

Town Recommendation System and Report on an Individual Data Science Project

Bipasha Lamsal

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

Data Science for Developers, ST5014CEM

Siddhartha Neupane

August 19, 2024

Table of Contents

| | |
|---------------------------------------|----|
| Introduction | 6 |
| Cleaning Data | 7 |
| House Price | 7 |
| Towns and Postcodes | 9 |
| Broadband Speed | 10 |
| Crime | 12 |
| School | 13 |
| Exploratory Analysis | 15 |
| House Prices | 15 |
| Broadband Speeds | 18 |
| Crime Rate | 21 |
| Schools | 26 |
| Linear Modeling | 30 |
| Recommendation System | 41 |
| Overview | 41 |
| Results | 42 |
| Reflection | 50 |
| Overall Score | 51 |
| Legal and Ethical Issues | 53 |

| | |
|-------------------------|----|
| Conclusion | 54 |
| Appendix..... | 56 |

Table of Figures

| | |
|--|----|
| Figure 1: House Prices Cleaning | 8 |
| Figure 2: Postcode to lsoa Cleaning | 9 |
| Figure 3: Broadband Speed Cleaning..... | 11 |
| Figure 4: Crime Cleaning..... | 12 |
| Figure 5: School Cleaning | 13 |
| Figure 6: Population Cleaning..... | 14 |
| Figure 7: House Price Visualization..... | 15 |
| Figure 8: BroadBand Speed Visualization | 18 |
| Figure 9: Crime Visualization..... | 21 |
| Figure 10: School Visualization | 26 |
| Figure 11: House Prices vs Download Speed..... | 30 |
| Figure 12: House Prices vs Drug Offence Rates | 33 |
| Figure 13: Attainment 8 Score vs House Price..... | 35 |
| Figure 14: Average download speed vs Attainment 8 Score..... | 37 |
| Figure 15: Average download speed vs drug offence rates (per ten thousand people) | 39 |
| Figure 16: Top Ten Towns | 47 |
| Figure 17: Top Ten Towns Output..... | 48 |
| Figure 18: Top Ten Towns Excel..... | 49 |
| Figure 19: Overall Score | 52 |
| Figure 20: House Prices Ranking | 56 |
| Figure 21: School Grades Ranking | 57 |
| Figure 22: Crime Ranking | 58 |

Figure 23: Broadband Speed Ranking..... 59

Introduction

This project focuses on analyzing data to find the best towns or cities for living or investing. We looked at important factors like house prices, internet speed, school grades, and crime rates. By putting all these factors together, we aimed to identify the top places that offer a good balance of affordability, quality of life, and safety.

We used different datasets, each containing information about these factors in various towns and cities. The data was cleaned, combined, and then analyzed to create an overall score for each location. This score helps in ranking the towns and cities based on how well they meet the criteria we set.

The purpose of this document is to explain the steps we took, the results we found, and the recommendations we can make based on this analysis. We also discuss any challenges we faced and how this analysis could be improved in the future.

Cleaning Data

House Price

The first step involved reading and combining datasets from 2020 to 2023 into a single data frame with consistent column names. The Price column was then converted to a numeric format, and any rows with missing values were eliminated in order to clean up the data. In order to concentrate on particular areas, the dataset was filtered so that it only contained records from "CITY OF BRISTOL" and "CORNWALL." A new column called Short Postcode was created by taking the first five characters out of the Postcode and simplifying the Date of Transfer to just the year. This preparation guarantees that the dataset is appropriate for the analysis and manageable. (Government, n.d.)

After the dataset was cleansed, just the important columns—S_No, Price, Date of Transfer, Postcode, Town/City, District, and County—were left. This approach makes the dataset more usable and less difficult. After processing, the data was put into a CSV file for further analysis. (GeeksforGeeks, n.d.)

Figure 1

House Prices Cleaning

```

1 #Cleaning the data for house prices
2 library(dplyr)
3 library(tidyverse)
4 library(lubridate)
5
6 # Define column names for the datasets
7 column_names = c("Transaction_ID", "Price", "Date of Transfer", "Postcode", "Property_Type", "Old/New", "Duration", "PAON",
8 "SAON", "Street", "Locality", "Town/City", "District", "County", "PPD_Category_type", "Record_Status")
9
10 # Read and combine data for the years 2020 to 2023
11 pp_2020 = read_csv("D:/Data_Science_Assignment/Obtained_Data/House Price Dataset/pp-2020.csv", col_names = FALSE) %>%
12 setNames(column_names)
13 pp_2021 = read_csv("D:/Data_Science_Assignment/Obtained_Data/House Price Dataset/pp-2021.csv", col_names = FALSE) %>%
14 setNames(column_names)
15 pp_2022 = read_csv("D:/Data_Science_Assignment/Obtained_Data/House Price Dataset/pp-2022.csv", col_names = FALSE) %>%
16 setNames(column_names)
17 pp_2023 = read_csv("D:/Data_Science_Assignment/Obtained_Data/House Price Dataset/pp-2023.csv", col_names = FALSE) %>%
18 setNames(column_names)
19
20 # Combine the datasets into one data frame
21 combined_data_houseprices = bind_rows(pp_2020,pp_2021,pp_2022,pp_2023)
22
23 # Clean the combined data of house prices
24 cleaned_data_houseprices = combined_data_houseprices %>%
25 as_tibble() %>% #convert to the tibble
26 na.omit() %>% #rows which have null values-removing that
27 mutate(Price = as.numeric(Price)) %>%
28 filter(County == "CITY OF BRISTOL" | County == "CORNWALL") %>% #filtering rows with bristol(city of bristol) and cornwall as county
29 mutate(Date of Transfer = year(ymd(`Date of Transfer`))) %>% #converting date of transfer column to a date format and then extract the year
30 mutate(S.No = row_number()) %>% #adding a S_No column
31 select(S.No,Price, `Date of Transfer`, Postcode, `Town/City`, District, County) %>% #selecting required columns only
32 mutate(Short Postcode = substr(Postcode, 1,5)) #now adding the another column to the combine dataset
33
34 # Define path to save the cleaned dataset
35 cleaned_houseprices_path = "D:/Data_Science_Assignment/Cleaned_Data/Cleaned_houseprices.csv"
36
37 # Define path to save the cleaned dataset
38 write.csv(cleaned_data_houseprices, cleaned_houseprices_path, row.names = FALSE)
39
40 print(cleaned_data_houseprices)
41 View(cleaned_data_houseprices)
42 str(cleaned_data_houseprices)

```

```

1 #Cleaning the data for house prices
2 library(dplyr)
3 library(tidyverse)
4 library(lubridate)
5
6 # Define column names for the datasets
7 column_names = c("Transaction_ID", "Price", "Date of Transfer", "Postcode", "Property_Type", "Old/New", "Duration", "PAON",
8 "SAON", "Street", "Locality", "Town/City", "District", "County", "PPD_Category_type", "Record_Status")
9
10 # Read and combine data for the years 2020 to 2023
11 pp_2020 = read_csv("D:/Data_Science_Assignment/Obtained_Data/House Price Dataset/pp-2020.csv", col_names = FALSE) %>%
12 setNames(column_names)
13 pp_2021 = read_csv("D:/Data_Science_Assignment/Obtained_Data/House Price Dataset/pp-2021.csv", col_names = FALSE) %>%
14 setNames(column_names)
15 pp_2022 = read_csv("D:/Data_Science_Assignment/Obtained_Data/House Price Dataset/pp-2022.csv", col_names = FALSE) %>%
16 setNames(column_names)
17 pp_2023 = read_csv("D:/Data_Science_Assignment/Obtained_Data/House Price Dataset/pp-2023.csv", col_names = FALSE) %>%
18 setNames(column_names)
19
20 # Combine the datasets into one data frame
21 combined_data_houseprices = bind_rows(pp_2020,pp_2021,pp_2022,pp_2023)
22
23 # Clean the combined data of house prices
24 cleaned_data_houseprices = combined_data_houseprices %>%
25 as_tibble() %>% #convert to the tibble
26 na.omit() %>% #rows which have null values-removing that
27 mutate(Price = as.numeric(Price)) %>%
28 filter(County == "CITY OF BRISTOL" | County == "CORNWALL") %>% #filtering rows with bristol(city of bristol) and cornwall as county
29 mutate(Date of Transfer = year(ymd(`Date of Transfer`))) %>% #converting date of transfer column to a date format and then extract the year
30 mutate(S.No = row_number()) %>% #adding a S_No column
31 select(S.No,Price, `Date of Transfer`, Postcode, `Town/City`, District, County) %>% #selecting required columns only
32 mutate(Short Postcode = substr(Postcode, 1,5)) #now adding the another column to the combine dataset
33
34 # Define path to save the cleaned dataset
35 cleaned_houseprices_path = "D:/Data_Science_Assignment/Cleaned_Data/Cleaned_houseprices.csv"
36
37 # Save the cleaned dataset
38 write.csv(cleaned_data_houseprices, cleaned_houseprices_path, row.names = FALSE)
39
40 print(cleaned_data_houseprices)
41 View(cleaned_data_houseprices)
42 str(cleaned_data_houseprices)

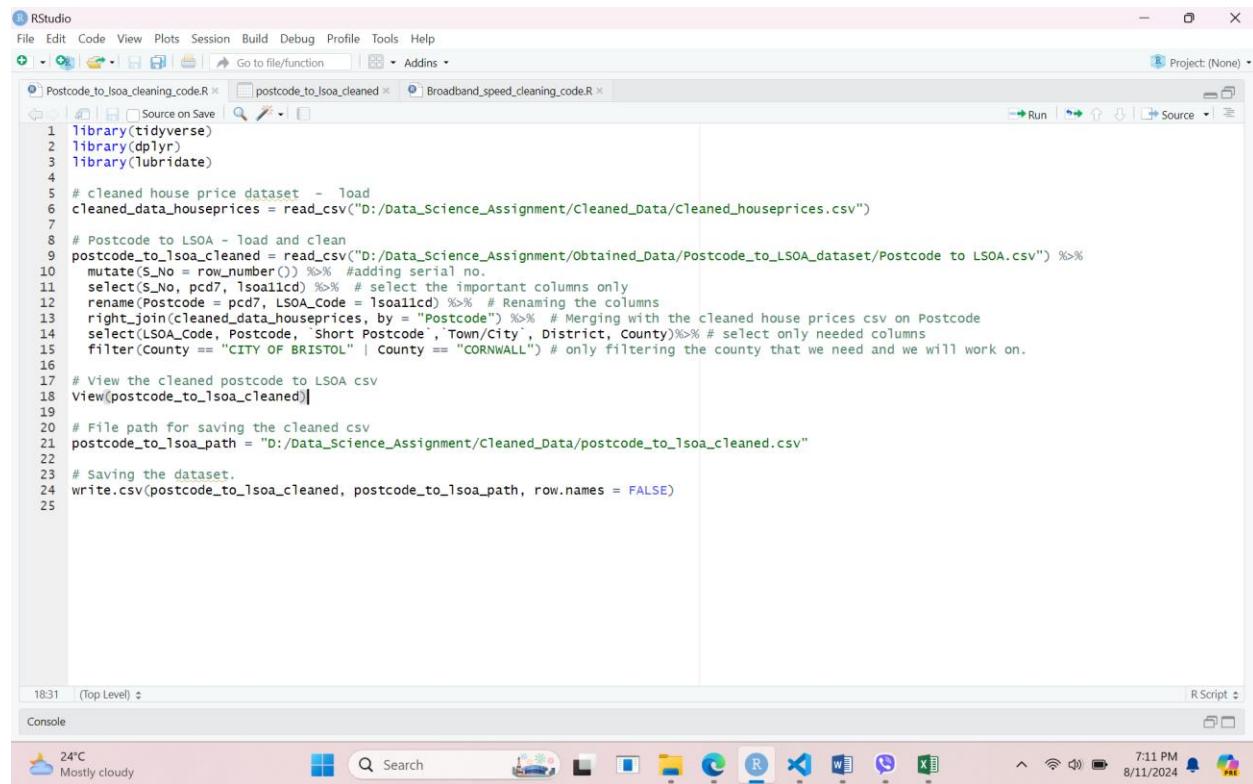
```

Towns and Postcodes

In order to align with the house price data, the postcode to LSOA dataset underwent preprocessing and cleaning. For distinct identity, a serial number column (S_No) was introduced. The only columns that were kept and renamed to comply with the naming conventions of the home price dataset were Postcode and LSOA_Code. This data was combined with the house price information using a right join based on the Postcode column. The two subsets of the combined dataset that were retained were "CITY OF BRISTOL" and "CORNWALL." Effective integration and clarity were ensured by saving the final cleaned dataset as a CSV file for additional analysis.

Figure 2

Postcode to Lsoa Cleaning



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - Project: (None) X
Postcode_to_lsoa_cleaning_code.R [ ] postcode_to_lsoa_cleaned [ ] Broadband_speed_cleaning_code.R [ ]
Source on Save Run Source
1 library(tidyverse)
2 library(dplyr)
3 library(lubridate)
4
5 # cleaned house price dataset - load
6 cleaned_data_houseprices = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Cleaned_houseprices.csv")
7
8 # Postcode to LSOA - load and clean
9 postcode_to_lsoa_cleaned = read_csv("D:/Data_Science_Assignment/Obtained_Data/Postcode_to_LSOA_dataset/Postcode_to_LSOA.csv") %>%
10   mutate(S_No = row_number()) %>% # adding serial no.
11   select(S_No, pcd7, lsoailcd) %>% # select the important columns only
12   rename(Postcode = pcd7, LSOA_Code = lsoailcd) %>% # Renaming the columns
13   right_join(cleaned_data_houseprices, by = "Postcode") %>% # Merging with the cleaned house prices csv on Postcode
14   select(LSOA_Code, Postcode, Short_Postcode, Town/City, District, County) %>% # select only needed columns
15   filter(County == "CITY OF BRISTOL" | County == "CORNWALL") # only filtering the county that we need and we will work on.
16
17 # View the cleaned postcode to LSOA csv
18 View(postcode_to_lsoa_cleaned)
19
20 # File path for saving the cleaned csv
21 postcode_to_lsoa_path = "D:/Data_Science_Assignment/Cleaned_Data/postcode_to_lsoa_cleaned.csv"
22
23 # Saving the dataset.
24 write.csv(postcode_to_lsoa_cleaned, postcode_to_lsoa_path, row.names = FALSE)
25

```

18:31 (Top Level) R Script

Console

24°C Mostly cloudy Search

7:11 PM 8/11/2024

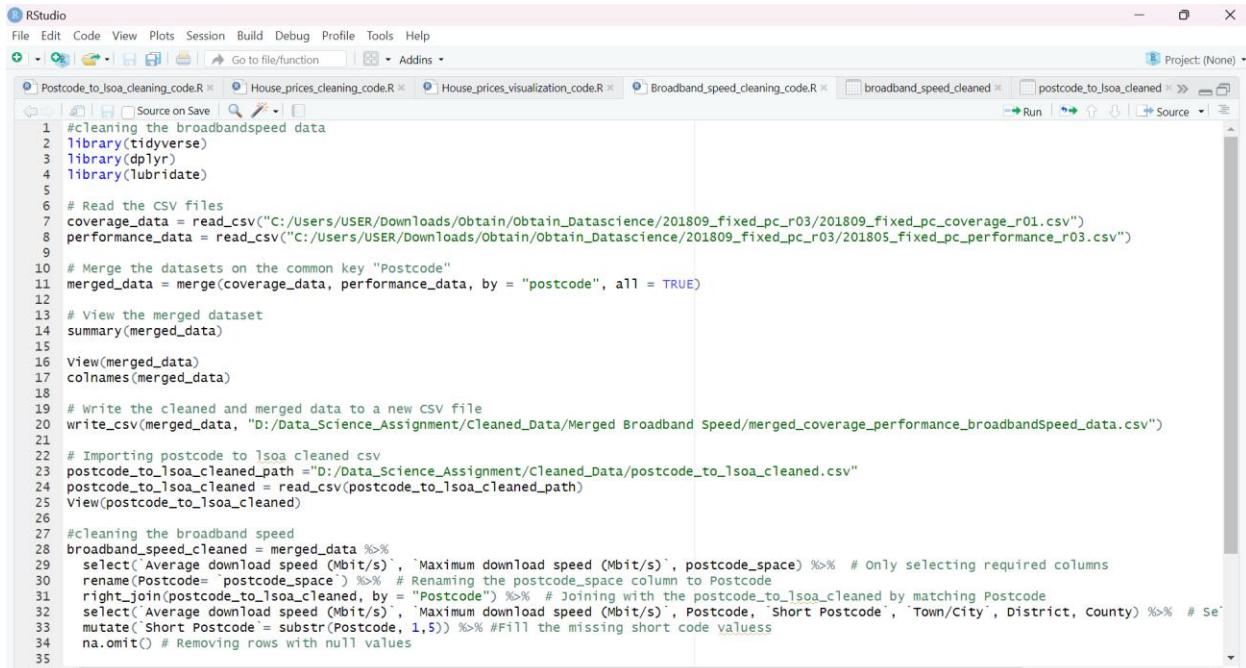
Broadband Speed

The broadband speed data required going through a number of steps of cleaning and preprocessing in order for it to be accurate and consistent with other datasets. Initially, the CSV files containing the coverage_data and performance_data were read and combined using the shared postcode key. Coverage and performance data were merged into a single dataset during this process. After merging, the dataset was reviewed to understand its structure and contents.

After that, the combined dataset was cleaned by selecting only important columns, such as postcode_space, Maximum download speed (Mbit/s), and Average download speed (Mbit/s). For uniformity, the postcode_space column was renamed to Postcode. After the postcode was cleaned, a right join was done with the LSOA dataset to match location data with broadband speed data. A Short Postcode column was added to the dataset, and any rows with missing values were eliminated. Ultimately, the dataset was cleaned and stored in a CSV file for later analysis. (Ofcom, n.d.)

Figure 3

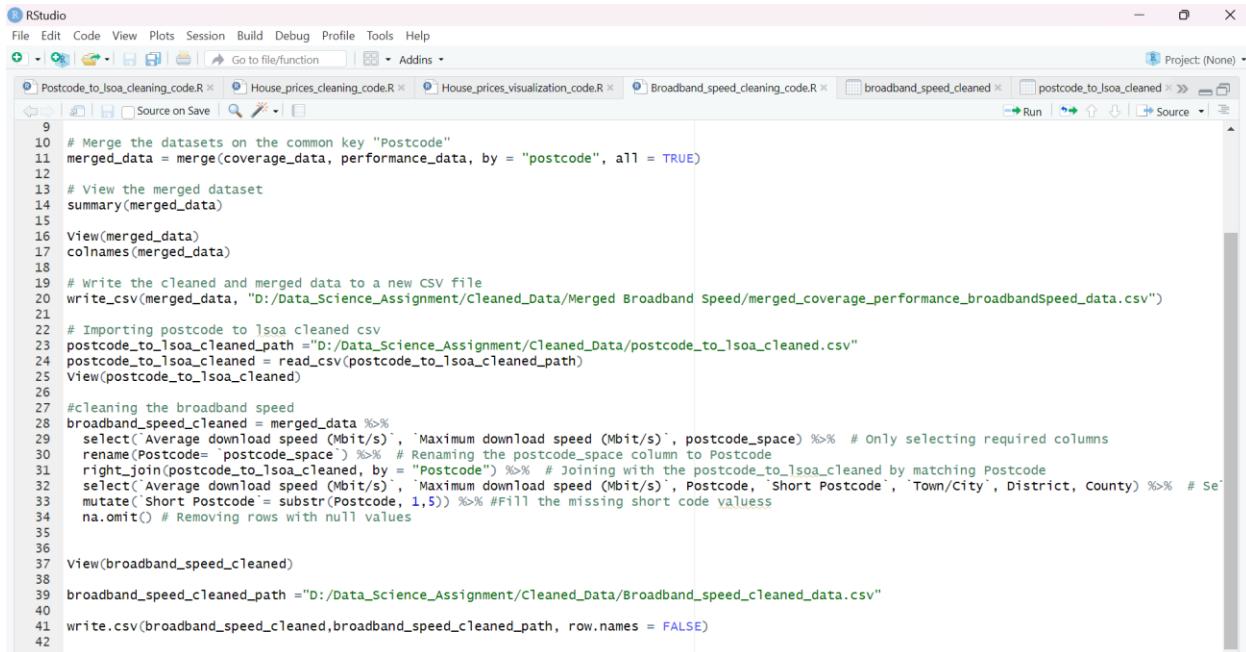
Broadband Speed Cleaning



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source on Save Go to file/function Addins Project: (None)
Postcode_to_lsoa_cleaning_code.R House_prices_cleaning_code.R House_prices_visualization_code.R Broadband_speed_cleaning_code.R broadband_speed_cleaned postcode_to_lsoa_cleaned >
1 #cleaning the broadband speed data
2 library(tidyverse)
3 library(dplyr)
4 library(lubridate)
5
6 # Read the CSV files
7 coverage_data = read_csv("C:/Users/USER/Downloads/Obtain/Obtain_Datascience/201809_fixed_pc_r03/201809_fixed_pc_coverage_r01.csv")
8 performance_data = read_csv("C:/Users/USER/Downloads/Obtain/Obtain_Datascience/201809_fixed_pc_r03/201805_fixed_pc_performance_r03.csv")
9
10 # Merge the datasets on the common key "Postcode"
11 merged_data = merge(coverage_data, performance_data, by = "postcode", all = TRUE)
12
13 # View the merged dataset
14 summary(merged_data)
15
16 View(merged_data)
17 colnames(merged_data)
18
19 # Write the cleaned and merged data to a new CSV file
20 write_csv(merged_data, "D:/Data_Science_Assignment/Cleaned_Data/Merged_Broadband_Speed/merged_coverage_performance_broadbandSpeed_data.csv")
21
22 # Importing postcode to lsoa cleaned csv
23 postcode_to_lsoa_cleaned_path ="D:/Data_Science_Assignment/Cleaned_Data/postcode_to_lsoa_cleaned.csv"
24 postcode_to_lsoa_cleaned = read_csv(postcode_to_lsoa_cleaned_path)
25 View(postcode_to_lsoa_cleaned)
26
27 #cleaning the broadband speed
28 broadband_speed_cleaned = merged_data %>%
29   select(`Average download speed (Mbit/s)`, `Maximum download speed (Mbit/s)`, postcode_space) %>% # Only selecting required columns
30   rename(Postcode= `postcode_space`) %>% # Renaming the postcode_space column to Postcode
31   right_join(postcode_to_lsoa_cleaned, by = "Postcode") %>% # Joining with the postcode_to_lsoa_cleaned by matching Postcode
32   select(`Average download speed (Mbit/s)`, `Maximum download speed (Mbit/s)`, Postcode, Short Postcode, `Town/City`, District, County) %>% # Selecting the required columns
33   mutate(`Short Postcode` = substr(Postcode, 1,5)) %>% #Fill the missing short code values
34   na.omit() # Removing rows with null values
35
36
37 View(broadband_speed_cleaned)
38
39 broadband_speed_cleaned_path ="D:/Data_Science_Assignment/Cleaned_Data/Broadband_speed_cleaned_data.csv"
40
41 write.csv(broadband_speed_cleaned,broadband_speed_cleaned_path, row.names = FALSE)
42

