Real Estate Town Recommendation System

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

ST5014CEM Data Science for Developer

Hasina kc khatri “13703510”

August 19th, 2024

Table of Contents

Introduction

Buying property can be tricky because there are many things to think about, like how much houses cost, internet speed, safety, and how good the schools are. This is especially hard for people who don't know a place well.

This report is about creating a tool to help people find good places to buy property in Bristol and Cornwall, two different areas in the UK. Bristol is a busy city, and Cornwall has beautiful countryside and beaches. Both places have good and bad things for people who want to invest in property. We're going to use information about house prices, internet speed, crime, and schools to figure out which towns are the best places to buy property. We'll start by collecting and cleaning data, then look for patterns in the data. After that, we'll use math to understand how different things are connected. Finally, we'll create a simple tool that gives points to each town based on how good it is. The idea is to help people choose the best place to buy property in Bristol or Cornwall.

Collecting the Dataset

The Town Recommendation System aims to assist potential homebuyers and investors in Bristol and Cornwall by leveraging data science to identify optimal locations. By analysing factors like housing costs, broadband speed, education quality, and crime rates, the project develops a scoring system to rank towns and cities based on desirability. This report outlines the data cleaning, analysis, modelling, and recommendation system development processes, providing insights into the methodology and potential improvements. data.gov.uk.

Folder Structure and Organization

To maintain a structured approach, the collected datasets were organized into the following folders within the Obtained directory:

Cleaning Dataset

Data cleaning is a crucial process that ensures the quality and integrity of the data used in analysis. In this project, we focused on cleaning datasets related to house sales, broadband speeds, crime rates, and school performance, covering the years 2020 to 2023 for Bristol and Cornwall. Each dataset underwent a thorough cleaning process, which involved several key steps to prepare the data for analysis. For cleaning process, we have used library (dplyr, purrr, readr) tidy verse.

House Sale (Cleaning Dataset)

The house sales data, covering the years 2020 to 2023, was collected in four separate CSV files. Handling each file individually would have been time-consuming and inefficient. To start cleaning process, all these files were first combined into a single dataset. While cleaning this dataset created a “Partial Postcode” column by extracting the first five characters of Postcode.

Combining Datasets:

- Loaded the necessary libraries (dplyr, purrr, readr).
- Defined consistent column names across all files.
- Combined the datasets using bind_rows () function.
- Saved the combined file as combined_housing_data_2020_2023.csv.

Cleaning the Combined Dataset

- Loaded tidy verse and lubricate for data cleaning and date manipulation.
- Used na.omit() to remove any rows with missing values.
- Converted Price to numeric, formatted Date of Transfer to extract the year, added a serial number column (S_No).
- Filtered the data to include only "CITY OF BRISTOL" and "CORNWALL".

- Selected required columns: S_No, Price, Date of Transfer, Postcode, Town, District, County.
- Created a Partial Postcode column by extracting the first five characters of Postcode.
- Removed duplicate rows using distinct ().

```r
library(tidyverse)
library(lubridate)

# Load data
cleaned_data_housing <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Housepricingcombine.csv")

# Data cleaning and processing
cleaned_data <- cleaned_data_housing %>%
  as_tibble() %>%
  na.omit() %>%
  mutate(
    Price = as.numeric(Price),
    `Date of Transfer` = ymd(Date),
    `Date of Transfer` = year(`Date of Transfer`),
    S_No = row_number()
  ) %>%
  filter(County %in% c("CITY OF BRISTOL", "CORNWALL")) %>%
  select(S_No, Price, `Date of Transfer`, Postcode, Town, District, County) %>%  # Select required columns
  mutate(
    `Partial_Postcode` = substr(Postcode, 1, 5)
  ) %>%
  distinct()

housing_path ="/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/House_rate_cleaned.csv"

# Save the cleaned dataset
write.csv(cleaned_data, housing_path, row.names = FALSE)

# View cleaned data structure and summary
str(cleaned_data)
View(cleaned_data)
summary(cleaned_data)
```

Figure 1

Broadband Speed Data

Combining the raw datasets of broadband speed: The broadband speed data for this project was collected from two separate sources: coverage data and performance data. These datasets were initially stored in coverage.csv and performance.csv files. These two datasets were combined into a single dataset by merging them on the postcode column.

```
#Broadband speed
coverage_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/broadband speed/broadband/coverage.csv")

performance_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/broadband speed/broadband/performance.csv")

merged_data <- merge(coverage_data, performance_data, by = "postcode", all = TRUE)

summary(merged_data)

output_path_broadband <- "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/combined_broadband_speed.csv"
write_csv(merged_data, output_path_broadband)
```

Figure 2

Cleaning Process of Broadband Speed : Key columns from the broadband data were then selected and renamed for clarity, focusing on the maximum and average download speeds. A right join was performed with the Postcode to LSOA dataset. A new column named Partial Postcode was created by extracting the first five characters of the Postcode column. The dataset was then refined by selecting only the essential columns: average and maximum download speeds, postcode, partial postcode, town, district, and county.

```
library(tidyverse)

# Set working directory (adjust as needed)
setwd("/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data")

# Step 1: Load the broadband data and the cleaned Postcode to LSOA dataset
broadband_data <- read_csv("combined_broadband_speed_data.csv")
postcode_lsoa_cleaned <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Postcode_cleaned_LSOA.csv")

# Step 2: Select and rename important columns in broadband data
broadband_cleaned <- broadband_data %>%
  select(Postcode = postcode_space,
         Maximum_Download_Speed = `Maximum download speed (Mbit/s)`,
         Average_Download_Speed = `Average download speed (Mbit/s)`)

# Perform a right join with the cleaned Postcode_to_lsoa
broadband_lsoa_combined <- broadband_cleaned %>%
  right_join(postcode_lsoa_cleaned, by = "Postcode")

# Step 4: Add a Partial Postcode column
broadband_lsoa_combined <- broadband_lsoa_combined %>%
  mutate(Partial_Postcode = substr(Postcode, 1, 5)) # Assuming 'Partial_Postcode' uses the first 5 characters

# Step 5: Select only the required columns
final_broadband_data <- broadband_lsoa_combined %>%
  select(Average_Download_Speed,
         Maximum_Download_Speed,
         Postcode,
         Partial_Postcode,
         Town,
         District,
         County)

# Step 6: Eliminate any rows with missing values
final_broadband_data <- final_broadband_data %>%
  na.omit()
```

Figure 3

Crime Rates Dataset

The crime rates data was collected from monthly reports covering the regions of Avon and Somerset, and Devon and Cornwall, spanning from May 2021 to December 2023. The data was initially split across multiple CSV files, each representing a different month and region. The first step in the cleaning process was to combine these individual files into a single comprehensive dataset for easier analysis.

Combining the Datasets:

To aggregate all monthly crime rate files into one comprehensive dataset. The files were listed, read, and combined using the following code :

```
#crimerate
combine_files <-function(file_paths){
  data_list <- lapply(file_paths, read_csv)
  combined_data <- bind_rows(data_list)
  return(combined_data)}# Get all file paths for Avon and Somerset and Devon and Cornwall
file_paths_avon <- list.files(path ="/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Crime_rates",
                              pattern ="avon-and-somerset-street.csv$", full.names =TRUE, recursive =TRUE)

file_paths_devon <- list.files(path ="/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Crime_rates",
                               pattern ="devon-and-cornwall-street.csv$", full.names =TRUE, recursive =TRUE)# Combine all Avon and
combined_avon <- combine_files(file_paths_avon)# Combine all Devon and Cornwall files
combined_devon <- combine_files(file_paths_devon)# Combine both regions into one dataframe
combined_crime_data <- bind_rows(combined_avon, combined_devon)# Save the combined data to a CSV file in the Obtaine_data folder
output_path_crime <-"/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/combined_crime_data.csv"
write_csv(combined_crime_data, output_path_crime)# Confirmation message
print("Combined crime dataset has been saved successfully.")
```

Cleaning the Crime Rate Dataset :

- To combine multiple monthly crime data files into one comprehensive dataset, clean and organize the data by removing duplicates, filling in missing geographical details, and preparing it for analysis to accurately understand crime patterns in Bristol and Cornwall.
- **Join with Postcode to LSOA Data:** Perform a right join with the Postcode to LSOA dataset to enrich the crime data with geographical details.

```
combined_crime_data <- combined_crime_data %>%
  right_join(postcode_lsoa_dataset, by = "LSOA_Code", relationship = "many-to-many") %>%
  mutate(Falls_Within = ifelse(is.na(Outcome_Category), "Status update unavailable", Outcome_Category),
         Serial_Number = row_number())
```

- **Select and Finalize Columns:**

  Select only the required columns for analysis and ensure no duplicate rows exist.

```
combined_crime_data <- combined_crime_data %>%
  select(Crime_Date, Falls_Within, Crime_Type, LSOA_Code,
         Postcode, Partial_Postcode, Town, County) %>%
  distinct() %>%
  as_tibble()
```

- **Check for Missing Values:**

  The code is counting how many missing values there are in each column of the combined_crime_data dataset and then displaying those counts.

```
missing_values_summary <- sapply(combined_crime_data, function(x) sum(is.na(x)))
print(missing_values_summary)
```

- **Check for and Remove Duplicates**

  Identify and remove any duplicate rows to ensure the uniqueness of the **data.**

```
row_count_before <- nrow(combined_crime_data)
combined_crime_data <- combined_crime_data %>% distinct()
row_count_after <- nrow(combined_crime_data)
print(row_count_before - row_count_after)
```

The crime rates data was meticulously cleaned and prepared for analysis. This clean dataset allowed for an accurate and detailed examination of crime patterns in Bristol and Cornwall, contributing to the overall understanding of safety in these regions.

Here is the complete code of crime rate cleaning :

```
library(tidyverse)

crime_rates_directory <- "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Crime_rates"
postcode_lsoa_dataset_path <- "/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Postcode_cleaned_LSOA.csv"

postcode_lsoa_dataset <- read_csv(postcode_lsoa_dataset_path,
                                  col_types = cols(LSOA_Code = col_character(),
                                                   Postcode = col_character(),
                                                   Partial_Postcode = col_character(),
                                                   Town = col_character(),
                                                   County = col_character()))

combined_crime_data <- list.files(crime_rates_directory, pattern = "\\.csv$", full.names = TRUE, recursive = TRUE) %>%
  lapply(read_csv) %>%
  bind_rows() %>%
  select(`LSOA code`, `Crime type`, Month, `Last outcome category`) %>%
  rename(LSOA_Code = `LSOA code`,
         Crime_Type = `Crime type`,
         Crime_Date = Month,
         Outcome_Category = `Last outcome category`) %>%
  mutate(Crime_Date = as.Date(paste0(Crime_Date, "-01"), format = "%Y-%m-%d"),
         Crime_Year = substr(Crime_Date, 1, 4),
         Crime_Month = substr(Crime_Date, 6, 7)) %>%
  filter(!is.na(LSOA_Code) & !is.na(Crime_Type)) %>%
  right_join(postcode_lsoa_dataset, by = "LSOA_Code", relationship = "many-to-many") %>%
  mutate(Falls_Within = ifelse(is.na(Outcome_Category), "Status update unavailable", Outcome_Category),
         Serial_Number = row_number()) %>%
  select(Crime_Date, Falls_Within, Crime_Type, LSOA_Code,
         Postcode, Partial_Postcode, Town, County) %>%
  distinct() %>%
  as_tibble()

# Check for missing values
missing_values_summary <- sapply(combined_crime_data, function(x) sum(is.na(x)))
print(missing_values_summary)

# Check for duplicates
row_count_before <- nrow(combined_crime_data)
combined_crime_data <- combined_crime_data %>% distinct()
row_count_after <- nrow(combined_crime_data)
print(row_count_before - row_count_after)  # Number of duplicates removed

write_csv(combined_crime_data, "/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Cleaned_Crime_Data.csv")

# Check the first few rows of the cleaned crime data
head(combined_crime_data)
```

Here is the combination code of uncleaned crime rate :

```
#crimerate
combine_files <-function(file_paths){
  data_list <- lapply(file_paths, read_csv)
  combined_data <- bind_rows(data_list)
  return(combined_data)}# Get all file paths for Avon and Somerset and Devon and Cornwall
file_paths_avon <- list.files(path ="/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Crime_rates",
                              pattern ="avon-and-somerset-street.csv$", full.names =TRUE, recursive =TRUE)

file_paths_devon <- list.files(path ="/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Crime_rates",
                               pattern ="devon-and-cornwall-street.csv$", full.names =TRUE, recursive =TRUE)
combined_avon <- combine_files(file_paths_avon)# Combine all Devon and Cornwall files
combined_devon <- combine_files(file_paths_devon)# Combine both regions into one dataframe
combined_crime_data <- bind_rows(combined_avon, combined_devon)# Save the combined data to a CSV file
output_path_crime <-"/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/combined_crime_data.csv"
write_csv(combined_crime_data, output_path_crime)# Confirmation message
print("Combined crime dataset has been saved successfully.")
```

School Dataset

The school datasets for Bristol and Cornwall were collected for the academic years 2021-2022 and 2022-2023. At first, this data was stored in different files depending on the region and year. The data includes things like census information, where students go after school, test results, and general school details. To make sure everything was consistent and easy to analyse, the raw data was cleaned and combined. This involved merging the files, picking out the important columns, dealing with missing information, and making sure everything matched up before saving it all as one cleaned dataset.

