

Aryan Shah(230511)

B.Sc. (Hons.) Computing, Softwarica College of IT and E-Commerce, Coventry University
ST5014CEM Data Science for Developers

Siddhartha Neupane

Table of Contents

Introduction.....	3
Cleaning Data.....	4
House Sales.....	4
Towns and Postcodes.....	5
Broadband Speeds.....	8
Crime.....	10
School.....	14
Exploratory Data Analysis.....	20
House Prices.....	20
Broadband Speed.....	27
Crime Rates.....	33
Schools.....	42
Linear Modelling.....	48
Recommendation System.....	63
Overview.....	67
Results.....	67
Reflection.....	67
Broadband Speeds.....	68
School Grades.....	68
House Prices.....	68
Crimes.....	68
Overall Score.....	68
Legal and Ethical Issues.....	69
Conclusion.....	69
References.....	70
Appendix.....	71

Introduction

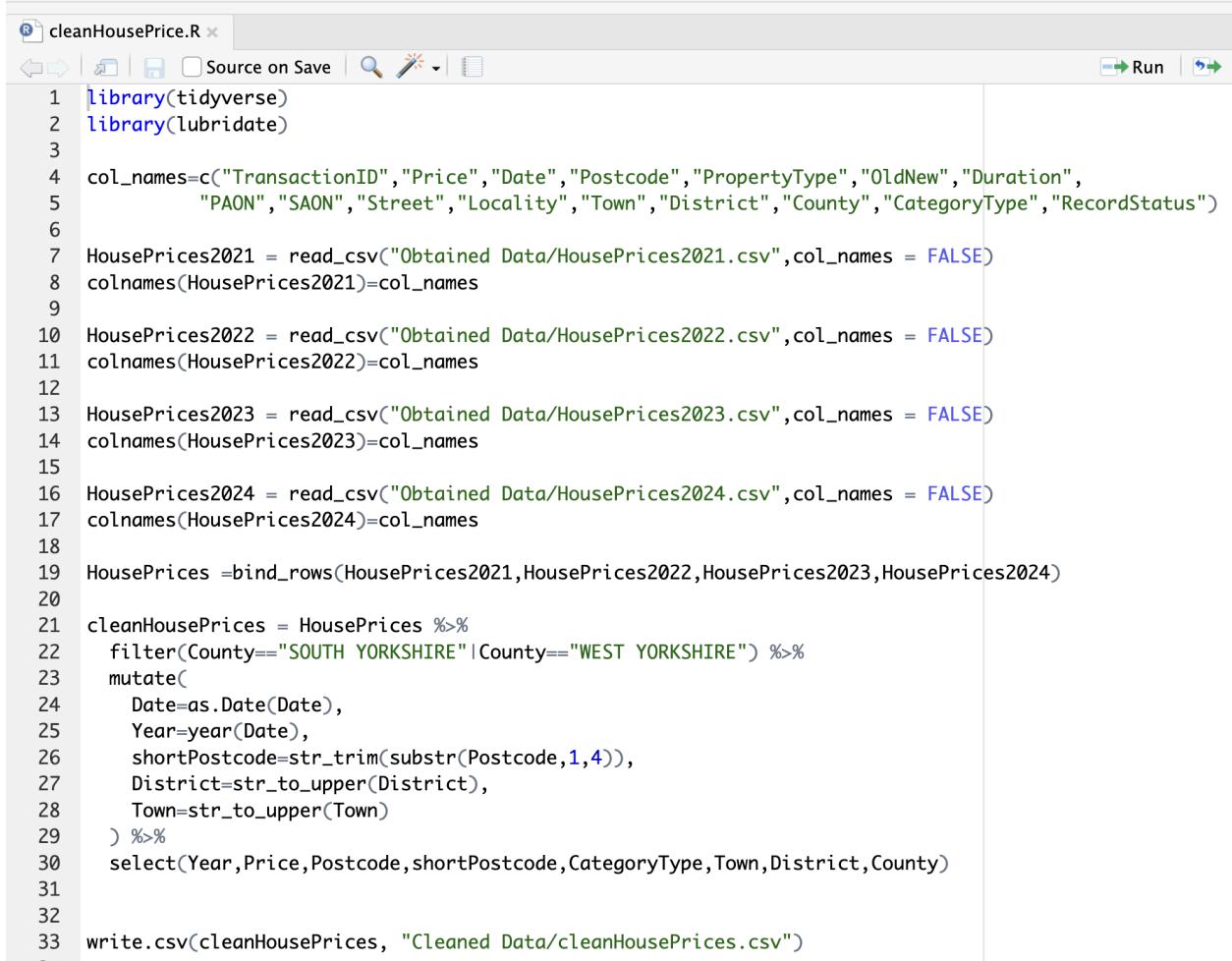
This document evaluates South and West Yorkshire urban areas for housing investment based on affordability, connectivity, safety, and quality of life according to house prices first, then internet speed, crime rates, and school performance. Using data from publicly available datasets on data.gov.uk, a data science workflow was applied: data cleaning and joining 2021–2024, followed by exploratory and statistical analysis. A weighted recommendation system (40% for affordability, 20% each for connectivity, safety, and quality of life) rates and ranks towns to identify the most suitable locations for investment. Analysis focuses on major Towns/District for relevance and completeness of data, making stable, evidence-based recommendations.

Cleaning Data

House Sales

2021–2024 House Price datasets were cleaned and joined together to support South and West Yorkshire property investment analysis. The steps included naming column conventions, year-to-year data joining, and filtering counties of interest. Date formatting was kept consistent, and a Year column was extracted. A shortPostcode was created in order to join with other datasets, and District and Town names were capitalized for uniformity. The cleaned dataset, with key fields (Year, Price, Postcode, shortPostcode, CategoryType, Town, District, County), was saved as cleanHousePrices.csv.

...1	Year	Price	Postcode	shortPostcode	CategoryType	Town	District	County
143	143	2021	86795	WF3 2GH	WF3	WAKEFIELD	WAKEFIELD	WEST YORKSHIRE
144	144	2021	132437	WF3 2GH	WF3	WAKEFIELD	WAKEFIELD	WEST YORKSHIRE
145	145	2021	102680	WF3 2GH	WF3	WAKEFIELD	WAKEFIELD	WEST YORKSHIRE
146	146	2021	86795	WF3 2GH	WF3	WAKEFIELD	WAKEFIELD	WEST YORKSHIRE
147	147	2021	86795	WF3 2GH	WF3	WAKEFIELD	WAKEFIELD	WEST YORKSHIRE
148	148	2021	100000	WF14 8AT	WF14	MIRFIELD	KIRKLEES	WEST YORKSHIRE
149	149	2021	76995	HX2 8QL	HX2	HALIFAX	CALDERDALE	WEST YORKSHIRE
150	150	2021	80000	WF10 1LH	WF10	CASTLEFORD	WAKEFIELD	WEST YORKSHIRE
151	151	2021	115297	LS29 6GZ	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE
152	152	2021	142000	S12 4AE	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE
153	153	2021	175000	S6 3NG	S6 3	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE
154	154	2021	280000	S63 6EU	S63	ROOTHERHAM	ROOTHERHAM	SOUTH YORKSHIRE
155	155	2021	243000	DN10 6QW	DN10	DONCASTER	DONCASTER	SOUTH YORKSHIRE
156	156	2021	103000	S71 5RE	S71	BARNESLEY	BARNESLEY	SOUTH YORKSHIRE
157	157	2021	122000	S63 9BY	S63	ROOTHERHAM	BARNESLEY	SOUTH YORKSHIRE
158	158	2021	116000	S36 1AT	S36	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE



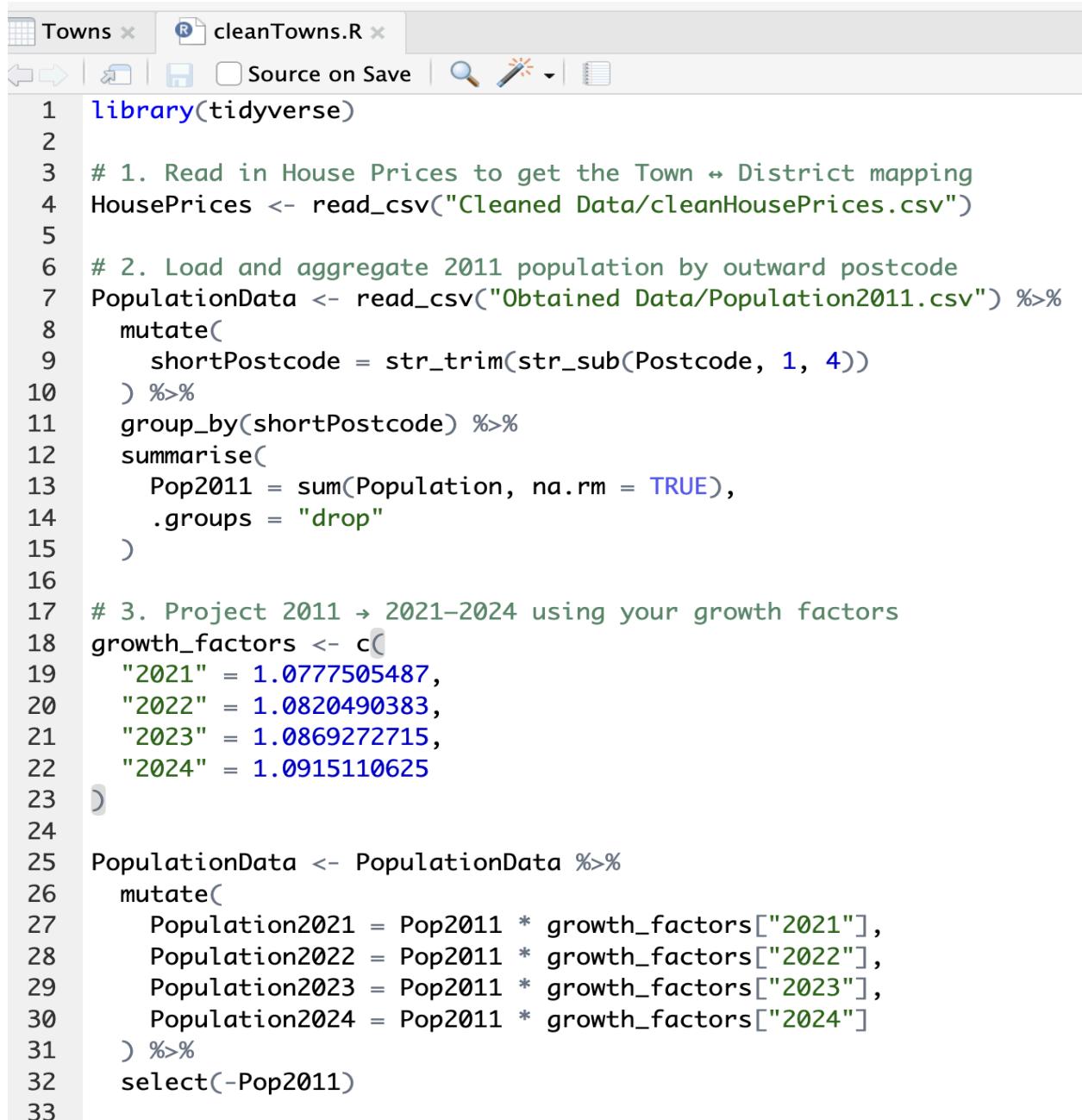
```

1 library(tidyverse)
2 library(lubridate)
3
4 col_names=c("TransactionID","Price","Date","Postcode","PropertyType","OldNew","Duration",
5 "PAON","SAON","Street","Locality","Town","District","County","CategoryType","RecordStatus")
6
7 HousePrices2021 = read_csv("Obtained Data/HousePrices2021.csv",col_names = FALSE)
8 colnames(HousePrices2021)=col_names
9
10 HousePrices2022 = read_csv("Obtained Data/HousePrices2022.csv",col_names = FALSE)
11 colnames(HousePrices2022)=col_names
12
13 HousePrices2023 = read_csv("Obtained Data/HousePrices2023.csv",col_names = FALSE)
14 colnames(HousePrices2023)=col_names
15
16 HousePrices2024 = read_csv("Obtained Data/HousePrices2024.csv",col_names = FALSE)
17 colnames(HousePrices2024)=col_names
18
19 HousePrices =bind_rows(HousePrices2021,HousePrices2022,HousePrices2023,HousePrices2024)
20
21 cleanHousePrices = HousePrices %>%
22   filter(County=="SOUTH YORKSHIRE" | County=="WEST YORKSHIRE") %>%
23   mutate(
24     Date=as.Date(Date),
25     Year=year(Date),
26     shortPostcode=str_trim(substr(Postcode,1,4)),
27     District=str_to_upper(District),
28     Town=str_to_upper(Town)
29   ) %>%
30   select(Year,Price,Postcode,shortPostcode,CategoryType,Town,District,County)
31
32
33 write.csv(cleanHousePrices, "Cleaned Data/cleanHousePrices.csv")

```

Towns and Postcodes

Town cleansing process generates a normalized lookup table of towns to West and South Yorkshire districts and counties, consolidating population data for 2021–2024. Starting with the cleansed dataset of house prices, individual occurrences of `shortPostcode`, `Town`, `District`, and `County` are selected. Processing 2011 population data leads to `shortPostcode` formation and population summation by `shortPostcode`. Population forecasts for 2021–2024 are forecasted based on pre-defined growth factors, and the population of 2011 is subtracted. The town data based on house prices is merged with population information based on `shortPostcode`, retaining only those records where there are proper 2021 population values. For consistency, `District` and `County` are made uppercase and trimmed. Counties are reassigned by reference to district tables for South Yorkshire and West Yorkshire and correcting any misassignments. The data is filtered to keep South and West Yorkshire only, sorted by `County`, `District`, and `Town`, and saved in `Towns.csv`.



```

1 library(tidyverse)
2
3 # 1. Read in House Prices to get the Town ↔ District mapping
4 HousePrices <- read_csv("Cleaned Data/cleanHousePrices.csv")
5
6 # 2. Load and aggregate 2011 population by outward postcode
7 PopulationData <- read_csv("Obtained Data/Population2011.csv") %>%
8   mutate(
9     shortPostcode = str_trim(str_sub(Postcode, 1, 4))
10    ) %>%
11   group_by(shortPostcode) %>%
12   summarise(
13     Pop2011 = sum(Population, na.rm = TRUE),
14     .groups = "drop"
15   )
16
17 # 3. Project 2011 → 2021–2024 using your growth factors
18 growth_factors <- c(
19   "2021" = 1.0777505487,
20   "2022" = 1.0820490383,
21   "2023" = 1.0869272715,
22   "2024" = 1.0915110625
23 )
24
25 PopulationData <- PopulationData %>%
26   mutate(
27     Population2021 = Pop2011 * growth_factors["2021"],
28     Population2022 = Pop2011 * growth_factors["2022"],
29     Population2023 = Pop2011 * growth_factors["2023"],
30     Population2024 = Pop2011 * growth_factors["2024"]
31   ) %>%
32   select(-Pop2011)
33

```

```

33
34 # 4. Build initial Town ↔ District ↔ County table
35 towns_raw <- HousePrices %>%
36   select(shortPostcode, Town, District, County) %>%
37   distinct() %>%
38   left_join(PopulationData, by = "shortPostcode") %>%
39   filter(!is.na(Population2021))
40
41 # 5. Explicitly correct Yorkshire districts
42 south_yorkshire <- c("SHEFFIELD", "BARNESLEY", "DONCASTER", "ROOTHERHAM")
43 west_yorkshire <- c("LEEDS", "BRADFORD", "CALDERDALE", "KIRKLEES", "WAKEFIELD")
44
45 towns_clean <- towns_raw %>%
46   # normalize case
47   mutate(
48     District = str_to_upper(str_trim(District)),
49     County   = str_to_upper(str_trim(County)))
50   ) %>%
51   # overwrite any mis-assigned county based on district
52   mutate(
53     County = case_when(
54       District %in% south_yorkshire ~ "SOUTH YORKSHIRE",
55       District %in% west_yorkshire ~ "WEST YORKSHIRE",
56       TRUE                      ~ County
57     )
58   ) %>%
59   # keep only the two target counties
60   filter(County %in% c("SOUTH YORKSHIRE", "WEST YORKSHIRE")) %>%
61   arrange(County, District, Town)
62
63 # 6. Save out your cleaned lookup
64 write_csv(towns_clean, "Cleaned Data/Towns.csv")

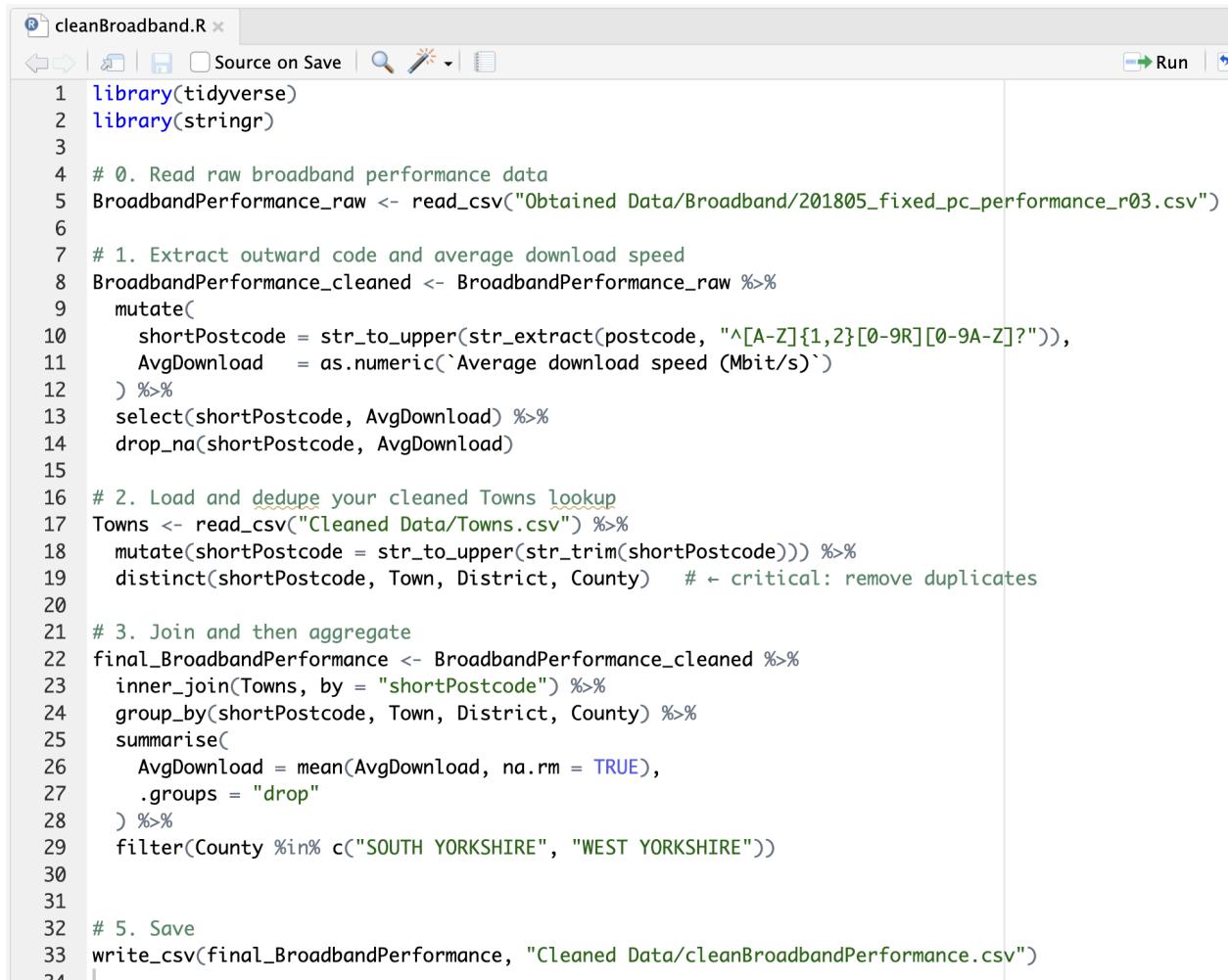
```

Towns x cleanTowns.R x Filter

	shortPostcode	Town	District	County	Population2021	Population2022	Population2023	Population2024
63	S17	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	16797.820	16864.816	16940.848	17012.291
64	S10	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	51093.998	51297.781	51529.048	51746.356
65	S13	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	33054.609	33186.444	33336.059	33476.644
66	S20	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	33821.968	33956.863	34109.952	34253.800
67	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	37012.109	37159.728	37327.256	37484.673
68	S11	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	37056.297	37204.092	37371.820	37529.425
69	S36	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	29473.244	29590.795	29724.200	29849.553
70	S35	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	44619.950	44797.912	44999.876	45189.649
71	S14	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	10416.459	10458.004	10505.152	10549.454
72	S26	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	27900.806	28012.086	28138.373	28257.038
73	S25	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	22741.614	22832.317	22935.252	23031.975
74	IP18	SOUTHWOLD	SHEFFIELD	SOUTH YORKSHIRE	4375.667	4393.119	4412.925	4431.535
75	DE11	SWADLINCOTE	SHEFFIELD	SOUTH YORKSHIRE	47215.174	47403.486	47617.197	47818.008
76	WF17	BATLEY	BRADFORD	WEST YORKSHIRE	47118.176	47306.102	47519.373	47719.772
77	BD16	BINGLEY	BRADFORD	WEST YORKSHIRE	27809.197	27920.111	28045.984	28164.260
78	BD4	BRADFORD	BRADFORD	WEST YORKSHIRE	33645.217	33779.407	33931.696	34074.792
79	BD10	BRADFORD	BRADFORD	WEST YORKSHIRE	27894.340	28005.593	28131.852	28250.489
80	BD1	BRADFORD	BRADFORD	WEST YORKSHIRE	4134.251	4150.740	4169.453	4187.036
81	BD9	BRADFORD	BRADFORD	WEST YORKSHIRE	31587.791	31713.775	31856.751	31991.098
82	BD12	BRADFORD	BRADFORD	WEST YORKSHIRE	17982.268	18053.988	18135.382	18211.862
83	BD13	BRADFORD	BRADFORD	WEST YORKSHIRE	26921.131	27028.503	27150.356	27264.855
84	BD8	BRADFORD	BRADFORD	WEST YORKSHIRE	33951.298	34086.709	34240.383	34384.781
85	BD6	BRADFORD	BRADFORD	WEST YORKSHIRE	32410.115	32539.379	32686.077	32823.921
86	BD3	BRADFORD	BRADFORD	WEST YORKSHIRE	36666.151	36812.390	36978.353	37134.298

Broadband Speeds

The Broadband performance dataset is cleaned and processed. It loads raw broadband and drops the outside part of postcodes (shortPostcode) with regex in uppercase to be consistent. It also converts the average download speed column into numeric form and retains only applicable columns (shortPostcode, AvgDownload). A cleaned town lookup table is then imported and deduplicated so that shortPostcode uniquely maps to a Town, District, and County. The cleaned town data and broadband data are then joined using the key shortPostcode. This allows for aggregation of average download speed by Town, District, and County. The dataset is then filtered to retain only records from South Yorkshire and West Yorkshire. Finally, the cleaned and combined broadband performance data set is stored as cleanBroadbandPerformance.csv.



