drug offence rate 2020-2023

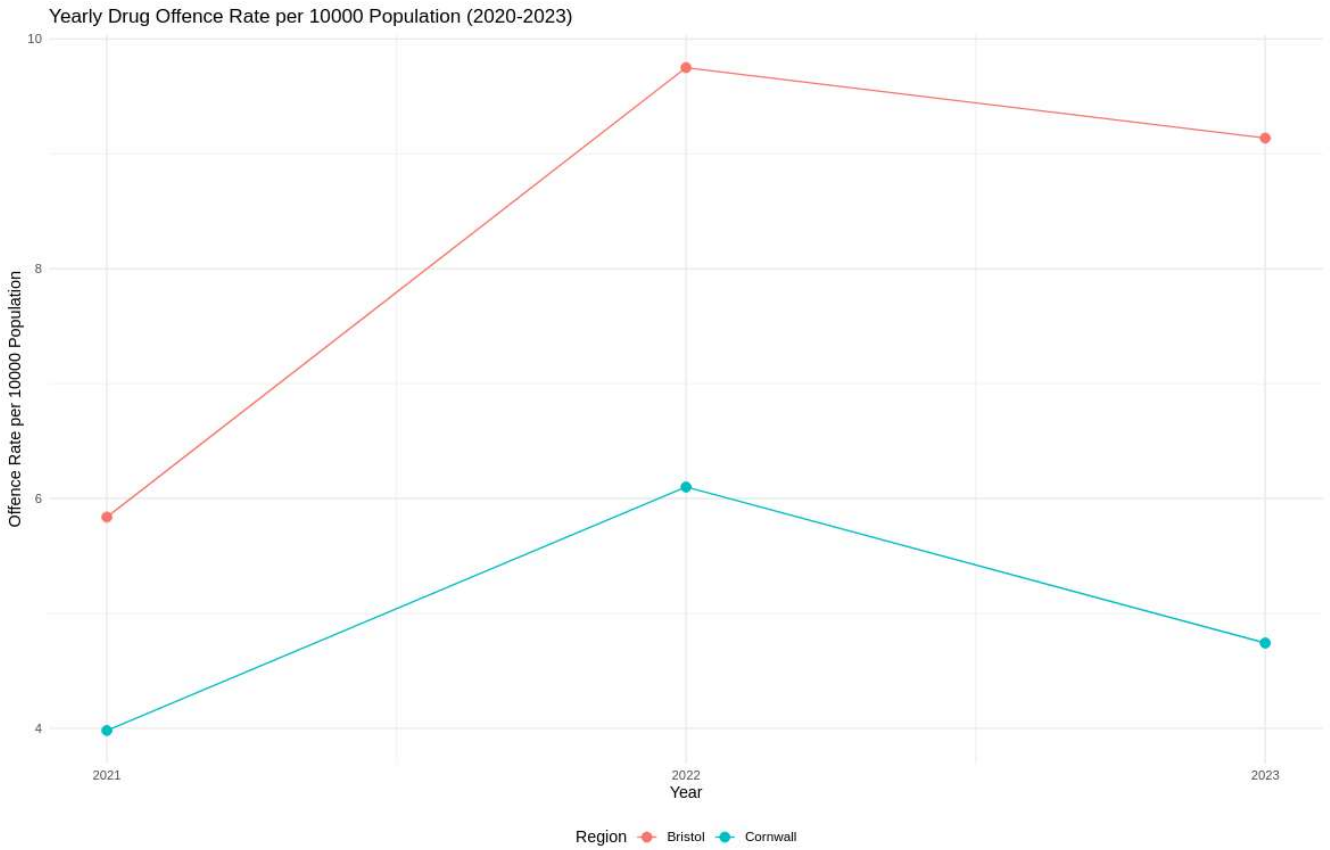


Figure 25: Code to generate Piechart for robberies on October 2022

```r
robbery_offences_oct_2023 <- crime_dataset %>%
  filter(
    Crime == "Robbery", # Filter for robbery offences
    year(date) == 2022, # Only data from the year 2022
    month(date) == 10 # Only data from October
  )

# Calculate robbery rates for October 2022
robbery_rates <- robbery_offences_oct_2023 %>%
  group_by(County) %>% # Group by county
  summarize(
    total_offences = n(), # Count total robbery offences
    .groups = "drop" # Drop the grouping after summarizing
  ) %>%
  # Join with population data for each county
  left_join(
    tibble(
      County = names(population_2023),
      population = as.numeric(population_2023)
    ),
    by = "County"
  ) %>%
  mutate(
    offence_rate = (total_offences / population) * 10000, # Calculate offence rate per 10,000 population
    percentage = total_offences / sum(total_offences) * 100
  ) # Calculate percentage of total offences


# Create a pie chart to visualize the distribution of robbery offences in October 2022
ggplot(robbery_rates, aes(x = "", y = percentage, fill = County)) +
  geom_bar(stat = "identity", width = 1) + # Use bars to represent proportions
  coord_polar("y", start = 0) + # Convert bar chart to a pie chart
  geom_text(aes(label = sprintf("%.1f%%\n(%d)", percentage, total_offences)), # Add labels with percentage and count
    position = position_stack(vjust = 0.5)
  ) +
  labs(
    title = "Robbery Offences Distribution (October 2022)", # Title of the plot
    subtitle = paste("Total offences:", sum(robbery_rates$total_offences)), # Subtitle with total offences
    fill = "Region" # Label for the fill legend
  ) +
  theme_void() + # Use a void theme to remove background and axes
  theme(legend.position = "bottom") # Position the legend at the bottom

# Save the pie chart as a PNG file
ggsave(paste0(getwd(), "/Graphs/Crime/Pi_Robbery_oct_2022.png"))
```