Combining the Datasets:

Combined data from schools in Bristol and Cornwall for the school years 2021-2022 and 2022-2023. The data was originally split into separate files for each region and year. To make it easier to analyse and compare the information, created four new datasets: one for Bristol in 2021-2022, one for Bristol in 2022-2023, one for Cornwall in 2021-2022, and one for Cornwall in 2022-2023.

```
# Combine Bristol datasets for 2021-2022
Bristoldataset2021to2022 <-
  bristol_2021_2022 %>%
  lapply(read_csv, show_col_types = FALSE) %>%
  lapply(function(df) df %>% mutate(across(everything(), as.character))) %>%
  bind_rows()
```

- For each region and year, I combined the datasets using lap ply to read each CSV file and then converted all columns to character type to ensure consistency. Finally, I used bind_rows() to merge the data into a single data frame.

Cleaning the School Dataset :

This involves selecting relevant columns, handling missing data, and filtering out unneeded information. The cleaned datasets are then combined into a single file for easier analysis.

- **Loading the Data and Selecting Important Columns:**

  Load the data from the specified CSV file and select only the columns that are important for the analysis: School Name, Attainment Score, Town, and Postcode. The select function is used to filter the columns, while rename provides more readable column names.

```
# Clean the 2022-2023 Bristol School Dataset
Bristol_2022_2023_school_cleaning <- read_csv(bristol_2022_2023_path) %>%
  select(SCHNAME, ATT8SCR, TOWN, PCODE) %>%
  rename(`School Name` = SCHNAME, Town = TOWN, `Postcode` = PCODE, `Attainment Score` = ATT8SCR) %>%
  as_tibble() %>%
```

- **Handling Missing Values:**

  Create a new column Partial Postcode by extracting the first five characters of the Postcode, which helps in grouping and analysing the data by broader postcode areas. The na.omit() function is then applied to remove any rows with missing values, ensuring the dataset is clean and complete.

```
  mutate(Partial_Postcode = substr(Postcode, 1, 5)) %>%
  na.omit() %>%
```

- **Filtering Out Unnecessary Data:**

  Exclude rows where the Attainment Score is not available or suppressed. NE and SUPP are placeholders that indicate missing or non-reportable scores. Removing these rows ensures that the dataset only includes valid and

```
  filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>%
```

Here is cleaned code of School dataset :

```r
library(tidyverse)
library(lubridate)
# Define the file paths for each dataset
bristol_2021_2022_path <- "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Bristoldataset2021to2022.csv"
bristol_2022_2023_path <- "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Infobristoldataset2022to2023.csv"
cornwall_2021_2022_path <- "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Cornwalldataset2021to2022.csv"
cornwall_2022_2023_path <- "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/cornwalldataset2022to2023.csv"

# Clean the 2021-2022 Bristol School Dataset
Bristol_2021_2022_school_cleaning <- read_csv(bristol_2021_2022_path) %>%
  select(SCHNAME, ATT8SCR, TOWN, PCODE) %>%
  rename(`School Name` = SCHNAME, Town = TOWN, `Postcode` = PCODE, `Attainment Score` = ATT8SCR) %>%
  as_tibble() %>%
  mutate(Partial_Postcode = substr(Postcode, 1, 5)) %>%
  na.omit() %>%
  filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>%
  mutate(County = "CITY OF BRISTOL") %>%
  mutate(Year = "2022")
# Clean the 2022-2023 Bristol School Dataset
Bristol_2022_2023_school_cleaning <- read_csv(bristol_2022_2023_path) %>%
  select(SCHNAME, ATT8SCR, TOWN, PCODE) %>%
  rename(`School Name` = SCHNAME, Town = TOWN, `Postcode` = PCODE, `Attainment Score` = ATT8SCR) %>%
  as_tibble() %>%
  mutate(Partial_Postcode = substr(Postcode, 1, 5)) %>%
  na.omit() %>%
  filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>%
  mutate(County = "CITY OF BRISTOL") %>%
  mutate(Year = "2023")

# Clean the 2021-2022 Cornwall School Dataset
Cornwall_2021_2022_school_cleaning <- read_csv(cornwall_2021_2022_path) %>%
  select(SCHNAME, ATT8SCR, TOWN, PCODE) %>%
  rename(`School Name` = SCHNAME, Town = TOWN, `Postcode` = PCODE, `Attainment Score` = ATT8SCR) %>%
  as_tibble() %>%
  mutate(Partial_Postcode = substr(Postcode, 1, 5)) %>%
  na.omit() %>%
  filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>%
  mutate(County = "CORNWALL") %>%
  mutate(Year = "2022")
# Clean the 2022-2023 Cornwall School Dataset
Cornwall_2022_2023_school_cleaning <- read_csv(cornwall_2022_2023_path) %>%
  select(SCHNAME, ATT8SCR, TOWN, PCODE) %>%
  rename(`School Name` = SCHNAME, Town = TOWN, `Postcode` = PCODE, `Attainment Score` = ATT8SCR) %>%
  as_tibble() %>%
  mutate(Partial_Postcode = substr(Postcode, 1, 5)) %>%
  na.omit() %>%
  filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>%
  mutate(County = "CORNWALL") %>%
  mutate(Year = "2023")

# Combine all cleaned datasets into a single tibble
school_dataset_cleaned_combined <- bind_rows(
  Bristol_2021_2022_school_cleaning,
  Bristol_2022_2023_school_cleaning,
  Cornwall_2021_2022_school_cleaning,
  Cornwall_2022_2023_school_cleaning
)

# Save the combined cleaned dataset to CSV
cleanedSchool_path <- "/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/School_Dataset_Cleaned.csv"
write_csv(school_dataset_cleaned_combined, cleanedSchool_path)

# View the cleaned and combined dataset
View(school_dataset_cleaned_combined)
```

School Dataset Combined raw dataset for cleaning :

```r
#school dataset
library(dplyr)
library(readr)

# Define the directories for Bristol and Cornwall
bristol_2021_2022 <- list.files("/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/School dataset/Bristol School info/2021-2022", full.names = TRUE, pattern = "\\.csv$")
bristol_2022_2023 <- list.files("/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/School dataset/Bristol School info/2022-2023", full.names = TRUE, pattern = "\\.csv$")
cornwall_2021_2022 <- list.files("/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/School dataset/Cornwall School info/2021-2022", full.names = TRUE, pattern = "\\.csv$")
cornwall_2022_2023 <- list.files("/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/School dataset/Cornwall School info/2022-2023", full.names = TRUE, pattern = "\\.csv$")

# Combine Bristol datasets for 2021-2022
Bristoldataset2021to2022 <-
  bristol_2021_2022 %>%
  lapply(read_csv, show_col_types = FALSE) %>%
  lapply(function(df) df %>% mutate(across(everything(), as.character))) %>%
  bind_rows()

# Combine Bristol datasets for 2022-2023
Infobristoldataset2022to2023 <-
  bristol_2022_2023 %>%
  lapply(read_csv, show_col_types = FALSE) %>%
  lapply(function(df) df %>% mutate(across(everything(), as.character))) %>%
  bind_rows()

# Combine Cornwall datasets for 2021-2022
Cornwalldataset2021to2022 <-
  cornwall_2021_2022 %>%
  lapply(read_csv, show_col_types = FALSE) %>%
  lapply(function(df) df %>% mutate(across(everything(), as.character))) %>%
  bind_rows()

# Combine Cornwall datasets for 2022-2023
cornwalldataset2022to2023 <-
  cornwall_2022_2023 %>%
  lapply(read_csv, show_col_types = FALSE) %>%
  lapply(function(df) df %>% mutate(across(everything(), as.character))) %>%
  bind_rows()

# Save the combined datasets to CSV
write_csv(Bristoldataset2021to2022, "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Bristoldataset2021to2022.csv")
write_csv(Infobristoldataset2022to2023, "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Infobristoldataset2022to2023.csv")
write_csv(Cornwalldataset2021to2022, "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/Cornwalldataset2021to2022.csv")
write_csv(cornwalldataset2022to2023, "/Users/hasu/Desktop/TownRecommendationSystem /Obtaine_data/cornwalldataset2022to2023.csv")


```

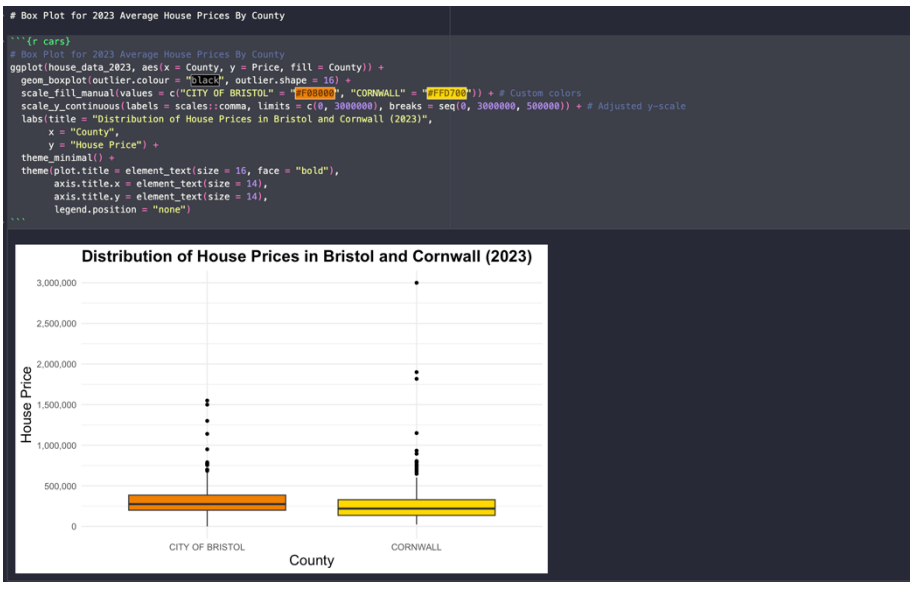
Exploratory Data Analysis

Before diving into detailed analysis or building models, EDA helps us see the main patterns, trends, and relationships in the data. It lets us spot any oddities or hidden insights by summarizing and visualizing the data. This helps us decide the best way to move forward with more complex analysis. Analysing broadband speeds allows us to evaluate internet quality and its impact on property attractiveness. Examining crime rates helps assess safety and desirability of different areas, while school performance metrics are crucial for families choosing where to live .EDA provides a comprehensive view of these factors, helping us spot trends, detect anomalies, and understand relationships, ensuring that our recommendations are based on a solid foundation of data insights.

EDA of House pricing

Average House price in 2023

1. Boxplot Analysis:
    - Objective**:** To visualize the distribution of average house prices in 2023 across different regions.
    - Visualizations: A boxplot showing the overall average house price for 2023. Separate boxplots for individual cities, allowing comparison of price distributions across different urban areas.

