Prabin Tiwari

Softwarica College of IT and E-Commerce

ST5014CEM Data Science for Developers

Siddhartha Neupane

August 19, 2024

Table of Contents

Table of Figures

Introduction

In the context of helping international friends make an informed property investment decision in the United Kingdom, this report aims to analyze various factors including housing prices, internet connectivity, crime rates, and other significant aspects in the regions of Bristol and Cornwall. This objective is to utilize data science techniques to process and analyze datasets relevant to these factors, ultimately leading to a recommendation of the top towns based on their suitability for investment.

This project involved the complete data mining lifecycle, data collection and cleaning exploratory data analysis (EDA) and the development of a basic recommendation system. The datasets utilized for this project are sourced from the UK government's open data repositories, ensuring their credibility and relevance. The analysis and recommendation system are developed using R language, leveraging libraries such as tidyverse, ggplot2, fmsb. Etc.

Data Cleaning

Data cleaning is a fundamental process in data analysis. It involves dealing with missing numbers, eliminating or fixing inaccurate data, and getting the data ready for analysis. Data cleansing is crucial since it provides accurate analysis, boosts model performance, and assures the validity of recommendations. For this project, we cleaned and worked with the data using R's tidyverse library. An extensive description of the data cleaning procedure is provided below.

Dataset Importing

In the initial phase of the analysis, the required datasets were imported into the environment using R. These datasets included housing prices, broadband speed, crime rates, and school data all of which are crucial for the analysis and recommendations. The “read.csv()” function in R was employed to load these datasets efficiently, enabling subsequent data manipulation and cleaning. Each dataset provided specific insights, housing prices offered average property costs across towns in Bristol and Cornwall, broadband speed data highlighted internet availability, crime rates detailed the frequency of various offenses, and school data provided metrics on educational performance.

Cleaning Datasets

After importing the datasets, the data cleaning process was undertaken to ensure the datasets were accurate and ready for analysis. This process involved handling missing values by filling gaps with appropriate substitutes, such as the median or mean values. Data normalization was applied to conform the datasets to the third normal form(3NF), reducing redundancy and enhancing data integrity. Additionally, categorical data was transformed into numerical formats for ease of analysis, and outliers were addressed using statistical methods like the interquartile

range (IQR) and Z-scores, either by capping or removing them. Finally, the cleaned datasets were integrated into a unified model, ensuring that each town was represented as a unique entity with associated attributes from all datasets.

Housing Dataset Cleaning

In this dataset cleaning, the primary objective was to clean and prepare the housing dataset for analysis. Essential columns such as Postcode and Town/City were extracted. The Postcode was standardized by trimming it to the first six characters to ensure consistency with other datasets. This cleaned data was subsequently used to join with cleaned crime dataset.

Broadband Dataset Cleaning

For the Broadband Dataset, the aim was to clean broadband availability and speed data. The dataset was filtered to focus on relevant geographic areas, such as Bristol and Cornwall. Column names and formats were standardized, duplicates and irrelevant rows were removed, and the data was merged with other datasets as needed, particularly by postcode.

Crime Dataset Cleaning

In the Crime Dataset cleaning process, the focus was on the data from Bristol and Cornwall for the year 2022 to 2024. Monthly crime datasets for each year and location were combined using "rbind" and the dataset was converted to a Tibble for better handling. Essential columns, including Month, LSOA code and Crime type, were selected. Duplicates based on LSOA code were removed, and the dataset was merged with LSOA data to include geographic information like streets and counties. Population data was also added to provide demographic context. Rows with missing values in key columns were filtered out to ensure data quality.

School Dataset Cleaning

The School Dataset was prepared by filtering data for schools within relevant geographic regions, cleaning and standardizing data formats, removing duplicates entries and merging other datasets using geographical identifiers like postcodes.

LSOA Dataset Cleaning

For the LSOA Dataset, the cleaning process involved selecting and renaming columns for consistency such as lsoa11cd, lsoa11nm, and pcds. To LSOA code, street, counties and postcode respectively. The data was filtered to retain only relevant areas like Bristol and Cornwall, and postcodes were standardized. Duplicates data were checked and removed.

Population Dataset Cleaning

The Population Dataset was cleaned by renaming columns for clarity e.g., renaming the postcode column to “postcode” and population count to “count” and postcodes were standardized to ensure proper joining with other datasets. This cleaned population data was used in conjunction with other datasets, such as crime data, to provide contextual information on population density.

Exploratory Data Analysis (EDA)

In the Exploratory Data Analysis (EDA) phase, the primary goal was to understand the underlying patterns, trends, and relationships within the datasets. This step is crucial in any data analysis project as it allows us to gain insights and prepare the data for further analysis, including modeling. We used various techniques such as summarizing data, visualizing distributions, and identifying outliers to explore the characteristics of the data.

Data Visualization

Data visualization is the process representation of data through use of common graphics, such as charts, graphs or diagrams to make information more accessible and easier to understand. These techniques enable us to grasp the insights quickly that might be buried in a raw number by recognizing the patterns and trends in the visual data. Various visualization methods, such as box plots and line graphs are crucial tools in this process to make informed decisions and better understand the factors influencing outcomes like customer behavior and market trends.

In this project, data visualization techniques have been employed effectively to present the cleaned and processed data. By using visual tools like box plots to display data distribution and line graphs to track trends over time. These visualizations were specifically designed to address key questions in our analysis, such as examining the average house prices and identifying trends in crime rates. Making the data both actionable and insightful.

EDA of Housing

House price data representation aims to visually analyze the real estate market by showing trends and variances in property values. For example, bar charts are used to show average property prices in different towns in 2023, and line graphs show price patterns over multiple years within several counties. Additionally, boxplots are used to compare housing prices across counties in 2023. These visualizations give analysts a full picture of the real estate sector, allowing them to detect patterns, analyze market movements, and make informed predictions.

Figure 1:

Bar chart of average house price in 2023

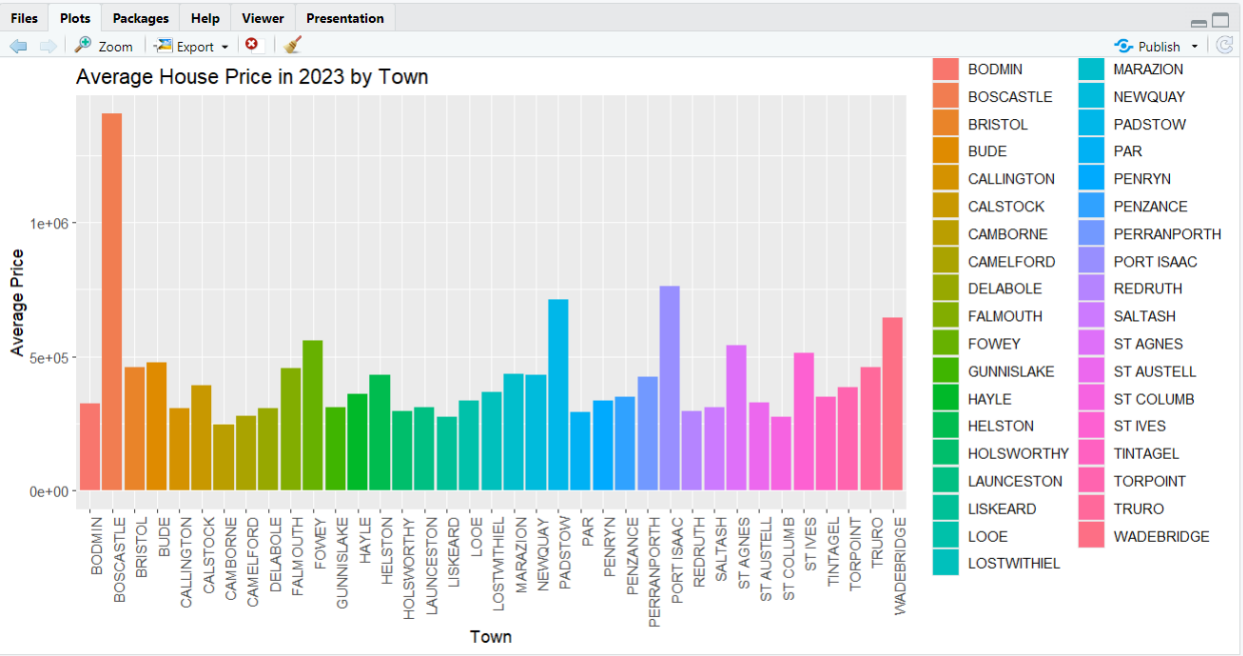


Figure 2:

Box plot of house prices

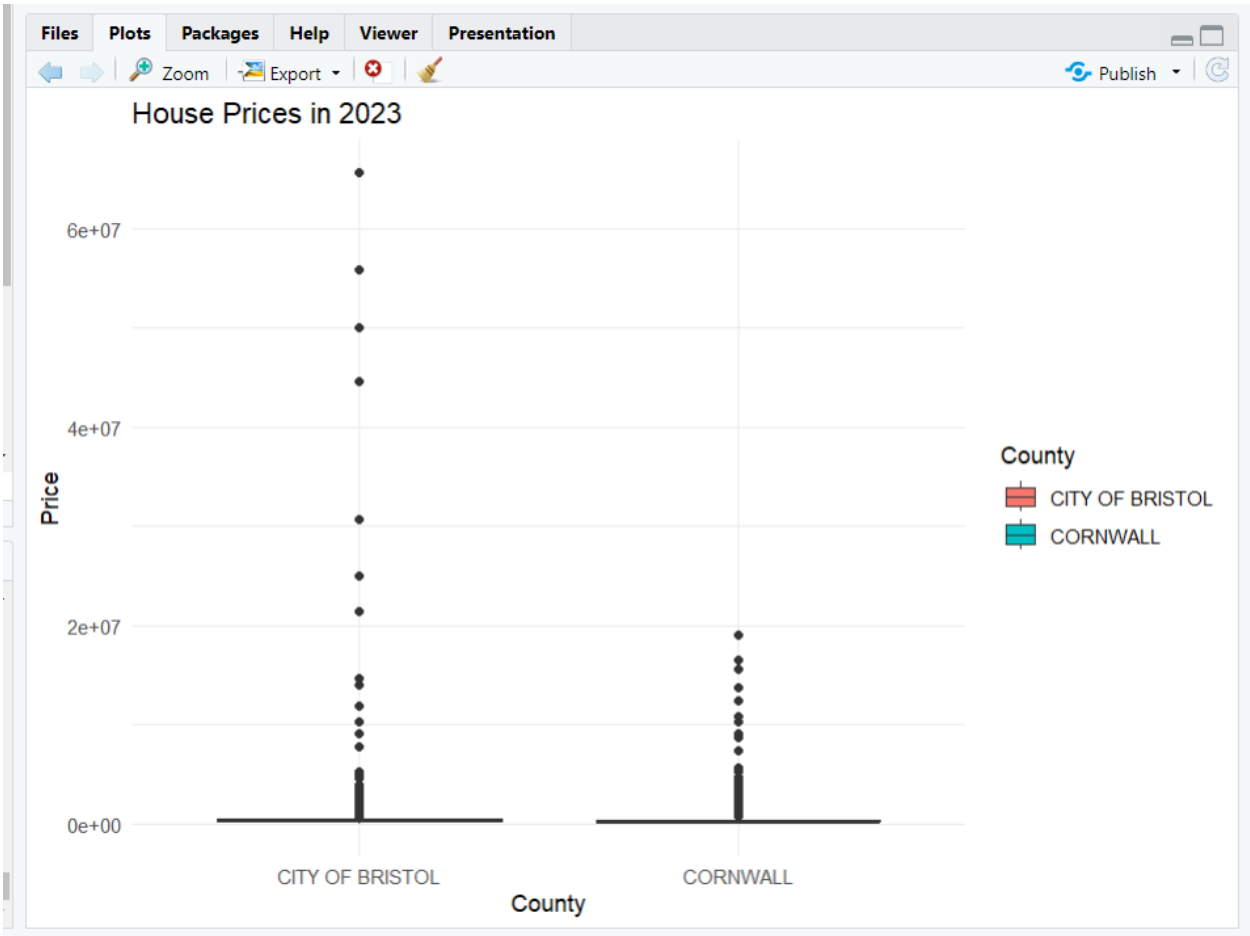
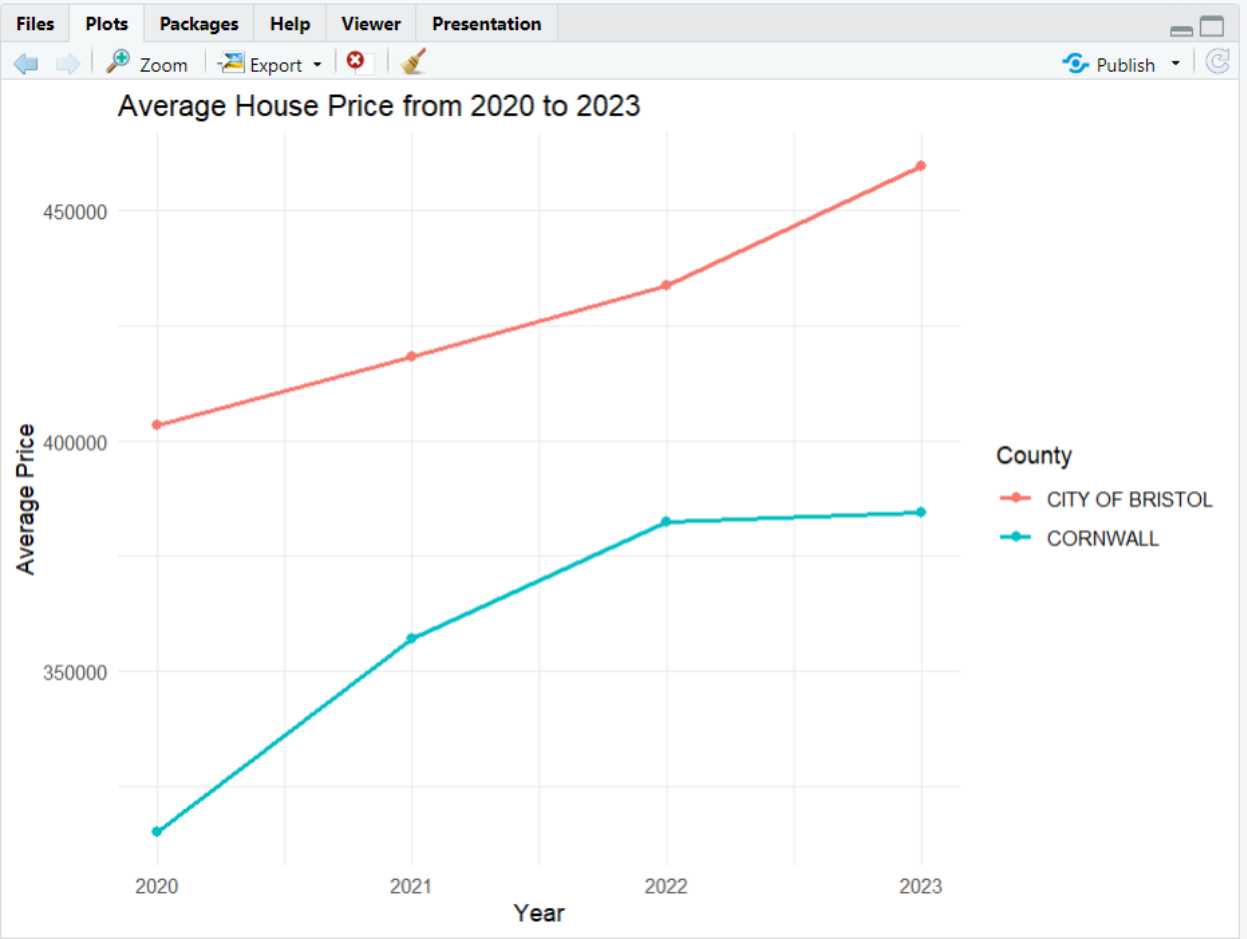


