Individual Report

Dipesh Kumar Khadgi

B.Sc. (Hons) in Computing, Softwarica College of IT and E-Commerce

ST5014CEM Data Science for Developers

Siddhartha Neupane

19 August, 2024

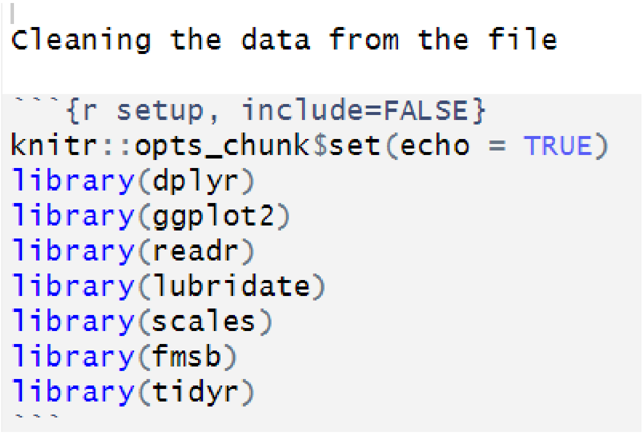
Table of Contents

Table of Figures

Introduction

The rapidly expanding field of data science has made ethical and legal considerations more crucial. These capacities will only increase along with the problems of ensuring that data collection, processing, and analysis are done correctly. Because of worries about discrimination, misuse of data, and invasion of privacy, public and private sectors have been increasingly stringent in enforcing ethical laws and regulations. This study examines the intricate relationships that exist between data science endeavors and the legal and moral frameworks that govern them. Through an examination of real-world applications, this research aims to show how important it is to maintain these principles in order to maintain public trust and produce socially responsible outcomes.

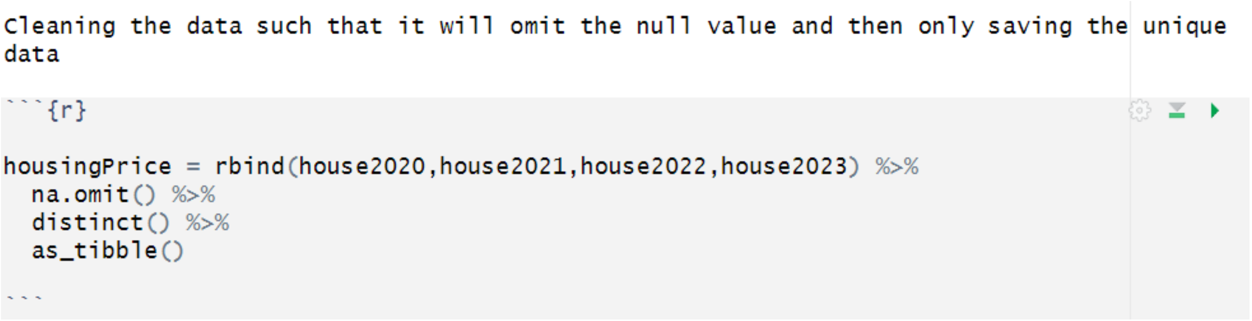
Cleaning Data



Cleaning the data from the file

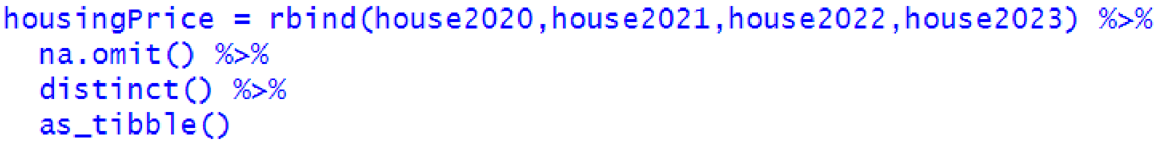
~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(ggplot2)
library(readr)
library(lubridate)
library(scales)
library(fmsb)
library(tidyr)
```
~~~

House Sales

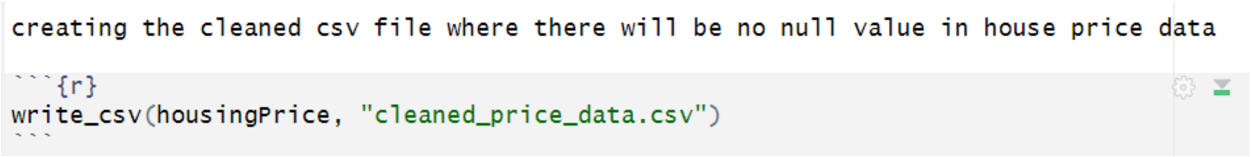


Cleaning the data such that it will omit the null value and then only saving the unique data

~~~
```{r}

housingPrice = rbind(house2020,house2021,house2022,house2023) %>%
  na.omit() %>%
  distinct() %>%
  as_tibble()

```
~~~

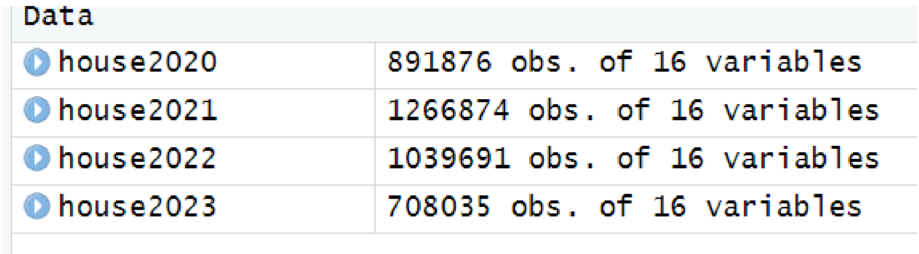


```
housingPrice = rbind(house2020,house2021,house2022,house2023) %>%
  na.omit() %>%
  distinct() %>%
  as_tibble()
```

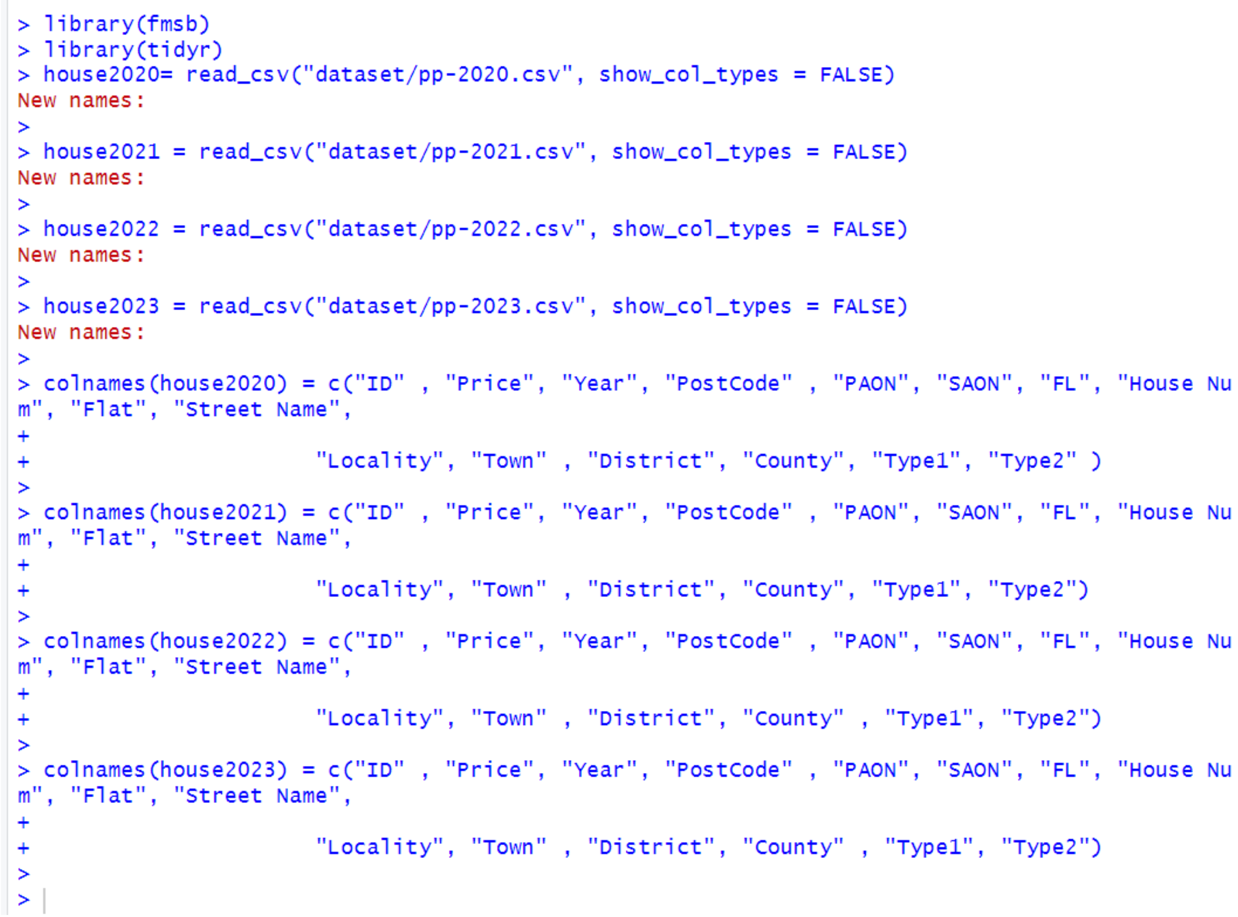


creating the cleaned csv file where there will be no null value in house price data

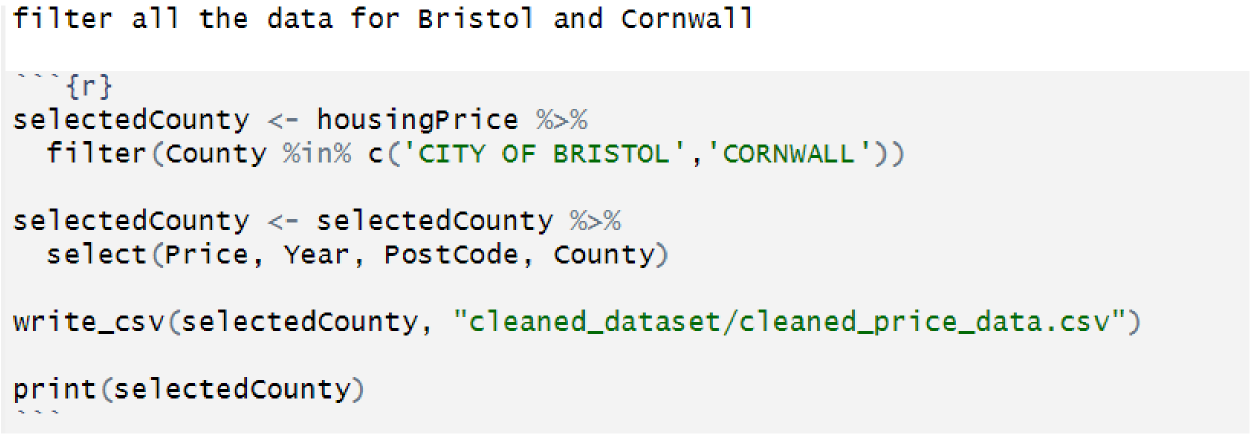
~~~
```{r}
write_csv(housingPrice, "cleaned_price_data.csv")
```
~~~



Data	
house2020	891876 obs. of 16 variables
house2021	1266874 obs. of 16 variables
house2022	1039691 obs. of 16 variables
house2023	708035 obs. of 16 variables



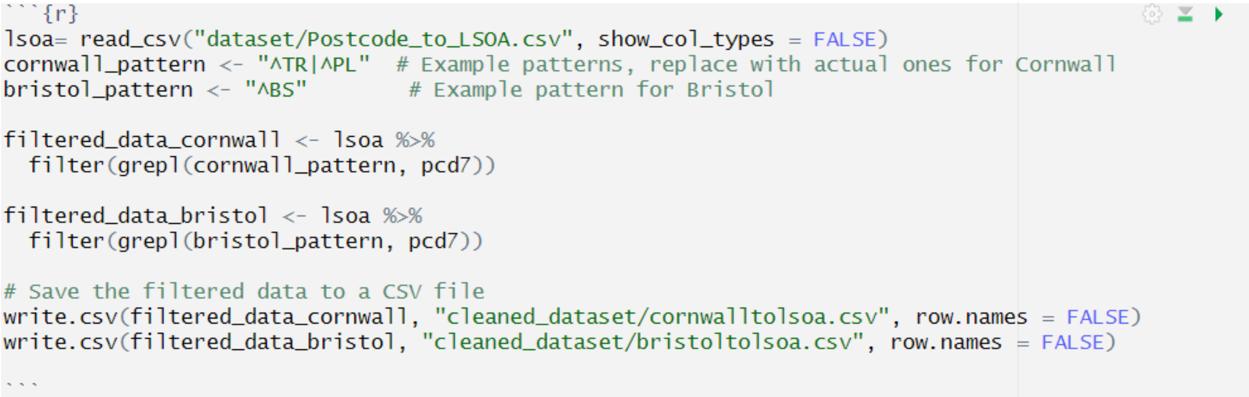
```
> library(fmsb)
> library(tidyr)
> house2020= read_csv("dataset/pp-2020.csv", show_col_types = FALSE)
New names:
>
> house2021 = read_csv("dataset/pp-2021.csv", show_col_types = FALSE)
New names:
>
> house2022 = read_csv("dataset/pp-2022.csv", show_col_types = FALSE)
New names:
>
> house2023 = read_csv("dataset/pp-2023.csv", show_col_types = FALSE)
New names:
>
> colnames(house2020) = c("ID" , "Price", "Year", "PostCode" , "PAON", "SAON", "FL", "House Nu
m", "Flat", "Street Name",
+
+                       "Locality", "Town" , "District", "County", "Type1", "Type2" )
>
> colnames(house2021) = c("ID" , "Price", "Year", "PostCode" , "PAON", "SAON", "FL", "House Nu
m", "Flat", "Street Name",
+
+                       "Locality", "Town" , "District", "County", "Type1", "Type2")
>
> colnames(house2022) = c("ID" , "Price", "Year", "PostCode" , "PAON", "SAON", "FL", "House Nu
m", "Flat", "Street Name",
+
+                       "Locality", "Town" , "District", "County" , "Type1", "Type2")
>
> colnames(house2023) = c("ID" , "Price", "Year", "PostCode" , "PAON", "SAON", "FL", "House Nu
m", "Flat", "Street Name",
+
+                       "Locality", "Town" , "District", "County" , "Type1", "Type2")
>
> 
```

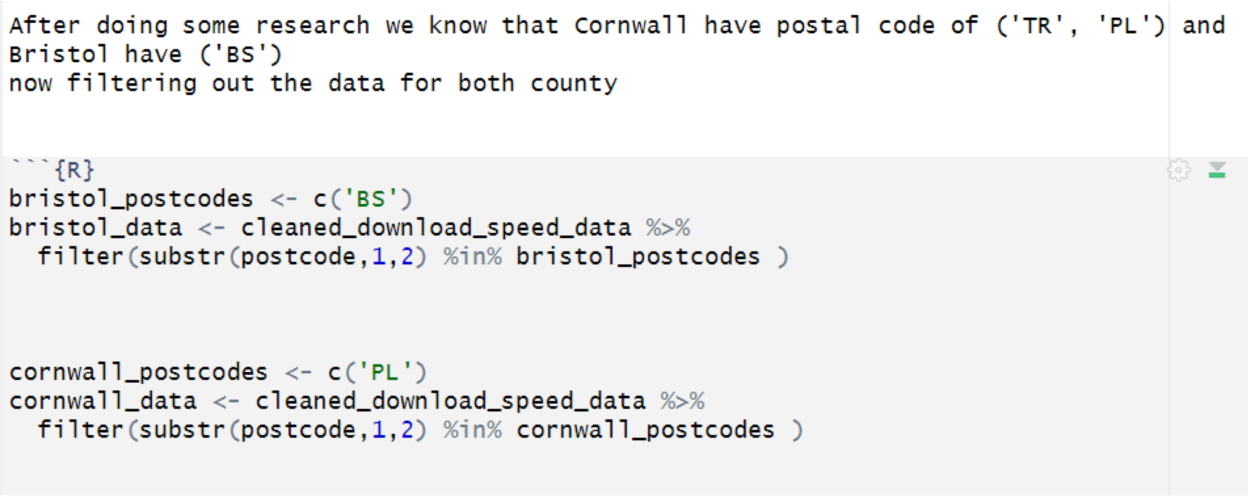


filter all the data for Bristol and Cornwall

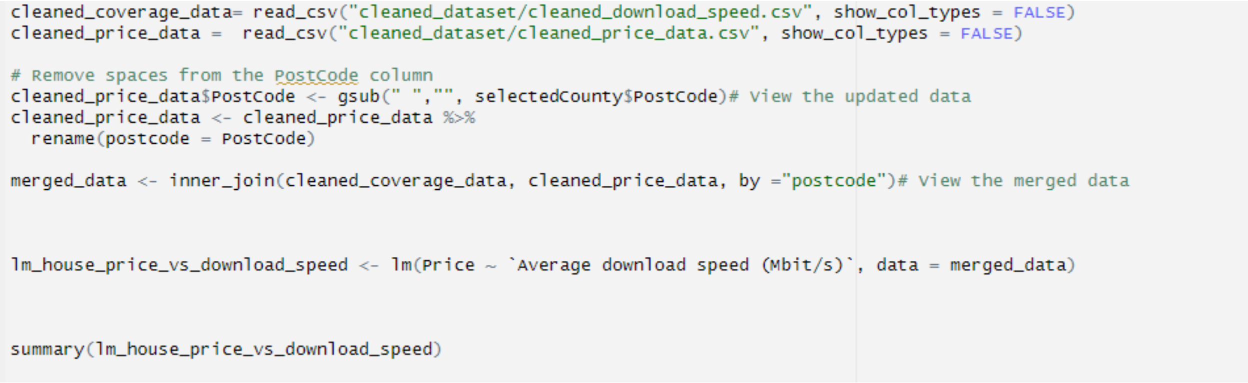
~~~
```{r}
selectedCounty <- housingPrice %>%
  filter(County %in% c('CITY OF BRISTOL','CORNWALL'))

