Table of Contents

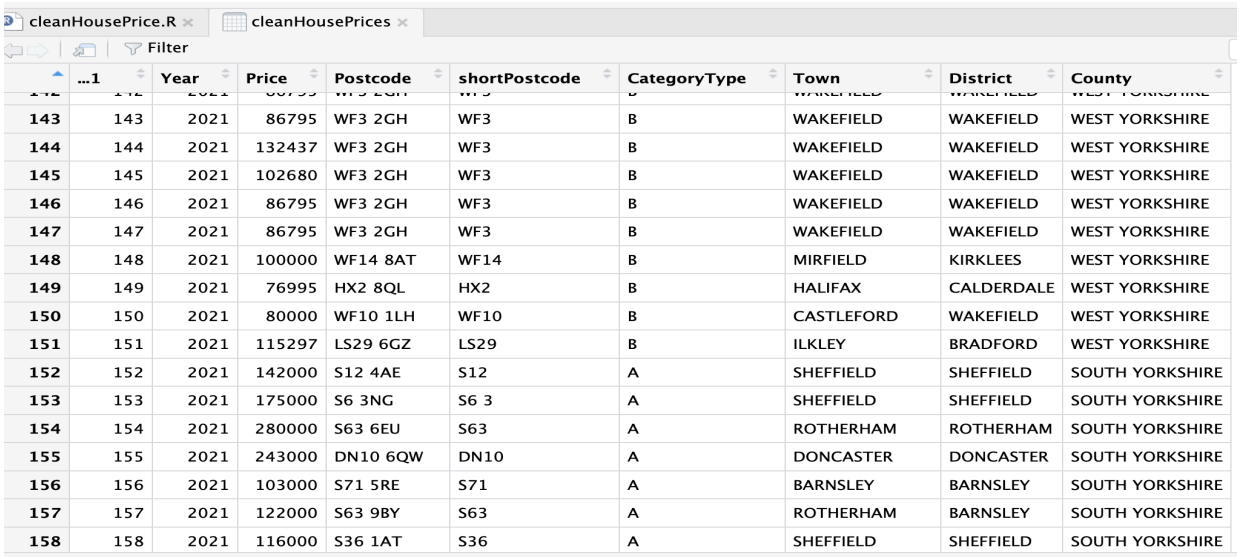
Introduction

This report evaluates towns in South and West Yorkshire for property investment, targeting affordability, connectivity, safety, and quality of life prioritizing house prices, followed by internet speed, crime rates, and school performance. Using publicly available datasets from data.gov.uk, a data science approach was applied: cleaning and merging data across 2021–2024, followed by exploratory and statistical analysis. A weighted recommendation system (40% affordability, 20% each for connectivity, safety, and quality of life) scores and ranks towns, identifying the most suitable locations for investment. The analysis focuses on major Towns/District for relevance and data completeness, ensuring reliable, evidence-based recommendations.

Cleaning Data

House Sales

The House Price datasets (2021–2024) were cleaned and combined to support property investment analysis in South and West Yorkshire. The process involved standardizing column names, merging annual data, and filtering for relevant counties. Dates were formatted consistently, and a Year column was extracted. A shortPostcode was generated for merging with other datasets, and Town and District names were converted to uppercase for consistency. The cleaned dataset, containing key fields (Year, Price, Postcode, shortPostcode, CategoryType, Town, District, County), was saved as cleanHousePrices.csv.


cleanHousePrice.R × cleanHousePrices ×

Filter

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

cleanHousePrice.R

Source on Save | Run

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

col_names=c("TransactionID","Price","Date","Postcode","PropertyType","OldNew","Duration",
            "PAON","SAON","Street","Locality","Town","District","County","CategoryType","RecordStatus")

HousePrices2021 = read_csv("Obtained Data/HousePrices2021.csv",col_names = FALSE)
colnames(HousePrices2021)=col_names

HousePrices2022 = read_csv("Obtained Data/HousePrices2022.csv",col_names = FALSE)
colnames(HousePrices2022)=col_names

HousePrices2023 = read_csv("Obtained Data/HousePrices2023.csv",col_names = FALSE)
colnames(HousePrices2023)=col_names

HousePrices2024 = read_csv("Obtained Data/HousePrices2024.csv",col_names = FALSE)
colnames(HousePrices2024)=col_names

HousePrices =bind_rows(HousePrices2021,HousePrices2022,HousePrices2023,HousePrices2024)

cleanHousePrices = HousePrices %>%
  filter(County=="SOUTH YORKSHIRE"|County=="WEST YORKSHIRE") %>%
  mutate(
    Date=as.Date(Date),
    Year=year(Date),
    shortPostcode=str_trim(substr(Postcode,1,4)),
    District=str_to_upper(District),
    Town=str_to_upper(Town)
  ) %>%
  select(Year,Price,Postcode,shortPostcode,CategoryType,Town,District,County)


write.csv(cleanHousePrices, "Cleaned Data/cleanHousePrices.csv")
```

Towns and Postcodes

The town data cleaning process creates a standardized lookup table mapping towns to districts and counties in South and West Yorkshire, integrating population data for 2021–2024. Starting with the cleaned house price dataset , unique combinations of `shortPostcode`, `Town`, `District`, and `County` are extracted. The 2011 population data is processed by creating `shortPostcode` and aggregating population by `shortPostcode`. Population estimates for 2021–2024 are projected using predefined growth factors, and the 2011 population is dropped. The house price-derived town data is merged with population data via `shortPostcode`, retaining only records with valid 2021 population values. To ensure consistency, `District` and `County` are normalized to uppercase and trimmed. Counties are corrected based on district lists for South Yorkshire and West Yorkshire, overwriting any misassignments. The dataset is filtered to include only South and West Yorkshire, sorted by `County`, `District`, and `Town`, and saved as `Towns.csv`.

Towns | cleanTowns.R

Source on Save