Figure 24: Pie chart for robbery rate on October 2022

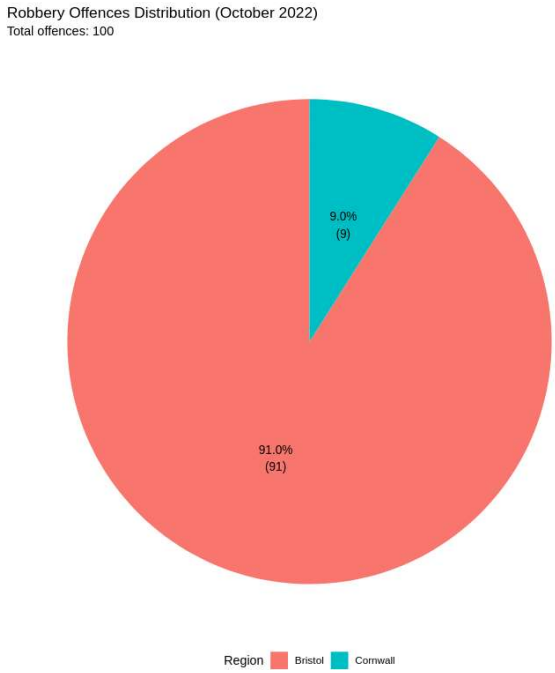


Figure 26: Radar chart for vehicle crime rates from 2020-2023

```r
population_2011 <- read_csv("Cleaned_datasets/Population_clean.csv") %>%
  deframe()

total_population <- sum(population_2011[c("Bristol", "Cornwall")], na.rm = TRUE)
# Add the total to the list with the name "Total"
population_2011 <- c(population_2011, Total = total_population)
# Remove the NA entry if it exists
population_2011 <- population_2011[!is.na(names(population_2011))]


population_2023 <- floor((1.00561255390388033 * population_2011))


crime_dataset <- read_csv("Cleaned_datasets/Crime/Crime_Data_Combined.csv")

vehicle_crime_2020_2023 <- crime_dataset %>%
  filter(Crime == "Vehicle crime", year(date) >= 2020 & year(date) <= 2023)

vehicle_crime_rate_2020_2023 <- vehicle_crime_2020_2023 %>%
  mutate(
    year = year(date)
  ) %>%
  group_by(year, County) %>%
  summarize(
    total_offences = n(),
    .groups = "drop"
  ) %>%
  left_join(
    tibble(
      County = names(population_2023),
      population = as.numeric(population_2023)
    ),
    by = "County"
  ) %>%
  mutate(offence_rate = (total_offences / population) * 10000) %>%
  select(year, County, offence_rate)

# Reshape the data for ggradar
vehicle_crime_rate_wide <- vehicle_crime_rate_2020_2023 %>%
  pivot_wider(names_from = year, values_from = offence_rate, names_prefix = "Year_") %>%
  mutate(across(starts_with("Year_"), ~replace_na(., 0)))

# Create radar chart
ggradar(vehicle_crime_rate_wide,
        values.radar = c("0", "5", "10"),
        grid.min = 0,
        grid.max = 10,
        grid.mid = 5,
        group.point.size = 3,
        group.line.width = 1.5,
        axis.label.size = 3,
        legend.text.size = 10,
        legend.position = "bottom",
        background.circle.colour = "white",
        axis.line.colour = "gray60",
        gridline.mid.colour = "gray60") +
  labs(title = "Vehicle Crime Rate per 10,000 Population (2020-2023)")

```

Figure 27: Radar chart for yearly vehicle crimes 2020-2023

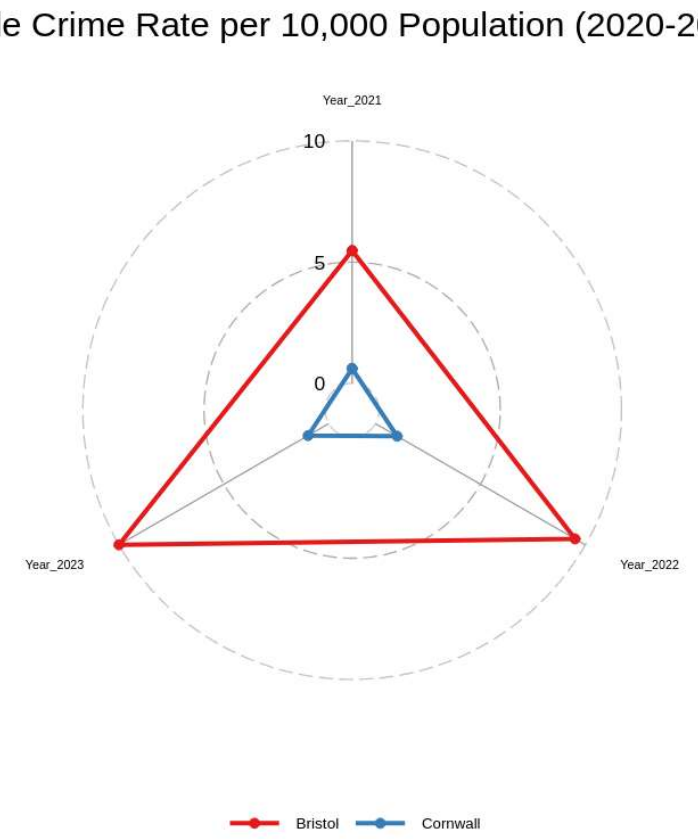


Figure 28: Code to generate Boxplot of attainment 8 score 2023

```r
 Read the cleaned school dataset and filter out data for Wiltshire
school_dataset <- read_csv("Cleaned_datasets/Schools/Schools_Clean.csv") %>%
  filter(County != "Wiltshire")

# Subset the dataset to include only records for the year 2023
school_data_2023 <- subset(school_dataset, Year == 2023)

# Create a boxplot of ATT8SCR (Attainment 8 Score) by County for the year 2023
ggplot(school_data_2023, aes(x = County, y = ATT8SCR, fill = County)) +
  geom_boxplot() + # Use a boxplot to visualize the distribution of ATT8SCR by County
  labs(
    title = "Boxplot of ATT8SCR by County",
    x = "County",
    y = "ATT8SCR"
  ) + # Label the plot
  theme(axis.text.x = element_text(hjust = 1)) # Adjust text alignment for x-axis labels

# Save the boxplot to a file
ggsave(paste0(getwd(), "/Graphs/Schools/BoxP_avg_att8_BL_CL_2023.png"))
```