Figure 3:

Line chart of average house price



EDA of Broadband

The broadband speed dataset was analyzed to compare average and maximum download speeds in various cities and counties. A boxplot was created to illustrate the average download speed by county. This graph was useful for comparing internet speeds in different counties. We utilized bar charts to compare average and maximum download speeds in several towns/cities around Cornwall and Bristol. These graphics allowed for a clear comparison of broadband speeds within specified regions.

Figure 4:

Box Plot of Average and Max Download Speed

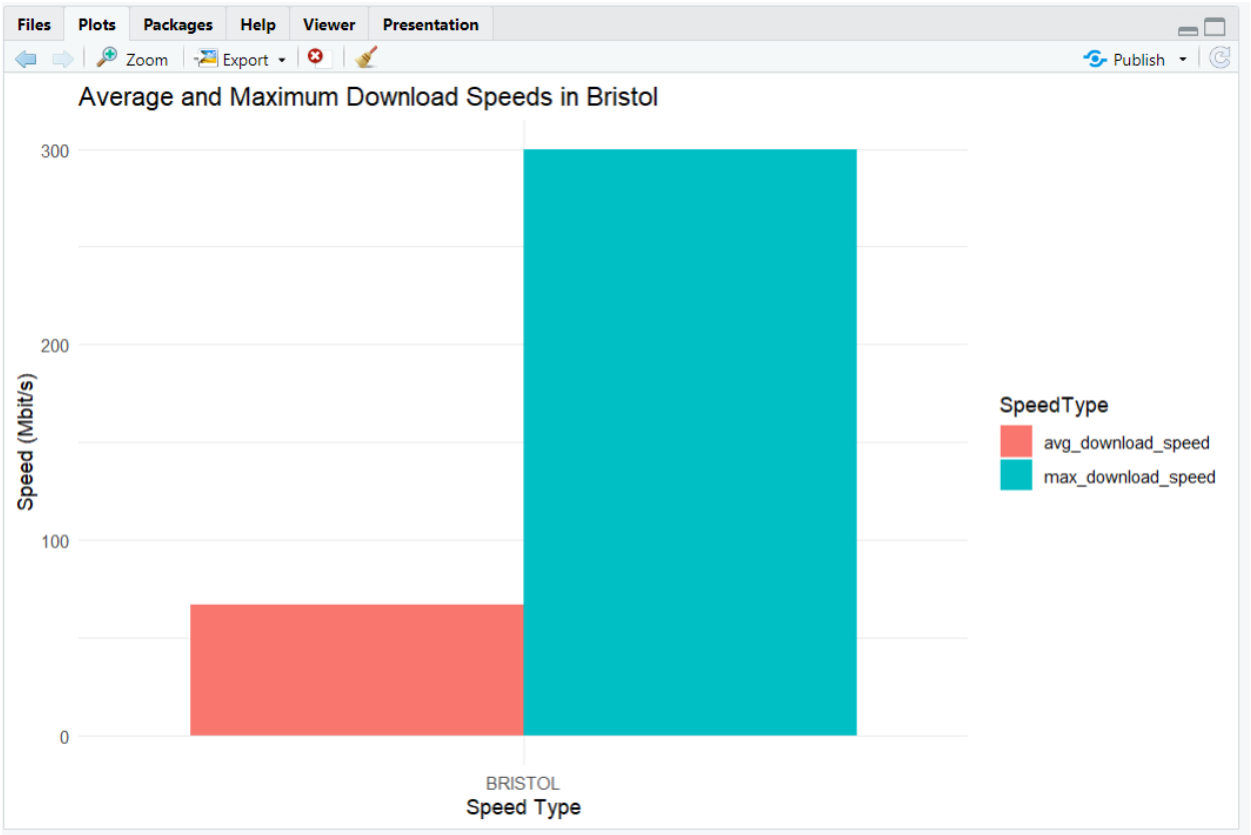


Figure 5:

Bar Chart of Avg and Max Download Speed

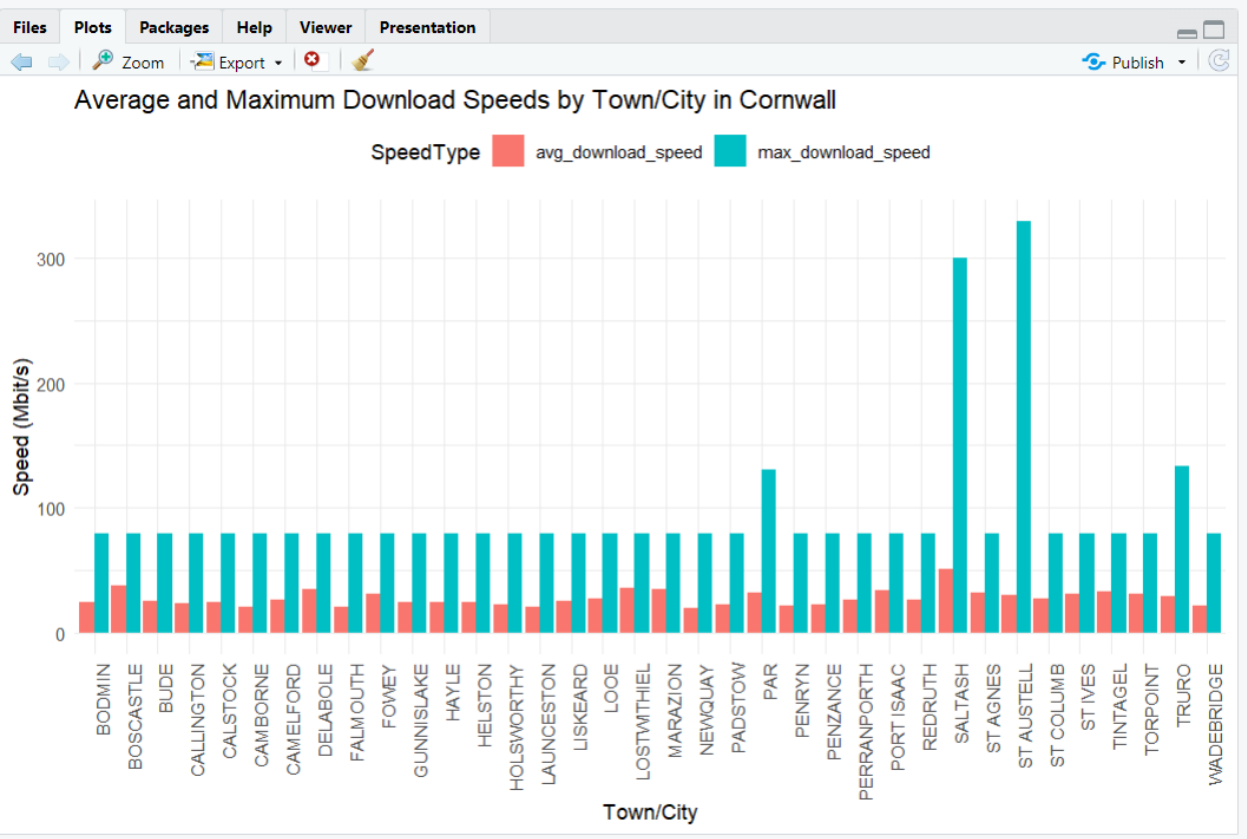
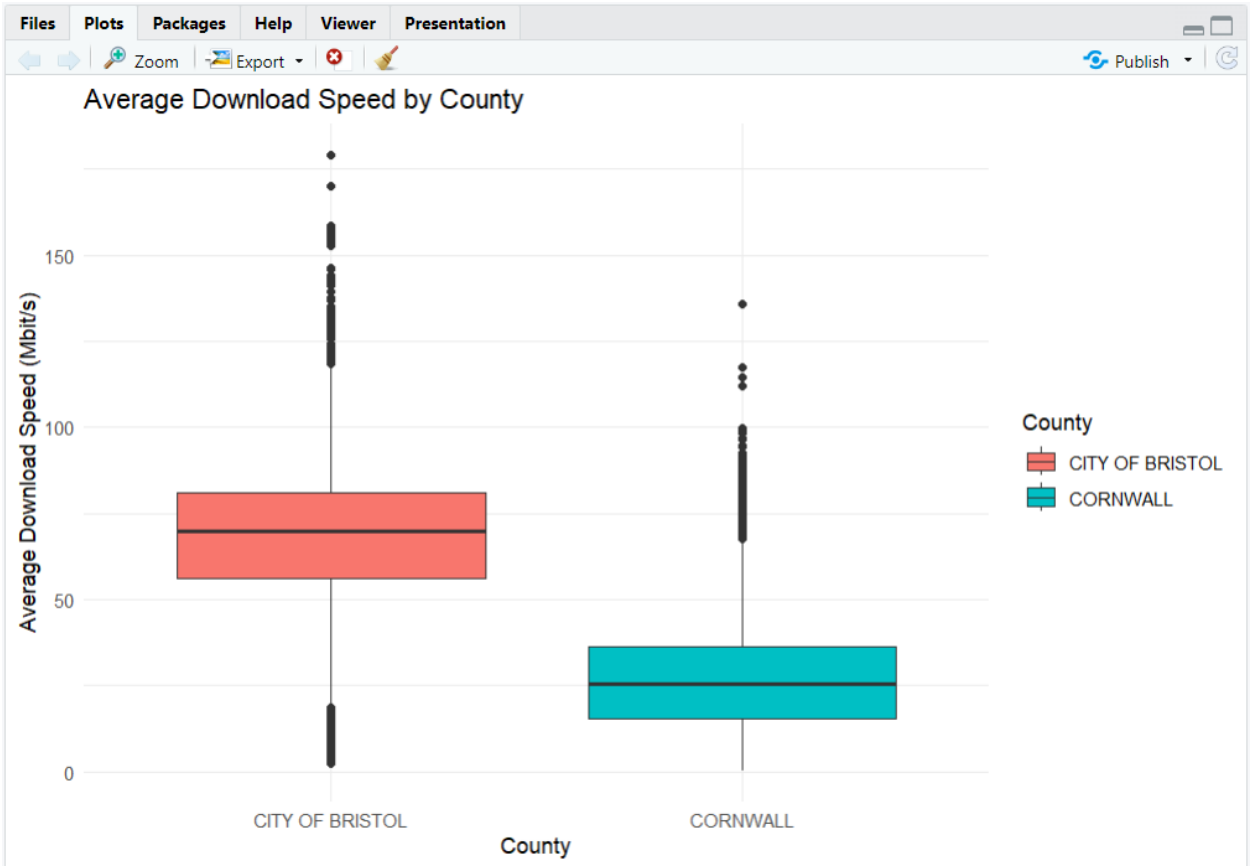