selectedCounty <- selectedCounty %>%
  select(Price, Year, PostCode, County)

write_csv(selectedCounty, "cleaned_dataset/cleaned_price_data.csv")

print(selectedCounty)
```
~~~

Towns and Postcodes

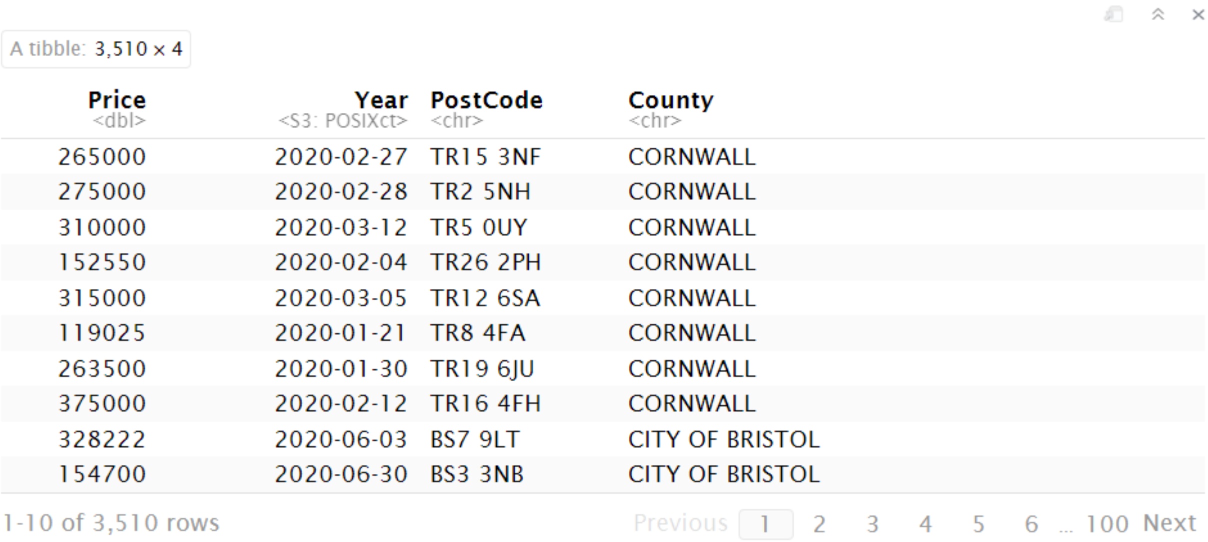


~~~
```{r}
lsoa= read_csv("dataset/Postcode_to_LSOA.csv", show_col_types = FALSE)
cornwall_pattern <- "^TR|^PL"  # Example patterns, replace with actual ones for Cornwall
bristol_pattern <- "^BS"        # Example pattern for Bristol

filtered_data_cornwall <- lsoa %>%
  filter(grepl(cornwall_pattern, pcd7))

filtered_data_bristol <- lsoa %>%
  filter(grepl(bristol_pattern, pcd7))

# Save the filtered data to a CSV file
write.csv(filtered_data_cornwall, "cleaned_dataset/cornwalltolsoa.csv", row.names = FALSE)
write.csv(filtered_data_bristol, "cleaned_dataset/bristoltolsoa.csv", row.names = FALSE)

~~~
```



```
After doing some research we know that Cornwall have postal code of ('TR', 'PL') and
Bristol have ('BS')
now filtering out the data for both county

```{R}
bristol_postcodes <- c('BS')
bristol_data <- cleaned_download_speed_data %>%
  filter(substr(postcode,1,2) %in% bristol_postcodes )



cornwall_postcodes <- c('PL')
cornwall_data <- cleaned_download_speed_data %>%
  filter(substr(postcode,1,2) %in% cornwall_postcodes )
```



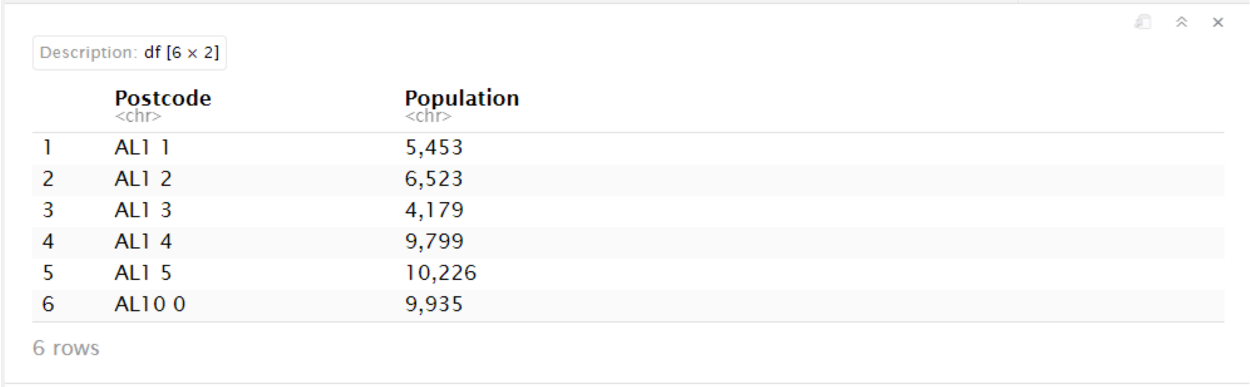
```
cleaned_coverage_data= read_csv("cleaned_dataset/cleaned_download_speed.csv", show_col_types = FALSE)
cleaned_price_data =  read_csv("cleaned_dataset/cleaned_price_data.csv", show_col_types = FALSE)

# Remove spaces from the PostCode column
cleaned_price_data$PostCode <- gsub(" ","", selectedCounty$PostCode)# View the updated data
cleaned_price_data <- cleaned_price_data %>%
  rename(postcode = PostCode)

merged_data <- inner_join(cleaned_coverage_data, cleaned_price_data, by ="postcode")# View the merged data


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


summary(lm_house_price_vs_download_speed)
```



A tibble: 3,510 × 4

Price	Year	PostCode	County
<dbl>	<S3: POSIXct>	<chr>	<chr>
265000	2020-02-27	TR15 3NF	CORNWALL
275000	2020-02-28	TR2 5NH	CORNWALL
310000	2020-03-12	TR5 0UY	CORNWALL
152550	2020-02-04	TR26 2PH	CORNWALL
315000	2020-03-05	TR12 6SA	CORNWALL
119025	2020-01-21	TR8 4FA	CORNWALL
263500	2020-01-30	TR19 6JU	CORNWALL
375000	2020-02-12	TR16 4FH	CORNWALL
328222	2020-06-03	BS7 9LT	CITY OF BRISTOL
154700	2020-06-30	BS3 3NB	CITY OF BRISTOL

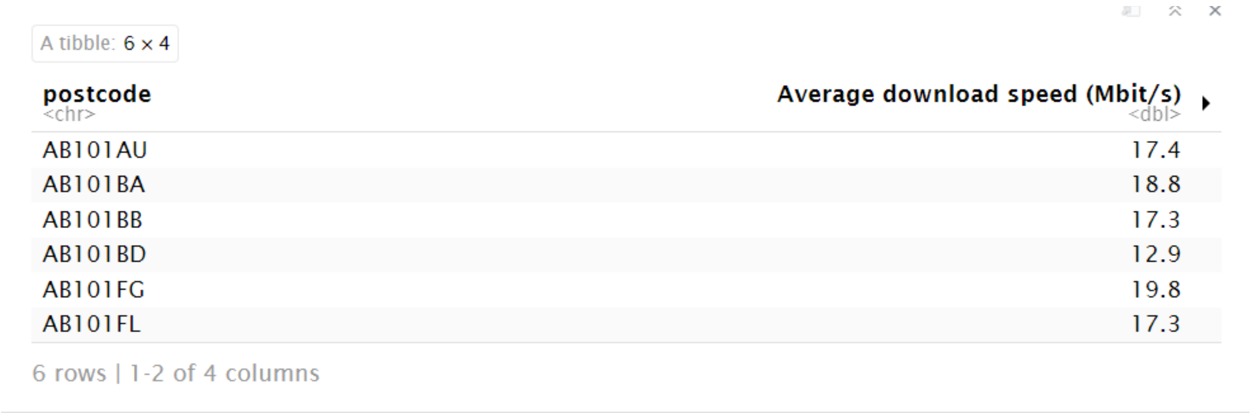
1-10 of 3,510 rows



Description: df [6 × 2]

	Postcode	Population
	<chr>	<chr>
1	AL1 1	5,453
2	AL1 2	6,523
3	AL1 3	4,179
4	AL1 4	9,799
5	AL1 5	10,226
6	AL10 0	9,935

6 rows



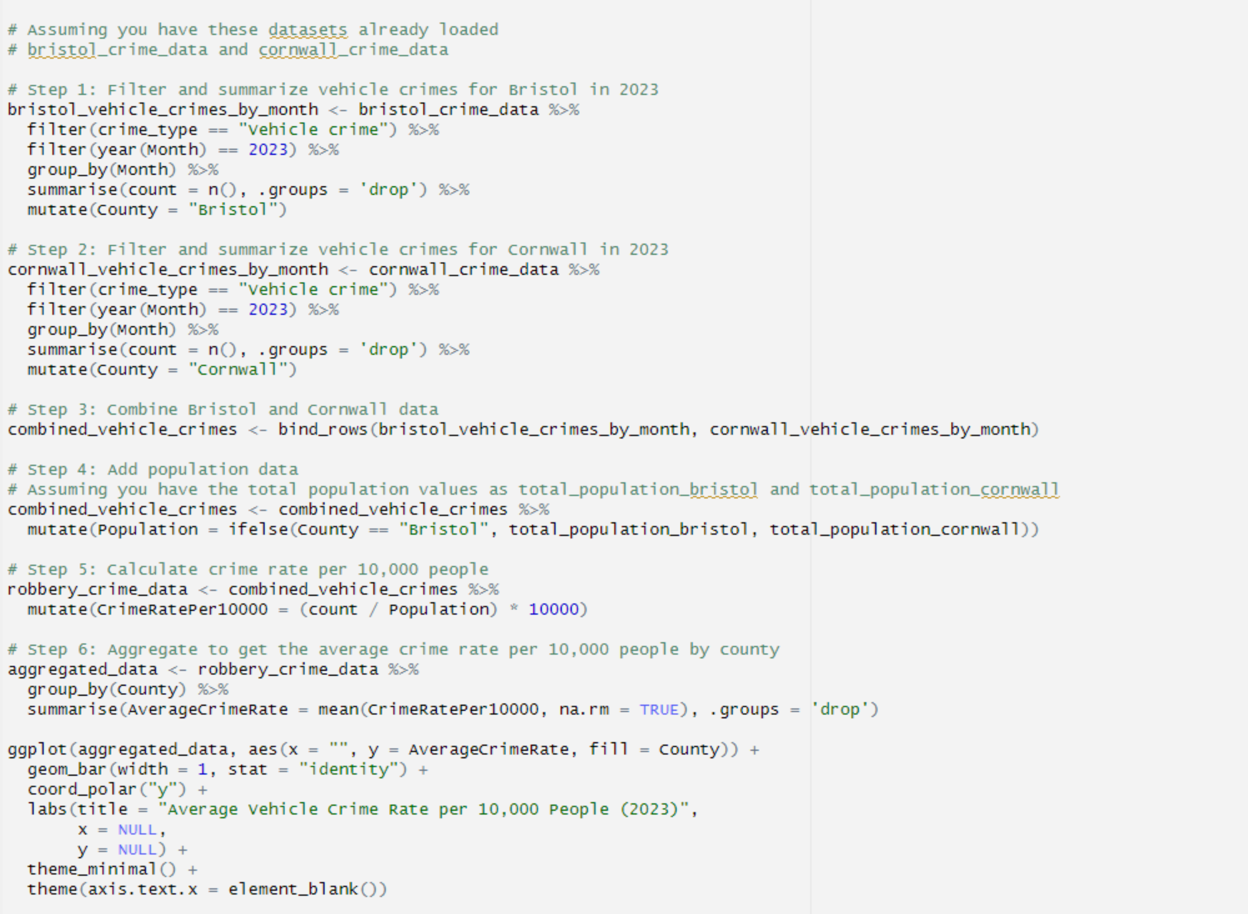
A tibble: 6 × 4

postcode	Average download speed (Mbit/s)
<chr>	<dbl>
AB101AU	17.4
AB101BA	18.8
AB101BB	17.3
AB101BD	12.9
AB101FG	19.8
AB101FL	17.3

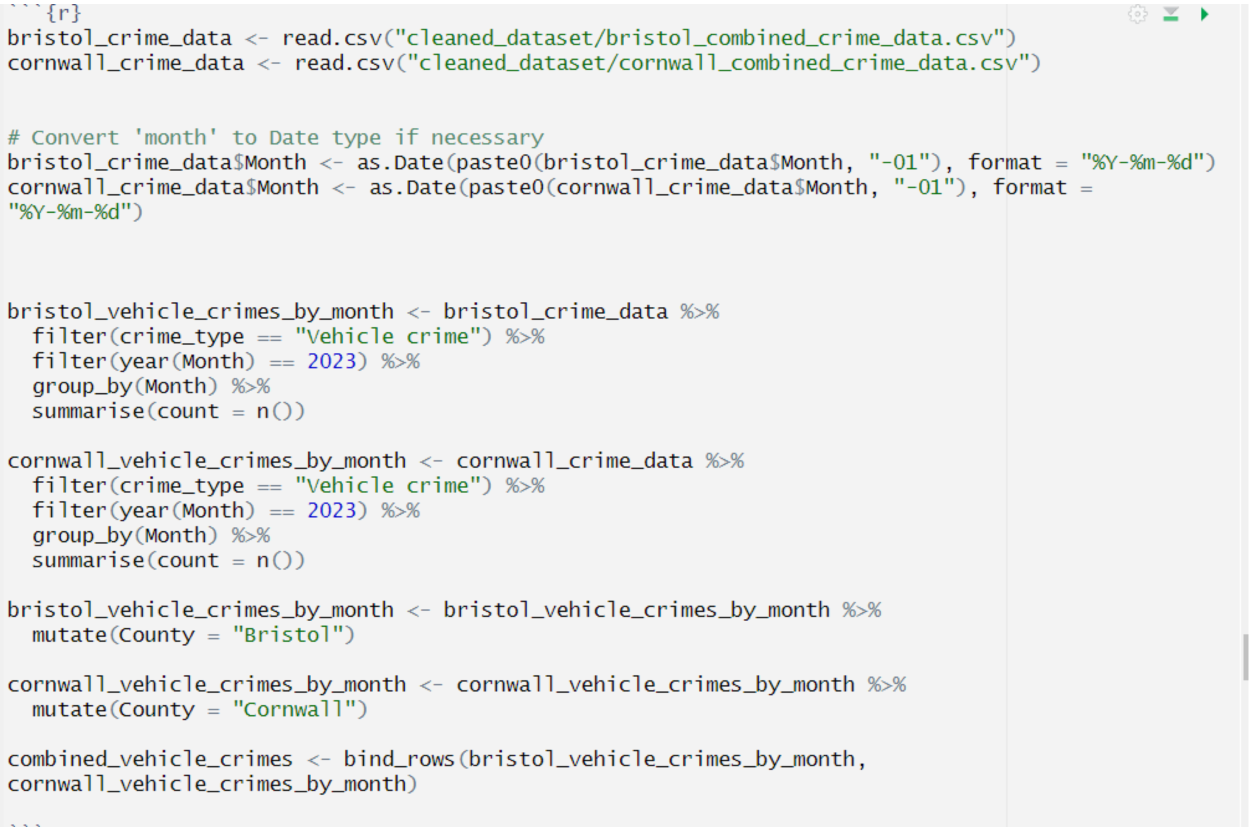
6 rows | 1-2 of 4 columns

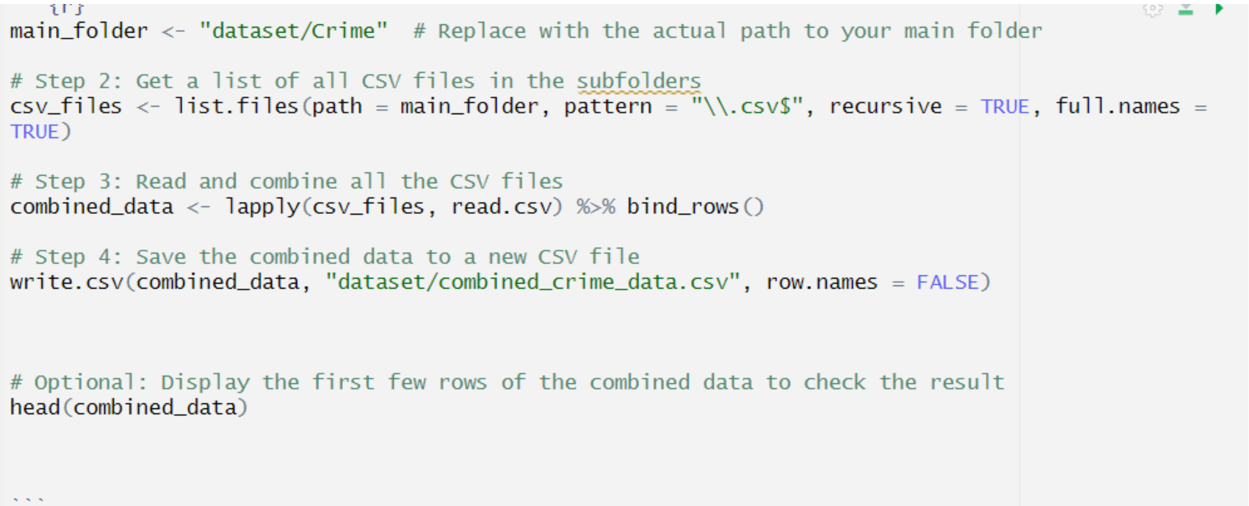
Board band Speeds

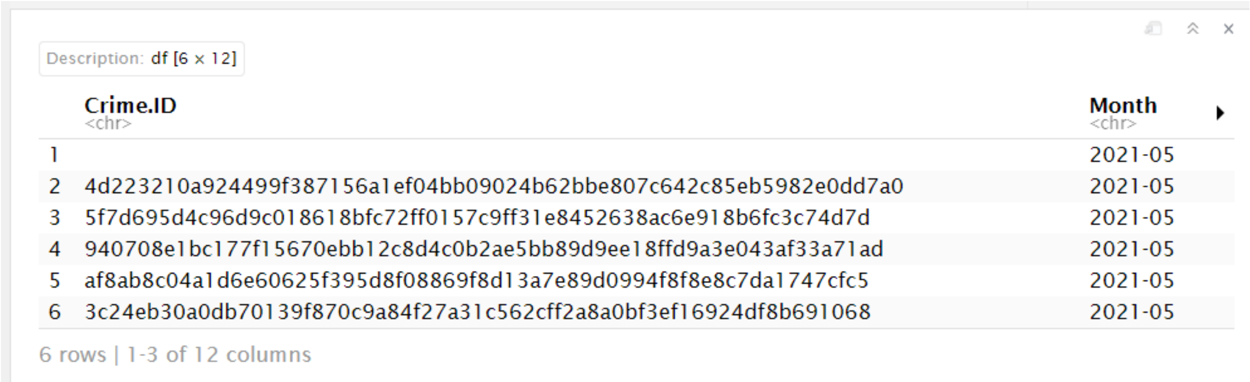
Crime