```r
library(tidyverse)

# 1. Read in House Prices to get the Town ↔ District mapping
HousePrices <- read_csv("Cleaned Data/cleanHousePrices.csv")

# 2. Load and aggregate 2011 population by outward postcode
PopulationData <- read_csv("Obtained Data/Population2011.csv") %>%
  mutate(
    shortPostcode = str_trim(str_sub(Postcode, 1, 4))
  ) %>%
  group_by(shortPostcode) %>%
  summarise(
    Pop2011 = sum(Population, na.rm = TRUE),
    .groups = "drop"
  )

# 3. Project 2011 → 2021–2024 using your growth factors
growth_factors <- c(
  "2021" = 1.0777505487,
  "2022" = 1.0820490383,
  "2023" = 1.0869272715,
  "2024" = 1.0915110625
)

PopulationData <- PopulationData %>%
  mutate(
    Population2021 = Pop2011 * growth_factors["2021"],
    Population2022 = Pop2011 * growth_factors["2022"],
    Population2023 = Pop2011 * growth_factors["2023"],
    Population2024 = Pop2011 * growth_factors["2024"]
  ) %>%
  select(-Pop2011)

```

```r
# 4. Build initial Town ↔ District ↔ County table
towns_raw <- HousePrices %>%
  select(shortPostcode, Town, District, County) %>%
  distinct() %>%
  left_join(PopulationData, by = "shortPostcode") %>%
  filter(!is.na(Population2021))

# 5. Explicitly correct Yorkshire districts
south_yorkshire <- c("SHEFFIELD", "BARNSLEY", "DONCASTER", "ROTHERHAM")
west_yorkshire  <- c("LEEDS", "BRADFORD", "CALDERDALE", "KIRKLEES", "WAKEFIELD")

towns_clean <- towns_raw %>%
  # normalize case
  mutate(
    District = str_to_upper(str_trim(District)),
    County   = str_to_upper(str_trim(County))
  ) %>%
  # overwrite any mis-assigned county based on district
  mutate(
    County = case_when(
      District %in% south_yorkshire ~ "SOUTH YORKSHIRE",
      District %in% west_yorkshire  ~ "WEST YORKSHIRE",
      TRUE                          ~ County
    )
  ) %>%
  # keep only the two target counties
  filter(County %in% c("SOUTH YORKSHIRE", "WEST YORKSHIRE")) %>%
  arrange(County, District, Town)

# 6. Save out your cleaned lookup
write_csv(towns_clean, "Cleaned Data/Towns.csv")
```

Towns | cleanTowns.R

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. First, it loads raw broadband data and extracts the outward portion of postcodes (shortPostcode) using regular expressions, ensuring uppercase formatting for consistency. It also converts the average download speed column to numeric format and retains only relevant columns (shortPostcode and AvgDownload). Next, a cleaned town lookup dataset is loaded and deduplicated to ensure each shortPostcode uniquely maps to a Town, District, and County. The cleaned broadband and town data are then joined on the shortPostcode key. This allows aggregation of average download speeds by Town, District, and County. The dataset is filtered to retain only records from South Yorkshire and West Yorkshire. Finally, the cleaned and aggregated broadband performance dataset is saved as cleanBroadbandPerformance.csv.

cleanBroadband.R

Source on Save | Run

```r
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 | cleanBroadbandPerformance

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

The raw crime data sets are processed from May 2022 to December 2024 for South and West Yorkshire.First, it loads a postcode-to-LSOA mapping file and extracts key columns including LSOA code, postcode, and district name. A shortPostcode is derived from the first four characters of each postcode to facilitate later merges.

It then loads a cleaned towns reference file to link short postcodes with their corresponding towns. A sequence of monthly file paths is generated for the required time period, and only those files that exist are retained. Each crime CSV file is read, with any parsing issues logged and problematic rows excluded to ensure clean data.

The crime data is cleaned by selecting relevant columns , filtering out empty or invalid entries, and focusing on three crime types: "Drugs", "Vehicle crime", and "Robbery". The County field is

standardized by removing trailing "Police" and converting to uppercase. A Year column is also extracted.

The cleaned crime records are enriched by joining them with the postcode–LSOA mapping and town datasets, allowing the addition of Postcode, shortPostcode, Town, and District information. Only data from South and West Yorkshire with valid town information is retained. The final cleaned and standardized dataset is saved as cleanCrime.csv.

cleanCrime | cleanCrime.R

Source on Save | Run

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

# Read LSOA-to-Postcode mapping
LSOA_to_Postcode <- read_csv("Obtained Data/Postcode to LSOA 1.csv") %>%
  select(lsoa11cd, pcds, ladnm) %>%
  rename(LSOA_code = lsoa11cd, Postcode = pcds, District = ladnm) %>%
  mutate(
    shortPostcode = str_trim(substring(Postcode, 1, 4)),
    District = str_to_upper(str_trim(District))
  ) %>%
  filter(!is.na(Postcode) & !is.na(shortPostcode) & !is.na(District)) %>%
  # Select one postcode per LSOA to avoid duplication
  group_by(LSOA_code) %>%
  slice(1) %>%
  ungroup()

# Read Towns.csv for Town mapping
Towns <- read_csv("Cleaned Data/Towns.csv") %>%
  select(shortPostcode, Town) %>%
  distinct(shortPostcode, .keep_all = TRUE)

# Generate list of months from May 2022 to December 2024
start_date <- ym("2022-05")
end_date <- ym("2024-12")
months <- seq(start_date, end_date, by = "month")
month_strings <- format(months, "%Y-%m")