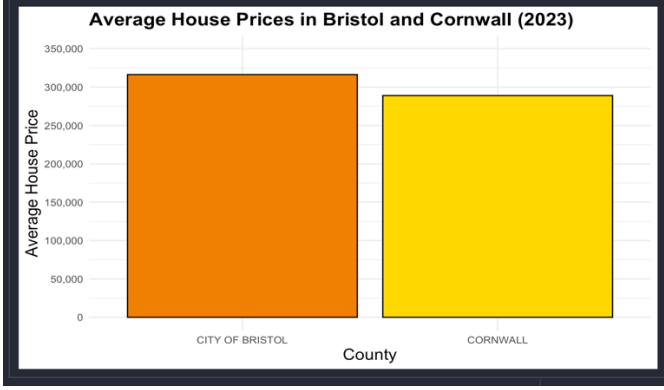


2. Bar Chart (Stacked):

    - Objective: To provide a clear comparison of average house prices in 2023 across different regions, with stacked bars showing how prices vary across cities within the same region.

```r
##  Bar Chart for 2023 Average House Prices by County

```{r pressure, echo=FALSE}
# Bar Chart for 2023 Average House Prices by County
ggplot(average_price_county, aes(x = County, y = Average_Price, fill = County)) +
  geom_bar(stat = "identity", color = "black") +
  scale_fill_manual(values = c("CITY OF BRISTOL" = "#F08000", "CORNWALL" = "#FFD700")) + # Custom colors
  scale_y_continuous(labels = scales::comma, limits = c(0, 350000), breaks = seq(0, 350000, 50000)) + # Adjusted y-scale
  labs(title = "Average House Prices in Bristol and Cornwall (2023)",
       x = "County",
       y = "Average House Price") +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 14),
        axis.title.y = element_text(size = 14),
        legend.position = "none")
```
```

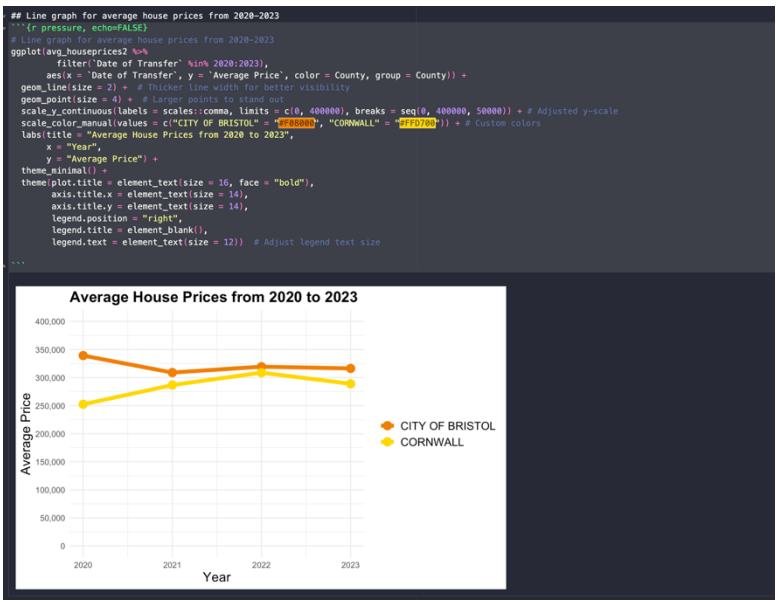


Average House Pricing from 2020 to 2023

3. Line Graph Analysis

    Objective**:** To examine the trends in average house prices over time, from 2020 to 2023. This line graph will allow us to see how prices have evolved in different cities, highlighting any significant increases or decreases that may inform future investment decisions.

```r
## Line graph for average house prices from 2020-2023
```{r pressure, echo=FALSE}
# Line graph for average house prices from 2020-2023
ggplot(avg_houseprices2 %>%
         filter(`Date of Transfer` %in% 2020:2023),
       aes(x = `Date of Transfer`, y = `Average Price`, color = County, group = County)) +
  geom_line(size = 2) +  # Thicker line width for better visibility
  geom_point(size = 4) +  # Larger points to stand out
  scale_y_continuous(labels = scales::comma, limits = c(0, 400000), breaks = seq(0, 400000, 50000)) + # Adjusted y-scale
  scale_color_manual(values = c("CITY OF BRISTOL" = "#F08000", "CORNWALL" = "#FFD700")) + # Custom colors
  labs(title = "Average House Prices from 2020 to 2023",
       x = "Year",
       y = "Average Price") +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 14),
        axis.title.y = element_text(size = 14),
        legend.position = "right",
        legend.title = element_blank(),
        legend.text = element_text(size = 12))  # Adjust legend text size
```
```



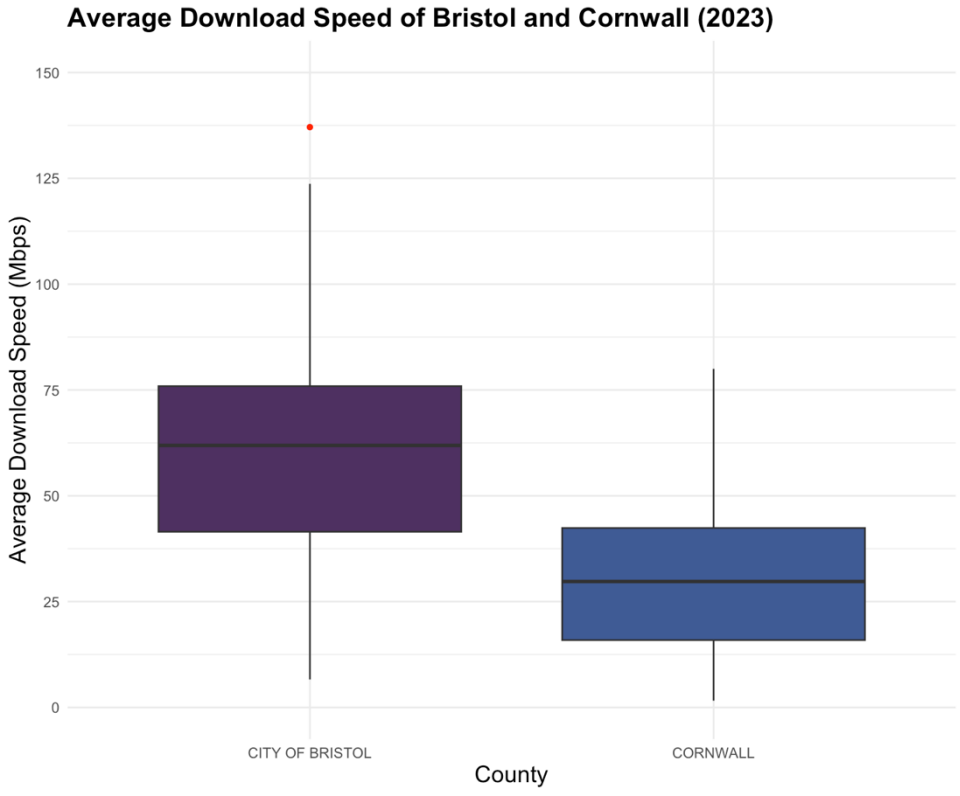
EDA of Broadband Speed

Average Download Speed

- Boxplot Analysis:

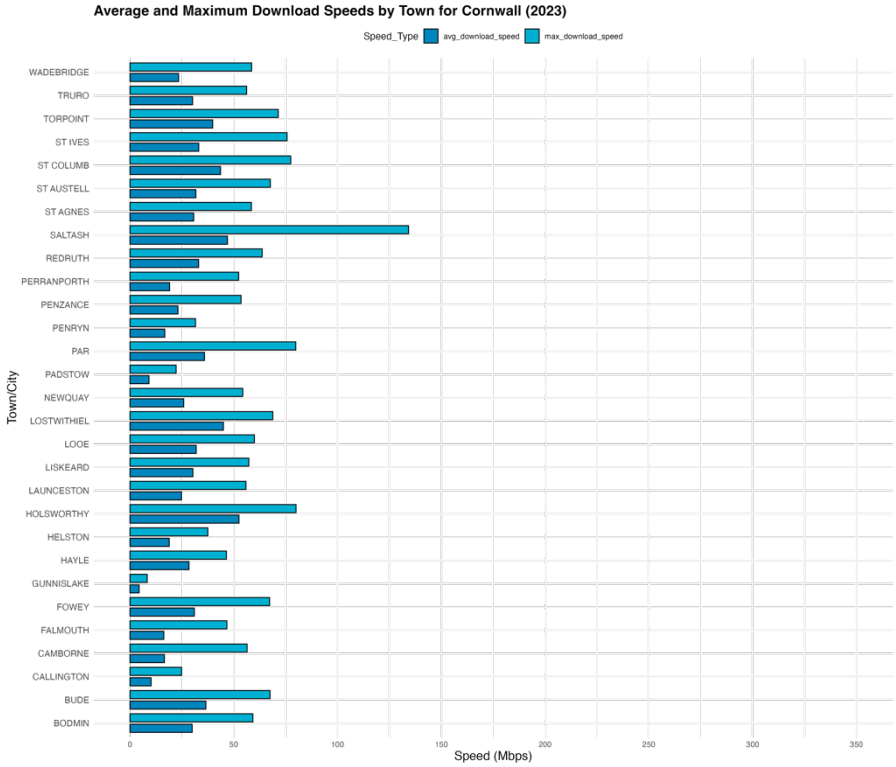
  To compare the average download speeds across different counties, particularly focusing on Bristol and Cornwall. This boxplot will help visualize the distribution and identify any significant outliers or areas with poor connectivity.

```r
# Load the broadband data
broadband_data <- read_csv('/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Broadband_speed_dataset_cleaned.csv')

# Filter data by county
bristol_data <- broadband_data %>% filter(County == 'CITY OF BRISTOL')
cornwall_data <- broadband_data %>% filter(County == 'CORNWALL')

# Define custom color palette
custom_palette <- c("#4d3061","#405a95","#0085bc","#00b0d2","#0fd9d5","#87ffcc")

# 1. Box Plot for Average Download Speed of Bristol and Cornwall Combined
box_plot <- ggplot(broadband_data, aes(x = County, y = Average_Download_Speed, fill = County)) +
  geom_boxplot(outlier.colour = "red", outlier.shape = 16) +
  scale_fill_manual(values = c("CITY OF BRISTOL" = custom_palette[1], "CORNWALL" = custom_palette[2])) +
  scale_y_continuous(labels = scales::comma, limits = c(0, 150), breaks = seq(0, 150, 25)) + # Adjusted y-scale for clarity
  labs(title = "Average Download Speed of Bristol and Cornwall (2023)",
       x = "County",
       y = "Average Download Speed (Mbps)") +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 14),
        axis.title.y = element_text(size = 14),
        legend.position = "none")

```



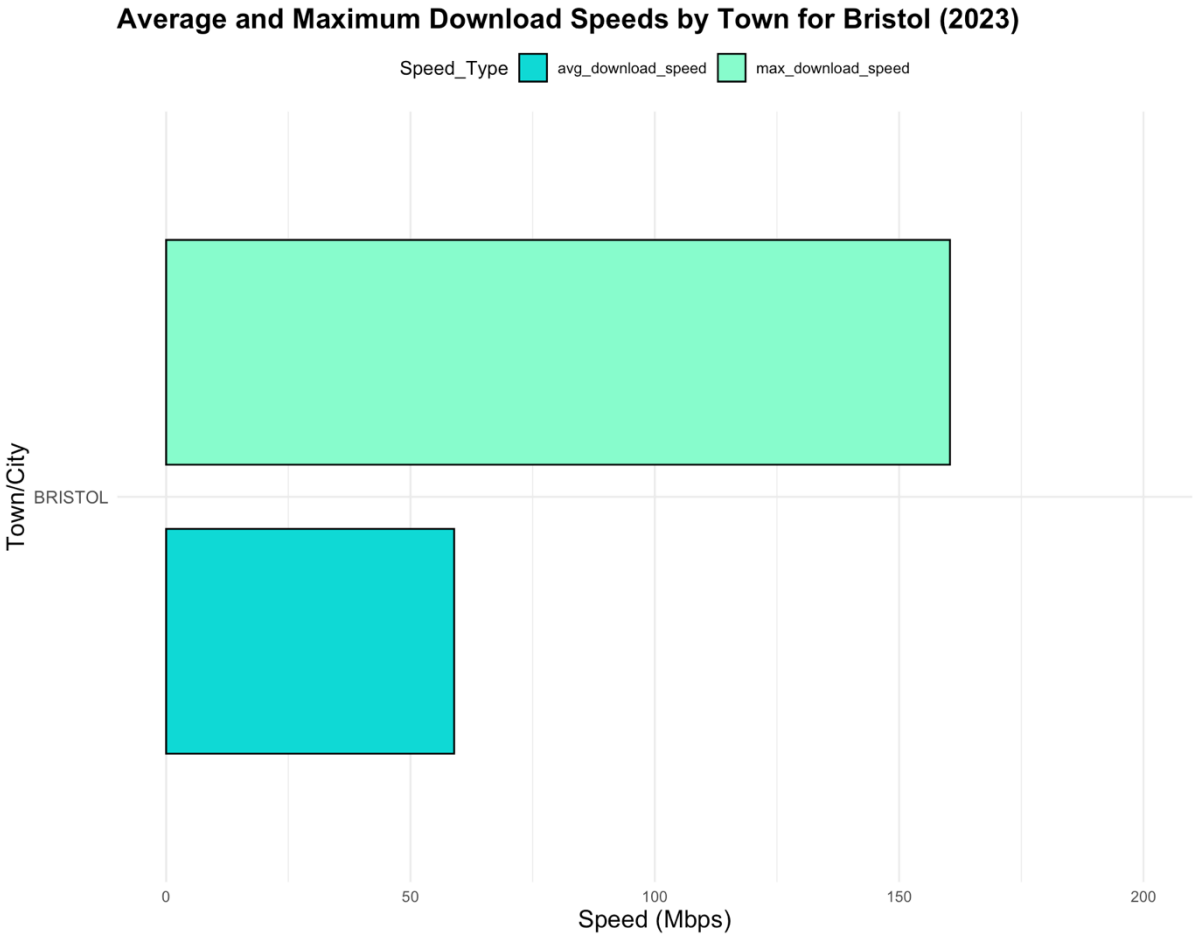
- Bar Charts for Cornwall and Bristol

  To separately visualize the average and maximum download speeds for Cornwall and Bristol. These bar charts will provide a clear comparison of internet speeds within these regions, which is crucial for evaluating the attractiveness of these areas for residents or businesses relying on high-speed internet.




```r
# 2. Bar Chart for Average and Maximum Download Speeds by Town for Cornwall
# Reshape data to long format
cornwall_long <- cornwall_data %>%
  group_by(Town) %>%
  summarise(avg_download_speed = mean(Average_Download_Speed, na.rm = TRUE),
            max_download_speed = mean(Maximum_Download_Speed, na.rm = TRUE)) %>%
  pivot_longer(cols = c(avg_download_speed, max_download_speed), names_to = "Speed_Type", values_to = "Speed")

# Bar Chart for Cornwall
cornwall_bar_chart <- ggplot(cornwall_long, aes(y = Town, x = Speed, fill = Speed_Type)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black", width = 0.7) +
  scale_fill_manual(name = "Speed_Type", values = custom_palette[3:4]) +
  scale_x_continuous(labels = scales::comma, limits = c(0, 350), breaks = seq(0, 350, 50)) + # Adjusted x-scale
  labs(title = "Average and Maximum Download Speeds by Town for Cornwall (2023)",
       x = "Speed (Mbps)",
       y = "Town/City") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10),
        plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 14),
        axis.title.y = element_text(size = 14),
        legend.position = "top")
```