Figure 29: Attainment 8 score by county 2023

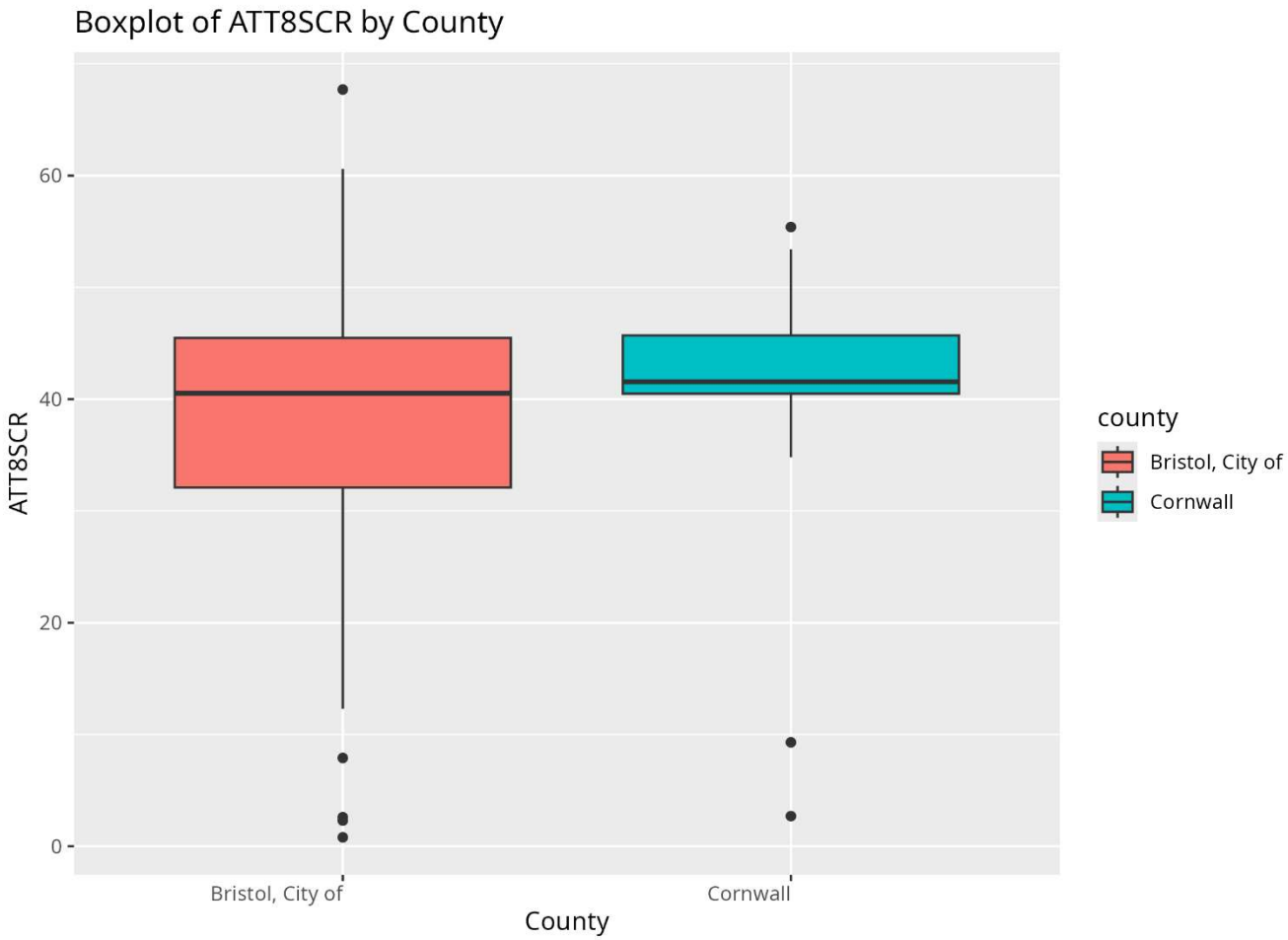


Figure 30: Code to lime graph of attainment 8 score city wise in Bristol and Cornwall

```r
cornwall_data <- school_dataset %>%
  filter(County == "Cornwall")

# Filter data for Bristol
bristol_data <- school_dataset %>%
  filter(County == "Bristol")

# Create a line graph of average Attainment 8 Score by City in Cornwall over the years
ggplot(cornwall_data, aes(x = Year, y = ATT8SCR, color = City)) +
  geom_line() + # Use a line graph to show the trend of ATT8SCR by City
  labs(
    title = "Average Attainment 8 Score by City in Cornwall",
    x = "Year",
    y = "Average Attainment 8 Score"
  ) # Label the plot

# Save the boxplot to a file
ggsave(paste0(getwd(), "/Graphs/Schools/line_avg_att8_CL.png"))

# Create a line graph of average Attainment 8 Score by City in Bristol over the years
ggplot(bristol_data, aes(x = Year, y = ATT8SCR, color = City)) +
  geom_line() + # Use a line graph to show the trend of ATT8SCR by City
  labs(
    title = "Average Attainment 8 Score by City in Bristol",
    x = "Year",
    y = "Average Attainment 8 Score"
  ) # Label the plot

# Save the boxplot to a file
ggsave(paste0(getwd(), "/Graphs/Schools/line_avg_att8_BL.png"))
```