# Get list of all crime CSV files
crime_files <- unlist(lapply(month_strings, function(month) {
  c(
    paste0("Obtained Data/crime/", month, "/", month, "-south-yorkshire-street.csv"),
    paste0("Obtained Data/crime/", month, "/", month, "-west-yorkshire-street.csv")
  )
```

cleanCrime | cleanCrime.R

Source on Save | Run

```
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")) %>%
```

```r
  # Filter for relevant crime types
  filter(CrimeType %in% c("Drugs", "Vehicle crime", "Robbery")) %>%
  # Clean and standardize columns
  mutate(
    Month = str_trim(Month),
    LSOA_code = str_trim(LSOA_code),
    LSOA_name = str_trim(LSOA_name),
    CrimeType = str_trim(CrimeType),
    County = str_to_upper(str_remove(str_trim(County), " Police$")),
    Year = as.numeric(substr(Month, 1, 4))
  ) %>%
  # Join with LSOA_to_Postcode to get Postcode, shortPostcode, District
  left_join(LSOA_to_Postcode, by = "LSOA_code", relationship = "many-to-one") %>%
  # Join with Towns to get Town
  left_join(Towns, by = "shortPostcode", relationship = "many-to-one") %>%
  # Filter for South and West Yorkshire
  filter(County %in% c("SOUTH YORKSHIRE", "WEST YORKSHIRE") & !is.na(Town)) %>%
  # Select final columns, **including Month**
  select(Postcode, shortPostcode, Town, District, County, CrimeType, Year, Month) %>%
  # Remove any remaining NA values
  filter(complete.cases(.))


# Write to CSV
write.csv(crime_clean, "Cleaned Data/cleanCrime.csv", row.names = FALSE)

View(crime_clean)
```

cleanCrime | cleanCrime.R

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 data cleaning process prepares Key Stage 4 (KS4) performance and school information datasets (2021-2022,2022-2023,2023-2024) trimmed and kept exam-year i.e.(2022-2024) for the South and West Yorkshire property investment analysis. For each year, school information datasets are processed to extract `URN`, `POSTCODE`, `shortPostcode` (uppercase, trimmed first four characters of `POSTCODE`), `District` (from `LANAME`), and `Town`. These are merged with the cleaned `Towns.csv` dataset to assign `County` using `shortPostcode`, filtering out records without valid counties. KS4 datasets are cleaned by selecting `URN`, `year`, `attainment_8_score` (from `ATT8SCR`), and `school_name`, filtering out missing `URN` or `attainment_8_score`. The cleaned school information and KS4 data are joined by `URN`, retaining records with valid `County`, `District`, `attainment_8_score`, and

`Town`. The annual datasets (2022–2024) are combined using `bind_rows`. Non-numeric `attainment_8_score` values (e.g., "SUPP", "NE", "NA") are removed, and the column is converted to numeric, with remaining `NA` values dropped. The final dataset, saved as `cleanSchool.csv`, ensures consistent, numeric school performance data for town-wise aggregation in the recommendation system and visualization.

cleanSchool.R

Source on Save | Run

```r
library(tidyverse)
library(stringr)

ks4_2022<-read_csv("Obtained Data/2021-2022/england_ks4final.csv")
school_info_2022<-read_csv("Obtained Data/2021-2022/england_school_information.csv")
Towns<-read_csv("Cleaned Data/TOWNS.csv")

cleanSchoolInfo2022<-school_info_2022 %>%
  mutate(
    District=LANAME,
    shortPostcode=toupper(str_trim(substr(POSTCODE,1,4))),
    Town=TOWN
    ) %>%
  select(URN,POSTCODE,shortPostcode,District,Town)

cleanTown<-Towns %>%
  mutate(shortPostcode=toupper(str_trim(shortPostcode)))

county_cleanSchoolInfo2022<-cleanSchoolInfo2022 %>%
  left_join(
    cleanTown %>% select(shortPostcode,County),
    by="shortPostcode"
  ) %>%
  drop_na(County)

ks4final<-ks4_2022 %>%
  mutate(
    year=2022,
    attainment_8_score=ATT8SCR,
    school_name=SCHNAME
  ) %>%
  select(URN,year,attainment_8_score,school_name) %>%
  filter(!is.na(URN), !is.na(attainment_8_score))

```

cleanSchool.R

Source on Save | Run

```
combined2022<-ks4final %>%
  left_join(county_cleanSchoolInfo2022, by="URN") %>%
  drop_na(County,District,attainment_8_score,Town)

ks4_2023<-read_csv("Obtained Data/2022-2023/england_ks4final.csv")
school_info_2023<-read_csv("Obtained Data/2022-2023/england_school_information.csv")

cleanSchoolInfo2023<-school_info_2023 %>%
  mutate(
    District=LANAME,
    shortPostcode=toupper(str_trim(substr(POSTCODE,1,4))),
    Town=TOWN
  ) %>%
  select(URN,POSTCODE,shortPostcode,District,Town)

cleanTown<-Towns %>%
  mutate(shortPostcode=toupper(str_trim(shortPostcode)))

county_cleanSchoolInfo2023<-cleanSchoolInfo2023 %>%
  left_join(
    cleanTown %>% select(shortPostcode,County),
    by="shortPostcode"
  ) %>%
  drop_na(County)

ks4final<-ks4_2023 %>%
  mutate(
    year=2023,
    attainment_8_score=ATT8SCR,
    school_name=SCHNAME
  ) %>%
  select(URN,year,attainment_8_score,school_name) %>%
  filter(!is.na(URN), !is.na(attainment_8_score))
```

cleanSchool.R

Source on Save | Run