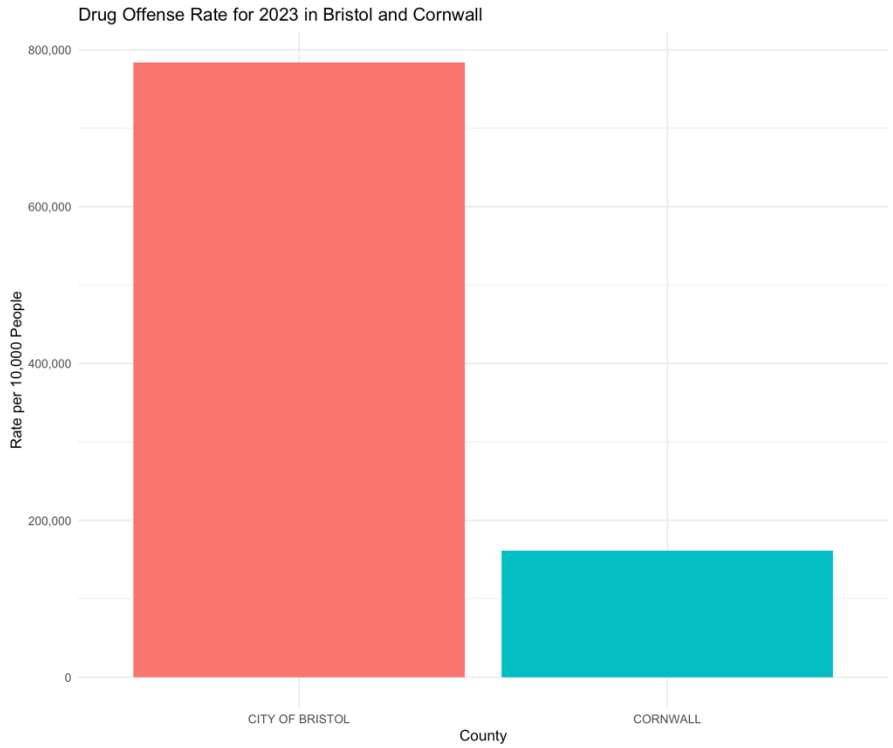
```r
# 3. Bar Chart for Average and Maximum Download Speeds by Town for Bristol
# Reshape data to long format
bristol_long <- bristol_data %>%
  group_by(Town) %>%
  summarise(avg_download_speed = mean(Average_Download_Speed, na.rm = TRUE),
            max_download_speed = mean(Maximum_Download_Speed, na.rm = TRUE)) %>%
  pivot_longer(cols = c(avg_download_speed, max_download_speed), names_to = "Speed_Type", values_to = "Speed")

# Bar Chart for Bristol
bristol_bar_chart <- ggplot(bristol_long, aes(y = Town, x = Speed, fill = Speed_Type)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black", width = 0.7) +
  scale_fill_manual(name = "Speed_Type", values = custom_palette[5:6]) +
  scale_x_continuous(labels = scales::comma, limits = c(0, 200), breaks = seq(0, 200, 50)) + # Adjusted x-scale
  labs(title = "Average and Maximum Download Speeds by Town for Bristol (2023)",
       x = "Speed (Mbps)",
       y = "Town/City") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10),
        plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 14),
        axis.title.y = element_text(size = 14),
        legend.position = "top")
```



EDA of Crime

Drug Offense Rate in 2023

- Bar plot Analysis: To analyse and compare the drug offense rates in 2023 between Bristol and Cornwall. This boxplot will help identify areas with higher drug-related crime rates, which could impact the desirability of these regions for potential residents or investors.

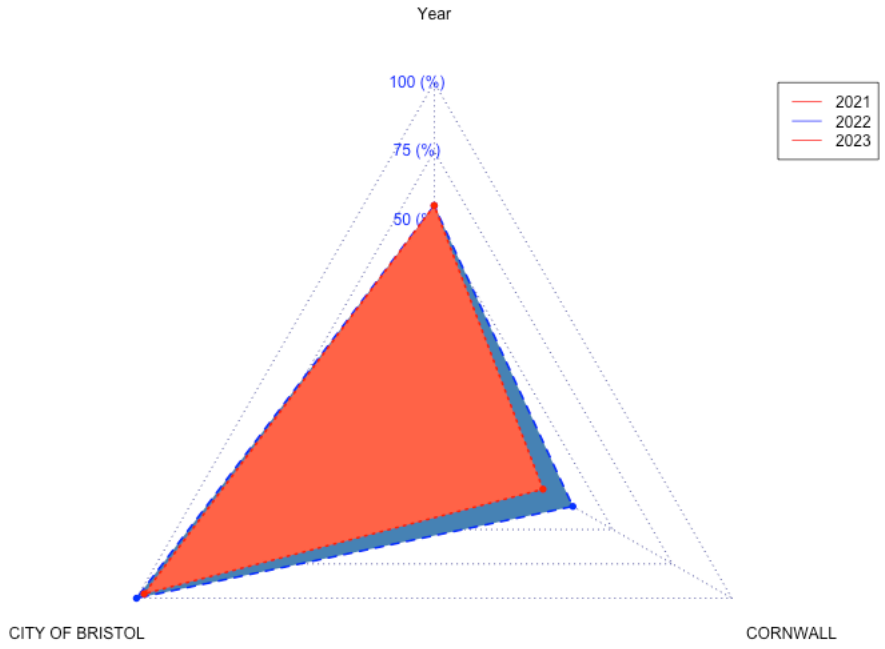



```r

# Filter for drug offenses in 2023 for Bristol and Cornwall
drug_offense_2023 <- crime_data_cleaned %>%
  filter(Year == 2023, Crime_Type == "Drugs", County %in% c("CITY OF BRISTOL", "CORNWALL"))

# Calculate the rate per 10,000 people for each county
drug_offense_rate <- drug_offense_2023 %>%
  group_by(County) %>%
  summarise(Total_Crimes = n()) %>%
  left_join(population_data_cleaned %>% select(County, Population2023), by = "County") %>%
  mutate(Rate_per_10K = (Total_Crimes / Population2023) * 10000)
print(drug_offense_rate)

# a. Bar Plot for Drug Offense Rate in 2023
ggplot(drug_offense_rate, aes(x = County, y = Rate_per_10K, fill = County)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Drug Offense Rate for 2023 in Bristol and Cornwall",
    x = "County",
    y = "Rate per 10,000 People"
  ) +
  theme_minimal()

```

Vehicle Crime Rate (2020-2023)

Radar Chart Analysis:

- To visualize the vehicle crime rate trends from 2020 to 2023. A radar chart will effectively show how different types of vehicle crimes have evolved over time, providing insights into the safety of various regions.

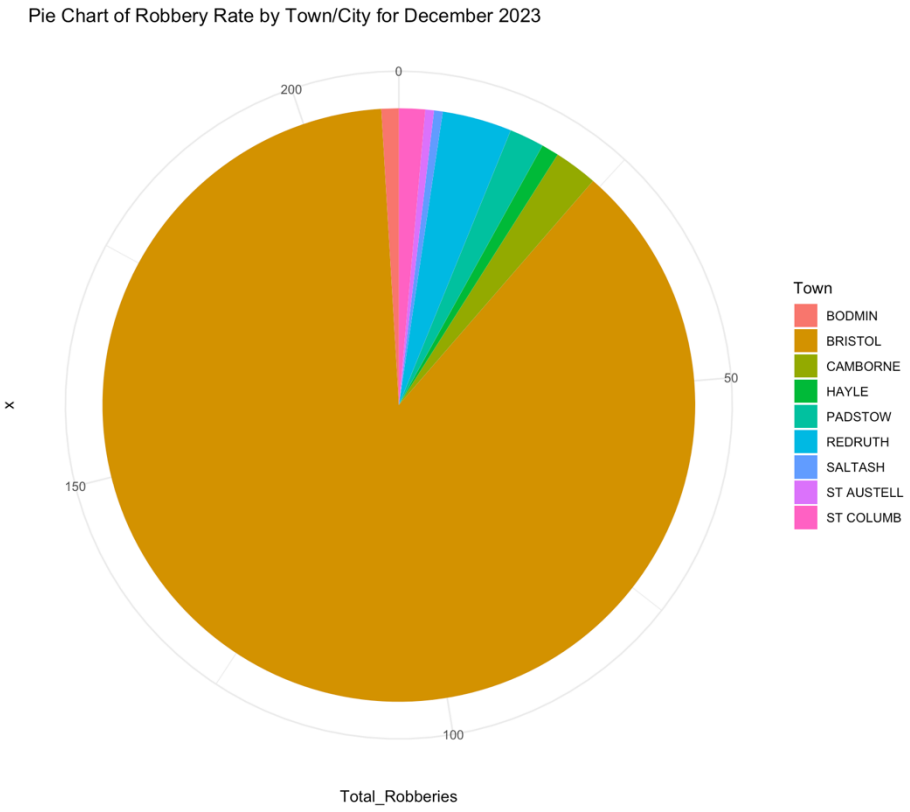



```r
# b. Radar Chart for an Alternative Crime Type (e.g., "Burglary")
alternative_crime_data <- crime_data_cleaned %>%
  filter(Year %in% 2020:2023, Crime_Type == "Burglary", County %in% c("CITY OF BRISTOL", "CORNWALL"))

# Aggregate data by year and county
alternative_crime_agg <- alternative_crime_data %>%
  group_by(Year, County) %>%
  summarise(Total_Crimes = n()) %>%
  spread(County, Total_Crimes, fill = 0)

# Prepare data for radar chart
radar_data <- alternative_crime_agg %>%
  select(-Year) %>%
  as.data.frame()
row.names(radar_data) <- alternative_crime_agg$Year

# Add columns for max and min
radar_data <- rbind(rep(max(radar_data, na.rm = TRUE), ncol(radar_data)),
                    rep(0, ncol(radar_data)),
                    radar_data)

# Save the radar chart to a PNG file
png(filename = "radar_chart.png", width = 800, height = 800)
radarchart(radar_data, axistype = 1,
           pcol = c("red", "blue"), pfcol = c("#FF6347", "#4682B4"), plwd = 2)
legend(x = 1, y = 1, legend = row.names(radar_data)[-c(1, 2)], col = c("red", "blue"), lty = 1)
dev.off()

```

Robbery Rate for a Specific Month in 2023

Pie Chart Analysis:

- To provide a snapshot of the robbery rate for a chosen month in 2023, using a pie chart to show the proportion of different robbery incidents across the region. This visualization will highlight the prevalence of robbery in specific areas during that period.

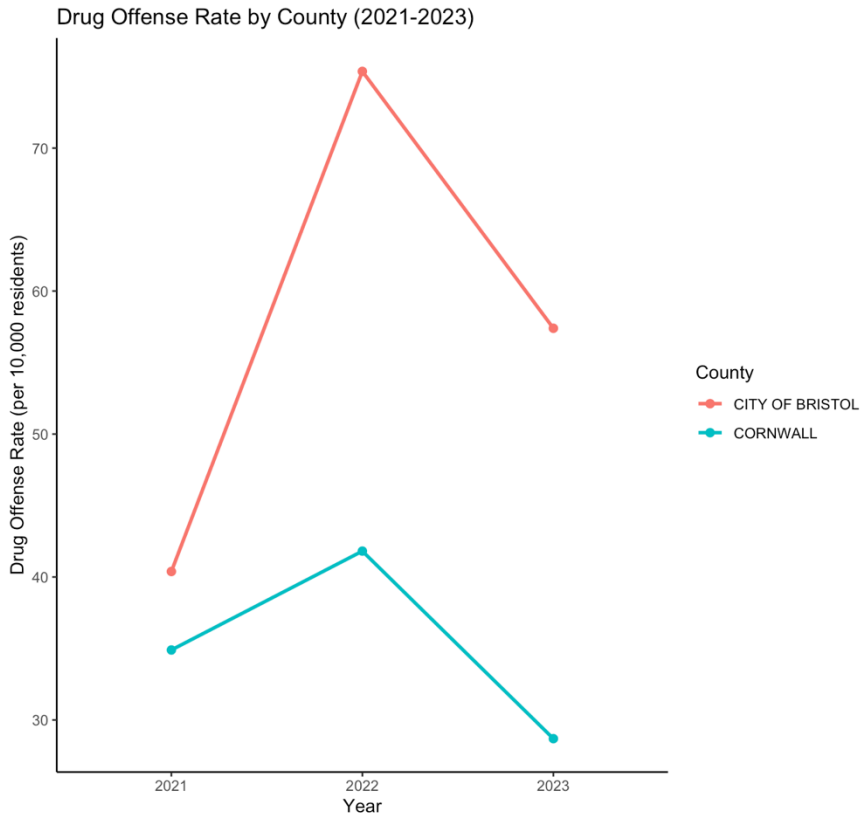



```r
# Ensure correct filtering for December 2023
robbery_data_dec_2023 <- crime_data_cleaned %>%
  filter(Year == 2023, Crime_Type == "Robbery", month(Crime_Date) == 12)

# Aggregate by Town
robbery_agg_dec_2023 <- robbery_data_dec_2023 %>%
  group_by(Town) %>%
  summarise(Total_Robberies = n())

# Generate pie chart
ggplot(robbery_agg_dec_2023, aes(x = "", y = Total_Robberies, fill = Town)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y") +
  labs(title = "Pie Chart of Robbery Rate by Town/City for December 2023") +
  theme_minimal()
```

Drug Offense Rate per 10,000 People

Line Chart Analysis

- To compare the drug offense rate per 10,000 people in both Bristol and Cornwall. By normalizing the data using population figures, this line chart will provide a fair comparison of drug crime rates between the two counties over time.