Figure 31: Line graph to see average attainment 8 score in Cornwall

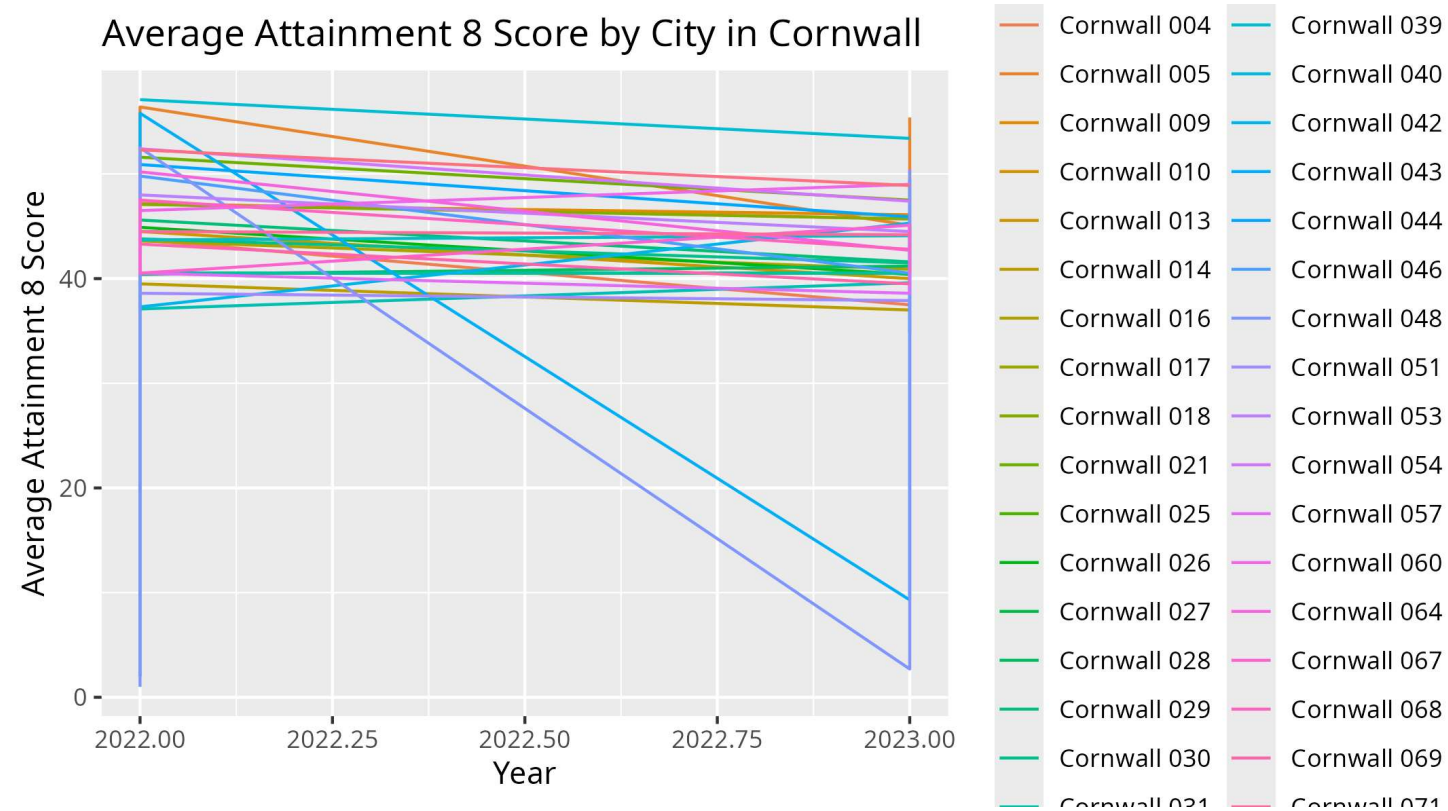


Figure 32: Line graph to see average attainment 8 score in Bristol

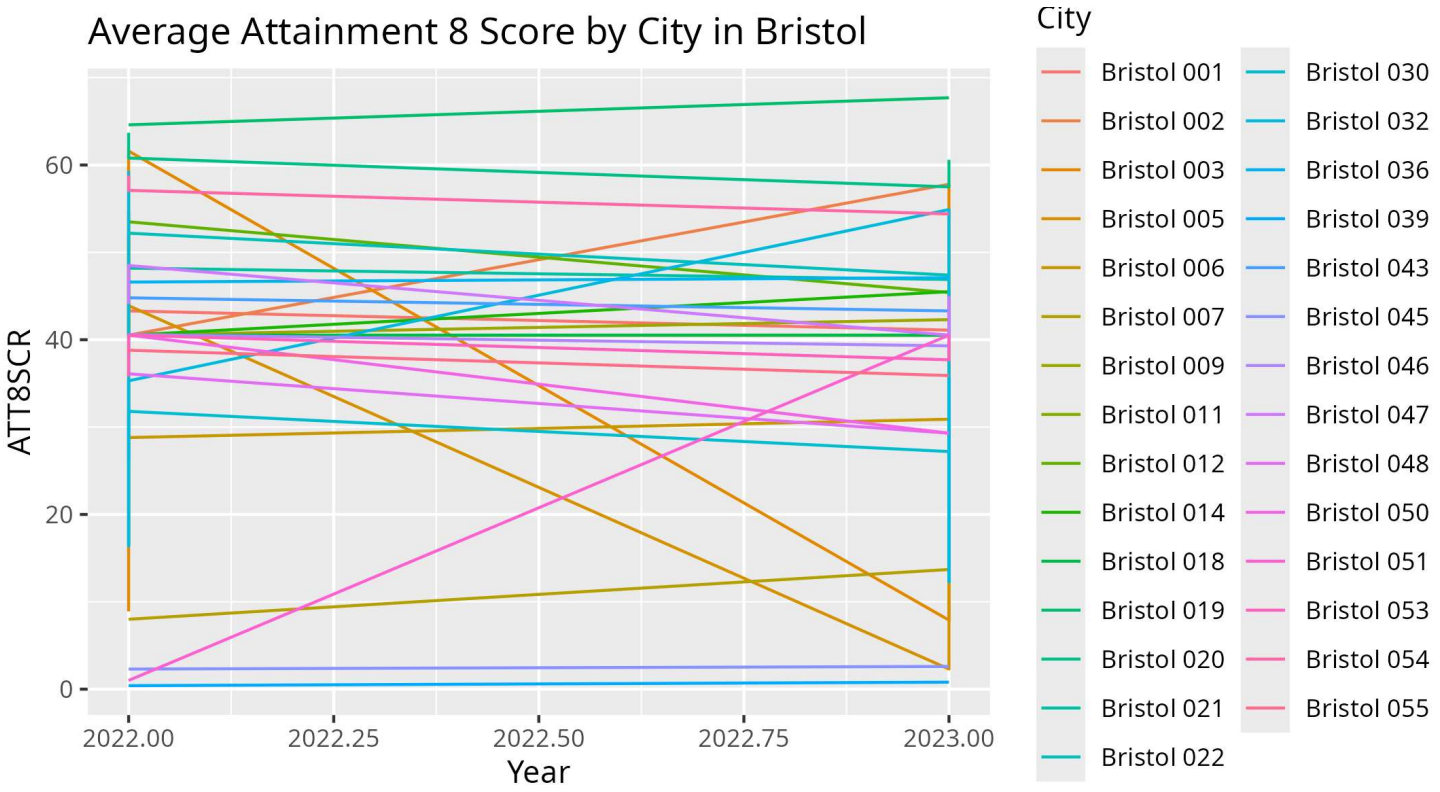


Figure 33: Code to perform linear modelling of attainment 8 score and house price

```r
school_dataset <- read_csv("Cleaned_datasets/Schools/Schools_Clean.csv")
house_dataset <- read_csv("Cleaned_datasets/House_Pricing_Data/clean_house_pricing_data.csv")


population_2011 <- read_csv("Cleaned_datasets/Population_clean.csv") %>%
  deframe()

total_population <- sum(population_2011[c("Bristol", "Cornwall")], na.rm = TRUE)
# Add the total to the list with the name "Total"
population_2011 <- c(population_2011, Total = total_population)
# Remove the NA entry if it exists
population_2011 <- population_2011[!is.na(names(population_2011))]


population_2023 <- floor((1.00561255390388033 * population_2011))

merged_dataset <- merge(
  school_dataset,
  house_dataset,
  by = "PostalCode", # Use the updated column name for matching
  all.x = TRUE, # Include all rows from the school_dataset even if there is no matching data in house_dataset
  all.y = FALSE # Exclude rows from house_dataset if there is no matching row in school_dataset
) %>%
  filter(!is.na(ATT8SCR), !is.na(Price))

model <- lm(ATT8SCR ~ Price, data = merged_dataset)

# Check the model summary
summary(model)
```

Figure 34: Code to perform Linear Modelling between Attainment 8 score and average download speed

```r
school_dataset <- read_csv("Cleaned_datasets/Schools/Schools_Clean.csv")
broadband_performance_dataset <- read_csv("Cleaned_datasets/Broadband/clean_broadband_performance.csv")

merged_dataset <- merge(
  broadband_performance_dataset,
  school_dataset,
  by = "County", # Use the updated column name for matching
  all.x = TRUE, # Include all rows from the school_dataset even if there is no matching data in house_dataset
  all.y = FALSE # Exclude rows from house_dataset if there is no matching row in school_dataset
) %>%
  filter(!is.na(`Average download speed (Mbit/s)`), !is.na(ATT8SCR))

# Fit a linear model
model <- lm(ATT8SCR ~ `Average download speed (Mbit/s)`, data = merged_dataset)

# Check the model summary
summary(model)
```

Figure 35: Code to perform linear modeling between drug offence rate per 1000 people and average download speed

```r
broadband_dataset <- read_csv("Cleaned_datasets/Broadband/clean_broadband_performance.csv")

drug_offences_2020_2023 <- read_csv("Cleaned_datasets/Crime/Crime_Data_Combined.csv") %>%
  filter(Crime == "Drugs", year(date) >= 2020 & year(date) <= 2023)

population_dataset <- read_csv("Obtained_Data/Population2011_1656567141570.csv") %>%
  mutate(Postcode = substr(Postcode, start = 1, stop = 2))

postcode_dataset <- read_csv("Cleaned_datasets/Postcode_clean.csv") %>%
  select(pcd7, County) %>%
  mutate(pcd7 = substr(pcd7, start = 1, stop = 2)) %>%
  distinct()

population_2011 <- read_csv("Cleaned_datasets/Population_clean.csv") %>%
  deframe()

total_population <- sum(population_2011[c("Bristol", "Cornwall")], na.rm = TRUE)
# Add the total to the list with the name "Total"
population_2011 <- c(population_2011, Total = total_population)
# Remove the NA entry if it exists
population_2011 <- population_2011[!is.na(names(population_2011))]


population_2023 <- floor((1.00561255390388033 * population_2011))

drug_offences_rate_2020_2023 <- drug_offences_2020_2023 %>%
  # Extract month and year from the date for grouping
  mutate(
    month = floor_date(ymd(date), "month"),
    year = year(date)
  ) %>%
  # Group by year and county, then count the total number of offences
  group_by(year, City, County) %>%
  summarize(
    total_offences = n(),
    .groups = "drop"
  ) %>%
  # Join with population data to get population numbers for each county
  left_join(
    tibble(
      County = names(population_2023),
      population = as.numeric(population_2023)
    ),
    by = "County"
  ) %>%
  # Calculate the offence rate per 10,000 population
  mutate(offence_rate = (total_offences / population) * 10000) %>%
  rename("Street" = "City")


# Merge datasets based on common columns, such as post codes or county
# Here, we'll merge on street as an example
merged_dataset <- merge(
  broadband_dataset,
  drug_offences_rate_2020_2023,
  by = "Street",
  all.x = TRUE,
  all.y = FALSE
) %>%
  na.omit()

# Fit a linear model
model <- lm(offence_rate ~ `Average download speed (Mbit/s)`, data = merged_dataset)

# Check the model summary
summary(model)
```