```



```

9
10 # Merge the datasets on the common key "Postcode"
11 merged_data = merge(coverage_data, performance_data, by = "postcode", all = TRUE)
12
13 # View the merged dataset
14 summary(merged_data)
15
16 View(merged_data)
17 colnames(merged_data)
18
19 # Write the cleaned and merged data to a new CSV file
20 write_csv(merged_data, "D:/Data_Science_Assignment/Cleaned_Data/Merged_Broadband_Speed/merged_coverage_performance_broadbandSpeed_data.csv")
21
22 # Importing postcode to lsoa cleaned csv
23 postcode_to_lsoa_cleaned_path ="D:/Data_Science_Assignment/Cleaned_Data/postcode_to_lsoa_cleaned.csv"
24 postcode_to_lsoa_cleaned = read_csv(postcode_to_lsoa_cleaned_path)
25 View(postcode_to_lsoa_cleaned)
26
27 #cleaning the broadband speed
28 broadband_speed_cleaned = merged_data %>%
29   select(`Average download speed (Mbit/s)`, `Maximum download speed (Mbit/s)`, postcode_space) %>% # Only selecting required columns
30   rename(Postcode= `postcode_space`) %>% # Renaming the postcode_space column to Postcode
31   right_join(postcode_to_lsoa_cleaned, by = "Postcode") %>% # Joining with the postcode_to_lsoa_cleaned by matching Postcode
32   select(`Average download speed (Mbit/s)`, `Maximum download speed (Mbit/s)`, Postcode, Short Postcode, `Town/City`, District, County) %>% # Selecting the required columns
33   mutate(`Short Postcode` = substr(Postcode, 1,5)) %>% #Fill the missing short code values
34   na.omit() # Removing rows with null values
35
36
37 View(broadband_speed_cleaned)
38
39 broadband_speed_cleaned_path ="D:/Data_Science_Assignment/Cleaned_Data/Broadband_speed_cleaned_data.csv"
40
41 write.csv(broadband_speed_cleaned,broadband_speed_cleaned_path, row.names = FALSE)
42

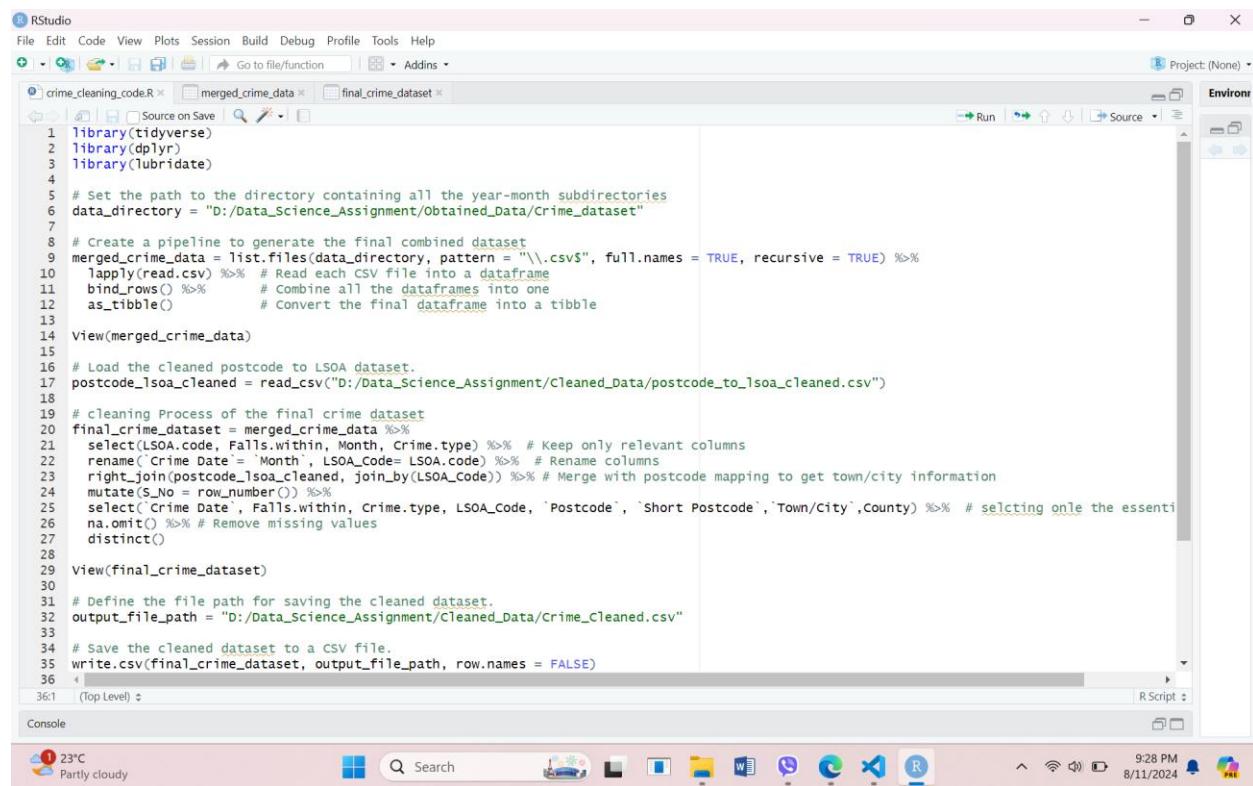
```

Crime

Using `list.files()` and `bind_rows()`, CSV files from several year-month subdirectories were combined into a single dataframe to clean and preprocess the crime dataset. After that, the data was transformed into a tibble for simpler management. For consistency, important columns like `LSOA.code`, `Falls.within`, `Month`, and `Crime.type` were kept but given new names (`Month` became `Crime Date`, for example). To include town/city information, the dataset was combined with postcode data that had been cleansed earlier and added to LSOA. Missing values were eliminated, unnecessary columns were eliminated, and a serial number (`S_No`) was inserted. For additional study, the cleaned dataset was stored as a CSV file. (Police, n.d.)

Figure 4

Crime Cleaning



The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project:** Project: (None)
- Code Editor:** The code is contained in a file named `crime_cleaning_code.R`. The code uses the tidyverse, dplyr, and lubridate packages to read CSV files from a directory, bind them into a tibble, and then clean the final dataset by selecting relevant columns, renaming them, merging with a postcode dataset, and saving it as a CSV file.
- Environment:** Shows the current environment variables.
- Console:** Shows the command `source("crime_cleaning_code.R")`.
- System Status:** Shows the date and time as 8/11/2024 9:28 PM, and the weather as 23°C Partly cloudy.

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - X
File Edit Code View Plots Session Build Debug Profile Tools Help
crime_cleaning_code.R merged_crime_data final_crime_dataset
Source on Save Go to file/function Run Source Environment
1 library(tidyverse)
2 library(dplyr)
3 library(lubridate)
4
5 # Set the path to the directory containing all the year-month subdirectories
6 data_directory = "D:/Data_Science_Assignment/Obtained_Data/Crime_dataset"
7
8 # Create a pipeline to generate the final combined dataset
9 merged_crime_data = list.files(data_directory, pattern = "\\.csv$", full.names = TRUE, recursive = TRUE) %>%
10   lapply(read.csv) %>% # Read each CSV file into a dataframe
11   bind_rows() %>% # Combine all the dataframes into one
12   as_tibble() # Convert the final dataframe into a tibble
13
14 View(merged_crime_data)
15
16 # Load the cleaned postcode to LSOA dataset.
17 postcode_lsoa_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/postcode_to_lsoa_cleaned.csv")
18
19 # cleaning Process of the final crime dataset
20 final_crime_dataset = merged_crime_data %>%
21   select(LSOA.code, Falls.within, Month, Crime.type) %>% # Keep only relevant columns
22   rename(`Crime Date` = Month, LSOA_Code= LSOA.code) %>% # Rename columns
23   right_join(postcode_lsoa_cleaned, join_by(LSOA.Code)) %>% # Merge with postcode mapping to get town/city information
24   mutate(S_No = row_number()) %>%
25   select(`Crime Date` , Falls.within, crime.type, LSOA_Code, `Postcode`, `Short Postcode`, `Town/City`, County) %>% # selecting only the essential
26   na.omit() %>% # Remove missing values
27   distinct()
28
29 View(final_crime_dataset)
30
31 # Define the file path for saving the cleaned dataset.
32 output_file_path = "D:/Data_Science_Assignment/Cleaned_Data/Crime_Cleaned.csv"
33
34 # Save the cleaned dataset to a CSV file.
35 write.csv(final_crime_dataset, output_file_path, row.names = FALSE)
36
37 (Top Level) R Script
Console
23°C Partly cloudy Search 9:28 PM 8/11/2024

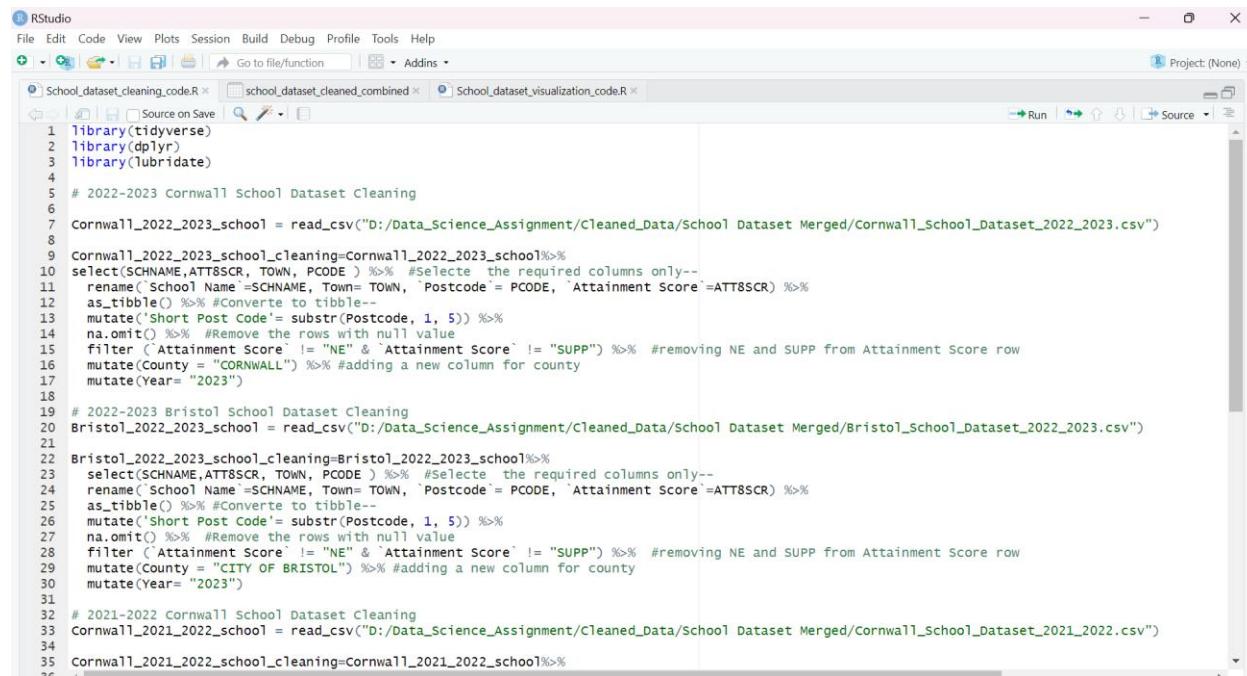
```

School

The school datasets for 2021–2022 and 2022–2023 were cleaned to ensure consistency between Bristol and Cornwall. Key columns (SCHNAME, ATT8SCR, TOWN, and PCODE) were selected, and column names were standardized. A new column, Short Post Code, was created from the first five characters of the Postcode. Rows with null values or "NE" or "SUPP" in the Attainment Score column were removed. The datasets were then combined into a single tibble, adding County and Year columns for context. The cleaned dataset was saved as a CSV file for further analysis. (School dataset, UK Government Compare School Performance Service, n.d.)

Figure 5

School Cleaning

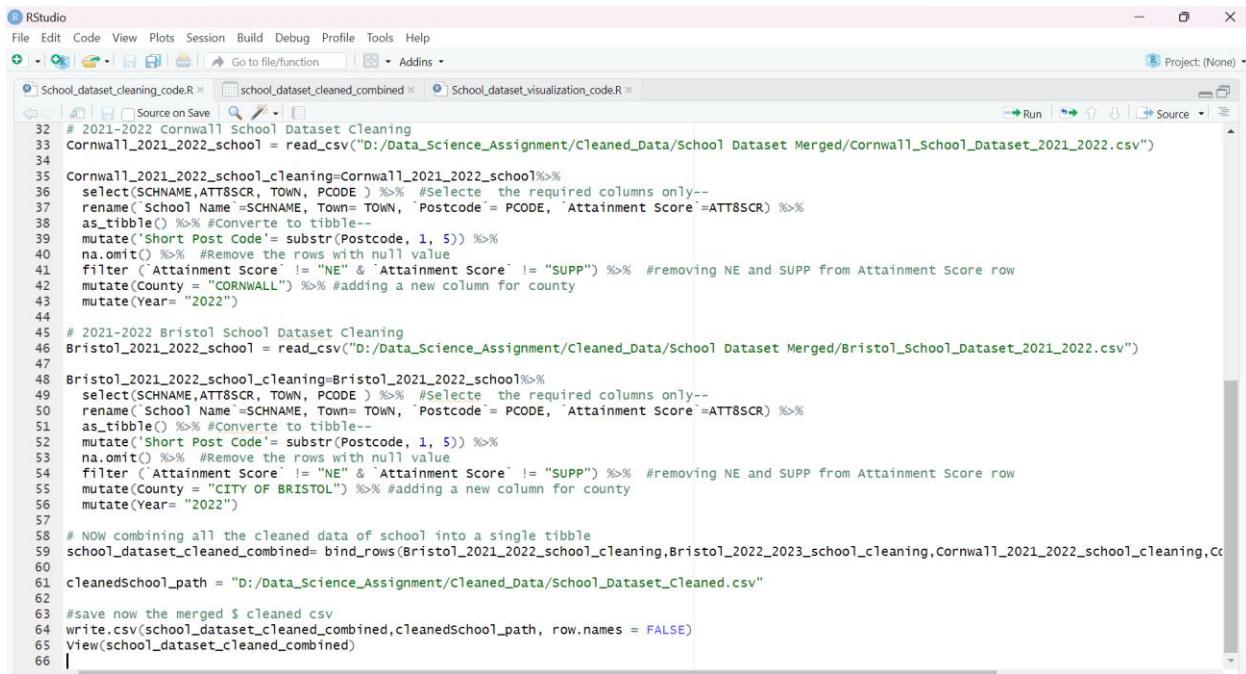


The screenshot shows the RStudio interface with three tabs open: 'School_dataset_cleaning_code.R', 'school_dataset_cleaned_combined', and 'School_dataset_visualization_code.R'. The 'School_dataset_cleaning_code.R' tab contains the following R code:

```

1 library(tidyverse)
2 library(dplyr)
3 library(lubridate)
4
5 # 2022-2023 Cornwall School Dataset Cleaning
6
7 Cornwall_2022_2023_school = read_csv("D:/Data_Science_Assignment/Cleaned_Data/School Dataset Merged/cornwall_School_Dataset_2022_2023.csv")
8
9 Cornwall_2022_2023_school_cleaning=Cornwall_2022_2023_school%>%
10 select(SCHNAME,ATT8SCR, TOWN, PCODE ) %>% #Selecte the required columns only--
11 rename(`School Name` =SCHNAME, `Town` = TOWN, `Postcode` = PCODE, `Attainment Score` =ATT8SCR) %>%
12 as_tibble() %>% #Converte to tibble--
13 mutate(`Short Post Code` = substr(Postcode, 1, 5)) %>%
14 na.omit() %>% #Remove the rows with null value
15 filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>% #removing NE and SUPP from Attainment Score row
16 mutate(`County` = "CORNWALL") %>% #adding a new column for county
17 mutate(`Year` = "2023")
18
19 # 2022-2023 Bristol School Dataset Cleaning
20 Bristol_2022_2023_school = read_csv("D:/Data_Science_Assignment/Cleaned_Data/School Dataset Merged/Bristol_School_Dataset_2022_2023.csv")
21
22 Bristol_2022_2023_school_cleaning=Bristol_2022_2023_school%>%
23 select(SCHNAME,ATT8SCR, TOWN, PCODE ) %>% #Selecte the required columns only--
24 rename(`School Name` =SCHNAME, `Town` = TOWN, `Postcode` = PCODE, `Attainment Score` =ATT8SCR) %>%
25 as_tibble() %>% #Converte to tibble--
26 mutate(`Short Post Code` = substr(Postcode, 1, 5)) %>%
27 na.omit() %>% #Remove the rows with null value
28 filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>% #removing NE and SUPP from Attainment Score row
29 mutate(`County` = "CITY OF BRISTOL") %>% #adding a new column for county
30 mutate(`Year` = "2023")
31
32 # 2021-2022 Cornwall School Dataset Cleaning
33 Cornwall_2021_2022_school = read_csv("D:/Data_Science_Assignment/Cleaned_Data/School Dataset Merged/cornwall_School_Dataset_2021_2022.csv")
34
35 Cornwall_2021_2022_school_cleaning=cornwall_2021_2022_school%>%
36

```



RStudio interface showing R code for school dataset cleaning. The code reads three CSV files (Cornwall, Bristol, and a merged version) and performs various data manipulations like selecting columns, converting to tibbles, and filtering rows. It then binds the cleaned datasets together and writes a final cleaned CSV file.

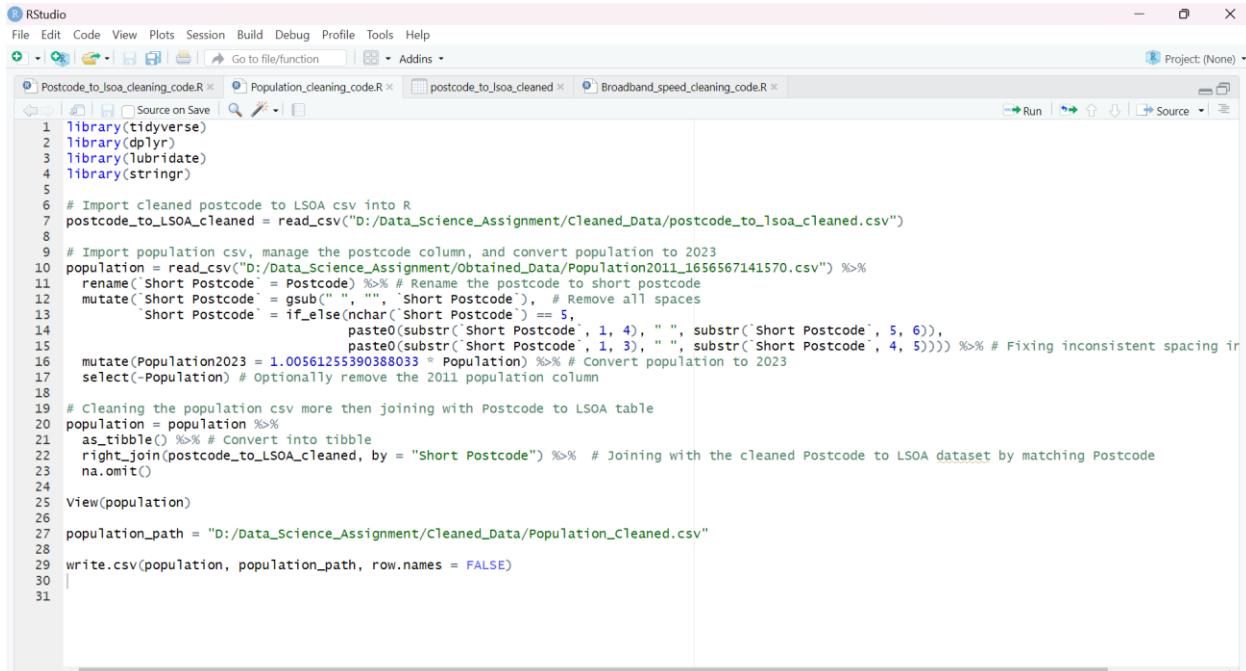
```

32 # 2021-2022 Cornwall School Dataset Cleaning
33 Cornwall_2021_2022_school = read_csv("D:/Data_Science_Assignment/Cleaned_Data/School Dataset Merged/cornwall_School_Dataset_2021_2022.csv")
34
35 Cornwall_2021_2022_school_cleaning<-Cornwall_2021_2022_school%>%
36   select(SCHNAME,ATT8SCR, TOWN, PCODE ) %>% #Selecte the required columns only-
37   rename(`School Name` =SCHNAME, `Town` = TOWN, `Postcode` = PCODE, `Attainment Score` =ATT8SCR) %>%
38   as_tibble() %>% #Converte to tibble-
39   mutate(`Short Post Code`= substr(Postcode, 1, 5)) %>%
40   na.omit() %>% #Remove the rows with null value
41   filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>% #removing NE and SUPP from Attainment Score row
42   mutate(County = "CORNWALL") %>% #adding a new column for county
43   mutate(Year= "2022")
44
45 # 2021-2022 Bristol School Dataset Cleaning
46 Bristol_2021_2022_school = read_csv("D:/Data_Science_Assignment/Cleaned_Data/School Dataset Merged/Bristol_School_Dataset_2021_2022.csv")
47
48 Bristol_2021_2022_school_cleaning<-Bristol_2021_2022_school%>%
49   select(SCHNAME,ATT8SCR, TOWN, PCODE ) %>% #Selecte the required columns only-
50   rename(`School Name` =SCHNAME, `Town` = TOWN, `Postcode` = PCODE, `Attainment Score` =ATT8SCR) %>%
51   as_tibble() %>% #Converte to tibble-
52   mutate(`Short Post Code`= substr(Postcode, 1, 5)) %>%
53   na.omit() %>% #Remove the rows with null value
54   filter(`Attainment Score` != "NE" & `Attainment Score` != "SUPP") %>% #removing NE and SUPP from Attainment Score row
55   mutate(County = "CITY OF BRISTOL") %>% #adding a new column for county
56   mutate(Year= "2022")
57
58 # NOW combining all the cleaned data of school into a single tibble
59 school_dataset_cleaned_combined<- bind_rows(Bristol_2021_2022_school_cleaning,Bristol_2022_2023_school_cleaning,Cornwall_2021_2022_school_cleaning,Co
60
61 cleanedSchool_path = "D:/Data_Science_Assignment/Cleaned_Data/School_Dataset_Cleaned.csv"
62
63 #save now the merged & cleaned csv
64 write.csv(school_dataset_cleaned_combined,cleanedSchool_path, row.names = FALSE)
65 View(school_dataset_cleaned_combined)
66

```

Figure 6

Population Cleaning



RStudio interface showing R code for population dataset cleaning. The code imports a CSV of postcodes to LSOAs, merges it with a population CSV, and then joins it with a cleaned postcode dataset to produce a final population CSV.