```
# Assuming you have these datasets already loaded
# bristol_crime_data and cornwall_crime_data

# Step 1: Filter and summarize vehicle crimes for Bristol in 2023
bristol_vehicle_crimes_by_month <- bristol_crime_data %>%
  filter(crime_type == "Vehicle crime") %>%
  filter(year(Month) == 2023) %>%
  group_by(Month) %>%
  summarise(count = n(), .groups = 'drop') %>%
  mutate(County = "Bristol")

# Step 2: Filter and summarize vehicle crimes for Cornwall in 2023
cornwall_vehicle_crimes_by_month <- cornwall_crime_data %>%
  filter(crime_type == "Vehicle crime") %>%
  filter(year(Month) == 2023) %>%
  group_by(Month) %>%
  summarise(count = n(), .groups = 'drop') %>%
  mutate(County = "Cornwall")

# Step 3: Combine Bristol and Cornwall data
combined_vehicle_crimes <- bind_rows(bristol_vehicle_crimes_by_month, cornwall_vehicle_crimes_by_month)

# Step 4: Add population data
# Assuming you have the total population values as total_population_bristol and total_population_cornwall
combined_vehicle_crimes <- combined_vehicle_crimes %>%
  mutate(Population = ifelse(County == "Bristol", total_population_bristol, total_population_cornwall))

# Step 5: Calculate crime rate per 10,000 people
robbery_crime_data <- combined_vehicle_crimes %>%
  mutate(CrimeRatePer10000 = (count / Population) * 10000)

# Step 6: Aggregate to get the average crime rate per 10,000 people by county
aggregated_data <- robbery_crime_data %>%
  group_by(County) %>%
  summarise(AverageCrimeRate = mean(CrimeRatePer10000, na.rm = TRUE), .groups = 'drop')

ggplot(aggregated_data, aes(x = "", y = AverageCrimeRate, fill = County)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y") +
  labs(title = "Average Vehicle Crime Rate per 10,000 People (2023)",
       x = NULL,
       y = NULL) +
  theme_minimal() +
  theme(axis.text.x = element_blank())
```



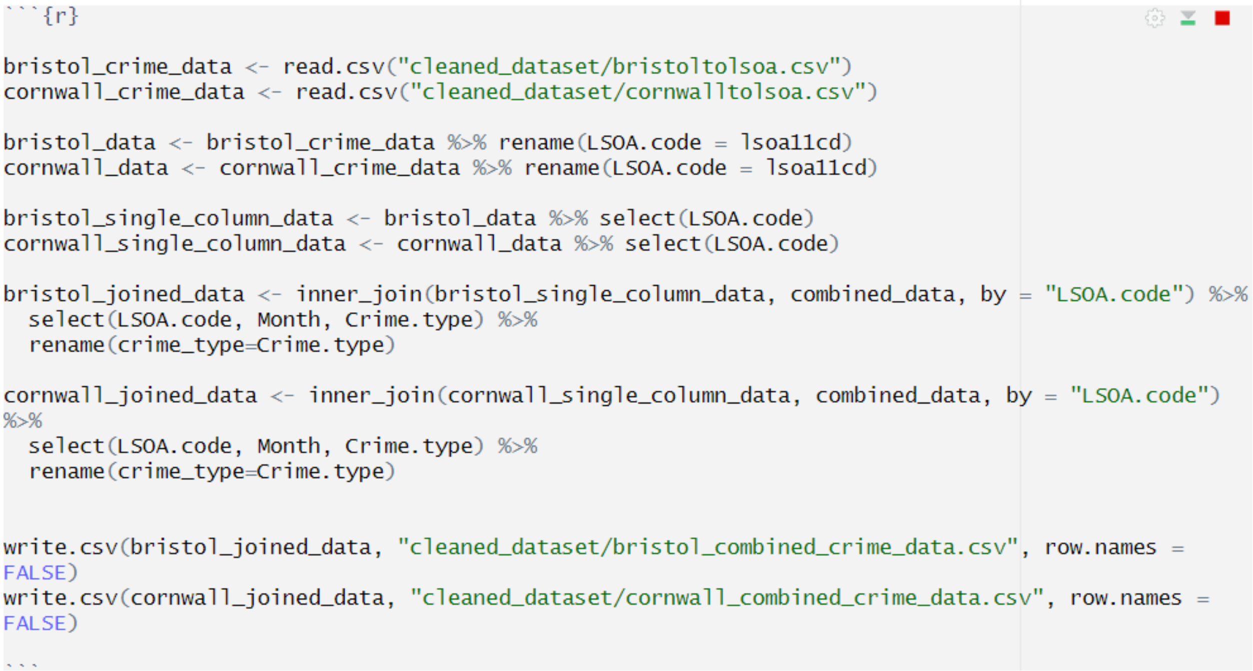
~~~r
```{r}
bristol_crime_data <- read.csv("cleaned_dataset/bristol_combined_crime_data.csv")
cornwall_crime_data <- read.csv("cleaned_dataset/cornwall_combined_crime_data.csv")


# Convert 'month' to Date type if necessary
bristol_crime_data$Month <- as.Date(paste0(bristol_crime_data$Month, "-01"), format = "%Y-%m-%d")
cornwall_crime_data$Month <- as.Date(paste0(cornwall_crime_data$Month, "-01"), format =
"%Y-%m-%d")



bristol_vehicle_crimes_by_month <- bristol_crime_data %>%
  filter(crime_type == "Vehicle crime") %>%
  filter(year(Month) == 2023) %>%
  group_by(Month) %>%
  summarise(count = n())

cornwall_vehicle_crimes_by_month <- cornwall_crime_data %>%
  filter(crime_type == "Vehicle crime") %>%
  filter(year(Month) == 2023) %>%
  group_by(Month) %>%
  summarise(count = n())

bristol_vehicle_crimes_by_month <- bristol_vehicle_crimes_by_month %>%
  mutate(County = "Bristol")

cornwall_vehicle_crimes_by_month <- cornwall_vehicle_crimes_by_month %>%
  mutate(County = "Cornwall")

combined_vehicle_crimes <- bind_rows(bristol_vehicle_crimes_by_month,
cornwall_vehicle_crimes_by_month)

~~~
~~~



```r
  {r}
main_folder <- "dataset/Crime"  # Replace with the actual path to your main folder

# Step 2: Get a list of all CSV files in the subfolders
csv_files <- list.files(path = main_folder, pattern = "\\.csv$", recursive = TRUE, full.names =
TRUE)

# Step 3: Read and combine all the CSV files
combined_data <- lapply(csv_files, read.csv) %>% bind_rows()

# Step 4: Save the combined data to a new CSV file
write.csv(combined_data, "dataset/combined_crime_data.csv", row.names = FALSE)



# Optional: Display the first few rows of the combined data to check the result
head(combined_data)



```
~~~



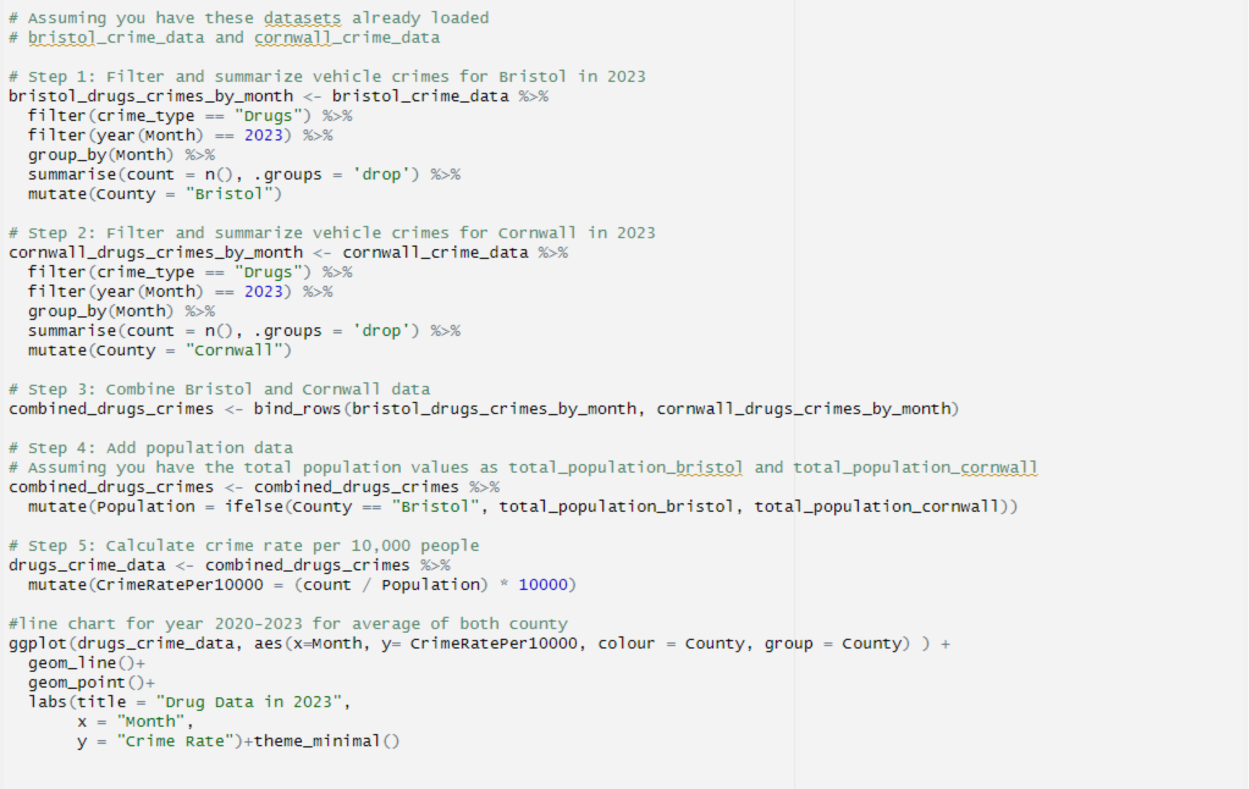
Description: df [6 × 12]

	Crime.ID <chr>	Month <chr>
1		2021-05
2	4d223210a924499f387156a1ef04bb09024b62bbe807c642c85eb5982e0dd7a0	2021-05
3	5f7d695d4c96d9c018618bfc72ff0157c9ff31e8452638ac6e918b6fc3c74d7d	2021-05
4	940708e1bc177f15670ebb12c8d4c0b2ae5bb89d9ee18ffd9a3e043af33a71ad	2021-05
5	af8ab8c04a1d6e60625f395d8f08869f8d13a7e89d0994f8f8e8c7da1747cfc5	2021-05
6	3c24eb30a0db70139f870c9a84f27a31c562cff2a8a0bf3ef16924df8b691068	2021-05

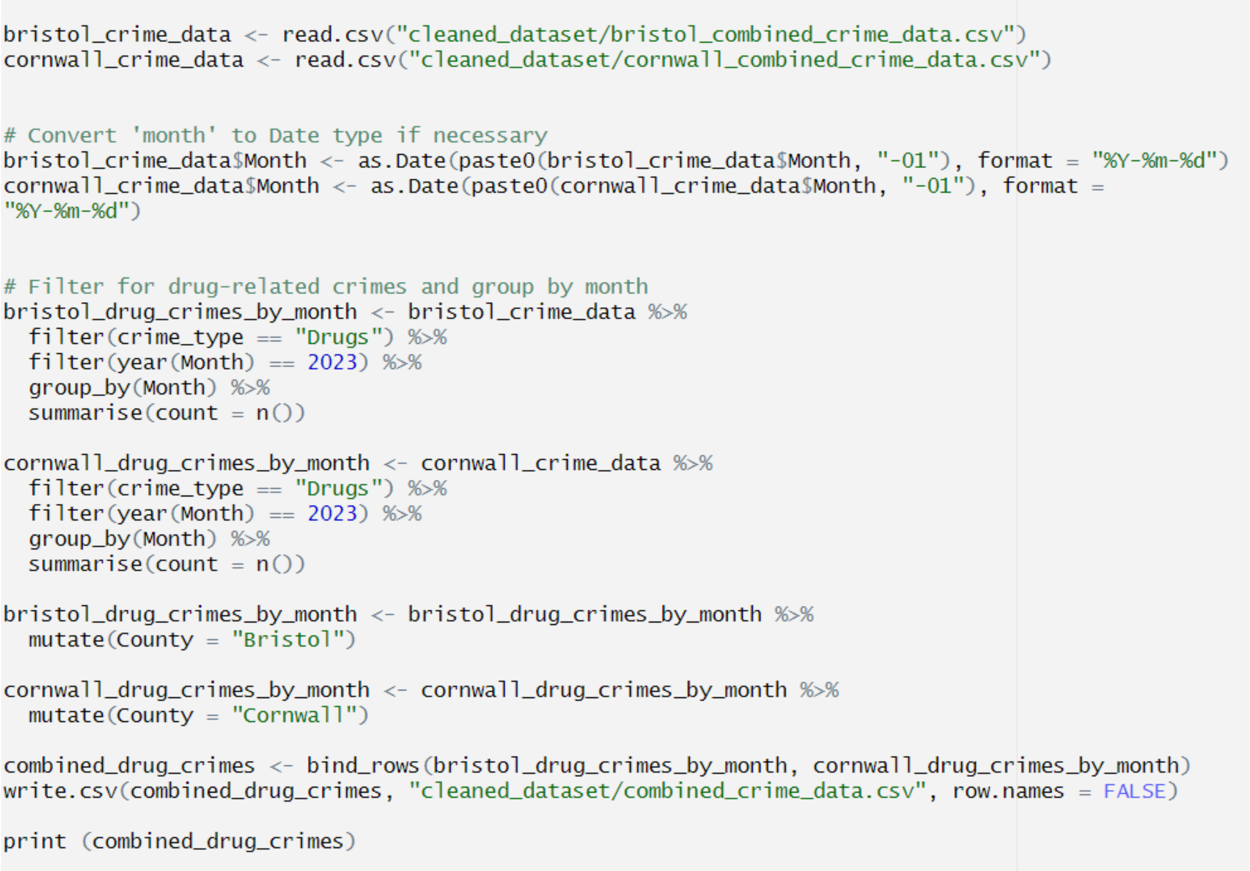
6 rows | 1-3 of 12 columns



~~~
```{r}

bristol_crime_data <- read.csv("cleaned_dataset/bristoltolsoa.csv")
cornwall_crime_data <- read.csv("cleaned_dataset/cornwalltolsoa.csv")

bristol_data <- bristol_crime_data %>% rename(LSOA.code = lsoa11cd)
cornwall_data <- cornwall_crime_data %>% rename(LSOA.code = lsoa11cd)

bristol_single_column_data <- bristol_data %>% select(LSOA.code)
cornwall_single_column_data <- cornwall_data %>% select(LSOA.code)

bristol_joined_data <- inner_join(bristol_single_column_data, combined_data, by = "LSOA.code") %>%
  select(LSOA.code, Month, Crime.type) %>%
  rename(crime_type=Crime.type)

cornwall_joined_data <- inner_join(cornwall_single_column_data, combined_data, by = "LSOA.code")
%>%
  select(LSOA.code, Month, Crime.type) %>%
  rename(crime_type=Crime.type)


write.csv(bristol_joined_data, "cleaned_dataset/bristol_combined_crime_data.csv", row.names =
FALSE)
write.csv(cornwall_joined_data, "cleaned_dataset/cornwall_combined_crime_data.csv", row.names =
FALSE)

```
~~~