Figure 6:

Box plot of Average Download Speed



EDA of Crime

Crime data representation use visual methods to investigate and analyze crime statistics, focusing on various crime categories and their distribution across counties. For example, radar charts show the pattern of vehicle offenses over time, indicating changes in crime rates. Pie charts depict the distribution of robberies by month in 2023, providing insight into temporal patterns. Additionally, boxplots compare drug offence rates across counties, allowing for a comparative analysis of crime severity.

Figure 7:

Radar Chart of Vehicle Crime Rate

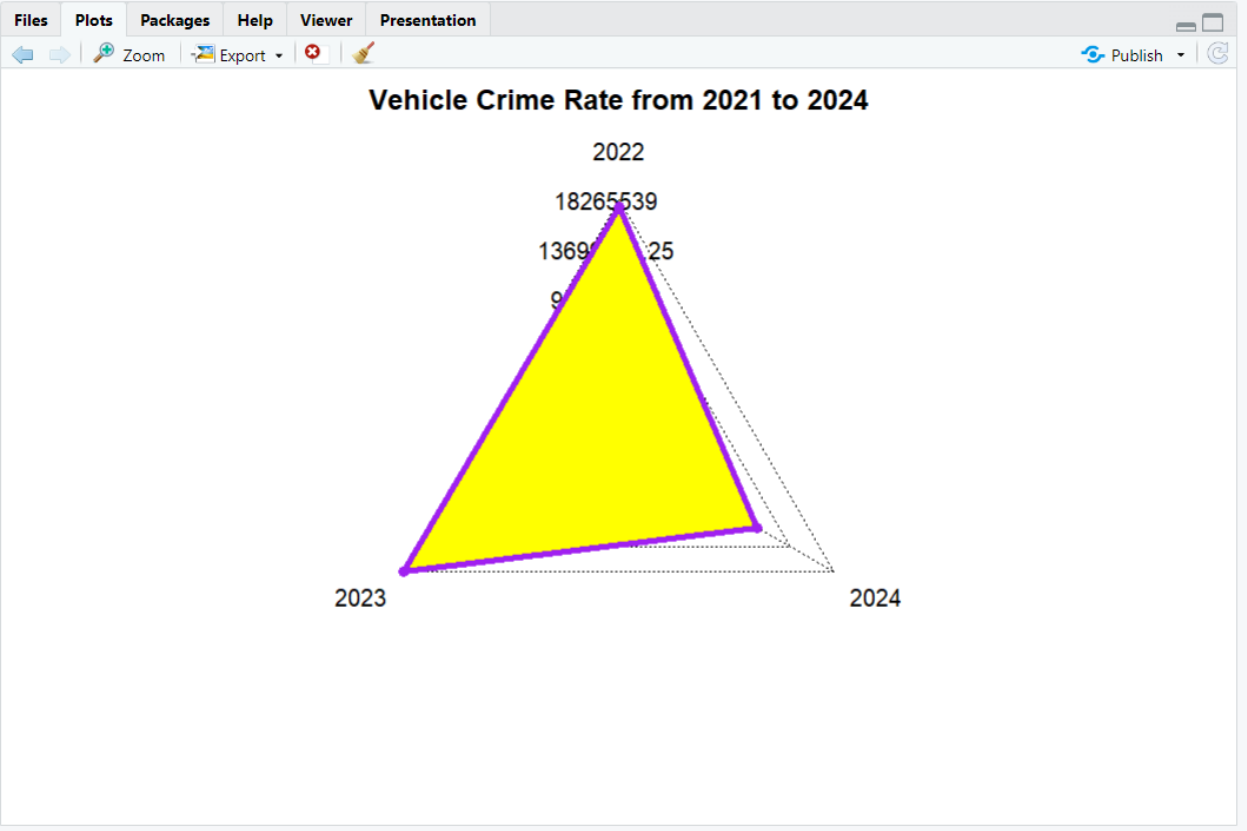


Figure 8:

Pie Chart of Robberies

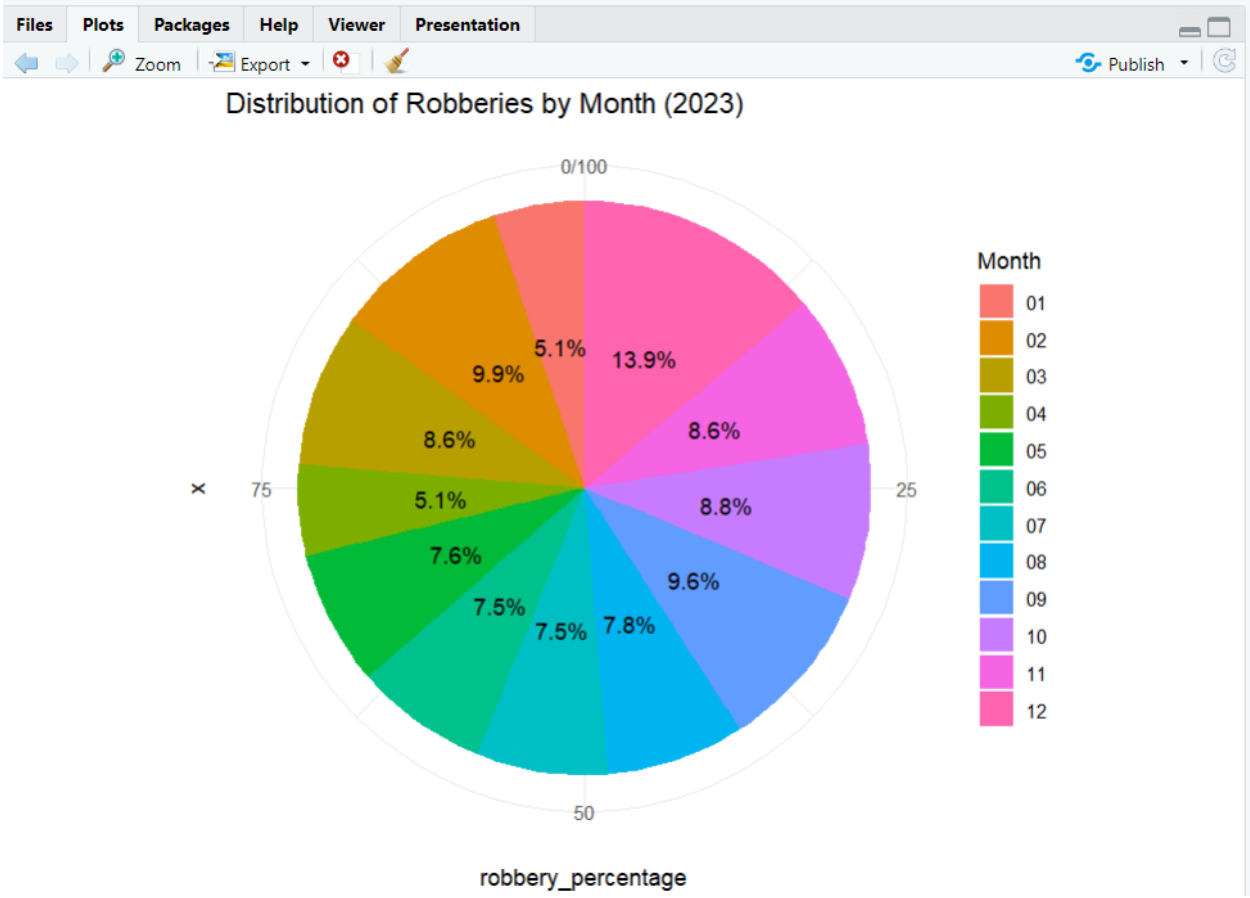
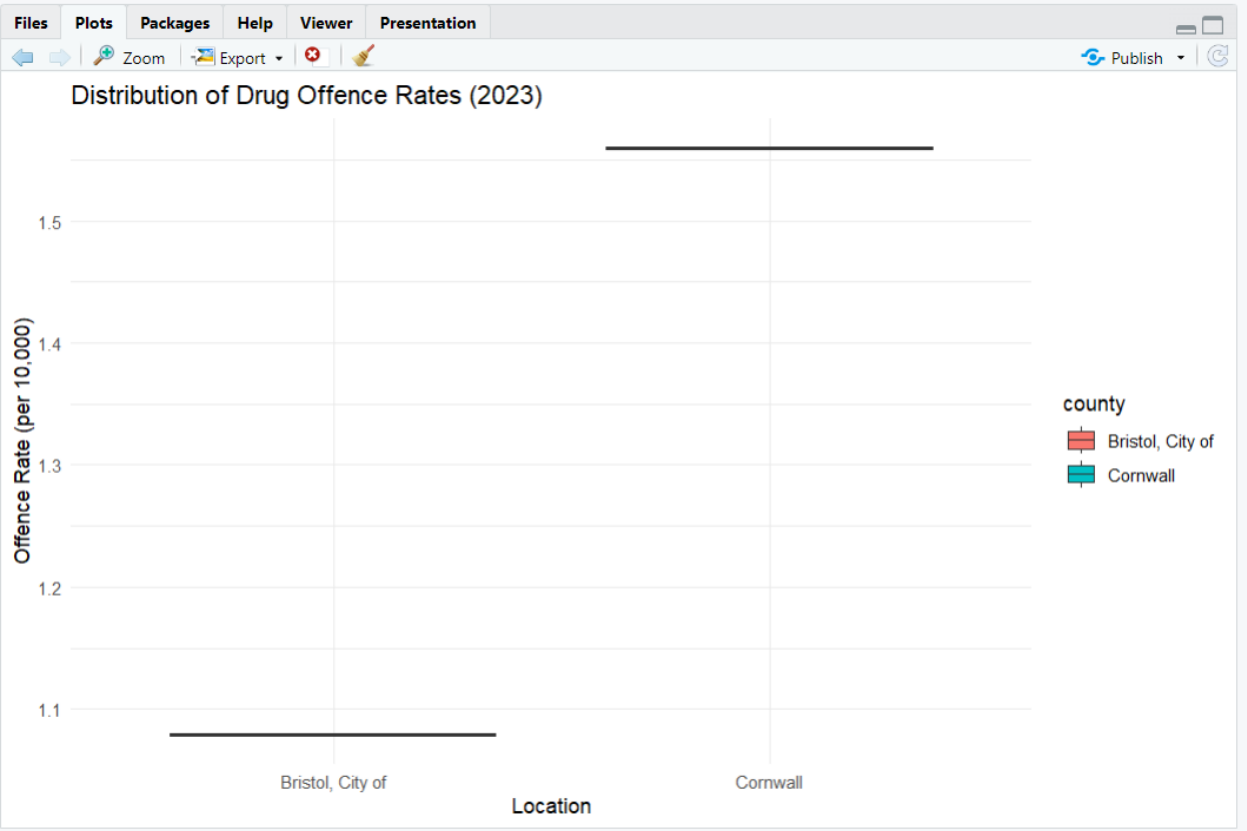


Figure 9:

Box Plot of Drug Offense Rate



EDA of School

School data representation comprises visualizing academic achievement metrics among educational institutions. In this context, boxplots are used to compare the average Attainment 8 scores for schools in various counties in 2022, and line graphs follow these scores over time for specific schools within counties such as Bristol and Cornwall.