```
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))
```

```r
combined2024<-ks4final %>%
  left_join(county_cleanSchoolInfo2024, by="URN") %>%
  drop_na(County,District,attainment_8_score,Town)

combinedSchool=bind_rows(combined2022,combined2023,combined2024)

# Clean the combinedSchool data by removing non-numeric values in attainment_8_score
combinedSchool_cleaned <- combinedSchool %>%
  filter(!attainment_8_score %in% c("SUPP", "NE", "NA")) %>%  # Remove unwanted codes
  mutate(attainment_8_score = as.numeric(attainment_8_score)) %>%  # Convert to numeric
  drop_na(attainment_8_score)  # Remove any remaining NA values

write_csv(combinedSchool_cleaned,"Cleaned Data/cleanSchool.csv")
```

cleanSchool.R × cleanSchool ×

Filter

	URN	year	attainment_8_score	school_name	POSTCODE	shortPostcode	District	Town	County
141	138361	2022	37.8	Notre Dame High School	S10 3BT	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	Belle Vue Girls' Academy	BD9 6NA	BD9	Bradford	Bradford	WEST YORKSHIRE

Exploratory Data Analysis

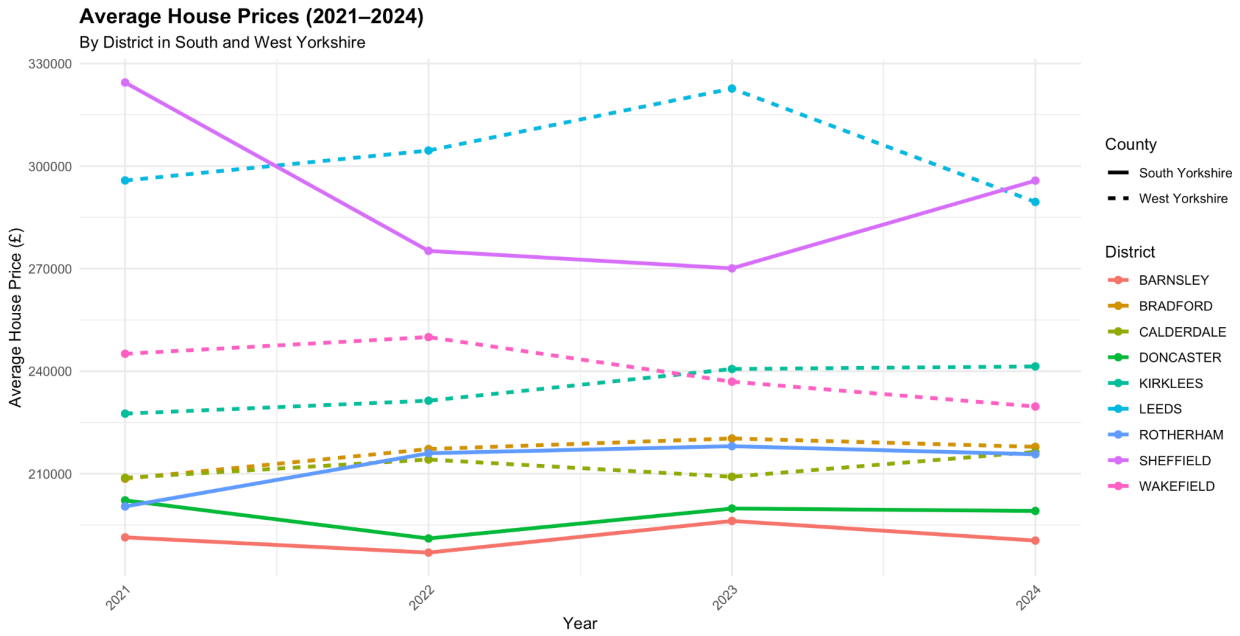
House Prices

1.Line Graphs

VisualizationHousePrice.R

Source on Save | Run

```r
library(tidyverse)
library(scales)

house<-read_csv("Cleaned Data/cleanHousePrices.csv")

# Define district groups
south_yorkshire_districts <- c("SHEFFIELD", "BARNSLEY", "DONCASTER", "ROTHERHAM")
west_yorkshire_districts  <- c("LEEDS", "BRADFORD", "CALDERDALE", "KIRKLEES", "WAKEFIELD")
yorkshire_districts <- c(south_yorkshire_districts, west_yorkshire_districts)

# Filter and clean data
house_filtered <- house %>%
  filter(District %in% yorkshire_districts, Year %in% 2021:2024) %>%
  mutate(District = toupper(District))

# Calculate average price per District and Year
avg_price_by_year <- house_filtered %>%
  group_by(Year, District) %>%
  summarise(avg_price = mean(Price, na.rm = TRUE), .groups = "drop") %>%
  mutate(
    County = case_when(
      District %in% south_yorkshire_districts ~ "South Yorkshire",
      District %in% west_yorkshire_districts ~ "West Yorkshire",
      TRUE ~ "Other"
    )
  )