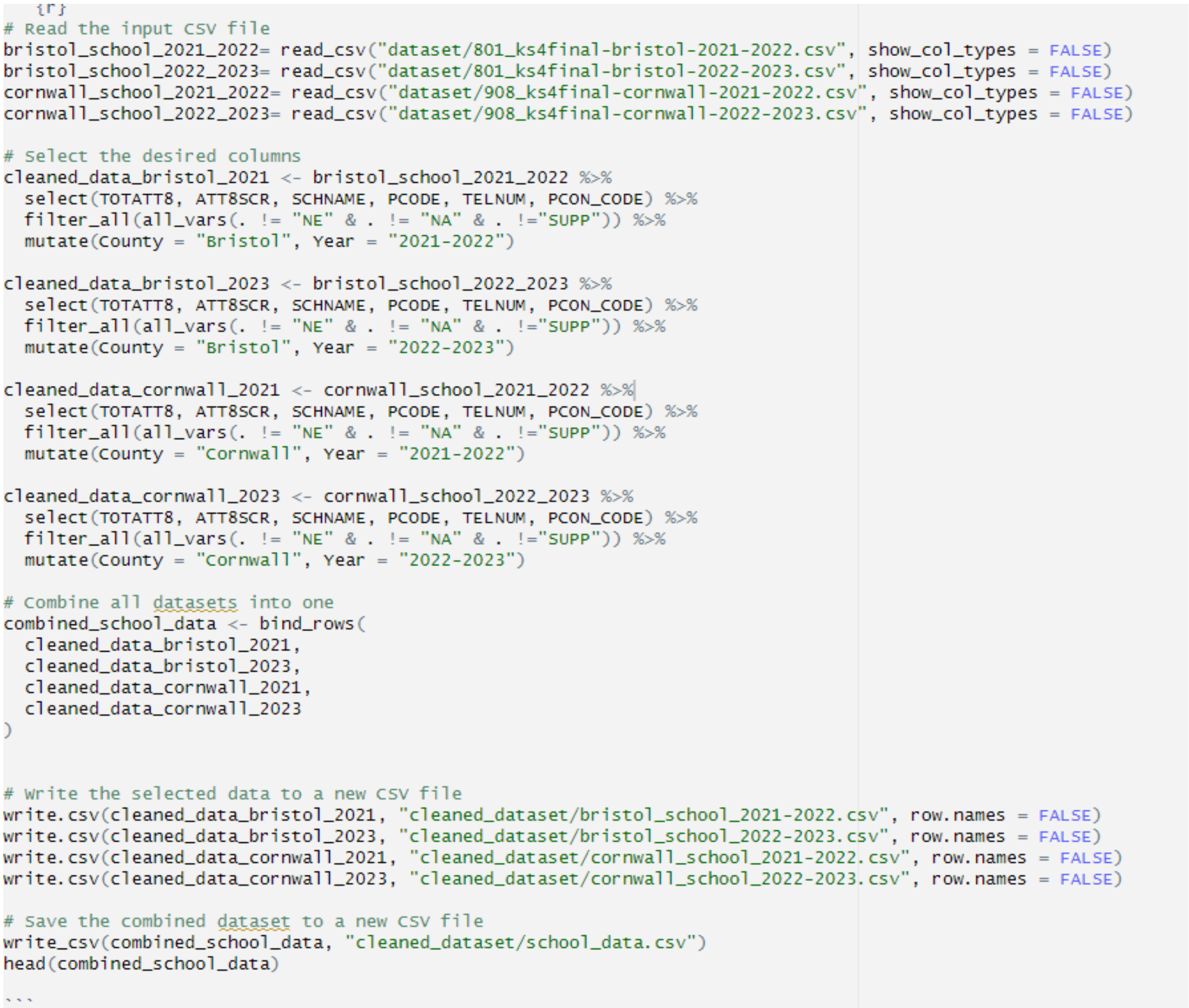


```
# Assuming you have these datasets already loaded
# bristol_crime_data and cornwall_crime_data

# Step 1: Filter and summarize vehicle crimes for Bristol in 2023
bristol_drugs_crimes_by_month <- bristol_crime_data %>%
  filter(crime_type == "Drugs") %>%
  filter(year(Month) == 2023) %>%
  group_by(Month) %>%
  summarise(count = n(), .groups = 'drop') %>%
  mutate(County = "Bristol")

# Step 2: Filter and summarize vehicle crimes for Cornwall in 2023
cornwall_drugs_crimes_by_month <- cornwall_crime_data %>%
  filter(crime_type == "Drugs") %>%
  filter(year(Month) == 2023) %>%
  group_by(Month) %>%
  summarise(count = n(), .groups = 'drop') %>%
  mutate(County = "Cornwall")

# Step 3: Combine Bristol and Cornwall data
combined_drugs_crimes <- bind_rows(bristol_drugs_crimes_by_month, cornwall_drugs_crimes_by_month)

# Step 4: Add population data
# Assuming you have the total population values as total_population_bristol and total_population_cornwall
combined_drugs_crimes <- combined_drugs_crimes %>%
  mutate(Population = ifelse(County == "Bristol", total_population_bristol, total_population_cornwall))

# Step 5: Calculate crime rate per 10,000 people
drugs_crime_data <- combined_drugs_crimes %>%
  mutate(CrimeRatePer10000 = (count / Population) * 10000)

#line chart for year 2020-2023 for average of both county
ggplot(drugs_crime_data, aes(x=Month, y= CrimeRatePer10000, colour = County, group = County) ) +
  geom_line()+
  geom_point()+
  labs(title = "Drug Data in 2023",
       x = "Month",
       y = "Crime Rate")+theme_minimal()
```

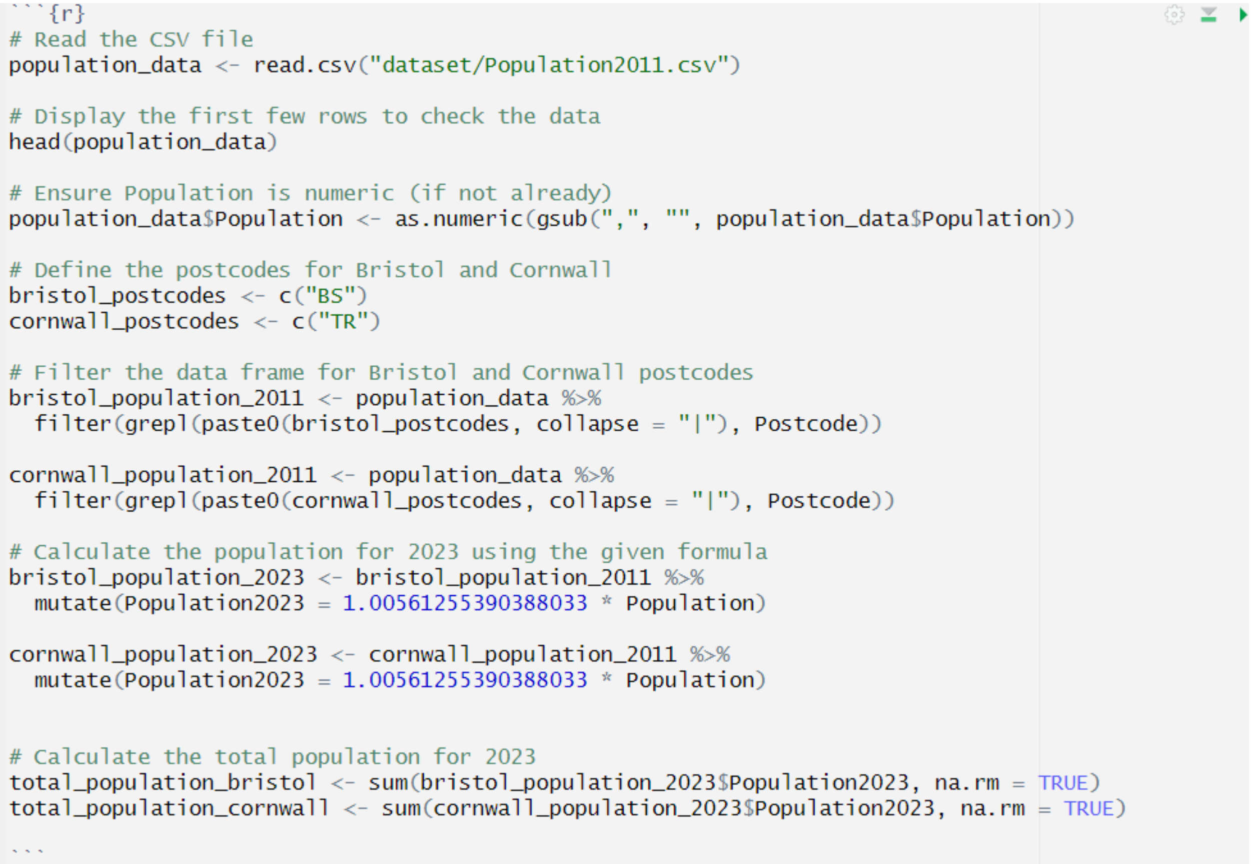


```
bristol_crime_data <- read.csv("cleaned_dataset/bristol_combined_crime_data.csv")
cornwall_crime_data <- read.csv("cleaned_dataset/cornwall_combined_crime_data.csv")


# Convert 'month' to Date type if necessary
bristol_crime_data$Month <- as.Date(paste0(bristol_crime_data$Month, "-01"), format = "%Y-%m-%d")
cornwall_crime_data$Month <- as.Date(paste0(cornwall_crime_data$Month, "-01"), format =
"%Y-%m-%d")


# Filter for drug-related crimes and group by month
bristol_drug_crimes_by_month <- bristol_crime_data %>%
  filter(crime_type == "Drugs") %>%
  filter(year(Month) == 2023) %>%
  group_by(Month) %>%
  summarise(count = n())

cornwall_drug_crimes_by_month <- cornwall_crime_data %>%
  filter(crime_type == "Drugs") %>%
  filter(year(Month) == 2023) %>%
  group_by(Month) %>%
  summarise(count = n())

bristol_drug_crimes_by_month <- bristol_drug_crimes_by_month %>%
  mutate(County = "Bristol")

cornwall_drug_crimes_by_month <- cornwall_drug_crimes_by_month %>%
  mutate(County = "Cornwall")

combined_drug_crimes <- bind_rows(bristol_drug_crimes_by_month, cornwall_drug_crimes_by_month)
write.csv(combined_drug_crimes, "cleaned_dataset/combined_crime_data.csv", row.names = FALSE)

print (combined_drug_crimes)
```

School



~~~
   {r}
# Read the input CSV file
bristol_school_2021_2022= read_csv("dataset/801_ks4final-bristol-2021-2022.csv", show_col_types = FALSE)
bristol_school_2022_2023= read_csv("dataset/801_ks4final-bristol-2022-2023.csv", show_col_types = FALSE)
cornwall_school_2021_2022= read_csv("dataset/908_ks4final-cornwall-2021-2022.csv", show_col_types = FALSE)
cornwall_school_2022_2023= read_csv("dataset/908_ks4final-cornwall-2022-2023.csv", show_col_types = FALSE)

# Select the desired columns
cleaned_data_bristol_2021 <- bristol_school_2021_2022 %>%
  select(TOTATT8, ATT8SCR, SCHNAME, PCODE, TELNUM, PCON_CODE) %>%
  filter_all(all_vars(. != "NE" & . != "NA" & . !="SUPP")) %>%
  mutate(County = "Bristol", Year = "2021-2022")

cleaned_data_bristol_2023 <- bristol_school_2022_2023 %>%
  select(TOTATT8, ATT8SCR, SCHNAME, PCODE, TELNUM, PCON_CODE) %>%
  filter_all(all_vars(. != "NE" & . != "NA" & . !="SUPP")) %>%
  mutate(County = "Bristol", Year = "2022-2023")

cleaned_data_cornwall_2021 <- cornwall_school_2021_2022 %>%
  select(TOTATT8, ATT8SCR, SCHNAME, PCODE, TELNUM, PCON_CODE) %>%
  filter_all(all_vars(. != "NE" & . != "NA" & . !="SUPP")) %>%
  mutate(County = "Cornwall", Year = "2021-2022")

cleaned_data_cornwall_2023 <- cornwall_school_2022_2023 %>%
  select(TOTATT8, ATT8SCR, SCHNAME, PCODE, TELNUM, PCON_CODE) %>%
  filter_all(all_vars(. != "NE" & . != "NA" & . !="SUPP")) %>%
  mutate(County = "Cornwall", Year = "2022-2023")

# Combine all datasets into one
combined_school_data <- bind_rows(
  cleaned_data_bristol_2021,
  cleaned_data_bristol_2023,
  cleaned_data_cornwall_2021,
  cleaned_data_cornwall_2023
)


# Write the selected data to a new CSV file
write.csv(cleaned_data_bristol_2021, "cleaned_dataset/bristol_school_2021-2022.csv", row.names = FALSE)
write.csv(cleaned_data_bristol_2023, "cleaned_dataset/bristol_school_2022-2023.csv", row.names = FALSE)
write.csv(cleaned_data_cornwall_2021, "cleaned_dataset/cornwall_school_2021-2022.csv", row.names = FALSE)
write.csv(cleaned_data_cornwall_2023, "cleaned_dataset/cornwall_school_2022-2023.csv", row.names = FALSE)

# Save the combined dataset to a new CSV file
write_csv(combined_school_data, "cleaned_dataset/school_data.csv")
head(combined_school_data)

```
~~~

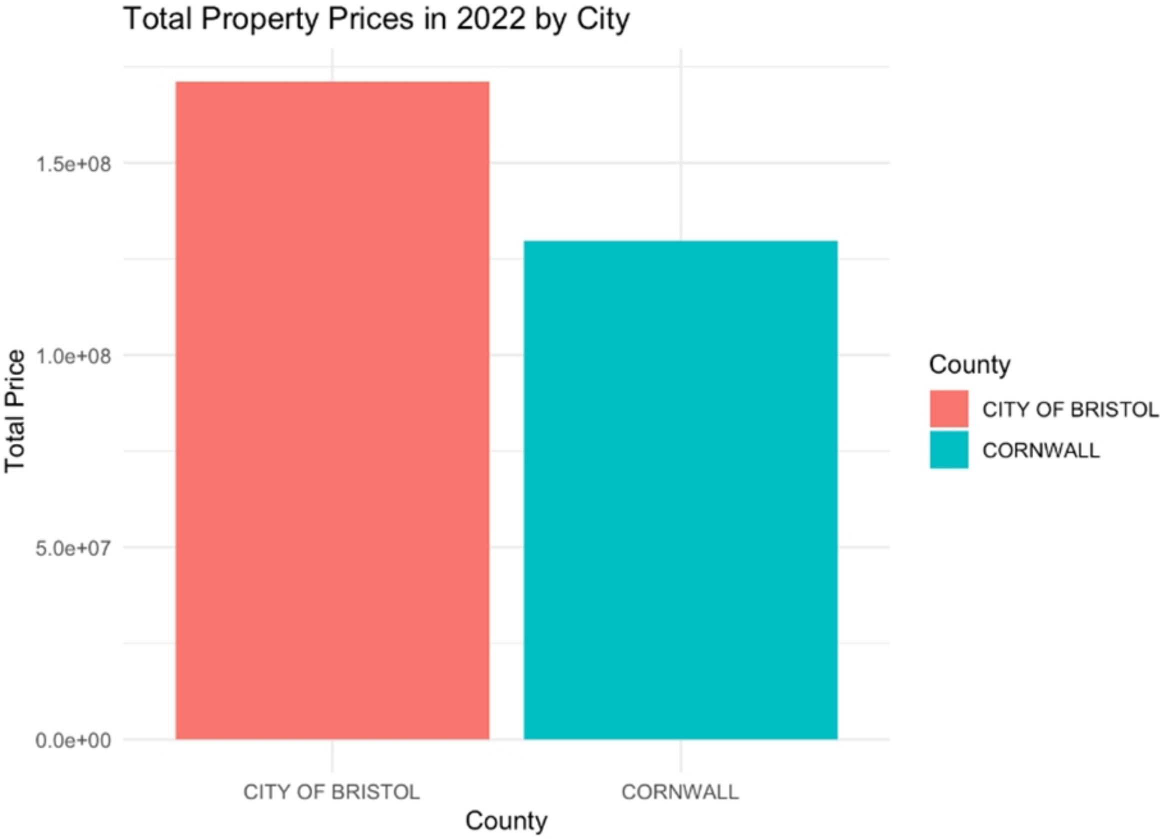


~~~
```{r}
# Read the CSV file
population_data <- read.csv("dataset/Population2011.csv")

# Display the first few rows to check the data
head(population_data)

# Ensure Population is numeric (if not already)
population_data$Population <- as.numeric(gsub(",", "", population_data$Population))

# Define the postcodes for Bristol and Cornwall
bristol_postcodes <- c("BS")
cornwall_postcodes <- c("TR")

# Filter the data frame for Bristol and Cornwall postcodes
bristol_population_2011 <- population_data %>%
  filter(grepl(paste0(bristol_postcodes, collapse = "|"), Postcode))

cornwall_population_2011 <- population_data %>%
  filter(grepl(paste0(cornwall_postcodes, collapse = "|"), Postcode))

# Calculate the population for 2023 using the given formula
bristol_population_2023 <- bristol_population_2011 %>%
  mutate(Population2023 = 1.00561255390388033 * Population)

cornwall_population_2023 <- cornwall_population_2011 %>%
  mutate(Population2023 = 1.00561255390388033 * Population)


# Calculate the total population for 2023
total_population_bristol <- sum(bristol_population_2023$Population2023, na.rm = TRUE)
total_population_cornwall <- sum(cornwall_population_2023$Population2023, na.rm = TRUE)