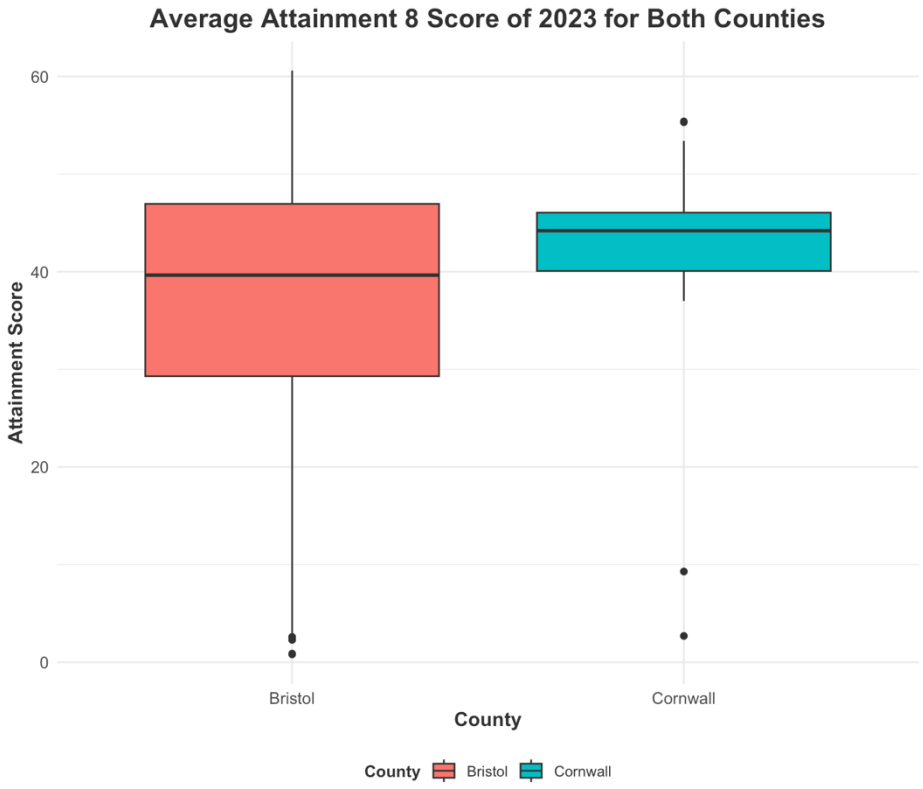

```r
# d. Line Chart for Drug Offense Rate per 10,000 People (2020-2023)
crime_summary <- crime_data_cleaned %>%
  mutate(`Crime Date` = year(Crime_Date)) %>%
  filter(`Crime Date` %in% c(2021, 2022, 2023)) %>%
  group_by(`Partial_Postcode`, `Crime_Type`, `County`, `Crime Date`) %>%
  tally() %>%
  rename(`Crime Count` = n) %>%
  filter(`Crime_Type` == "Drugs")  # Filter for drug-related crimes

# Join with population data and rename columns
joined_data <- crime_summary %>%
  left_join(population_data_cleaned, by = "Partial_Postcode", suffix = c("_crime", "_pop")) %>%
  rename(`County` = `County_crime`)

# Calculate the Drug Offense Rate
crime_dataset_with_rate <- joined_data %>%
  mutate(`Drug Offense Rate` = (`Crime Count` / Population2023) * 10000) %>%
  select(`County`, `Crime Date`, `Drug Offense Rate`, `Partial_Postcode`, `Crime_Type`) %>%
  group_by(`County`, `Crime Date`) %>%
  summarize(`Drug Offense Rate` = mean(`Drug Offense Rate`, na.rm = TRUE), .groups = 'drop')

# Create a line chart for drug offense rates by county from 2021 to 2023
ggplot(crime_dataset_with_rate, aes(x = `Crime Date`, y = `Drug Offense Rate`, color = `County`, group = `County`)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(title = "Drug Offense Rate by County (2021-2023)",
       x = "Year",
       y = "Drug Offense Rate (per 10,000 residents)") +
  scale_y_continuous(labels = scales::comma) +
  scale_x_discrete(limits = c(2021, 2022, 2023)) +
  theme_classic()  # Using a clean and simple theme
```

EDA of Schools

Average Attainment 8 score for 2023

- Boxplot Analysis:

  Compare the average Attainment 8 scores in 2023 between Bristol and Cornwall.

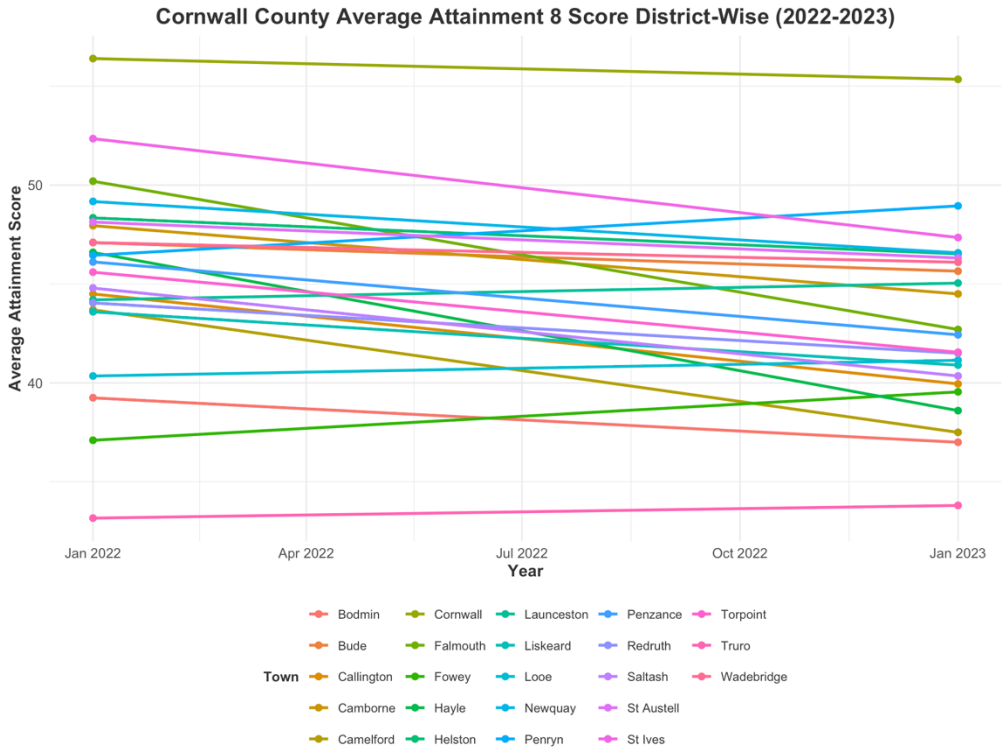



```r
# Filter data for the year 2023
school_data_2023 <- school_data_cleaned %>%
  filter(format(Year, "%Y") == "2023")

# Custom theme for the visualizations
custom_theme <- theme_minimal() +
  theme(
    text = element_text(family = "Arial", color = "gray20"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 10),
    legend.position = "bottom",
    legend.title = element_text(size = 10, face = "bold"),
    legend.text = element_text(size = 9)
  )

# Create a box plot for average attainment 8 score of 2023 for both counties
ggplot(school_data_2023, aes(x = County, y = Attainment_Score, fill = County)) +
  geom_boxplot() +
  ggtitle('Average Attainment 8 Score of 2023 for Both Counties') +
  ylab('Attainment Score') +
  xlab('County') +
  custom_theme
```

District-wise Attainment 8 score Analysis

- Line Graph for Cornwall:

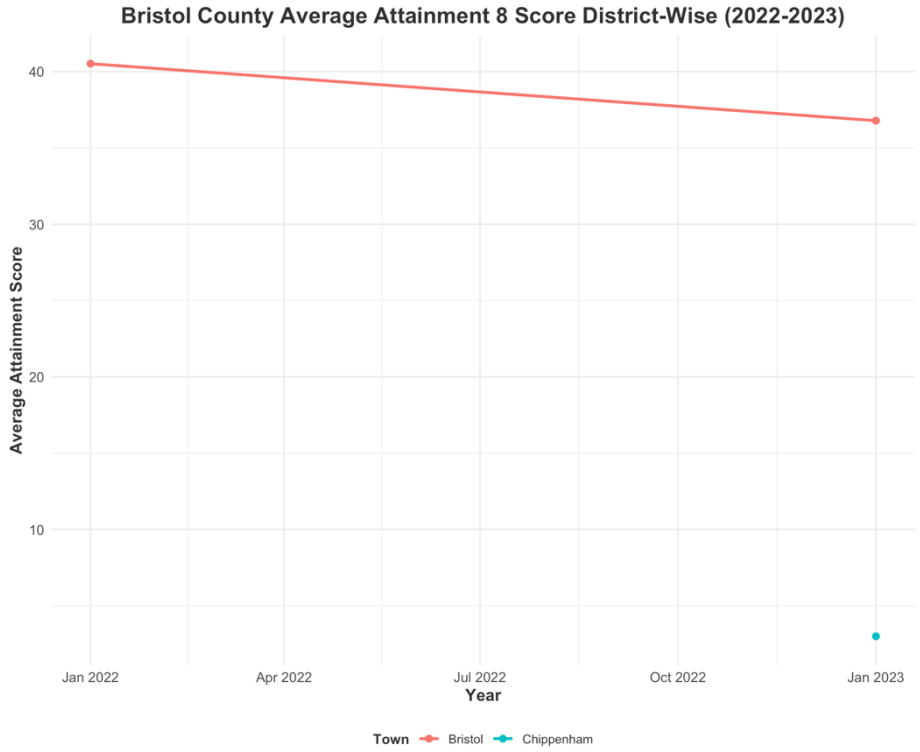
To track the average Attainment 8 scores across different districts in Cornwall from 2020 to 2023. This line graph will highlight any trends or changes in educational performance at the district level.




```r
# Line graph for Cornwall County's average attainment 8 score district-wise

# Filter data for Cornwall
cornwall_data <- school_data_cleaned %>%
  filter(County == 'Cornwall', format(Year, "%Y") %in% c("2022", "2023"))

# Group by Town and Year and calculate the average attainment score
cornwall_grouped <- cornwall_data %>%
  group_by(Town, Year) %>%
  summarise(Average_Attainment_Score = mean(Attainment_Score, na.rm = TRUE), .groups = 'drop')

# Create a line graph for Cornwall
ggplot(cornwall_grouped, aes(x = Year, y = Average_Attainment_Score, color = Town, group = Town)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  ggtitle('Cornwall County Average Attainment 8 Score District-Wise (2022-2023)') +
  ylab('Average Attainment Score') +
  xlab('Year') +
  custom_theme
```

- Line Graph for Bristol:

  Like Cornwall, this line graph will focus on the average Attainment 8 scores in Bristol, allowing us to compare educational outcomes across different districts within the city.