```

1 library(tidyverse)
2 library(dplyr)
3 library(lubridate)
4 library(stringr)
5
6 # Import cleaned postcode to LSOA csv into R
7 postcode_to_lsoa_cleaned = read_csv("D:/Data_Science_Assignment/cleaned_Data/postcode_to_lsoa_cleaned.csv")
8
9 # Import population csv , manage the postcode column, and convert population to 2023
10 population = read_csv("D:/Data_Science_Assignment/Obtained_Data/Population2011_1656567141570.csv") %>%
11   rename(`Short Postcode` = Postcode) %>% # Rename the postcode to short postcode
12   mutate(`Short Postcode` = gsub(" ", "", `Short Postcode`), # Remove all spaces
13     `Short Postcode` = if_else(nchar(`Short Postcode` ) == 5,
14       paste0(substr(`Short Postcode` , 1, 4), " ", substr(`Short Postcode` , 5, 6)),
15       paste0(substr(`Short Postcode` , 1, 3), " ", substr(`Short Postcode` , 4, 5))) %>% # Fixing inconsistent spacing in
16   mutate(Population2023 = 1.00561255390388033 * Population) %>% # Convert population to 2023
17   select(-Population) # Optionally remove the 2011 population column
18
19 # Cleaning the population csv more then joining with Postcode to LSOA table
20 population = population %>%
21   as_tibble() %>% # Convert into tibble
22   right_join(postcode_to_lsoa_cleaned, by = "Short Postcode") %>% # Joining with the cleaned Postcode to LSOA dataset by matching Postcode
23   na.omit()
24
25 View(population)
26
27 population_path = "D:/Data_Science_Assignment/Cleaned_Data/Population_Cleaned.csv"
28
29 write.csv(population, population_path, row.names = FALSE)
30

```

Exploratory Analysis

House Prices

In the EDA of the house prices dataset, we used several visualizations and summary statistics. A box plot was created to show the range of average house prices for Bristol and Cornwall in 2023, highlighting typical price ranges and outliers. A bar chart compared average prices between the two counties, and a line graph tracked price trends from 2020-2023. By summarizing prices across towns, districts, and counties, we identified key patterns and trends in the data. (Wickham, n.d.)

Figure 7

House Price Visualization

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None) Environment

Files

Box Plot for 2023 Average House Prices By County

County

CITY OF CORNWALL

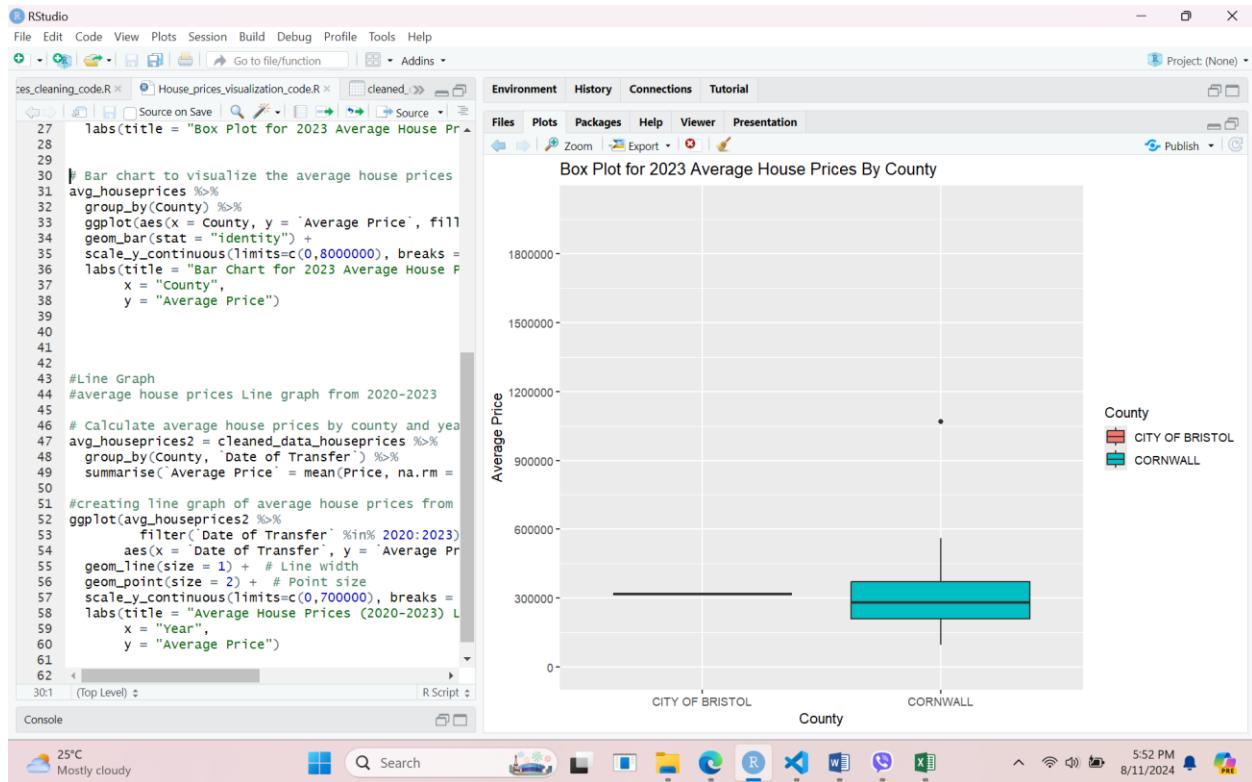
```
1 #Exploratory data analysis
2 library(ggplot2)
3 library(scales)
4 library(tidyverse)
5 library(dplyr)
6 library(lubridate)
7
8 cleaned_data_houseprices=read_csv("D:/Data_Science_Assignment/Cleaned_Data/Cleaned_houseprices.csv")
9
10 # Filter data for the year 2023
11 data_2023 = cleaned_data_houseprices %>%
12   filter(`Date of Transfer` == 2023)
13
14 # Find the average house price for each Town/City, District, and County in 2023
15 avg_houseprices = data_2023 %>%
16   group_by(Town/City ,District,County, `Date of Transfer`) %>%
17   summarise(`Average Price` = mean(Price)) %>%
18   ungroup(Town/City',District,County, `Date of Transfer`)
19
20 # Box plot to visualize average house prices of bristol and cornwall in 2023.
21 avg_houseprices %>%
22   group_by(County) %>%
23   ggplot(aes(x =County, y = `Average Price`, fill = County)) +
24   geom_boxplot() +
25   scale_y_continuous(limits=c(0,2000000), breaks = seq(0,2000000,300000))+ #setting limits and breaks
26   labs(title = "Box Plot for 2023 Average House Prices By County")
27
28
29 # Bar chart to visualize the average house prices of bristol and cornwall in 2023.
30 avg_houseprices %>%
31   group_by(County) %>%
32   ggplot(aes(x =County, y = `Average Price`, fill = County)) +
33   geom_bar(stat = "identity") +
34   scale_y_continuous(limits=c(0,8000000), breaks = seq(0,8000000,800000))+ #setting limits and breaks
35   labs(title = "Bar Chart for 2023 Average House Prices by County",
36   (Top Level) :
```

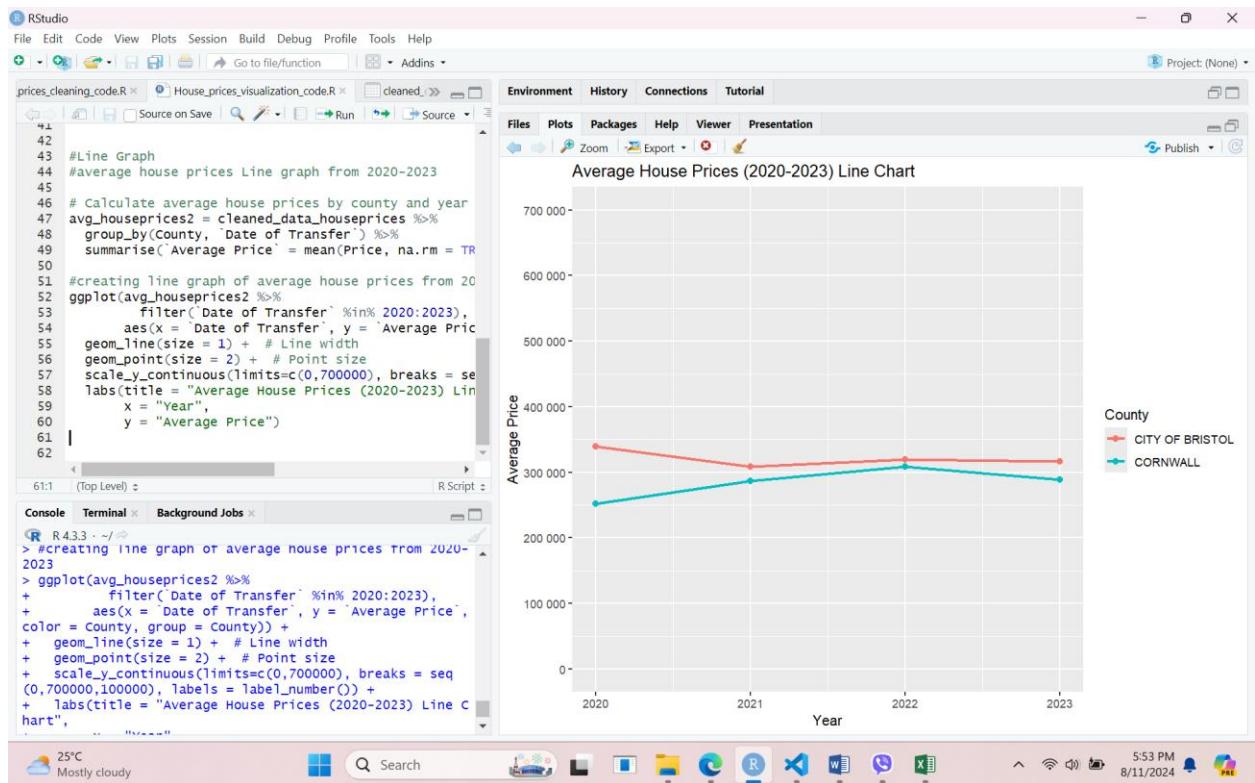
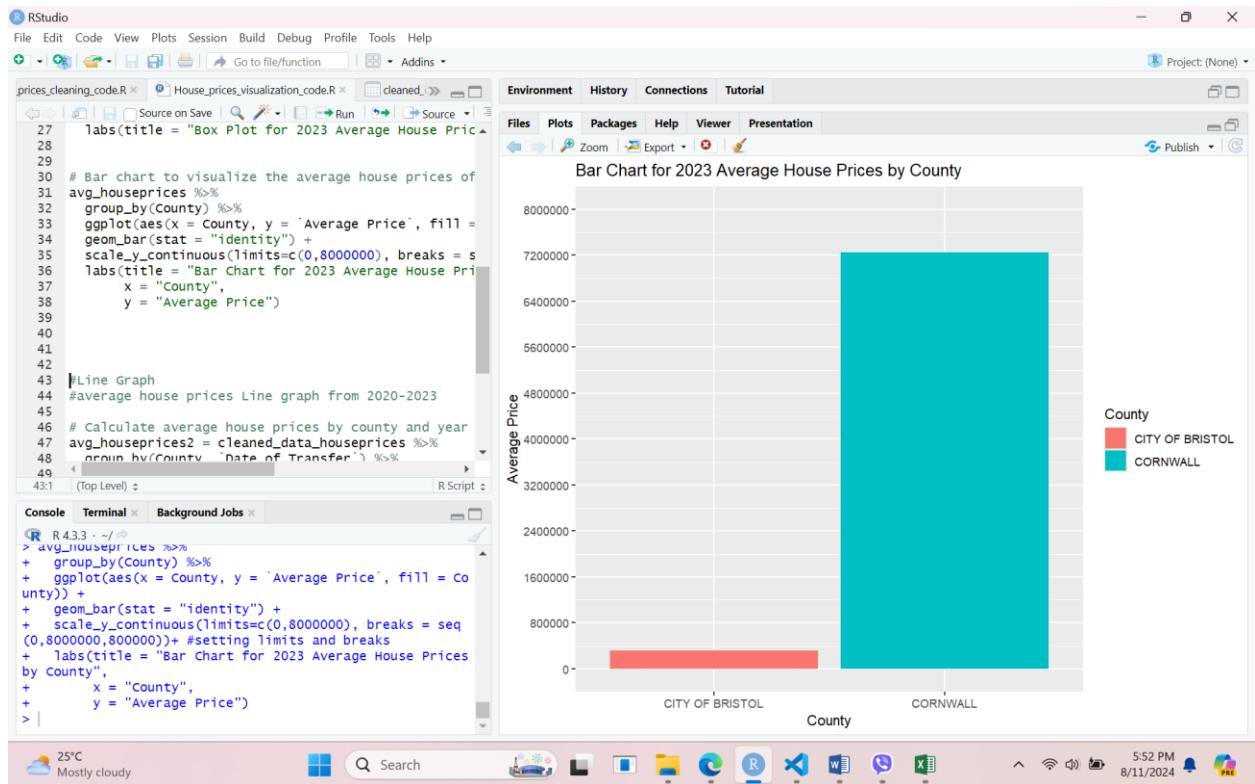
RStudio

```

File Edit Code View Plots Session Build Debug Profile Tools Help
+ - ○ X
Source on Save Go to file/function Addins
Project: (None)
Environment
Files Box Plot for:
27   labs(title = "Box Plot for 2023 Average House Prices By County")
28
29
30 # Bar chart to visualize the average house prices of bristol and cornwall in 2023.
31 avg_houseprices %>%
32   group_by(County) %>%
33   ggplot(aes(x = County, y = `Average Price`, fill = County)) +
34   geom_bar(stat = "identity") +
35   scale_y_continuous(limits=c(0,8000000), breaks = seq(0,8000000,800000))+ #setting limits and breaks
36   labs(title = "Bar Chart for 2023 Average House Prices by County",
37     x = "County",
38     y = "Average Price")
39
40
41
42
43 #Line Graph
44 #average house prices Line graph from 2020-2023
45
46 # Calculate average house prices by county and year
47 avg_houseprices2 = cleaned_data_houseprices %>%
48   group_by(County, `Date of Transfer`) %>%
49   summarise(`Average Price` = mean(Price, na.rm = TRUE), .groups = "drop")
50
51 #creating line graph of average house prices from 2020-2023
52 ggplot(avg_houseprices2 %>%
53   filter(`Date of Transfer` %in% 2020:2023),
54   aes(x = `Date of Transfer` , y = `Average Price`, color = County, group = County)) +
55   geom_line(size = 1) + # Line width
56   geom_point(size = 2) + # Point size
57   scale_y_continuous(limits=c(0,700000), breaks = seq(0,700000,100000), labels = label_number()) +
58   labs(title = "Average House Prices (2020-2023) Line Chart",
59     x = "Year",
60     y = "Average Price")
61
62

```



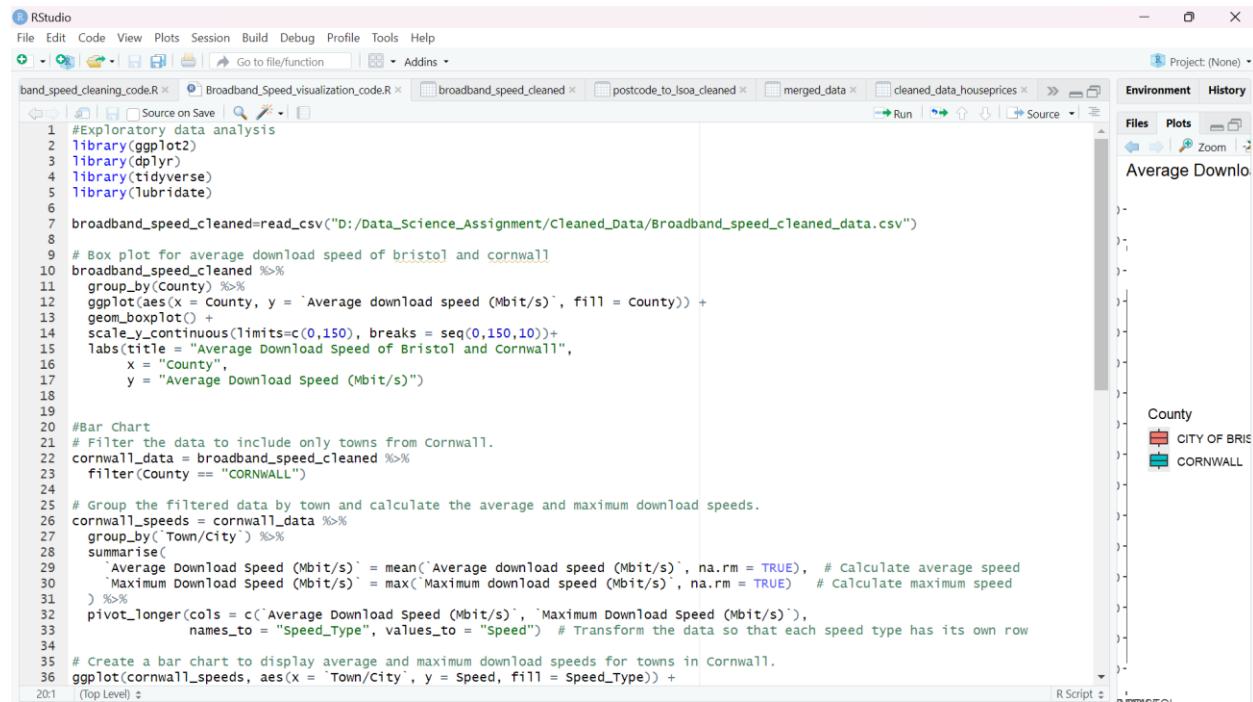


Broadband Speeds

In the exploratory data analysis of broadband speeds for Bristol and Cornwall, a box plot was created to compare average download speeds, highlighting variations and outliers between the two counties. Bar charts were also made to compare average and maximum download speeds for towns within each county, revealing which towns have the best and worst broadband performance. These visualizations provided insights into broadband connectivity across Bristol and Cornwall.

Figure 8

Broadband Speed Visualization



The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Plots Tab:** Active tab, showing a box plot titled "Average Download Speed (Mbit/s)" comparing "CITY OF BRIS" (red) and "CORNWALL" (blue).
- Code Editor:** Displays R code for data cleaning and visualization.
- Environment Tab:** Shows variables and objects available in the environment.
- History Tab:** Shows the history of R commands run.
- Code Area:**

```

1 #Exploratory data analysis
2 library(ggplot2)
3 library(dplyr)
4 library(tidyverse)
5 library(lubridate)
6
7 broadband_speed_cleaned=read_csv("D:/Data_Science_Assignment/Cleaned_Data/Broadband_speed_cleaned_data.csv")
8
9 # Box plot for average download speed of bristol and cornwall
10 broadband_speed_cleaned %>%
11   group_by(County) %>%
12   ggplot(aes(x = County, y = `Average download speed (Mbit/s)`, fill = County)) +
13   geom_boxplot() +
14   scale_y_continuous(limits=c(0,150), breaks = seq(0,150,10))+
15   labs(title = "Average Download Speed of Bristol and Cornwall",
16       x = "County",
17       y = "Average Download Speed (Mbit/s)")
18
19
20 #Bar Chart
21 # Filter the data to include only towns from Cornwall.
22 cornwall_data = broadband_speed_cleaned %>%
23   filter(County == "CORNWALL")
24
25 # Group the filtered data by town and calculate the average and maximum download speeds.
26 cornwall_speeds = cornwall_data %>%
27   group_by(Town/City) %>%
28   summarise(
29     `Average Download Speed (Mbit/s)` = mean(`Average download speed (Mbit/s)`, na.rm = TRUE), # Calculate average speed
30     `Maximum Download Speed (Mbit/s)` = max(`Maximum download speed (Mbit/s)`, na.rm = TRUE) # Calculate maximum speed
31   ) %>%
32   pivot_longer(cols = c(`Average Download Speed (Mbit/s)`, `Maximum Download Speed (Mbit/s)`),
33                 names_to = "Speed_Type", values_to = "Speed") # Transform the data so that each speed type has its own row
34
35 # Create a bar chart to display average and maximum download speeds for towns in cornwall.
36 ggplot(cornwall_speeds, aes(x = `Town/City`, y = Speed, fill = Speed_Type)) +
37   (Top Level) +
  
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

band_speed_cleaning_code.R broadband_Speed_visualization_code.R broadband_speed_cleaned postcode_to_lsoa_cleaned merged_data cleaned_data_houseprices Run Source

Create a bar chart to display average and maximum download speeds for towns in Cornwall.

ggplot(cornwall_speeds, aes(x = "Town/City", y = Speed, fill = Speed_Type)) +
geom_bar(stat = "identity", position = position_dodge()) + # Create the bars and separate them by speed type
scale_y_continuous(limits=c(0,350), breaks = seq(0,350,30)) +
coord_flip() +
labs(title = "Average and Maximum Download Speeds by Town for Cornwall",
x = "Town/City",
y = "Speed (Mbit/s)")

Filter the data to include only towns from Bristol.

bristol_data = broadband_speed_cleaned %>%
filter(County == "CITY OF BRISTOL")

Group the filtered data by town and calculate the average and maximum download speeds.

bristol_speeds = bristol_data %>%
group_by("Town/City") %>%
summarise(
"Average Download Speed (Mbit/s)" = mean(`Average download speed (Mbit/s)`, na.rm = TRUE), # Calculate average speed
"Maximum Download Speed (Mbit/s)" = mean(`Maximum download speed (Mbit/s)`, na.rm = TRUE) # Calculate maximum speed
) %>%
pivot_longer(cols = c(`Average Download Speed (Mbit/s)`, `Maximum Download Speed (Mbit/s)`),
names_to = "Speed_Type", values_to = "Speed") # Transform the data so that each speed type has its own row

Bar chart to display average and maximum download speeds for towns in Bristol.

ggplot(bristol_speeds, aes(x = "Town/City", y = Speed, fill = Speed_Type)) +
geom_bar(stat = "identity", position = position_dodge()) + # Create the bars and separate them by speed type
coord_flip() +
labs(title = "Average and Maximum Download Speeds by Town for Bristol",
x = "Town/City",
y = "Speed (Mbit/s)") +
scale_y_continuous(limits=c(0,200), breaks = seq(0,200,10))

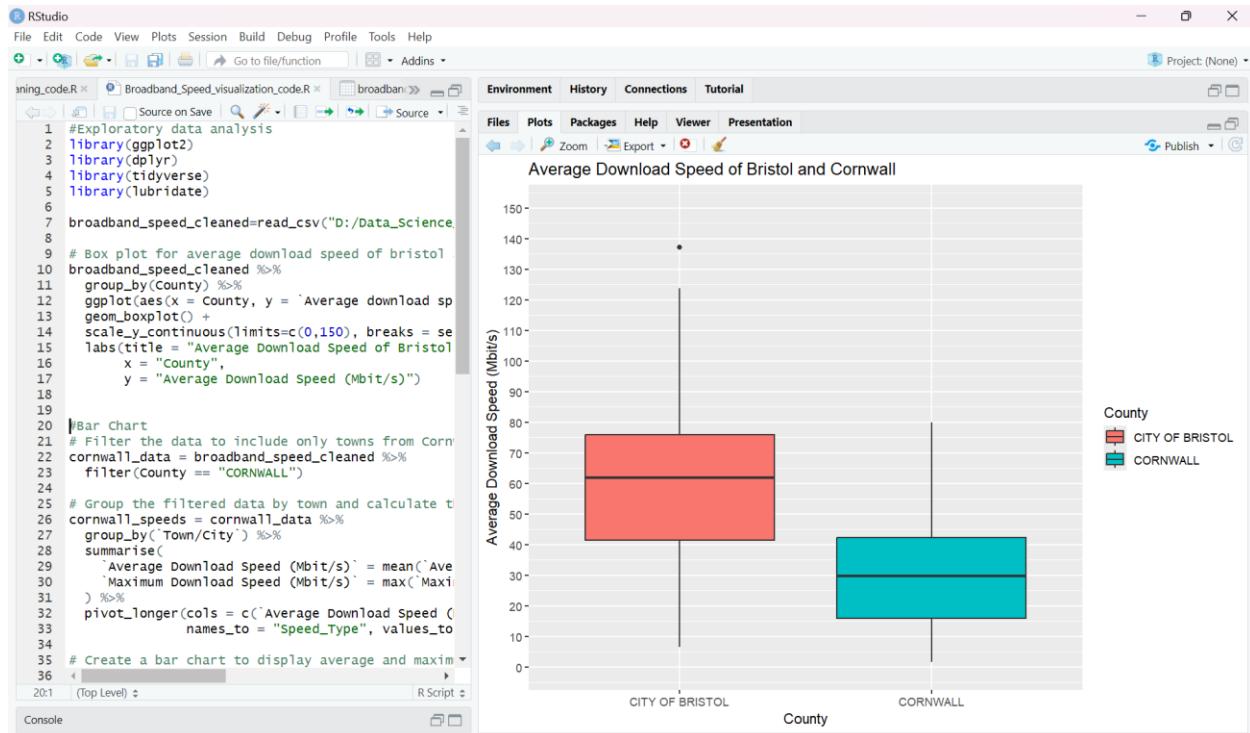
Environment History

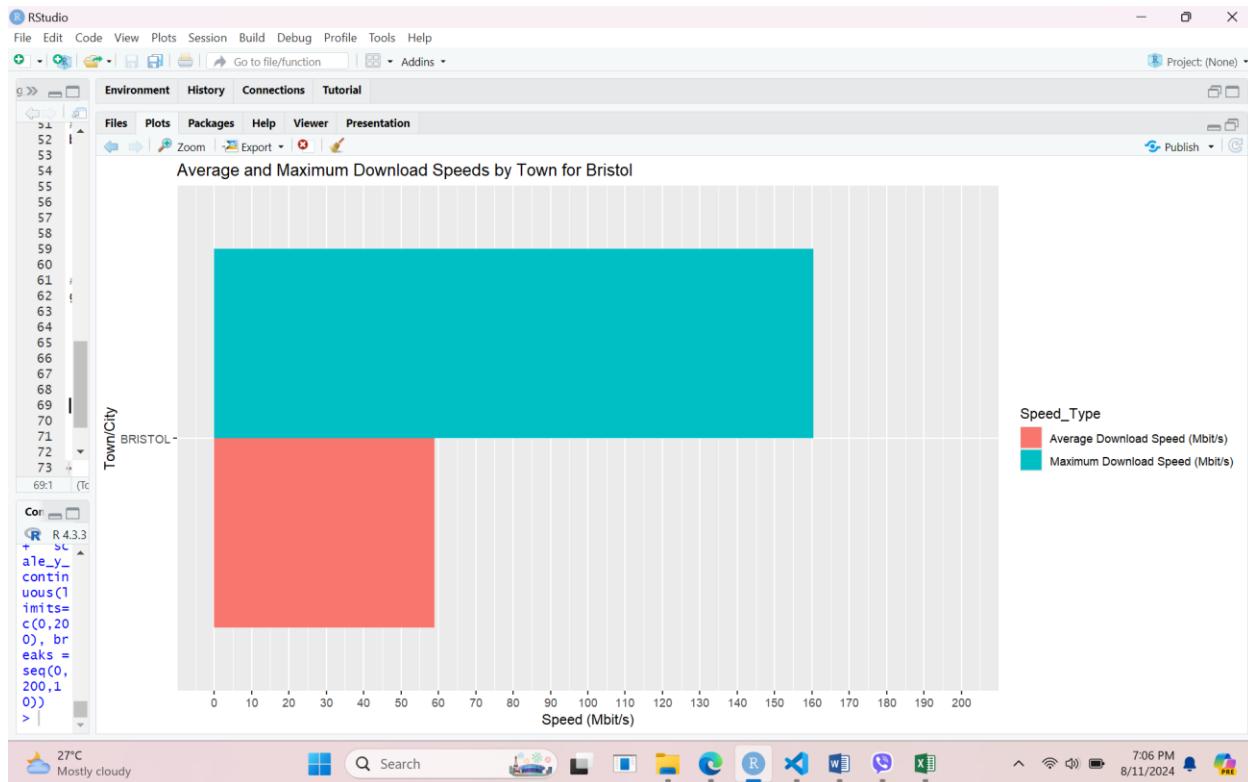
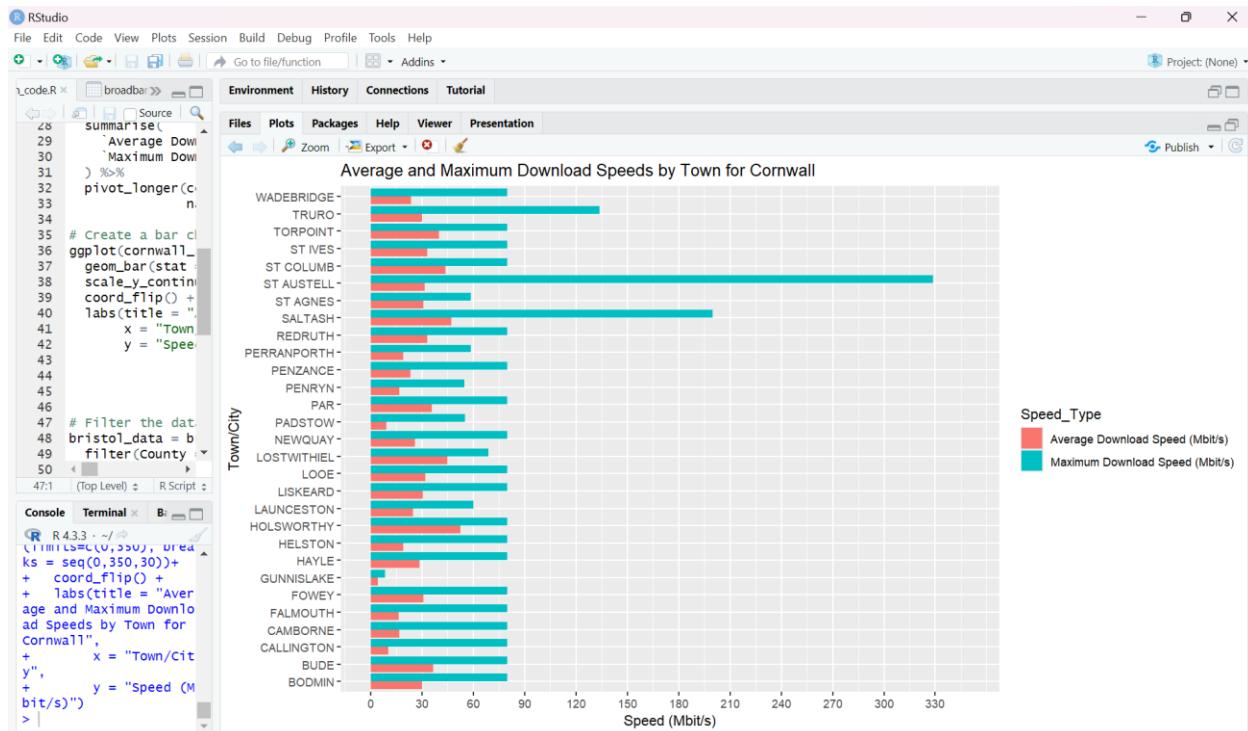
Files Plots Zoom