Figure 10:

Box Plot of Average Attainment 8 Scores

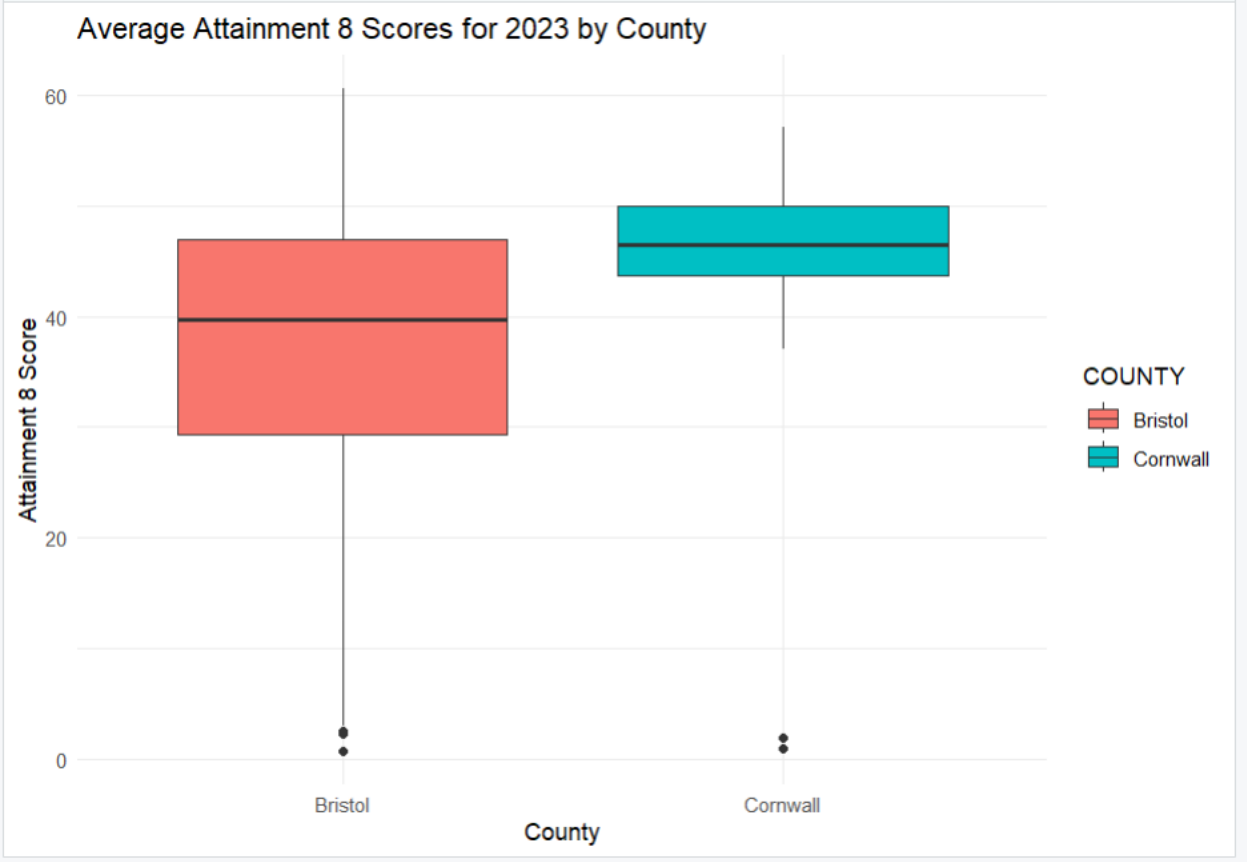


Figure 11:

Average Attainment score of Bristol

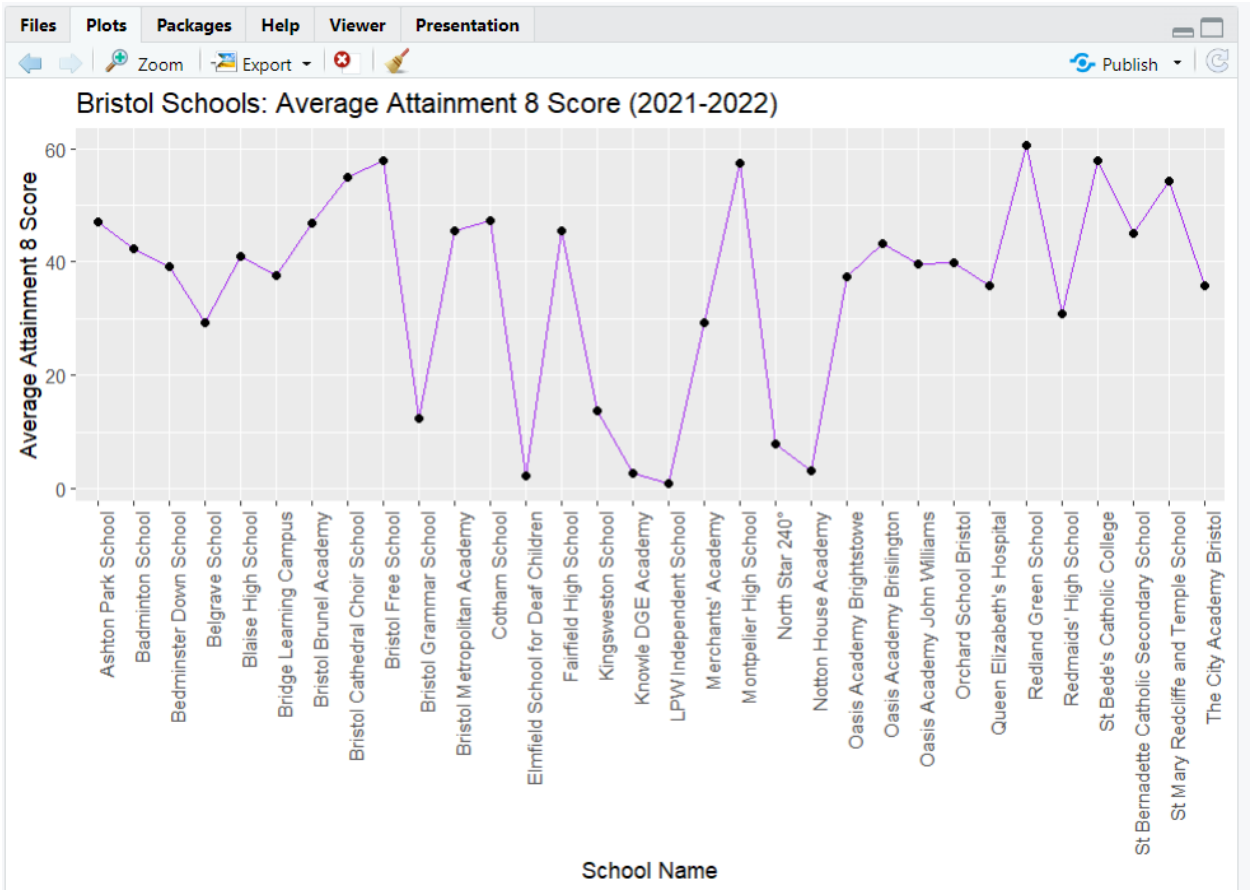
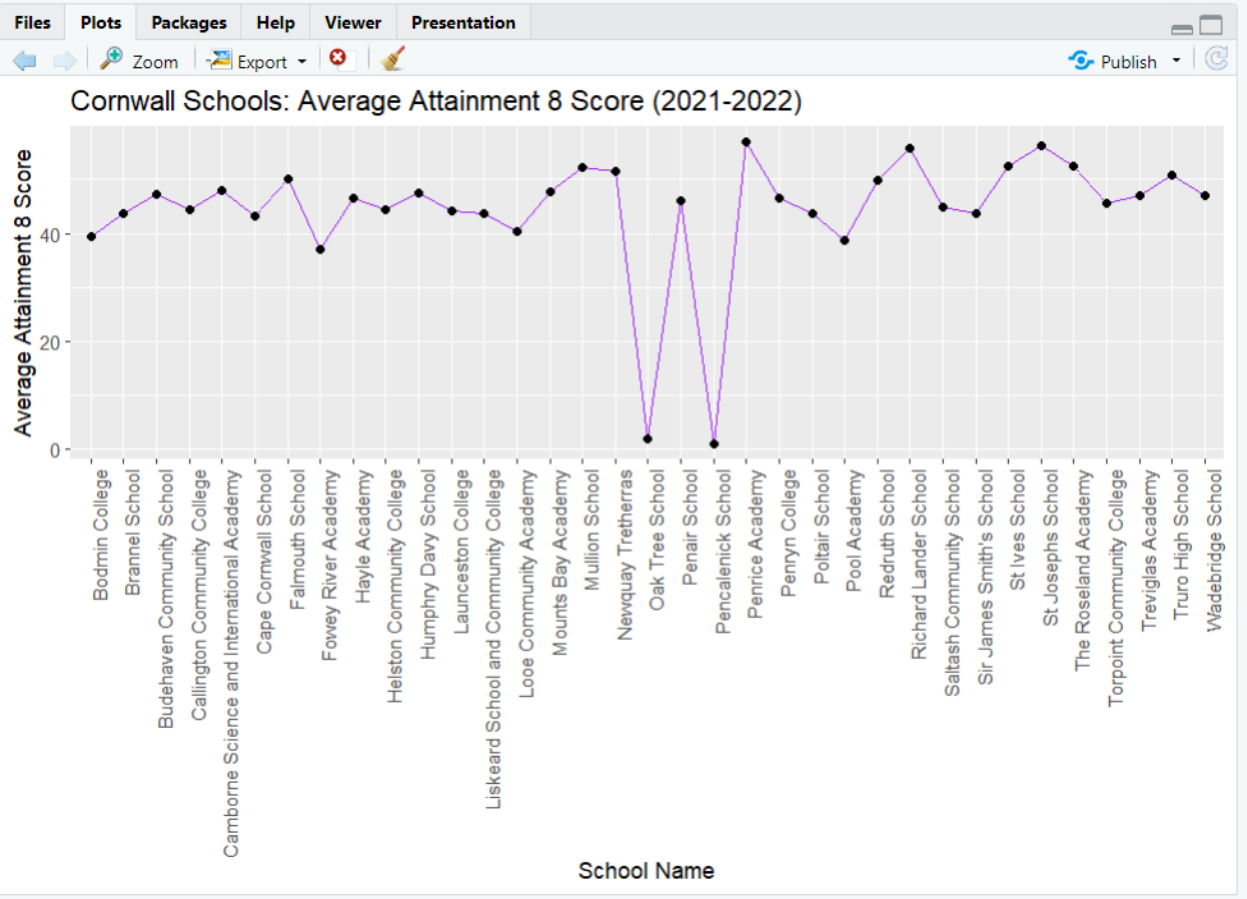


Figure 12:

Average Attainment Score of Cornwall



Linear Model

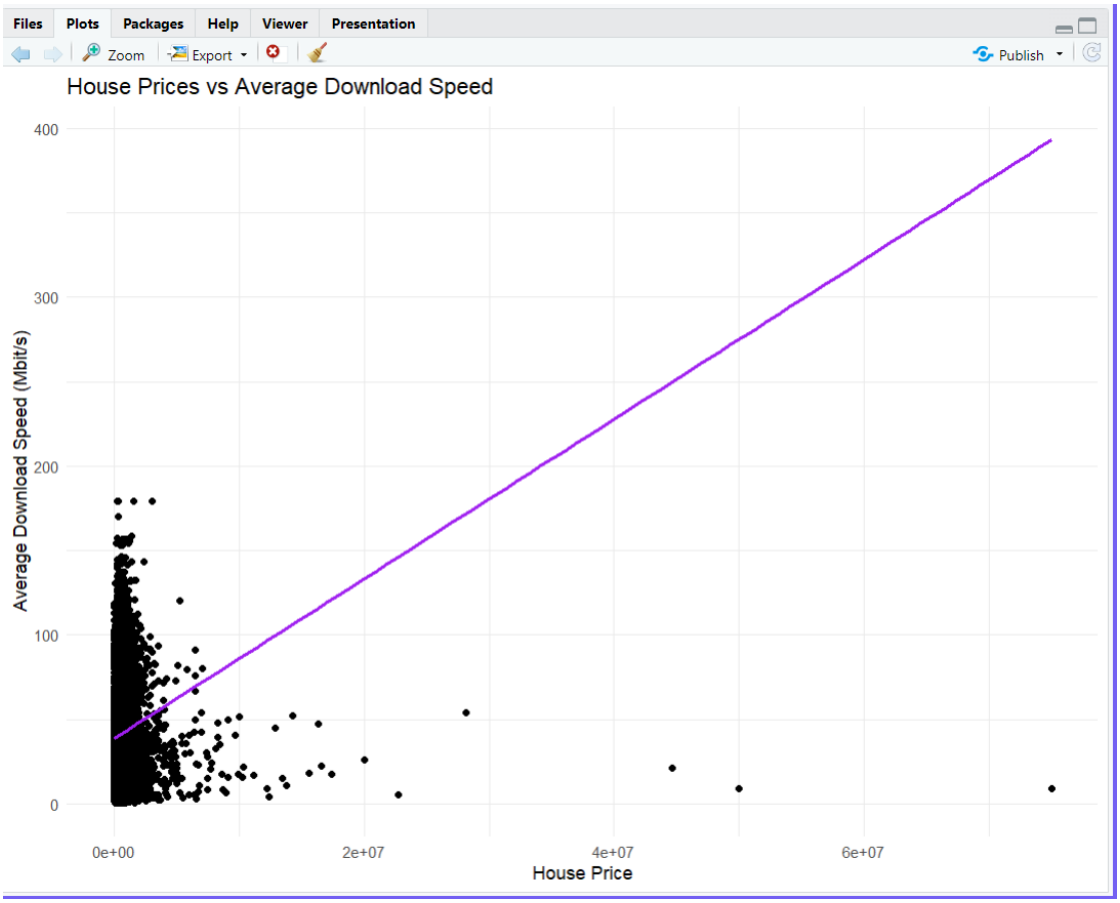
Linear Model also known as “Linear Regression” is a statistical modeling technique used to estimate the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. With one explanatory variable, it is a simple linear regression, while the inclusion of multiple variables is termed multiple linear regression. The model assumes that the relationship between the variables is linear, and it aims to estimate the parameters that best describe this relationship by minimizing the differences between observed and predicted values. Linear regression models are widely used due to their simplicity and the ease with which they can be fitted and interpreted, making them fundamental tools in statistical analysis and predictive modeling.

House Price vs Average Download Speed

In this analysis, the code explores the relationship between house prices and average download speeds. It selects relevant columns from the housing and broadband datasets, merges them by postcode and creates a scatter plot. The plot includes a linear regression line to illustrate the potential influence of internet speed on house prices.

Figure 13:

House Price Vs Average Download Speed

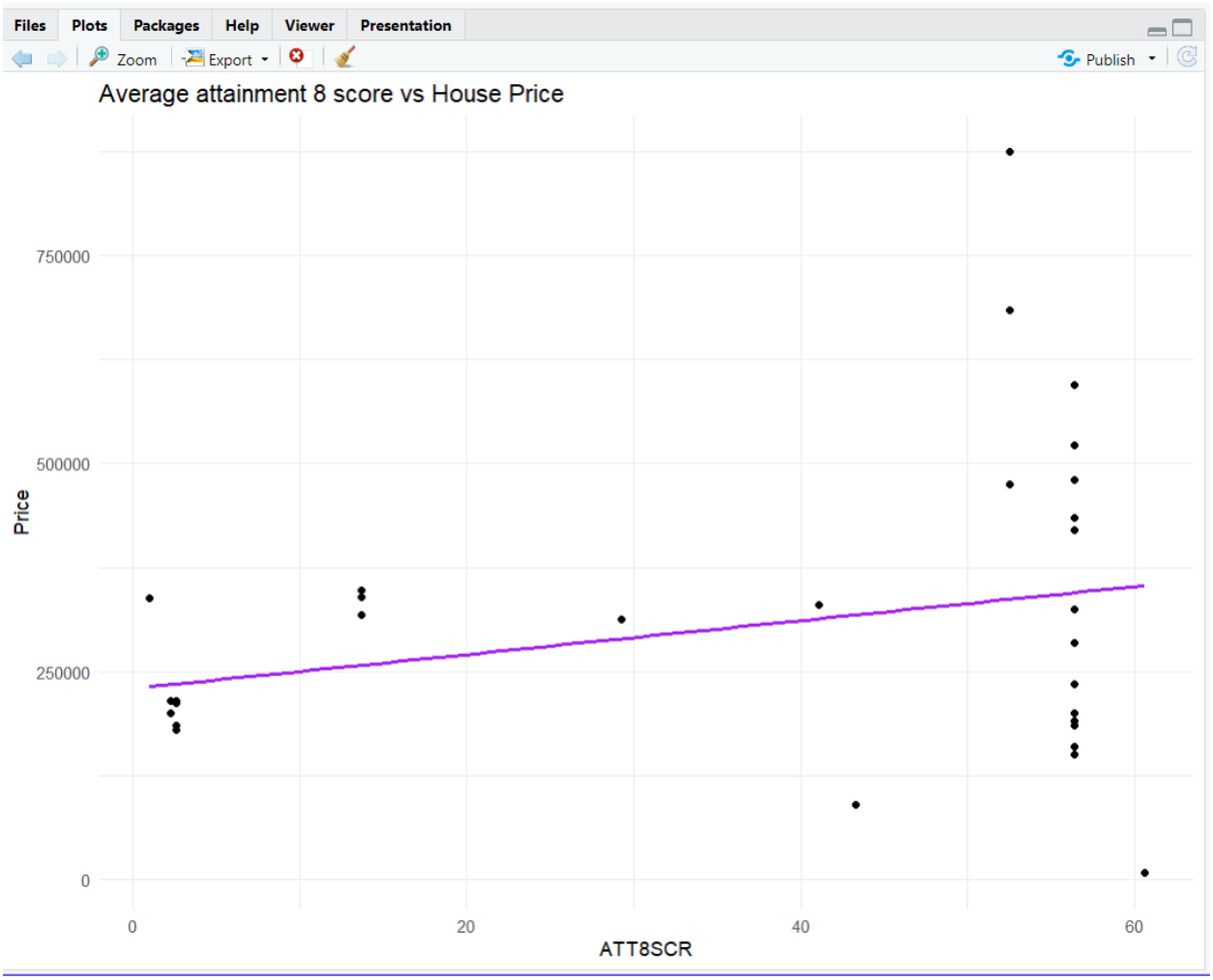


Attainment 8 score VS House Price

This section examines the connection between educational attainment and house prices. It filters the school data for the year 2022 and merges it with housing data by postcode. The resulting scatterplot, enhanced with a linear regression line, which visualizes how the average Attainment 8 score, measure of student performance, correlates with house prices, potentially indicating the value of living in areas with better educational outcomes.

Figure 14:

Attainment 8 Score Vs House Price

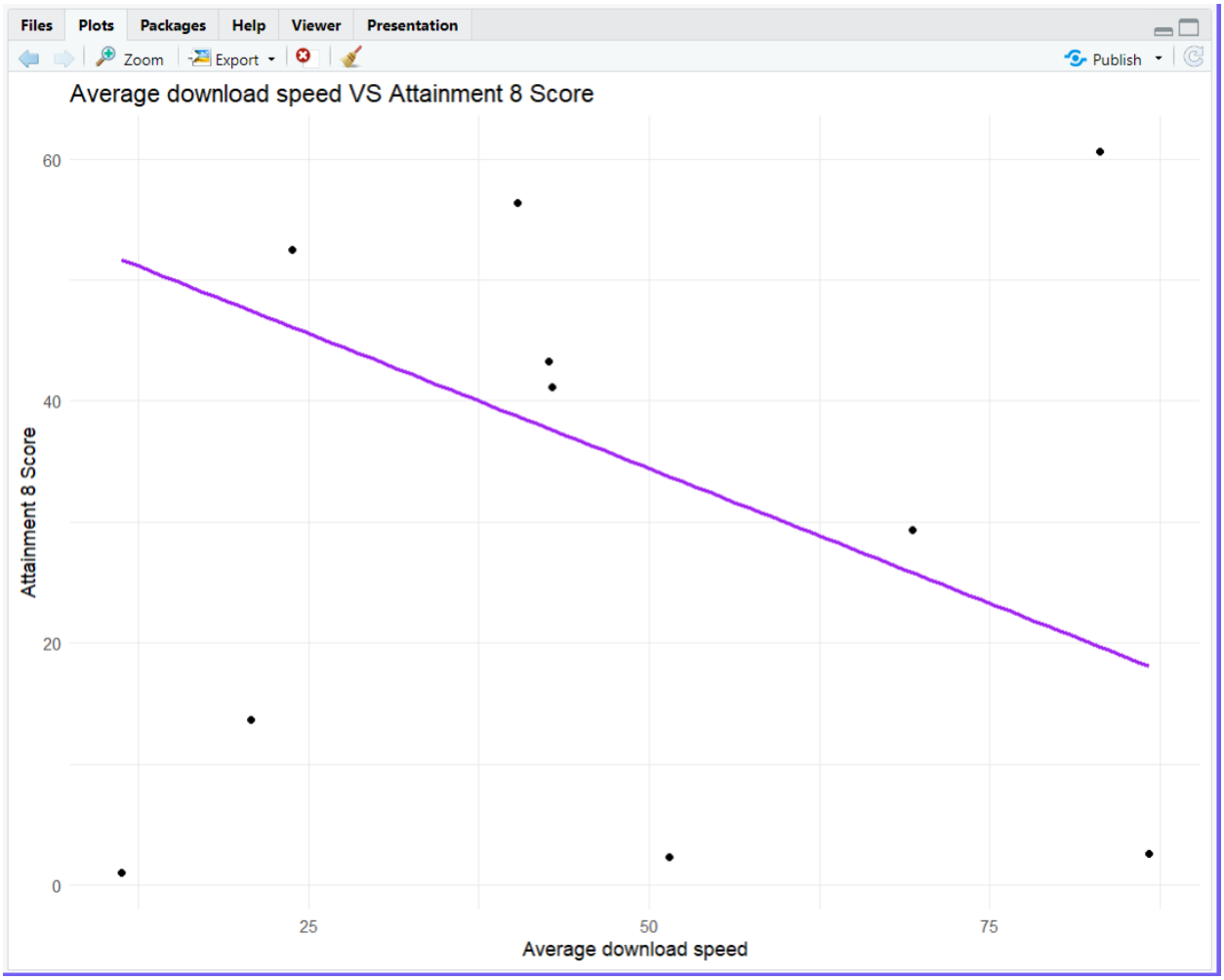


Average Download Speed VS Attainment 8 score

The analysis examines whether there was a relationship between broadband speed and educational performance. It merges the broadband data with school data for 2022 by postcode and generates scatterplot. The plot includes a linear regression line which helps to assess whether faster download speeds are associated with higher Attainment 8 scores, exploring the impact of internet access on education.