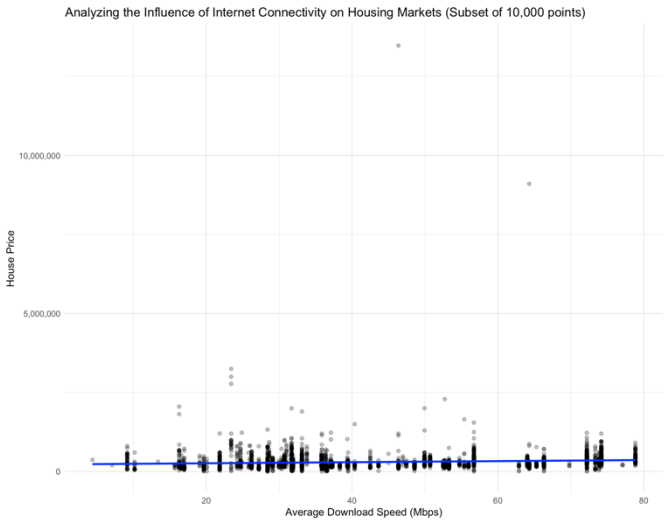

```r
# Line graph for Bristol County's average attainment 8 score district-wise

# Filter data for Bristol
bristol_data <- school_data_cleaned %>%
  filter(County == 'Bristol', format(Year, "%Y") %in% c("2022", "2023"))

# Group by Town and Year and calculate the average attainment score
bristol_grouped <- bristol_data %>%
  group_by(Town, Year) %>%
  summarise(Average_Attainment_Score = mean(Attainment_Score, na.rm = TRUE), .groups = 'drop')

# Create a line graph for Bristol
ggplot(bristol_grouped, aes(x = Year, y = Average_Attainment_Score, color = Town, group = Town)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  ggtitle('Bristol County Average Attainment 8 Score District-Wise (2022-2023)') +
  ylab('Average Attainment Score') +
  xlab('Year') +
  custom_theme
```

Linear Modelling

Explore relationships between variables and make predictions based on the data. In this project, the focus is on understanding how different factors influence each other, particularly in the context of housing, crime rates, broadband speeds, and education outcomes. A linear model will be constructed with house price as the dependent variable and download speed as the independent variable. The model will use house price as the dependent variable and drug rate as the independent variable. To assess the relationship between educational outcomes (as measured by the Attainment 8 score) and house prices, combining data from both Bristol and Cornwall. These are the modelling we have done from the Cleaned datasets

House Price vs. Download Speed

- **Objective:** To determine if there is a significant relationship between the average house price and the download speed in different regions.

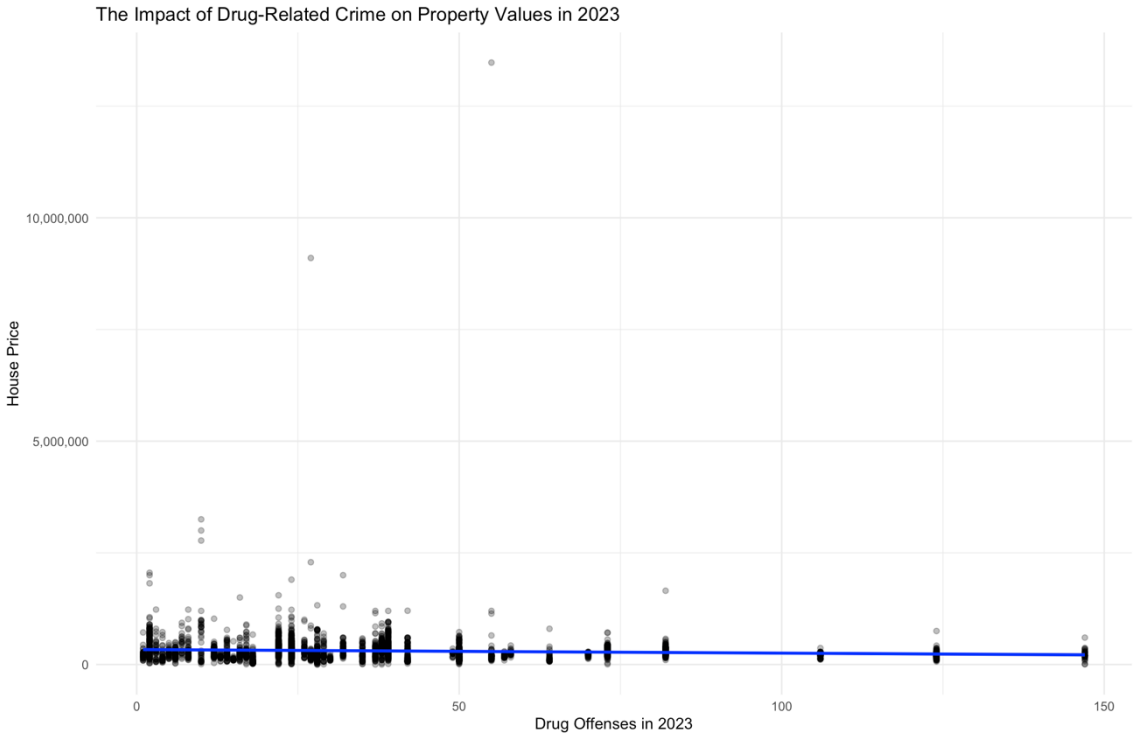



```r
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(readr)
library(scales)  # This library helps with formatting

# Load the datasets
house_price_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/House_rate_cleaned.csv")
broadband_speed_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Broadband_speed_dataset_cleaned.csv")

# Aggregate broadband data to get the mean download speed per Partial_Postcode
aggregated_broadband_data <- broadband_speed_data %>%
  group_by(Partial_Postcode) %>%
  summarize(Average_Download_Speed = mean(Average_Download_Speed, na.rm = TRUE),
            Maximum_Download_Speed = mean(Maximum_Download_Speed, na.rm = TRUE))

merged_data <- inner_join(house_price_data, aggregated_broadband_data, by = "Partial_Postcode")

# Drop any rows with missing values in the relevant columns
merged_data_cleaned <- merged_data %>%
  filter(!is.na(Price) & !is.na(Average_Download_Speed))

# Take a random subset of available points (without replacement)
set.seed(42)
subset_data <- merged_data_cleaned %>% sample_n(size = nrow(merged_data_cleaned))

# Create the scatter plot with a regression line and formatted y-axis
ggplot(subset_data, aes(x = Average_Download_Speed, y = Price)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Analyzing the Influence of Internet Connectivity on Housing Markets (Subset of 10,000 points)",
       x = "Average Download Speed (Mbps)",
       y = "House Price") +
  scale_y_continuous(labels = scales::comma) +  # Format y-axis with commas
  theme_minimal()
```

This code analyses the relationship between internet connectivity and housing prices by merging two datasets: one for house prices and another for broadband speeds. It aggregates the broadband data by postcode, calculating the mean download speeds, and then merges this with the house price data. After cleaning the merged data to remove any missing values, a random subset is taken for analysis. Finally, it creates a scatter plot with a regression line to visualize the influence of average download speed on house prices, helping to explore the potential impact of internet connectivity on the housing market.

House Price vs. Drug Rate (2023)

- **Objective:** To explore how drug offense rates in 2023 affect house prices in various regions. The model will use house price as the dependent variable and drug rate as the independent variable. The results will show if areas with higher drug crime rates tend to have lower property values.




```r
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(readr)
library(scales)

# Load the datasets
house_price_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/House_rate_cleaned.csv")
crime_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Cleaned_Crime_Data.csv")

# Filter crime data for drug-related offenses in 2023
drug_data_2023 <- crime_data %>%
  filter(grepl("^2023", Crime_Date) & grepl("Drugs", Crime_Type, ignore.case = TRUE))

# Aggregate drug offenses by Partial_Postcode
drug_rate_2023 <- drug_data_2023 %>%
  group_by(Partial_Postcode) %>%
  summarize(Drug_Offenses_2023 = n())

# Merge the aggregated drug rate data with the house price data using the Partial_Postcode column
merged_data <- inner_join(house_price_data, drug_rate_2023, by = "Partial_Postcode")

# Drop any rows with missing values in the relevant columns
merged_data_cleaned <- merged_data %>%
  filter(!is.na(Price) & !is.na(Drug_Offenses_2023))

# Create the scatter plot with a regression line
ggplot(merged_data_cleaned, aes(x = Drug_Offenses_2023, y = Price)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "The Impact of Drug-Related Crime on Property Values in 2023",
       x = "Drug Offenses in 2023",
       y = "House Price") +
  scale_y_continuous(labels = scales::comma) +  # Format y-axis with commas
  theme_minimal()

# Perform linear regression
model <- lm(Price ~ Drug_Offenses_2023, data = merged_data_cleaned)

# Summarize the model
summary(model)
```

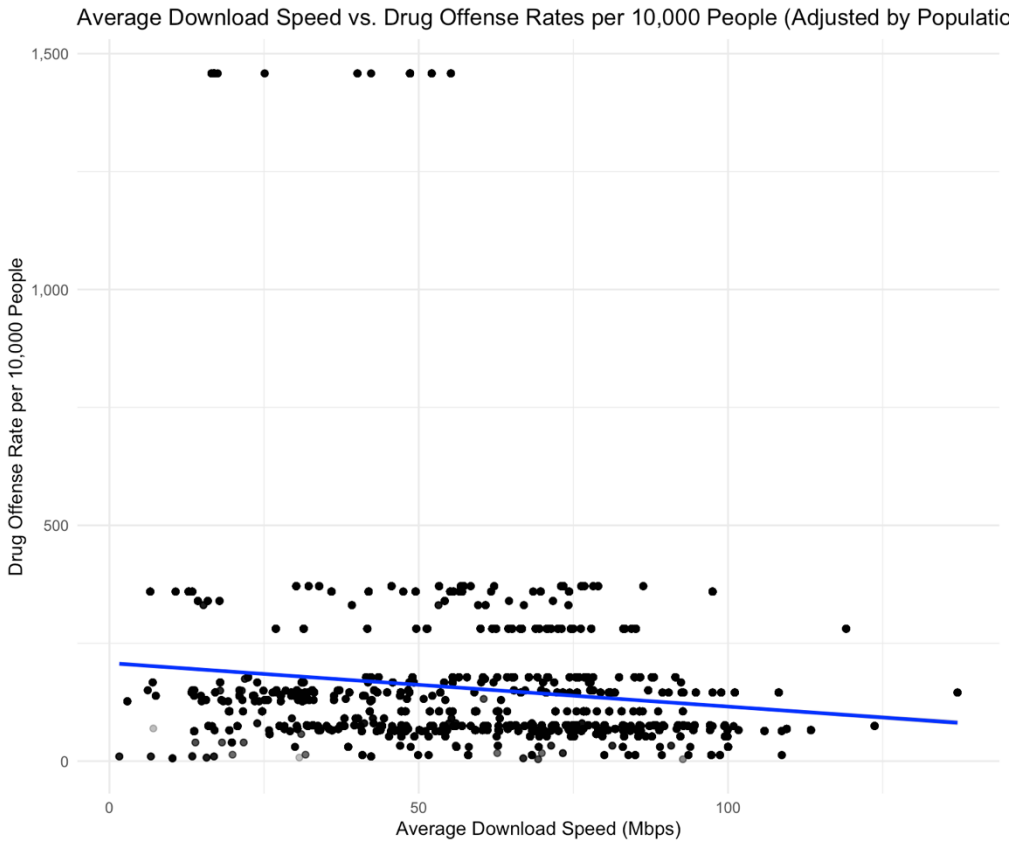
Attainment 8 score vs. House Price (Combined Counties)

Objective: To assess the relationship between educational outcomes (as measured by the Attainment 8 score) and house prices, combining data from both Bristol and Cornwall.


```r
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(readr)
library(scales)

# Load the datasets
school_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/School_Dataset_Cleaned.csv")
house_price_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/House_rate_cleaned.csv")

# Merge the school dataset with the house price dataset using the Partial_Postcode column
merged_data <- inner_join(school_data, house_price_data, by = "Partial_Postcode")

# Drop any rows with missing values in the relevant columns
merged_data_cleaned <- merged_data %>%
  filter(!is.na(`Attainment Score`) & !is.na(Price))

# Create the scatter plot with a regression line
ggplot(merged_data_cleaned, aes(x = `Attainment Score`, y = Price)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Exploring the Relationship Between School Performance and Property Values",
       x = "Attainment 8 Score",
       y = "House Price") +
  scale_y_continuous(labels = scales::comma) +  # Format y-axis with commas
  theme_minimal()

```

Average Download Speed vs. Drug Offense Rates per 10,000 People

To investigate whether internet connectivity (download speed) is related to crime rates, specifically drug offenses per 10,000 people.




```r
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(readr)
library(scales)

# Load the datasets
broadband_speed_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Broadband_speed_dataset_cleaned.csv")
crime_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Cleaned_Crime_Data.csv")
population_data <- read_csv("/Users/hasu/Desktop/TownRecommendationSystem /cleaned_datasets/Cleaned_Population_Data.csv")

# Step 1: Filter crime data for drug-related offenses
drug_offense_data <- crime_data %>%
  filter(grepl("Drugs", Crime_Type, ignore.case = TRUE))

# Aggregate drug offenses by Partial_Postcode
drug_offense_counts <- drug_offense_data %>%
  group_by(Partial_Postcode) %>%
  summarize(Drug_Offenses_Count = n())

# Aggregate broadband data by Partial_Postcode to avoid duplication
aggregated_broadband_data <- broadband_speed_data %>%
  group_by(Partial_Postcode) %>%
  summarize(Average_Download_Speed = mean(Average_Download_Speed))

# Step 2: Merge population data with drug offense counts to calculate drug offense rates
merged_crime_population <- inner_join(drug_offense_counts, population_data, by = "Partial_Postcode")

# Calculate the drug offense rate per 10,000 people using the actual population
merged_crime_population <- merged_crime_population %>%
  mutate(Drug_Offense_Rate_per_10000 = (Drug_Offenses_Count / Population2023) * 10000)

# Step 3: Merge with Broadband Speed Data
merged_data_final <- inner_join(broadband_speed_data,
                                merged_crime_population %>% select(Partial_Postcode, Drug_Offense_Rate_per_10000),
                                by = "Partial_Postcode")

# Step 4: Generate the updated visualization
ggplot(merged_data_final, aes(x = Average_Download_Speed, y = Drug_Offense_Rate_per_10000)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Average Download Speed vs. Drug Offense Rates per 10,000 People (Adjusted by Population)",
       x = "Average Download Speed (Mbps)",
       y = "Drug Offense Rate per 10,000 People") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma)  # Format y-axis with commas

```

Recommendation System

The recommendation system was created to help people make smart choices when investing in property in Cornwall and Bristol. It looks at important things like house prices, internet speed, school quality, and crime rates to give a clear picture of the best towns or cities to invest in or live in. The system is also flexible, so it can be adjusted to fit what each person cares about most.