Average Download Speeds by Town for Cornwall

County

CITY OF BRISTOL CORNWALL



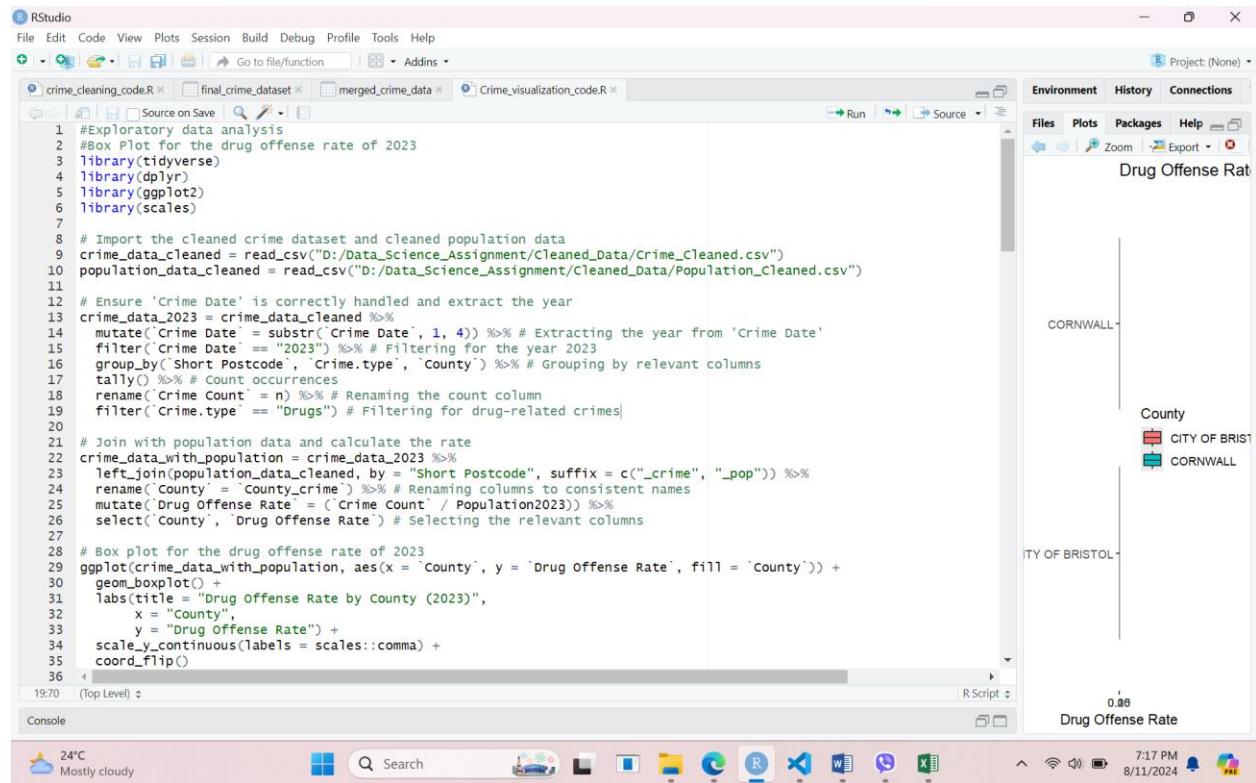


Crime Rate

The analysis first focused on drug offenses in 2023, using a box plot to compare drug offense rates by county, revealing variations and outliers. A radar chart was then used to analyze vehicle crime data from 2020 to 2023, showing different crime types over time. A pie chart illustrated robbery rates by county for a specific month in 2023. Finally, a line chart was plotted to track trends in drug crime rates over time.

Figure 9

Crime Visualization



```
40 # Vehicle crime rate from 2020 to 2023 (Radar chart)
41 library(dplyr)
42 library(tidyr)
43 library(fmsb)
44
45 crime_data_cleaned = read.csv("D:/Data_Science_Assignment/Cleaned_Data/Crime_Cleaned.csv")
46 vehicle_crime_types = c("Bicycle theft", "Vehicle crime", "Criminal damage and arson")
47
48 # Filtering and summarizing data for the years 2020 to 2023
49 crime_counts = crime_data_cleaned %>%
50   filter(Crime.type %in% vehicle_crime_types) %>%
51   mutate(year = substr(`Crime.Date`, 1, 4)) %>%
52   filter(year >= "2020" & year <= "2023") %>%
53   group_by(year, Crime.type) %>%
54   summarise(count = n(), .groups = 'drop') %>%
55   pivot_wider(names_from = Crime.type, values_from = count, values_fill = list(count = 0))
56
57 # Prepare data for radar chart
58 radar_data = as.data.frame(crime_counts)
59 row.names(radar_data) = radar_data$year
60 radar_data = radar_data %>% select(-Year)
61
62 # Ensure at least three columns (crime types)
63 if (ncol(radar_data) < 3) {
64   stop("Not enough variables for radar chart. Ensure you have at least three crime types.")
65 }
66
67 max_values = rep(max(radar_data, na.rm = TRUE), ncol(radar_data))
68 min_values = rep(0, ncol(radar_data))
69 radar_data = rbind(max_values, min_values, radar_data)
70
71 # Plot the radar chart
72 radarchart(radar_data, axistype = 1,
73             pcol = rgb(0.2, 0.5, 0.5, 0.7), pfcol = rgb(0.2, 0.5, 0.5, 0.5),
74             scol = "#0072BD", scol2 = "#0072BD", scol3 = "#0072BD", scol4 = "#0072BD",
75             max_radar_data = 100, max_radar_label = 100, max_radar_label2 = 100, max_radar_label3 = 100,
76             max_radar_label4 = 100)
77
```

(Top Level) ▾ R Script ▾

Console



```
Source on Save Run Source
71 # Plot the radar chart
72 radarchart(radar_data, axistype = 1,
73             pc1 = rgb(0.2, 0.5, 0.5, 0.7), pfcol = rgb(0.2, 0.5, 0.5, 0.5),
74             cg1col = "grey", cg1ty = 1, axislabcol = "grey", caxislabels = seq(0, max(radar_data[-c(1,2),]), na.rm = TRUE), length.out = 5),
75             title = "Vehicle Crime Rate (Theft, Accident) 2020 to 2023")
76
77
78 #Pie chart for a specific month of 2023 (robbery rate)
79 library(dplyr)
80 library(ggplot2)
81
82 # Read the cleaned crime and population data
83 crime_data_Cleaned = read.csv("D:/Data_Science_Assignment/Cleaned_Data/Crime_Cleaned.csv")
84 population_data = read.csv("D:/Data_Science_Assignment/cleaned_Data/Population_Cleaned.csv")
85
86 # The month and year to analyze
87 specific_month = "2023-12"
88
89 # Filtering the data for the specific month and year
90 monthly_data = crime_data_cleaned %>%
91   filter(substr(Crime.Date, 1, 7) == specific_month)
92
93 # Counting the number of robberies by county
94 robbery_count = monthly_data %>%
95   filter(Crime.type == "Robbery") %>%
96   group_by(County.Town.City) %>%
97   summarise(Count = n()) %>%
98   arrange(desc(Count))
99
100 # Joining with the population data
101 robbery_rate = robbery_count %>%
102   left_join(population_data, by = "County", "Town.City.y") %>%
103   mutate(Robbery_Rate = (Count / Population2023)) %>%
104   rename(`Town/City` = Town.City.y)
105
106 # Creating the pie chart for robbery rate
```

Source on Save | Run | Source | Run | Source

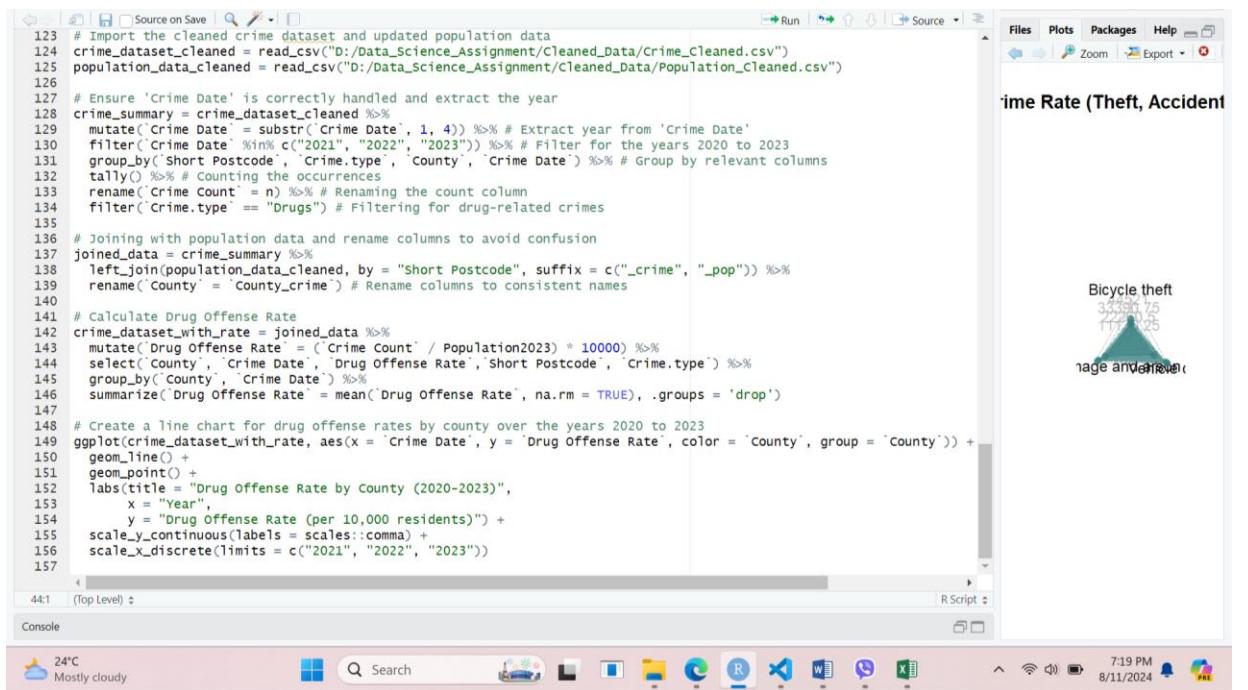
```

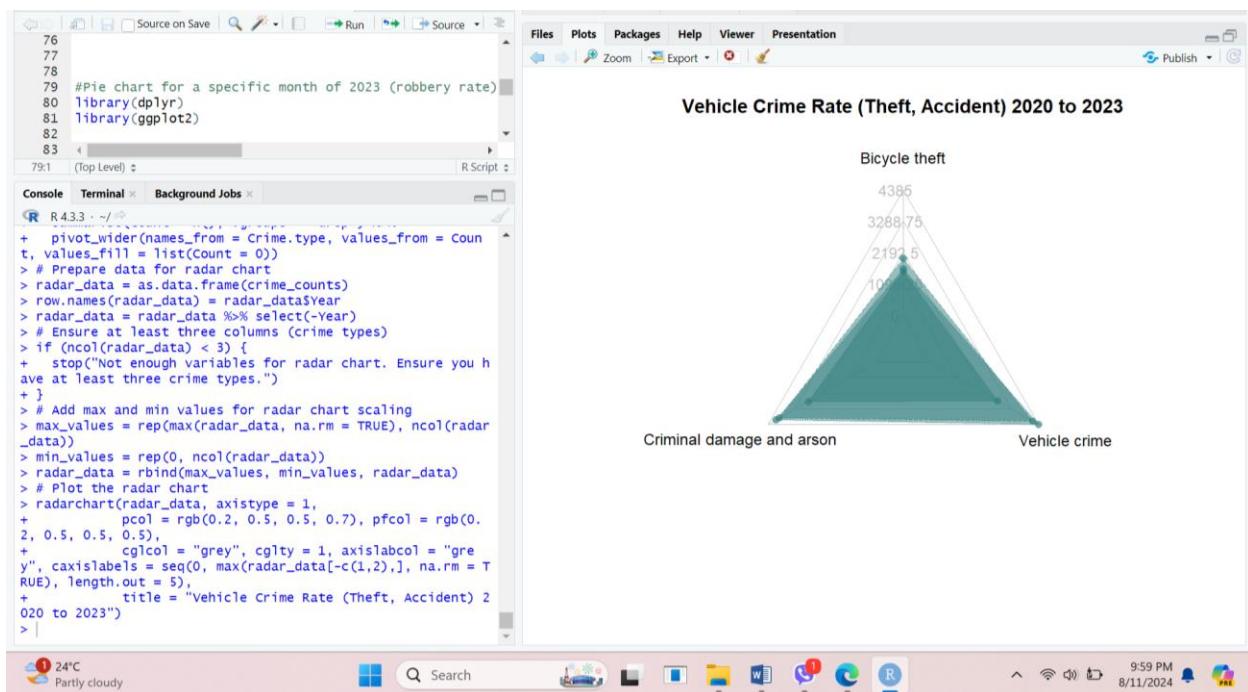
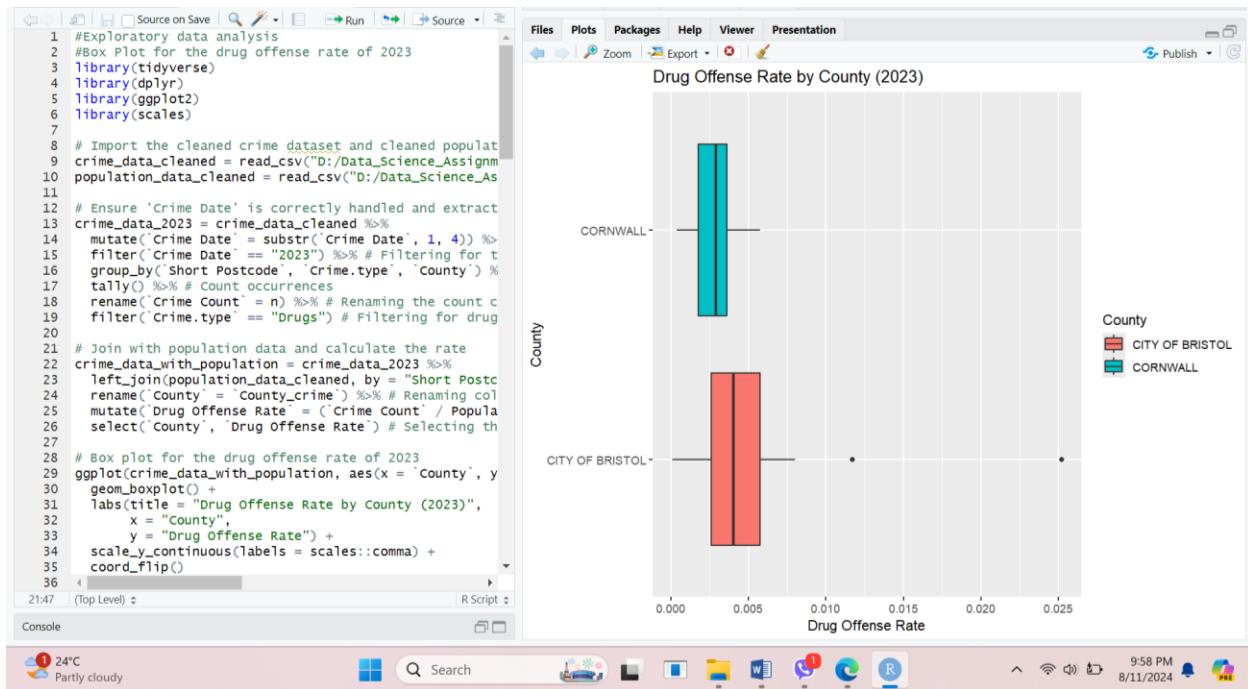
106 # Creating the pie chart for robbery rate
107 #in the month of 2023-12, the crime rate of Town/city in the county cornwall is very low.
108 #So, in pie chart, it is not clear.
109 ggplot(robbery_rate, aes(x = "", y = Robbery_Rate, fill = 'Town/City')) +
110   geom_bar(width = 1, stat = "identity") +
111   coord_polar("y") +
112   theme_void() +
113   labs(title = paste("Pie Chart of Robbery Rate by Town/City for 2023-DECEMBER"))
114
115
116
117
118
119 # Drug offense rate - Line chart for both county's per 10k people
120 library(tidyverse)
121 library(dplyr)
122 library(ggplot2)
123 library(scales)
124
125 # Import the cleaned crime dataset and updated population data
126 crime_dataset_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Crime_Cleaned.csv")
127 population_data_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Population_Cleaned.csv")
128
129 # Ensure 'Crime Date' is correctly handled and extract the year
130 crime_summary = crime_dataset_cleaned %>%
131   mutate(Crime Date = substr('Crime Date', 1, 4)) %>% # Extract year from 'Crime Date'
132   filter(Crime Date %in% c("2021", "2022", "2023")) %>% # Filter for the years 2020 to 2023
133   group_by(Short Postcode, 'Crime.type', 'County', 'Crime Date') %>% # Group by relevant columns
134   tally() %>% # Counting the occurrences
135   rename('Crime Count' = n) %>% # Renaming the count column
136   filter('Crime.type' == "Drugs") # Filtering for drug-related crimes
137
138 # Joining with population data and rename columns to avoid confusion
139 joined_data = crime_summary %>%
140   left_join(population_data_cleaned, by = "Short Postcode", suffix = c("_crime", "_pop")) %>%
141   rename('County' = 'County_crime') # Rename columns to consistent names
9038 (Top Level) : R Script

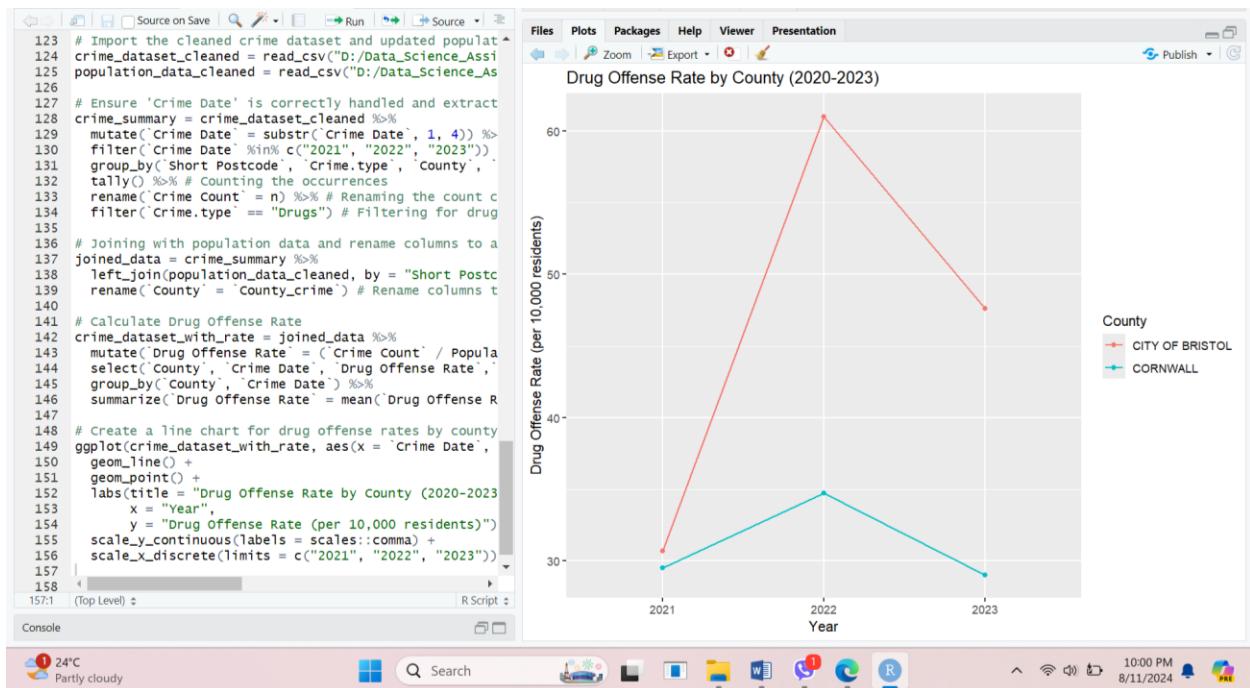
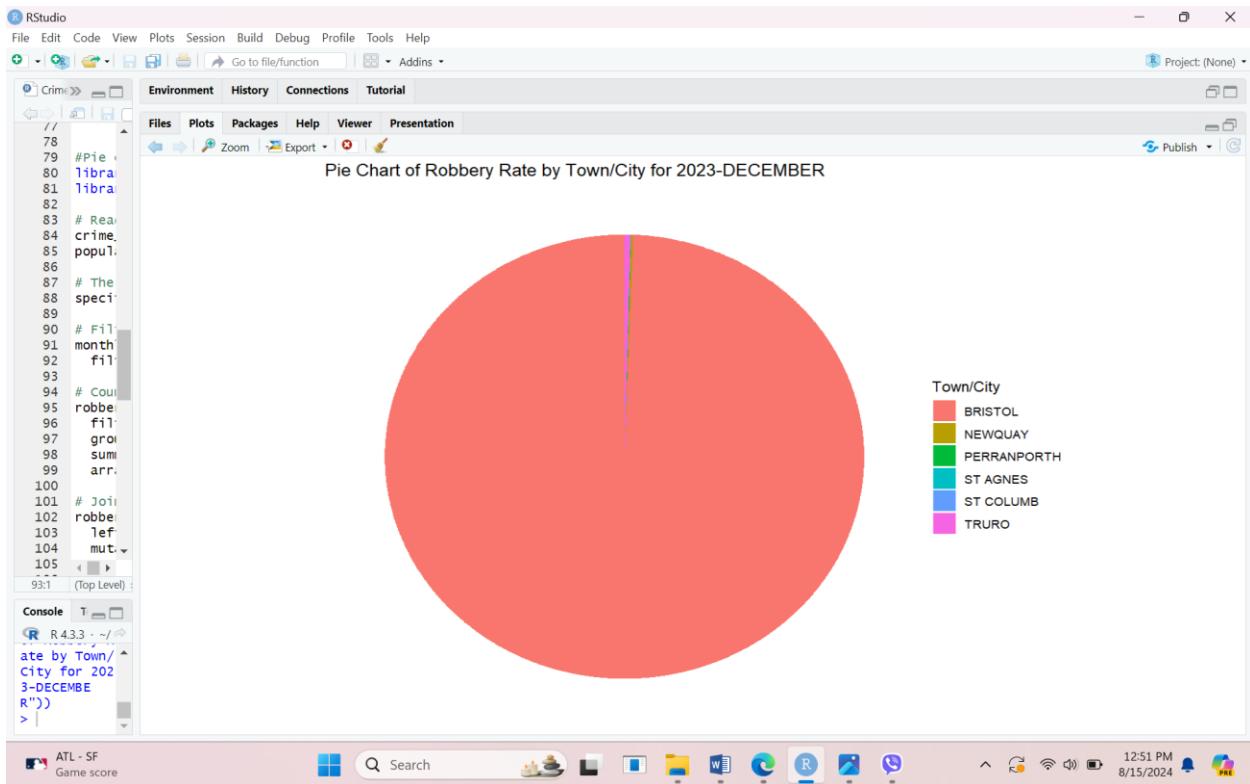
```