```
~~~

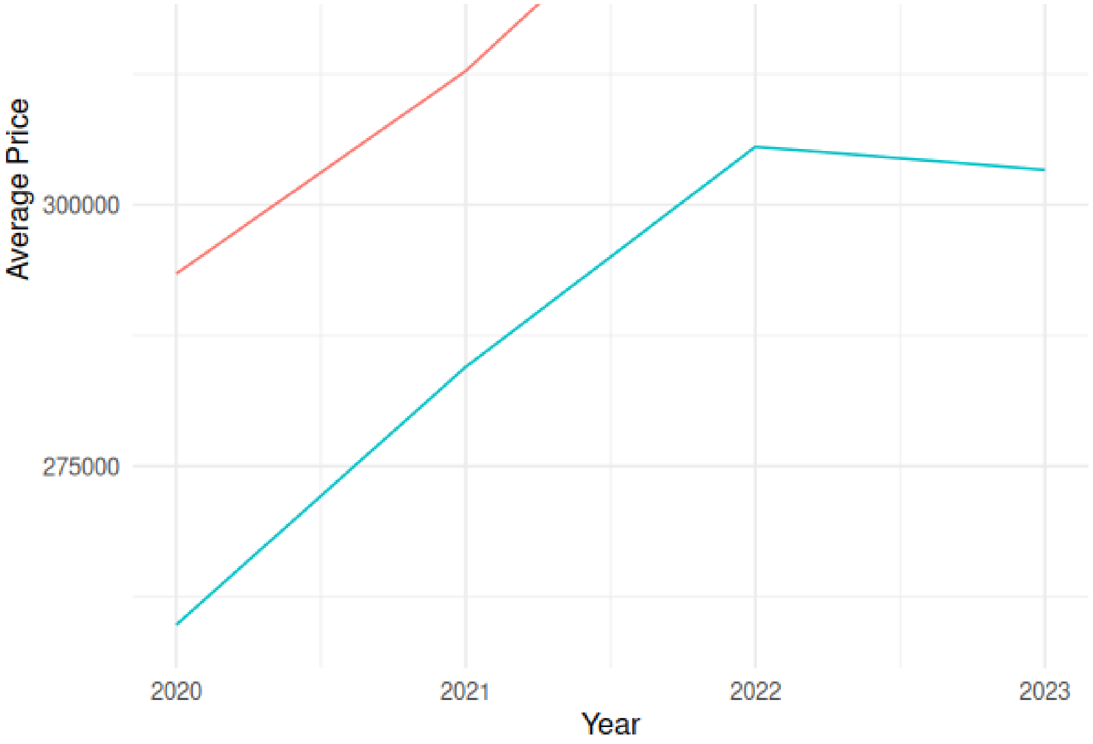
Exploratory Data Analysis

House Pricing

Figure 1: house price



Fig



Board Band Speeds

Figure 3: average download speed

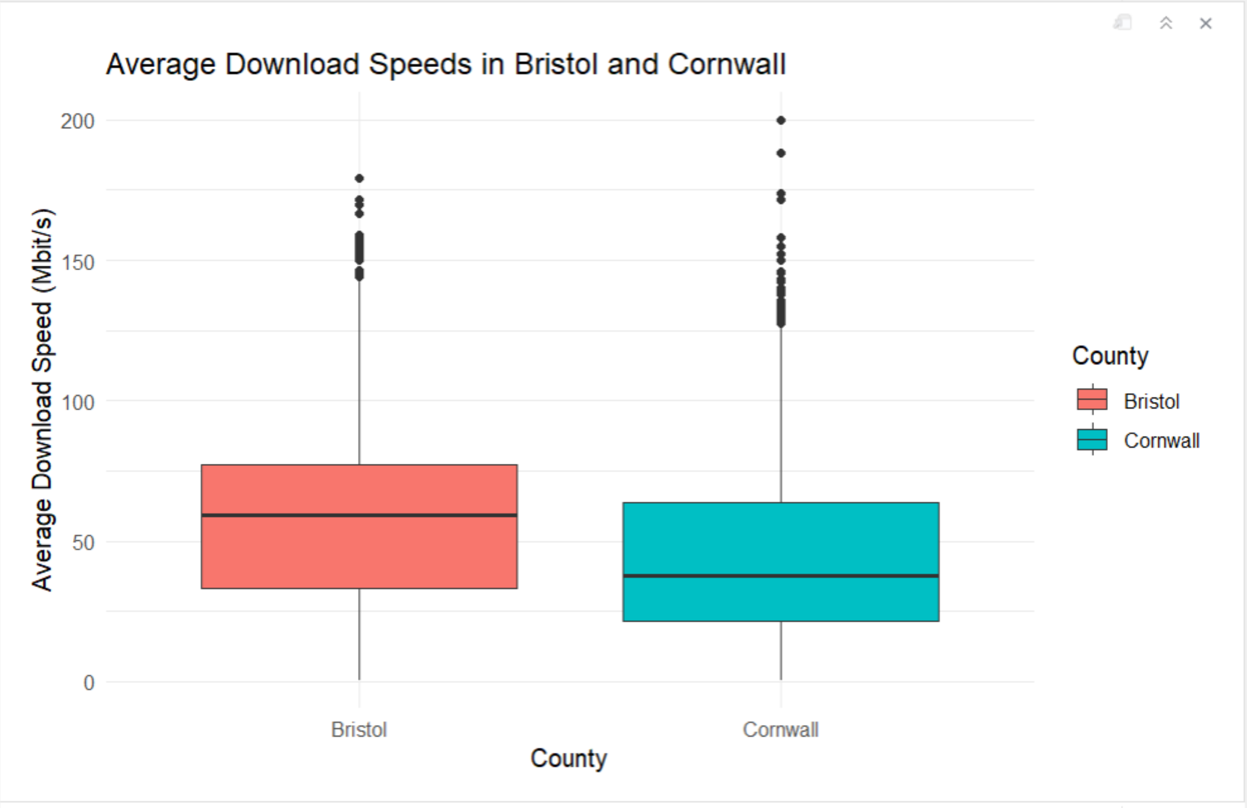


Figure 4: average and maximum download speed

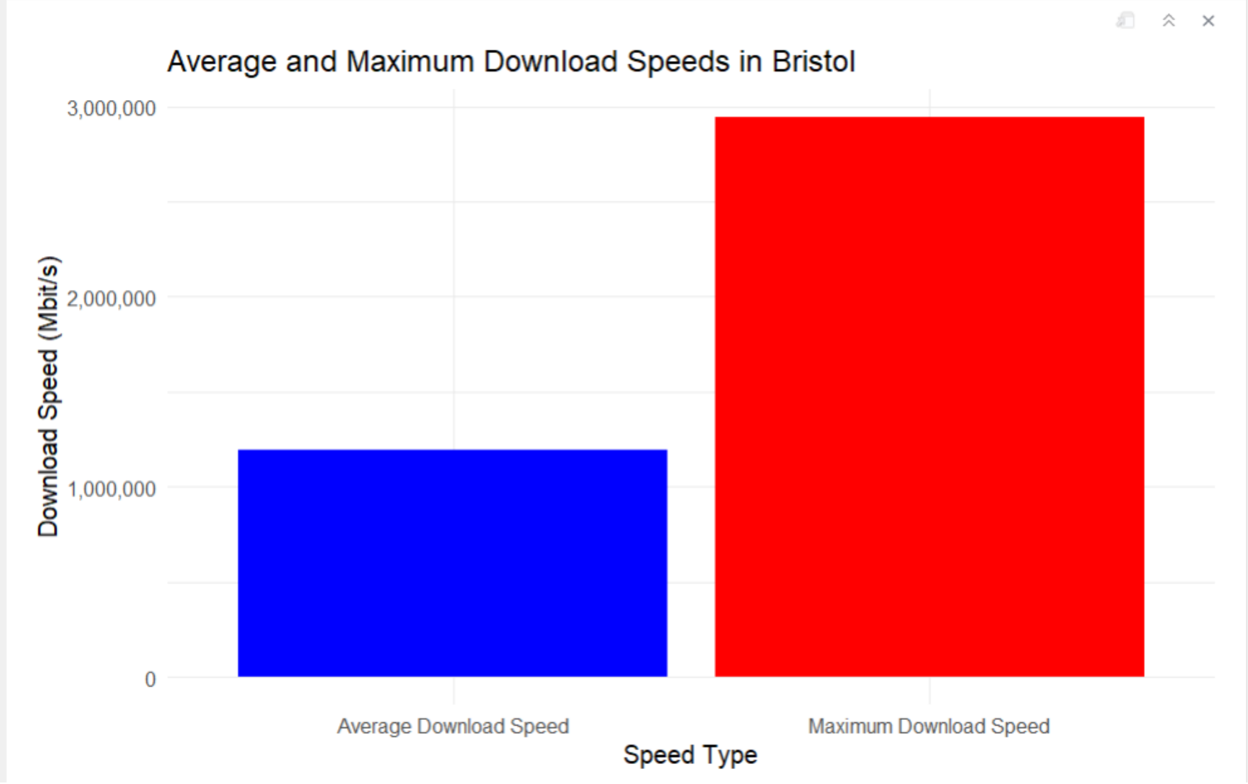
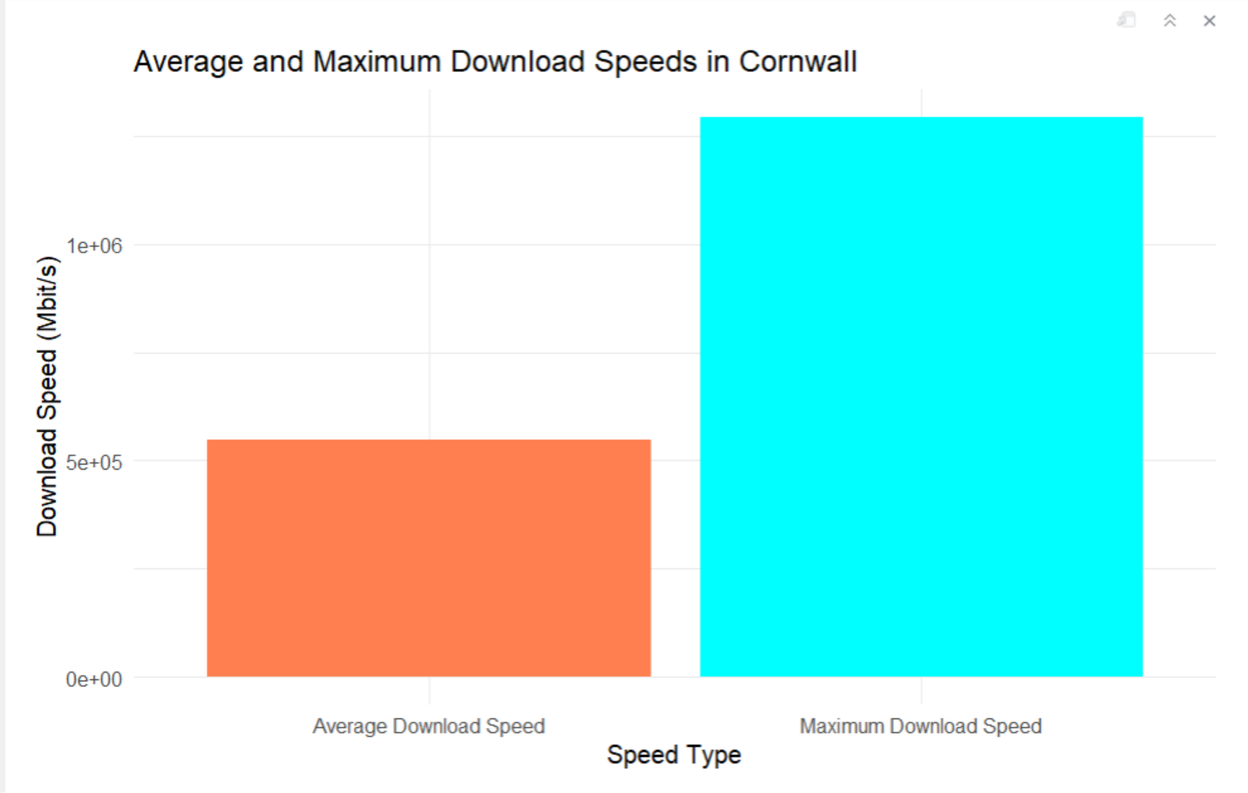


Figure 5: average and maximum download speed in cornwall



Crime

Figure 6: drug offence rate in both counties

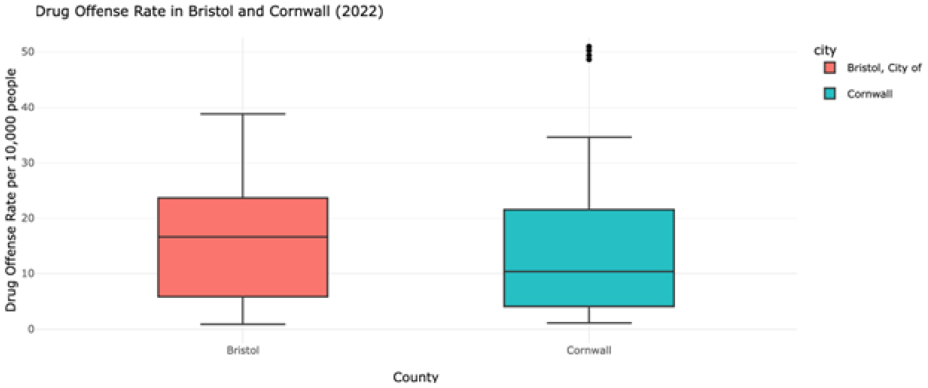
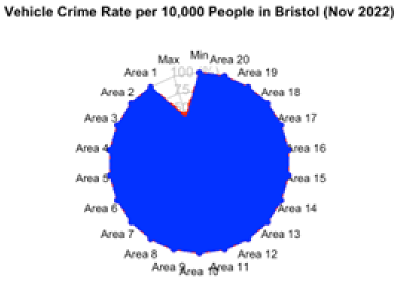


Figure 7: vehicle Crime Rate per 10000 people



School

Figure 10: average Attainment 8 score in the year 2021-2022 academic year

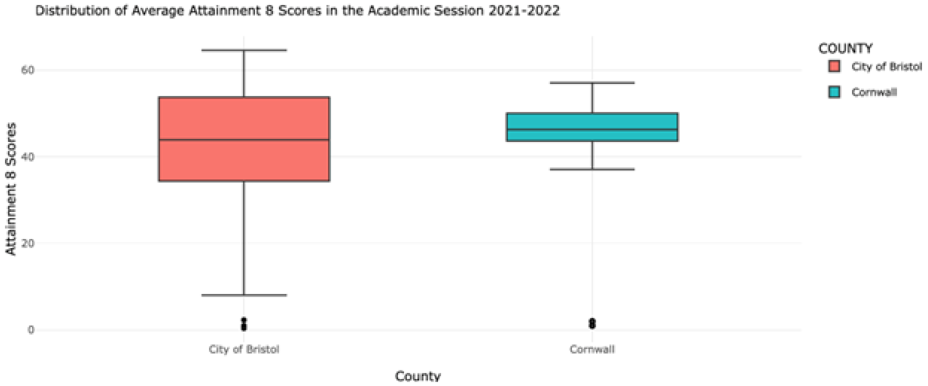


Figure 11: bristol average attainment 8 score in academic year 2021-2022

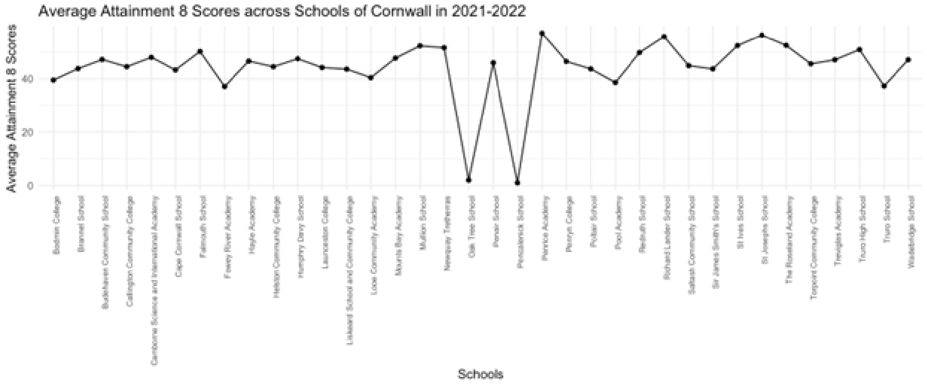
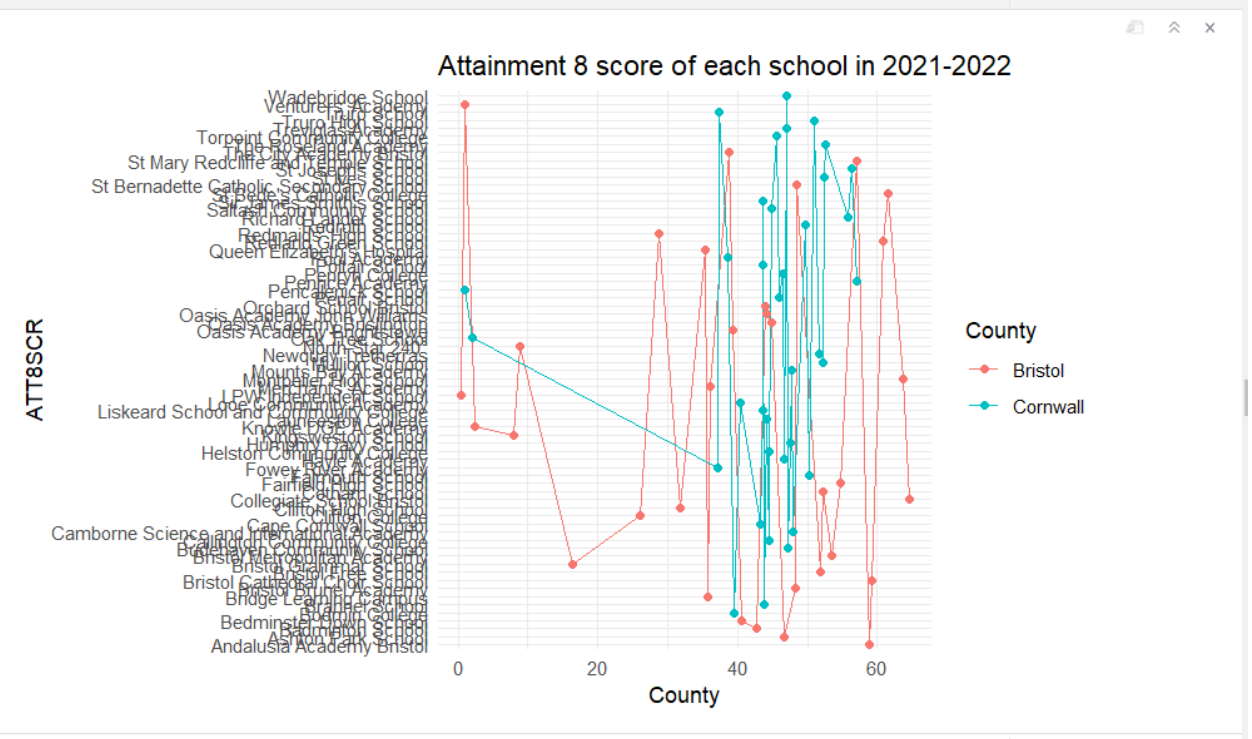


Figure 12: cornwall average Attainment 8 score



Linear Modelling

Figure 13: house price vs download speed in cornwall

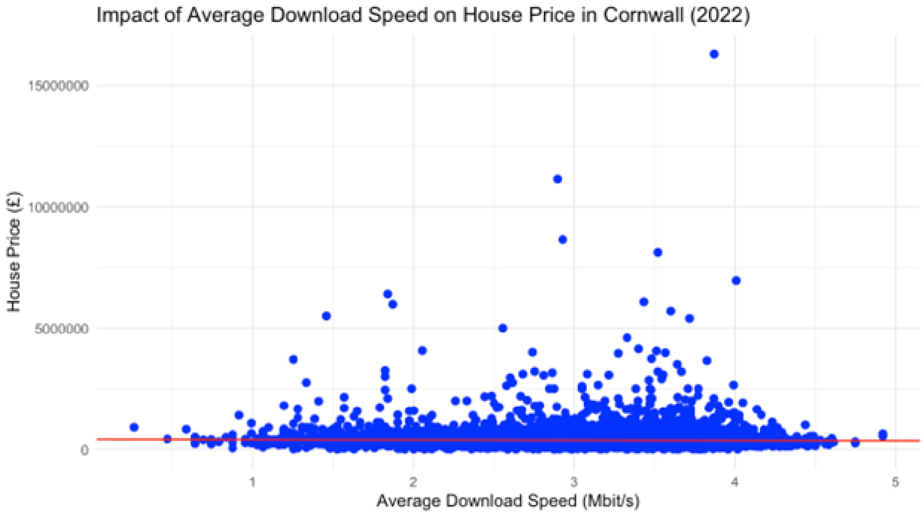
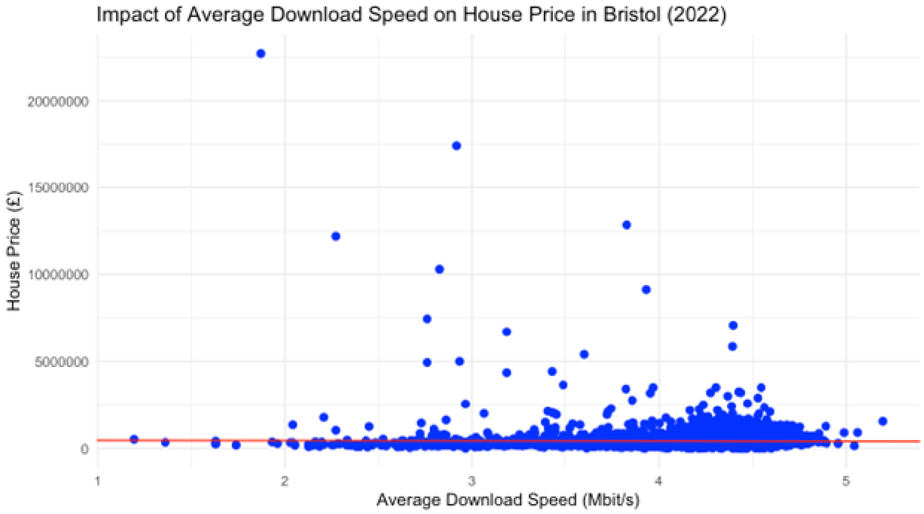


Figure 14: house price vs download speed in bristol



The scatter figure displays the correlation between Bristol's Attainment 8 scores for the year 2022 and the average download speed, expressed in megabits per second. The Attainment 8 scores, which range from 0 to 60, are displayed on the y-axis, while the x-axis indicates the average download speed, which ranges from roughly 3.0 to 4.6 Mbit/s. A number of blue data points on the plot show the download speed vs the attainment score for various Bristol entities or locations. A linear regression is shown by a red trend line that descends diagonally and shows a negative correlation between the two variables. This implies that the Attainment 8 scores tend to decline with increasing average download speed.

Figure 15: house price vs Drug Rate in cornwall

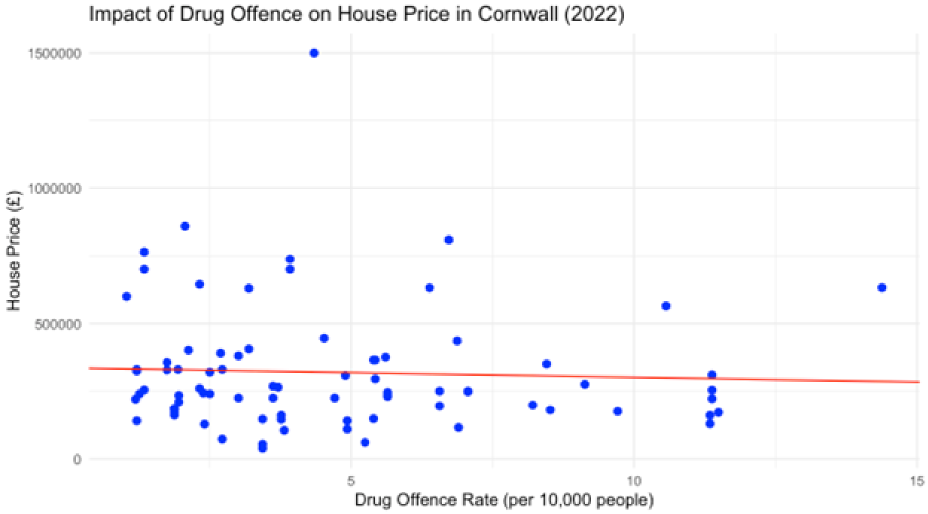
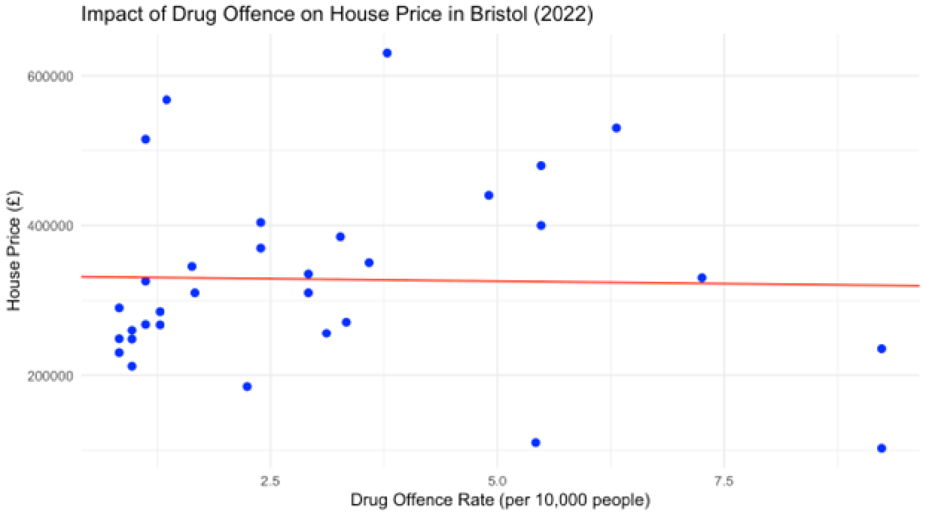


Figure 16: house price vs Drug Rate in bristol



The association between Bristol's average download speeds and house prices in 2022 is depicted in the scatter plot. The y-axis displays housing prices, which span from 0 to more than 20 million, while the x-axis displays the average download speed, which is roughly between 1.0 and 5.0 Mbit/s. A number of blue data points, representing the cost of dwelling in various districts of Bristol in relation to download speed, populate the figure. The red trend line, which virtually horizontally crosses the graph, indicates a very weak or nonexistent correlation between download speed and property prices despite the large number of data points. The bulk of real estate values are centered in the lower levels, particularly between 0 and 5 million, with a tiny percentage of extraordinarily high values.

Figure 17: attainment 8 score vs House Price in cornwall

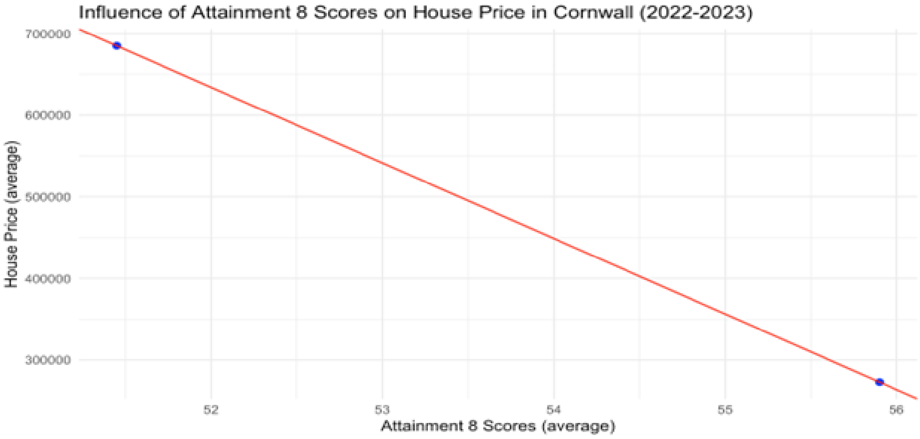
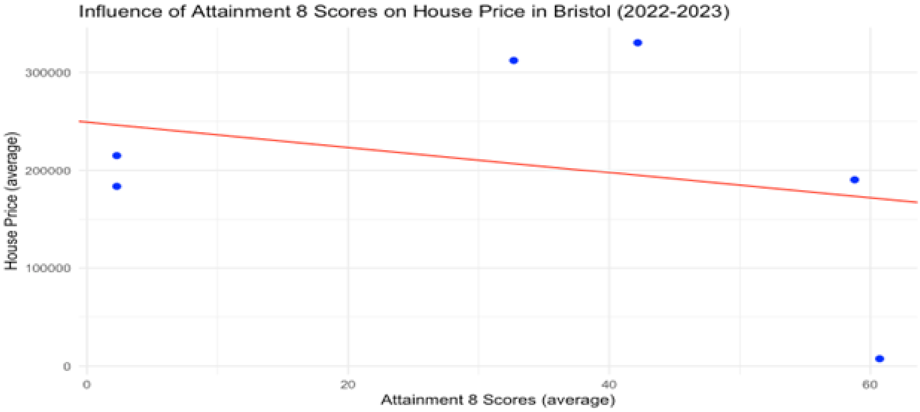
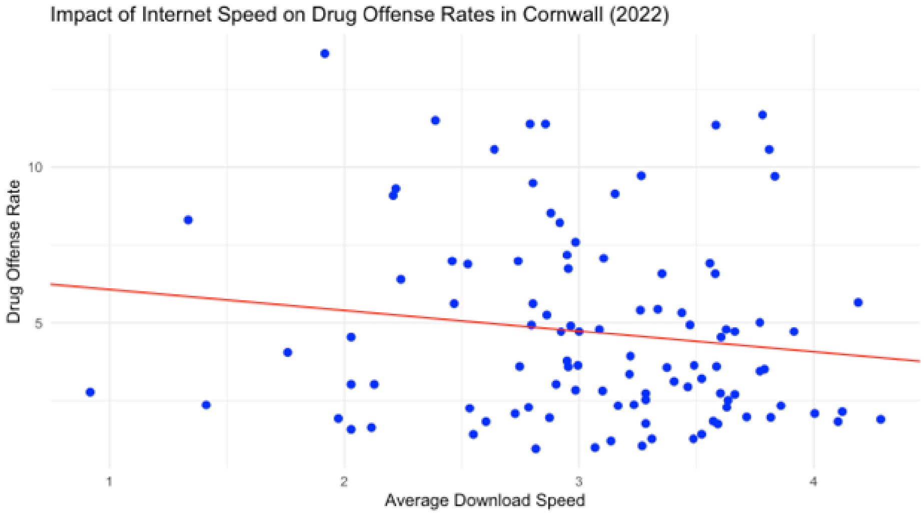


Figure 18: attainment 8 score vs House Price in bristol



The scatter plot shows the correlation between Bristol's drug offense rate and home prices in 2022. The drug offense rate is represented by the x-axis, which ranges from roughly 1 to 9, while house prices are represented by the y-axis, which ranges from 100,000 to 600,000. Every blue dot on the plot represents a data point that illustrates the correlation between the rate of drug offenses and the cost of homes in various Bristol neighborhoods. The nearly horizontal red trend line shows that there is extremely little or no association between these two variables. Given that the correlation coefficient is -0.026, it is confirmed that the relationship between the rate of drug offenses and property prices is not nearly linear.

Figure 19: average Download Speed vs Drug Offense Rate in cornwall



The graph shows the relationship between achievement 8 scores and housing prices in Bristol, UK in 2022. The data points are represented by the blue dots, and the line of greatest fit is shown by the red line. The negative correlation of the line indicates a relationship between lower property prices and higher achievement 8 scores. The data point with the highest accomplishment 8 score is also the one with the lowest dwelling price. The data point with the lowest attainment 8 score is also the one with the highest property price. A data point with a house price of 207,000 pounds and an approximate attainment 8 score of 33 is found.