Figure 36: Code to perform linear modelling between house prices and drug offence rate in 2023

```r
house_dataset_2023 <- read_csv("Cleaned_datasets/House_Pricing_Data/clean_house_pricing_data.csv") %>%
  filter(format(as.Date(Date), "%Y") == "2023")

drug_offences_2023 <- read_csv("Cleaned_datasets/Crime/Crime_Data_Combined.csv") %>%
  filter(Crime == "Drugs", year(date) == 2023)

population_2011 <- read_csv("Cleaned_datasets/Population_clean.csv") %>%
  deframe()

total_population <- sum(population_2011[c("Bristol", "Cornwall")], na.rm = TRUE)
# Add the total to the list with the name "Total"
population_2011 <- c(population_2011, Total = total_population)
# Remove the NA entry if it exists
population_2011 <- population_2011[!is.na(names(population_2011))]


population_2023 <- floor((1.00561255390388033 * population_2011))

drug_offences_rate_2023 <- drug_offences_2023 %>%
  # Extract month and year from the date for grouping
  mutate(
    month = floor_date(ymd(date), "month"),
    year = year(date)
  ) %>%
  # Group by year and county, then count the total number of offences
  group_by(year, City, County) %>%
  summarize(
    total_offences = n(),
    .groups = "drop"
  ) %>%
  # Join with population data to get population numbers for each county
  left_join(
    tibble(
      County = names(population_2023),
      population = as.numeric(population_2023)
    ),
    by = "County"
  ) %>%
  # Calculate the offence rate per 10,000 population
  mutate(offence_rate = (total_offences / population)) %>%
  rename("Street" = "City")

merged_data <- left_join(house_dataset_2023, drug_offences_rate_2023, by = c("Street", "County")) %>%
  na.omit()



model <- lm(Price ~ offence_rate, data = merged_data)

# Summary of the model to view details
summary(model)
```

Figure 37: Code to perform linear modelling of house price and Average download speed

```r
house_dataset <- read_csv("Cleaned_datasets/House_Pricing_Data/clean_house_pricing_data.csv")
broadband_performance_dataset <- read_csv("Cleaned_datasets/Broadband/clean_broadband_performance.csv")

merged_data <- left_join(house_dataset, broadband_performance_dataset, by = "PostalCode") %>%
  na.omit()


model <- lm(Price ~ `Average download speed (Mbit/s)`, data = merged_data)

# Summary of the model to view details
summary(model)
```

Figure 38: Code to generate line of best fit between download speed and House Price

```r
ggplot(merged_data, aes(x = `Average download speed (Mbit/s)`, y = Price)) +
  geom_point(color = "blue") + # Scatter plot of the data
  scale_y_continuous(trans = "log10") + # Plot the data points
  geom_smooth(method = "lm", color = "red", se = FALSE) + # Line of best fit
  labs(
    title = "House Price vs Average Download Speed",
    x = "Average Download Speed (Mbit/s)",
    y = "House Price"
  ) +
  theme_minimal()
```

Figure 39: Line of best fit between download speed and house price

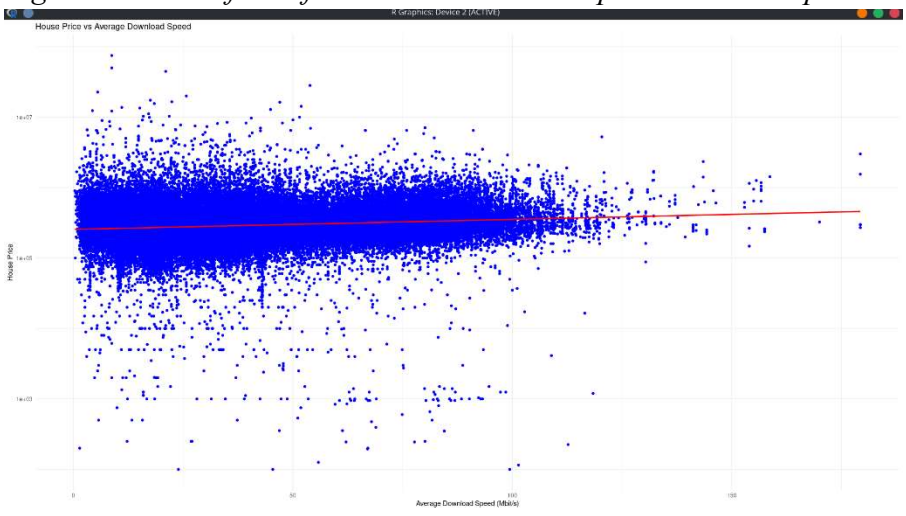


Figure 40: Code to generate line of best fit between drug offence rate and house price 2023

```r
ggplot(merged_data, aes(x = offence_rate, y = Price)) +
  geom_point(color = "blue") + # Scatter plot of the data
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  scale_y_continuous(trans = "log10") + # Plot the data points# Line of best fit
  labs(
    title = "House Price vs Drug Offence Rate (2023)",
    x = "Offence Rate",
    y = "House Price"
  ) +
  theme_minimal()
```