# Plot: Line graph by District over Year
ggplot(avg_price_by_year, aes(x = Year, y = avg_price, color = District, linetype = County)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(
    title = "Average House Prices (2021–2024)",
    subtitle = "By District in South and West Yorkshire",
    x = "Year",
    y = "Average House Price (£)",
    color = "District"
  ) +
  theme_minimal() +
  theme(
```

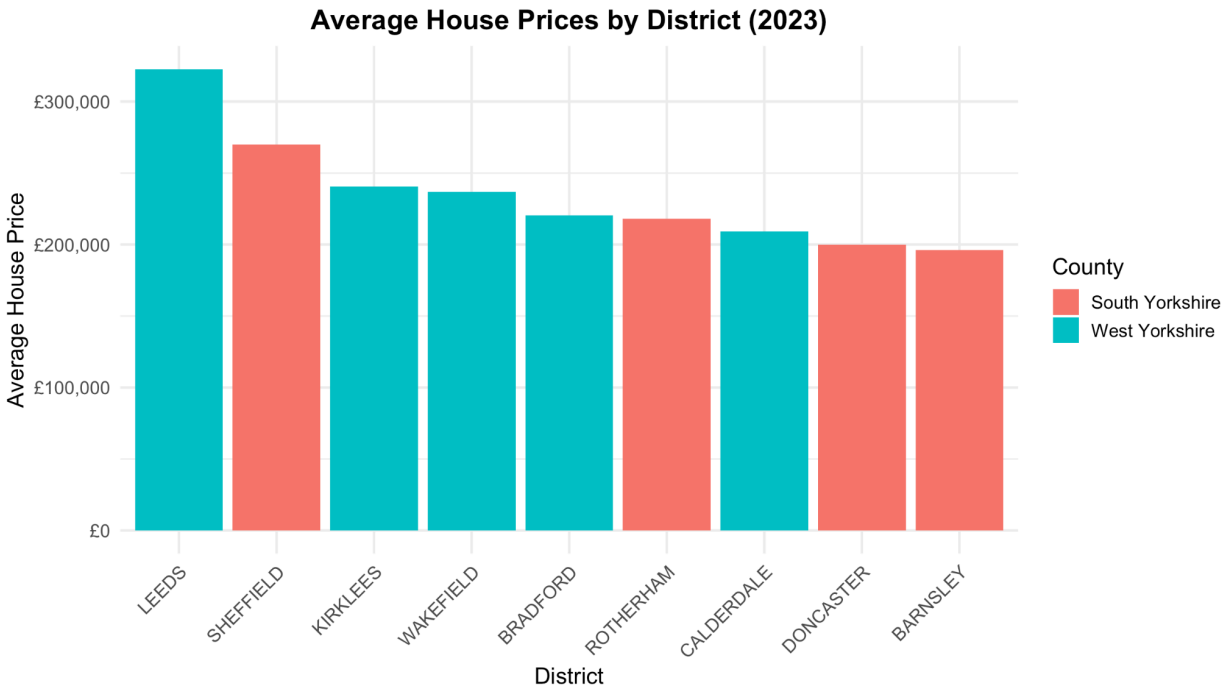


Interpretation:

The graph shows average house prices (2021–2024) across districts in South and West Yorkshire. Leeds consistently had the highest prices, peaking in 2023, while Barnsley had the lowest, with a slight dip in 2024. South Yorkshire districts, indicated by solid lines, generally remained more affordable than West Yorkshire districts (dashed lines). Notably, Sheffield saw a significant drop from 2021 to 2022 but began recovering by 2024. Overall, West Yorkshire showed greater price stability and gradual growth, while South Yorkshire exhibited more fluctuation, highlighting varied investment potential across the region.

2.Bar chart

```r
# 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)
  )
```

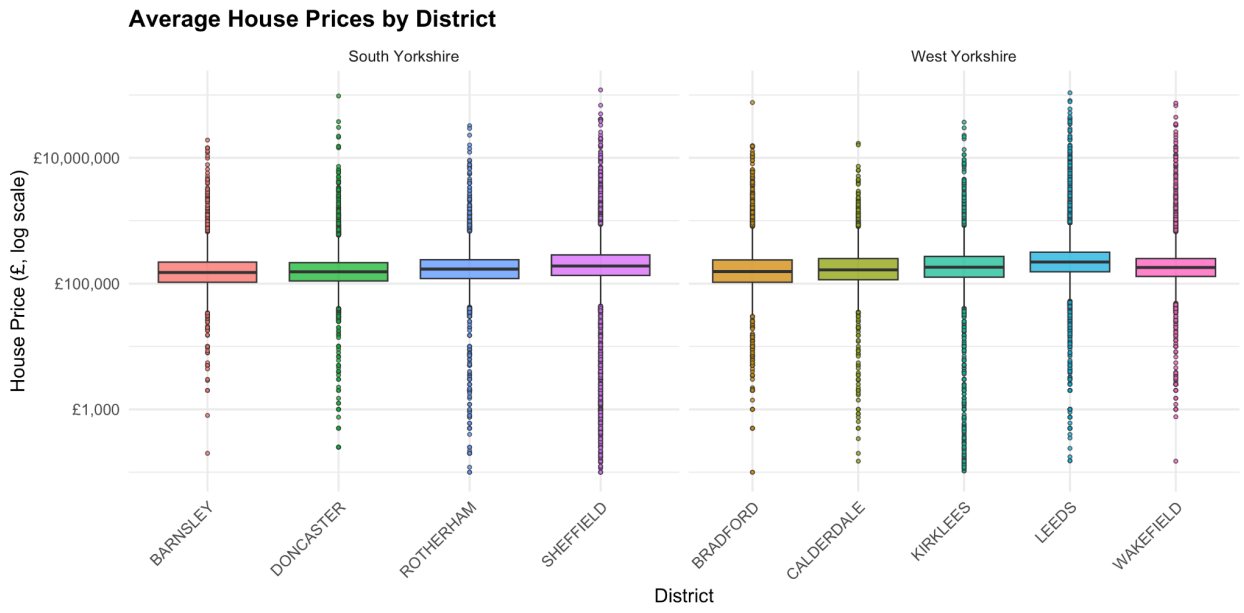


Interpretation:

The bar chart displays average house prices by district in 2023. Leeds (West Yorkshire) had the highest average price, exceeding £300,000, followed by Sheffield (South Yorkshire). Barnsley and Doncaster (both in South Yorkshire) had the lowest average prices, making them potentially more affordable for investors. West Yorkshire districts generally showed higher prices than those in South Yorkshire, except for Sheffield. This distribution highlights regional price disparities and suggests greater affordability in South Yorkshire, while Leeds stands out as the most expensive district across both counties.

3.Boxplot