The screenshot shows the RStudio interface with the script file 'cleanBroadband.R' open. The code is written in R and performs the following steps:

- Imports the tidyverse and stringr packages.
- Reads raw broadband performance data from '201805_fixed_pc_performance_r03.csv'.
- Extracts outward code and average download speed, creating a new dataset 'BroadbandPerformance_cleaned'.
- Loads and deduplicates a 'Towns' lookup dataset.
- Joins the cleaned broadband data with the towns lookup, groups by short postcode, and aggregates the average download speed.
- Filters the results to include only data from South Yorkshire and West Yorkshire.
- Saves the final cleaned broadband performance data to 'cleanBroadbandPerformance.csv'.

```
library(tidyverse)
library(stringr)

# 0. Read raw broadband performance data
BroadbandPerformance_raw <- read_csv("Obtained Data/Broadband/201805_fixed_pc_performance_r03.csv")

# 1. Extract outward code and average download speed
BroadbandPerformance_cleaned <- BroadbandPerformance_raw %>%
  mutate(
    shortPostcode = str_to_upper(str_extract(postcode, "^[A-Z]{1,2}[0-9R][0-9A-Z]?")),
    AvgDownload   = as.numeric(`Average download speed (Mbit/s)`)
  ) %>%
  select(shortPostcode, AvgDownload) %>%
  drop_na(shortPostcode, AvgDownload)

# 2. Load and dedupe your cleaned Towns lookup
Towns <- read_csv("Cleaned Data/Towns.csv") %>%
  mutate(shortPostcode = str_to_upper(str_trim(shortPostcode))) %>%
  distinct(shortPostcode, Town, District, County) # ← critical: remove duplicates

# 3. Join and then aggregate
final_BroadbandPerformance <- BroadbandPerformance_cleaned %>%
  inner_join(Towns, by = "shortPostcode") %>%
  group_by(shortPostcode, Town, District, County) %>%
  summarise(
    AvgDownload = mean(AvgDownload, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  filter(County %in% c("SOUTH YORKSHIRE", "WEST YORKSHIRE"))

# 5. Save
write_csv(final_BroadbandPerformance, "Cleaned Data/cleanBroadbandPerformance.csv")
```

cleanBroadband.R x cleanBroadbandPerformance x Filter

	shortPostcode	Town	District	County	AvgDownload
18	BD19	CLECKHEATON	CALDERDALE	WEST YORKSHIRE	29.99905
19	BD19	CLECKHEATON	KIRKLEES	WEST YORKSHIRE	29.99905
20	BD20	KEIGHLEY	BRADFORD	WEST YORKSHIRE	38.48827
21	BD21	KEIGHLEY	BRADFORD	WEST YORKSHIRE	50.45494
22	BD21	KEIGHLEY	KIRKLEES	WEST YORKSHIRE	50.45494
23	BD22	KEIGHLEY	BRADFORD	WEST YORKSHIRE	42.48948
24	BD22	KEIGHLEY	LEEDS	WEST YORKSHIRE	42.48948
25	DE11	SWADLINCOTE	SHEFFIELD	SOUTH YORKSHIRE	34.69866
26	DN10	DONCASTER	DONCASTER	SOUTH YORKSHIRE	30.31514
27	DN11	DONCASTER	DONCASTER	SOUTH YORKSHIRE	28.86158
28	DN12	DONCASTER	DONCASTER	SOUTH YORKSHIRE	50.15452
29	DN12	DONCASTER	ROTHERHAM	SOUTH YORKSHIRE	50.15452
30	DN14	GOOLE	DONCASTER	SOUTH YORKSHIRE	29.18143
31	IP18	SOUTHWOLD	SHEFFIELD	SOUTH YORKSHIRE	33.75238
32	LS10	LEEDS	LEEDS	WEST YORKSHIRE	49.15421
33	LS11	LEEDS	LEEDS	WEST YORKSHIRE	55.18745

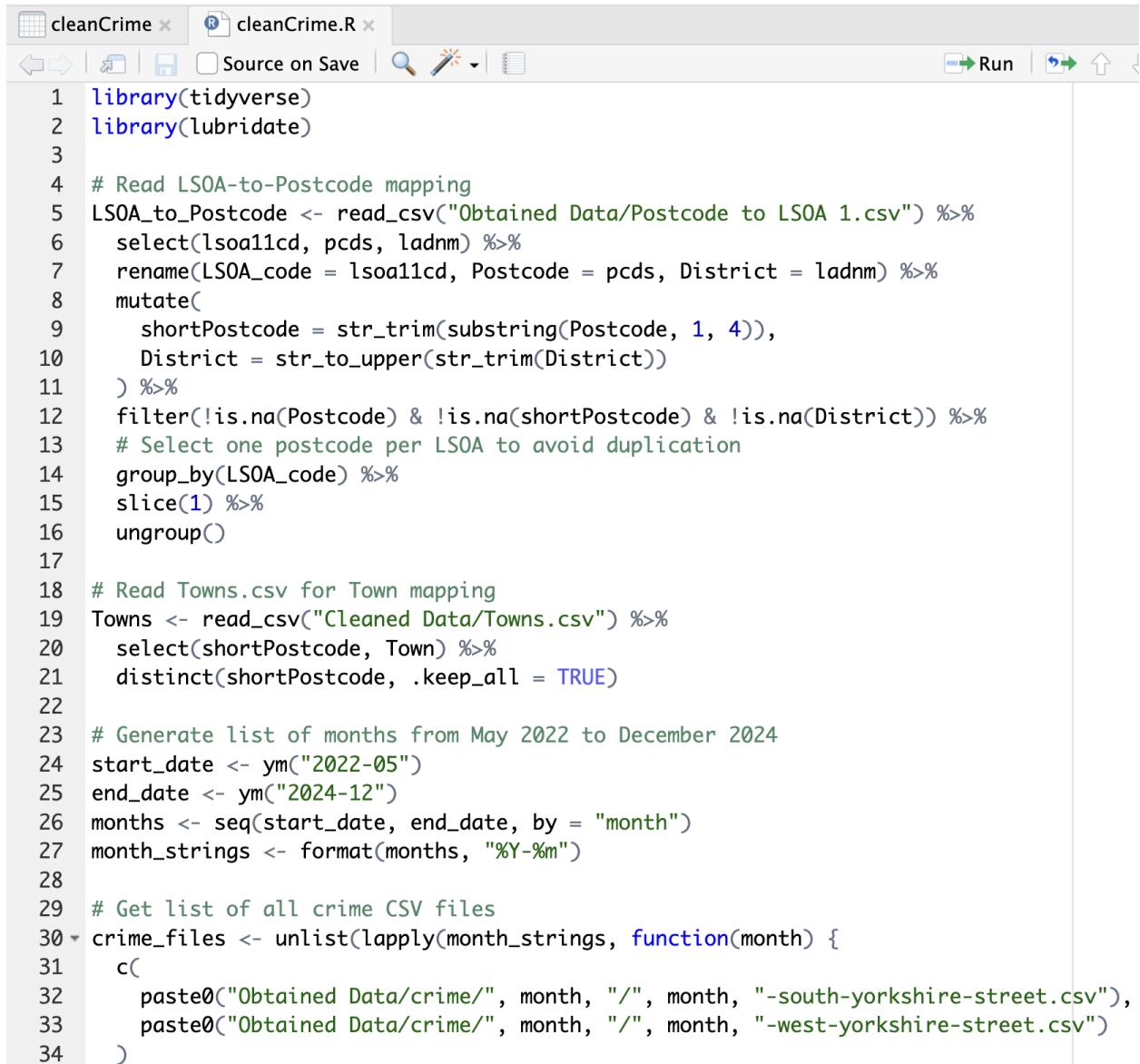
Crime

Raw crime data sets are pre-cleaned from May 2022 to December 2024 for West and South Yorkshire. It initially loads a postcode-to-LSOA map file and extracts significant columns like LSOA code, postcode, and district name. A shortPostcode is derived by extracting the first four characters of all postcodes to facilitate easier merges later.

It subsequently loads a sanitized towns ref file to translate short postcodes to the corresponding towns. An array of file paths for months within the specified time interval is created, and only files that exist are retained. Each crime CSV file is read, parsing errors being written to log and rows containing error rows being discarded in order to prepare the data in a clean format.

The crime data is cleansed by selecting proper columns, removing empty or null rows, and focusing on three kinds of crime: "Drugs", "Vehicle crime", and "Robbery". The County column is normalized by removing trailing "Police" and converting it to uppercase. A Year column is also extracted.

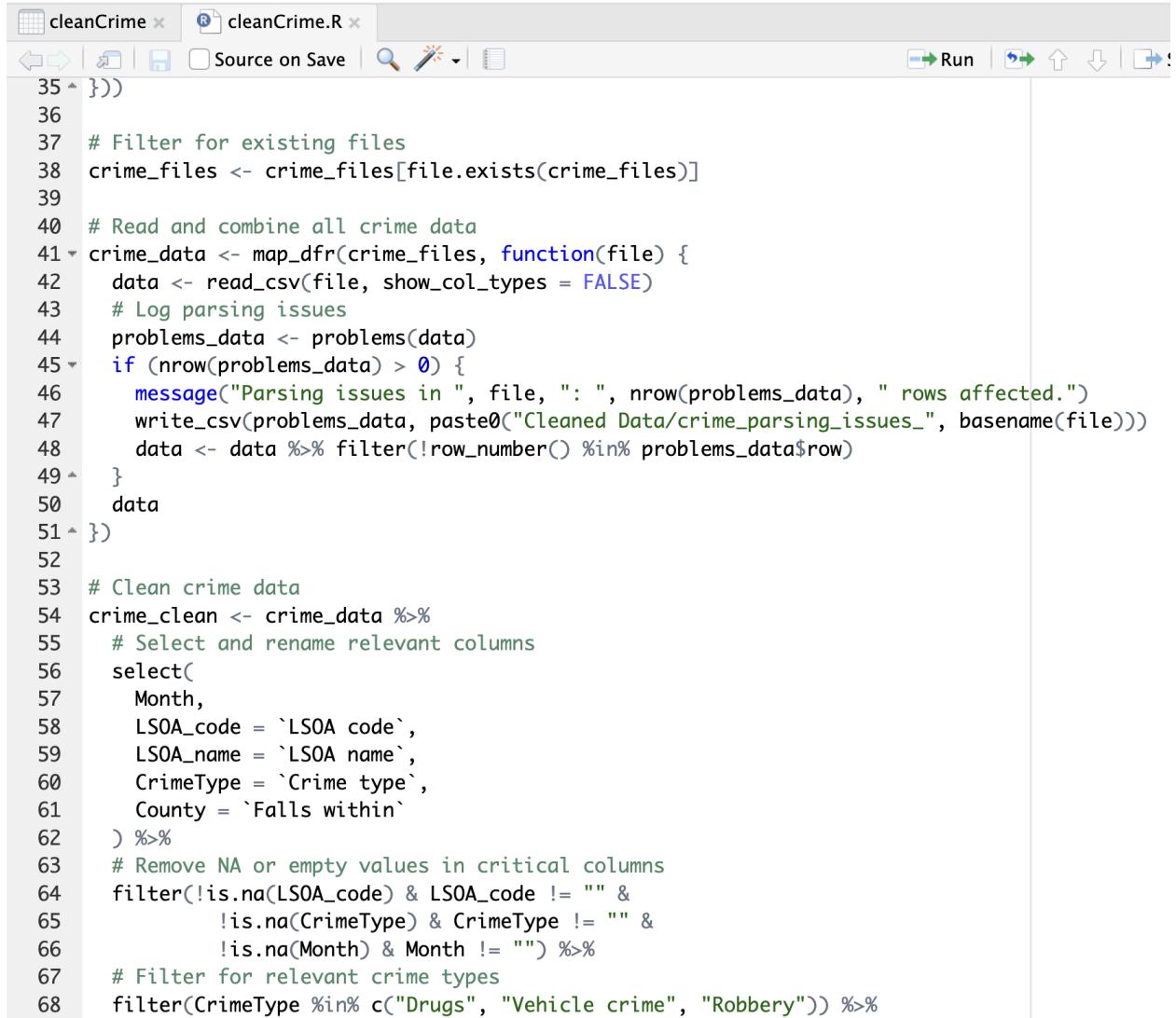
The cleaned crime records are augmented by joining them with postcode–LSOA mapping and town dataframes to include Postcode, shortPostcode, Town, and District information. Valid data with town information from South and West Yorkshire only is retained. The cleaned and standardized dataset is saved in the file cleanCrime.csv.



```

1 library(tidyverse)
2 library(lubridate)
3
4 # Read LSOA-to-Postcode mapping
5 LSOA_to_Postcode <- read_csv("Obtained Data/Postcode to LSOA 1.csv") %>%
6   select(lsoa11cd, pcds, ladnm) %>%
7   rename(LSOA_code = lsoa11cd, Postcode = pcds, District = ladnm) %>%
8   mutate(
9     shortPostcode = str_trim(substring(Postcode, 1, 4)),
10    District = str_to_upper(str_trim(District)))
11 ) %>%
12 filter(!is.na(Postcode) & !is.na(shortPostcode) & !is.na(District)) %>%
13 # Select one postcode per LSOA to avoid duplication
14 group_by(LSOA_code) %>%
15 slice(1) %>%
16 ungroup()
17
18 # Read Towns.csv for Town mapping
19 Towns <- read_csv("Cleaned Data/Towns.csv") %>%
20   select(shortPostcode, Town) %>%
21   distinct(shortPostcode, .keep_all = TRUE)
22
23 # Generate list of months from May 2022 to December 2024
24 start_date <- ym("2022-05")
25 end_date <- ym("2024-12")
26 months <- seq(start_date, end_date, by = "month")
27 month_strings <- format(months, "%Y-%m")
28
29 # Get list of all crime CSV files
30 crime_files <- unlist(lapply(month_strings, function(month) {
31   c(
32     paste0("Obtained Data/crime/", month, "/", month, "-south-yorkshire-street.csv"),
33     paste0("Obtained Data/crime/", month, "/", month, "-west-yorkshire-street.csv")
34   )
35 }))

```



The screenshot shows the RStudio interface with the 'cleanCrime.R' script open. The code is a series of R commands for cleaning crime data. It starts by filtering existing files, then reads and combines all crime data, logs parsing issues, and filters out rows with problems. It then cleans the crime data by selecting relevant columns, removing NA or empty values from critical columns, and filtering for relevant crime types.

```
35 })
36
37 # Filter for existing files
38 crime_files <- crime_files[file.exists(crime_files)]
39
40 # Read and combine all crime data
41 crime_data <- map_dfr(crime_files, function(file) {
42   data <- read_csv(file, show_col_types = FALSE)
43   # Log parsing issues
44   problems_data <- problems(data)
45   if (nrow(problems_data) > 0) {
46     message("Parsing issues in ", file, ":", nrow(problems_data), " rows affected.")
47     write_csv(problems_data, paste0("Cleaned Data/crime_parsing_issues_", basename(file)))
48     data <- data %>% filter(!row_number() %in% problems_data$row)
49   }
50   data
51 })
52
53 # Clean crime data
54 crime_clean <- crime_data %>%
55   # Select and rename relevant columns
56   select(
57     Month,
58     LSOA_code = `LSOA code`,
59     LSOA_name = `LSOA name`,
60     CrimeType = `Crime type`,
61     County = `Falls within`
62   ) %>%
63   # Remove NA or empty values in critical columns
64   filter(!is.na(LSOA_code) & LSOA_code != "" &
65         !is.na(CrimeType) & CrimeType != "" &
66         !is.na(Month) & Month != "") %>%
67   # Filter for relevant crime types
68   filter(CrimeType %in% c("Drugs", "Vehicle crime", "Robbery")) %>%
```

```
67 # Filter for relevant crime types
68 filter(CrimeType %in% c("Drugs", "Vehicle crime", "Robbery")) %>%
69 # Clean and standardize columns
70 mutate(
71   Month = str_trim(Month),
72   LSOA_code = str_trim(LSOA_code),
73   LSOA_name = str_trim(LSOA_name),
74   CrimeType = str_trim(CrimeType),
75   County = str_to_upper(str_remove(str_trim(County), " Police$")),
76   Year = as.numeric(substr(Month, 1, 4))
77 ) %>%
78 # Join with LSOA_to_Postcode to get Postcode, shortPostcode, District
79 left_join(LSOA_to_Postcode, by = "LSOA_code", relationship = "many-to-one") %>%
80 # Join with Towns to get Town
81 left_join(Towns, by = "shortPostcode", relationship = "many-to-one") %>%
82 # Filter for South and West Yorkshire
83 filter(County %in% c("SOUTH YORKSHIRE", "WEST YORKSHIRE") & !is.na(Town)) %>%
84 # Select final columns, **including Month**
85 select(Postcode, shortPostcode, Town, District, County, CrimeType, Year, Month) %>%
86 # Remove any remaining NA values
87 filter(complete.cases(.))
88
89
90 # Write to CSV
91 write.csv(crime_clean, "Cleaned Data/cleanCrime.csv", row.names = FALSE)
92
93 View(crime_clean)
```

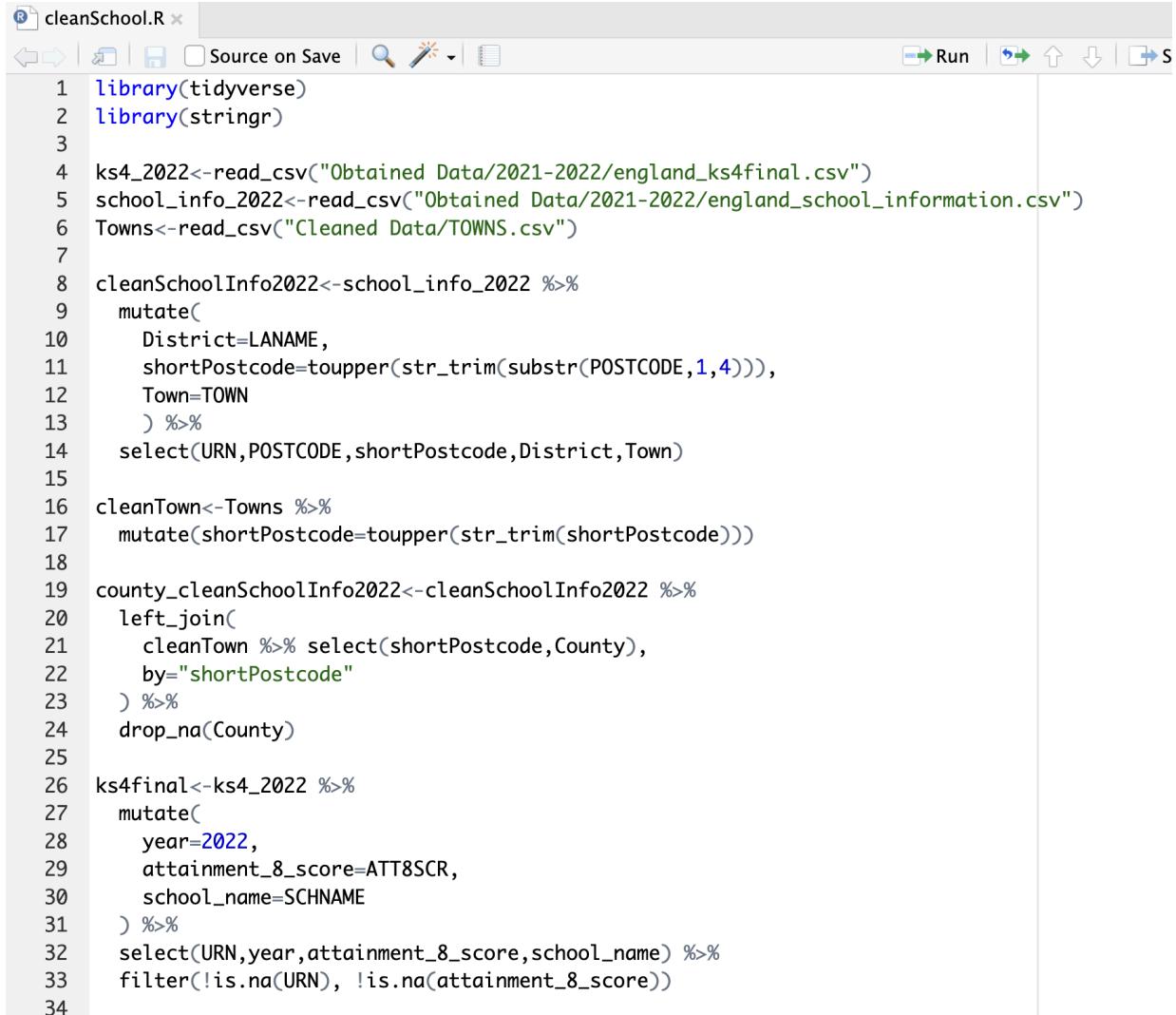
cleanCrime x cleanCrime.R x Filter

	Postcode	shortPostcode	Town	District	County	CrimeType	Year	Month
4434	S12 3AA	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	Robbery	2022	2022-06
4435	S12 3AA	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	Vehicle crime	2022	2022-06
4436	S12 3AA	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	Vehicle crime	2022	2022-06
4437	S12 3AS	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	Vehicle crime	2022	2022-06
4438	S12 3AS	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	Vehicle crime	2022	2022-06
4439	S12 3AS	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	Vehicle crime	2022	2022-06
4440	S12 2DY	S12	SHEFFIELD	SHEFFIELD	SOUTH YORKSHIRE	Drugs	2022	2022-06
4441	S75 4BX	S75	BARNSLEY	WAKEFIELD	SOUTH YORKSHIRE	Vehicle crime	2022	2022-06
4442	LS29 0UB	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4443	LS29 0UB	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4444	LS29 0UB	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4445	LS29 8AH	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Drugs	2022	2022-06
4446	LS29 0AB	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Drugs	2022	2022-06
4447	LS29 0AB	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4448	LS29 0AA	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4449	LS29 8AY	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4450	LS29 9AB	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4451	LS29 9AB	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4452	LS29 8AT	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4453	LS29 7EG	LS29	ILKLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4454	BD20 0AB	BD20	KEIGHLEY	BRADFORD	WEST YORKSHIRE	Robbery	2022	2022-06
4455	BD20 0AB	BD20	KEIGHLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4456	BD20 0AA	BD20	KEIGHLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06
4457	BD20 6FR	BD20	KEIGHLEY	BRADFORD	WEST YORKSHIRE	Vehicle crime	2022	2022-06

School

The school cleaning operation of data cleanses datasets of Key Stage 4 (KS4) performance and school data (2021-2022,2022-2023,2023-2024) truncated and stored exam-year i.e.(2022-2024) for property investment analysis of the South and West Yorkshire. School data datasets are cleansed year-wise to fetch 'URN', 'POSTCODE', 'shortPostcode' (uppercase, first four characters truncated of 'POSTCODE'), 'District' (calculated from 'LANAME'), and 'Town'. These are joined with the cleaned 'Towns.csv' data to match on 'County' using 'shortPostcode', removing invalid county records. KS4 datasets are cleaned by selecting 'URN', 'year', 'attainment_8_score' (from 'ATT8SCR'), and 'school_name', removing missing 'URN' or 'attainment_8_score'. School data and cleaned KS4 data are joined on 'URN', retaining records with valid 'County', 'District', 'attainment_8_score', and 'Town'. The yearly datasets