Console

30°C Mostly sunny Search Run Source 12:53 PM 8/15/2024





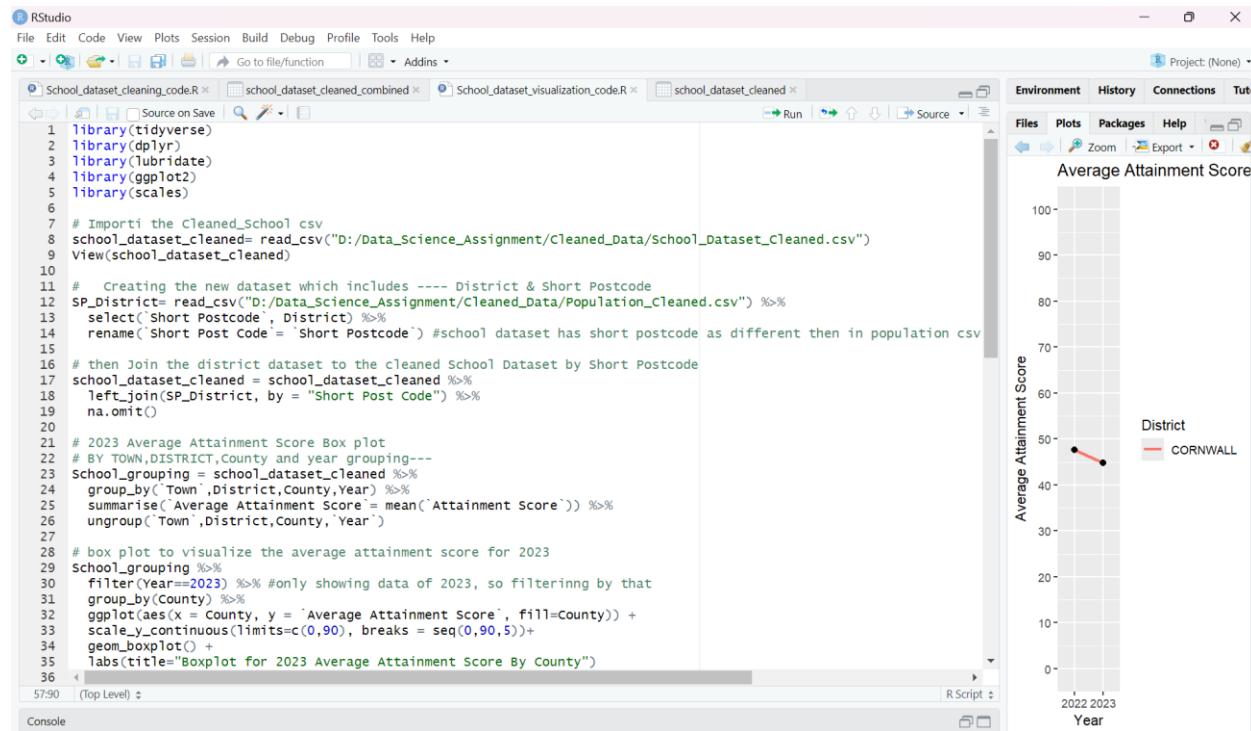


Schools

The cleaned school dataset was merged with a population dataset to include district information, with standardized column names and missing values removed. The analysis then visualized average attainment scores for 2023 using a box plot, grouped by Town, District, County, and Year. This revealed score distributions across counties. Additionally, line graphs were created to show trends in average attainment scores from 2022-2023 by district, highlighting performance changes over time.

Figure 10

School Visualization



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

School_dataset_cleaning_code.R school_dataset_cleaned_combined.R School_dataset_visualization_code.R school_dataset_cleaned.R

Run Source

```
38 # Average Attainment Score for 2022-2023 Line Graph For District --CITY OF BRISTOL
39 # Group cleaned school csv
40 school_grouping_next <- school_dataset_cleaned %>%
41   filter(County=="CITY OF BRISTOL") %>% #filter to show only rows with county as city of bristol
42   group_by(District,Year) %>%
43   summarise(Average Attainment Score` = mean(`Attainment Score`))
44
45 # Convert Year to a factor
46 school_grouping_next = school_grouping_next %>%
47   mutate(Year = as.factor(Year))
48
49 # Line graph of average Attainment score - 2022-2023
50 school_grouping_next %>%
51   group_by(District, Year) %>% # By District and Year -grouping in order to compare the average Attainment Score across districts over multiple years
52   ggplot(aes(x = `Year` , y = `Average Attainment Score` , group = District, color = District)) +
53     geom_line(linewidth = 1) +
54     geom_point(size = 2, color = "black") +
55     scale_y_continuous(limits=c(0,100), breaks = seq(0,100,10)) +
56     labs(title = "Average Attainment Score Line Graph For CITY OF BRISTOL - District - 2022-2023",
57       x = "Year",
58       y = "Average Attainment Score")
59
60 # Average Attainment Score for 2022-2023 Line Graph For District -- CORNWALL
61 # Group cleaned school csv
62 school_grouping_next2 = school_dataset_cleaned %>%
63   filter(County=="CORNWALL") %>% #filter to show only rows with county as cornwall
64   group_by(District,Year) %>%
65   summarise(Average Attainment Score` = mean(`Attainment Score`))
66
67 # Convert Year to a factor
68 school_grouping_next2 = school_grouping_next2 %>%
69   mutate(Year = as.factor(Year))
70
71 # Line graph of average Attainment score - 2022-2023
72
73 (Top Level) ▾
```

RStudio

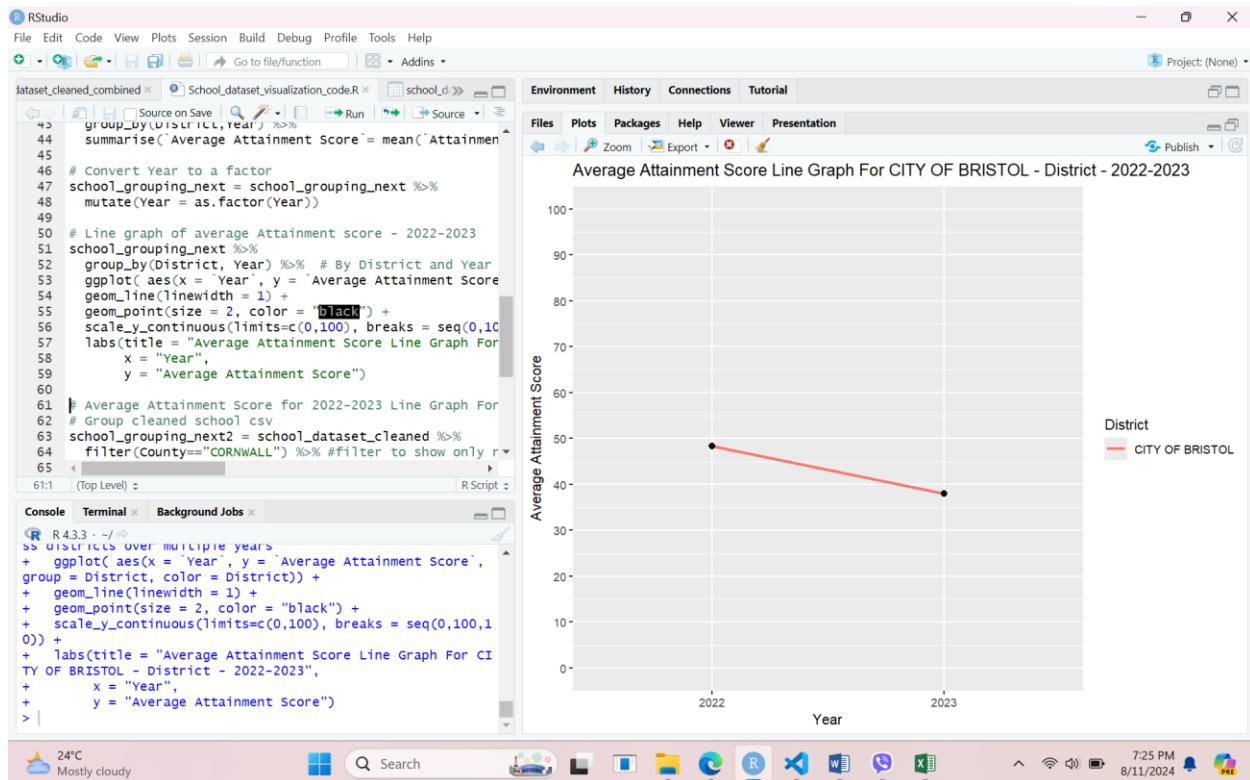
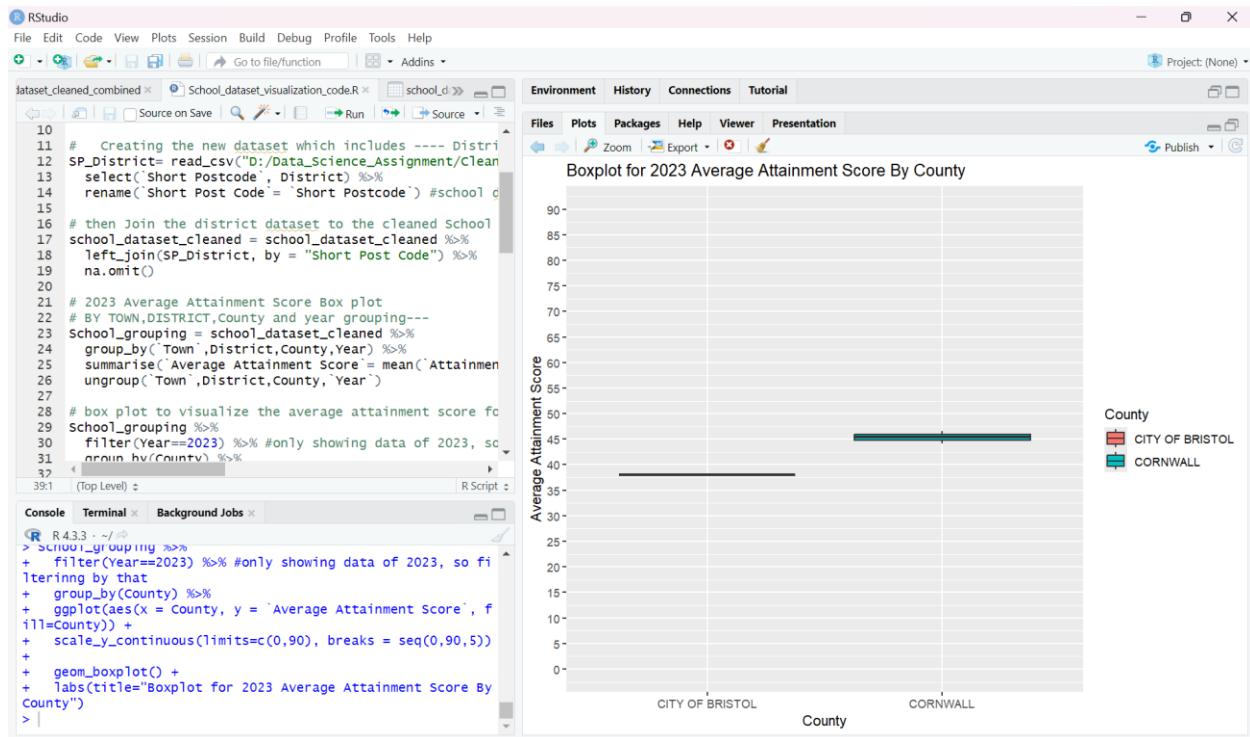
File Edit Code View Plots Session Build Debug Profile Tools Help

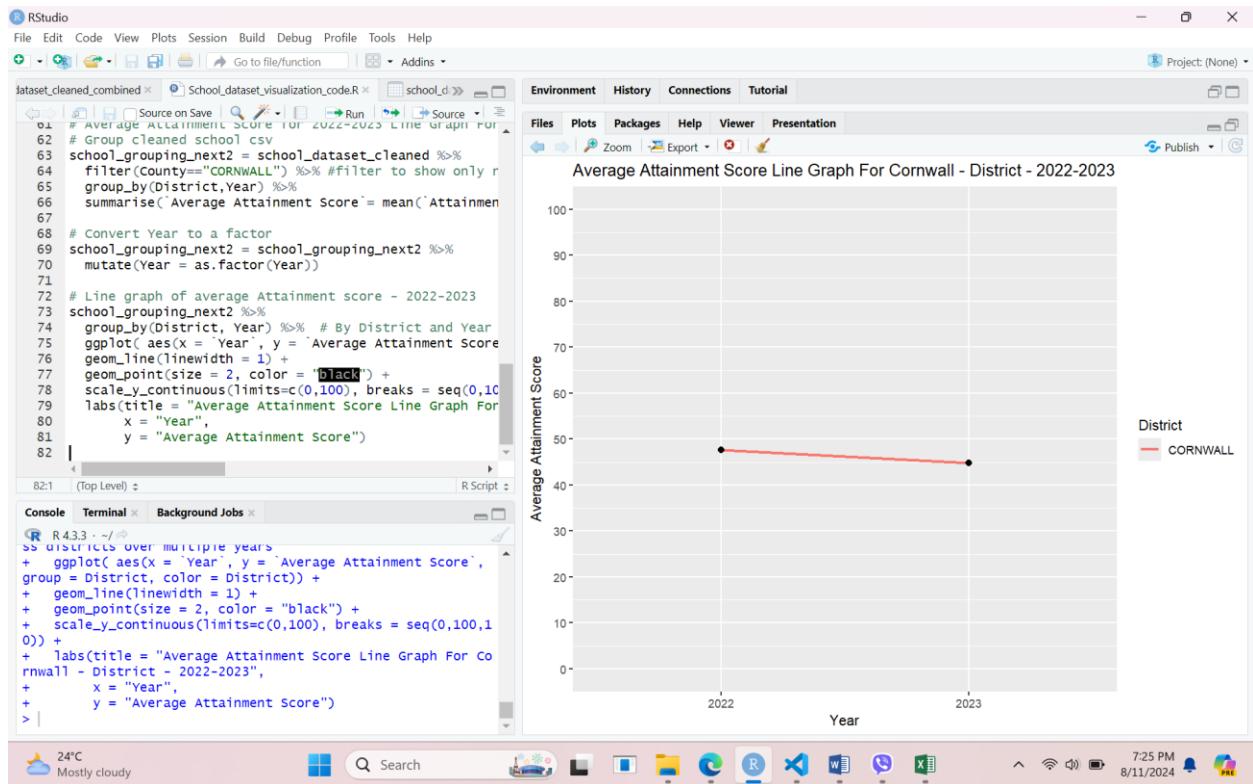
Go to file/function Addins

School_dataset_cleaning_code.R school_dataset_cleaned_combined School_dataset_visualization_code.R school_dataset_cleaned

Run Source Envir

```
48 mutate(Year = as.factor(year))
49
50 # Line graph of average Attainment score - 2022-2023
51 school_grouping_next %>%
52   group_by(District, Year) %% # By District and Year -grouping in order to compare the average Attainment Score across districts over multiple
53   ggplot(aes(x = `Year` , y = `Average Attainment Score` , group = District, color = District)) +
54   geom_line(linewidth = 1) +
55   geom_point(size = 2, color = 'black') +
56   scale_y_continuous(limits=c(0,100), breaks = seq(0,100,10)) +
57   labs(title = "Average Attainment Score Line Graph For CITY OF BRISTOL - District - 2022-2023",
58       x = "Year",
59       y = "Average Attainment Score")
60
61 # Average Attainment Score for 2022-2023 Line Graph For District -- CORNWALL
62 # Group cleaned school csv
63 school_grouping_next2 = school_dataset_cleaned %>%
64   filter(County=="CORNWALL") %% #filter to show only rows with county as cornwall
65   group_by(District,year) %%%
66   summarise(`Average Attainment Score` = mean(`Attainment Score`))
67
68 # Convert Year to a factor
69 school_grouping_next2 = school_grouping_next2 %>%
70   mutate(Year = as.factor(year))
71
72 # Line graph of average Attainment score - 2022-2023
73 school_grouping_next2 %>%
74   group_by(District, Year) %% # By District and Year -grouping in order to compare the average Attainment Score across districts over multiple
75   ggplot(aes(x = `Year` , y = `Average Attainment Score` , group = District, color = District)) +
76   geom_line(linewidth = 1) +
77   geom_point(size = 2, color = 'black') +
78   scale_y_continuous(limits=c(0,100), breaks = seq(0,100,10)) +
79   labs(title = "Average Attainment Score Line Graph For Cornwall - District - 2022-2023",
80       x = "Year",
81       y = "Average Attainment Score")
```





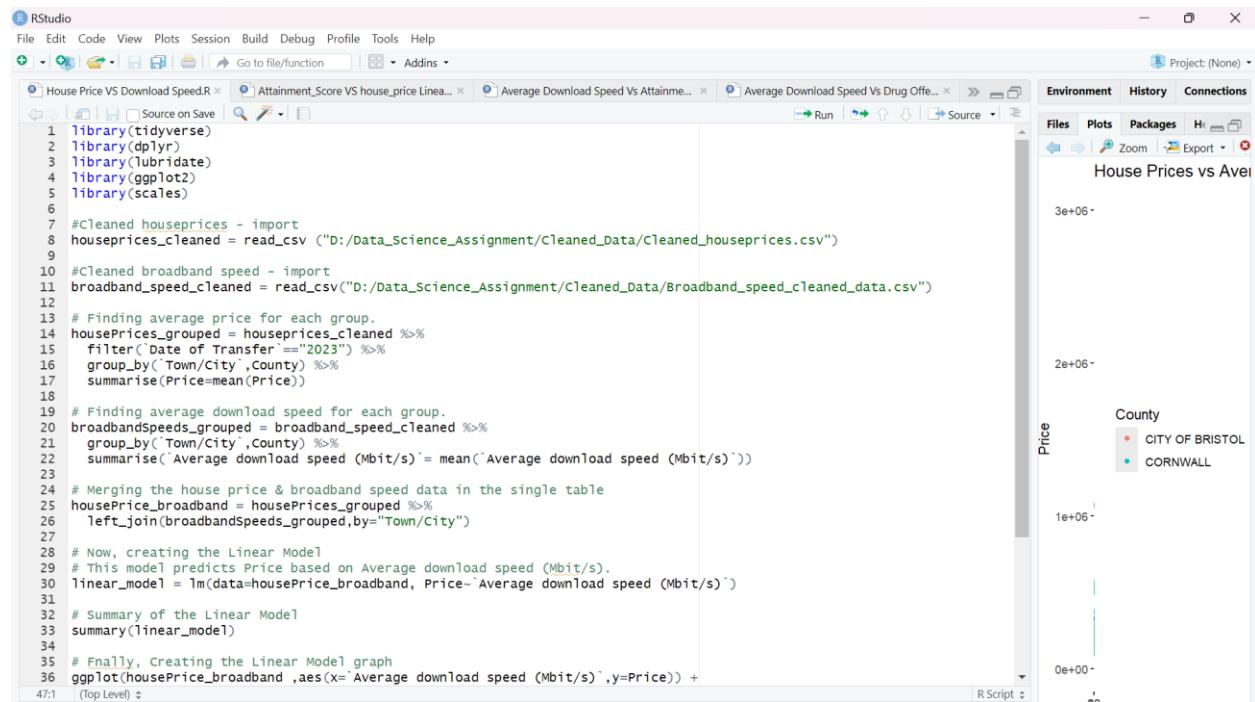
Linear Modeling

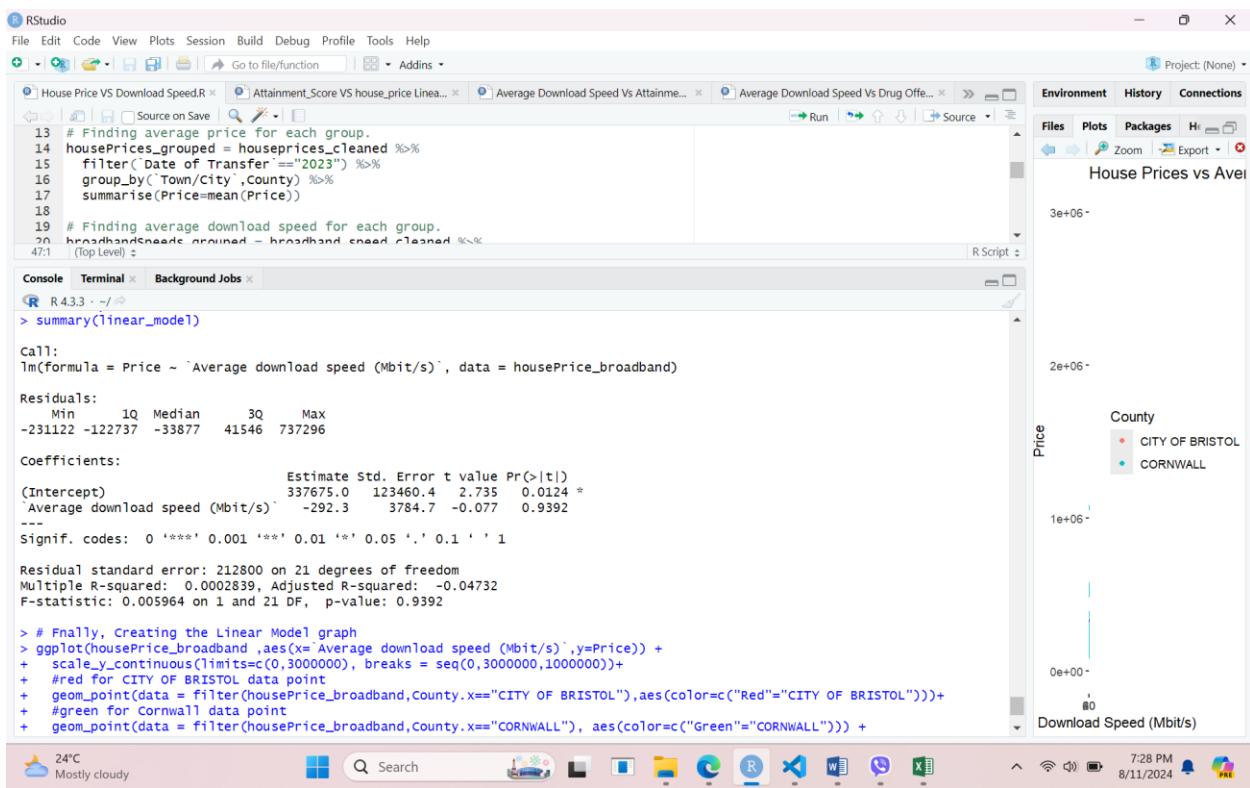
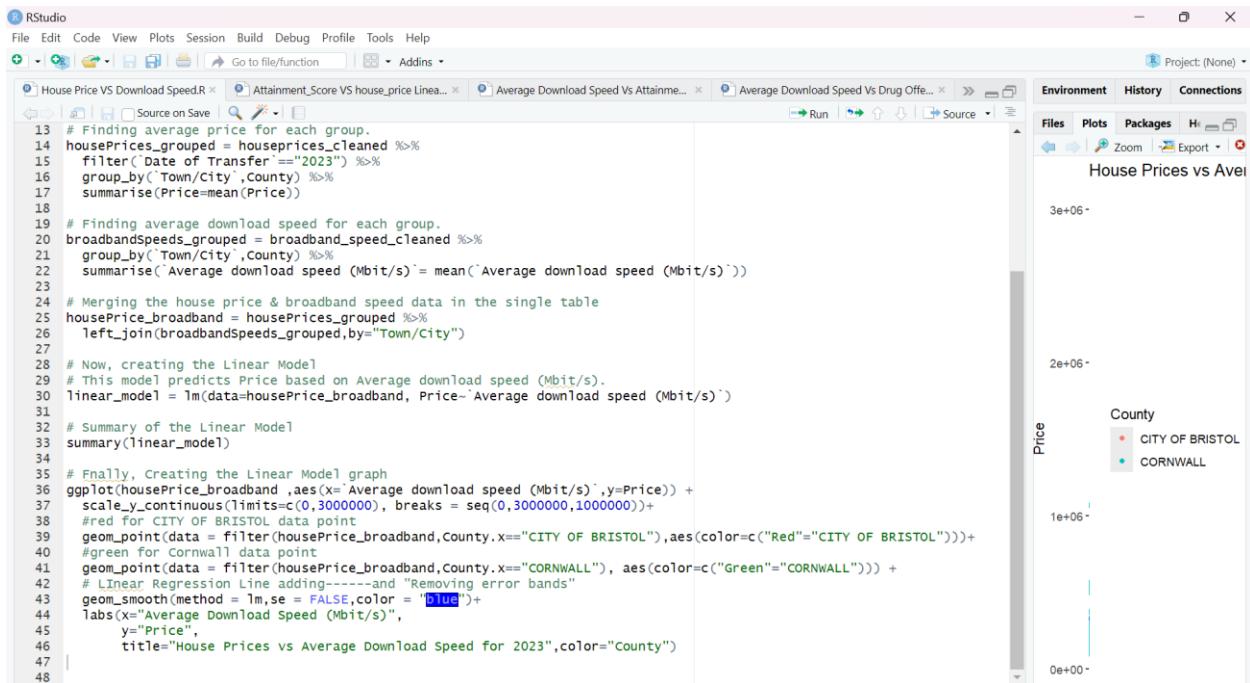
1. House prices vs Download Speed