```
# Filter for Yorkshire districts
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"
    )
  )

# Create boxplots faceted by County
ggplot(house_filtered, 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 by District",
    x = "District",
    y = "House Price (£, log scale)"
  ) +
  facet_wrap(~ County, scales = "free_x") +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(face = "bold"),
    legend.position = "none"
  )
```



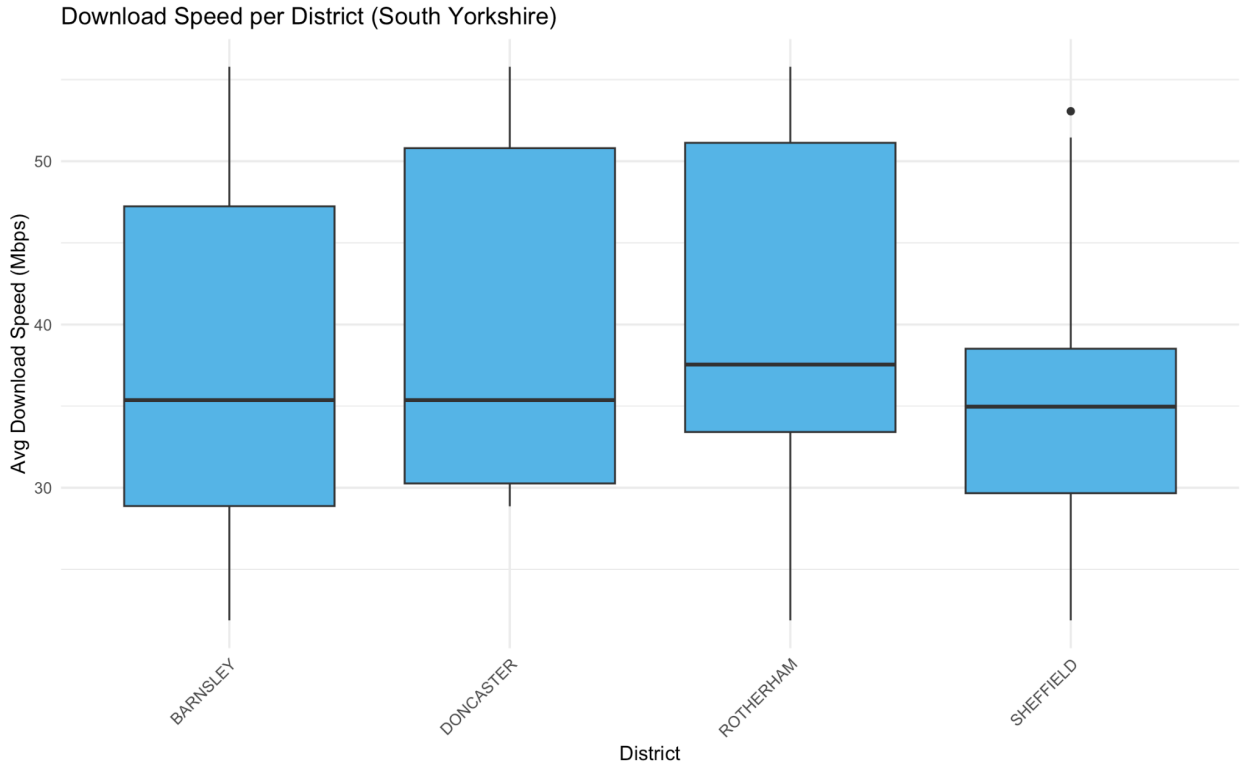
Interpretation:

The boxplot visualizes the distribution of house prices across districts in South and West Yorkshire on a log scale. Most districts show a median house price between £100,000 and £300,000. Leeds and Sheffield have higher medians and wider spreads, indicating a larger range of property values and more premium listings. All districts feature significant outliers, with some properties exceeding £1 million, particularly in Sheffield and Leeds. West Yorkshire districts like Leeds, Kirklees, and Wakefield show greater variability, suggesting diverse property markets. South Yorkshire districts, especially Barnsley and Doncaster, have narrower interquartile ranges and lower medians, indicating more affordability and consistency. The log scale highlights the skewed nature of house prices, with most transactions clustered in the mid-range and a few high-value outliers. This distribution is important for understanding local affordability and investment potential.

Broadband Speed

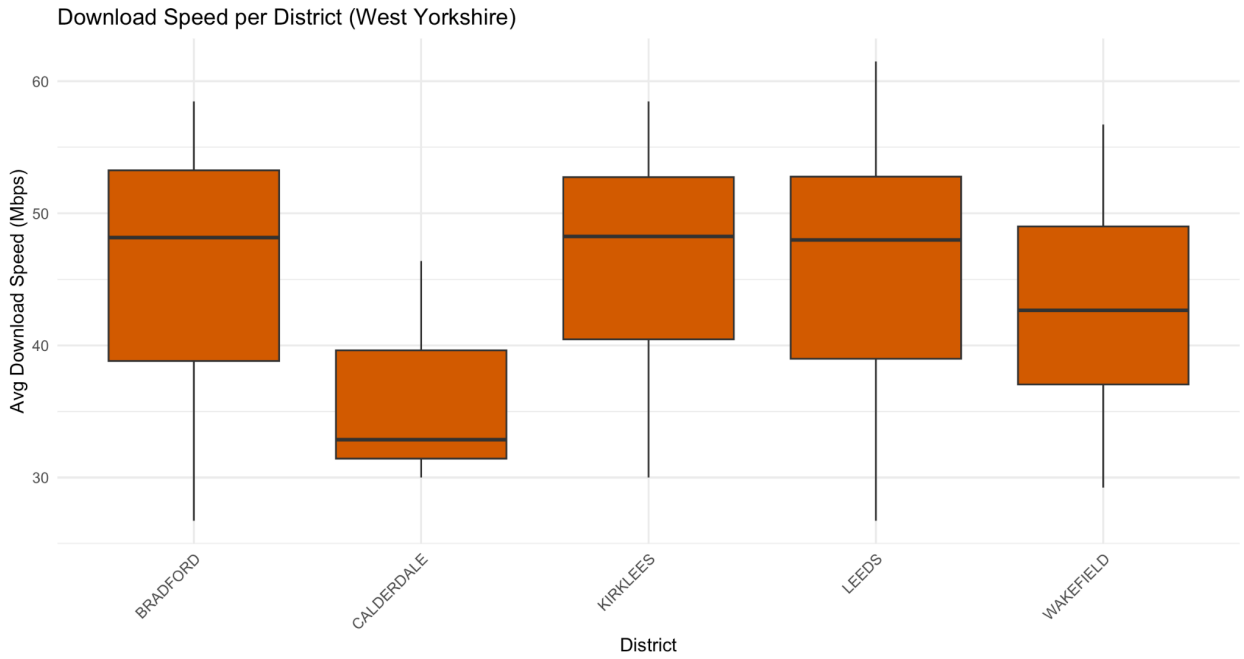
1.Boxplots