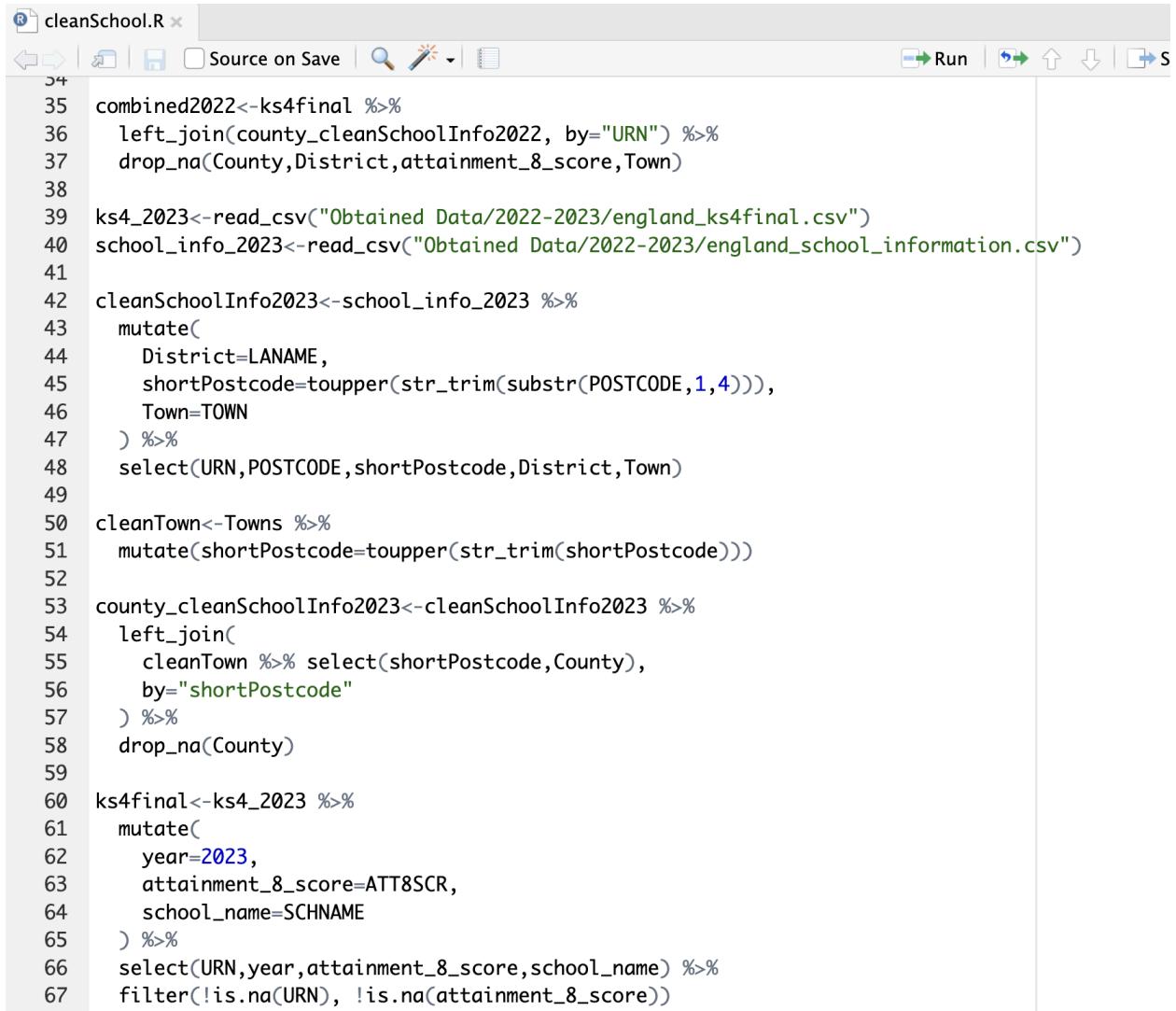
(2022–2024) are merged with `bind_rows`. Non-numeric `attainment_8_score` values (e.g., "SUPP", "NE", "NA") are excluded, and the column is converted to numeric with all 'NA' removed. The final dataset, saved as `cleanSchool.csv`, gives consistent, numeric school performance for town-wise aggregation by the recommendation system and for plotting.



The screenshot shows an RStudio interface with the following details:

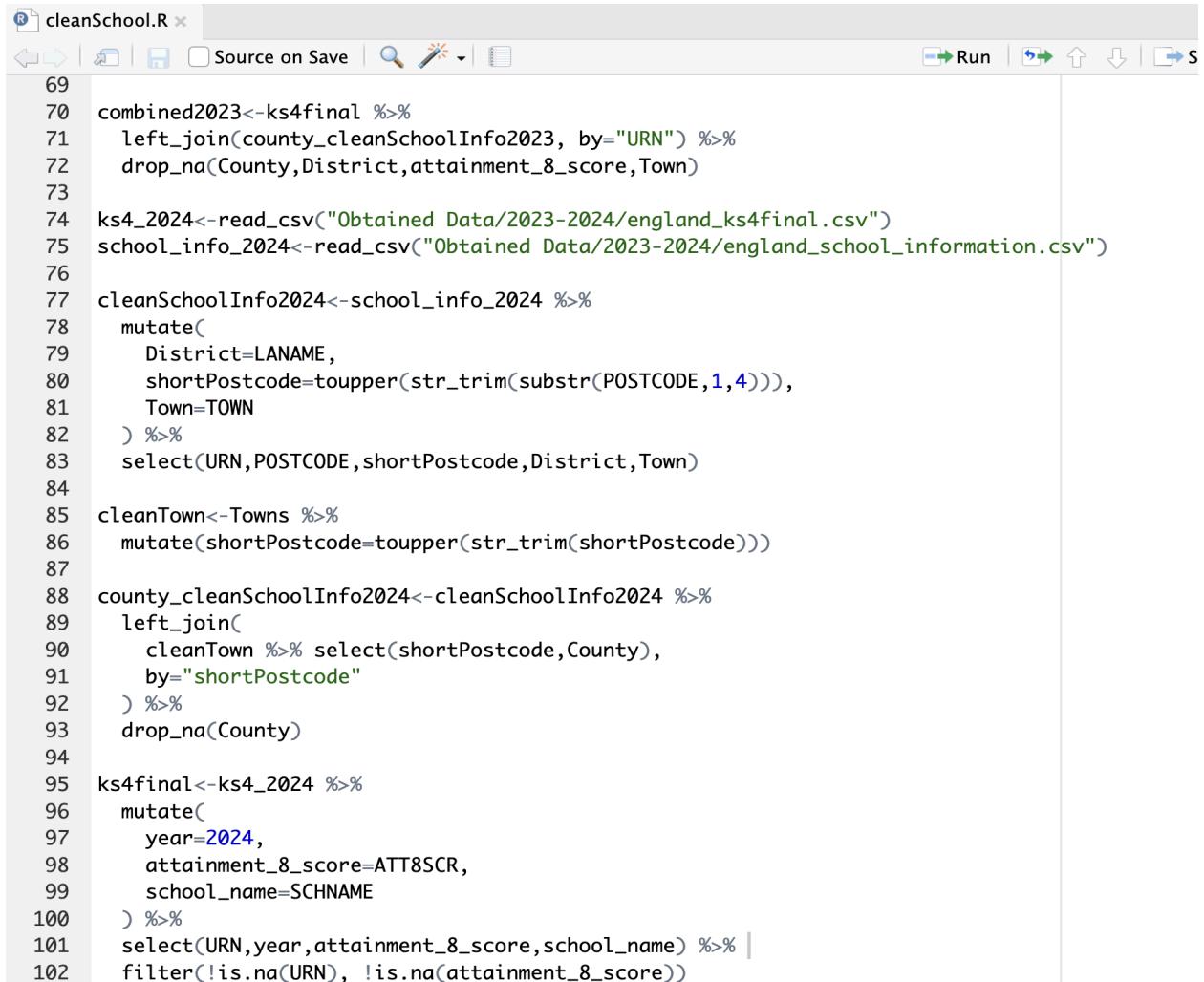
- Title Bar:** The title bar displays "cleanSchool.R x".
- Toolbar:** The toolbar includes standard icons for file operations (New, Open, Save, Print), search, and run.
- Code Area:** The main area contains the R code for the "cleanSchool" script. The code uses the tidyverse and stringr packages to read CSV files, manipulate data frames, and perform joins.

```
1 library(tidyverse)
2 library(stringr)
3
4 ks4_2022<-read_csv("Obtained Data/2021-2022/england_ks4final.csv")
5 school_info_2022<-read_csv("Obtained Data/2021-2022/england_school_information.csv")
6 Towns<-read_csv("Cleaned Data/TOWNS.csv")
7
8 cleanSchoolInfo2022<-school_info_2022 %>%
9   mutate(
10     District=LANAME,
11     shortPostcode=toupper(str_trim(substr(POSTCODE,1,4))),
12     Town=TOWN
13   ) %>%
14   select(URN,POSTCODE,shortPostcode,District,Town)
15
16 cleanTown<-Towns %>%
17   mutate(shortPostcode=toupper(str_trim(shortPostcode)))
18
19 county_cleanSchoolInfo2022<-cleanSchoolInfo2022 %>%
20   left_join(
21     cleanTown %>% select(shortPostcode,County),
22     by="shortPostcode"
23   ) %>%
24   drop_na(County)
25
26 ks4final<-ks4_2022 %>%
27   mutate(
28     year=2022,
29     attainment_8_score=ATT8SCR,
30     school_name=SCHNAME
31   ) %>%
32   select(URN,year,attainment_8_score,school_name) %>%
33   filter(!is.na(URN), !is.na(attainment_8_score))
```



The screenshot shows an RStudio interface with the file 'cleanSchool.R' open. The code is a pipeline for data cleaning, starting with combining two datasets from 2022 and 2023, then reading in school information CSV files, and finally performing various mutations and joins to standardize the data. The code uses the dplyr package's pipe operator (%>%).

```
34 combined2022<-ks4final %>%
35   left_join(county_cleanSchoolInfo2022, by="URN") %>%
36   drop_na(County,District,attainment_8_score,Town)
37
38 ks4_2023<-read_csv("Obtained Data/2022-2023/england_ks4final.csv")
39 school_info_2023<-read_csv("Obtained Data/2022-2023/england_school_information.csv")
40
41 cleanSchoolInfo2023<-school_info_2023 %>%
42   mutate(
43     District=LANAME,
44     shortPostcode=toupper(str_trim(substr(POSTCODE,1,4))),
45     Town=TOWN
46   ) %>%
47   select(URN,POSTCODE,shortPostcode,District,Town)
48
49 cleanTown<-Towns %>%
50   mutate(shortPostcode=toupper(str_trim(shortPostcode)))
51
52 county_cleanSchoolInfo2023<-cleanSchoolInfo2023 %>%
53   left_join(
54     cleanTown %>% select(shortPostcode,County),
55     by="shortPostcode"
56   ) %>%
57   drop_na(County)
58
59 ks4final<-ks4_2023 %>%
60   mutate(
61     year=2023,
62     attainment_8_score=ATT8SCR,
63     school_name=SCHNAME
64   ) %>%
65   select(URN,year,attainment_8_score,school_name) %>%
66   filter(!is.na(URN), !is.na(attainment_8_score))
```



The screenshot shows the RStudio interface with the script file "cleanSchool.R" open. The code is written in R and performs data cleaning and joining operations. It starts by reading two CSV files: "england_ks4final.csv" and "england_school_information.csv". It then creates a "cleanSchoolInfo2024" dataset by joining the two files on the "URN" column. The code uses dplyr functions like left_join, drop_na, mutate, and select. It also performs string manipulation on the "POSTCODE" column to create a shorter version. Finally, it joins this cleaned information with the original "cleanSchoolInfo2024" dataset on the "shortPostcode" column, drops rows with missing values in the "County" column, and filters out rows where "URN" or "attainment_8_score" are missing.

```
69
70 combined2023<-ks4final %>%
71   left_join(county_cleanSchoolInfo2023, by="URN") %>%
72   drop_na(County,District,attainment_8_score,Town)
73
74 ks4_2024<-read_csv("Obtained Data/2023-2024/england_ks4final.csv")
75 school_info_2024<-read_csv("Obtained Data/2023-2024/england_school_information.csv")
76
77 cleanSchoolInfo2024<-school_info_2024 %>%
78   mutate(
79     District=LANAME,
80     shortPostcode=toupper(str_trim(substr(POSTCODE,1,4))),
81     Town=TOWN
82   ) %>%
83   select(URN,POSTCODE,shortPostcode,District,Town)
84
85 cleanTown<-Towns %>%
86   mutate(shortPostcode=toupper(str_trim(shortPostcode)))
87
88 county_cleanSchoolInfo2024<-cleanSchoolInfo2024 %>%
89   left_join(
90     cleanTown %>% select(shortPostcode,County),
91     by="shortPostcode"
92   ) %>%
93   drop_na(County)
94
95 ks4final<-ks4_2024 %>%
96   mutate(
97     year=2024,
98     attainment_8_score=ATT8SCR,
99     school_name=SCHNAME
100   ) %>%
101   select(URN,year,attainment_8_score,school_name) %>% |
102   filter(!is.na(URN), !is.na(attainment_8_score))
```

```

105 combined2024<-ks4final %>%
106   left_join(county_cleanSchoolInfo2024, by="URN") %>%
107   drop_na(County,District,attainment_8_score,Town)
108
109 combinedSchool=bind_rows(combined2022,combined2023,combined2024)
110
111 # Clean the combinedSchool data by removing non-numeric values in attainment_8_score
112 combinedSchool_cleaned <- combinedSchool %>%
113   filter(!attainment_8_score %in% c("SUPP", "NE", "NA")) %>% # Remove unwanted codes
114   mutate(attainment_8_score = as.numeric(attainment_8_score)) %>% # Convert to numeric
115   drop_na(attainment_8_score) # Remove any remaining NA values
116
117 write_csv(combinedSchool_cleaned,"Cleaned Data/cleanSchool.csv")
118

```

cleanSchool.R x cleanSchool x

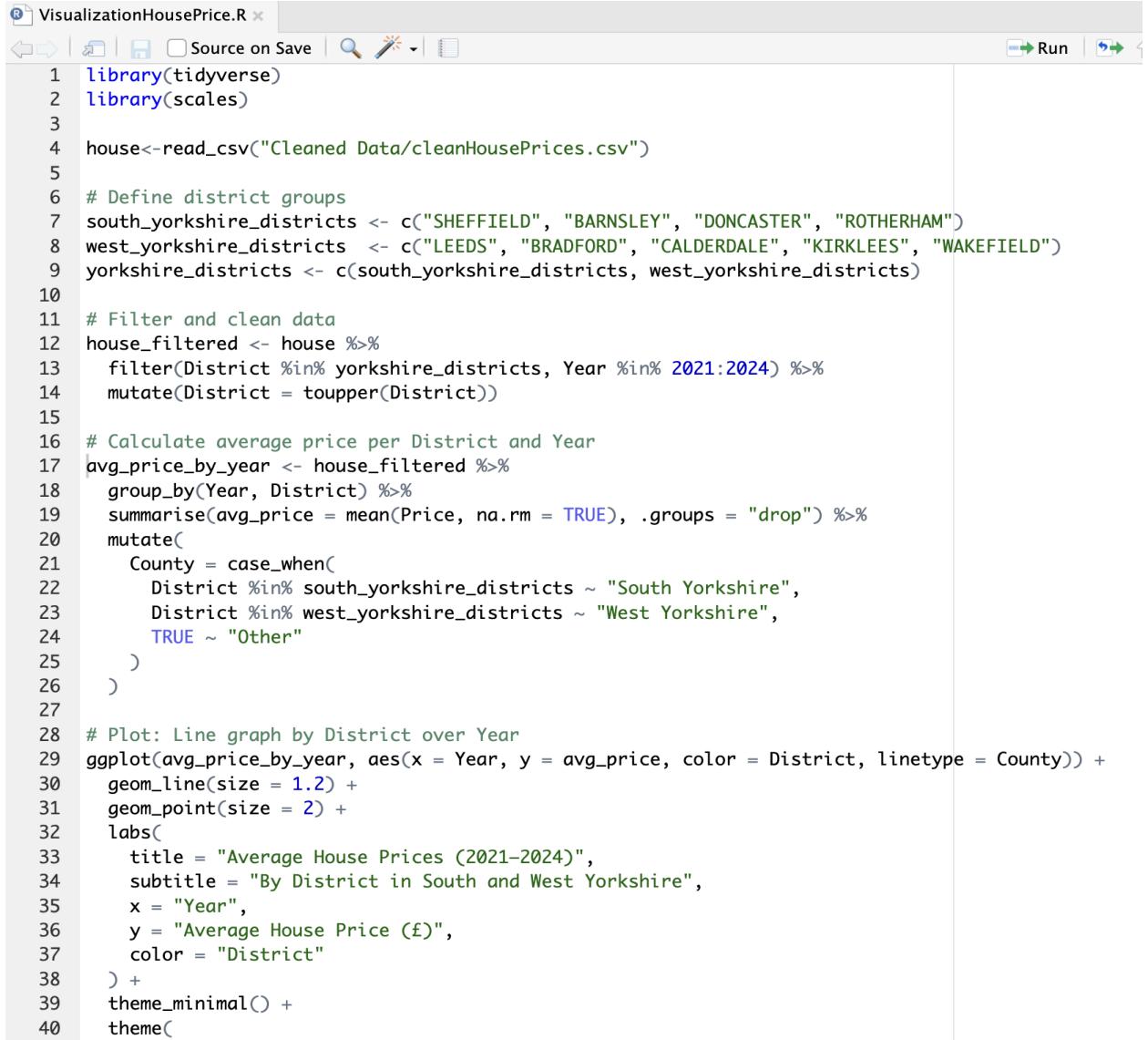
Filter

	URN	year	attainment_8_score	school_name	POSTCODE	shortPostcode	District	Town	County
140	130001	2022	37.0	Rivulet Dame High School	S10 5DT	S10	Sheffield	Sheffield	SOUTH YORKSHIRE
142	140415	2022	44.4	Outwood Academy City	S13 8SS	S13	Sheffield	Sheffield	SOUTH YORKSHIRE
143	140415	2022	44.4	Outwood Academy City	S13 8SS	S13	Sheffield	Sheffield	SOUTH YORKSHIRE
144	107166	2022	17.1	Sheffield High School	S10 2PE	S10	Sheffield	Sheffield	SOUTH YORKSHIRE
145	131896	2022	38.4	Sheffield Springs Academy	S12 2SF	S12	Sheffield	Sheffield	SOUTH YORKSHIRE
146	139167	2022	63.0	Silverdale School	S11 9QH	S11	Sheffield	Sheffield	SOUTH YORKSHIRE
147	145274	2022	51.7	Stocksbridge High School	S36 1FD	S36	Sheffield	Sheffield	SOUTH YORKSHIRE
148	145274	2022	51.7	Stocksbridge High School	S36 1FD	S36	Sheffield	Sheffield	SOUTH YORKSHIRE
149	138069	2022	60.6	Tapton School	S10 5RG	S10	Sheffield	Sheffield	SOUTH YORKSHIRE
150	145562	2022	43.9	Westfield School	S20 1HQ	S20	Sheffield	Sheffield	SOUTH YORKSHIRE
151	145562	2022	43.9	Westfield School	S20 1HQ	S20	Sheffield	Sheffield	SOUTH YORKSHIRE
152	145943	2022	39.1	Yewlands Academy	S35 8NN	S35	Sheffield	Sheffield	SOUTH YORKSHIRE
153	145943	2022	39.1	Yewlands Academy	S35 8NN	S35	Sheffield	Sheffield	SOUTH YORKSHIRE
154	145943	2022	39.1	Yewlands Academy	S35 8NN	S35	Sheffield	Sheffield	SOUTH YORKSHIRE
155	134429	2022	49.7	Al Mumin Primary and Secondary School	BD8 7DA	BD8	Bradford	Bradford	WEST YORKSHIRE
156	145173	2022	39.0	Appleton Academy	BD12 8AL	BD12	Bradford	Bradford	WEST YORKSHIRE
157	145173	2022	39.0	Appleton Academy	BD12 8AL	BD12	Bradford	Bradford	WEST YORKSHIRE
158	143112	2022	39.3	Beckfoot Oakbank	BD22 7DU	BD22	Bradford	Keighley	WEST YORKSHIRE
159	143112	2022	39.3	Beckfoot Oakbank	BD22 7DU	BD22	Bradford	Keighley	WEST YORKSHIRE
160	139975	2022	56.1	Beckfoot School	BD16 1EE	BD16	Bradford	Bingley	WEST YORKSHIRE
161	143114	2022	45.6	Beckfoot Thornton	BD13 3BH	BD13	Bradford	Bradford	WEST YORKSHIRE
162	143114	2022	45.6	Beckfoot Thornton	BD13 3BH	BD13	Bradford	Bradford	WEST YORKSHIRE
163	142031	2022	38.7	Beckfoot Upper Heaton	BD9 6AL	BD9	Bradford	Bradford	WEST YORKSHIRE
164	135229	2022	0.5	Beechcliffe Special School	BD20 6ED	BD20	Bradford	Keighley	WEST YORKSHIRE
165	138087	2022	50.1	Bella Vista Circle Academy	BD9 6NA	BD9	Bradford	Bradford	WEST YORKSHIRE

Exploratory Data Analysis

House Prices

1. Line Graphs



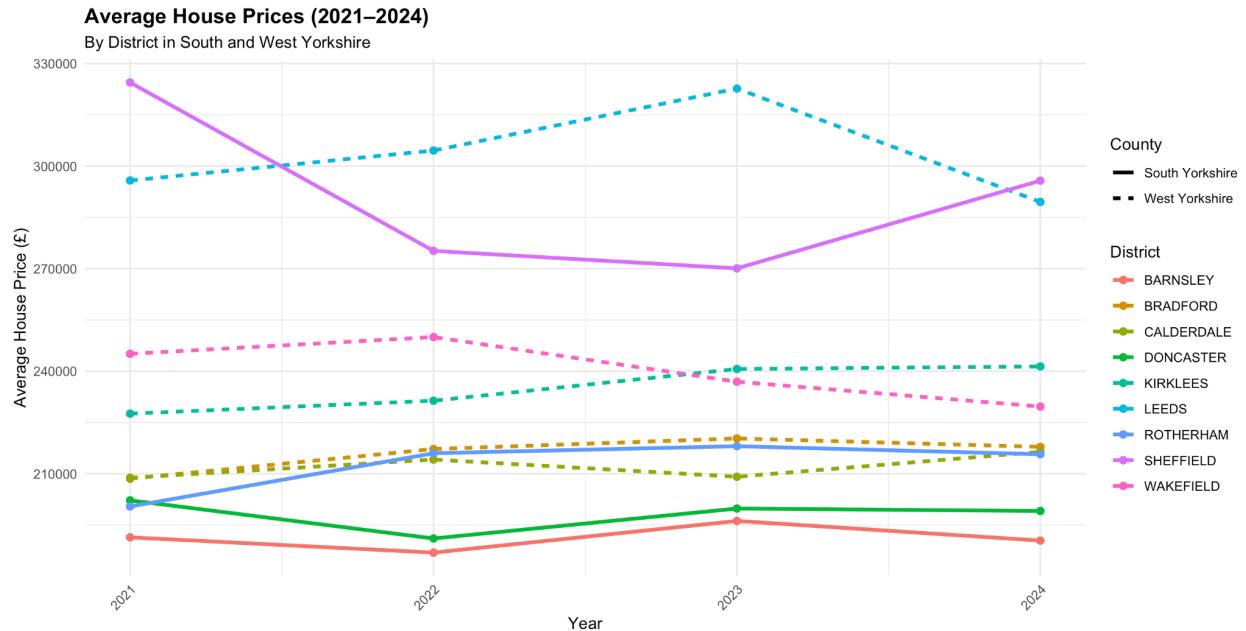
The screenshot shows an RStudio interface with a code editor window titled "VisualizationHousePrice.R". The code is written in R and performs the following steps:

- Imports the tidyverse and scales packages.
- Reads a CSV file named "cleanHousePrices.csv" into a data frame called "house".
- Defines district groups: south_yorkshire_districts, west_yorkshire_districts, and yorkshire_districts (which is a combination of the two).
- Filters and cleans the data to include districts in Yorkshire from 2021 to 2024, and converts district names to uppercase.
- Calculates the average price per District and Year.
- Groups the data by Year and District, then summarises the average price (mean) while dropping groups (groups = "drop"). It also creates a new column "County" using case_when to map districts to "South Yorkshire", "West Yorkshire", or "Other".
- Plots the average price over time for each district, using a line for the trend and points for individual years. The plot includes a title ("Average House Prices (2021–2024)"), subtitle ("By District in South and West Yorkshire"), x-axis label ("Year"), y-axis label ("Average House Price (£)"), and color mapping for districts.

```

1 library(tidyverse)
2 library(scales)
3
4 house<-read_csv("Cleaned Data/cleanHousePrices.csv")
5
6 # Define district groups
7 south_yorkshire_districts <- c("SHEFFIELD", "BARNSLEY", "DONCASTER", "ROOTHERHAM")
8 west_yorkshire_districts <- c("LEEDS", "BRADFORD", "CALDERDALE", "KIRKLEES", "WAKEFIELD")
9 yorkshire_districts <- c(south_yorkshire_districts, west_yorkshire_districts)
10
11 # Filter and clean data
12 house_filtered <- house %>%
13   filter(District %in% yorkshire_districts, Year %in% 2021:2024) %>%
14   mutate(District = toupper(District))
15
16 # Calculate average price per District and Year
17 avg_price_by_year <- house_filtered %>%
18   group_by(Year, District) %>%
19   summarise(avg_price = mean(Price, na.rm = TRUE), .groups = "drop") %>%
20   mutate(
21     County = case_when(
22       District %in% south_yorkshire_districts ~ "South Yorkshire",
23       District %in% west_yorkshire_districts ~ "West Yorkshire",
24       TRUE ~ "Other"
25     )
26   )
27
28 # Plot: Line graph by District over Year
29 ggplot(avg_price_by_year, aes(x = Year, y = avg_price, color = District, linetype = County)) +
30   geom_line(size = 1.2) +
31   geom_point(size = 2) +
32   labs(
33     title = "Average House Prices (2021–2024)",
34     subtitle = "By District in South and West Yorkshire",
35     x = "Year",
36     y = "Average House Price (£)",
37     color = "District"
38   ) +
39   theme_minimal() +
40   theme(

```



Interpretation:

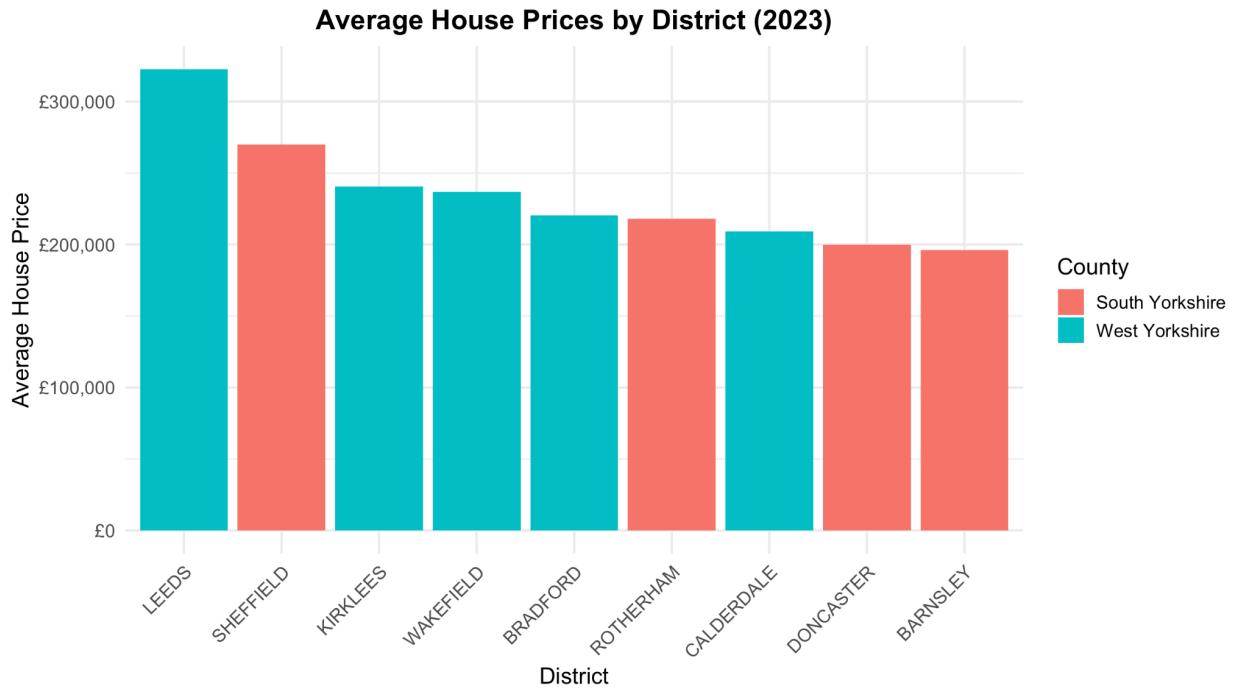
The graph charts average house prices (2021–2024) by South and West Yorkshire districts. Leeds was the most expensive throughout, peaking in 2023, while Barnsley was the least expensive, with a slight drop in 2024. South Yorkshire councils, indicated by solid lines, were generally less expensive than West Yorkshire councils (dashed lines). Notably, Sheffield had a steep dip from 2021 to 2022 but began recovering by 2024. West Yorkshire overall showed greater price stability and consistent growth, while South Yorkshire was more fluctuant, reflecting varied investment potential within the area.

2. Bar chart

```
# Filter for 2023 and relevant districts
house_2023 <- house %>%
  filter(District %in% yorkshire_districts, Year == 2023) %>%
  mutate(
    District = toupper(District),
    County = case_when(
      District %in% south_yorkshire_districts ~ "South Yorkshire",
      District %in% west_yorkshire_districts ~ "West Yorkshire"
    )
  )

# Compute average price per district
avg_price_2023 <- house_2023 %>%
  group_by(District, County) %>%
  summarise(avg_price = mean(Price, na.rm = TRUE), .groups = "drop")

# Bar chart
ggplot(avg_price_2023, aes(x = reorder(District, -avg_price), y = avg_price, fill = County)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  scale_y_continuous(labels = label_number(prefix = "£", big.mark = ",")) +
  labs(
    title = "Average House Prices by District (2023)",
    x = "District",
    y = "Average House Price",
    fill = "County"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
)
```



Interpretation:

The bar chart shows average house prices per district in 2023. The highest average price over £300,000 was in Leeds (West Yorkshire), then Sheffield (South Yorkshire). Barnsley and Doncaster (both South Yorkshire) had the lowest average prices, and thus the cheapest for investors. West Yorkshire districts generally had higher prices than South Yorkshire districts, except for Sheffield alone. This distribution highlights regional price variations and suggests greater affordability in South Yorkshire, while Leeds is the most expensive district in both counties.

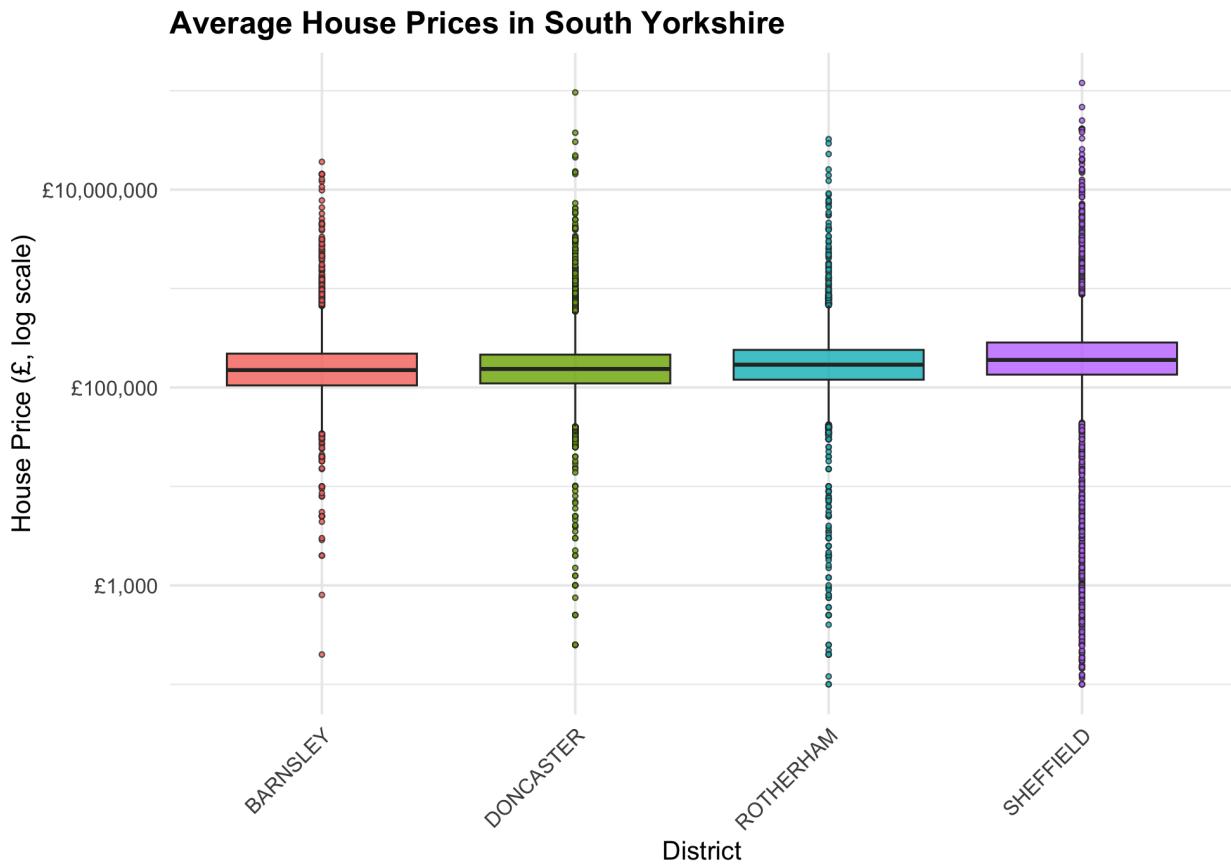
3.Boxplot

```
# Filter and annotate the data
house_filtered <- house %>%
  filter(District %in% yorkshire_districts) %>%
  mutate(
    District = toupper(District),
    County = case_when(
      District %in% south_yorkshire_districts ~ "South Yorkshire",
      District %in% west_yorkshire_districts ~ "West Yorkshire"
    )
  )

# Separate data for each county
south_data <- house_filtered %>% filter(County == "South Yorkshire")
west_data <- house_filtered %>% filter(County == "West Yorkshire")

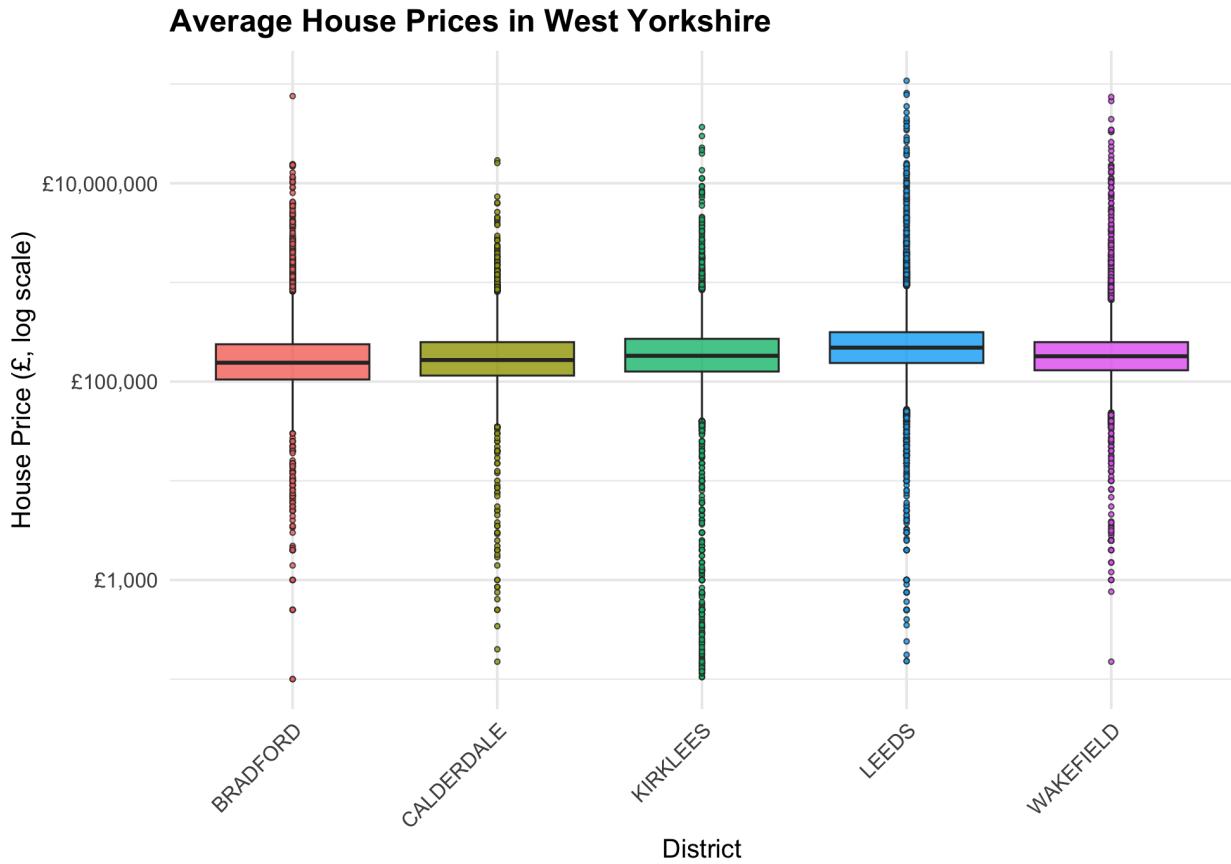
# South Yorkshire plot
p_south <- ggplot(south_data, aes(x = District, y = Price, fill = District)) +
  geom_boxplot(outlier.shape = 21, outlier.size = 1, alpha = 0.8) +
  scale_y_log10(labels = scales::label_number(prefix = "£", big.mark = ",")) +
  labs(
    title = "Average House Prices in South Yorkshire",
    x = "District",
    y = "House Price (£, log scale)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(face = "bold"),
    legend.position = "none"
  )

# West Yorkshire plot
p_west <- ggplot(west_data, aes(x = District, y = Price, fill = District)) +
  geom_boxplot(outlier.shape = 21, outlier.size = 1, alpha = 0.8) +
  scale_y_log10(labels = scales::label_number(prefix = "£", big.mark = ",")) +
  labs(
    title = "Average House Prices in West Yorkshire",
    x = "District",
    y = "House Price (£, log scale)"
  ) +
```



Interpretation:

The chart shows average house prices in South Yorkshire (Barnsley, Doncaster, Rotherham, Sheffield) on a log scale. Barnsley and Doncaster have the lowest averages, around £100,000, with wide price ranges. Rotherham's average is slightly higher, also around £100,000, with a broad distribution. Sheffield has the highest average, exceeding £100,000, with prices extending up to £10,000,000. The data indicates significant variation, with Sheffield showing the greatest upper range, while Barnsley and Doncaster have more concentrated lower values. Overall, house prices vary widely across districts, with Sheffield standing out as the most expensive.



Interpretation:

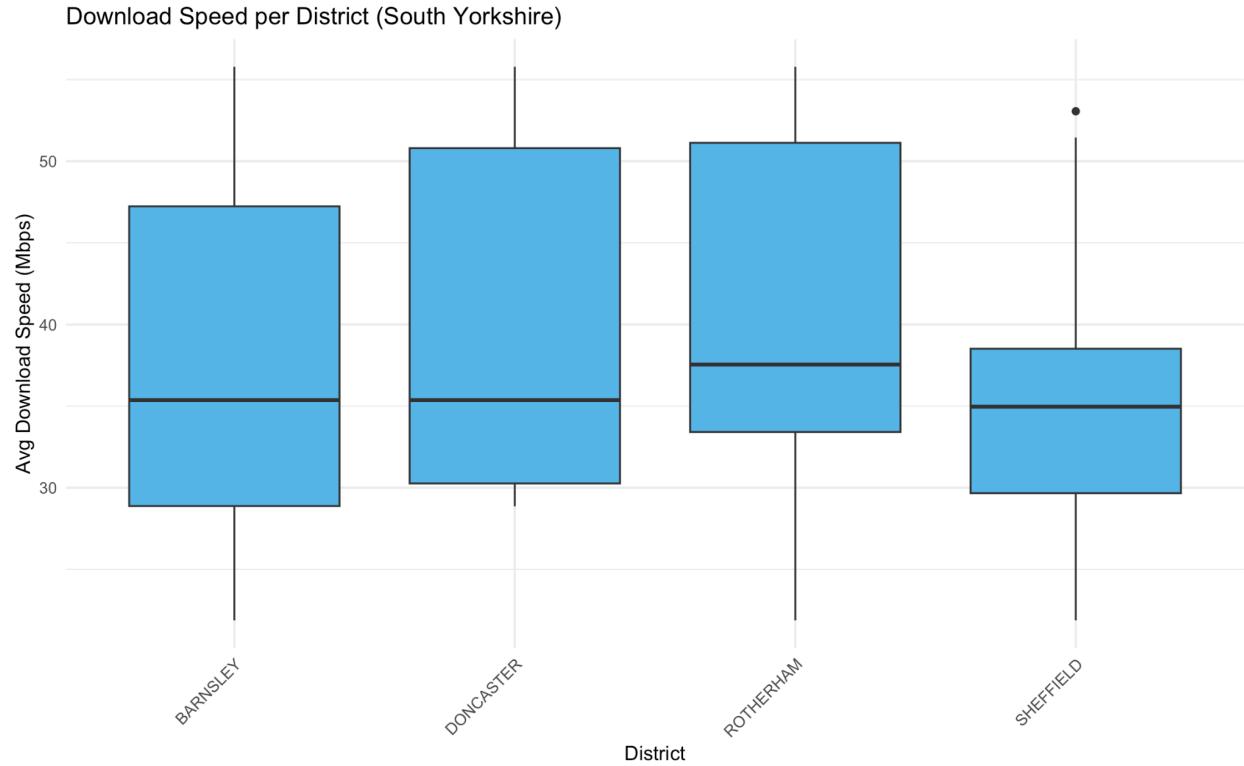
The chart displays average house prices in West Yorkshire (Bradford, Calderdale, Kirklees, Leeds, Wakefield) on a log scale. All districts have averages around £100,000, with wide price ranges. Bradford and Calderdale show the broadest spreads, with prices reaching up to £10,000,000. Kirklees and Leeds have similar distributions, while Wakefield's range extends highest, also nearing £10,000,000. The data suggests significant price variability, with all districts having a central tendency around £100,000, but upper limits vary widely. Overall, house prices in West Yorkshire exhibit diverse extremes, with no single district dominating the high end.

Broadband Speed

1. Boxplots

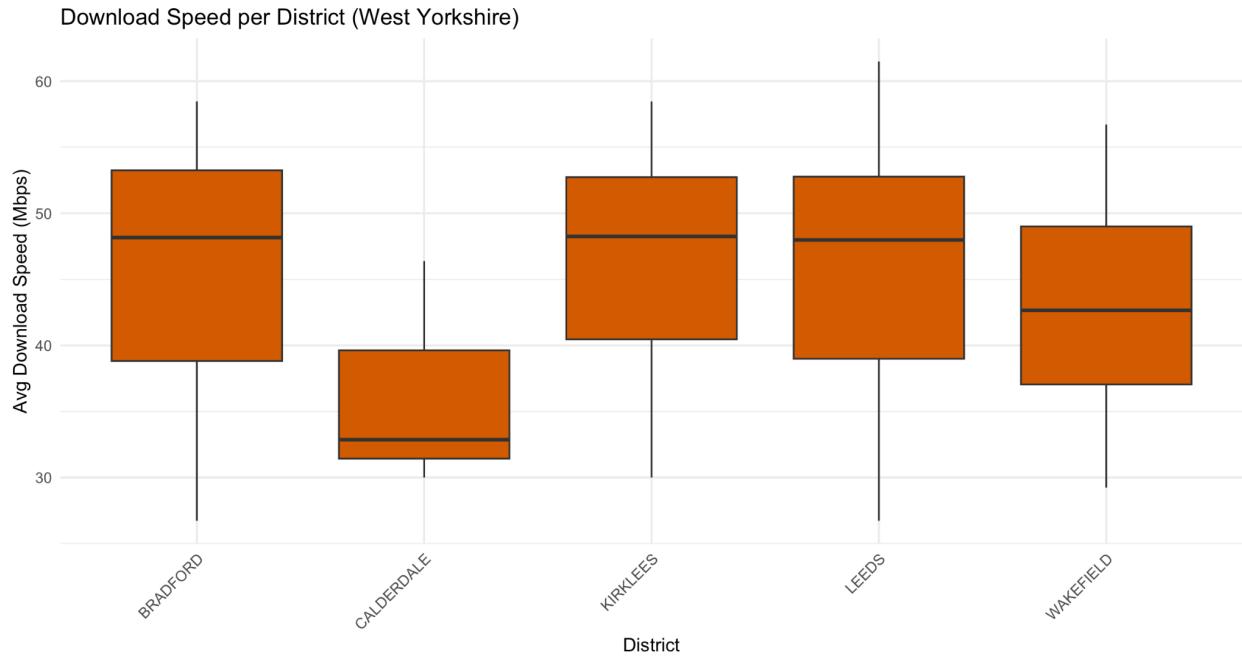
```
#Boxplot
final_BroadbandPerformance %>%
  filter(County == "SOUTH YORKSHIRE") %>%
  ggplot(aes(x = District, y = AvgDownload)) +
  geom_boxplot(fill = "#56B4E9") +
  labs(
    title = "Download Speed per District (South Yorkshire)",
    x = "District",
    y = "Avg Download Speed (Mbps)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

final_BroadbandPerformance %>%
  filter(County == "WEST YORKSHIRE") %>%
  ggplot(aes(x = District, y = AvgDownload)) +
  geom_boxplot(fill = "#D55E00") +
  labs(
    title = "Download Speed per District (West Yorkshire)",
    x = "District",
    y = "Avg Download Speed (Mbps)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Interpretation:

The box plot presents average download speeds (Mbps) in four South Yorkshire areas: Barnsley, Doncaster, Rotherham, and Sheffield. A single box indicates the interquartile range (IQR), and the horizontal line within it is the median speed. The whiskers extend to the lowest and highest speeds, and the outliers are represented by dots. Barnsley, Doncaster, and Rotherham all have median speeds around 40 Mbps with similar IQRs, indicating consistent performance. Sheffield shows a lower median speed at around 35 Mbps with a larger IQR and an outlier for variability. The plot places Sheffield as having the lowest and most variable of the districts' download speeds.



Interpretation:

The box plot shows mean download speeds (Mbps) for five West Yorkshire boroughs: Bradford, Calderdale, Kirklees, Leeds, and Wakefield. Each box shows the interquartile range (IQR), and a line shows the median speed. Bradford and Kirklees both report median speeds of around 45-50 Mbps with similar IQRs. Calderdale shows a lower median of around 35-40 Mbps with a smaller IQR. Leeds and Wakefield have medians of around 40-45 Mbps, Wakefield fluctuating slightly. Bradford and Kirklees are shown to have the highest and most variable download speeds, followed by Calderdale being the lowest.

2.Barchart