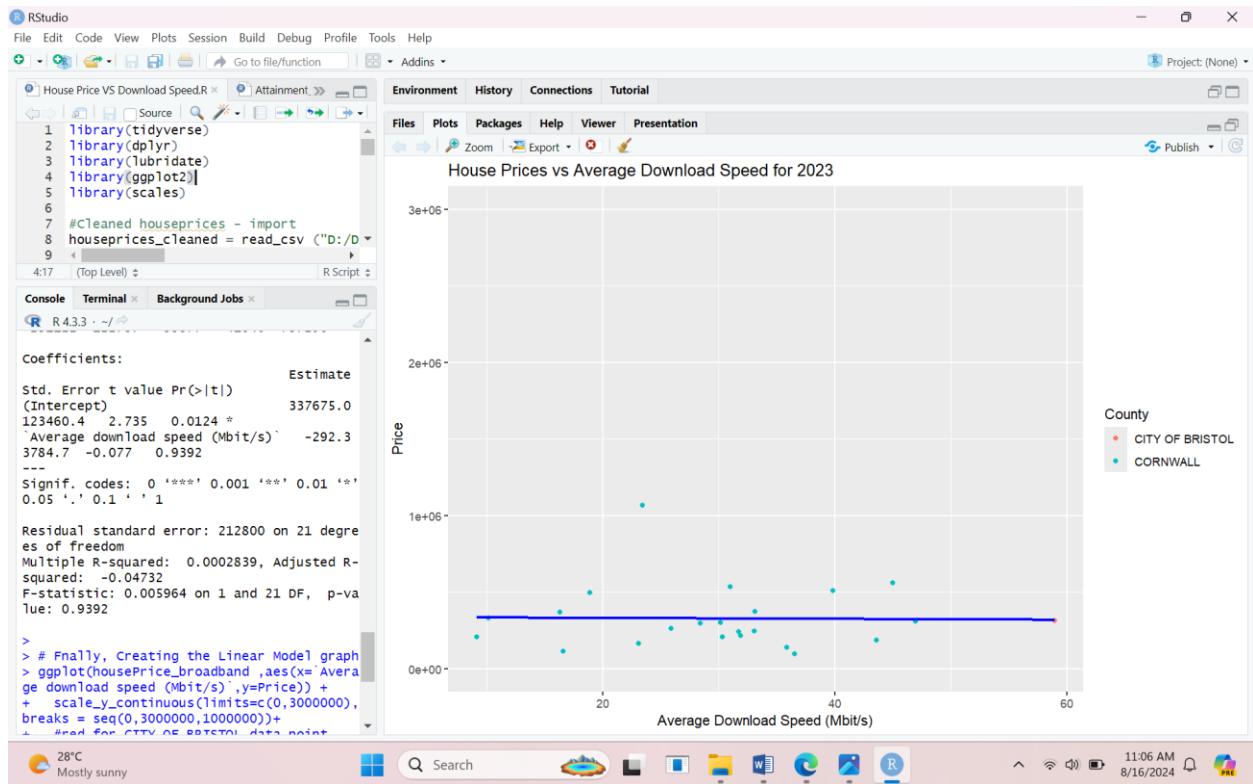
The analysis shows no statistically significant relationship between average download speed and house prices in the dataset. The p-value for the average download speed is very high (0.9392), and the R-squared value is near zero, indicating that average download speed does not explain the variation in house prices. The flat trend line in the plot further supports the conclusion that there is no strong relationship between house prices and average download speed in the data provided.

Figure 11

House prices vs Download Speed





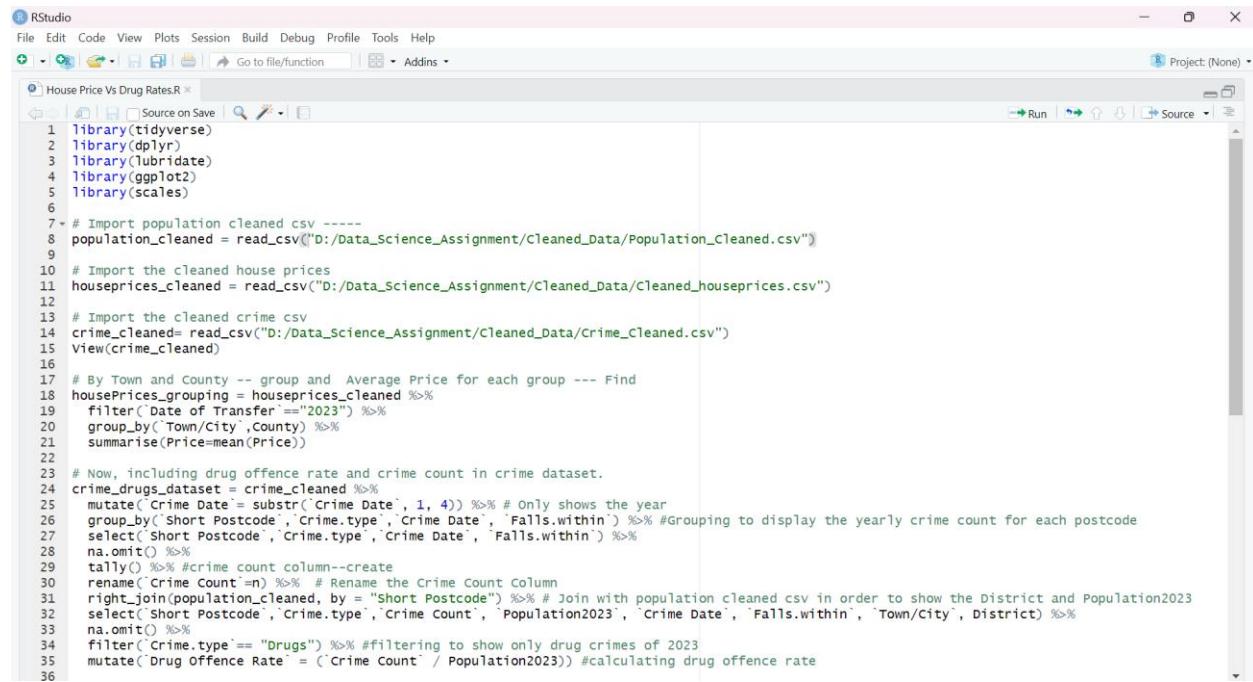


2. House prices vs drug rates

The linear model graph displays the correlation between drug offense rates and home prices in Cornwall and the City of Bristol, two UK counties. According to the graph, there is a very weak positive link between drug offense rates and home prices, which means that when drug offense rates rise, house values typically rise as well. But the correlation is so weak that it is essentially insignificant.

Figure 12

House prices vs Drug Offence Rates



The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project:** Project: (None)
- Code Editor:** The code is named "House Price Vs Drug Rates.R". It contains R code for data cleaning and analysis, including imports for tidyverse, dplyr, lubridate, ggplot2, and scales. It reads CSV files for population, house prices, and crime data, filters for 2023, groups by town and county, calculates average house prices, and joins the crime dataset to calculate drug offence rates. The code uses %>% from the tidyverse package.
- Tools Bar:** Includes Run, Source, and other standard RStudio icons.

```

library(tidyverse)
library(dplyr)
library(lubridate)
library(ggplot2)
library(scales)

# Import population cleaned csv ----
population_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Population_Cleaned.csv")

# Import the cleaned house prices
houseprices_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Cleaned_houseprices.csv")

# Import the cleaned crime csv
crime_cleaned= read_csv("D:/Data_Science_Assignment/Cleaned_Data/Crime_Cleaned.csv")
View(crime_cleaned)

# By Town and County -- group and Average Price for each group --- Find
housePrices_grouping = houseprices_cleaned %>%
  filter(`Date of Transfer`=="2023") %>%
  group_by(`Town/City`, `County`) %>%
  summarise(Price=mean(Price))

# Now, including drug offence rate and crime count in crime dataset.
crime_drugs_dataset = crime_cleaned %>%
  mutate(`Crime Date` = substr(`Crime Date`, 1, 4)) %>% # Only shows the year
  group_by(`Short Postcode`, `Crime.type`, `Crime Date`, `Falls.within`) %>% #Grouping to display the yearly crime count for each postcode
  select(`Short Postcode`, `Crime.type`, `Crime Date`, `Falls.within`) %>%
  na.omit() %>%
  tally() %>% #Crime count column--create
  rename(`Crime Count`=n) %>% # Rename the Crime Count Column
  right_join(population_cleaned, by = "Short Postcode") %>% # Join with population cleaned csv in order to show the District and Population2023
  select(`Short Postcode`, `Crime.type`, `Crime Count`, `Population2023`, `Crime Date`, `Falls.within`, `Town/City`, `District`) %>%
  na.omit() %>%
  filter(`Crime.type`== "Drugs") %>% #filtering to show only drug crimes of 2023
  mutate(`Drug Offence Rate` = (`Crime Count` / Population2023)) #calculating drug offence rate

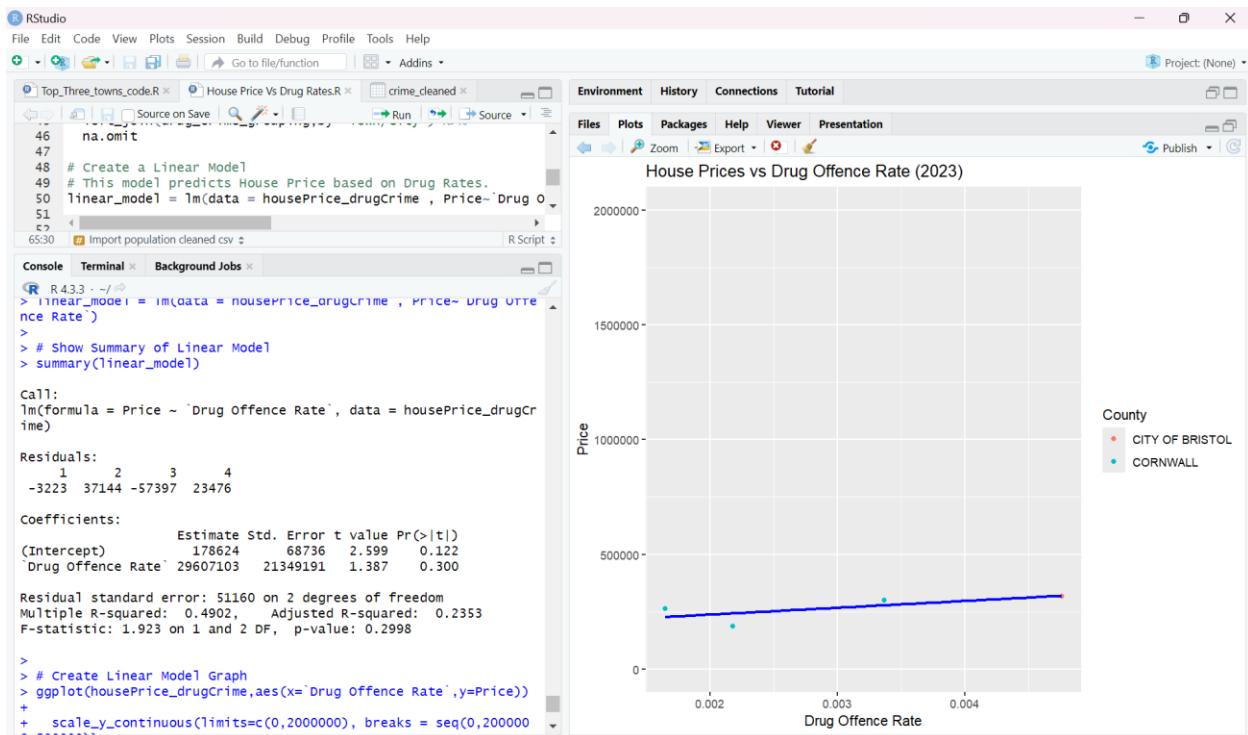
```

RStudio

```

File Edit Code View Plots Session Build Debug Profile Tools Help
+ - Go to file/function Addins Project: (None)
House Price Vs Drug Rates.R
Source on Save Run Source
34 filter(Crime.type == "Drugs") %>% #filtering to show only drug crimes of 2023
35 mutate(Drug Offence Rate = (Crime Count / Population2023)) #calculating drug offence rate
36
37 # By county and town grouping the crime and drug csv & showing the rate for each group for the year 2023
38 drug_crime_grouping = crime_drugs_dataset %>%
39 filter(Crime Date=="2023") %>%
40 group_by(Falls.within, Town/City) %>%
41 summarise(Drug Offence Rate = mean(Drug Offence Rate))
42
43 # House price data and drug crime rate data ----JOINING---in a single table
44 housePrice_drugCrime = housePrices_grouping %>%
45 left_join(drug_crime_grouping, by="Town/City") %>%
46 na.omit
47
48 # Create a Linear Model
49 # This model predicts House Price based on Drug Rates.
50 linear_model = lm(data = housePrice_drugCrime , Price~Drug Offence Rate)
51
52 # Show Summary of Linear Model
53 summary(linear_model)
54
55 # Create Linear Model Graph
56 ggplot(housePrice_drugCrime,aes(x=Drug Offence Rate,y=Price)) +
57 scale_y_continuous(limits=c(0,2000000), breaks = seq(0,2000000,500000))+ # Format the y-axis labels
58 scale_x_continuous(labels = label_number(scale = 1, big.mark = ",")) + # Format the x-axis labels
59 # Red as City of Bristol data point.
60 geom_point(data = filter(housePrice_drugCrime,County=="CITY OF BRISTOL"),aes(color=c("Red"="CITY OF BRISTOL")))+ # Green for Cornwall data point.
61 geom_point(data = filter(housePrice_drugCrime,County=="CORNWALL"), aes(color=c("Green"="CORNWALL")))+ # Linear Regression Line adding----and "Removing error bands"
62 geom_smooth(method = lm , se = FALSE , color = "Blue")+
63 labs(x="Drug Offence Rate",
64 y="Price",
65 title="House Prices vs Drug Offence Rate (2023)",color="County")
66
67
68

```



3. Attainment 8 Score vs house price

Most data points are clustered around lower house prices (below 500,000) and attainment scores between 30 and 50. The trend line is nearly flat, suggesting that there is little to no significant correlation between house prices and attainment scores in the data provided. This means that higher house prices do not necessarily correlate with higher or lower attainment scores.

Figure 13

Attainment 8 Score vs House Price



The screenshot shows an RStudio interface with the following details:

- Code Editor:** Displays R code for data cleaning and combining two datasets. The code includes imports for tidyverse, dplyr, lubridate, ggplot2, and scales. It reads CSV files for cleaned house prices and school dataset, filters for 2023 transfers, converts column names to lowercase, and groups by town and county to calculate average prices and attainment scores.
- File Explorer:** Shows a file named "Attainment Score.R" in the sidebar.
- Variables View:** A legend titled "County" indicates two categories: "CITY OF E" (red dot) and "CORNWAI" (green dot).

```

1 #Attainment 8 Score vs house price (both counties combined)
2 library(tidyverse)
3 library(dplyr)
4 library(lubridate)
5 library(ggplot2)
6 library(scales)
7
8 # Import Cleaned House Prices
9 houseprices_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Cleaned_houseprices.csv")
10
11 # Import Cleaned School csv
12 school_dataset_cleaned= read_csv("D:/Data_Science_Assignment/Cleaned_Data/School_Dataset_Cleaned.csv")
13
14 # By Town and County grouping the House Prices & Average Price --- Finding for|each group
15 house_prices_grouping = houseprices_cleaned %>%
16   filter(`Date of Transfer`=="2023") %>%
17   group_by(`Town/City`, `County`) %>%
18   # From uppercase to all lowercase to the Town column
19   mutate(`Town/City` = tolower(`Town/City`)) %>%
20   summarise(`Price`=mean(Price))
21
22 #grouping school data by town and county and finding average score for each group
23 school_dataset_grouped = school_dataset_cleaned %>%
24   filter(`Year` == "2023") %>%
25   group_by(`Town` ,`County`) %>%
26   # Convert town to all lowercase
27   mutate(`Town`= tolower(Town)) %>%
28   summarise(`Attainment Score`=mean(`Attainment Score`))
29
30 # School and House Price Combining - in a single table
31 school_houseprice = school_dataset_grouped %>%
32   left_join(house_prices_grouping,by=c(`Town`="Town/City")) %>%
33   na.omit
34
35 # Create a Linear Model
36 # This model predicts Average attainment score based on Average House Prices

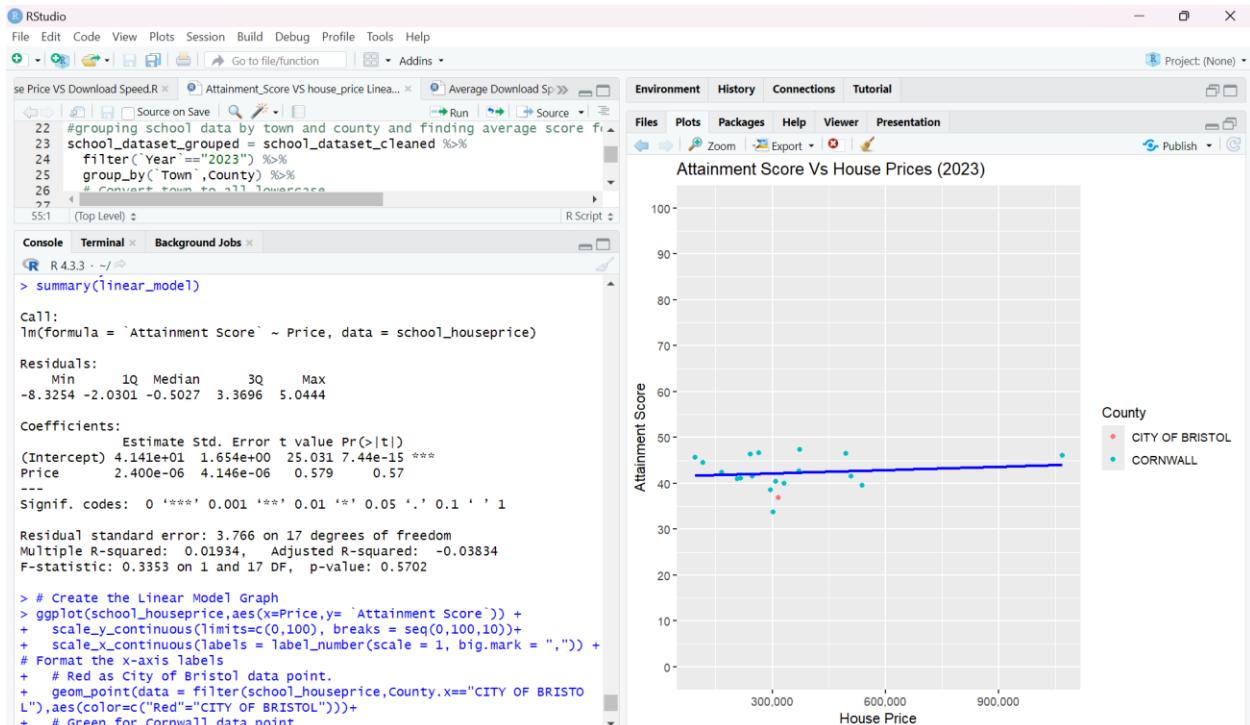
```

RStudio

```

22 #grouping school data by town and county and finding average score for each group
23 school_dataset_grouped = school_dataset_cleaned %>%
24   filter(`Year` == "2023") %>%
25   group_by(`Town`, `County`) %>%
26   # Convert town to all lowercase
27   mutate(Town= tolower(Town)) %>%
28   summarise(`Attainment Score` = mean(`Attainment Score`))
29
30 # School and House Price Combining - in a single table
31 school_houseprice = school_dataset_grouped %>%
32   left_join(house_prices_grouping, by=c("Town"="Town/City")) %>%
33   na.omit
34
35 # Create a Linear Model
36 # This model predicts Average attainment score based on Average House Prices
37 linear_model = lm(data = school_houseprice, `Attainment Score` ~ Price)
38
39 # Summary of the Linear Model
40 summary(linear_model)
41
42 # Create the Linear Model Graph
43 ggplot(school_houseprice,aes(x=Price,y= `Attainment Score`)) +
44   scale_y_continuous(limits=c(0,100), breaks = seq(0,100,10))+
45   scale_x_continuous(labels = label_number(scale = 1, big.mark = ",")) + # Format the x-axis labels
46   # Red as City of Bristol data point.
47   geom_point(data = filter(school_houseprice,County.x=="CITY OF BRISTOL"),aes(color=c("Red"="CITY OF BRISTOL")))+ # Green for Cornwall data point.
48   geom_point(data = filter(school_houseprice,County.x=="CORNWALL"), aes(color=c("Green"="CORNWALL")))+ # Linear Regression Line adding-----and "Removing error bands"
49   geom_smooth(method = lm , se = FALSE,color = "blue")+
50   labs(x="House Price",
51       y="Attainment Score",
52       title="Attainment Score Vs House Prices (2023)",color="County")
53
54
55

```

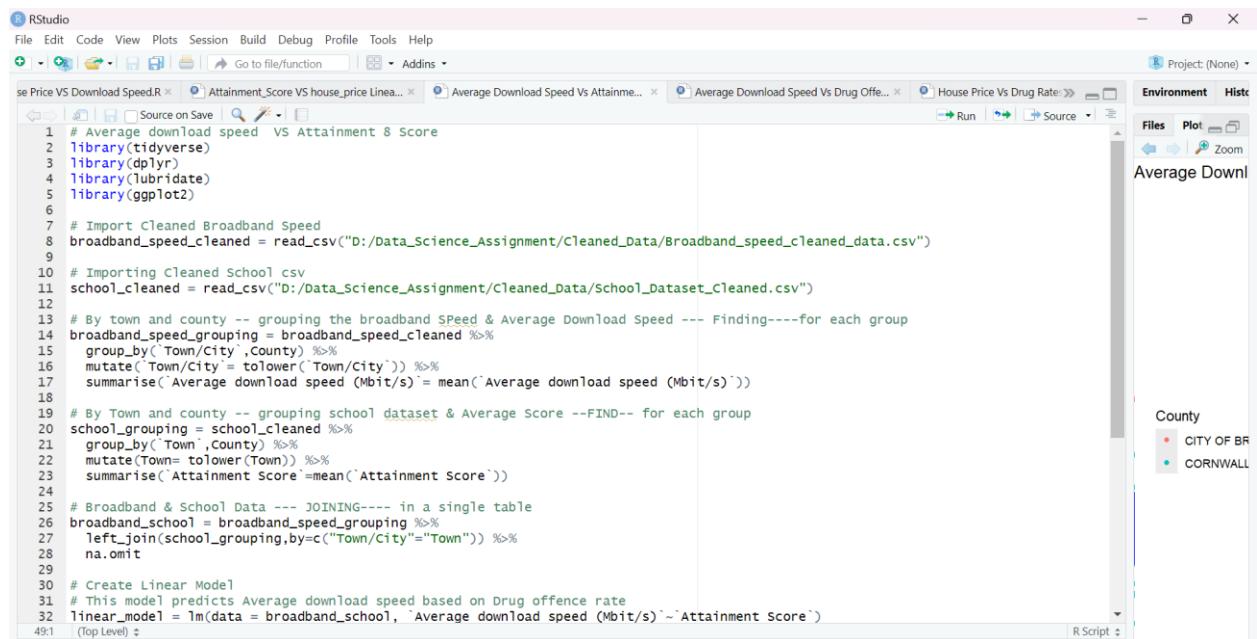


4. Average download speed vs Attainment 8 Score

City of Bristol has a relatively low attainment score but the highest average download speed among all data points. Cornwall shows a wider spread of attainment scores, mostly around 40-45, but the download speeds are consistently lower than the City of Bristol. The slight negative slope of the trend line suggests that, on average, regions with higher attainment scores tend to have lower average download speeds.