Figure 15:

Download Speed Vs Attainment 8 Score

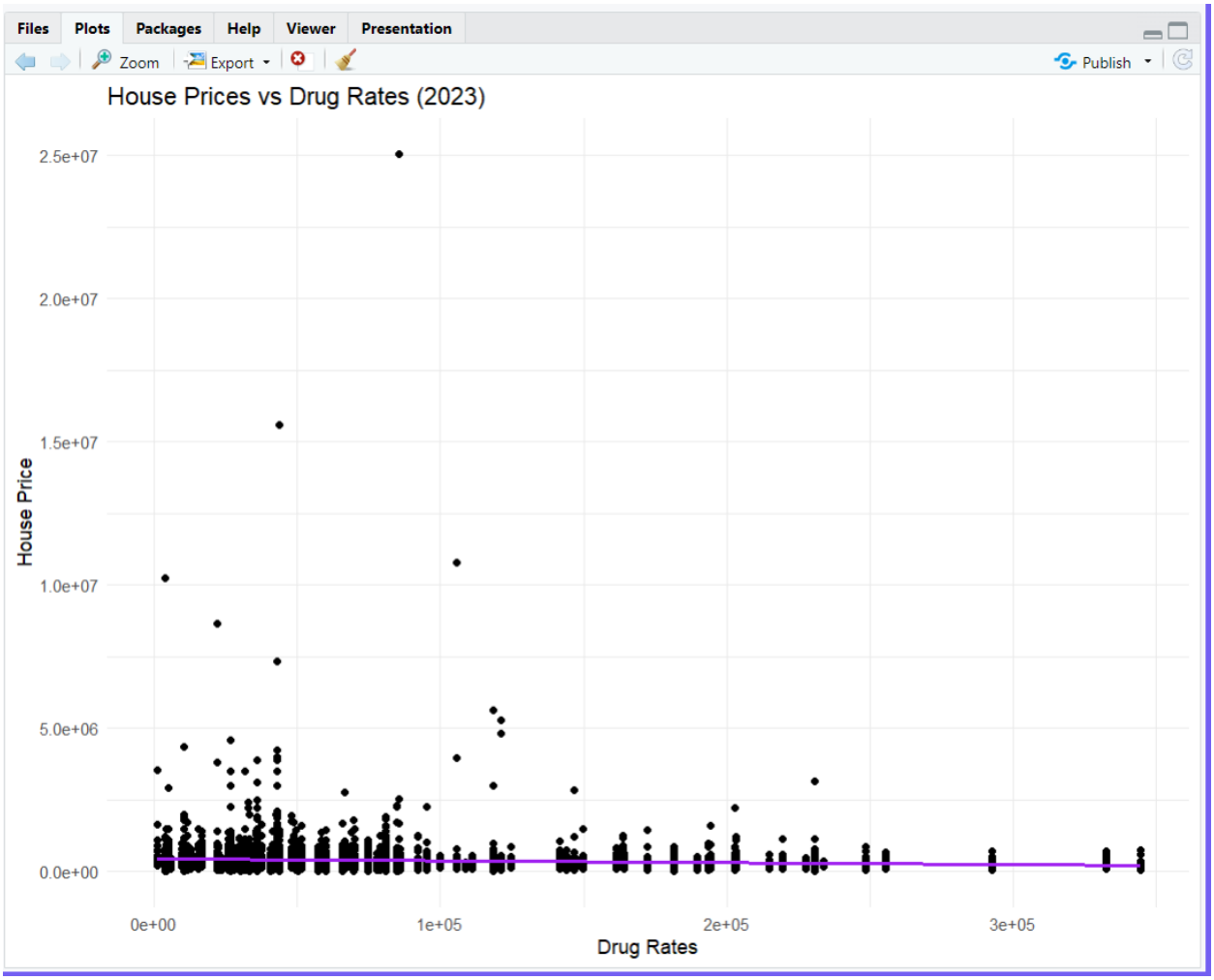


House Price vs Drug Rates

In this analysis, the focus was on how drug crime rates affect house prices. The code filters and aggregates crime data for drug-related offenses in 2023, then merges it with housing data by postcode. A scatter plot with a linear regression line is created to examine whether higher drug crime rates in an area are linked to lower house prices, reflecting the impact on property values.

Figure 16:

House Price Vs Drug Rates



Recommendation System

House Price Ranking

The housing recommendation system identifies the town with the highest average housing price, which is Camborne. The system ranks towns based on the normalized housing prices, with the highest normalized score indicating the most expensive area. The recommendation is made by analyzing the housing data, filtering it to include only relevant fields like the average price for the year 2023, and then ranking the towns based on the normalized values. Camborne tops the list with an average price of £245368.7 and a normalized price score of 1.0000000, making it the most expensive town for housing in this analysis.

Figure 17:

House Price Ranking

	TOWN	avg_price	norm_price
1	CAMBORNE	245368.7	1.0000000
2	LISKEARD	275131.8	0.9256203
3	CAMELFORD	277289.9	0.9202270
4	REDRUTH	294412.6	0.8774366
5	CALLINGTON	306231.3	0.8479010
6	SALTASH	308047.3	0.8433626
7	LAUNCESTON	310652.7	0.8368516
8	BODMIN	322766.0	0.8065798
9	ST AUSTELL	325716.0	0.7992077
10	PENRYN	334556.3	0.7771151
11	LOOE	335359.1	0.7751090
12	PENZANCE	349984.7	0.7385587
13	HAYLE	359057.7	0.7158847
14	TORPOINT	384276.0	0.6528629
15	HELSTON	430092.1	0.5383658
16	FALMOUTH	455765.9	0.4742057
17	BRISTOL	459828.8	0.4640524
18	BUDE	475364.3	0.4252281
19	ST IVES	512576.7	0.3322324
20	FOWEY	558521.7	0.2174133
21	WADEBRIDGE	645519.8	0.0000000

Broadband Speed Ranking

The broadband recommendation system ranks towns based on average download speeds. The recommendation identifies Bristol as the top town, with an average download speed of 66.75 Mbit/s and a normalized download speed score of 1.0000000. This ranking is determined by grouping the broadband data by town, calculating the mean download speed, and then normalizing these values. Bristol's high-speed internet makes it the top recommendation for broadband.

Figure 18:

Broadband Ranking

	TOWN	avg_down_speed	norm_download_speed
1	BRISTOL	66.75668	1.000000000
2	SALTASH	51.58548	0.671416675
3	FOWEY	31.64851	0.239614721
4	ST IVES	31.51296	0.236678986
5	TORPOINT	30.95377	0.224567804
6	ST AUSTELL	29.99783	0.203863749
7	LOOE	27.81727	0.156636342
8	CAMELFORD	26.95526	0.137966605
9	REDRUTH	26.26193	0.122950360
10	BUDE	25.91393	0.115413196
11	LISKEARD	25.78350	0.112588262
12	HELSTON	25.10586	0.097911817
13	BODMIN	25.06855	0.097103584
14	HAYLE	24.42739	0.083217251
15	CALLINGTON	23.58414	0.064953730
16	PENZANCE	22.59383	0.043505221
17	PENRYN	22.22517	0.035520668
18	WADEBRIDGE	21.66013	0.023282925
19	LAUNCESTON	20.76376	0.003868954
20	CAMBORNE	20.69015	0.002274762
21	FALMOUTH	20.58512	0.000000000

Crime Score Ranking

For crime rates, the system recommends Fowey as the town with the highest crime rate, with a crime rate of 98 and a normalized crime rate score of 1.0000000. This recommendation is derived from filtering crime data for the year 2023, calculating the total number of crimes by postcode, and aggregating these values by town. Towns are then ranked based on their normalized crime rates, highlighting Fowey as the area with the most concerning crime statistics.

Figure 19:

Crime Ranking

	TOWN	norm_crimerate	crimerate
1	FOWEY	1.0000000	98
2	CAMELFORD	0.9913297	199
3	LOOE	0.9800841	330
4	CALLINGTON	0.9683235	467
5	WADEBRIDGE	0.9663490	490
6	TORPOINT	0.9634303	524
7	PENRYN	0.9592240	573
8	ST IVES	0.9545884	627
9	HAYLE	0.9478067	706
10	BUDE	0.9420551	773
11	SALTASH	0.9366469	836
12	LISKEARD	0.9131256	1110
13	LAUNCESTON	0.9022234	1237
14	HELSTON	0.8915787	1361
15	BODMIN	0.8552665	1784
16	FALMOUTH	0.8544081	1794
17	CAMBORNE	0.8350931	2019
18	REDRUTH	0.7901966	2542
19	PENZANCE	0.7831574	2624
20	ST AUSTELL	0.7728560	2744
21	BRISTOL	0.0000000	11747

School Score Ranking

The school recommendation system evaluates towns based on their average attainment 8 scores, with St Ives ranking at the top. St Ives has an average attainment 8 score of 52.40 and a normalized score of 1.0000000. This ranking was obtained by grouping school data by town, calculating the mean attainment 8 scores, and normalizing these values to identify the top-performing educational areas.

Figure 20:

School Ranking

	TOWN	av_att8	norm_att8_score
1	ST IVES	52.40000	1.00000000
2	FALMOUTH	50.20000	0.85874068
3	HELSTON	48.40000	0.74316487
4	ST AUSTELL	48.20000	0.73032312
5	CAMBORNE	48.00000	0.71748136
6	BUDE	47.20000	0.66611433
7	WADEBRIDGE	47.10000	0.65969345
8	HAYLE	46.60000	0.62758906
9	PENRYN	46.50000	0.62116819
10	PENZANCE	46.16667	0.59976526
11	TORPOINT	45.60000	0.56338028
12	SALTASH	44.90000	0.51843413
13	CALLINGTON	44.50000	0.49275062
14	REDRUTH	44.20000	0.47348799
15	LAUNCESTON	44.20000	0.47348799
16	CAMELFORD	43.70000	0.44138360
17	LISKEARD	43.60000	0.43496272
18	LOOE	40.40000	0.22949461
19	BODMIN	39.50000	0.17170671
20	FOWEY	37.10000	0.01760563
21	BRISTOL	36.82581	0.00000000

Overall Ranking

The overall recommendation system combines the scores from housing, broadband, crime, and schools to provide a comprehensive ranking of towns. Saltash emerges as the top town with a balanced performance across all categories. It has an average housing price of £30,8074.20, a crime rate of 836, an average attainment 8 score of 44.90, and an average download speed of 51.58548 Mbit/s. The final score of 3.969860 makes Saltash the top overall recommendation for living, considering all the factors analyzed followed by St Ives and St Austell.

Figure 21:

Final Ranking

	TOWN	avg_price	crimerate	av_att8	avg_down_speed	final_score
1	SALTASH	308047.3	836	44.90000	51.58548	3.969860
2	ST IVES	512576.7	627	52.40000	31.51296	3.520322
3	ST AUSTELL	325716.0	2744	48.20000	29.99783	3.387235
4	TORPOINT	384276.0	524	45.60000	30.95377	3.377302
5	CAMELFORD	277289.9	199	43.70000	26.95526	3.202755
6	LISKEARD	275131.8	1110	43.60000	25.78350	2.907460
7	REDRUTH	294412.6	2542	44.20000	26.26193	2.871253
8	HELSTON	430092.1	1361	48.40000	25.10586	2.843046
9	HAYLE	359057.7	706	46.60000	24.42739	2.829539
10	LOOE	335359.1	330	40.40000	27.81727	2.824622
11	BUDE	475364.3	773	47.20000	25.91393	2.805581
12	CALLINGTON	306231.3	467	44.50000	23.58414	2.632246
13	CAMBORNE	245368.7	2019	48.00000	20.69015	2.554849
14	PENRYN	334556.3	573	46.50000	22.22517	2.462829
15	BODMIN	322766.0	1784	39.50000	25.06855	2.415638
16	FOWEY	558521.7	98	37.10000	31.64851	2.404770
17	PENZANCE	349984.7	2624	46.16667	22.59383	2.343645
18	BRISTOL	459828.8	11747	36.82581	66.75668	2.247678
19	LAUNCESTON	310652.7	1237	44.20000	20.76376	2.228909
20	FALMOUTH	455765.9	1794	50.20000	20.58512	2.225749
21	WADEBRIDGE	645519.8	490	47.10000	21.66013	1.790639

Legal and Ethical Issues

The datasets for this analysis were obtained from the official UK government website, ensuring that all data is publicly available and utilized in accordance with legal standards. The materials were obtained with the explicit intention of being used for educational purposes, in accordance with the terms of use specified on the government's website. This method ensures that the analysis is carried out within the legal and ethical parameters, while also protecting data privacy and intellectual property rights. The datasets, which include information on housing, broadband speeds, school performance, and crime rates, are intended solely for academic and research purposes, demonstrating the commitment to appropriate data handling and usage.

Reflection

To ensure the accuracy of the results, data was carefully gathered from trustworthy sources, with a focus on acquiring information directly from the official UK government website. The project used R programming, a well-known data science tool, to carry out the analysis. Recognizing the possibility of inaccuracies in raw data, a strong emphasis was placed on the cleaning process. This involved removing irrelevant, incomplete, or inaccurate data to create polished datasets, which were then stored as .csv files.

Throughout this project, various visualization techniques, such as box plots, bar graphs, and linear graphs, were used to make the cleaned data easier to understand. Linear modeling was employed to explore relationships between different factors, including the connection between house prices and download speeds, as well as crime rates and educational attainment. This analysis helped in categorizing counties by their performance in these areas. In the end, the county with the

most favorable outcomes was identified, and recommendations were made based on this comprehensive comparison.