```
#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 displays average download speeds (Mbps) across four districts in South Yorkshire: Barnsley, Doncaster, Rotherham, and Sheffield. Each box represents the interquartile range (IQR), with the median speed marked by a horizontal line. Whiskers extend to the minimum and maximum speeds, with outliers shown as dots. Barnsley, Doncaster, and Rotherham have median speeds around 40 Mbps, with similar IQRs, indicating consistent performance. Sheffield shows a lower median speed, around 35 Mbps, with a wider IQR and an outlier suggesting variability. The plot highlights Sheffield as having the lowest and most variable download speeds among the districts.

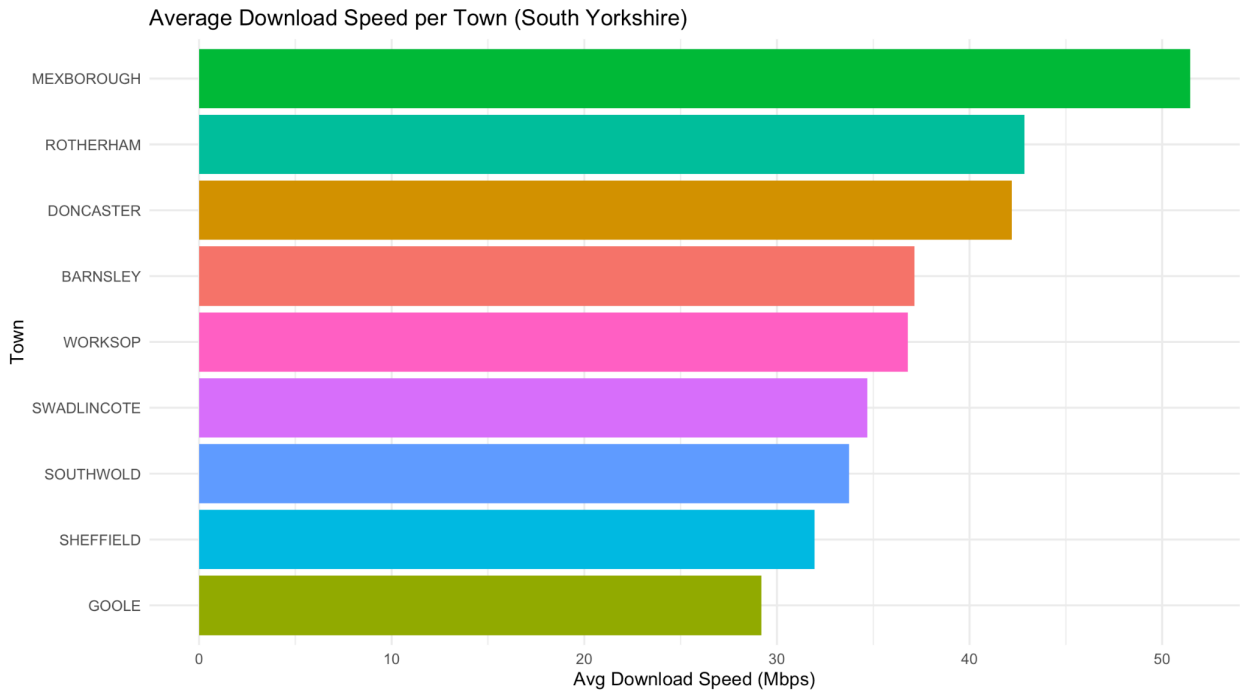


Interpretation:

The box plot illustrates average download speeds (Mbps) across five districts in West Yorkshire: Bradford, Calderdale, Kirklees, Leeds, and Wakefield. Each box shows the interquartile range (IQR), with the median speed marked by a line. Whiskers indicate the range, with outliers as dots. Bradford and Kirklees have median speeds around 45-50 Mbps, with similar IQRs. Calderdale shows a lower median, around 35-40 Mbps, with a tighter IQR. Leeds and Wakefield have medians around 40-45 Mbps, with Wakefield showing slight variability. The plot suggests Bradford and Kirklees offer the highest and most consistent download speeds, while Calderdale is the lowest.