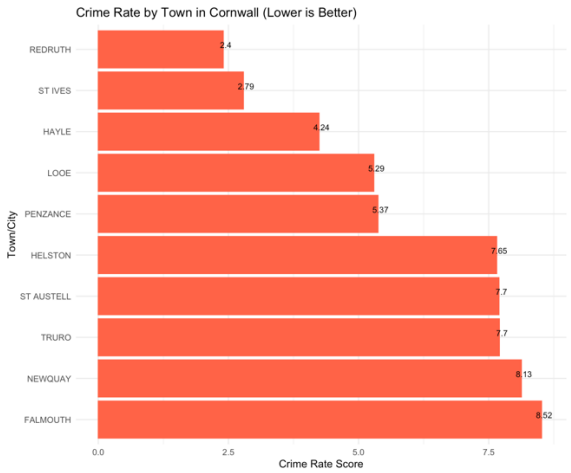
Results(Pricing, Broadband, Schools, Crime)

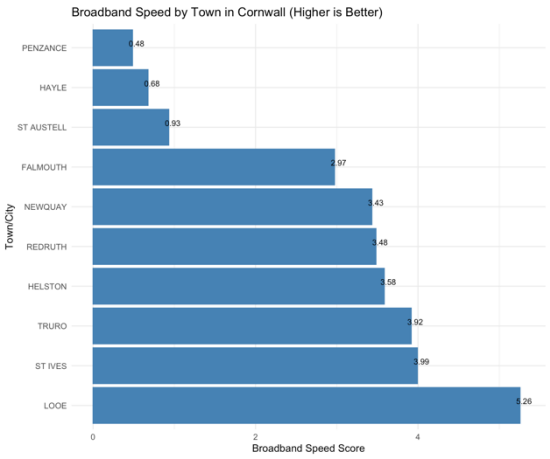
Cornwall:

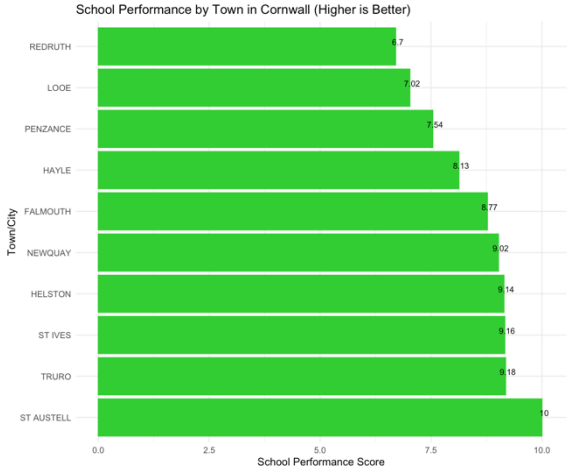
- **Top Towns**: Truro, Helston, and Newquay stand out as the best towns in Cornwall, with Truro leading the list with a balanced overall score.
- **Pricing**: House prices in Cornwall tend to be more affordable, with St. Austell showing the highest house score.
- **Broadband**: Broadband speed in Cornwall is relatively moderate, with Looe performing the best in this category.
- **Schools**: School performance is particularly strong in St. Austell, which scored the highest in this category.
- **Crime**: Crime rates are lowest in Truro and Helston, making them safer choices for potential residents.

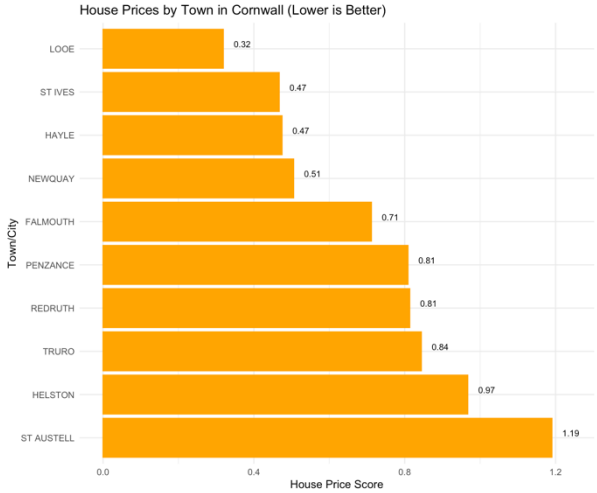
A tibble: 10 × 7

	Partial_Postcode	Town	HouseScore	BroadbandScore	SchoolScore	CrimeScore	Overall_Score
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	TR15	REDRUTH	0.813	3.48	6.70	2.40	3.35
2	TR2 5	TRURO	0.844	3.92	9.18	7.70	5.41
3	TR26	ST IVES	0.467	3.99	9.16	2.79	4.10
4	TR12	HELSTON	0.967	3.58	9.14	7.65	5.34
5	TR19	PENZANCE	0.809	0.485	7.54	5.37	3.55
6	TR7 3	NEWQUAY	0.506	3.43	9.02	8.13	5.27
7	TR11	FALMOUTH	0.712	2.97	8.77	8.52	5.24
8	TR27	HAYLE	0.475	0.676	8.13	4.24	3.38
9	PL13	LOOE	0.319	5.26	7.02	5.29	4.47
10	PL25	ST AUSTELL	1.19	0.931	10	7.70	4.95


Crime Rate by Town in Cornwall (Lower is Better)
REDRUTH
ST IVES
HAYLE
LOOE
PENZANCE
HELSTON
ST AUSTELL
TRURO
NEWQUAY
FALMOUTH
2.4
2.79
4.24
5.29
5.37
7.65
7.7
7.7
8.13
8.52
Town/City
0.0
2.5
5.0
7.5
Crime Rate Score


Broadband Speed by Town in Cornwall (Higher is Better)
PENZANCE
HAYLE
ST AUSTELL
FALMOUTH
NEWQUAY
REDRUTH
HELSTON
TRURO
ST IVES
LOOE
0.48
0.68
0.93
2.97
3.43
3.48
3.58
3.92
3.99
5.26
Town/City
0
2
4
Broadband Speed Score


School Performance by Town in Cornwall (Higher is Better)
REDRUTH
LOOE
PENZANCE
HAYLE
FALMOUTH
NEWQUAY
HELSTON
ST IVES
TRURO
ST AUSTELL
6.7
7.02
7.54
8.13
8.77
9.02
9.14
9.16
9.18
10
Town/City
0.0
2.5
5.0
7.5
10.0
School Performance Score


House Prices by Town in Cornwall (Lower is Better)
LOOE
ST IVES
HAYLE
NEWQUAY
FALMOUTH
PENZANCE
REDRUTH
TRURO
HELSTON
ST AUSTELL
0.32
0.47
0.47
0.51
0.71
0.81
0.81
0.84
0.97
1.19
Town/City
0.0
0.4
0.8
1.2
House Price Score

Bristol:

- **Top Town**: Bristol city was analysed, and it performed moderately across all factors.
- **Pricing**: House prices in Bristol are higher compared to Cornwall, but the affordability is relatively low.
- **Broadband**: Broadband speed in Bristol is better than in most towns in Cornwall, making it an attractive option for those requiring high-speed internet.
- **Schools**: School performance in Bristol is commendable, particularly in the BS7 9 area, which scored the highest.
- **Crime**: Crime rates in Bristol are moderate, with some areas performing better than others.

```
> print(final_scores_bristol)
# A tibble: 10 × 7
   Partial_Postcode Town    HouseScore BroadbandScore SchoolScore CrimeScore Overall_Score
   <chr>            <chr>        <dbl>          <dbl>       <dbl>      <dbl>         <dbl>
 1 BS7 9            BRISTOL    0.0243           0.575       0.858      0.649         0.527
 2 BS16             BRISTOL    0.0185           0.383       0.839      0.224         0.366
 3 BS6 6            BRISTOL    0.0281           0.502       0.818      0.359         0.427
 4 BS8 1            BRISTOL    0.0253           0.375       0.253      0.428         0.270
 5 BS13             BRISTOL    0.00551          0.257       0.635      0.826         0.431
 6 BS6 5            BRISTOL    0.0290           0.606       1          0.195         0.457
 7 BS10             BRISTOL    0.0148           0.868       0.678      0.814         0.594
 8 BS3 4            BRISTOL    0.0153           0.299       0          0.269         0.146
 9 BS5 9            BRISTOL    0.0160           0.379       0.607      0.812         0.453
10 BS9 3            BRISTOL    0.0306           0.716       0.668      0.838         0.563
>
```

Reflection

In this analysis, we aimed to identify the best town in Cornwall based on key factors such as house prices, school performance, broadband speed, and crime rate. Each of these factors plays a significant role in determining the overall quality of life and suitability of a town for living.

Key Factors :

House Prices: Looe, St Ives, Hayle, and Newquay stood out as the most affordable towns, making them attractive for individuals looking for lower living costs.

School Performance: Towns like St Austell, Truro, St Ives, and Helston showed strong educational outcomes, which is crucial for families prioritizing good schools.

Broadband Speed: Looe, St Ives, Truro, and Helston excelled in internet connectivity, a key factor in today's digital age.

Crime Rate: Redruth, St Ives, and Hayle were among the safest towns, offering peace of mind to residents.

Among all the towns analysed, **St Ives** consistently ranked high across all categories. It combines affordability, excellent school performance, strong broadband speed, and a low crime rate, making it a well-rounded choice for both living and working. **Looe** and **Helston** also performed well, but St Ives stands out as the best overall option due to its balanced strengths across all important factors.

Overall score

St Ives is the top town according to our recommendation system, earning the highest overall score. This town stands out because it offers a great balance of important factors. The schools in St Ives are particularly strong, making it an excellent choice for families who value education. The housing market here is also attractive, with good property values that suggest it's a smart place to invest. Additionally, St Ives has fast broadband speeds, which is crucial for staying connected in today's digital world. The town also has a low crime rate, providing a safe environment for residents. Overall, St Ives's combination of excellent schools, solid property values, reliable internet, and safety makes it the best town for living or investing in Cornwall.

Ranking_Table_Cornwall_Towns

Town	House Price Score (Lower is Better)	School Performance Score (Higher is Better)	Broadband Speed Score (Higher is Better)	Crime Rate Score (Lower is Better)
St Ives	0.47	9.16	3.99	2.79
Looe	0.32	7.02	5.26	5.29
Helston	0.97	9.14	3.58	7.65
Truro	0.84	9.18	3.92	7.7
Hayle	0.47	8.13	0.68	4.24
Falmouth	0.71	8.77	2.97	8.52
Penzance	0.81	7.54	0.48	5.37
Redruth	0.81	6.7	3.48	2.4
St Austell	1.19	10.0	0.93	7.7
Newquay	0.51	9.02	3.43	7.13

Legal and ethical

When creating a recommendation system like this, it's important to think about legal and ethical issues. Legally, we need to make sure that the data we use follows data protection laws, like the GDPR. This means protecting people's privacy and making sure any personal data is made anonymous and handled responsibly.

The system should be fair and not biased. It's important to avoid discrimination by making sure the criteria used don’t Favor or harm any group. Transparency is also crucial; users should know how the recommendations are made and be able to question them if needed. Additionally, we should consider how the recommendations might affect the communities involved, making sure the system helps people and doesn’t cause problems like pushing people out of their neighbourhoods.