```

library(ggplot2)
library(tidyverse)

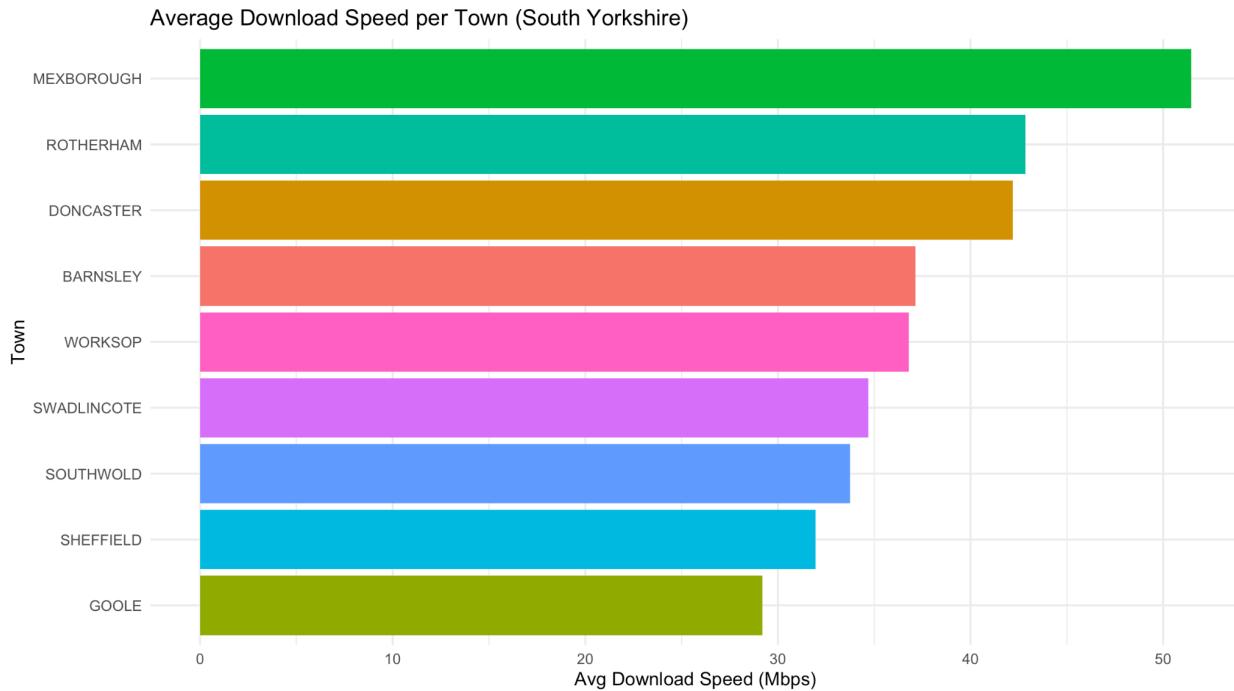
final_BroadbandPerformance<-read_csv("Cleaned Data/cleanBroadbandPerformance.csv")

broadband_by_town <- final_BroadbandPerformance %>%
  group_by(Town, County) %>%
  summarise(AvgDownload = mean(AvgDownload, na.rm = TRUE), .groups = "drop")

#Barchart
broadband_by_town %>%
  filter(County == "SOUTH YORKSHIRE") %>%
  ggplot(aes(x = reorder(Town, AvgDownload), y = AvgDownload, fill = Town)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  labs(
    title = "Average Download Speed per Town (South Yorkshire)",
    x = "Town",
    y = "Avg Download Speed (Mbps)"
  ) +
  theme_minimal()

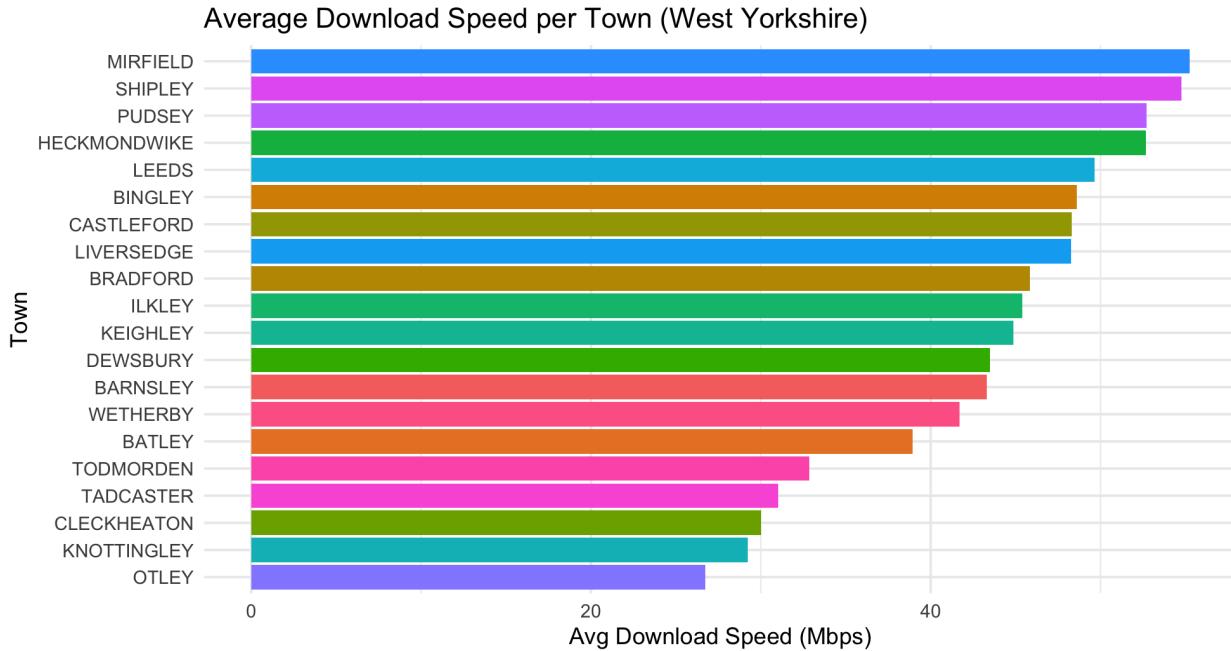
broadband_by_town %>%
  filter(County == "WEST YORKSHIRE") %>%
  ggplot(aes(x = reorder(Town, AvgDownload), y = AvgDownload, fill = Town)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  labs(
    title = "Average Download Speed per Town (West Yorkshire)",
    x = "Town",
    y = "Avg Download Speed (Mbps)"
  ) +
  theme_minimal()

```



Interpretation:

The bar chart displays average download speeds (Mbps) across nine towns in South Yorkshire. Mexborough leads with the highest speed, exceeding 50 Mbps, followed by Rotherham at around 45 Mbps. Doncaster and Barnsley show speeds near 40 Mbps, while Worksop and Swadlincote are slightly lower, around 35-37 Mbps. Southwold and Sheffield have speeds around 30-32 Mbps, and Goole records the lowest at approximately 30 Mbps. The chart indicates a range of performance, with Mexborough offering the best connectivity and Goole the least. Overall, speeds vary significantly, with northern towns generally outperforming southern ones in this region.



Interpretation:

The bar chart shows average download speeds (Mbps) across 16 towns in West Yorkshire. Mirfield and Shipley lead with speeds near or above 40 Mbps, followed closely by Pudsey and Heckmondwike, also around 40 Mbps. Leeds, Bingley, and Castleford range between 35-40 Mbps, while Liversedge, Bradford, Ilkley, Keighley, and Dewsbury fall around 30-35 Mbps. Wetherby, Batley, Todmorden, Tadcaster, Cleckheaton, Knottingley, and Otley show speeds between 25-30 Mbps. The chart highlights a gradient of performance, with northern towns like Mirfield and Shipley offering the fastest speeds, while southern towns like Otley and Knottingley lag, indicating regional connectivity disparities.

Crime Rates

1.Boxplot

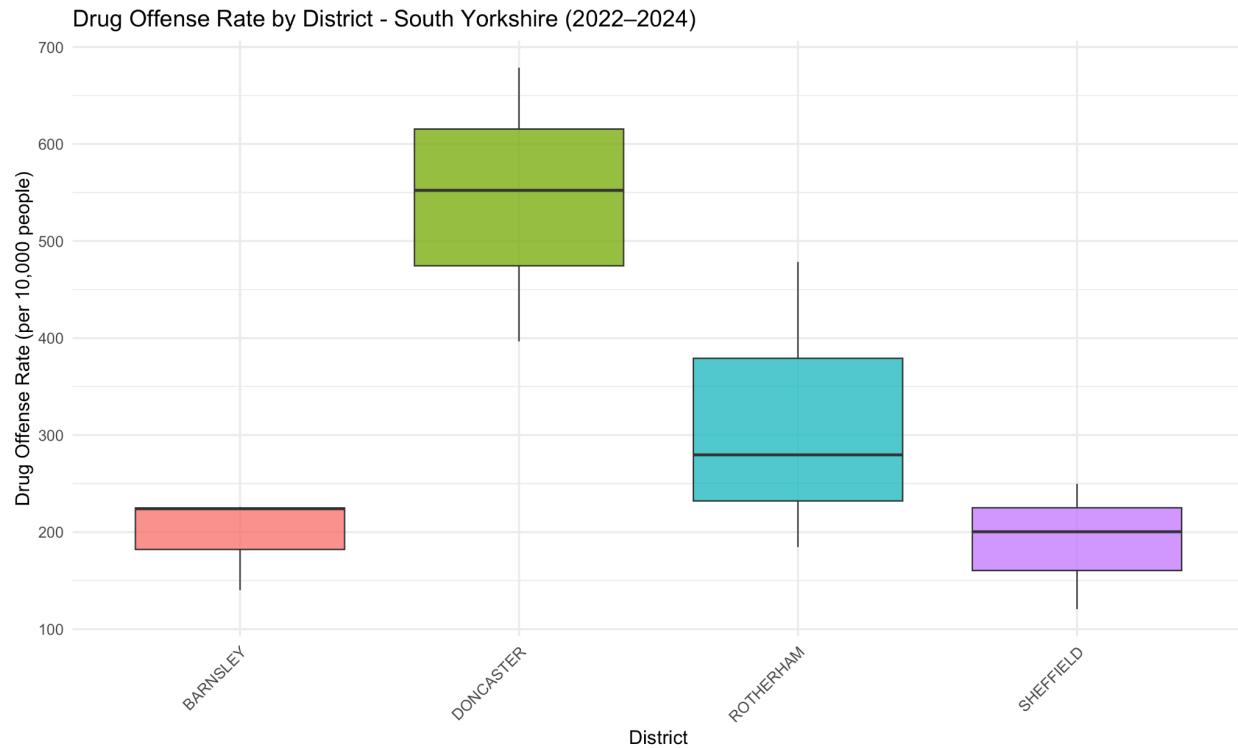
```
# Define district groups
south_yorkshire <- c("SHEFFIELD", "BARNSLEY", "ROTHERHAM", "DONCASTER")
west_yorkshire <- c("LEEDS", "BRADFORD", "WAKEFIELD", "KIRKLEES", "CALDERDALE")

# Classify CountyGroup
crime_rates <- crime_rates %>%
  mutate(
    CountyGroup = case_when(
      toupper(District_Town) %in% south_yorkshire ~ "South Yorkshire",
      toupper(District_Town) %in% west_yorkshire ~ "West Yorkshire",
      TRUE ~ "Other"
    )
  ) %>%
  filter(CountyGroup != "Other")

View(crime_rates)

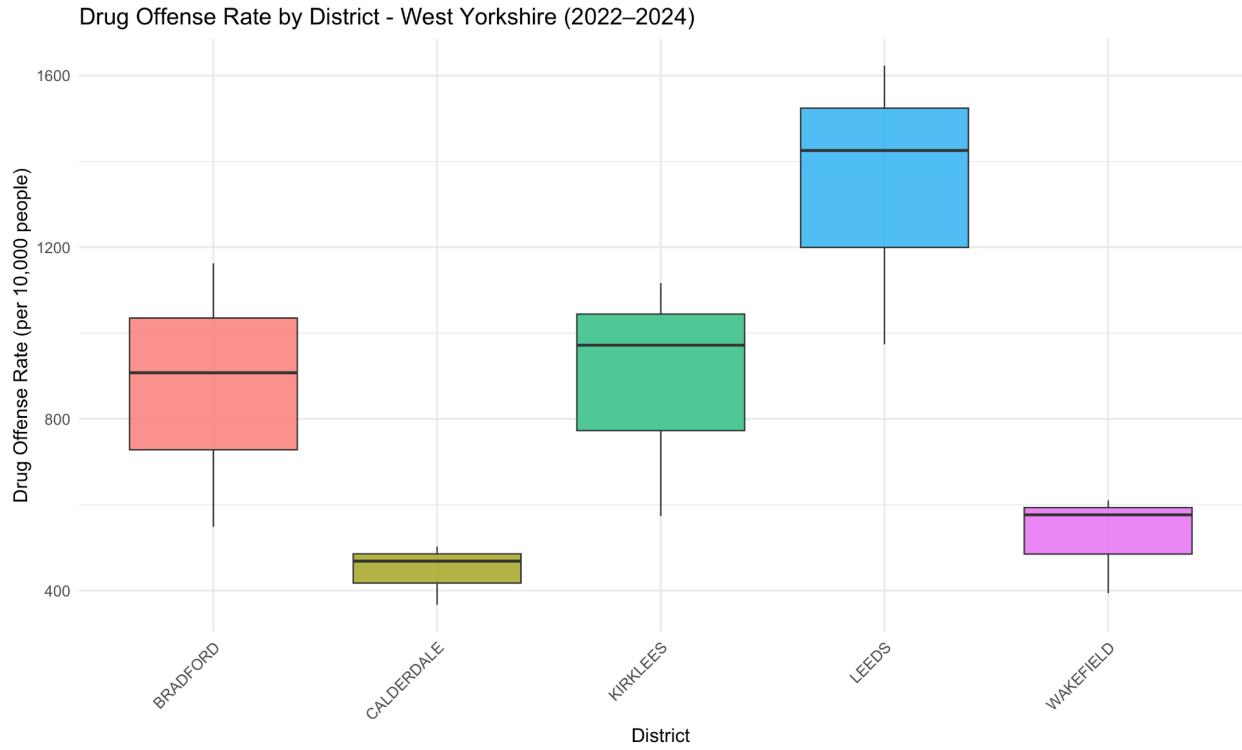
# South Yorkshire
south_plot <- crime_rates %>%
  filter(County_Town == "SOUTH YORKSHIRE") %>%
  ggplot(aes(x = District_Town, y = DrugOffenseRate, fill = District_Town)) +
  geom_boxplot(alpha = 0.8, outlier.shape = 21) +
  labs(
    title = "Drug Offense Rate by District - South Yorkshire (2022-2024)",
    x = "District",
    y = "Drug Offense Rate (per 10,000 people)"
  ) +
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  guides(fill = "none")

# West Yorkshire
west_plot <- crime_rates %>%
  filter(County_Town == "WEST YORKSHIRE") %>%
  ggplot(aes(x = District_Town, y = DrugOffenseRate, fill = District_Town)) +
  geom_boxplot(alpha = 0.8, outlier.shape = 21) +
  labs(
    title = "Drug Offense Rate by District - West Yorkshire (2022-2024)",
    x = "District",
```



Interpretation:

Box plot illustrates drug offense rate per 10,000 persons in four South Yorkshire local districts (2022-2024). Doncaster possesses the highest median rate at around 600, with a wide interquartile range (IQR) and extreme variability. Rotherham has a median of around 350 with a fairly moderate IQR. Barnsley has a median of around 200 with a slender IQR, indicating stability at a low rate. Sheffield has the lowest median value, about 200, with a similar close IQR. Whiskers and outliers (e.g., in Doncaster and Rotherham) show occasional higher rates. The values show that Doncaster has the most drug offense issues, and Sheffield and Barnsley are lower.



Interpretation:

The box plot illustrates the rate of drug offense per 10,000 persons in five West Yorkshire areas (2022-2024). Leeds has the highest median rate of about 1400 with wide interquartile range (IQR) and hence, large variability. Bradford's median rate is nearly 900 with wide IQR. Kirklees is about 800 with average IQR. Calderdale's median is the lowest at about 400 with narrow IQR, which suggests uniformity. Wakefield's median is around 500 with a close IQR. Outliers such as Leeds are pointers for the infrequent peak. Leeds and Bradford have the most drug offenses according to the data displayed, Calderdale having the lowest.

2.Radar chart

```

#Radar chart for south yorkshire(2023-06)
selected_year <- 2023
selected_month <- "2023-06"
selected_county <- "SOUTH YORKSHIRE"

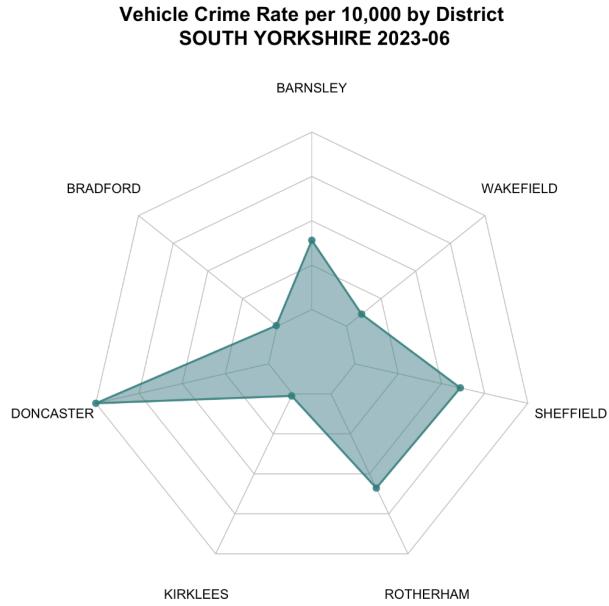
crime_filtered <- crime %>%
  filter(CrimeType == "Vehicle crime", Year == selected_year, Month == selected_month, County
  left_join(
    towns %>%
      select(shortPostcode, District_Town = District, County_Town = County,
             Population2023),
    by = "shortPostcode"
  )

crime_rates <- crime_filtered %>%
  mutate(
    Population = case_when(
      Year == 2023 ~ Population2023,
      TRUE ~ NA_real_
    )
  ) %>%
  drop_na(Population, District_Town) %>%
  group_by(District_Town) %>%
  summarise(
    VehicleCrimeCount = n(),
    Population = first(Population),
    VehicleCrimeRate = (VehicleCrimeCount / Population) * 10000,
    .groups = "drop"
  )

# Prepare data for radar chart
# Radar chart needs max, min rows on top
max_rate <- max(crime_rates$VehicleCrimeRate, na.rm = TRUE)
radar_data <- rbind(
  max = rep(max_rate, nrow(crime_rates)),
  min = rep(0, nrow(crime_rates)),
  crime_rates$VehicleCrimeRate
)

# Plot radar chart
radarchart(radar_data,
  pcol = rgb(0.2, 0.5, 0.5, 0.9),
  pfcol = rgb(0.2, 0.5, 0.5, 0.5),
  plwd = 2,
  cglcol = "grey",
  cglty = 1,
  axislabcol = "grey",
  axislabels = seq(0, ceiling(max_rate), length.out = 5),
  cglwd = 0.8,
  vlcex = 0.8,
  title = paste("Vehicle Crime Rate per 10,000 by District\n",
               selected_county, selected_month)
)

```



Interpretation:

The radar chart graphically represents the vehicle crime rate per 10,000 population for six districts in South Yorkshire in 2023-06. Doncaster's rate is highest, a steep spike radiating out, indicating that it's a serious issue. Barnsley and Wakefield's rates are moderate, Barnsley more so. Sheffield and Kirklees have lower rates, Sheffield lowest. Rotherham's rate is mid-range with an even distribution. The graph suggests a wide variability in car crime, with the highest being Doncaster and the lowest being Sheffield. The radial distribution shows discrepancies, emphasizing Doncaster's outlier status among the regions.

3.Pie chart

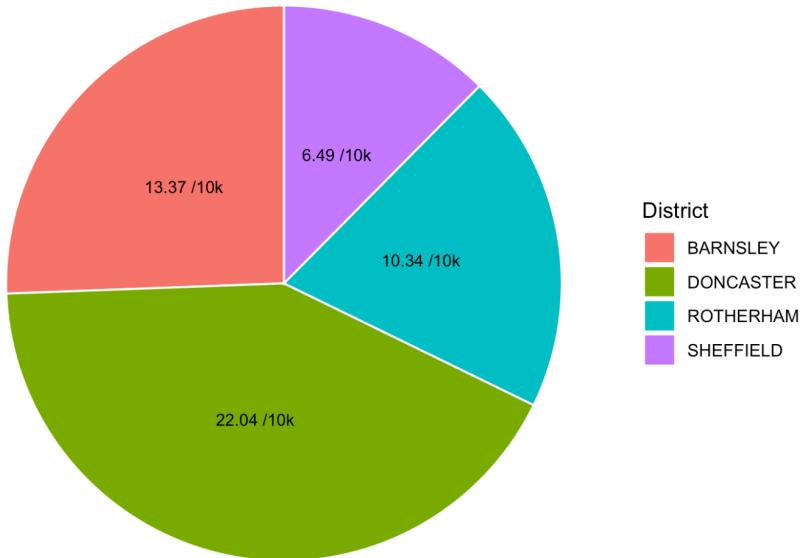
```
# Define South Yorkshire districts
south_yorkshire_districts <- c("SHEFFIELD", "BARNESLEY", "ROOTHERHAM", "DONCASTER")

# Parameters
selected_year <- 2024
selected_month <- "2024-06"
selected_county <- "SOUTH YORKSHIRE"

# Filter for Robbery crimes
crime_filtered <- crime %>%
  filter(
    CrimeType == "Robbery",
    Year == selected_year,
    Month == selected_month,
    County == selected_county
  ) %>%
  left_join(
    towns %>%
      select(shortPostcode, District_Town = District, County_Town = County, Population2024),
    by = "shortPostcode"
  ) %>%
  filter(District_Town %in% south_yorkshire_districts) # <- Filter only South Yorkshire dist

# Compute Robbery Rates
crime_rates <- crime_filtered %>%
  mutate(
    Population = case_when(
      Year == 2024 ~ Population2024,
      TRUE ~ NA_real_
    )
  ) %>%
  drop_na(Population, District_Town) %>%
  group_by(District_Town) %>%
  summarise(
    Population = first(Population),
    RobberyCount = n(),
    .groups = "drop"
  ) %>%
  mutate(robberryRate = (RobberyCount / Population) * 10000)
```