Figure 20: attainment 8 score vs Drug Offense Rate

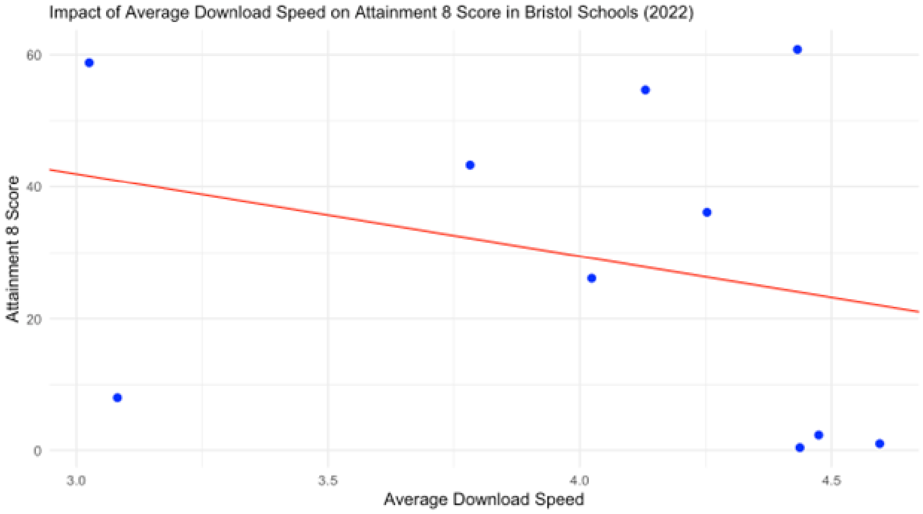
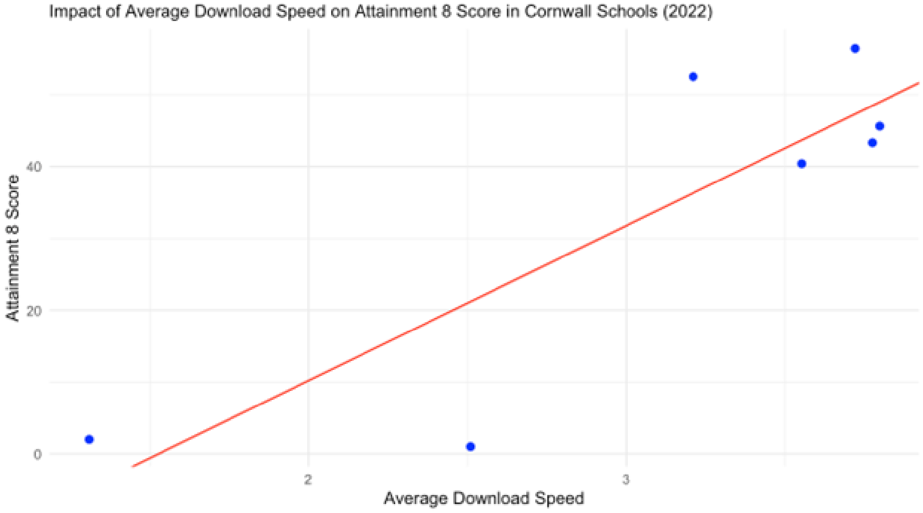


Figure 21: attainment 8 score vs Drug Offense Rate



The graph shows the relationship between Attainment 8 scores and housing prices in Bristol, 2022. Based on the data points that have been shown on the graph, a linear regression line has been created to depict the trend. The regression line shows that there is a negative correlation between Attainment 8 scores and real estate prices. This suggests that when Attainment 8 scores increase, property values frequently decrease. The following is the equation for the regression line: 242.7142 - 1.2902 * Average Attainment 8 Scores = Average House Price. Based on statistical data, the average house price in Bristol was expected to be approximately 206,906 pounds in 2022, when the Attainment 8 score was 32.7. Keep in mind that this does not imply a cause-and-effect relationship; rather, it is merely a correlation.

Legal and ethical isssue

Data Privacy: The use of sensitive data, such as drug offense rates and Attainment 8 scores, raises concerns about data privacy and the potential for identifying particular companies or locations.

Discrimination: Given the correlation between Attainment 8 scores and property prices, concerns of discrimination against particular areas or groups may arise, perhaps exacerbating already-existing social and economic divides.

Data Bias: If the analysis is conducted using biased data, the conclusions may be inaccurate. For example, Bristol's academic achievement may not be accurately represented by the Attainment 8 scores overall.

Accountability and openness: Using data analytics and machine learning models in decision-making raises concerns about openness and accountability.

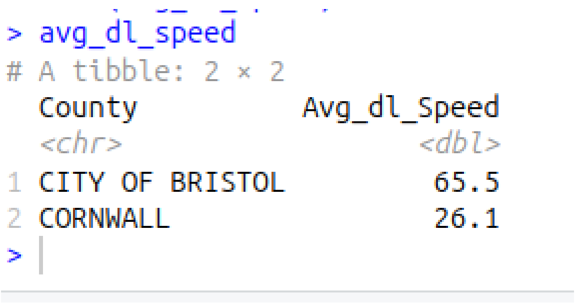
Conclusion

The analysis of the visualizations provides numerous significant insights into the relationships between the various factors impacting Bristol, UK, house prices in 2022. First off, a negative correlation between average download speed and Attainment 8 scores suggests that regions with better educational attainment are probably going to have slower internet speeds. This study could have an impact on lawmakers and educators since it highlights how important it is to close the digital divide and ensure that all students have fair access to digital infrastructure and high-quality education. Second, it appears that internet speed has no effect on Bristol real estate values based on the weak link between download speed and real estate values. Analyzing the images yields important new insights into how the various factors affecting Bristol, UK, real estate values in 2022 interact with one another. First off, the negative correlation between average download speed and Attainment 8 scores suggests that areas with better educational attainment also tend to have slower internet connections. This research highlights the need to close the digital divide and ensure that all students have fair access to digital infrastructure and high-quality education. It may have implications for policymakers and educators. Secondly, the little link shown between download speed and property values implies that internet speed is not a major factor influencing Bristol house prices.

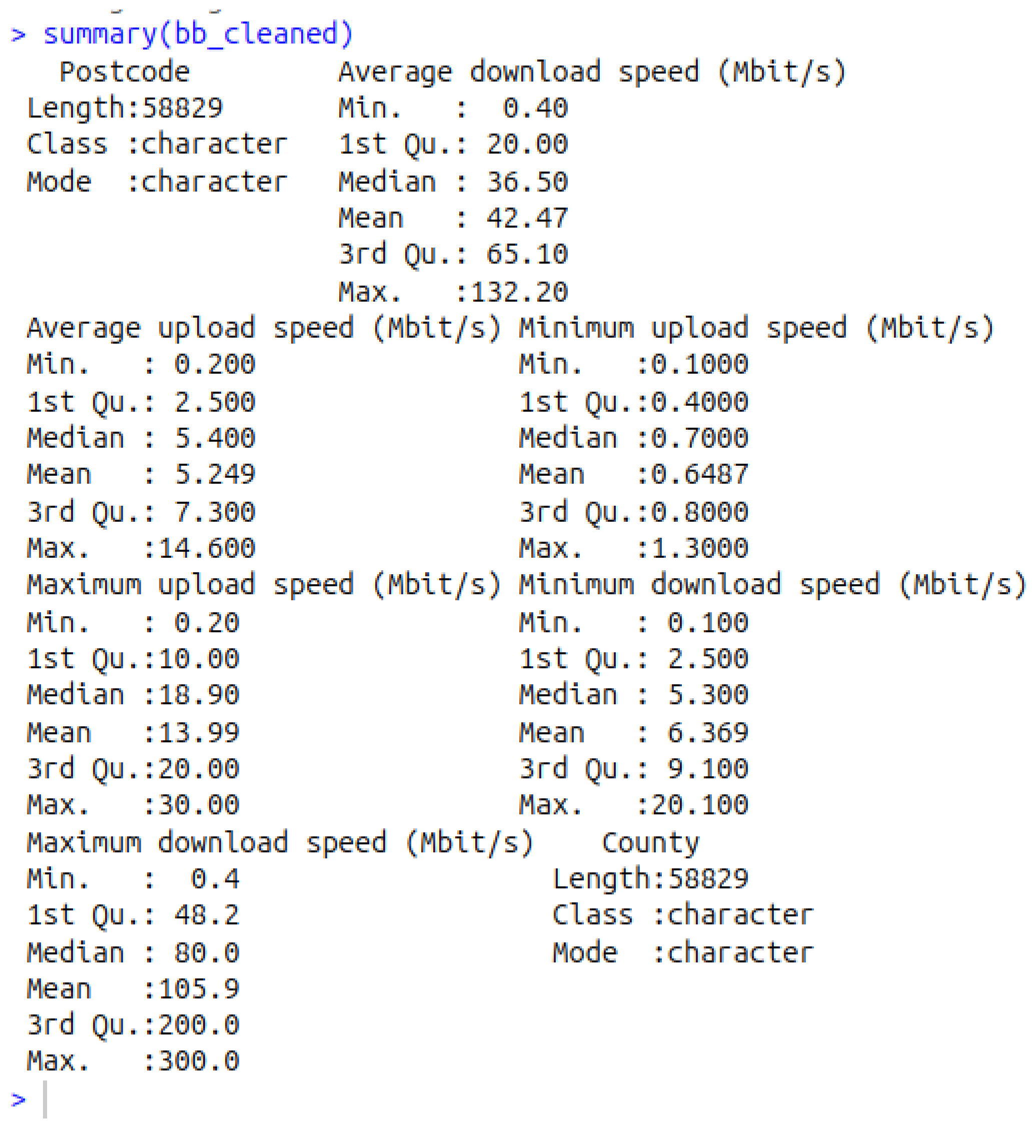
References

Data Science Ethics: "Ethics in Data Science" by Mike Loukides, Hilary Mason, and DJ Patil (O'Reilly Media, 2018) Data Privacy: "Data Privacy: A Practical Guide" by Paul Breitbarth and Allan Castle (Springer, 2020) Discrimination and Bias: "Fairness and Machine Learning" by Solon Barocas and Moritz Hardt (fairmlbook.org, 2019) Transparency and Accountability: "Transparent and Accountable AI" by the European Union's High-Level Expert Group on Artificial Intelligence (europa.eu, 2019) ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham: A comprehensive guide to data visualization with ggplot2, covering topics such as data visualization principles, ggplot2 syntax, and customization. Data Visualization: A Handbook for Data Driven Design by Andy Kirk: A comprehensive guide to data visualization, covering topics such as data preparation, visualization principles, and visualization tools. Visualize This: The FlowingData Guide to Design, Visualization, and Statistics by Nathan Yau: A practical guide to data visualization, covering topics such as data preparation, visualization principles, and visualization tools.