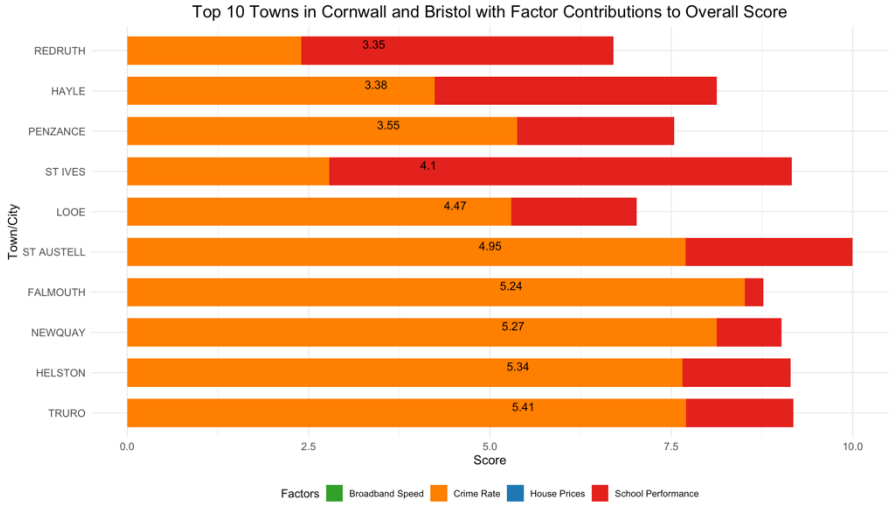
.

Conclusion

This project successfully developed a comprehensive recommendation system to assist individuals in making informed decisions about property investments in Cornwall and Bristol. By carefully analysing key factors such as house prices, internet speed, school performance, and crime rates, the system offers a clear and balanced view of the best towns or cities for living or investing. Throughout the process, we prioritized both legal and ethical considerations to ensure the system's fairness, transparency, and compliance with data protection laws. The project also emphasized flexibility, allowing the system to adapt to the specific preferences of users.

Overall, this project not only provides a valuable tool for property investors but also demonstrates the power of data-driven decision-making. By considering multiple aspects of community life, the system helps promote positive outcomes for both individuals and the wider community. Future improvements could further enhance the system’s accuracy and usability, ensuring it remains a trusted resource for anyone looking to invest in property.

Appendix




```r
# Combine the final scores from Cornwall and Bristol into one dataframe
combined_final_scores <- bind_rows(final_unique_towns, final_scores_bristol)
 View(combined_final_scores)

# Select the overall top 10 towns based on the Overall_Score
overall_top_10_towns <- combined_final_scores %>%
  arrange(desc(Overall_Score)) %>%
  slice(1:10)

# Print the final top 10 towns
print(overall_top_10_towns)

# Generate the bar chart for the top 10 towns across both counties
ggplot(overall_top_10_towns, aes(x = reorder(Town, -Overall_Score), y = Overall_Score)) +
  geom_bar(aes(y = HouseScore, fill = "House Prices"), stat = "identity", width = 0.7) +
  geom_bar(aes(y = BroadbandScore, fill = "Broadband Speed"), stat = "identity", width = 0.7, position = "stack") +
  geom_bar(aes(y = SchoolScore, fill = "School Performance"), stat = "identity", width = 0.7, position = "stack") +
  geom_bar(aes(y = CrimeScore, fill = "Crime Rate"), stat = "identity", width = 0.7, position = "stack") +
  geom_text(aes(label = round(Overall_Score, 2), y = Overall_Score + 0.05),
            position = position_dodge(width = 0.9), vjust = -0.25) +
  scale_fill_manual(values = c("House Prices" = "#1f78b4",  # Blue
                               "Broadband Speed" = "#33a02c", # Green
                               "School Performance" = "#e31a1c", # Red
                               "Crime Rate" = "#ff7f00", # Orange
                               "Overall Score" = "#6a3d9a")) + # Purple
  theme_minimal() +
  labs(title = "Top 10 Towns in Cornwall and Bristol with Factor Contributions to Overall Score",
       x = "Town/City",
       y = "Score",
       fill = "Factors") +
  coord_flip() +
  theme(legend.position = "bottom", # Move legend to the bottom
        plot.title = element_text(hjust = 0.5, size = 16), # Center and size the title
        axis.title.x = element_text(size = 12), # Size the X axis title
        axis.title.y = element_text(size = 12), # Size the Y axis title
        axis.text = element_text(size = 10)) # Size the axis text
```

```r
# Filter the data to include only Bristol
house_price_data_bristol <- house_price_data %>% filter(County == "CITY OF BRISTOL")
broadband_speed_data_bristol <- broadband_speed_data %>% filter(County == "CITY OF BRISTOL")
school_data_bristol <- school_data %>% filter(County == "CITY OF BRISTOL")
crime_data_bristol <- crime_data %>% filter(County == "CITY OF BRISTOL")

# Normalize each factor using Min-Max Scaling for Bristol
house_price_data_bristol$HouseScore <- (house_price_data_bristol$Price - min(house_price_data_bristol$Price, na.rm = TRUE)) /
  (max(house_price_data_bristol$Price, na.rm = TRUE) - min(house_price_data_bristol$Price, na.rm = TRUE))

broadband_speed_data_bristol$BroadbandScore <- (broadband_speed_data_bristol$Average_Download_Speed - min(broadband_speed_data_bristol$Average_Download_Speed, na.rm = TRUE)) /
  (max(broadband_speed_data_bristol$Average_Download_Speed, na.rm = TRUE) - min(broadband_speed_data_bristol$Average_Download_Speed, na.rm = TRUE))

school_data_bristol$SchoolScore <- (school_data_bristol$`Attainment Score` - min(school_data_bristol$`Attainment Score`, na.rm = TRUE)) /
  (max(school_data_bristol$`Attainment Score`, na.rm = TRUE) - min(school_data_bristol$`Attainment Score`, na.rm = TRUE))

# Aggregate crime data by Partial_Postcode and calculate CrimeScore for Bristol
crime_summary_bristol <- crime_data_bristol %>%
  group_by(Partial_Postcode) %>%
  summarize(Crime_Count = n(), .groups = 'drop')

crime_summary_bristol$CrimeScore <- (max(crime_summary_bristol$Crime_Count, na.rm = TRUE) - crime_summary_bristol$Crime_Count) /
  (max(crime_summary_bristol$Crime_Count, na.rm = TRUE) - min(crime_summary_bristol$Crime_Count, na.rm = TRUE))

# Weighted Scoring: Assign weights to each factor (adjust as needed based on client preferences)
house_weight <- 0.25
broadband_weight <- 0.25
school_weight <- 0.25
crime_weight <- 0.25

# Combine all scores into one dataframe using the Partial_Postcode as the key
combined_scores_bristol <- house_price_data_bristol %>%
  select(Partial_Postcode, Town, HouseScore) %>%
  left_join(broadband_speed_data_bristol %>% select(Partial_Postcode, BroadbandScore), by = "Partial_Postcode") %>%
  left_join(school_data_bristol %>% select(Partial_Postcode, SchoolScore), by = "Partial_Postcode") %>%
  left_join(crime_summary_bristol %>% select(Partial_Postcode, CrimeScore), by = "Partial_Postcode") %>%
  mutate(Overall_Score = house_weight * HouseScore + broadband_weight * BroadbandScore +
           school_weight * SchoolScore + crime_weight * CrimeScore)

# Filter to remove any rows with missing values in key factors
combined_scores_bristol <- combined_scores_bristol %>%
  filter(!is.na(HouseScore) & !is.na(BroadbandScore) & !is.na(SchoolScore) & !is.na(CrimeScore))

# Select the top 10 unique partial postcodes in Bristol based on the Overall Score
final_scores_bristol <- combined_scores_bristol %>%
  distinct(Partial_Postcode, .keep_all = TRUE) %>%
  slice(1:10)
```

```r
# Color palette
crime_color <- "#FF6347"  # Red
broadband_color <- "#4682B4"  # Blue
school_color <- "#32CD32"  # Green
house_price_color <- "#FFA500"  # Orange

# Create a bar chart for Crime Rate (Lower is Better)
ggplot(final_unique_towns, aes(x = reorder(Town, -CrimeScore), y = CrimeScore, fill = crime_color)) +
  geom_bar(stat = "identity", color = crime_color) +
  theme_minimal() +
  labs(title = "Crime Rate by Town in Cornwall (Lower is Better)",
       x = "Town/City",
       y = "Crime Rate Score",
       fill = "Crime Rate") +
  scale_fill_identity() +
  coord_flip() +
  geom_text(aes(label = round(CrimeScore, 2), y = CrimeScore + 0.05),
            color = "black", size = 3, vjust = -0.25)

# Create a bar chart for Broadband Speed (Higher is Better)
ggplot(final_unique_towns, aes(x = reorder(Town, -BroadbandScore), y = BroadbandScore, fill = broadband_color)) +
  geom_bar(stat = "identity", color = broadband_color) +
  theme_minimal() +
  labs(title = "Broadband Speed by Town in Cornwall (Higher is Better)",
       x = "Town/City",
       y = "Broadband Speed Score",
       fill = "Broadband Speed") +
  scale_fill_identity() +
  coord_flip() +
  geom_text(aes(label = round(BroadbandScore, 2), y = BroadbandScore + 0.05),
            color = "black", size = 3, vjust = -0.25)

# Create a bar chart for School Performance (Higher is Better)
ggplot(final_unique_towns, aes(x = reorder(Town, -SchoolScore), y = SchoolScore, fill = school_color)) +
  geom_bar(stat = "identity", color = school_color) +
  theme_minimal() +
  labs(title = "School Performance by Town in Cornwall (Higher is Better)",
       x = "Town/City",
       y = "School Performance Score",
       fill = "School Performance") +
  scale_fill_identity() +
  coord_flip() +
  geom_text(aes(label = round(SchoolScore, 2), y = SchoolScore + 0.05),
            color = "black", size = 3, vjust = -0.25)
```

Recommendation system filtering of Cornwall

```r
# Filter the data to include only Cornwall
house_price_data <- house_price_data %>% filter(County == "CORNWALL")
broadband_speed_data <- broadband_speed_data %>% filter(County == "CORNWALL")
school_data <- school_data %>% filter(County == "CORNWALL")
crime_data <- crime_data %>% filter(County == "CORNWALL")

# Normalize each factor and scale from 0 to 10
house_price_data$HouseScore <- 10 * (house_price_data$Price - min(house_price_data$Price, na.rm = TRUE)) /
  (max(house_price_data$Price, na.rm = TRUE) - min(house_price_data$Price, na.rm = TRUE))

broadband_speed_data$BroadbandScore <- 10 * (broadband_speed_data$Average_Download_Speed - min(broadband_speed_data$Average_Download_Speed, na.rm = TRUE)) /
  (max(broadband_speed_data$Average_Download_Speed, na.rm = TRUE) - min(broadband_speed_data$Average_Download_Speed, na.rm = TRUE))

school_data$SchoolScore <- 10 * (school_data$`Attainment Score` - min(school_data$`Attainment Score`, na.rm = TRUE)) /
  (max(school_data$`Attainment Score`, na.rm = TRUE) - min(school_data$`Attainment Score`, na.rm = TRUE))

# Aggregate crime data by Partial_Postcode and calculate CrimeScore (inverted since lower crime is better)
crime_summary <- crime_data %>%
  group_by(Partial_Postcode) %>%
  summarize(Crime_Count = n(), .groups = 'drop')

crime_summary$CrimeScore <- 10 * (max(crime_summary$Crime_Count, na.rm = TRUE) - crime_summary$Crime_Count) /
  (max(crime_summary$Crime_Count, na.rm = TRUE) - min(crime_summary$Crime_Count, na.rm = TRUE))

# Combine all scores into one dataframe using the Partial_Postcode as the key
combined_scores <- house_price_data %>%
  select(Partial_Postcode, Town, HouseScore) %>%
  left_join(broadband_speed_data %>% select(Partial_Postcode, BroadbandScore), by = "Partial_Postcode") %>%
  left_join(school_data %>% select(Partial_Postcode, SchoolScore), by = "Partial_Postcode") %>%
  left_join(crime_summary %>% select(Partial_Postcode, CrimeScore), by = "Partial_Postcode") %>%
  mutate(Overall_Score = rowMeans(select(., HouseScore, BroadbandScore, SchoolScore, CrimeScore), na.rm = TRUE))

# Filter to remove any rows with missing values in key factors
combined_scores <- combined_scores %>%
  filter(!is.na(HouseScore) & !is.na(BroadbandScore) & !is.na(SchoolScore) & !is.na(CrimeScore))

# Remove duplicates by selecting distinct rows based on Town
final_unique_towns <- combined_scores %>%
  distinct(Town, .keep_all = TRUE) %>%
  slice(1:10)

# Print the final list of unique top 10 towns
print(final_unique_towns)
```

Recommendation system of Cornwall

References

Data science cleaning - Effective data cleaning is a vital part of the data analytics process.

But what is it, why is it important, and how do you do it?

https://schoolworkspro.com/

House Pricing Dataset (Download From 2020 to 2023)-

https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads

Broadband Speed -https://www.ofcom.org.uk/phones-and-broadband/coverage-and-speeds/data-downloads

Crime dataset -https://data.police.uk/data/

EDA analysis -https://www.youtube.com/watch?v=j1pkPsjYw5s

https://www.youtube.com/watch?v=3iz-2iM4RFE