Figure 14

Average download speed vs Attainment 8 Score



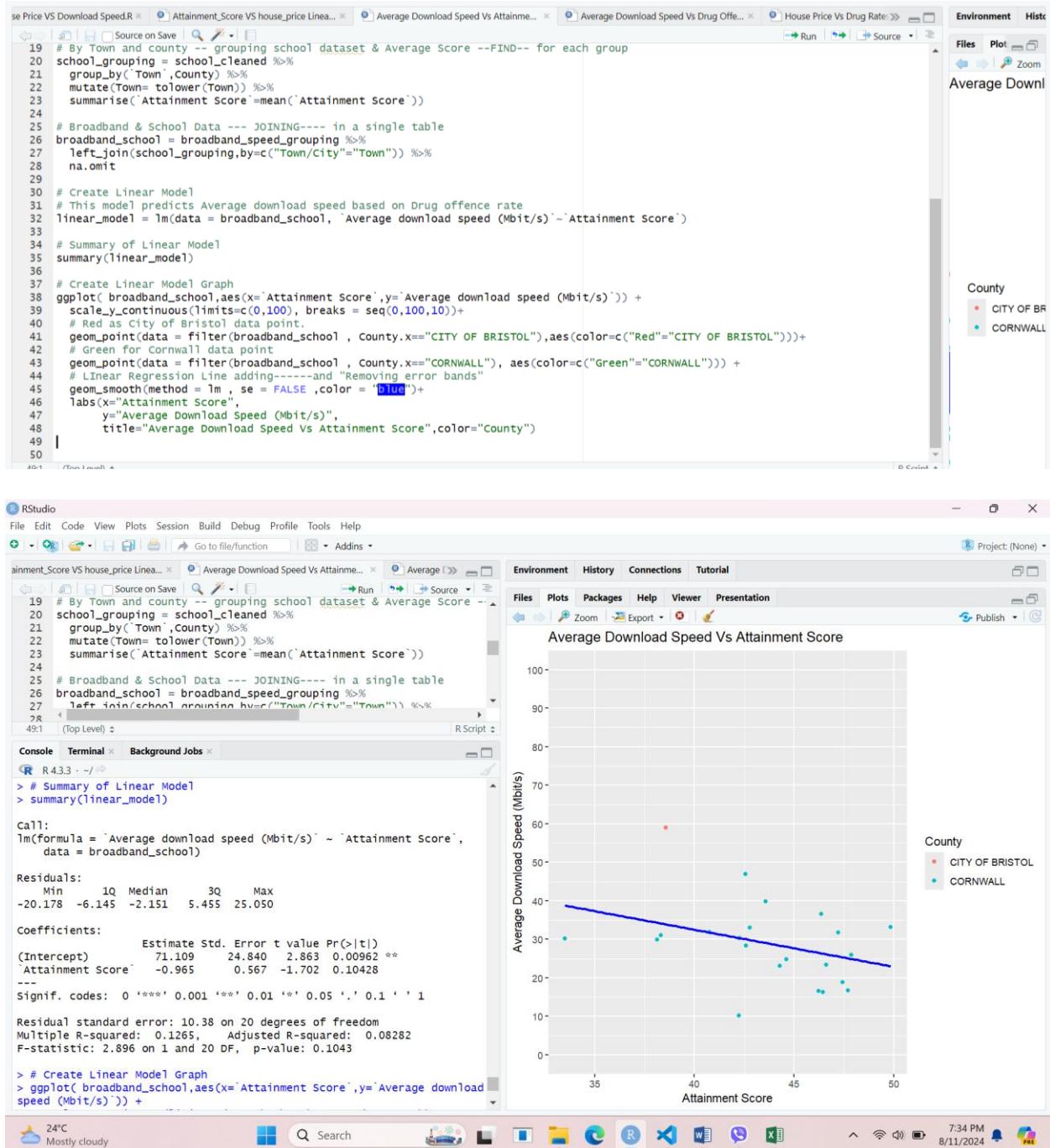
The screenshot shows the RStudio interface with an R script open. The script is titled "Average download speed VS Attainment 8 Score" and contains the following code:

```

1 # Average download speed VS Attainment 8 Score
2 library(tidyverse)
3 library(dplyr)
4 library(lubridate)
5 library(ggplot2)
6
7 # Import Cleaned Broadband Speed
8 broadband_speed_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Broadband_speed_cleaned_data.csv")
9
10 # Importing Cleaned School.csv
11 school_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/School_Dataset_Cleaned.csv")
12
13 # By town and county -- grouping the broadband speed & Average Download Speed --- Finding----for each group
14 broadband_speed_grouping = broadband_speed_cleaned %>%
15   group_by(Town/City, County) %>%
16   mutate(Town=tolower(Town)) %>%
17   summarise(`Average download speed (Mbit/s)` = mean(`Average download speed (Mbit/s)`))
18
19 # By Town and county -- grouping school dataset & Average Score --FIND-- for each group
20 school_grouping = school_cleaned %>%
21   group_by(Town, County) %>%
22   mutate(Town=tolower(Town)) %>%
23   summarise(`Attainment Score` = mean(`Attainment Score`))
24
25 # Broadband & School Data --- JOINING---- in a single table
26 broadband_school = broadband_speed_grouping %>%
27   left_join(school_grouping, by=c("Town/City"="Town")) %>%
28   na.omit
29
30 # Create Linear Model
31 # This model predicts Average download speed based on Drug offence rate
32 linear_model = lm(data = broadband_school, `Average download speed (Mbit/s)` ~ `Attainment Score`)
49:1 (Top Level) :

```

The RStudio environment pane shows a legend for "County" with two entries: "CITY OF BR" (red dot) and "CORNWALL" (blue dot). The plot pane is currently empty, indicating no plot has been generated yet.

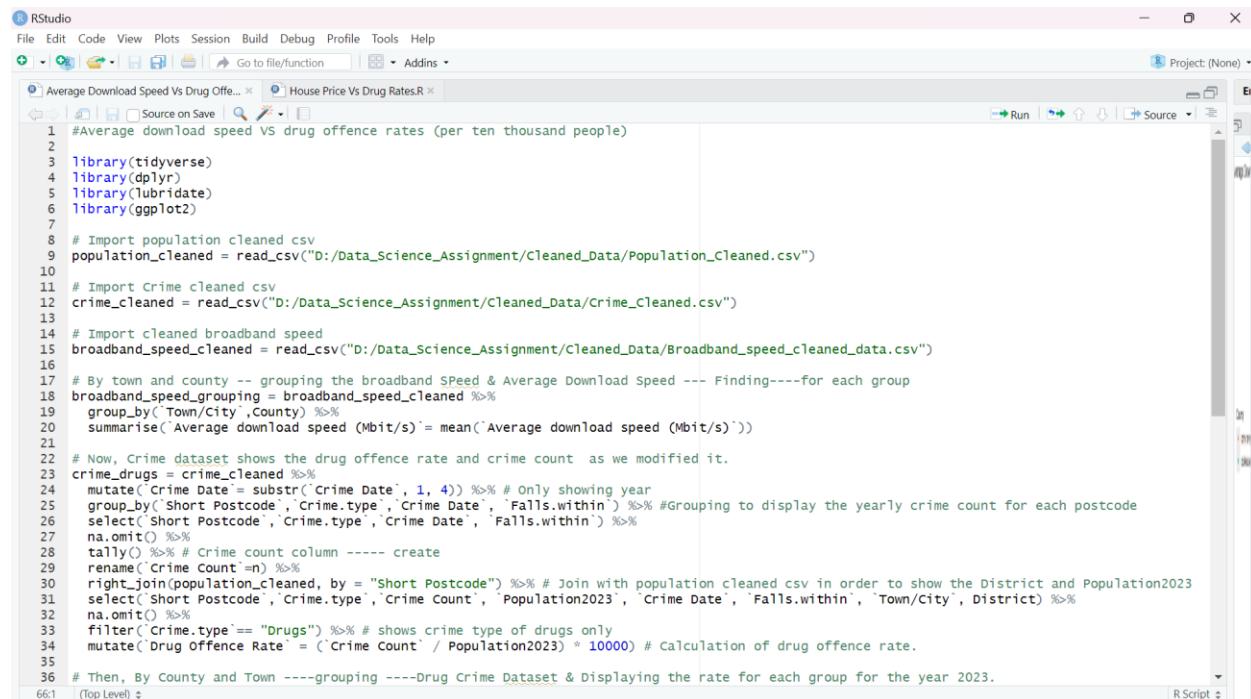


5. Average download speed vs drug offence rates (per ten thousand people)

City of Bristol has the highest drug offense rate and the highest average download speed among the two counties. Cornwall has lower drug offense rates and lower average download speeds in comparison. The positive slope of the trend line suggests a potential relationship where higher drug offense rates are associated with higher average download speeds.

Figure 15

Average download speed vs drug offence rates (per ten thousand people)



The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Standard RStudio toolbar with icons for file operations, search, and run.
- Project Bar:** Shows 'Project: (None)'.
- Code Editor:** Displays the R script 'Average Download Speed Vs Drug Offense Rates.R'.
- Output Panel:** Not visible in the screenshot.
- Environment Tab:** Shows the current environment variables.
- Help Tab:** Shows help documentation for 'mean'.
- File Tab:** Shows the file 'House Price Vs Drug Rates.R'.
- Run Tab:** Shows the status of the last run.
- Source Tab:** Shows the source code of the current file.
- Console Tab:** Shows the R console output.
- Plots Tab:** Shows the current plots.
- Session Tab:** Shows session information.
- Build Tab:** Shows build logs.
- Debug Tab:** Shows debug logs.
- Profile Tab:** Shows profile logs.
- Tools Tab:** Shows various tools and options.
- Help Tab:** Shows help documentation for 'mean'.
- File Tab:** Shows the file 'House Price Vs Drug Rates.R'.
- Run Tab:** Shows the status of the last run.
- Source Tab:** Shows the source code of the current file.

```

1 # Average download speed VS drug offence rates (per ten thousand people)
2
3 library(tidyverse)
4 library(dplyr)
5 library(lubridate)
6 library(ggplot2)
7
8 # Import population cleaned csv
9 population_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Population_Cleaned.csv")
10
11 # Import Crime cleaned csv
12 crime_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Crime_Cleaned.csv")
13
14 # Import cleaned broadband speed
15 broadband_speed_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Broadband_speed_cleaned_data.csv")
16
17 # By town and county -- grouping the broadband Speed & Average Download Speed --- Finding---for each group
18 broadband_speed_grouping = broadband_speed_cleaned %>%
19   group_by(Town/City ,County) %>%
20   summarise(`Average download speed (Mbit/s)` = mean(`Average download speed (Mbit/s)`))
21
22 # Now, Crime dataset shows the drug offence rate and crime count as we modified it.
23 crime_drugs = crime_cleaned %>%
24   mutate(Crime.Date = substr(`Crime Date`, 1, 4)) %>% # Only showing year
25   group_by(Short.Postcode ,`Crime.type` ,`Crime.Date` ,`Falls.within`) %>% #Grouping to display the yearly crime count for each postcode
26   select(Short.Postcode ,`Crime.type` ,`Crime.Date` ,`Falls.within` ) %>%
27   na.omit() %>%
28   tally() %>% # Crime count column ----- create
29   rename(`Crime Count` =n) %>%
30   right_join(population_cleaned, by = "Short.Postcode") %>% # Join with population cleaned csv in order to show the District and Population2023
31   select(`Short.Postcode` ,`Crime.type` ,`Crime.Count` ,`Population2023` ,`Crime.Date` ,`Falls.within` ,`Town/City` ,`District` ) %>%
32   na.omit() %>%
33   filter(`Crime.type` == "Drugs") %>% # shows crime type of drugs only
34   mutate(`Drug Offence Rate` = (`Crime.Count` / `Population2023` ) * 10000) # Calculation of drug offence rate.
35
36 # Then, By County and Town ---grouping ---Drug Crime Dataset & Displaying the rate for each group for the year 2023.
66:1 (Top Level) ▾

```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Source on Save Go to file/function Addins Project: (None)

```

34   mutate(`Drug Offence Rate` = `Crime Count` / Population2023) * 10000) # Calculation of drug offence rate.
35
36 # Then, By County and Town ---grouping ----Drug Crime Dataset & Displaying the rate for each group for the year 2023.
37 drug_crime_grouping = crime_crimes %>%
38   filter(`Crime Date` == "2023") %>%
39   group_by(`Falls.within`, `Town/City`) %>%
40   summarise(`Drug Offence Rate` = mean(`Drug Offence Rate`))
41
42 # Broadband and Drug crime rate data -- JOIN----in a single table
43 broadband_crime = broadband_speed_grouping %>%
44   left_join(drug_crime_grouping, by = "Town/City") %>%
45   na.omit
46
47 # Create a Linear Model
48 # This model predicts Average download speed based on Drug offence rate
49 linear_model1 = lm(data = broadband_crime, `Average download speed (Mbit/s)` ~ `Drug Offence Rate`)
50
51 # Summary of Linear Model
52 summary(linear_model1)
53
54 # Create Linear Model Graph
55 ggplot(broadband_crime, aes(x = `Drug Offence Rate`, y = `Average download speed (Mbit/s)`)) +
56   scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, 10)) +
57   # Red as City of Bristol data point.
58   geom_point(data = filter(broadband_crime, County == "CITY OF BRISTOL"), aes(color = c("Red" = "CITY OF BRISTOL"))) +
59   # Green for Cornwall data point
60   geom_point(data = filter(broadband_crime, County == "CORNWALL"), aes(color = c("Green" = "CORNWALL"))) +
61   # Linear Regression Line adding-----and "Removing error bands"
62   geom_smooth(method = lm, se = FALSE, color = "Blue") +
63   labs(x = "Drug Offence Rate",
64        y = "Average Download Speed (Mbit/s)",
65        title = "Average Download Speed Vs Drug Offence Rate (2023)", color = "County")
66
67

```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Source on Save Go to file/function Addins Project: (None)

Top_Three_towns_code.R Average Download Speed Vs Drug Offe... Environment History Connections Tutorial

Files Plots Packages Help Viewer Presentation

Console Terminal Background Jobs

R 4.3.3 · ~/

```

63   labs(x = "Drug Offence Rate",
64        y = "Average Download Speed (Mbit/s)",
65        title = "Average Download Speed Vs Drug Offence Rate")
66
67
68
69
66:1 (Top Level) R Script

```

Call:

```

lm(formula = `Average download speed (Mbit/s)` ~ `Drug Offence Rate`,
   data = broadband_crime)

```

Residuals:

| | 1 | 2 | 3 | 4 | 5 |
|--------|--------|---------|---------|----------|---|
| 4.1644 | 0.6078 | -6.6595 | 13.3250 | -11.4378 | |

Coefficients:

| | Estimate | Std. Error | t value | Pr(> t) |
|-------------------|----------|------------|---------|----------|
| (Intercept) | 9.6858 | 12.4389 | 0.779 | 0.493 |
| Drug Offence Rate | 0.9472 | 0.4177 | 2.268 | 0.108 |

Residual standard error: 11.11 on 3 degrees of freedom

Multiple R-squared: 0.6316, Adjusted R-squared: 0.5087

F-statistic: 5.142 on 1 and 3 DF, p-value: 0.1082

```

>
> # Create Linear Model Graph
> ggplot(broadband_crime, aes(x = `Drug Offence Rate`, y = `Average do
wnload speed (Mbit/s)`)) +
+   scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, 1
0)) +

```

Average Download Speed Vs Drug Offence Rate (2023)

County

- CITY OF BRISTOL
- CORNWALL

24°C Partly cloudy Search 9:56 PM 8/11/2024

Recommendation System

Overview

In this documentation, we have developed a comprehensive recommendation system to guide potential homebuyers and investors in Bristol and Cornwall. Our analysis integrates multiple key factors including house prices, broadband speeds, school grades, and crime rates to identify the most favorable towns and cities. By normalizing and ranking each factor—where lower house prices and lower crime rates receive higher scores, and higher broadband speeds and better school grades contribute positively—we have created a unified scoring system. This system enables us to objectively assess and compare various locations based on critical quality-of-life and investment criteria.

The final recommendations are based on the combined scores from our detailed analysis. We present the top ten recommended towns or cities in Bristol and Cornwall, each selected for their exceptional attributes, to assist you in making an informed decision for your future home and investment opportunities.

Results

In our analysis, we assessed various towns and cities across City of Bristol and Cornwall based on four key factors: house pricing, broadband speed, school grades, and crime rates. The goal was to identify the towns and cities with the best overall living conditions by normalizing and combining these factors into a single score.

Top Ten Recommended Towns/Cities

Based on the combined scores, the top ten towns/cities are on the County - Cornwall:

1. Bude

House Pricing Score: 9.04

Broadband Speed Score: 3.66

School Grades Score: 4.57

Crime Score: 0

Overall Score: 4.32

Recommendation: Bude is a great place to live because it has very good house prices, excellent broadband speed, and good schools. Plus, it has no significant crime problems. Overall, it's a top choice for a comfortable and safe living environment.

2. Saltash

House Pricing Score: 4.70

Broadband Speed Score: 4.70

School Grades Score: 4.04

Crime Score: 0

Overall Score: 3.91

Recommendation: Saltash offers a good balance of affordable house prices and strong broadband speeds. It also has decent school grades and no major crime issues. It's a solid choice for those looking for a well-rounded living environment.

3. St Austell

House Pricing Score: 7.58

Broadband Speed Score: 3.17

School Grades Score: 4.63

Crime Score: 0

Overall Score: 3.85

Recommendation: St Austell provides good house prices and school grades. The broadband speed is lower compared to the other top towns, but it still has no significant crime problems. It's a good option if you're looking for affordable housing and quality schools.

4. Looe

House Pricing Score: 7.84

Broadband Speed Score: 3.19

School Grades Score: 4.12

Crime Score: 0

Overall Score: 3.79

Recommendation: Looe offers excellent house pricing and school grades. The broadband speed is slightly lower compared to some other towns, but it remains favorable. With no crime issues, Looe is a strong choice for affordable living and good educational opportunities.

5. Liskeard

House Pricing Score: 7.92

Broadband Speed Score: 3.03

School Grades Score: 4.09

Crime Score: 0

Overall Score: 3.76

Recommendation: Liskeard features strong house pricing and solid school grades. While the broadband speed is somewhat lower, the absence of crime makes it an attractive location for those prioritizing affordability and education over internet speed.

6. Redruth

House Pricing Score: 7.53

Broadband Speed Score: 3.31

School Grades Score: 4.15

Crime Score: 0

Overall Score: 3.75

Recommendation: Redruth stands out with competitive house prices and good school grades.

Although broadband speed is modest, the town's lack of crime and overall value make it a practical choice for residents seeking a balance between cost and quality.

7. Camborne

House Pricing Score: 8.84

Broadband Speed Score: 1.66

School Grades Score: 4.45

Crime Score: 0

Overall Score: 3.74

Recommendation: Camborne excels in house pricing and school grades, though broadband speed is significantly lower. Despite this, its zero crime rate and strong property value make it a convincing option for those who prioritize housing affordability and educational standards.

8. Penzance

House Pricing Score: 8.34

Broadband Speed Score: 2.31

School Grades Score: 4.24

Crime Score: 0

Overall Score: 3.72

Recommendation: Penzance offers very competitive house pricing and school grades, with broadband speed slightly below average. The town's low crime rate enhances its appeal for those seeking a mix of good property value and quality education.

9. Newquay

House Pricing Score: 7.35

Broadband Speed Score: 2.59

School Grades Score: 4.66

Crime Score: 0.23

Overall Score: 3.59

Recommendation: Newquay features solid house pricing and school grades with a relatively lower broadband speed. Although it has a minor crime score, the overall quality of life, including strong educational performance, makes it a suitable option.

10. St Ives

House Pricing Score: 6.27

Broadband Speed Score: 3.31

School Grades Score: 4.74

Crime Score: 0

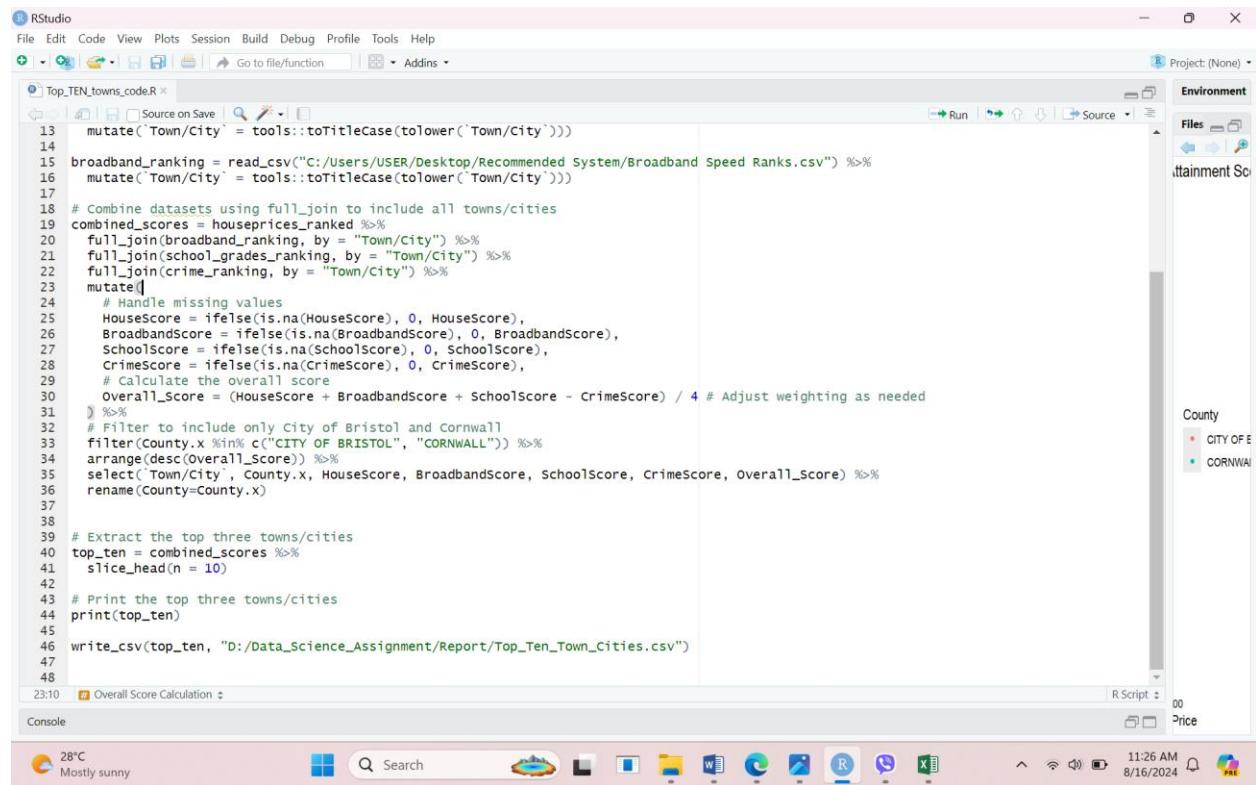
Overall Score: 3.58

Recommendation: St Ives offers very good school grades and reasonable broadband speed.

Despite its lower house pricing score, its zero crime rate and strong educational environment make it an attractive choice for families looking for a safe and educationally robust community.