```
# Pie Chart
ggplot(crime_rates, aes(x = "", y = robberyRate, fill = District_Town)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  geom_text(aes(label = paste0(round(robberyRate, 2), " /10k")),
            position = position_stack(vjust = 0.5),
            color = "black",
            size = 3) +
  coord_polar("y", start = 0) +
  labs(
    title = paste("Robbery Rate by District - ", selected_county, "(", selected_month, ")"),
    fill = "District"
  ) +
  theme_void() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    legend.position = "right"
  )
```

Robbery Rate by District - SOUTH YORKSHIRE (2024-06)**Interpretation:**

The pie chart shows the rate of robbery per 10,000 for four South Yorkshire districts in 2024-06. Doncaster has the highest at 22.04/10k, occupying the largest slice. Then there is Barnsley at 13.37/10k, followed by Rotherham with 10.34/10k. Sheffield has the lowest at 6.49/10k, occupying the smallest slice. The chart displays a vast difference, with Doncaster experiencing the highest rate of robberies, nearly double the rate of Sheffield. Barnsley and Rotherham have medium levels, with Barnsley barely ahead. This shows that Doncaster is robbed the most out of all the areas, with Sheffield being the safest of them all.

4. Line chart

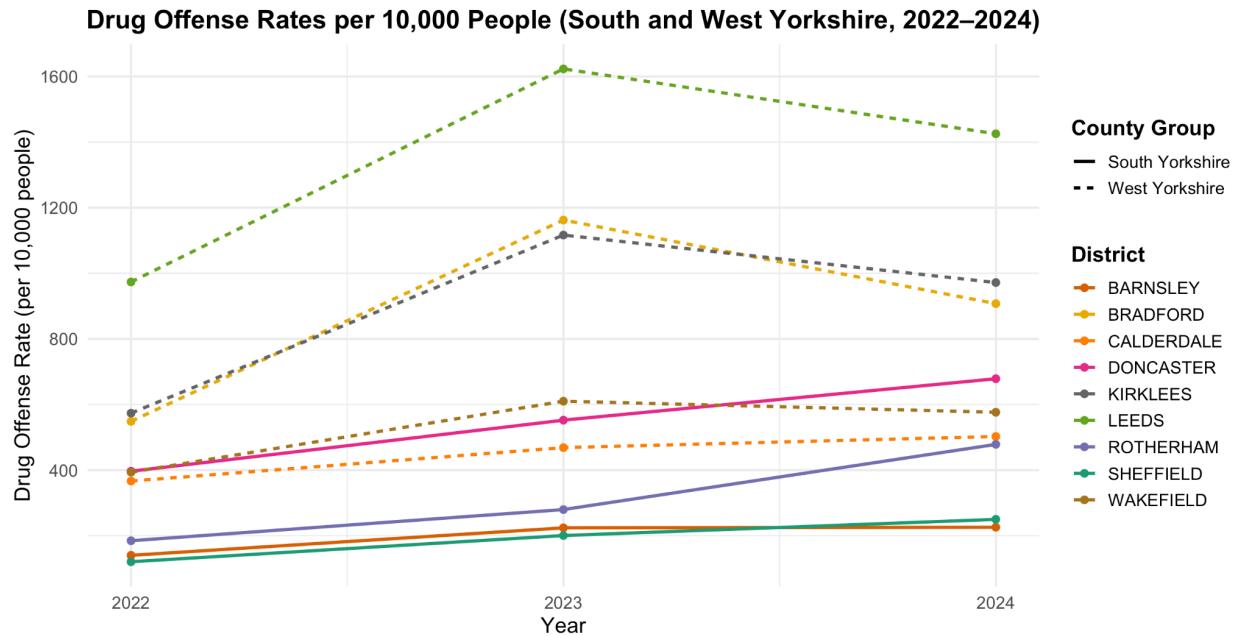
```

crime_rates <- crime_rates %>%
  mutate(
    CountyGroup = case_when(
      toupper(County_Town) %in% south_yorkshire ~ "SOUTH YORKSHIRE",
      toupper(County_Town) %in% west_yorkshire ~ "WEST YORKSHIRE",
      TRUE ~ "Other"
    )
  ) %>%
  filter(CountyGroup != "Other")

# Line chart for Drug Offense Rates (South and West Yorkshire, 2022–2024)
line_plot <- crime_rates %>%
  ggplot(aes(x = Year, y = DrugOffenseRate, color = District_Town, linetype = CountyGroup)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  labs(
    title = "Drug Offense Rates per 10,000 People (South and West Yorkshire, 2022–2024)",
    x = "Year",
    y = "Drug Offense Rate (per 10,000 people)",
    color = "District",
    linetype = "County Group"
  ) +
  theme_minimal(base_size = 14) +
  scale_x_continuous(breaks = c(2022, 2023, 2024)) +
  scale_color_manual(values = c(
    "SHEFFIELD" = "#1b9e77", "BARNESLEY" = "#d95f02", "ROOTHERHAM" = "#7570b3", "DONCASTER" = "#666666",
    "LEEDS" = "#66a61e", "BRADFORD" = "#e6ab02", "WAKEFIELD" = "#a6761d", "KIRKLEES" = "#666666"
  )) +
  theme(
    legend.position = "right",
    legend.title = element_text(face = "bold"),
    plot.title = element_text(face = "bold", hjust = 0.5)
  )

# Display the plot
print(line_plot)

```



Interpretation:

The line graph tracks drug offence rates per 10,000 people in South and West Yorkshire (2022-2024). South Yorkshire (solid line) registers a peak of 1600 in 2023, which drops to 1400 in 2024. West Yorkshire (dashed line) keeps on rising from 1000 to 1200. On the district level, Leeds registered a peak of 1400 in 2023, then declining, while Bradford and Kirklees rose from 400 to 600-700. Barnsley, Calderdale, Doncaster, Rotherham, Sheffield, and Wakefield record moderate growths of between 300 and 600. The figures show the total growth in 2023, which has the highest rates in South Yorkshire and Leeds, and then a corresponding decline in 2024.

Schools

1.Boxplot SY

```
library(tidyverse)

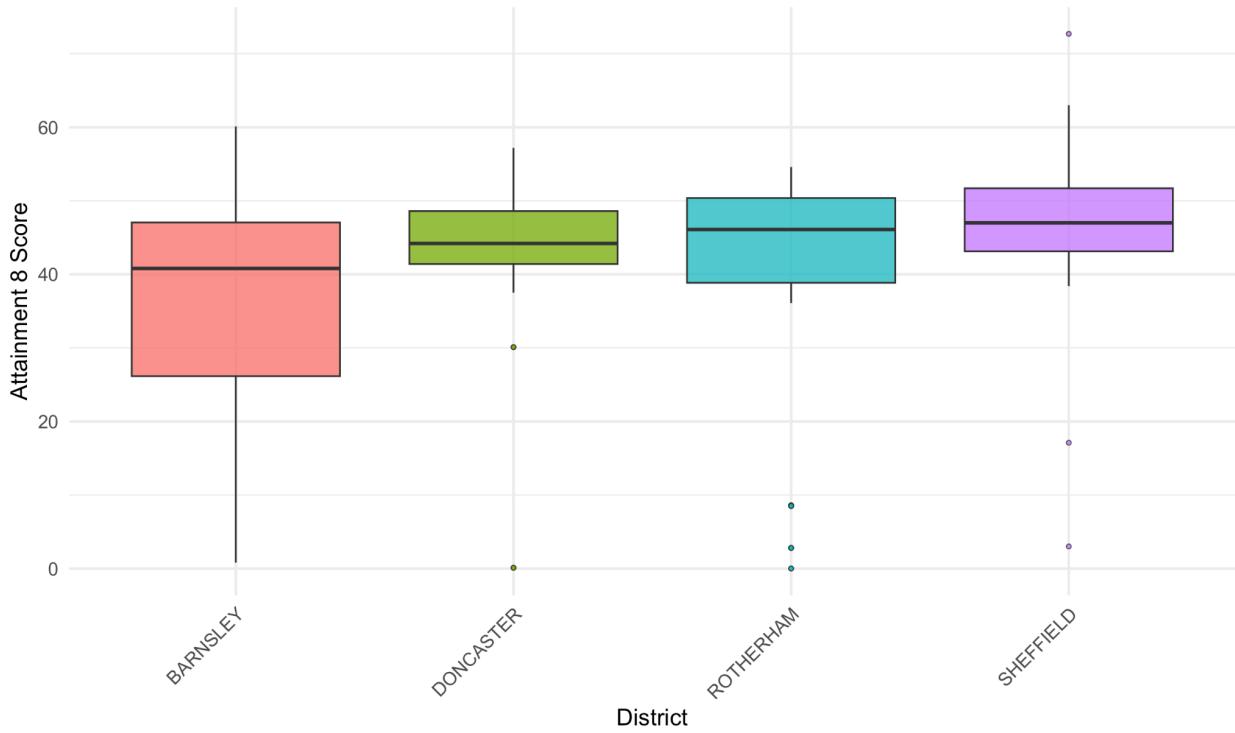
school<-read_csv("Cleaned Data/cleanSchool.csv")

# Define South Yorkshire districts
south_yorkshire_districts <- c("SHEFFIELD", "BARNESLEY", "DONCASTER", "ROOTHERHAM")

# Filter for 2022 and South Yorkshire
school_filtered_sy <- school %>%
  filter(year == 2022, toupper(District) %in% south_yorkshire_districts) %>%
  mutate(District = toupper(District))

# Plot: Boxplot of Attainment 8 score by District
ggplot(school_filtered_sy, aes(x = District, y = attainment_8_score, fill = District)) +
  geom_boxplot(outlier.shape = 21, outlier.size = 1, alpha = 0.8) +
  labs(
    title = "Attainment 8 Scores (2022) - South Yorkshire",
    x = "District",
    y = "Attainment 8 Score"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(face = "bold"),
    legend.position = "none"
  )
)
```

Attainment 8 Scores (2022) - South Yorkshire



Interpretation:

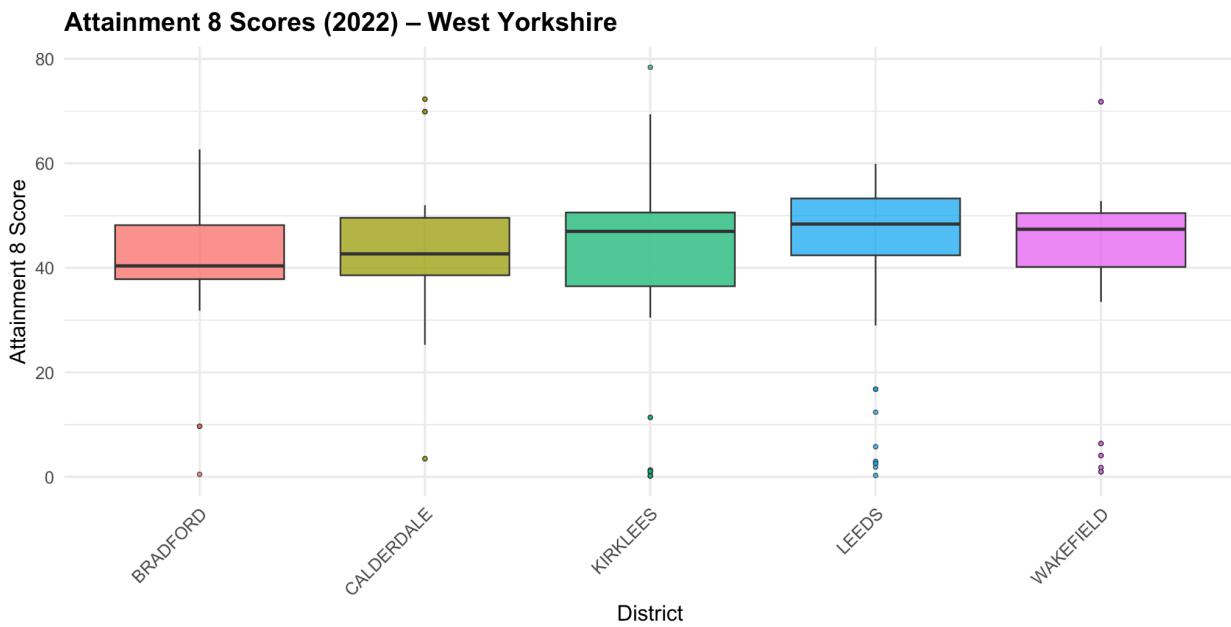
The box plot illustrates 2022 attainment 8 scores for four South Yorkshire districts: Barnsley, Doncaster, Rotherham, and Sheffield. The lowest median score is in Barnsley with a score of about 40 and a narrow interquartile range (IQR). Doncaster's median is slightly greater, approximately 45, and has an outlier at less than 20. Rotherham's median is about 50, quite a moderate IQR, and an outlier of about 10. Sheffield leads the way with a median of about 50, a wide IQR, and an outlier of more than 60. The plot shows that Sheffield and Rotherham performed best, while Barnsley lagged behind, with variability indicating skewed school performance in districts.

2.Boxplot WY

```
# Define West Yorkshire districts
west_yorkshire_districts <- c("LEEDS", "BRADFORD", "CALDERDALE", "KIRKLEES", "WAKEFIELD")

# Filter for 2022 and West Yorkshire districts
school_filtered_wy <- school %>%
  filter(year == 2022, toupper(District) %in% west_yorkshire_districts) %>%
  mutate(District = toupper(District))

# Plot: Boxplot of Attainment 8 score by District (West Yorkshire)
ggplot(school_filtered_wy, aes(x = District, y = attainment_8_score, fill = District)) +
  geom_boxplot(outlier.shape = 21, outlier.size = 1, alpha = 0.8) +
  labs(
    title = "Attainment 8 Scores (2022) – West Yorkshire",
    x = "District",
    y = "Attainment 8 Score"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(face = "bold"),
    legend.position = "none"
  )
)
```



Interpretation:

The box plot reveals achievement 8 scores for 2022 in five West Yorkshire localities: Bradford, Calderdale, Kirklees, Leeds, and Wakefield. Bradford is lowest at around 40, with a narrow interquartile range (IQR) and an outlier at around 0. Calderdale's is at around 45, with an outlier at around 70. Kirklees is highest at a level at around 50, with an expansive IQR. Leeds and Wakefield have medians of approximately 50, with Wakefield having one outlier in excess of 70. The graph shows Kirklees, Leeds, and Wakefield performing best, but Bradford falling behind due to variability in unpredictable education outcomes in districts.

3. Line Graph

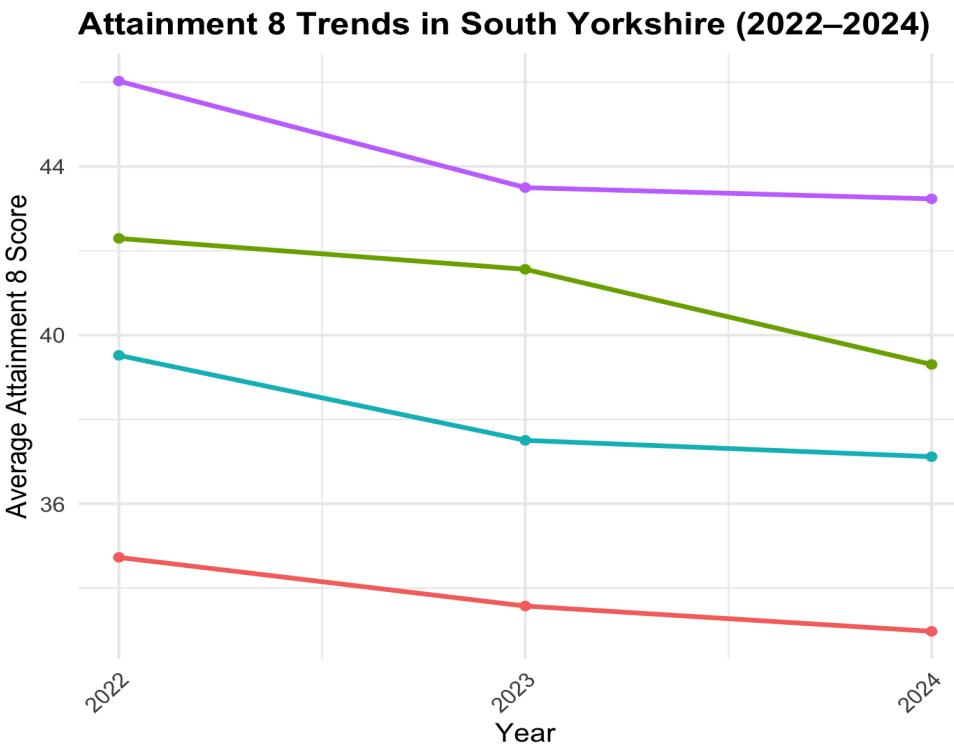
```
# Filter and label counties
school_filtered <- school %>%
  filter(toupper(District) %in% yorkshire_districts, year %in% c(2022, 2023, 2024)) %>%
  mutate(
    District = toupper(District),
    County = case_when(
      District %in% south_yorkshire_districts ~ "South Yorkshire",
      District %in% west_yorkshire_districts ~ "West Yorkshire"
    )
  )

# Summarize Attainment 8 scores
school_summary <- school_filtered %>%
  group_by(County, District, year) %>%
  summarise(AvgAttainment8 = mean(attainment_8_score, na.rm = TRUE), .groups = "drop")

# Split data
south_school <- school_summary %>% filter(County == "South Yorkshire")
west_school <- school_summary %>% filter(County == "West Yorkshire")

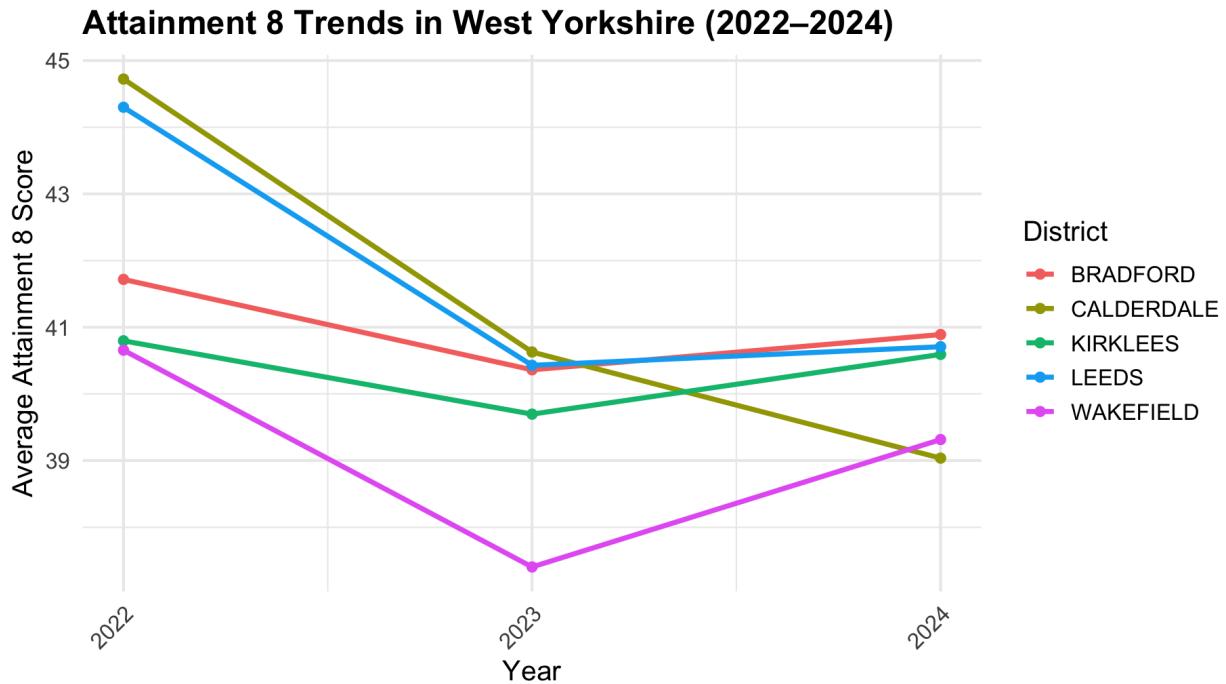
# South Yorkshire Plot
p_south_school <- ggplot(south_school, aes(x = year, y = AvgAttainment8, color = District, group = District)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 2) +
  scale_x_continuous(breaks = c(2022, 2023, 2024)) +
  labs(
    title = "Attainment 8 Trends in South Yorkshire (2022-2024)",
    x = "Year",
    y = "Average Attainment 8 Score",
    color = "District"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )

# West Yorkshire Plot
p_west_school <- ggplot(west_school, aes(x = year, y = AvgAttainment8, color = District, group = District)) +
  geom_line(linewidth = 1.2) +
```



Interpretation:

The chart shows Attainment 8 trends in South Yorkshire (2022-2024) for Barnsley, Doncaster, Rotherham, and Sheffield. All districts exhibit a downward trend in average scores. Sheffield starts highest at 45 in 2022, dropping to 43 by 2024. Doncaster declined from 42 to 40, Rotherham from 39 to 37, and Barnsley from 34 to 33. The overall trend indicates a gradual decrease in educational attainment across the region over the three years.



Interpretation:

The chart displays Attainment 8 trends in West Yorkshire (2022-2024) for Bradford, Calderdale, Kirklees, Leeds, and Wakefield. Leeds starts highest at 45, dropping to 41 by 2024. Bradford declined from 42 to 41, Calderdale from 45 to 39, Kirklees from 41 to 42, and Wakefield from 41 to 40. Trends vary: Leeds and Calderdale show significant drops, Bradford stabilizes, Kirklees slightly rises, and Wakefield dips then recovers. Overall, attainment scores fluctuate with a general downward trend across the region over the three years.

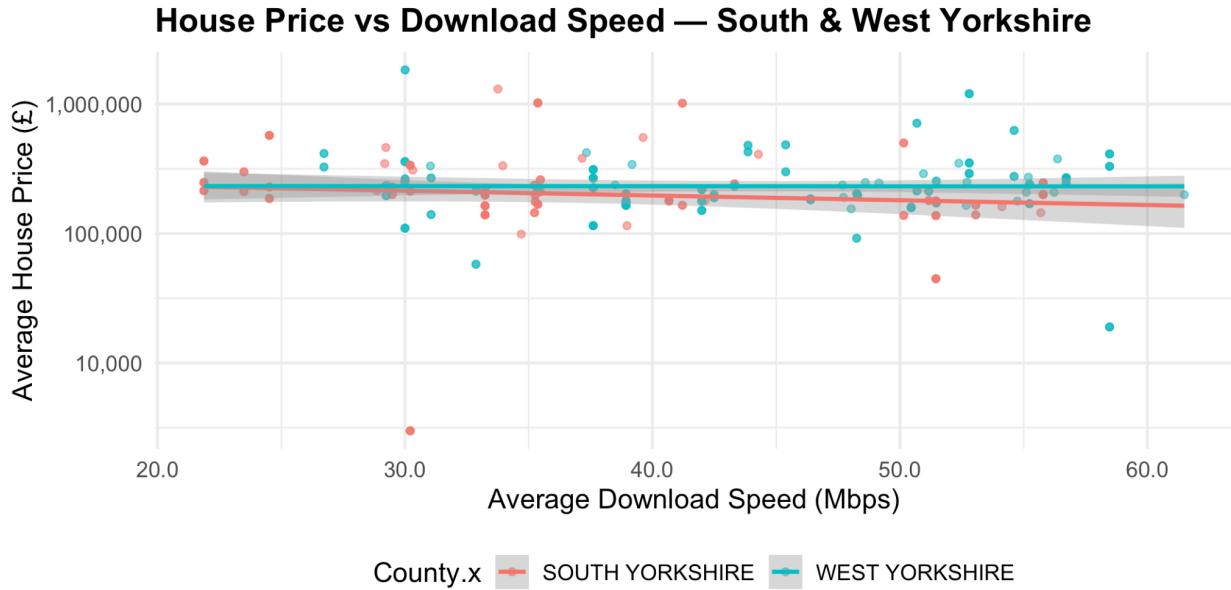
Linear Modelling

1. House Price vs Download Speed