Figure 41: line of best fit between drug offence rate and house price 2023

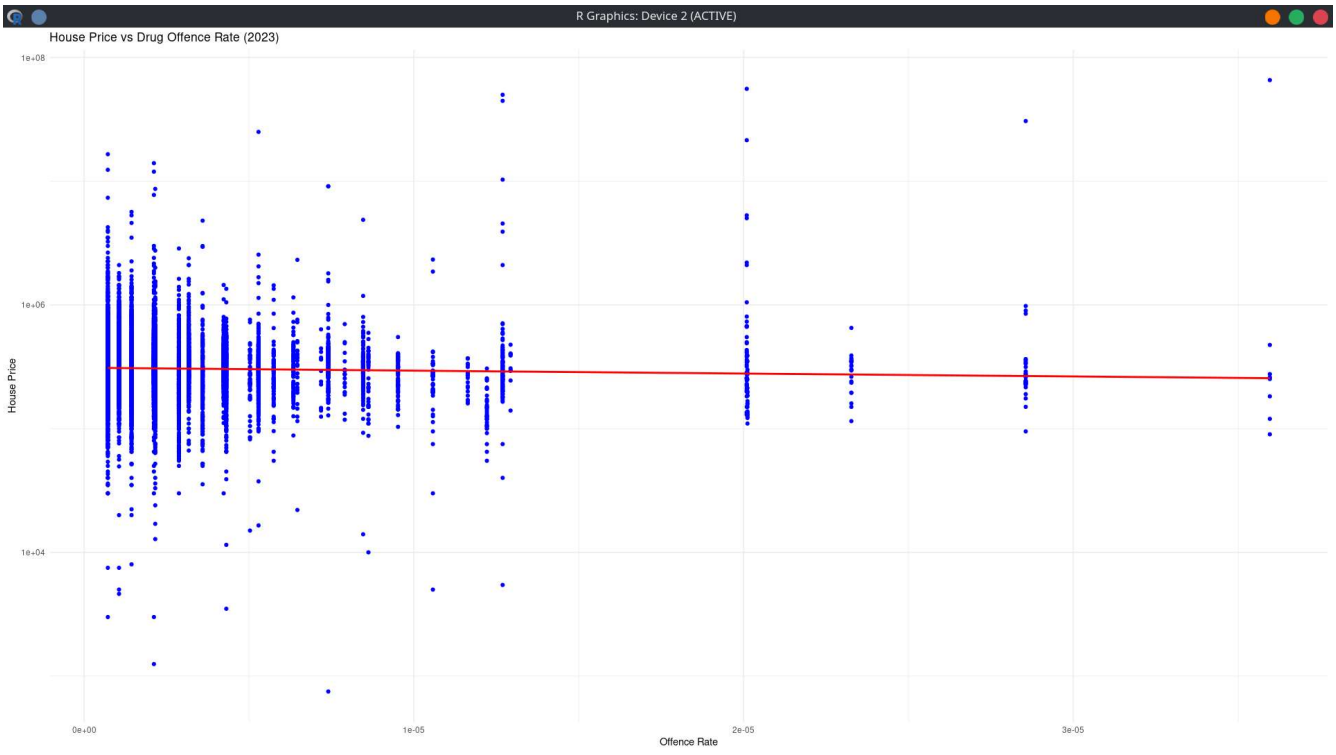


Figure 42: Code to generate line of best fit for house price and attainment 8 score

```r
ggplot(merged_dataset, aes(x = Price, y = ATT8SCR)) +
  geom_point(color = "blue", alpha = 0.5) + # Scatter plot of data points
  geom_smooth(method = "lm", color = "red", se = FALSE) + # Line of best fit
  labs(
    title = "Relationship between School ATT8 Scores and House Prices",
    x = "House Price",
    y = "ATT8 Score"
  ) +
  theme_minimal()
```

Figure 43: line of best fit for house price and attainment 8 score

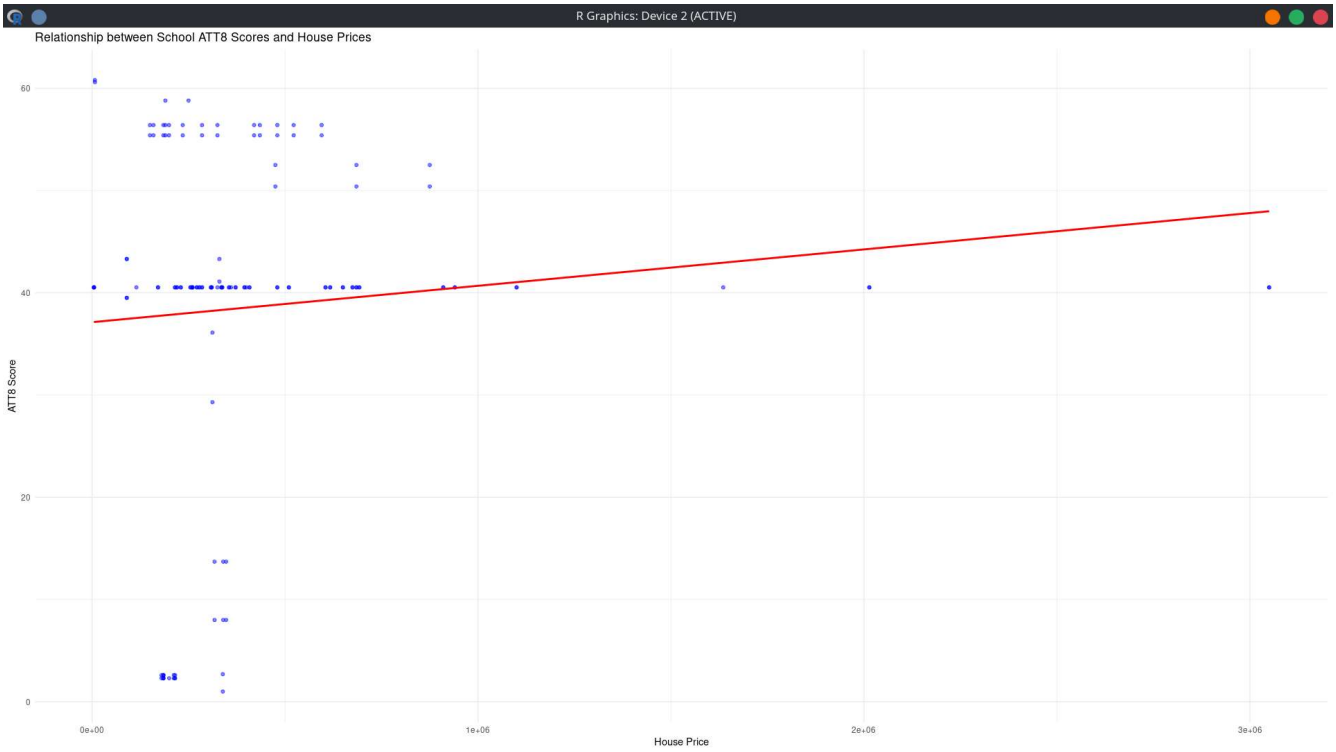


Figure 44: Code to generate line of best fit for download speed and drug offence rate per 1000 people

```r
# Create a scatter plot with the line of best fit
ggplot(merged_dataset, aes(x = `Average download speed (Mbit/s)`, y = offence_rate)) +
  geom_point(color = "blue", alpha = 0.5) + # Scatter plot of data points
  geom_smooth(method = "lm", color = "red", se = FALSE) + # Line of best fit
  labs(
    title = "Relationship between Average Download Speed and Drug Offence Rates per 10000 people",
    x = "Average Download Speed (Mbit/s)",
    y = "Drug Offence Rate per 1000 Population"
  ) +
  theme_minimal()

```

Figure 45:line of best fit for download speed and drug offence rate per 10000 people