Conclusion

The integration of several datasets and the use of modern data science techniques were critical in this project. By using R programming for complete data cleaning and visualization, insightful patterns emerged, revealing the complex links between property values, broadband speeds, crime rates, and academic accomplishment. The linear modeling approach provides a quantitative foundation for comprehending these relationships, demonstrating how each element affects county outcomes. This study successfully illustrated how merging numerous data sources can provide a comprehensive understanding of regional dynamics. The findings underline the importance of data science in translating complex facts into actionable insights, directing decision-making, and identifying major predictors of living conditions. Finally, this research provides a strong framework for evaluating and comparing county performance, allowing for better informed, data-driven decisions.

References

- *Price paid data*. (2024, July 26). GOV.UK. https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads
- Takyar, A., & Takyar, A. (2023, May 6). *Exploratory Data Analysis - a Comprehensive guide to EDA*. LeewayHertz - AI Development Company. https://www.leewayhertz.com/what-is-exploratory-data-analysis/
- Wikipedia contributors. (2024, August 9). *Linear regression*. Wikipedia. https://en.wikipedia.org/wiki/Linear_regression
- Zabrodski, A. (2024, June 14). *Introduction to data cleaning in R – dataquest*. Dataquest. https://www.dataquest.io/course/r-data-viz-2/
- Hashemi-Pour, C., Brush, K., & Burns, E. (2024, August 16). *What is data visualization and why is it important?* Business Analytics. https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization
- Hall, D. (2024, June 18). *Data Cleaning: Definition, Techniques & Best Practices for 2024*. TechnologyAdvice. https://technologyadvice.com/blog/information-technology/data-cleaning/

Appendix

GitHub Link: https://github.com/ItachiPrabin/Data-Science

Google Drive Link: https://drive.google.com/drive/folders/1te--i11iZGIE90UxuDT4-3a79klfw7kO?usp=drive_link

Data Cleaning

Figure 22:

House dataset cleaning

```
#housing
col_n=c("ID", "Price", "Year", "Postcode", "Property Type", "Old/New", "Duration", "PAON",
        "SAON", "Street", "Locality", "Town/City", "District", "County", "PPD Category Type")

h20 = read_csv("X:/College projects/4th sem/Data Science/Obtain/Housing/pp-2020.csv", col_names = col_n)
h21 = read_csv("X:/College projects/4th sem/Data Science/Obtain/Housing/pp-2021.csv", col_names = col_n)
h22 = read_csv("X:/College projects/4th sem/Data Science/Obtain/Housing/pp-2022.csv", col_names = col_n)
h23 = read_csv("X:/College projects/4th sem/Data Science/Obtain/Housing/pp-2023.csv", col_names = col_n)

#merging
housing_data = rbind(h20, h21, h22, h23)

#cleaning
clean_housing = housing_data %>%
  filter(County %in% c('CORNWALL', 'CITY OF BRISTOL')) %>%
  select(Price, Year, Postcode, `Town/City`, County) %>%
  mutate(Year = str_trim(substring(Year, 1, 4))) %>%
  filter(!is.na(Price) & !is.na(Year) & !is.na(Postcode) & !is.na(`Town/City`) & !is.na(County))

view(clean_housing)

write_csv(clean_housing, "X:/College projects/4th sem/Data Science/Cleaned/cleanedhousing.csv")
```

Figure 23:

Broadband Dataset Cleaning

```
#broadbandspeed data

broadbandspeed = read_csv("X:/College projects/4th sem/Data Science/Obtain/broadband speed/201809_fixed_pc_r03/201805_fixed_

broadband_sel = broadbandspeed %>%
  select("postcode_space", "Median download speed (Mbit/s)", "Median upload speed (Mbit/s)", "Average upload speed (Mbit/s)"
  filter(!is.na(postcode_space) & !is.na(`Median download speed (Mbit/s)`) & !is.na(`Median upload speed (Mbit/s)`) & !is.na

#view(broadband_sel)
broadjoin = broadband_sel %>%
  left_join(clean_housing %>%
              select(Postcode, `Town/City`, County),
            by = c("postcode_space" = "Postcode"))


broadbandspeed_clean = broadjoin %>%
  filter(!is.na(`Town/City`) & !is.na(County))


write_csv(broadbandspeed_clean, "X:/College projects/4th sem/Data Science/Cleaned/cleanedbroadband_speed.csv")
```

Figure 24:

Crime Dataset Cleaning

```
#selecting only required one
cleancrime = crime_comb %>%
  select(Month, `LSOA code`, `Crime type`, `Falls within`)
cleanlsoa = post_lsoa %>%
  select(`lsoa11cd`, `lsoa11nm`, `ladnm`, `pcds`)

#changing the columns name
colnames(cleanlsoa) = c('LSOA code', 'street', 'counties', "postcode")
colnames(population) = c("postcode", "count")

#filtering the required counties
pl = cleanlsoa %>%
  filter(counties %in% c("Bristol, City of", "Cornwall")) %>%
  mutate(postcode = str_trim((substring(postcode, 1, 6))))

#checking for duplicates
any(duplicated(cleancrime$`LSOA code`))
any(duplicated(pl$`LSOA code`))

#removing the duplicates
cleancrime1=unique(cleancrime, by = "LSOA code")
pl1=unique(pl, by = "LSOA code")

#cleaning & merging
final_cleaned_crime = cleancrime1 %>%
  left_join(pl1, by = "LSOA code", relationship = "many-to-many") %>%
  mutate(Year  = str_trim(substring(Month,  1, 4))) %>%
  mutate(Month  = str_trim(substring(Month,  6, 7))) %>%
  left_join(population, by = "postcode") %>%
  filter(!is.na(`Crime type`) & !is.na(`Month`) & !is.na(`Falls within`) & !is.na(`LSOA code`)& !is.na(`street`) & !is.na(`counties`)

fi_clean_crime = clean_housing %>%
  select(Postcode, `Town/City`) %>%
  mutate(postcode = str_trim(substring(Postcode, 1, 6))) %>%
  left_join(final_cleaned_crime, by="postcode", relationship = "many-to-many")

#view(fi_clean_crime)
write.csv(final_cleaned_crime, "X:/College projects/4th sem/Data Science/Cleaned/cleanedcrime.csv", row.names = FALSE)
#view(final_cleaned_crime)

#
```

Figure 25:

School Dataset Cleaning

```
#School Data
bristol_data21=read_csv("X:/College projects/4th sem/Data Science/Obtain/School/Bristol/2021-2022/801_ks4final.csv")
bristol_data21=read_csv("X:/College projects/4th sem/Data Science/Obtain/School/Bristol/2022-2023/801_ks4final.csv")
cornwall_data21=read_csv("X:/College projects/4th sem/Data Science/Obtain/School/Cornwall/2021-2022/908_ks4final.csv")
cornwall_data22=read_csv("X:/College projects/4th sem/Data Science/Obtain/School/Cornwall/2022-2023/908_ks4final.csv")

bristol_data21 = bristol_data21 %>%
  select(SCHNAME, PCODE, ATT8SCR, TOWN)%>%
  mutate(YEAR=2021,  COUNTY="Bristol")

bristol_data22 = bristol_data21 %>%
  select(SCHNAME, PCODE, ATT8SCR, TOWN)%>%
  mutate(YEAR=2022, COUNTY="Bristol")

cornwall_data21 = cornwall_data21 %>%
  select(SCHNAME, PCODE, ATT8SCR, TOWN)%>%
  mutate(YEAR=2021, COUNTY="Cornwall")

cornwall_data22 = cornwall_data21 %>%
  select(SCHNAME, PCODE, ATT8SCR, TOWN)%>%
  mutate(YEAR=2022, COUNTY="Cornwall")

#combinig school data
combined_bristol = rbind(bristol_data21, bristol_data22)
combined_cornwall = rbind(cornwall_data21, cornwall_data22)

view(combined_bristol)
view(combined_cornwall)

cleaned_bristol = combined_bristol %>%
  filter(!is.na(SCHNAME) & !is.na(PCODE) & !is.na(ATT8SCR)  & !is.na(TOWN)) %>%
  filter(ATT8SCR != "NE" & ATT8SCR != "SUPP")

cleaned_cornwall = combined_cornwall %>%
  filter(!is.na(SCHNAME) & !is.na(PCODE) & !is.na(ATT8SCR) & !is.na(TOWN)) %>%
  filter(ATT8SCR != "NE" & ATT8SCR != "SUPP") %>%
  distinct()

view(cleaned_cornwall)

both = rbind(cleaned_bristol, cleaned_cornwall)
view(both)
dim(both)
write_csv(both, "X:/College projects/4th sem/Data Science/Cleaned/cleanedschool.csv")
```

Graphs

Figure 26:

Housing Dataset

```r
# Load necessary libraries for data manipulation and visualization
library(tidyverse)
library(ggplot2)
library(fmsb)       # For radar charts and related plotting
library(scales)

# Read in the cleaned housing data from a CSV file
housing_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedhousing.csv")
head(housing_data)

# Filter data for years 2020 and onwards, and calculate the average price by Year and Town/City
housing_summary = housing_data %>%
  filter(Year >= 2020) %>%
  group_by(Year, `Town/City`) %>%
  summarise(avg_price = mean(Price))

# Filter the summarized data to include only the year 2023
housing_data_2023 = housing_summary %>%
  filter(Year == 2023)

view(housing_data_2023)  # View the data for 2023 to verify the filtered results

# Create a bar chart for average house prices in 2023, categorized by Town/City
ggplot(housing_data_2023, aes(x = `Town/City`, y = avg_price, fill = `Town/City`)) +
  geom_bar(stat = "identity") +                 # stat = "identity" means heights of bars represent values in the data
  ggtitle("Average House Price in 2023 by Town") +
  ylab("Average Price") +
  xlab("Town") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for better readability

# Create a boxplot to visualize the distribution of house prices in 2023, categorized by County
ggplot(housing_data %>% filter(Year == 2023), aes(x = County, y = Price, fill = County)) +
  geom_boxplot() +
  ggtitle("House Prices in 2023") +
  ylab("Price") +
  xlab("County") +
  theme_minimal()

# Group the data by Year and County
house_years_c = housing_data %>%
  filter(Year >= 2020) %>%
  group_by(Year, County) %>%
  summarise(avg_price = mean(Price))

# Create a line graph showing the trend of average house prices from 2020 to 2023, categorized by County
ggplot(house_years_c, aes(x = Year, y = avg_price, color = County)) +
  geom_line(size = 1) +                         # set line thickness with size
  geom_point(size = 2) +                        # Add points to emphasize data points
  ggtitle("Average House Price from 2020 to 2023") +
```

Figure 27:

Broadband dataset

```r
# Load necessary libraries for data manipulation and visualization
library(tidyverse)
library(ggplot2)
library(fmsb)
library(scales)

# Read in the broadband speed data from a CSV file
broadband_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedbroadband_speed.csv")

# Create a boxplot to compare the average download speeds between counties
ggplot(broadband_data, aes(x = County, y = `Average download speed (Mbit/s)`, fill = County)) +
  geom_boxplot() +
  labs(title = "Average Download Speed by County",
       x = "County",
       y = "Average Download Speed (Mbit/s)") +
  theme_minimal()

# Calculate average and maximum download speeds for each town/city in Cornwall
cornwall_speed = broadband_data %>%
  filter(County == "CORNWALL") %>%
  group_by(`Town/City`) %>%
  summarize(
    avg_download_speed = mean(`Average download speed (Mbit/s)`),
    max_download_speed = max(`Maximum download speed (Mbit/s)`)
  ) %>%
  pivot_longer(cols = c(avg_download_speed, max_download_speed), names_to = "SpeedType", values_to = "Speed")

# Bar chart for average and maximum download speeds by town/city in Cornwall
ggplot(cornwall_speed, aes(x = `Town/City`, y = Speed, fill = SpeedType)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average and Maximum Download Speeds by Town/City in Cornwall",
       x = "Town/City",
       y = "Speed (Mbit/s)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),  # Rotate x-axis labels for better readability
        legend.position = "top")
```

```r
# Calculate average and maximum download speeds for each town/city in Bristol
bristol_speed = broadband_data %>%
  filter(County == "CITY OF BRISTOL") %>%
  group_by(`Town/City`) %>%
  summarize(
    avg_download_speed = mean(`Average download speed (Mbit/s)`),
    max_download_speed = max(`Maximum download speed (Mbit/s)`)
  ) %>%
  pivot_longer(cols = c(avg_download_speed, max_download_speed), names_to = "SpeedType", values_to = "Speed")     # Reshape the dat

# Bar chart for average and maximum download speeds by town/city in Bristol
ggplot(bristol_speed, aes(x = `Town/City`, y = Speed, fill = SpeedType)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average and Maximum Download Speeds in Bristol",
       x = "Speed Type",    # Note: The x-axis label should probably be "Town/City" instead of "Speed Type"
       y = "Speed (Mbit/s)") +
  theme_minimal()
```