```

1 library(tidyverse)
2
3 house<-read_csv("Cleaned Data/cleanHousePrices.csv")
4 broadband<-read_csv("Cleaned Data/cleanBroadbandPerformance.csv")
5
6 # Step 1: Average house price per shortPostcode
7 avg_price <- house %>%
8   group_by(shortPostcode, District, County) %>%
9   summarise(AvgPrice = mean(Price, na.rm = TRUE), .groups = "drop")
0
1 avg_broadband<- broadband %>%
2   group_by(shortPostcode, District, County) %>%
3   summarise(AvgDownload = mean(AvgDownload, na.rm = TRUE), .groups = "drop")
4
5
6 # Step 2: Merge house price and broadband data
7 merged_df <- avg_price %>%
8   inner_join(avg_broadband, by = "shortPostcode") %>%
9   filter(!is.na(AvgDownload), !is.na(AvgPrice))
0
1
2 # Step 3: Correlation
3 cor_test <- cor.test(merged_df$AvgPrice, merged_df$AvgDownload)
4 print(cor_test)
5
6 # Step 4: Linear model
7 model <- lm(AvgPrice ~ AvgDownload, data = merged_df)
8 summary_model <- summary(model)
9 print(summary_model)
0
1
2 # Step 5: Scatter plot with regression line
3 ggplot(merged_df, aes(x = AvgDownload, y = AvgPrice, color = County.x)) +
4   geom_point(alpha = 0.6) +
5   geom_smooth(method = "lm", se = TRUE, fullrange=TRUE) +
6   scale_x_continuous(name = "Average Download Speed (Mbps)", labels = label_number(accuracy = 0.1)) +
7   scale_y_log10(name = "Average House Price (£)", labels = label_number(big.mark = ",")) +
8   labs(
9     title = "House Price vs Download Speed – South & West Yorkshire",
0   ) +
1   theme_minimal(base_size = 14) +

```



Linear Model Summary

- Call:** lm(formula = AvgPrice ~ AvgDownload, data = merged_df)
- Residuals:** Min = -280378, 1Q = -96562, Median = -58992, 3Q = 11896, Max = 1548726
- Coefficients:**
 - (Intercept) = 312184.5, Std. Error = 54055.7, t = 5.775, Pr(>|t|) = 1.97e-08 ***
 - AvgDownload = -953.6, Std. Error = 1265.1, t = -0.754, Pr(>|t|) = 0.452
- Residual Std Error:** 227700 on 291 DF
- Multiple R-squared:** 0.001948, Adjusted R-squared: -0.001481
- F-statistic:** 0.5681 on 1 and 291 DF, p-value: 0.4516

Correlation

- Pearson's product-moment correlation:**
 - t = -0.75373, df = 291, p-value = 0.4516
 - 95% CI: -0.15793002 to 0.07080436
 - Sample correlation: -0.0441413

Analysis:

The scatter plot provides a graphical depiction of download speed and average house price in South and West Yorkshire, with each county having a linear model. The trend in the graph is not discernible, as confirmed by weak negative correlation ($r = -0.044$, $p = 0.4516$), i.e., no evident linear relationship. Linear model summary indicates an intercept of 312,184.5, a non-significant slope of -953.6 ($p = 0.452$), and a very low R-squared (0.001948) implying that the model captures barely any variation. The p-value (0.4516) confirms the lack of prediction capability, suggesting that download speed has no significant influence on house prices.

2.Attainment 8 score vs House Price

```
# Load libraries
library(tidyverse)

south_yorkshire_districts <- c("SHEFFIELD", "BARNESLEY", "DONCASTER", "ROTHERHAM")
west_yorkshire_districts <- c("LEEDS", "BRADFORD", "CALDERDALE", "KIRKLEES", "WAKEFIELD")

school<-read_csv("Cleaned Data/cleanSchool.csv")

school <- school %>%
  mutate(
    District = toupper(District),      # Standardize case for merging
    Year = year                         # Rename `year` to `Year` for consistency
  )

house_filtered <- house %>%
  filter(Year %in% 2022:2024) %>%
  select(Year, Price, shortPostcode, District, County)

school_filtered <- school %>%
  filter(Year %in% 2022:2024) %>%
  select(Year, attainment_8_score, shortPostcode, District, County)

combined <- inner_join(house_filtered, school_filtered,
                        by = c("shortPostcode", "District", "County", "Year"))

combined <- combined %>%
  mutate(CountyGroup = case_when(
    District %in% south_yorkshire_districts ~ "South Yorkshire",
    District %in% west_yorkshire_districts ~ "West Yorkshire",
    TRUE ~ NA_character_
  )) %>%
  filter(!is.na(CountyGroup)) # Keep only selected counties

district_avg <- combined %>%
  group_by(CountyGroup, District, Year) %>%
  summarise(
```

```

combined <- combined %>%
  mutate(CountyGroup = case_when(
    District %in% south_yorkshire_districts ~ "South Yorkshire",
    District %in% west_yorkshire_districts ~ "West Yorkshire",
    TRUE ~ NA_character_
  )) %>%
  filter(!is.na(CountyGroup)) # Keep only selected counties

district_avg <- combined %>%
  group_by(CountyGroup, District, Year) %>%
  summarise(
    avg_attainment8 = mean(attainment_8_score, na.rm = TRUE),
    avg_price = mean(Price, na.rm = TRUE),
    .groups = "drop"
  )

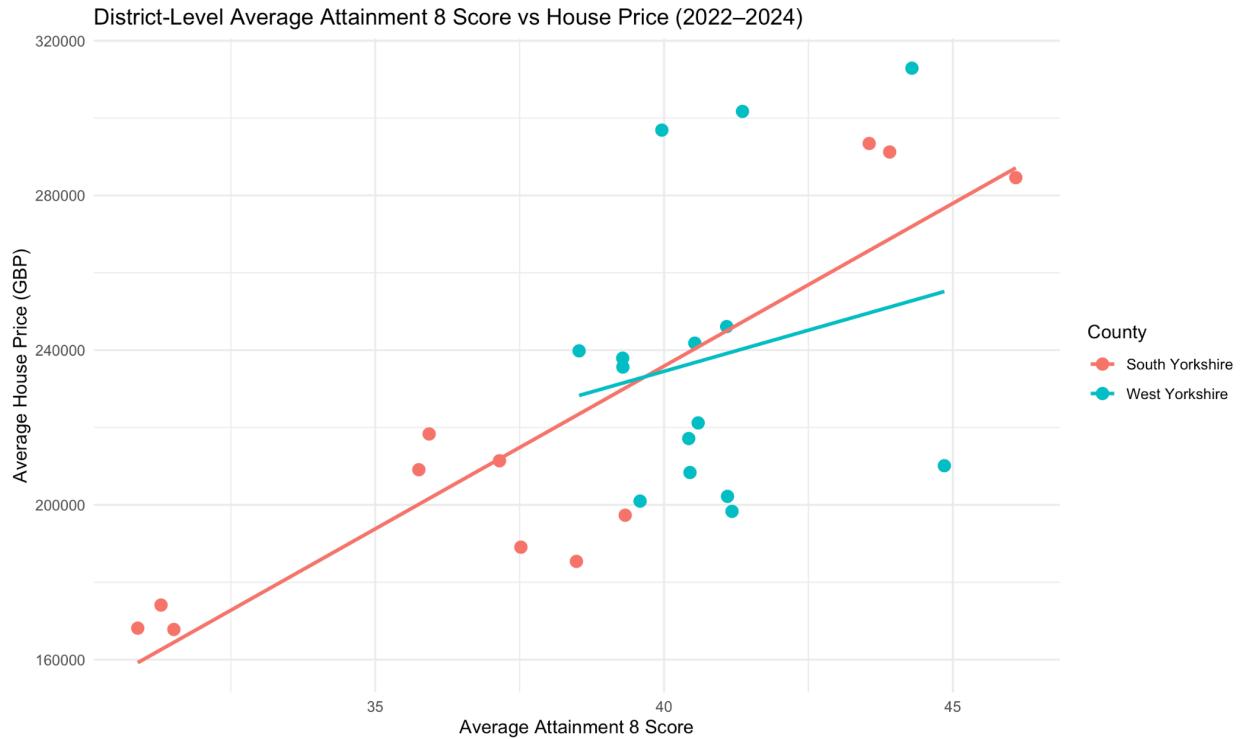
correlation <- district_avg %>%
  group_by(CountyGroup) %>%
  summarise(correlation = cor(avg_attainment8, avg_price, use = "complete.obs"))

print(correlation)
cor_test <- cor.test(district_avg$avg_attainment8, district_avg$avg_price)
print(cor_test)

model <- lm(avg_price ~ avg_attainment8, data = district_avg)
summary_model <- summary(model)
print(summary_model)

ggplot(district_avg, aes(x = avg_attainment8, y = avg_price, color = CountyGroup)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "District-Level Average Attainment 8 Score vs House Price (2022-2024)",
    x = "Average Attainment 8 Score",
    y = "Average House Price (GBP)",
    color = "County"
  ) +
  theme_minimal()

```



Linear Model Summary

- Call:** lm(formula = avg_price ~ avg_attainment8, data = district_avg)
- Residuals:** Min = -60056, 1Q = -26239, Median = 4941, 3Q = 13782, Max = 64408
- Coefficients:**
 - (Intercept) = -75499, Std. Error = 63141, t = -1.196, Pr(>|t|) = 0.243
 - avg_attainment8 = 7707, Std. Error = 1595, t = 4.832, Pr(>|t|) = 5.76e-05 ***
- Residual Std Error:** 31490 on 25 DF
- Multiple R-squared:** 0.4829, Adjusted R-squared: 0.4622
- F-statistic:** 23.35 on 1 and 25 DF, p-value: 5.761e-05

Correlation

- Pearson's product-moment correlation:**
 - t = 4.8319, df = 25, p-value = 5.761e-05
 - 95% CI: 0.4278969 to 0.8503655
 - Sample correlation: 0.6949132

Analysis:

The scatter plot is of the relationship between district-average Attainment 8 scores and house prices in South and West Yorkshire with a linear fit plotted for each. There is a good positive trend with evidence of a Pearson correlation coefficient of 0.695 ($p = 5.761\text{e-}05$), which indicates a significant, strong positive relationship. The linear model (`lm(avg_price ~ avg_attainment8)`) has an intercept of -75,499, a slope of 7,707 ($p = 5.76\text{e-}05$), and an R-squared of 0.4829, explaining almost half the variance. The small p-value confirms that Attainment 8 scores are an extremely significant predictor of house prices.

3.Attainment 8 score vs Drug (2023)

```

library(tidyverse)
library(scales)

# Load datasets
crime <- read_csv("Cleaned Data/cleanCrime.csv")
towns <- read_csv("Cleaned Data/Towns.csv")
school<-read_csv("Cleaned Data/cleanSchool.csv")

# Standardize short postcodes
crime <- crime %>%
  mutate(shortPostcode = str_trim(toupper(shortPostcode))) %>%
  select(-District, -County) # Remove to avoid .x/.y issues

towns <- towns %>%
  mutate(shortPostcode = str_trim(toupper(shortPostcode)))

# Step 1: Deduplicate towns to avoid postcode ambiguity
towns_clean <- towns %>%
  group_by(shortPostcode, District, County) %>%
  summarise(n = n(), Population2023 = mean(Population2023, na.rm = TRUE), .groups = "drop") %>%
  group_by(shortPostcode) %>%
  slice_max(n, n = 1) %>% # Take most common mapping
  ungroup()

# Step 2: Join towns to crime (after filtering for Drugs 2023)
crime_filtered <- crime %>%
  filter(CrimeType == "Drugs", Year == 2023) %>%
  left_join(
    towns_clean %>% select(shortPostcode, District, County, Population2023),
    by = "shortPostcode"
  )

# Step 3: Drop rows with missing District/Population, compute drug rate
crime_rates <- crime_filtered %>%
  drop_na(Population2023, District) %>%
  group_by(District, County, Year) %>%
  summarise(
    DrugOffenseCount = n(),
    Population = first(Population2023),
    DrugOffenseRate = (DrugOffenseCount / Population) * 10000,
  )

```

```

south_yorkshire <- c("SHEFFIELD", "BARNESLEY", "ROTHERHAM", "DONCASTER")
west_yorkshire <- c("LEEDS", "BRADFORD", "WAKEFIELD", "KIRKLEES", "CALDERDALE")
yorkshire_districts <- c(south_yorkshire, west_yorkshire)

school_clean <- school %>%
  mutate(
    District = toupper(District),
    Year = year
  ) %>%
  filter(Year == 2023) %>%
  group_by(District, County) %>%
  summarise(
    avg_attainment8 = mean(attendance_8_score, na.rm = TRUE),
    .groups = "drop"
  )

# Filter and combine datasets for only the defined districts
edu_crime_combined <- inner_join(
  school_clean %>% filter(District %in% yorkshire_districts),
  crime_rates %>% filter(District %in% yorkshire_districts),
  by = c("District", "County")
)

# Add county group label
edu_crime_combined <- edu_crime_combined %>%
  mutate(CountyGroup = case_when(
    District %in% south_yorkshire ~ "South Yorkshire",
    District %in% west_yorkshire ~ "West Yorkshire"
  ))

correlation <- edu_crime_combined %>%
  group_by(CountyGroup) %>%
  summarise(
    correlation = cor(avg_attainment8, DrugOffenseRate, use = "complete.obs")
  )
print("📊 Correlation by CountyGroup:")
print(correlation)

```

44 12 weeks Model

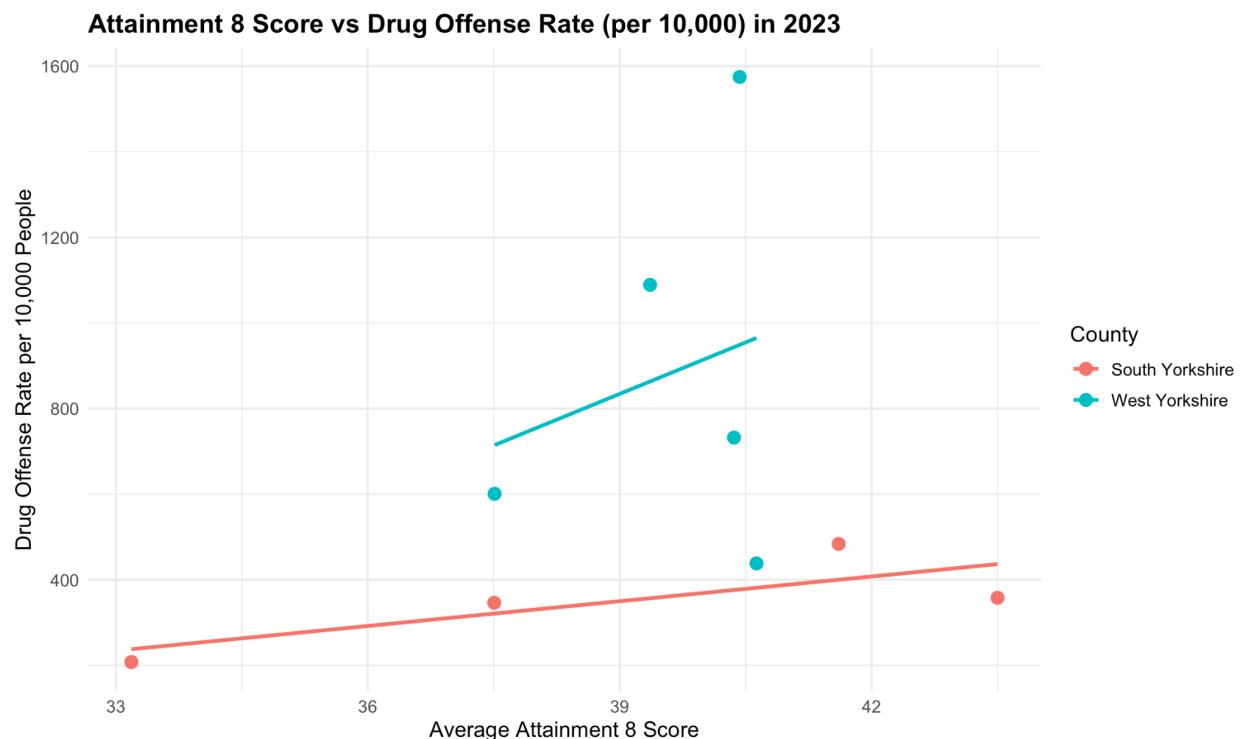
```

correlation <- edu_crime_combined %>%
  group_by(CountyGroup) %>%
  summarise(
    correlation = cor(avg_attainment8, DrugOffenseRate, use = "complete.obs")
  )
print("📊 Correlation by CountyGroup:")
print(correlation)

# --- Linear Model ---
model <- lm(DrugOffenseRate ~ avg_attainment8 * CountyGroup, data = edu_crime_combined)
print("📈 Linear Model Summary:")
summary(model)

ggplot(edu_crime_combined, aes(x = avg_attainment8, y = DrugOffenseRate, color = CountyGroup)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Attainment 8 Score vs Drug Offense Rate (per 10,000) in 2023",
    x = "Average Attainment 8 Score",
    y = "Drug Offense Rate per 10,000 People",
    color = "County"
  ) +
  theme_minimal() +
  theme(
    text = element_text(size = 12),
    plot.title = element_text(face = "bold")
  )

```



Linear Model Summary

- **Call:** lm(formula = DrugOffenseRate ~ avg_attainment8 * CountyGroup, data = edu_crime_combined)
- **Residuals:** 1 = -29.98, 2 = -211.15, 3 = -526.65, 4 = 83.50, 5 = 225.67, 6 = 625.82, 7 = 25.23, 8 = -78.75, 9 = -113.69
- **Coefficients:**
 - (Intercept) = -401.84, Std. Error = 1961.93, t = -0.205, Pr(>|t|) = 0.846
 - avg_attainment8 = 19.28, Std. Error = 50.11, t = 0.385, Pr(>|t|) = 0.716
 - CountyGroupWest Yorkshire = -1893.44, Std. Error = 6394.73, t = -0.296, Pr(>|t|) = 0.779
 - avg_attainment8:CountyGroupWest Yorkshire = 60.97, Std. Error = 161.38, t = 0.378, Pr(>|t|) = 0.721
- **Residual Std Error:** 398 on 5 DF
- **Multiple R-squared:** 0.4727, Adjusted R-squared: 0.1564
- **F-statistic:** 1.494 on 3 and 5 DF, p-value: 0.3234

Correlation

- **South Yorkshire:** 0.784
- **West Yorkshire:** 0.230

Analysis:

The following scatter plot illustrates the relationship between mean Attainment 8 scores and drug offense rates per 10,000 citizens in 2023 for West and South Yorkshire, with separate linear fits. There is a rising trend, with high correlation in South Yorkshire ($r = 0.784$) and low correlation in West Yorkshire ($r = 0.230$). The linear model ($\text{lm}(\text{DrugOffenseRate} \sim \text{avg_attainment8} * \text{CountyGroup})$) yields an intercept of -401.84, a slope of 19.28 (which is non-significant, $p = 0.716$), and an R-squared of 0.4727, indicating moderate fit but low significance ($p = 0.3234$). The interaction terms with CountyGroup are non-significant.

4. Download speed vs Drug

```

library(tidyverse)
library(scales)

# Load datasets
crime <- read_csv("Cleaned Data/cleanCrime.csv")
towns <- read_csv("Cleaned Data/Towns.csv")

# Standardize short postcodes
crime <- crime %>%
  mutate(shortPostcode = str_trim(toupper(shortPostcode))) %>%
  select(-District, -County)

towns <- towns %>%
  mutate(shortPostcode = str_trim(toupper(shortPostcode)))

# Step 1: Deduplicate towns to avoid postcode ambiguity
towns_clean <- towns %>%
  group_by(shortPostcode, District, County) %>%
  summarise(n = n(), Population2023 = mean(Population2023, na.rm = TRUE), .groups = "drop") %>%
  group_by(shortPostcode) %>%
  slice_max(n, n = 1) %>%
  ungroup()

crime_filtered <- crime %>%
  filter(CrimeType == "Drugs", Year == 2023) %>%
  left_join(
    towns_clean %>% select(shortPostcode, District, County, Population2023),
    by = "shortPostcode"
  )

crime_rates <- crime_filtered %>%
  drop_na(Population2023, District) %>%
  group_by(District, County, Year) %>%
  summarise(
    DrugOffenseCount = n(),
    Population = first(Population2023),
    DrugOffenseRate = (DrugOffenseCount / Population) * 10000,
    .groups = "drop"
  )

```

```

# Load cleaned broadband data (assumed to be mostly 2023)
broadband <- read_csv("Cleaned Data/cleanBroadbandPerformance.csv")

# Define Yorkshire districts
south_yorkshire <- c("SHEFFIELD", "BARNESLEY", "ROTHERHAM", "DONCASTER")
west_yorkshire <- c("LEEDS", "BRADFORD", "WAKEFIELD", "KIRKLEES", "CALDERDALE")
yorkshire_districts <- c(south_yorkshire, west_yorkshire)

# Filter and prepare broadband data (already at district level)
broadband_clean <- broadband %>%
  filter(District %in% yorkshire_districts) %>%
  group_by(District, County) %>%
  summarise(avg_download = mean(AvgDownload, na.rm = TRUE), .groups = "drop")

# Filter and prepare crime rates (from previous script)
crime_rates_2023 <- crime_rates %>%
  filter(Year == 2023, District %in% yorkshire_districts)

# Join broadband and crime
bb_crime_joined <- inner_join(broadband_clean, crime_rates_2023, by = c("District", "County"))

# Label CountyGroup
bb_crime_joined <- bb_crime_joined %>%
  mutate(CountyGroup = case_when(
    District %in% south_yorkshire ~ "South Yorkshire",
    District %in% west_yorkshire ~ "West Yorkshire"
  ))