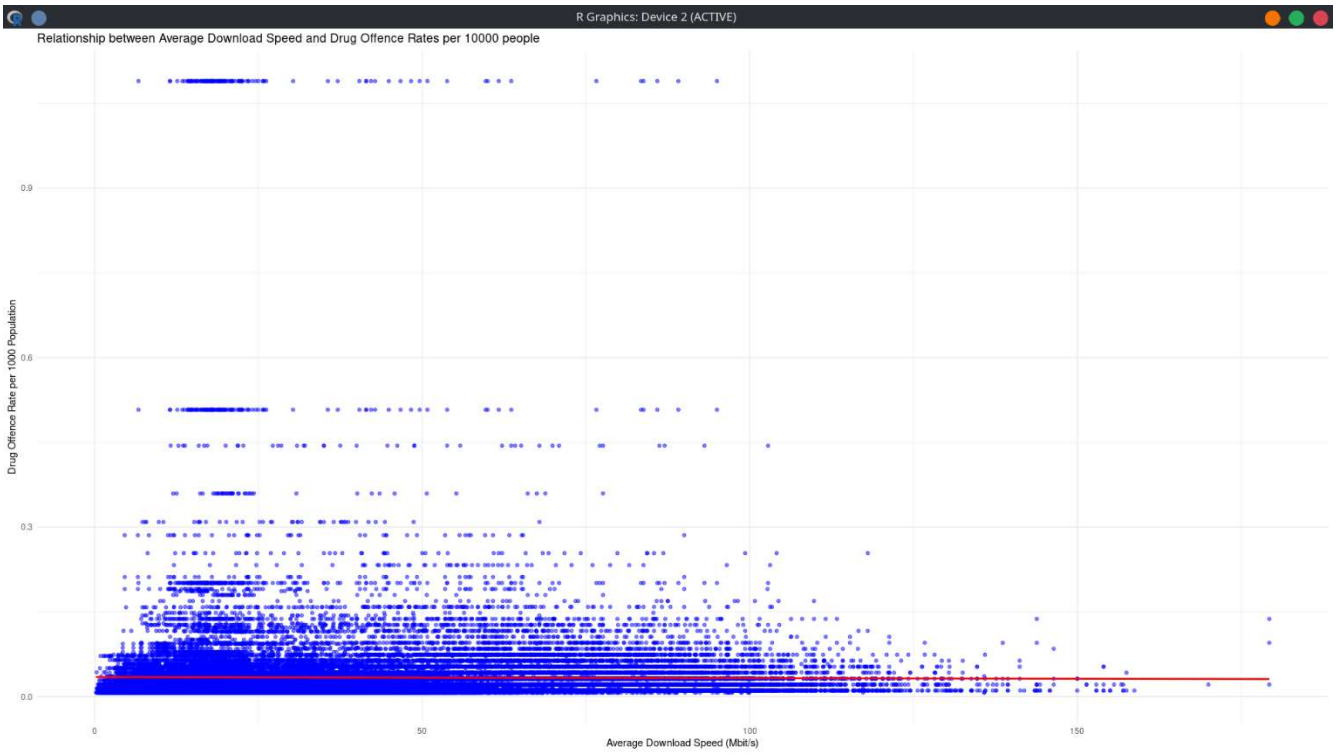


Figure 46:Code to generate line of best fit for average download speed and attainment 8 score

```r
ggplot(merged_dataset, aes(x = `Average download speed (Mbit/s)`, y = ATT8SCR)) +
  geom_point(color = "blue", alpha = 0.5) + # Scatter plot of data points
  geom_smooth(method = "lm", color = "red", se = FALSE) + # Line of best fit
  labs(
    title = "Relationship between Average Download Speed and Attainment 8 Score",
    x = "Average Download Speed (Mbit/s)",
    y = "Attainment 8 Score"
  ) +
  theme_minimal()
```

Figure 47: line of best fit for average download speed and attainment 8 score

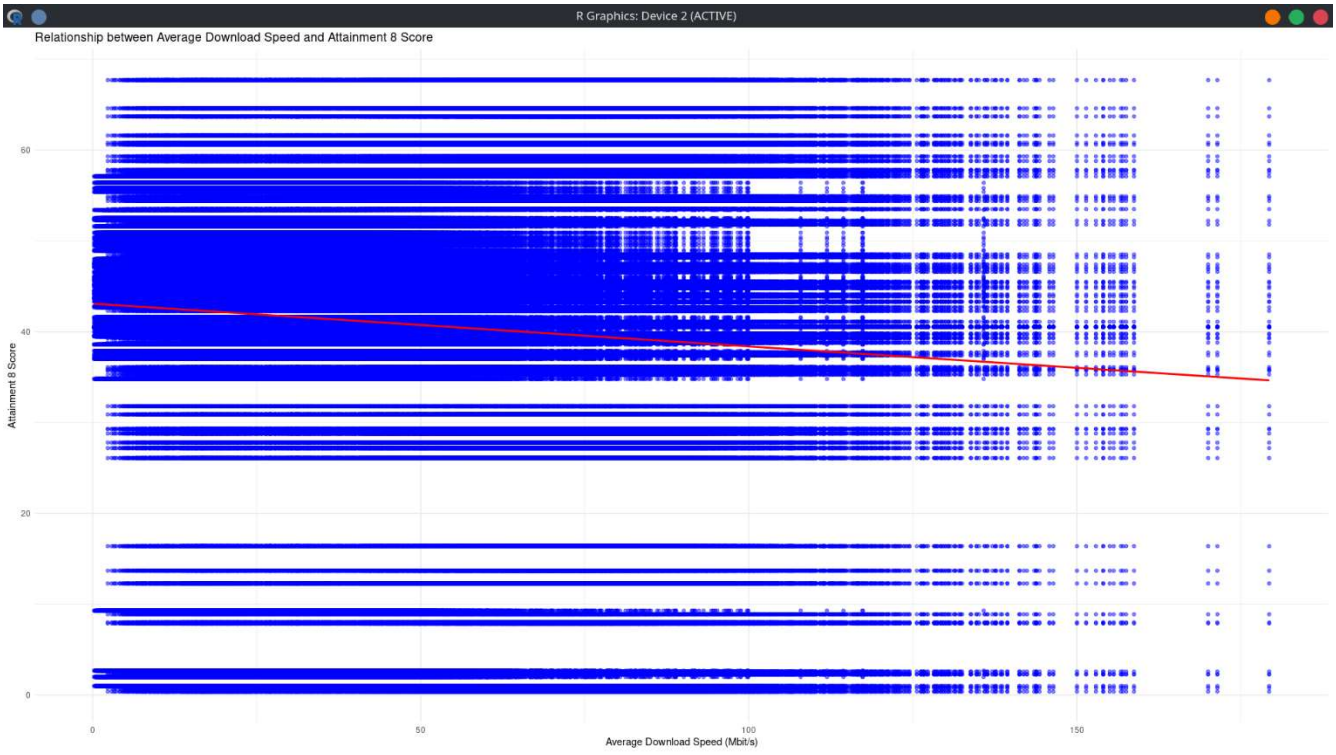


Figure 48: Code to perform city ranking for houses

```r
house_pricing_dataset <- read_csv("Cleaned_datasets/House_Pricing_Data/clean_house_pricing_data.csv")
house_rankings <- house_pricing_dataset %>%
  group_by(County, Town.City) %>%
  # Calculate percentile values for price distribution
  summarize(
    p25 = quantile(Price, 0.25, na.rm = TRUE),
    p50 = quantile(Price, 0.50, na.rm = TRUE),
    p75 = quantile(Price, 0.75, na.rm = TRUE),
    skewness = (mean(Price, na.rm = TRUE) - median(Price, na.rm = TRUE)) / sd(Price, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  # Calculate a score based on your formula
  mutate(score = p25 * 0.25 + p50 * 0.35 + p75 * 0.4 - skewness * 1000) %>%
  # Rank cities within each county based on score
  group_by(County) %>%
  arrange(desc(score)) %>%
  mutate(rank = row_number()) %>%
  # Filter for top 10 cities in each county
  filter(rank <= 10) %>%
  ungroup()

# Print the result
print(house_rankings)
```