Figure 16

Top Ten Towns



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ [ ] Go to file/function Addins
Top_Ten_towns_code.R
Source on Save | Run | Source
13 mutate(`Town/City` = tools::toTitleCase(tolower(`Town/City`)))
14
15 broadband_ranking = read_csv("C:/Users/USER/Desktop/Recommended System/Broadband Speed Ranks.csv") %>%
16   mutate(`Town/city` = tools::toTitlecase(tolower(`Town/City`)))
17
18 # Combine datasets using full_join to include all towns/cities
19 combined_scores = houseprices_ranked %>%
20   full_join(broadband_ranking, by = "Town/City") %>%
21   full_join(school_grades_ranking, by = "Town/City") %>%
22   full_join(crime_ranking, by = "Town/City") %>%
23   mutate(|)
24   # Handle missing values
25   HouseScore = ifelse(is.na(HouseScore), 0, HouseScore),
26   BroadbandScore = ifelse(is.na(BroadbandScore), 0, BroadbandScore),
27   SchoolScore = ifelse(is.na(SchoolScore), 0, SchoolScore),
28   CrimeScore = ifelse(is.na(CrimeScore), 0, CrimeScore),
29   # Calculate the overall score
30   Overall_Score = (HouseScore + BroadbandScore + SchoolScore - CrimeScore) / 4 # Adjust weighting as needed
31 %>%
32 # Filter to include only City of Bristol and Cornwall
33 filter(County %in% c("CITY OF BRISTOL", "CORNWALL")) %>%
34 arrange(desc(Overall_Score)) %>%
35 select(`Town/city`, County, HouseScore, BroadbandScore, SchoolScore, CrimeScore, Overall_Score) %>%
36 rename(County=County.X)
37
38
39 # Extract the top three towns/cities
40 top_ten = combined_scores %>%
41   slice_head(n = 10)
42
43 # Print the top three towns/cities
44 print(top_ten)
45
46 write_csv(top_ten, "D:/Data_Science_Assignment/Report/Top_Ten_Town_Cities.csv")
47
48
23:10 Overall Score Calculation
Console
28°C Mostly sunny
Search
11:26 AM 8/16/2024

```

Figure 17*Top Ten Towns Output*

The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Source Editor:** Shows the code for generating the top ten towns. The code filters for 'CITY OF BRISTOL' and 'CORNWALL', then selects columns and renames 'County' to 'County.x'. It then prints the top ten rows and writes them to a CSV file.
- Console:** Displays the R session history, including the command to write the CSV file.
- Environment:** Shows a legend for 'County' with two entries: 'CITY OF E' (red dot) and 'CORNWA' (green dot).
- Data View:** A preview of the data frame 'top_ten' showing 10 rows of data with columns: 'Town/City', 'County', 'HouseScore', 'BroadbandScore', 'SchoolScore', 'CrimeScore', and 'Overall_Score'.
- Bottom Status Bar:** Shows the date (8/16/2024), time (11:26 AM), and system icons.

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source on Save Go to file/function Addins
Top_TEN_towns_code.R
13   mutate(`Town/City` = tools::toTitleCase(tolower(`Town/City`)))
14
15 broadband_ranking = read_csv("c:/Users/USER/Desktop/Recommended System/Broadband Speed Ranks.csv") %>%
16   mutate(`Town/City` = tools::toTitleCase(tolower(`Town/City`)))
17
23:10 Overall Score Calculation
Overall Score Calculation

R 4.3.3 · ~/~
+ # Filter to include only City of Bristol and Cornwall
+ filter(County.x %in% c("CITY OF BRISTOL", "CORNWALL")) %>%
+ arrange(desc(Overall_Score)) %>%
+ select(-Town/City, -County.x, HouseScore, BroadbandScore, SchoolScore, CrimeScore, Overall_Score) %>%
+ rename(County=County.x)
>
>
> # Extract the top three towns/cities
> top_ten = combined_scores %>%
+ slice_head(n = 10)
>
> # Print the top three towns/cities
> print(top_ten)
# A tibble: 10 × 7
  `Town/City` County HouseScore BroadbandScore SchoolScore CrimeScore Overall_Score
    <chr>     <chr>      <dbl>        <dbl>       <dbl>      <dbl>        <dbl>
1 Bude       CORNWALL  9.04        3.66       4.57       0          4.32
2 Saltash    CORNWALL  6.9         4.70       4.04       0          3.91
3 St Austell CORNWALL  7.58        3.17       4.63       0          3.85
4 Looe       CORNWALL  7.84        3.19       4.12       0          3.79
5 Liskeard   CORNWALL  7.92        3.03       4.09       0          3.76
6 Redruth    CORNWALL  7.53        3.31       4.15       0          3.75
7 Camborne   CORNWALL  8.84        1.66       4.45       0          3.74
8 Penzance   CORNWALL  8.34        2.31       4.24       0          3.72
9 Newquay   CORNWALL  7.35        2.59       4.66       0.233     3.59
10 St Ives    CORNWALL 6.27        3.31       4.74       0          3.58
>
> write_csv(top_ten, "D:/Data_Science_Assignment/Report/Top_Ten_Town_Cities.csv")
>

```

Figure 18*Top Ten Towns Excel*

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|----|------------|---------|----------|----------|-----------|-----------|---------------|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Town/City | County | HouseSco | Broadban | SchoolSco | CrimeScor | Overall_Score | | | | | | | | | | | |
| 2 | Bude | CORNWAL | 9.041346 | 3.656522 | 4.565 | 0 | 4.315717 | | | | | | | | | | | |
| 3 | Saltash | CORNWAL | 6.9 | 4.6975 | 4.035 | 0 | 3.908125 | | | | | | | | | | | |
| 4 | St Austell | CORNWAL | 7.58136 | 3.173185 | 4.631667 | 0 | 3.846553 | | | | | | | | | | | |
| 5 | Looe | CORNWAL | 7.841402 | 3.190426 | 4.115 | 0 | 3.786707 | | | | | | | | | | | |
| 6 | Liskeard | CORNWAL | 7.916667 | 3.032778 | 4.09 | 0 | 3.759861 | | | | | | | | | | | |
| 7 | Redruth | CORNWAL | 7.527667 | 3.306452 | 4.15 | 0 | 3.74603 | | | | | | | | | | | |
| 8 | Camborne | CORNWAL | 8.84 | 1.657826 | 4.45 | 0 | 3.736957 | | | | | | | | | | | |
| 9 | Penzance | CORNWAL | 8.336708 | 2.31089 | 4.243333 | 0 | 3.722733 | | | | | | | | | | | |
| 10 | Newquay | CORNWAL | 7.354583 | 2.589744 | 4.6575 | 0.23322 | 3.592152 | | | | | | | | | | | |
| 11 | St Ives | CORNWAL | 6.2655 | 3.312645 | 4.735 | 0 | 3.578286 | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | | | | | | |
| 31 | | | | | | | | | | | | | | | | | | |

READY 28°C Mostly sunny

Search File Home Insert Page Layout Formulas Data Review View
Sign in 11:29 AM 8/16/2024

Reflection

This analysis aimed to identify the top towns and cities for potential relocation or investment by evaluating key factors such as house prices, broadband speed, school grades, and crime rates. By combining these factors into a single overall score, we were able to highlight the best options according to our criteria. The insights gained are as below:

Top Choices Identified: The towns of Bude, Saltash, St Austell, Looe, Liskeard, Redruth, Camborne, Penzance, Newquay and St Ives emerged as the top ten recommendations. Bude scored the highest overall due to its strong performance in house pricing, broadband speed, and school grades, combined with a lack of significant crime issues and St Ives showed the lowest score overall.

Balanced Criteria: The use of a combined score allowed us to weigh each factor equally, offering a balanced view of each town's strengths and weaknesses.

Limitations:

Data Completeness: The analysis is limited by the data available. For instance, if some towns or cities were missing from any of the datasets because of LSOA_code being NA, then the towns associated with that Lsoa_code were also NA. So, they could not be evaluated which lead to towns/city not being present in the merged data. Overall, this analysis provides a useful starting point for evaluating potential relocation or investment options.

Overall Score

The Overall Score is a combined measure that reflects the desirability of each town or city based on four key factors: house prices, broadband speed, school grades, and crime rates.

Here's how the score is calculated:

House Pricing Score: This reflects how affordable or expensive houses are in the town. Higher scores indicate better value for money.

Broadband Speed Score: This measures the quality of internet service. Higher scores indicate faster and more reliable broadband.

School Grades Score: This shows the quality of education in the area. Higher scores represent better school performance.

Crime Score: This reflects the safety of the town. Higher scores indicate higher crime rates, which is treated as a negative factor. We subtract this score to account for the impact of crime on the overall desirability.

Figure 19*Overall Score*

```

1 # Overall Score Calculation -----
2 library(tidyverse)
3
4 # Load and standardize datasets
5 houseprices_ranked = read_csv("C:/Users/USER/Desktop/Recommended System/House Pricing Ranks.csv") %>%
6   mutate(`Town/City` = tools::toTitleCase(tolower(`Town/City`)))
7
8 school_grades_ranking = read_csv("C:/Users/USER/Desktop/Recommended System/School Grades Ranks.csv") %>%
9   mutate(`Town/City` = tools::toTitleCase(tolower(`Town/City`)))
10
11 crime_ranking = read_csv("C:/Users/USER/Desktop/Recommended System/Crime Ranking.csv") %>%
12   mutate(`Town/City` = tools::toTitleCase(tolower(`Town/City`)))
13
14 broadband_ranking = read_csv("C:/Users/USER/Desktop/Recommended System/Broadband Speed Ranks.csv") %>%
15   mutate(`Town/City` = tools::toTitleCase(tolower(`Town/City`)))
16
17 # Combine datasets using full_join to include all towns/cities
18 combined_scores = houseprices_ranked %>%
19   full_join(broadband_ranking, by = "Town/City") %>%
20   full_join(school_grades_ranking, by = "Town/City") %>%
21   full_join(crime_ranking, by = "Town/City") %>%
22   mutate(
23     # Handle missing values
24     HouseScore = ifelse(is.na(HouseScore), 0, HouseScore),
25     Broadbandscore = ifelse(is.na(Broadbandscore), 0, Broadbandscore),
26     SchoolScore = ifelse(is.na(SchoolScore), 0, SchoolScore),
27     CrimeScore = ifelse(is.na(CrimeScore), 0, CrimeScore),
28     # Calculate the overall score
29     Overall_Score = (HouseScore + Broadbandscore + SchoolScore - CrimeScore) / 4 # Adjust weighting as needed
30   ) %>%
31   # Filter to include only City of Bristol and Cornwall
32   filter(County.x %in% c("CITY OF BRISTOL", "CORNWALL")) %>%
33   arrange(desc(Overall_Score)) %>%
34   select(`Town/City`, County.x, HouseScore, Broadbandscore, SchoolScore, CrimeScore, Overall_Score)
35
36 write_csv(combined_scores, "C:/Users/USER/Desktop/Recommended System/Overall_Scores.csv")
891 Overall Score Calculation

```

Console

24°C Partly cloudy

File Edit View Insert Cell Help

Run Source

House Price

County

CITY OF BRISTOL CORNWALL

R Script

8:00 PM 8/11/2024

Legal and Ethical Issues

In conducting this analysis, it is essential to address both legal and ethical considerations. The datasets used for this study are publicly available, which means they have been provided by governmental or authorized entities for public access and use. Utilizing publicly available datasets ensures compliance with legal requirements concerning data access and use.

Conclusion

This analysis used the data mining lifecycle to assess towns and cities based on house prices, broadband speed, school grades, and crime rates. Bude, Saltash, St Austell, Looe, Liskeard, Redruth, Camborne, Penzance, Newquay and St Ives as top recommendations due to their overall favorable scores by integrating these factors into a combined score. The analysis was based on publicly available datasets, ensuring relevance and accessibility.

References

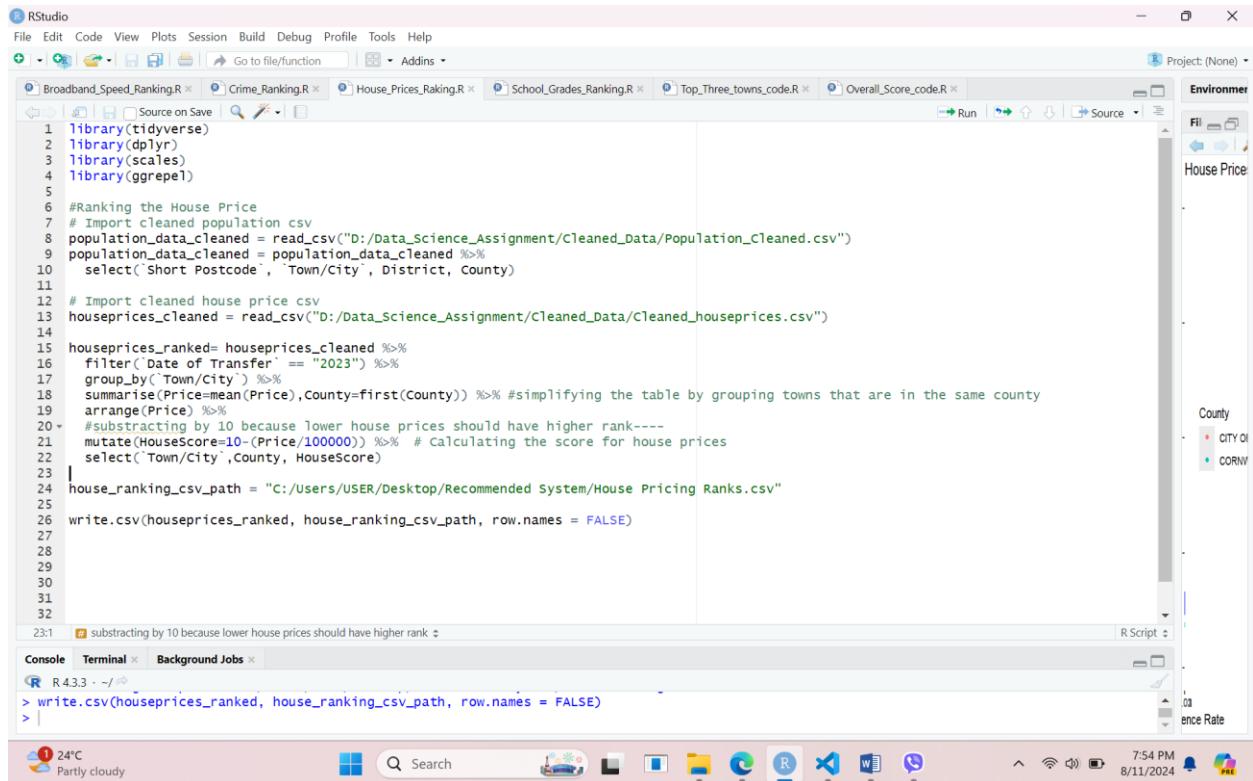
- GeeksforGeeks. (n.d.). *Data Cleaning in R - GeeksforGeeks*. Retrieved from <https://www.geeksforgeeks.org/data-cleaning-in-r/>
- Government, U. (n.d.). *House pricing dataset. UK Government Statistical Data Sets*. Retrieved from <https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads>
- Ofcom. (n.d.). *Broadband speed data*. Retrieved from <https://www.ofcom.org.uk/research-and-data/multi-sector-research/infrastructure-research/connected-nations-2018/data-downloads>
- Police, U. (n.d.). *Crime dataset*. Retrieved from <https://data.police.uk/data/>
- School dataset, UK Government Compare School Performance Service. (n.d.). Retrieved from <https://www.compare-school-performance.service.gov.uk/download-data?currentstep=datatypes®iontype=all&la=0&downloadYear=2018-2019&datatypes=ks5>
- Wickham, H. (n.d.). *Exploratory Data analysis. R for Data Science*. Retrieved from <https://r4ds.had.co.nz/exploratory-data-analysis.html>

Appendix

1. House Prices Ranking

Figure 20

House Prices Ranking



The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project:** Project (None).
- Code Editor:** The main pane displays R code for ranking house prices. The code includes imports for tidyverse, dplyr, scales, and ggrepel. It reads cleaned population and house price data, filters for 2023 transfers, groups by town/city, calculates mean price per town, arranges by price, and subtracts 10 from the price to get a house score. Finally, it writes the ranked data to a CSV file.
- Environment:** A sidebar showing variables and their values, including 'County' with entries 'CITY OF' and 'CORNW'.
- Console:** The bottom pane shows the R console output, confirming the execution of the code and the creation of the CSV file.
- System Status:** The taskbar at the bottom shows the date (8/11/2024), time (7:54 PM), battery level (24%), and system icons.

```

library(tidyverse)
library(dplyr)
library(scales)
library(ggrepel)
#Ranking the House Price
# Import cleaned population csv
population_data_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Population_Cleaned.csv")
population_data_cleaned = population_data_cleaned %>%
  select(-Short Postcode, -Town/City, District, County)
# Import cleaned house price csv
houseprices_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Cleaned_houseprices.csv")
houseprices_ranked= houseprices_cleaned %>%
  filter(Date of Transfer == "2023") %>%
  group_by(Town/City) %>%
  summarise(Price=mean(Price),County=first(County)) %>% #simplifying the table by grouping towns that are in the same county
  arrange(Price) %>%
  #subtracting by 10 because lower house prices should have higher rank----
  mutate(HouseScore=10-(Price/100000)) %>% # Calculating the score for house prices
  select(-Town/City,-County,HouseScore)
house_ranking_csv_path = "C:/Users/USER/Desktop/Recommended System/House Pricing Ranks.csv"
write.csv(houseprices_ranked, house_ranking_csv_path, row.names = FALSE)

```

2. School Grades Ranking

Figure 21

School Grades Ranking

```

library(tidyverse)
library(dplyr)
library(scales)
library(ggrepel)

# Ranking the School Grades

# Import cleaned school grades csv
school_grades_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/School_Dataset_Cleaned.csv")
View(school_grades_cleaned)
# Rank school grades by Town/City
school_grades_ranked = school_grades_cleaned %>%
  filter(Year == 2023) %>% # Filtering for the year 2023
  group_by(Town) %>%
  summarise(`Average Attainment Score` = mean(`Attainment Score`),
  County = first(County)) %>% # Simplifying the table by grouping towns that are in the same county
  arrange(desc(`Average Attainment Score`)) %>%
  rename(Town/city = Town) %>% # Renaming Town to Town/City
  # Calculate the score, assuming higher attainment scores should have a higher rank
  mutate(SchoolScore = (`Average Attainment Score` / 10)) %>% # Calculating the score for school grades
  select(-Town/city, -County, SchoolScore)
  
# Define path to save the school grades ranking csv
school_grades_ranking_csv_path = "C:/Users/USER/Desktop/Recommended System/School Grades Ranks.csv"

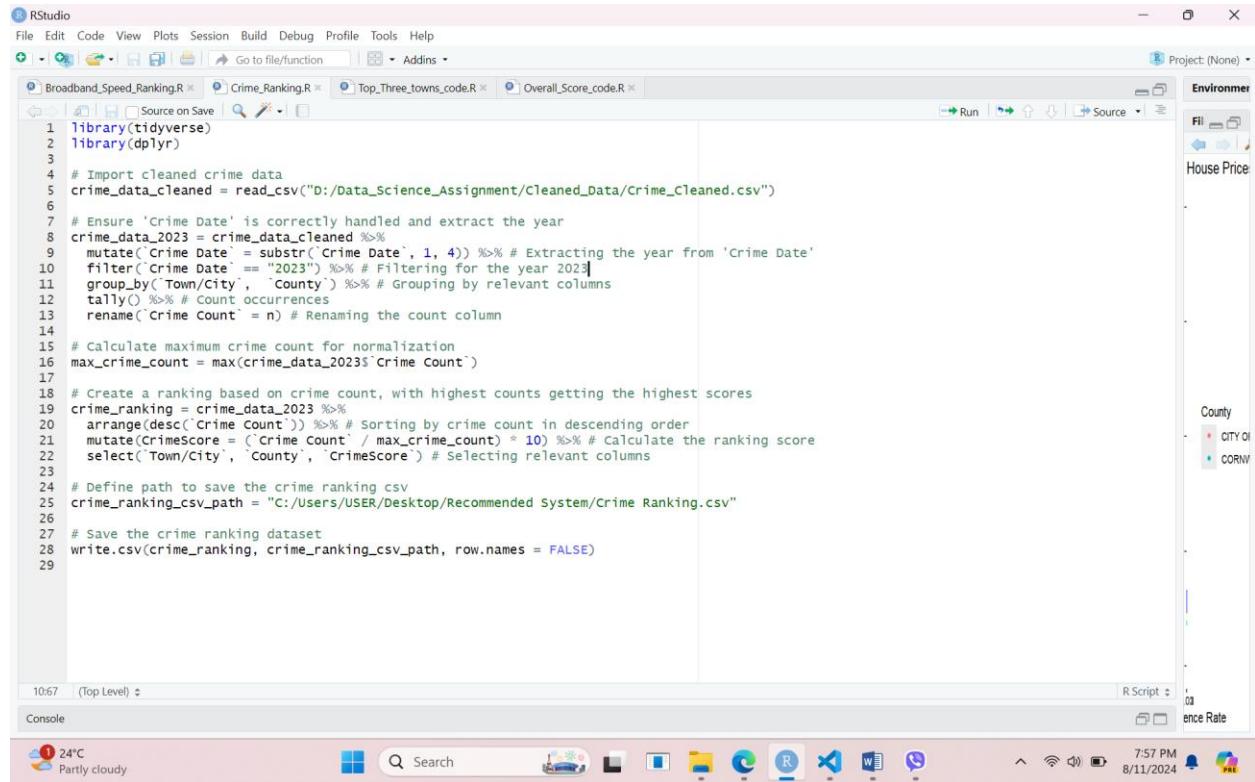
# Save the cleaned dataset
write.csv(school_grades_ranked, school_grades_ranking_csv_path, row.names = FALSE)

```

3. Crime Ranking

Figure 22

Crime Ranking



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - X
Broadband_Speed_Ranking.R Crime_Ranking.R Top_Three_towns_code.R Overall_Score_code.R
Source on Save Run Source
1 library(tidyverse)
2 library(dplyr)
3
4 # Import cleaned crime data
5 crime_data_cleaned = read_csv("D:/Data_Science_Assignment/cleaned_Data/crime_Cleaned.csv")
6
7 # Ensure 'Crime Date' is correctly handled and extract the year
8 crime_data_2023 = crime_data_cleaned %>%
9   mutate(Crime Date = substr('Crime Date', 1, 4)) %>% # Extracting the year from 'Crime Date'
10  filter(Crime Date == "2023") %>% # Filtering for the year 2023
11  group_by(Town/City, County) %>% # Grouping by relevant columns
12  tally() %>% # Count occurrences
13  rename(`Crime Count` = n) # Renaming the count column
14
15 # Calculate maximum crime count for normalization
16 max_crime_count = max(crime_data_2023$`Crime Count`)
17
18 # Create a ranking based on crime count, with highest counts getting the highest scores
19 crime_ranking = crime_data_2023 %>%
20   arrange(desc(`Crime Count`)) %>% # Sorting by crime count in descending order
21   mutate(CrimeScore = (`Crime Count` / max_crime_count) * 10) %>% # Calculate the ranking score
22   select(`Town/City`, `County`, `CrimeScore`) # Selecting relevant columns
23
24 # Define path to save the crime ranking csv
25 crime_ranking_csv_path = "C:/Users/USER/Desktop/Recommended System/Crime Ranking.csv"
26
27 # Save the crime ranking dataset
28 write.csv(crime_ranking, crime_ranking_csv_path, row.names = FALSE)
29

```

10:57 (Top Level) R Script

Console

24°C Partly cloudy Search

7:57 PM 8/11/2024

4. Broadband Ranking

Figure 23

Broadband Speed Ranking

```
library(tidyverse)
library(dplyr)
library(scales)
library(ggrepel)

# Ranking the Broadband Speed

# Import cleaned broadband speed csv
broadband_speed_cleaned = read_csv("D:/Data_Science_Assignment/Cleaned_Data/Broadband_speed_cleaned_data.csv")

# Rank broadband speeds by Town/City
broadband_speed_ranked = broadband_speed_cleaned %>%
  group_by(`Town/City`) %>%
  summarise(`Average Download Speed` = mean(`Average download speed (Mbit/s)`),
            County = first(County)) %>% # Simplifying the table by grouping towns that are in the same county
  arrange(`Average Download Speed`) %>%
  #lower speeds should have a lower rank---
  mutate(BroadbandScore = `Average Download Speed` / 10) %>% # Calculating the score for broadband speed
  select(`Town/City`, County, BroadbandScore)

# Define path to save the broadband ranking csv
broadband_ranking_csv_path = "C:/Users/USER/Desktop/Recommended System/Broadband Speed Ranks.csv"

# Save the cleaned dataset
write.csv(broadband_speed_ranked, broadband_ranking_csv_path, row.names = FALSE)
```