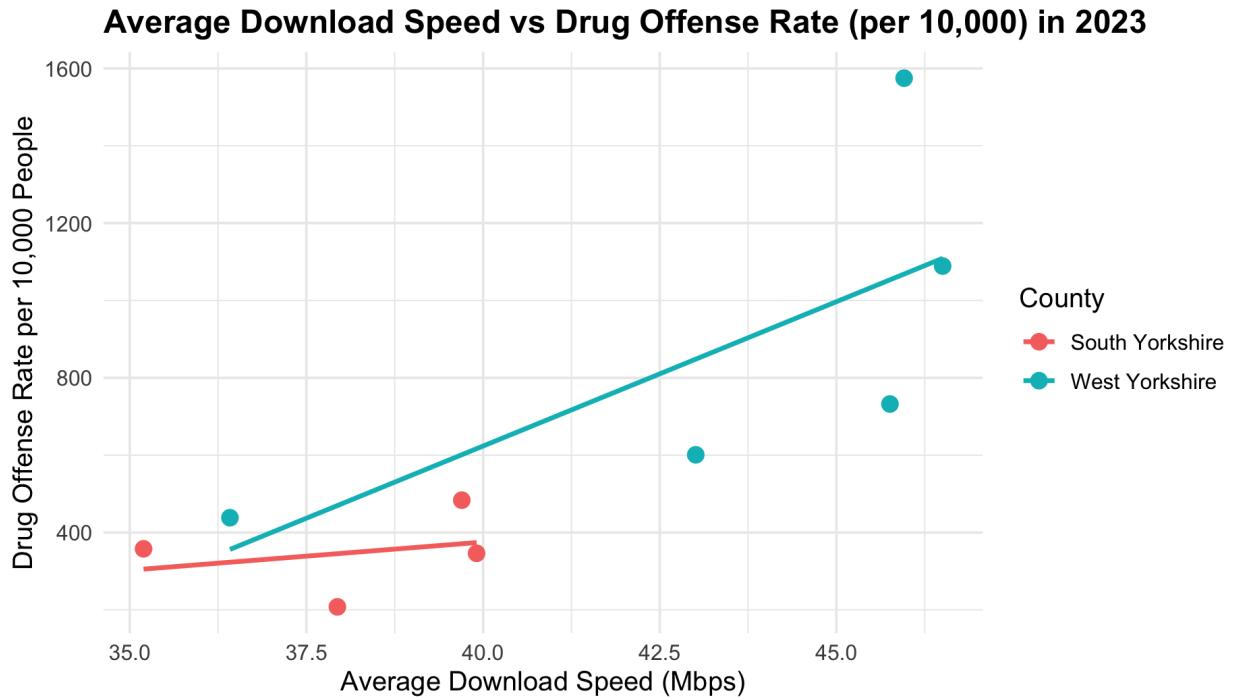
# --- Correlation ---
correlation <- bb_crime_joined %>%
  group_by(CountyGroup) %>%
  summarise(
    correlation = cor(avg_download, DrugOffenseRate, use = "complete.obs")
  )
print("📊 Correlation by CountyGroup:")
print(correlation)

# --- Linear Model ---
model <- lm(DrugOffenseRate ~ avg_download * CountyGroup, data = bb_crime_joined)

# --- Linear Model ---
model <- lm(DrugOffenseRate ~ avg_download * CountyGroup, data = bb_crime_joined)
print("🔗 Linear Model Summary:")
summary(model)

# --- Plot ---
ggplot(bb_crime_joined, aes(x = avg_download, y = DrugOffenseRate, color = CountyGroup)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Average Download Speed vs Drug Offense Rate (per 10,000) in 2023",
    x = "Average Download Speed (Mbps)",
    y = "Drug Offense Rate per 10,000 People",
    color = "County"
  ) +
  theme_minimal() +
  theme(
    text = element_text(size = 12),
    plot.title = element_text(face = "bold")
  )

```



Linear Model Summary

- **Call:** lm(formula = DrugOffenseRate ~ avg_download * CountyGroup, data = bb_crime_joined)
- **Residuals:** 1 = -137.52, 2 = -321.04, 3 = 82.03, 4 = 112.72, 5 = -20.09, 6 = 506.37, 7 = -27.67, 8 = 52.47, 9 = -247.27
- **Coefficients:**
 - (Intercept) = -206.58, Std. Error = 3080.37, t = -0.067, Pr(>|t|) = 0.949
 - avg_download = 14.55, Std. Error = 80.57, t = 0.181, Pr(>|t|) = 0.864
 - CountyGroupWest Yorkshire = -2155.30, Std. Error = 3463.16, t = -0.622, Pr(>|t|) = 0.561
 - avg_download:CountyGroupWest Yorkshire = 60.09, Std. Error = 88.34, t = 0.680, Pr(>|t|) = 0.527
- **Residual Std Error:** 304.3 on 5 DF
- **Multiple R-squared:** 0.6919, Adjusted R-squared: 0.507
- **F-statistic:** 3.743 on 3 and 5 DF, p-value: 0.09471

Correlation

- **South Yorkshire:** 0.281
- **West Yorkshire:** 0.692

Analysis:

The scatter plot illustrates the relationship between average download speed (Mbps) and drug offense rates per 10,000 people in 2023 for West and South Yorkshire with individual linear fits. There is an upward trend, with West Yorkshire showing greater correlation ($r = 0.692$) and South Yorkshire a lesser one ($r = 0.281$). The linear model $\text{lm}(\text{DrugOffenseRate} \sim \text{avg_download} * \text{CountyGroup})$ has an intercept of -206.58, a non-significant slope of 14.55 ($p = 0.864$), and an R-squared of 0.6919, indicating a moderate fit ($p = 0.09471$). Interaction terms with CountyGroup are also non-significant.

5. Average Download speed vs Attainment 8 score

```

1 library(tidyverse)
2 library(scales)
3
4
5 broadband <- read_csv("Cleaned Data/cleanBroadbandPerformance.csv") %>%
6   mutate(District = toupper(District), County = toupper(County)) %>%
7   group_by(District, County) %>%
8   summarise(
9     avg_download = mean(AvgDownload, na.rm = TRUE),
10    .groups = "drop"
11  )
12
13 # --- 2. Load and prepare school data for Attainment 8 (2023) ---
14 school <- read_csv("Cleaned Data/cleanSchool.csv") %>%
15   mutate(
16     District = toupper(District),
17     County = toupper(County),
18     Year = year
19   ) %>%
20   filter(Year == 2023) %>%
21   group_by(District, County) %>%
22   summarise(
23     avg_attainment8 = mean(attainment_8_score, na.rm = TRUE),
24     .groups = "drop"
25   )
26
27 # --- 3. Define target districts ---
28 south_yorkshire <- c("SHEFFIELD", "BARNESLEY", "ROTHERHAM", "DONCASTER")
29 west_yorkshire <- c("LEEDS", "BRADFORD", "WAKEFIELD", "KIRKLEES", "CALDERDALE")
30 yorkshire_districts <- c(south_yorkshire, west_yorkshire)
31
32 # --- 4. Filter to only those districts ---
33 broadband_clean <- broadband %>% filter(District %in% yorkshire_districts)
34 school_clean <- school %>% filter(District %in% yorkshire_districts)
35

```

```

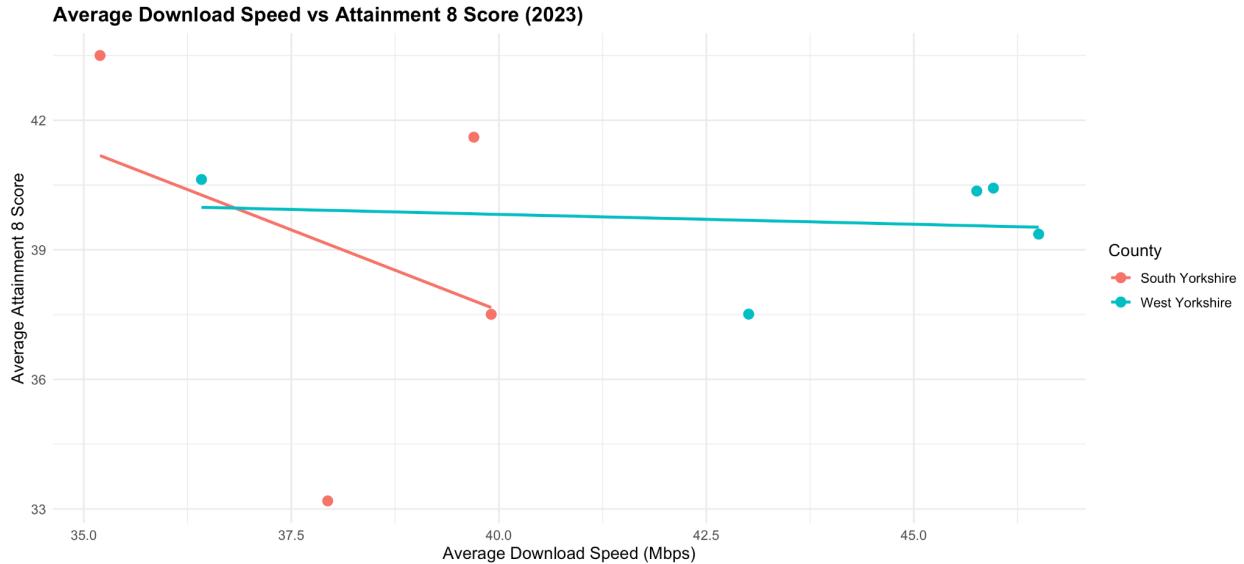
bb_school <- inner_join(
  broadband_clean,
  school_clean,
  by = c("District", "County")
) %>%
  mutate(CountyGroup = case_when(
    District %in% south_yorkshire ~ "South Yorkshire",
    District %in% west_yorkshire ~ "West Yorkshire"
  ))
)

# --- 6. Compute correlations per county ---
correlation <- bb_school %>%
  group_by(CountyGroup) %>%
  summarise(
    correlation = cor(avg_download, avg_attainment8, use = "complete.obs")
  )
print("📊 Pearson correlation by County:")
print(correlation)

# --- 7. Fit linear model with interaction ---
model <- lm(avg_attainment8 ~ avg_download * CountyGroup, data = bb_school)
print("✍️ Linear model summary:")
summary(model)

# --- 8. Plotting ---
ggplot(bb_school, aes(x = avg_download, y = avg_attainment8, color = CountyGroup)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Average Download Speed vs Attainment.8 Score (2023)",
    x = "Average Download Speed (Mbps)",
    y = "Average Attainment.8 Score",
    color = "County"
  ) +
  theme_minimal() +
  theme(
    text = element_text(size = 12),
    plot.title = element_text(face = "bold")
)

```



Linear Model Summary

- Call:** lm(formula = avg_attainment8 ~ avg_download * CountyGroup, data = bb_school)
- Residuals:** 1 = -5.9491, 2 = 0.8050, 3 = 0.6452, 4 = 3.7882, 5 = -0.1600, 6 = 0.8817, 7 = -0.1573, 8 = 2.3181, 9 = -2.1719
- Coefficients:**
 - (Intercept) = 67.4609, Std. Error = 35.5712, t = 1.897, Pr(>|t|) = 0.116
 - avg_download = -0.7467, Std. Error = 0.9304, t = -0.803, Pr(>|t|) = 0.459
 - CountyGroupWest Yorkshire = -25.8104, Std. Error = 39.9914, t = -0.645, Pr(>|t|) = 0.547
 - avg_download:CountyGroupWest Yorkshire = 0.7009, Std. Error = 1.0201, t = 0.687, Pr(>|t|) = 0.523
- Residual Std Error:** 3.514 on 5 DF
- Multiple R-squared:** 0.1299, Adjusted R-squared: -0.3922
- F-statistic:** 0.2488 on 3 and 5 DF, p-value: 0.8592

Correlation

- South Yorkshire:** -0.355
- West Yorkshire:** -0.148

Analysis:

The scatter plot shows the relationship between mean download speed (Mbps) and Attainment 8 scores in 2023 for South and West Yorkshire, with separate linear fits. There is a weak negative trend, where South Yorkshire has a correlation of -0.355 and West Yorkshire -0.148, indicating that they are low relationships. The linear model (`lm(avg_attainment8 ~ avg_download * CountyGroup)`) provides an intercept of 67.4609, non-significant slope of -0.7467 ($p = 0.459$), and R-squared of 0.1299, suggesting poor fit ($p = 0.8592$). Interaction terms with CountyGroup are also non-significant.

Recommendation System

```
# Loading required libraries
library(tidyverse)
library(ggplot2)

# Loading cleaned datasets (adjust file paths as needed)
house<-read_csv("Cleaned Data/cleanHousePrices.csv")
crime<-read_csv("Cleaned Data/cleanCrime.csv")
schools<-read_csv("Cleaned Data/cleanSchool.csv")
broadband<-read_csv("Cleaned Data/cleanBroadbandPerformance.csv")
towns<-read_csv("Cleaned Data/Towns.csv") # Columns: Town, County, Year, Attainment8Score

# Step 1: Aggregate crime data (2022-2024)
crime_agg <- crime %>%
  filter(Year %in% 2022:2024) %>%
  group_by(Town, County, shortPostcode) %>%
  summarise(CrimeCount = n(), .groups = "drop") %>%
  left_join(select(towns, shortPostcode, Town, County, Population2024), by = c("shortPostcode", "Town", "County")) %>%
  filter(!is.na(Population2024)) %>%
  mutate(CrimeRate = (CrimeCount / Population2024) * 1000) %>%
  # Remove duplicates, keeping first occurrence
  distinct(Town, County, .keep_all = TRUE) %>%
  select(Town, County, CrimeRate)

# Check for duplicates
print("Duplicates in crime_agg:")
print(sum(duplicated(crime_agg[, c("Town", "County")])))"

# Step 2: Aggregate house prices (2021-2024)
house_prices_agg <- house %>%
  filter(Year %in% 2021:2024) %>%
  group_by(Town, County) %>%
  summarise(avg_price = mean(Price, na.rm = TRUE), .groups = "drop") %>%
  # Remove duplicates
  distinct(Town, County, .keep_all = TRUE) %>%
  select(Town, County, avg_price)

# Check for duplicates
print("Duplicates in house_prices_agg:")
print(sum(duplicated(house_prices_agg[, c("Town", "County")])))
```

```

# Step 3: Aggregate school scores (2022–2024)
schools_agg <- school %>%
  mutate(
    Town = toupper(Town),
    County = toupper(County),
    Year = year
  ) %>%
  filter(Year %in% 2022:2024) %>%
  group_by(Town, County) %>%
  summarise(AvgAttainment8 = mean(attainment_8_score, na.rm = TRUE), .groups = "drop") %>%
  # Remove duplicates
  distinct(Town, County, .keep_all = TRUE) %>%
  select(Town, County, AvgAttainment8)

# Check for duplicates
print("Duplicates in schools_agg:")
print(sum(duplicated(schools_agg[, c("Town", "County")])))>

# Step 4: Aggregate broadband data (assumed 2023)
broadband_agg <- broadband %>%
  group_by(Town, County) %>%
  summarise(AvgDownload = mean(AvgDownload, na.rm = TRUE), .groups = "drop") %>%
  # Remove duplicates
  distinct(Town, County, .keep_all = TRUE) %>%
  select(Town, County, AvgDownload)

# Check for duplicates
print("Duplicates in broadband_agg:")
print(sum(duplicated(broadband_agg[, c("Town", "County")])))>

# Step 5: Merge all datasets
merged_data <- crime_agg %>%
  left_join(house_prices_agg, by = c("Town", "County")) %>%
  left_join(broadband_agg, by = c("Town", "County"), relationship = "one-to-one") %>%
  left_join(schools_agg, by = c("Town", "County")) %>%
  filter(!is.na(avg_price) & !is.na(AvgDownload) & !is.na(AvgAttainment8)) # Remove incomplete rows

```

```

# Check for duplicates in merged data
print("Duplicates in merged_data:")
print(sum(duplicated(merged_data[, c("Town", "County")])))

View(merged_data)

# Step 6: Normalize metrics to 0-10 scale, handling edge cases
merged_data <- merged_data %>%
  mutate(
    # Affordability (lower price is better)
    AffordabilityScore = case_when(
      max(avg_price, na.rm = TRUE) == min(avg_price, na.rm = TRUE) ~ 5,
      TRUE ~ 10 * (max(avg_price, na.rm = TRUE) - avg_price) /
        (max(avg_price, na.rm = TRUE) - min(avg_price, na.rm = TRUE))
    ),
    # Connectivity (higher speed is better)
    ConnectivityScore = case_when(
      max(AvgDownload, na.rm = TRUE) == min(AvgDownload, na.rm = TRUE) ~ 5,
      TRUE ~ 10 * (AvgDownload - min(AvgDownload, na.rm = TRUE)) /
        (max(AvgDownload, na.rm = TRUE) - min(AvgDownload, na.rm = TRUE))
    ),
    # Safety (lower crime rate is better)
    SafetyScore = case_when(
      max(CrimeRate, na.rm = TRUE) == min(CrimeRate, na.rm = TRUE) ~ 5,
      TRUE ~ 10 * (max(CrimeRate, na.rm = TRUE) - CrimeRate) /
        (max(CrimeRate, na.rm = TRUE) - min(CrimeRate, na.rm = TRUE))
    ),
    # Quality of Life (higher school score is better)
    QualityScore = case_when(
      max(AvgAttainment8, na.rm = TRUE) == min(AvgAttainment8, na.rm = TRUE) ~ 5,
      TRUE ~ 10 * (AvgAttainment8 - min(AvgAttainment8, na.rm = TRUE)) /
        (max(AvgAttainment8, na.rm = TRUE) - min(AvgAttainment8, na.rm = TRUE))
    )
  )

# Diagnostic: Check for zero or NA scores
print("Rows with zero or NA scores:")
print(merged_data %>%
  filter(AffordabilityScore == 0 | ConnectivityScore == 0 |
    SafetyScore == 0 | QualityScore == 0 |
```

```

# Diagnostic: Check for zero or NA scores
print("Rows with zero or NA scores:")
print(merged_data %>%
      filter(AffordabilityScore == 0 | ConnectivityScore == 0 |
             SafetyScore == 0 | QualityScore == 0 |
             is.na(AffordabilityScore) | is.na(ConnectivityScore) |
             is.na(SafetyScore) | is.na(QualityScore)) %>%
      select(Town, County, avg_price, AvgDownload, CrimeRate, AvgAttainment8,
             AffordabilityScore, ConnectivityScore, SafetyScore, QualityScore))

# Step 7: Calculate composite score with weights (40% affordability, 20% each for others)
merged_data <- merged_data %>%
  mutate(CompositeScore = 0.4 * AffordabilityScore +
        0.2 * ConnectivityScore +
        0.2 * SafetyScore +
        0.2 * QualityScore)

# Step 8: Select top 3 towns
top_towns <- merged_data %>%
  arrange(desc(CompositeScore)) %>%
  select(Town, County, AffordabilityScore, ConnectivityScore, SafetyScore, QualityScore, CompositeScore) %>%
  head(10)

# Print results
print("Top 3 Towns for Property Investment:")
print(top_towns)
View(top_towns)

write.csv(top_towns, "recommendation system/recommendation_data.csv", row.names = FALSE)

```

Town	County	AffordabilityScore	ConnectivityScore	SafetyScore	QualityScore	CompositeScore
MEXBOROUGH	SOUTH YORKSHIRE	10	8.67346919215254	9.77589336740754	4.75065368742715	8.64000324939745
HECKMONDWIKE	WEST YORKSHIRE	9.73721141173315	9.1026701029599	9.7985251947537	3.57149607787544	8.38942283981107
MIRFIELD	WEST YORKSHIRE	7.37877933040529	10	9.77884680740072	6.09142173077529	8.12556543979732
CASTLEFORD	WEST YORKSHIRE	8.69556907119	7.56222866188537	9.67666951811291	5.39709542261286	8.00542634899823
BINGLEY	WEST YORKSHIRE	7.12350350511688	7.67235631946473	9.7784790797434	7.14887271734451	7.76934302535728
LIVERSEDGE	WEST YORKSHIRE	8.48644518959773	7.55011871971936	9.70985076811794	2.73950162240494	7.39447229788754
DEWSBURY	WEST YORKSHIRE	8.76561655355841	5.88434950262452	9.51656415632784	3.86136876071665	7.35870310535717
ROOTHERHAM	SOUTH YORKSHIRE	8.84179376191704	5.65410634630876	9.42792508616613	3.25137722185316	7.20339923563243
BATLEY	WEST YORKSHIRE	8.52018061427123	4.28072455402386	9.77636285108798	4.19698831238383	7.05888738920763
PUDSEY	WEST YORKSHIRE	7.07645208671341	9.11268626569026	9.80594789199272	1.80575244935891	6.97545815609374

Overview

This recommendation system ranks South and West Yorkshire towns on four major parameters: affordability, broadband speed, crime rate, and school performance. Each of the four parameters was normalized to 0–10, and affordability was given the highest weight of 40%, while the remaining three were equal weights of 20%. By looking at different aspects of livability, the system recommends locations most appropriate to invest in property.

Results

Mexborough in South Yorkshire ranked highest with a great mix of affordability and safety. Heckmondwike and Mirfield in West Yorkshire followed, both with excellent broadband connectivity and low crime. The top 10 towns consist of a mix of both counties, reflecting the quality of West Yorkshire's connectivity and South Yorkshire's affordability. Overall positions were based on a composite score of all four measures.

Reflection

The method offers a balanced, data-based perspective, though limited by having missing or incongruent data in specific towns. Localized local characteristics may not be detected by averaging and population projections. More interactive weights based on user preference or the incorporation of public transport and health access are possible future improvements. However, the approach provides a clear, comparable method of assessing multiple towns holistically.

Broadband Speeds

West Yorkshire towns also generally fared better in broadband speed compared to South Yorkshire. Heckmondwike and Mirfield topped the list with over 9 scores. These high speeds are a major reason for their popularity for remote work or streaming, which helps them achieve high composite scores in the final recommendation.

School Grades

School performance varied widely between towns, with Bingley and Mirfield leading the pack in educational quality. Mexborough and Rotherham, while overall top-ranked, had relatively humble school scores. This would suggest that high affordability or safety can override education in the final score based on priorities.

House Prices

South Yorkshire persistently offers lower priced housing, with Mexborough and Rotherham being the lowest. This is in line with investment aims seeking lower entry costs. West Yorkshire towns had more expensive average prices that decreased their affordability score although all other areas were impressive.

Crimes

Safety scores were derived from crime per 1,000 residents, and rates were averaged over 2022-2024. Most of the top villages had crime rates low enough to score above 9 in safety, most significantly Mexborough, Mirfield, and Bingley. This factor greatly influenced final ranks, especially when crime was low and data was available.

Overall Score

Composite scores balanced all four factors to rank towns fairly. Mexborough achieved the highest overall score of 8.64, excelling in affordability and safety. Heckmondwike and Mirfield closely followed, combining high broadband and low crime. These towns provide an optimal mix for property investment based on a multi-criteria decision framework.

Legal and Ethical Issues

Legal and ethical considerations include GDPR compliance and ensuring all UK government datasets are properly licensed for reuse. Anonymizing any personal data prevents privacy breaches. It's essential to acknowledge data source limitations and avoid biased weighting that might disadvantage certain communities. Transparency in methodology preserves public trust.

Conclusion

In summary, our multi-criteria analysis highlights Mexborough, Heckmondwike, and Mirfield as prime investment towns, balancing affordability, connectivity, safety, and school performance. While data gaps exist, this transparent framework can be refined with additional metrics and user-specific weightings to guide informed, community-sensitive property decisions in Yorkshire.

References

- Rogers, J. and Jonker, A. (nd) *What is data cleaning?*, IBM. Available at:
<https://www.ibm.com/think/topics/data-cleaning>
- Registry, H.L. (2025) *Price paid data*, GOV.UK. Available at:
<https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads>
- (nd) *How to do linear regression in R* | datacamp. Available at:
<https://www.datacamp.com/tutorial/linear-regression-R>
- Biswal, A. (2025) *What is Exploratory Data Analysis: Data Preparation Guide 2024*, Simplilearn.com. Available at:
<https://www.simplilearn.com/tutorials/data-analytics-tutorial/exploratory-data-analysis>

Appendix

Cleaned Data(csv),code(R),graphs(visualization-jpeg),recommendation system(csv),Report(pdf)
https://github.com/Aryanshah010/Coursework_AryanShah_230511

Obtained Data(.zip)

<https://www.dropbox.com/scl/fi/fduz2ccieuvhia14z9kp0/Obtained-Data.zip?rlkey=4v50k6org8wsxk0rfm2jexfsl&st=qxsvisti&dl=0>