Figure 49: Code to perform city ranking by broadband

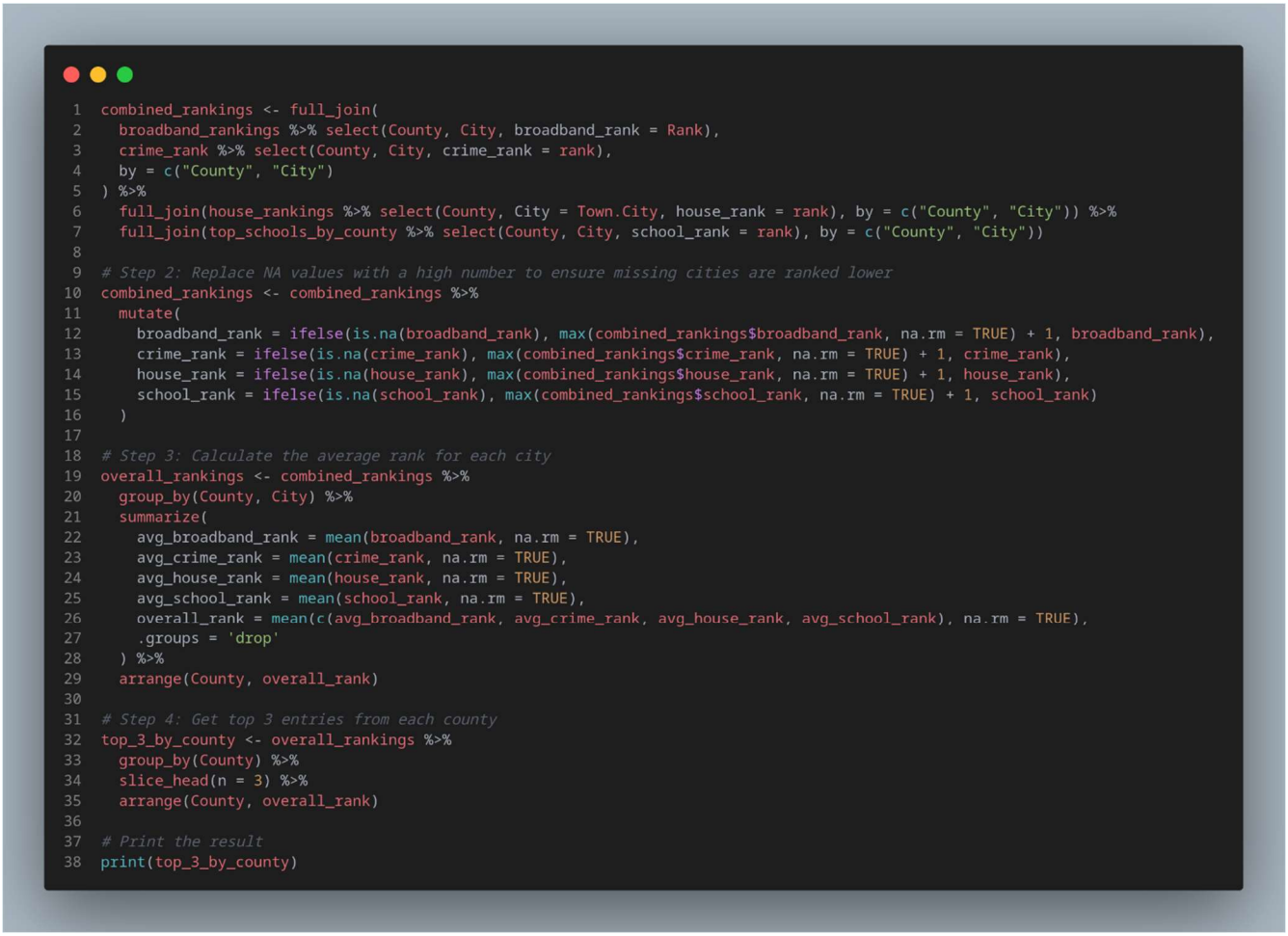
```r
broadband_coverage_dataset <- read_csv("Cleaned_datasets/Broadband/clean_broadband_coverage.csv")
broadband_performance_dataset <- read_csv("Cleaned_datasets/Broadband/clean_broadband_performance.csv")



# Normalize the data
normalize <- function(x) {
  (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))
}

# Merge the datasets by PostalCode
combined_dataset <- broadband_coverage_dataset %>%
  inner_join(broadband_performance_dataset, by = c("PostalCode", "oa11cd", "Street", "City", "County"))


combined_dataset <- combined_dataset %>%
  mutate(
    SFBB_norm = normalize(`SFBB availability (% premises)`),
    UFBB_norm = normalize(`UFBB availability (% premises)`),
    FTTP_norm = normalize(`FTTP availability (% premises)`),
    MaxDownload_norm = normalize(`Maximum download speed (Mbit/s)`),
    AvgDownload_norm = normalize(`Average download speed (Mbit/s)`),
    Weighted_Score = SFBB_norm * 0.2 + UFBB_norm * 0.25 + FTTP_norm * 0.25 +
                     MaxDownload_norm * 0.15 + AvgDownload_norm * 0.15
  )
```

Figure 50: Code to perform city ranking via schools

Figure 51:Code to perform city ranking via crime

```r
crime_dataset <- read_csv("Cleaned_datasets/Crime/Crime_Data_Combined.csv")

crime_totals <- crime_dataset %>%
  group_by(County, City, Crime) %>%
  summarize(total_offences = n(), .groups = 'drop') %>%
  # Pivot the data to have total offences of each crime type as columns
  pivot_wider(names_from = Crime, values_from = total_offences, values_fill = 0) %>%
  # Calculate the total offences across all crime types for each city
  rowwise() %>%
  mutate(total_crime_offences = sum(c_across(where(is.numeric)))) %>%
  ungroup() %>%
  # Rank cities within each county based on total crime offences
  group_by(County) %>%
  arrange(total_crime_offences) %>%
  mutate(rank = row_number()) %>%
  # Select top 10 cities with the fewest total crime offences for each county
  filter(rank <= 10) %>%
  ungroup()

# Print the result
print(crime_totals)

crime_rank <- crime_totals %>%
  select(County, City, rank)

```

Figure 52: Code to perform final ranking of cities



```r
combined_rankings <- full_join(
  broadband_rankings %>% select(County, City, broadband_rank = Rank),
  crime_rank %>% select(County, City, crime_rank = rank),
  by = c("County", "City")
) %>%
  full_join(house_rankings %>% select(County, City = Town.City, house_rank = rank), by = c("County", "City")) %>%
  full_join(top_schools_by_county %>% select(County, City, school_rank = rank), by = c("County", "City"))

# Step 2: Replace NA values with a high number to ensure missing cities are ranked lower
combined_rankings <- combined_rankings %>%
  mutate(
    broadband_rank = ifelse(is.na(broadband_rank), max(combined_rankings$broadband_rank, na.rm = TRUE) + 1, broadband_rank),
    crime_rank = ifelse(is.na(crime_rank), max(combined_rankings$crime_rank, na.rm = TRUE) + 1, crime_rank),
    house_rank = ifelse(is.na(house_rank), max(combined_rankings$house_rank, na.rm = TRUE) + 1, house_rank),
    school_rank = ifelse(is.na(school_rank), max(combined_rankings$school_rank, na.rm = TRUE) + 1, school_rank)
  )

# Step 3: Calculate the average rank for each city
overall_rankings <- combined_rankings %>%
  group_by(County, City) %>%
  summarize(
    avg_broadband_rank = mean(broadband_rank, na.rm = TRUE),
    avg_crime_rank = mean(crime_rank, na.rm = TRUE),
    avg_house_rank = mean(house_rank, na.rm = TRUE),
    avg_school_rank = mean(school_rank, na.rm = TRUE),
    overall_rank = mean(c(avg_broadband_rank, avg_crime_rank, avg_house_rank, avg_school_rank), na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  arrange(County, overall_rank)

# Step 4: Get top 3 entries from each county
top_3_by_county <- overall_rankings %>%
  group_by(County) %>%
  slice_head(n = 3) %>%
  arrange(County, overall_rank)

# Print the result
print(top_3_by_county)
```