Figure 28:

Crime Dataset

```r
# Load necessary libraries for data manipulation and visualization
library(tidyverse)
library(ggplot2)
library(fmsb)
library(scales)

# Read in the crime data from a CSV file
crime_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedcrime.csv")

# Filter data specifically for vehicle crime
vehicle_crime_data = crime_data %>%
  filter(`Crime type` == "Vehicle crime")

# Summarize vehicle crime data by year
vehicle_crime_summary = vehicle_crime_data %>%
  group_by(Year) %>%
  summarise(total_vehicle_crime = sum(count, na.rm = TRUE))

# Prepare data for radar chart (transposing and renaming columns)
radar_chart_data = as.data.frame(t(vehicle_crime_summary$total_vehicle_crime))
colnames(radar_chart_data) = vehicle_crime_summary$Year

# Add max and min rows for proper scaling in the radar chart
radar_chart_data = rbind(rep(max(vehicle_crime_summary$total_vehicle_crime), length(years)),
                         rep(0, length(years)),
                         radar_chart_data)

# Adjust plot margins and create the radar chart
par(mar = c(2, 2, 2, 2))  # Adjust plot margins for better fit

radarchart(radar_chart_data,
           axistype = 1,            # Axis type (1 = clockwise)
           pcol = "purple",
           pfcol = "yellow",
           plwd = 4,                # Line width
           cglcol = "black",
           axislabcol = "black",
           caxislabels = seq(0, max(vehicle_crime_summary$total_vehicle_crime), length.out = 5),
           title = "Vehicle Crime Rate from 2021 to 2024"
)

# Create a pie chart to show the distribution of robberies by month in 2023
robbery_data = crime_data %>%
  filter(`Crime type` == "Robbery" & Year == "2023") %>%
  group_by(Month) %>%
  summarise(robbery_count = n()) %>%
  mutate(robbery_percentage = robbery_count / sum(robbery_count) * 100)
```

```r
# Generate the pie chart with percentage labels
ggplot(robbery_data, aes(x = "", y = robbery_percentage, fill = as.factor(Month))) +
  geom_bar(width = 1, stat = "identity") +       # Use a bar chart in polar coordinates
  coord_polar("y") +                             # Convert bar chart to pie chart
  geom_text(aes(label = paste0(round(robbery_percentage, 1), "%")),
            position = position_stack(vjust = 0.5)) +
  labs(title = "Distribution of Robberies by Month (2023)", fill = "Month") +
  theme_minimal()

# Filter and summarize drug offence data for Cornwall
cornwall_drug_data = drug_offence_data %>%
  filter(counties == "Cornwall") %>%
  distinct(`LSOA code`, .keep_all = TRUE) %>%    # Remove duplicate rows based on LSOA code
  summarise(total_population_cornwall = sum(count),
            total_drug_offences_cornwall = n())

# Filter and summarize drug offence data for Bristol
bristol_drug_data = drug_offence_data %>%
  filter(counties == "Bristol, City of") %>%
  distinct(`LSOA code`, .keep_all = TRUE) %>%
  summarise(total_population_bristol = sum(count),
            total_drug_offences_bristol = n())

# Calculate drug offence rates per 10,000 people for Cornwall
cornwall_drug_data = cornwall_drug_data %>%
  mutate(drug_offence_rate_cornwall = (total_drug_offences_cornwall / total_population_cornwall) * 10000)

# Calculate drug offence rates per 10,000 people for Bristol
bristol_drug_data = bristol_drug_data %>%
  mutate(drug_offence_rate_bristol = (total_drug_offences_bristol / total_population_bristol) * 10000)

# Combine data for Cornwall and Bristol
combined_drug_data = bind_rows(
  cornwall_drug_data %>% mutate(county = "Cornwall"),
  bristol_drug_data %>% mutate(county = "Bristol, City of")
)

# Create a boxplot to compare drug offence rates between Cornwall and Bristol
ggplot(combined_data, aes(x = county, y = offence_rate, fill = county)) +
  geom_boxplot() +
  labs(title = "Distribution of Drug Offence Rates (2023)",
       x = "Location",
       y = "Offence Rate (per 10,000)") +
  theme_minimal()
```

Figure 29:

School Dataset

```
# Load necessary libraries for data manipulation and visualization
library(tidyverse)
library(ggplot2)
library(scales)

# Read in the school data from a CSV file
school_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedschool.csv")

# Filter the school data to only include records from the year 2022
data_2022 = school_data %>%
  filter(YEAR == 2022)

# Create a boxplot to compare Attainment 8 Scores between counties for the year 2022
ggplot(data_2022, aes(x = COUNTY, y = ATT8SCR, fill = COUNTY)) +
  geom_boxplot() +
  labs(title = "Average Attainment 8 Scores for 2023 by County",
       x = "County",
       y = "Attainment 8 Score") +
  theme_minimal()

# Filter the school data for Bristol schools in the year 2021
bristol_attainment = school_data %>%
  filter(YEAR == 2021) %>%
  filter(COUNTY == "Bristol")

# Line graph showing the Attainment 8 Scores for individual schools in Bristol in 2021
ggplot(bristol_attainment, aes(x = SCHNAME, y = ATT8SCR, group = 1)) +
  geom_line(color = "purple") +
  geom_point() +
  labs(title = "Bristol Schools: Average Attainment 8 Score (2021-2022)",
       x = "School Name",
       y = "Average Attainment 8 Score") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate school names on x-axis for readability

# Filter the school data for Cornwall schools in the year 2021
cornwall_attainment = school_data %>%
  filter(YEAR == 2021) %>%
  filter(COUNTY == "Cornwall")

# Line graph showing the Attainment 8 Scores for individual schools in Cornwall in 2021
ggplot(cornwall_attainment, aes(x = SCHNAME, y = ATT8SCR, group = 1)) +
  geom_line(color = "purple") +
  geom_point() +
  labs(title = "Cornwall Schools: Average Attainment 8 Score (2021-2022)",
       x = "School Name",
       y = "Average Attainment 8 Score") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate school names on x-axis for readability
```

Ranking Code

Figure 30:

House Ranking Code

```
# Load necessary libraries
library(tidyverse)

# Load the data
housing_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedhousing.csv")

# Preprocessing for Housing Ranking
house_price = housing_data %>%
  filter(Year == 2023) %>%
  mutate(TOWN = str_trim(toupper(`Town/City`))) %>%
  group_by(TOWN) %>%
  summarise(avg_price = mean(Price)) %>%
  select(avg_price, TOWN) %>%
  na.omit() %>%
  distinct()

# Calculate normalized price
min_price = min(house_price$avg_price)
max_price = max(house_price$avg_price)

houserank = house_price %>%
  mutate(norm_price = 1 - (avg_price - min_price) / (max_price - min_price)) %>%
  arrange(desc(norm_price))

# View the housing ranking
View(houserank)
```

Figure 31:

Broadband ranking code

```r
# Load necessary libraries
library(tidyverse)

# Load the data
broadband_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedbroadband_speed.csv")

# Preprocessing for Broadband Speed Ranking
broadband_sel = broadband_data %>%
  group_by(`Town/City`) %>%
  summarise(
    avg_upl_speed = mean(`Average upload speed (Mbit/s)`),
    avg_down_speed = mean(`Average download speed (Mbit/s)`)
  ) %>%
  mutate(TOWN = str_trim(toupper(`Town/City`))) %>%
  select(TOWN, avg_upl_speed, avg_down_speed)

# Calculate normalized download speed
min_download_speed = min(broadband_sel$avg_down_speed)
max_download_speed = max(broadband_sel$avg_down_speed)

broadbandrank = broadband_sel %>%
  mutate(norm_download_speed = (avg_down_speed - min_download_speed) / (max_download_speed - min_download_speed)) %>%
  arrange(desc(norm_download_speed))

# View the broadband speed ranking
View(broadbandrank)
```

Figure 32:

Crime ranking code

```
library(tidyverse)
# Load the data
crime_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedcrime.csv")
housing_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedhousing.csv")

# Preprocessing for Crime Rate Ranking
town = housing_data %>%
  mutate(postcode = str_trim(substring(Postcode, 1, 6))) %>%
  mutate(TOWN = str_trim(toupper(`Town/City`))) %>%
  select(postcode, TOWN) %>%
  distinct()

sel_crime = crime_data %>%
  filter(Year == 2023) %>%
  group_by(postcode) %>%
  summarise(crimeno = n()) %>%
  arrange(desc(crimeno)) %>%
  select(postcode, crimeno)

final_crime = sel_crime %>%
  left_join(town, by = "postcode") %>%
  na.omit() %>%
  distinct()

last_crime = final_crime %>%
  group_by(TOWN) %>%
  summarise(crimerate = sum(crimeno)) %>%
  select(TOWN, crimerate)

# Calculate normalized crime rate
min_crimerate = min(last_crime$crimerate)
max_crimerate = max(last_crime$crimerate)

crimerank = last_crime %>%
  mutate(norm_crimerate = 1 - (crimerate - min_crimerate) / (max_crimerate - min_crimerate)) %>%
  arrange(desc(norm_crimerate))

# View the crime rate ranking
View(crimerank)
```

Figure 33:

School ranking code

```
# Load necessary libraries
library(tidyverse)

# Load the data
school_data = read_csv("X:/College projects/4th sem/Data Science/Cleaned/cleanedschool.csv")

# Preprocessing for School Ranking
att8 = school_data %>%
  group_by(TOWN) %>%
  summarise(av_att8 = mean(ATT8SCR)) %>%
  select(TOWN, av_att8) %>%
  distinct() %>%
  mutate(TOWN = str_trim(toupper(TOWN)))

# Calculate normalized Attainment 8 score
min_att8_score = min(att8$av_att8)
max_att8_score = max(att8$av_att8)

schoolrank = att8 %>%
  mutate(norm_att8_score = (av_att8 - min_att8_score) / (max_att8_score - min_att8_score)) %>%
  arrange(desc(norm_att8_score))

# View the school ranking
View(schoolrank)
```

Figure 34:

Final Ranking code

```
library(tidyverse)
# Final Overall Ranking
ranking_data = house_price %>%
  left_join(att8, by = "TOWN") %>%
  left_join(broadband_sel, by = "TOWN") %>%
  left_join(last_crime, by = "TOWN") %>%
  na.omit()

# Calculate normalized scores and final score
min_max <- ranking_data %>%
  summarise(
    min_download_speed = min(avg_down_speed),
    max_download_speed = max(avg_down_speed),
    min_upload_speed = min(avg_upl_speed),
    max_upload_speed = max(avg_upl_speed),
    min_att8_score = min(av_att8),
    max_att8_score = max(av_att8),
    min_price = min(avg_price),
    max_price = max(avg_price),
    min_crimerate = min(crimerate),
    max_crimerate = max(crimerate)
  )

ranking_data <- ranking_data %>%
  mutate(
    norm_download_speed = (avg_down_speed - min_max$min_download_speed) / (min_max$max_download_speed - min_max$min_download_speed),
    norm_upload_speed = (avg_upl_speed - min_max$min_upload_speed) / (min_max$max_upload_speed - min_max$min_upload_speed),
    norm_att8_score = (av_att8 - min_max$min_att8_score) / (min_max$max_att8_score - min_max$min_att8_score),
    norm_price = 1 - (avg_price - min_max$min_price) / (min_max$max_price - min_max$min_price),
    norm_crimerate = 1 - (crimerate - min_max$min_crimerate) / (min_max$max_crimerate - min_max$min_crimerate),
    final_score = norm_download_speed + norm_upload_speed + norm_att8_score + norm_price + norm_crimerate
  )

final_rank = ranking_data %>%
  select(TOWN, avg_price, crimerate, av_att8, avg_down_speed, final_score) %>%
  arrange(desc(final_score))

# View the final overall ranking
View(final_rank)
```