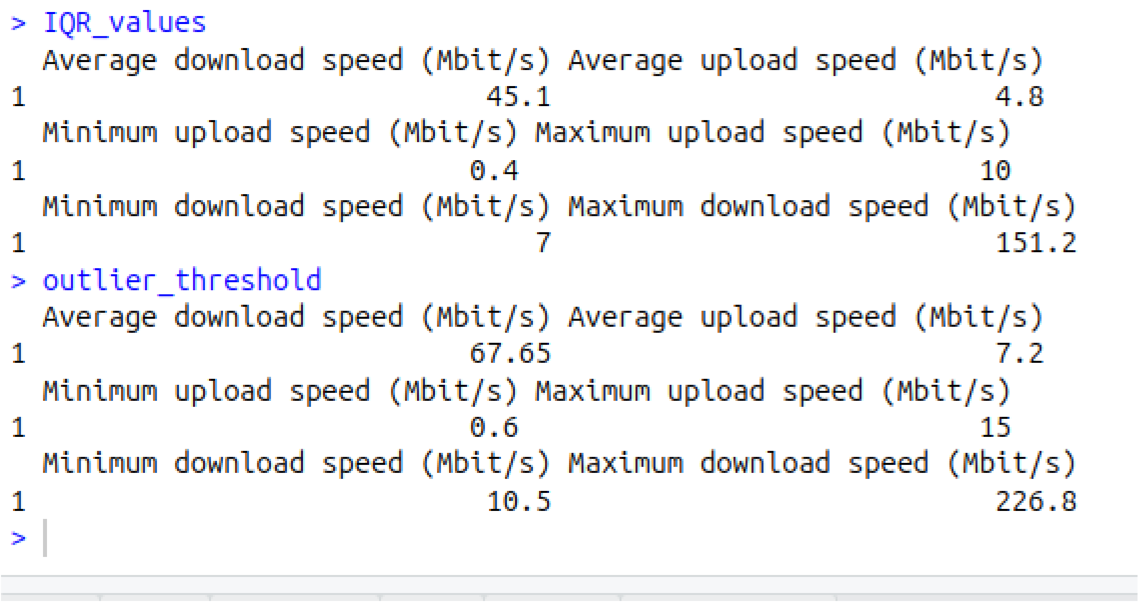
Appendix



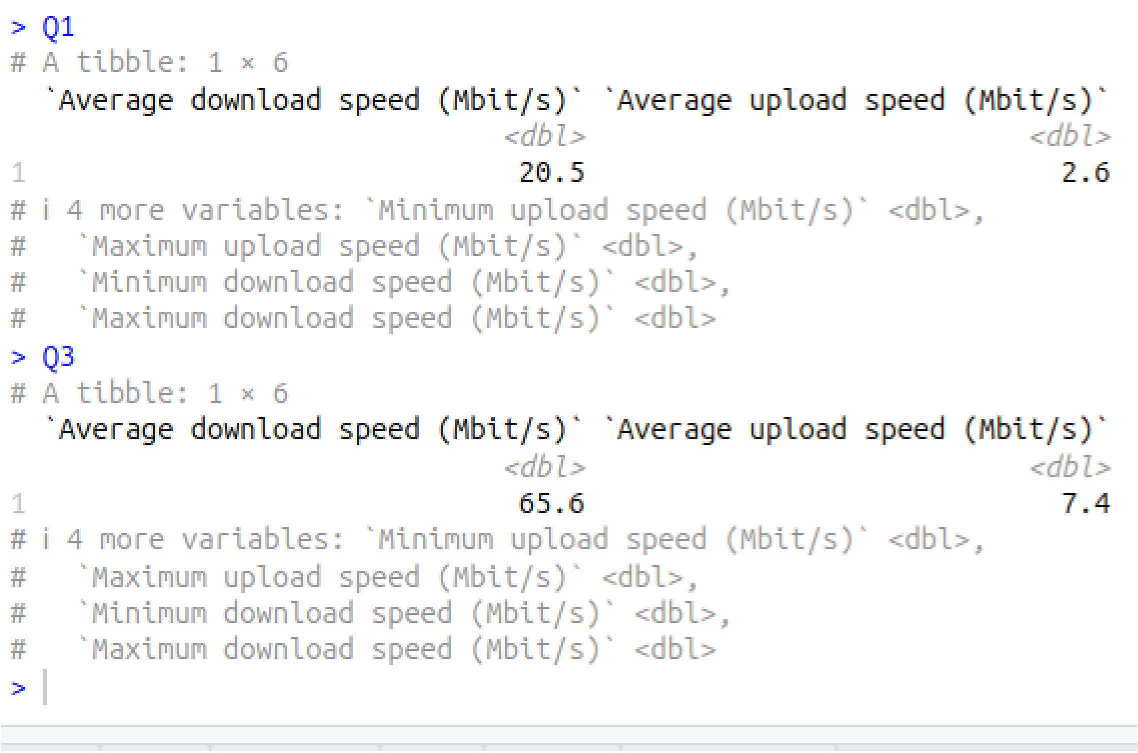
```
> avg_dl_speed
# A tibble: 2 × 2
  County          Avg_dl_Speed
  <chr>                  <dbl>
1 CITY OF BRISTOL         65.5
2 CORNWALL                26.1
> 
```



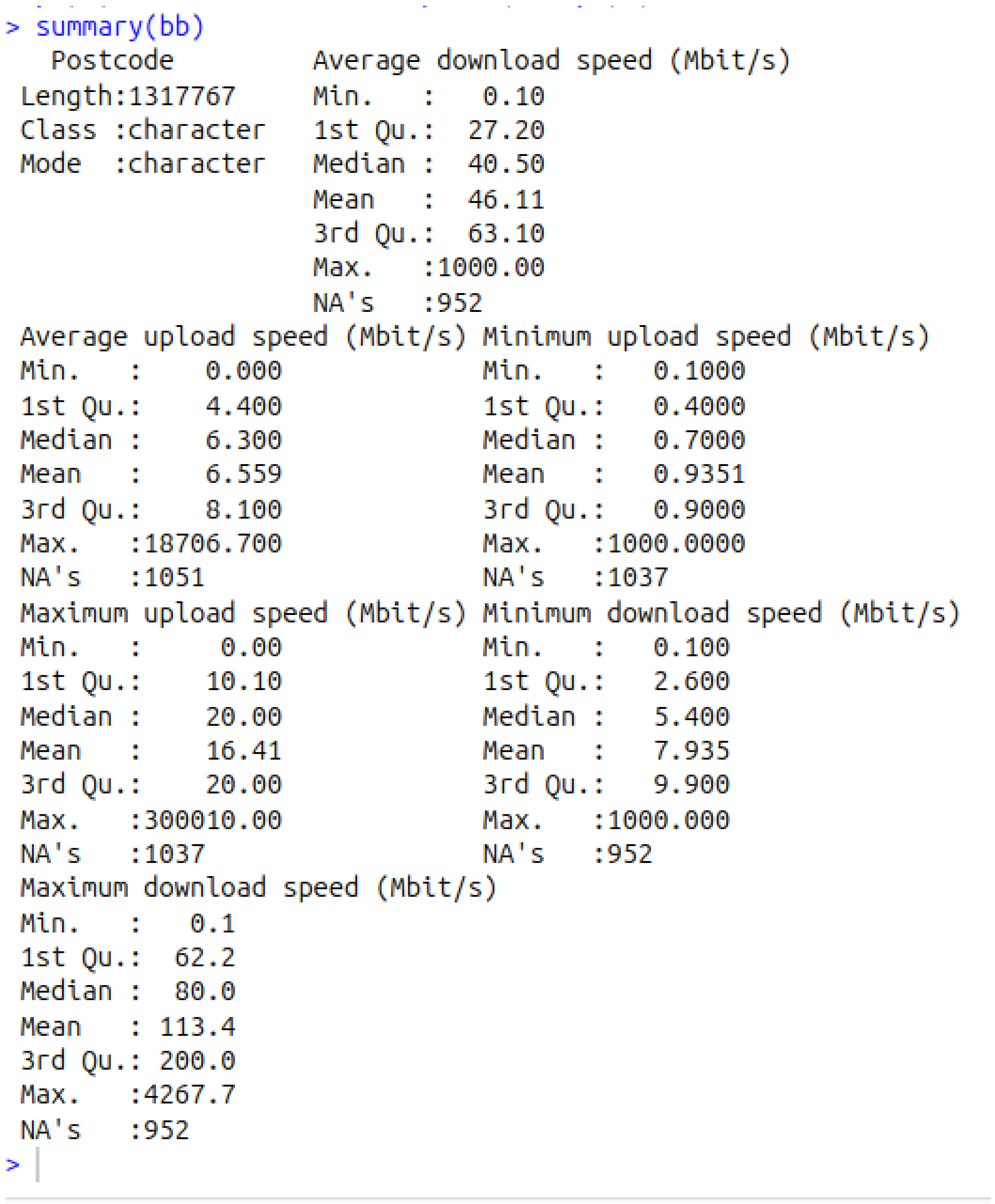
```
> summary(bb_cleaned)
   Postcode          Average download speed (Mbit/s)
 Length:58829       Min.   :  0.40                 
 Class :character   1st Qu.: 20.00                 
 Mode  :character   Median : 36.50                 
                    Mean   : 42.47                 
                    3rd Qu.: 65.10                 
                    Max.   :132.20                 
 Average upload speed (Mbit/s) Minimum upload speed (Mbit/s)
 Min.   : 0.200                Min.   :0.1000               
 1st Qu.: 2.500                1st Qu.:0.4000               
 Median : 5.400                Median :0.7000               
 Mean   : 5.249                Mean   :0.6487               
 3rd Qu.: 7.300                3rd Qu.:0.8000               
 Max.   :14.600                Max.   :1.3000               
 Maximum upload speed (Mbit/s) Minimum download speed (Mbit/s)
 Min.   : 0.20                 Min.   : 0.100                 
 1st Qu.:10.00                 1st Qu.: 2.500                 
 Median :18.90                 Median : 5.300                 
 Mean   :13.99                 Mean   : 6.369                 
 3rd Qu.:20.00                 3rd Qu.: 9.100                 
 Max.   :30.00                 Max.   :20.100                 
 Maximum download speed (Mbit/s)    County         
 Min.   :  0.4                   Length:58829      
 1st Qu.: 48.2                   Class :character  
 Median : 80.0                   Mode  :character  
 Mean   :105.9                                     
 3rd Qu.:200.0                                     
 Max.   :300.0                                     
> 
```



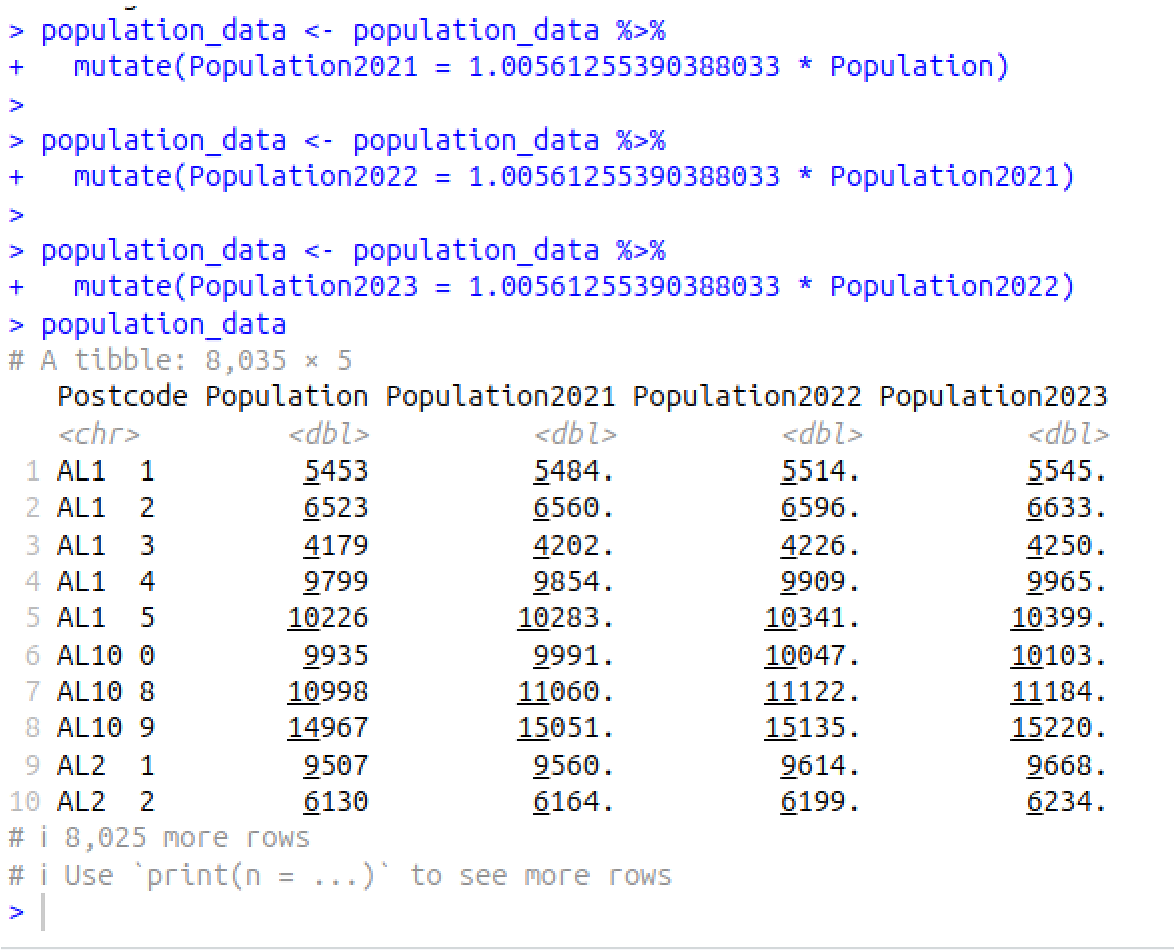
```
> IQR_values
  Average download speed (Mbit/s) Average upload speed (Mbit/s)
1                            45.1                           4.8
  Minimum upload speed (Mbit/s) Maximum upload speed (Mbit/s)
1                           0.4                            10
  Minimum download speed (Mbit/s) Maximum download speed (Mbit/s)
1                               7                           151.2
> outlier_threshold
  Average download speed (Mbit/s) Average upload speed (Mbit/s)
1                           67.65                           7.2
  Minimum upload speed (Mbit/s) Maximum upload speed (Mbit/s)
1                           0.6                            15
  Minimum download speed (Mbit/s) Maximum download speed (Mbit/s)
1                            10.5                           226.8
> 
```



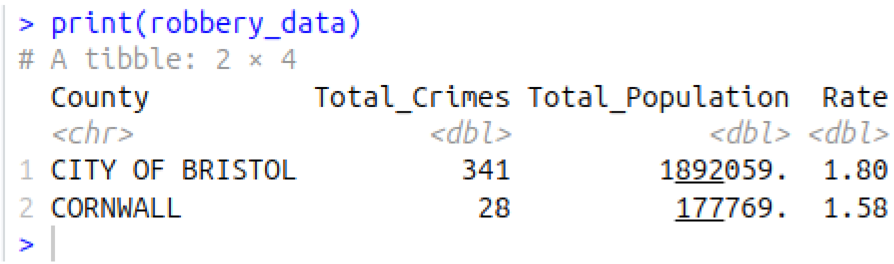
```
> Q1
# A tibble: 1 × 6
  `Average download speed (Mbit/s)` `Average upload speed (Mbit/s)`
                              <dbl>                            <dbl>
1                              20.5                              2.6
# ℹ 4 more variables: `Minimum upload speed (Mbit/s)` <dbl>,
#   `Maximum upload speed (Mbit/s)` <dbl>,
#   `Minimum download speed (Mbit/s)` <dbl>,
#   `Maximum download speed (Mbit/s)` <dbl>
> Q3
# A tibble: 1 × 6
  `Average download speed (Mbit/s)` `Average upload speed (Mbit/s)`
                              <dbl>                            <dbl>
1                              65.6                              7.4
# ℹ 4 more variables: `Minimum upload speed (Mbit/s)` <dbl>,
#   `Maximum upload speed (Mbit/s)` <dbl>,
#   `Minimum download speed (Mbit/s)` <dbl>,
#   `Maximum download speed (Mbit/s)` <dbl>
> 
```



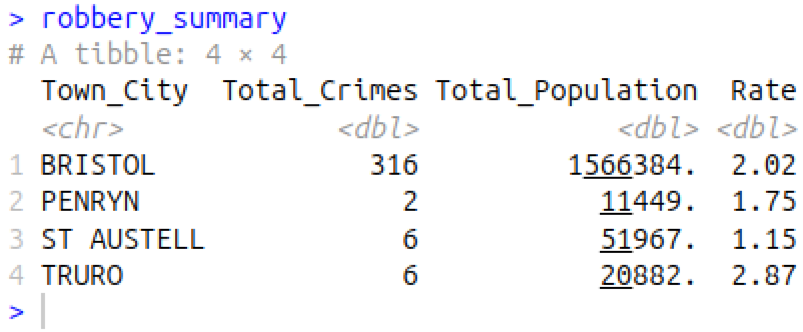
```
> summary(bb)
   Postcode          Average download speed (Mbit/s)
 Length:1317767      Min.   :   0.10                
 Class :character    1st Qu.:  27.20                
 Mode  :character    Median :  40.50                
                     Mean   :  46.11                
                     3rd Qu.:  63.10                
                     Max.   :1000.00                
                     NA's   :952                    
 Average upload speed (Mbit/s) Minimum upload speed (Mbit/s)
 Min.   :    0.000             Min.   :   0.1000            
 1st Qu.:    4.400             1st Qu.:   0.4000            
 Median :    6.300             Median :   0.7000            
 Mean   :    6.559             Mean   :   0.9351            
 3rd Qu.:    8.100             3rd Qu.:   0.9000            
 Max.   :18706.700             Max.   :1000.0000            
 NA's   :1051                  NA's   :1037                 
 Maximum upload speed (Mbit/s) Minimum download speed (Mbit/s)
 Min.   :     0.00             Min.   :   0.100               
 1st Qu.:    10.10             1st Qu.:   2.600               
 Median :    20.00             Median :   5.400               
 Mean   :    16.41             Mean   :   7.935               
 3rd Qu.:    20.00             3rd Qu.:   9.900               
 Max.   :300010.00             Max.   :1000.000               
 NA's   :1037                  NA's   :952                    
 Maximum download speed (Mbit/s)
 Min.   :   0.1                 
 1st Qu.:  62.2                 
 Median :  80.0                 
 Mean   : 113.4                 
 3rd Qu.: 200.0                 
 Max.   :4267.7                 
 NA's   :952                    
> 
```



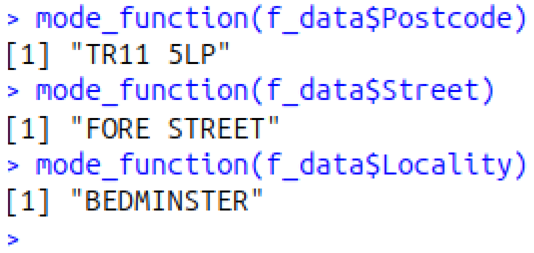
```
> population_data <- population_data %>%
+   mutate(Population2021 = 1.00561255390388033 * Population)
>
> population_data <- population_data %>%
+   mutate(Population2022 = 1.00561255390388033 * Population2021)
>
> population_data <- population_data %>%
+   mutate(Population2023 = 1.00561255390388033 * Population2022)
> population_data
# A tibble: 8,035 × 5
   Postcode Population Population2021 Population2022 Population2023
   <chr>         <dbl>          <dbl>          <dbl>          <dbl>
 1 AL1  1         5453          5484.          5514.          5545.
 2 AL1  2         6523          6560.          6596.          6633.
 3 AL1  3         4179          4202.          4226.          4250.
 4 AL1  4         9799          9854.          9909.          9965.
 5 AL1  5        10226         10283.         10341.         10399.
 6 AL10 0         9935          9991.         10047.         10103.
 7 AL10 8        10998         11060.         11122.         11184.
 8 AL10 9        14967         15051.         15135.         15220.
 9 AL2  1         9507          9560.          9614.          9668.
10 AL2  2         6130          6164.          6199.          6234.
# ℹ 8,025 more rows
# ℹ Use `print(n = ...)` to see more rows
> 
```



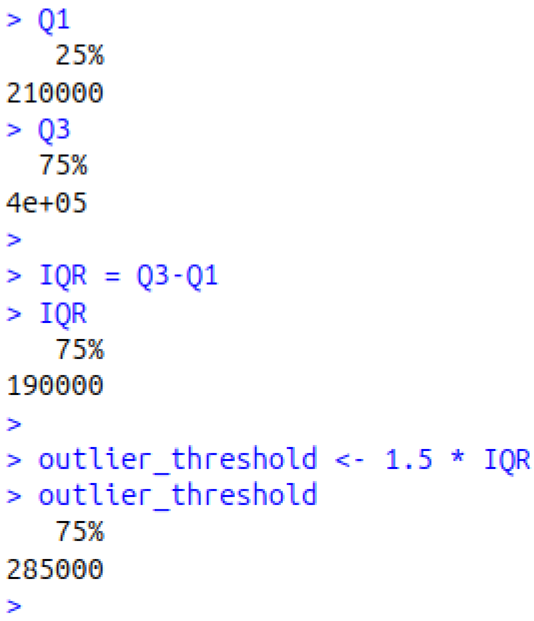
```
> print(robbery_data)
# A tibble: 2 × 4
  County          Total_Crimes Total_Population  Rate
  <chr>                  <dbl>            <dbl> <dbl>
1 CITY OF BRISTOL          341         1892059.  1.80
2 CORNWALL                  28          177769.  1.58
> 
```



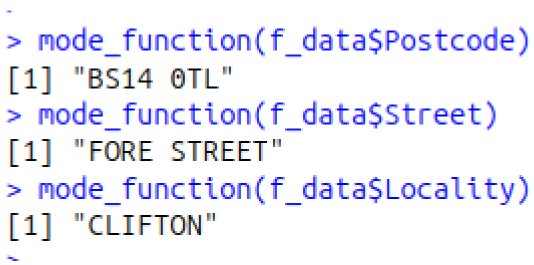
```
> robbery_summary
# A tibble: 4 × 4
  Town_City  Total_Crimes Total_Population  Rate
  <chr>             <dbl>            <dbl> <dbl>
1 BRISTOL             316         1566384.  2.02
2 PENRYN                2           11449.  1.75
3 ST AUSTELL            6           51967.  1.15
4 TRURO                 6           20882.  2.87
> 
```



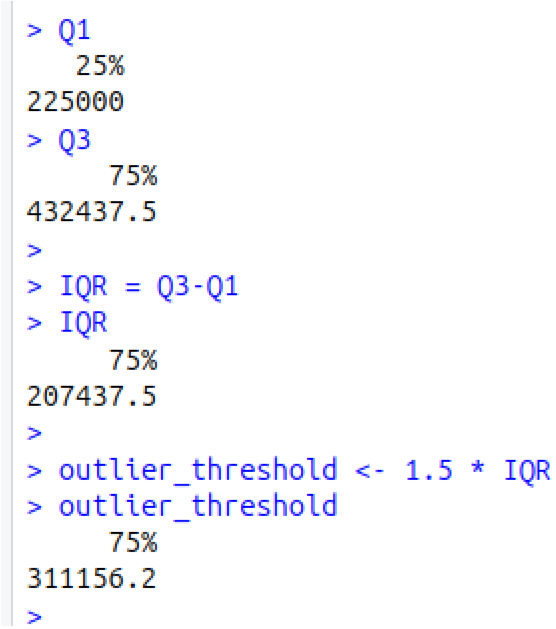
```
> mode_function(f_data$Postcode)
[1] "TR11 5LP"
> mode_function(f_data$Street)
[1] "FORE STREET"
> mode_function(f_data$Locality)
[1] "BEDMINSTER"
>
```



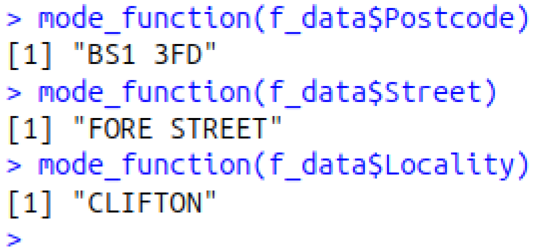
```
> Q1
   25%
210000
> Q3
  75%
4e+05
>
> IQR = Q3-Q1
> IQR
   75%
190000
>
> outlier_threshold <- 1.5 * IQR
> outlier_threshold
   75%
285000
>
```



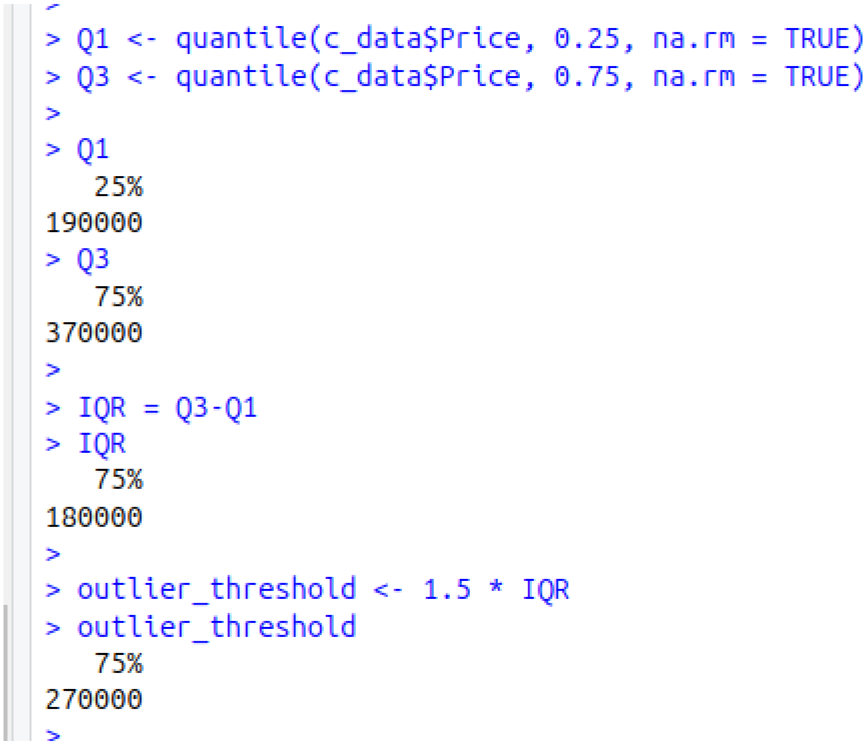
```
> mode_function(f_data$Postcode)
[1] "BS14 0TL"
> mode_function(f_data$Street)
[1] "FORE STREET"
> mode_function(f_data$Locality)
[1] "CLIFTON"
```



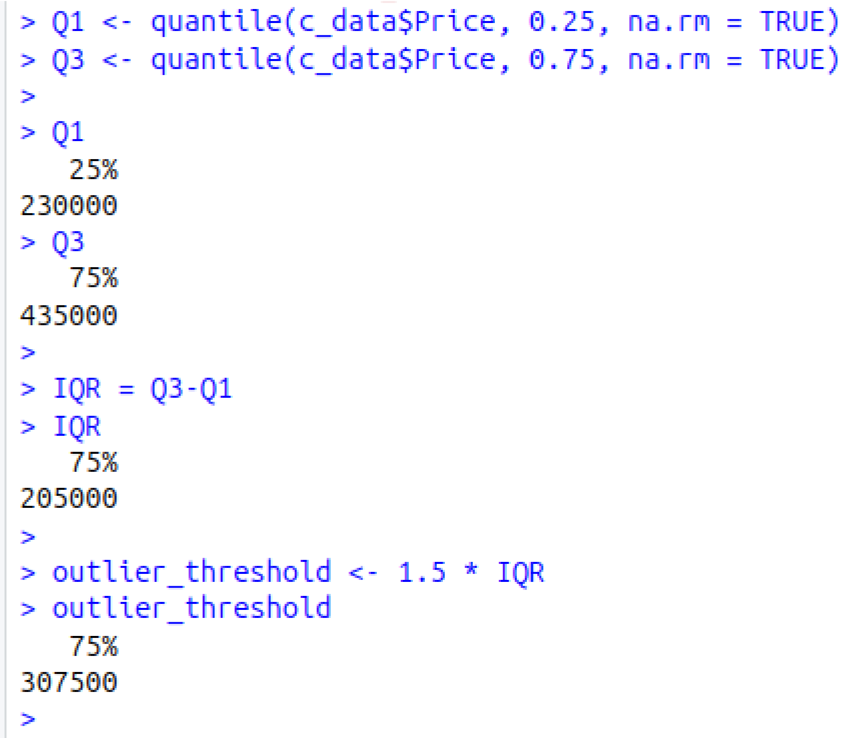
```
> Q1
   25%
225000
> Q3
     75%
432437.5
>
> IQR = Q3-Q1
> IQR
     75%
207437.5
>
> outlier_threshold <- 1.5 * IQR
> outlier_threshold
     75%
311156.2
>
```



```
> mode_function(f_data$Postcode)
[1] "BS1 3FD"
> mode_function(f_data$Street)
[1] "FORE STREET"
> mode_function(f_data$Locality)
[1] "CLIFTON"
>
```



```
> Q1 <- quantile(c_data$Price, 0.25, na.rm = TRUE)
> Q3 <- quantile(c_data$Price, 0.75, na.rm = TRUE)
>
> Q1
   25%
190000
> Q3
   75%
370000
>
> IQR = Q3-Q1
> IQR
   75%
180000
>
> outlier_threshold <- 1.5 * IQR
> outlier_threshold
   75%
270000
>
```



```
> Q1 <- quantile(c_data$Price, 0.25, na.rm = TRUE)
> Q3 <- quantile(c_data$Price, 0.75, na.rm = TRUE)
>
> Q1
   25%
230000
> Q3
   75%
435000
>
> IQR = Q3-Q1
> IQR
   75%
205000
>
> outlier_threshold <- 1.5 * IQR
> outlier_threshold
   75%
307500
>
```

Git hub link:-

https://github.com/Dipeshkhadgi/data-science-coursework-documentation