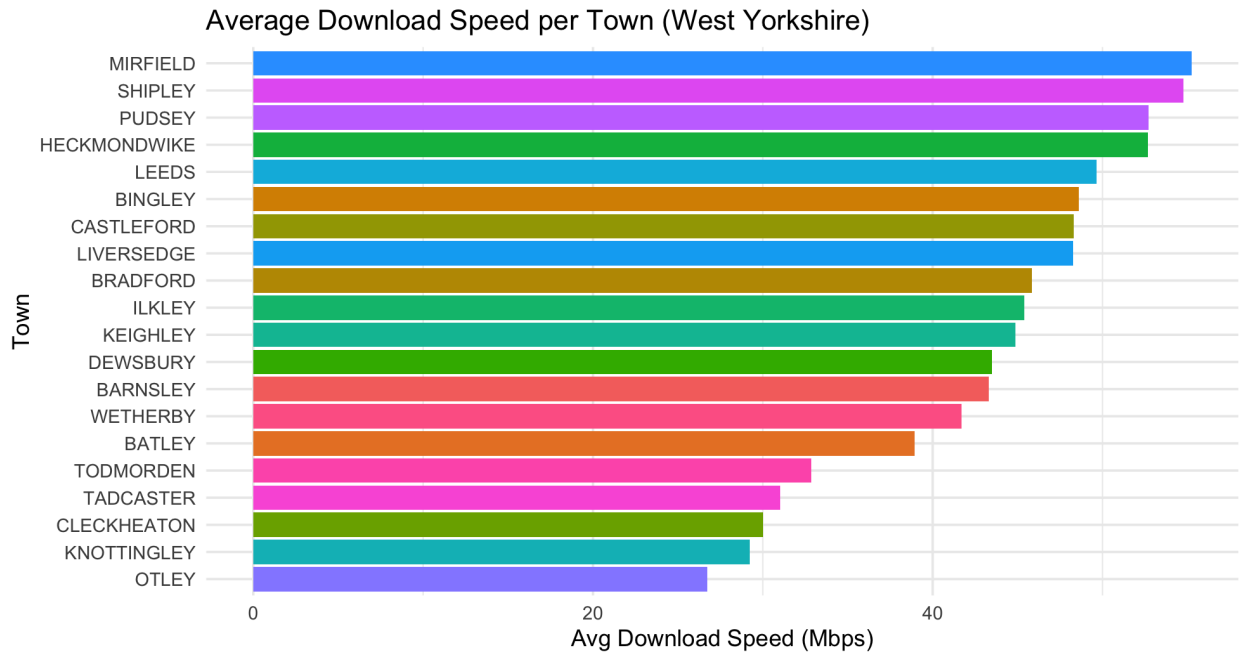
2.Barchart

```
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.



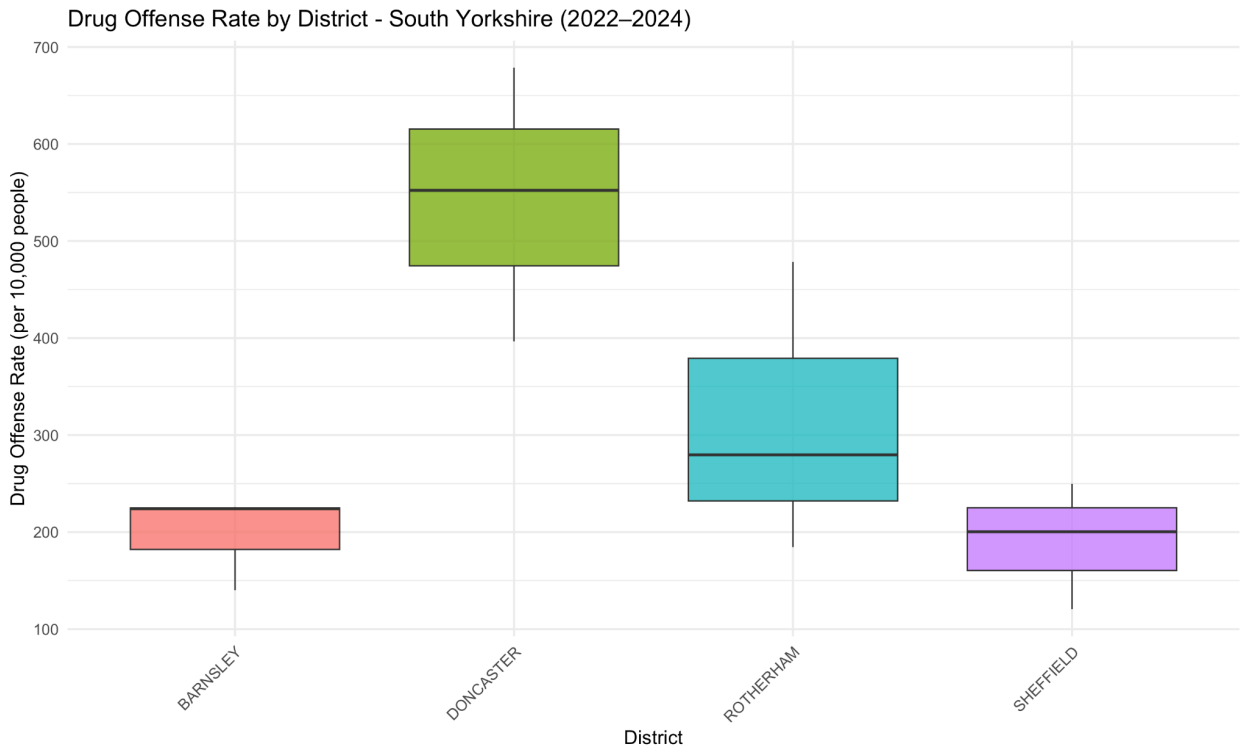
Interpretation:

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

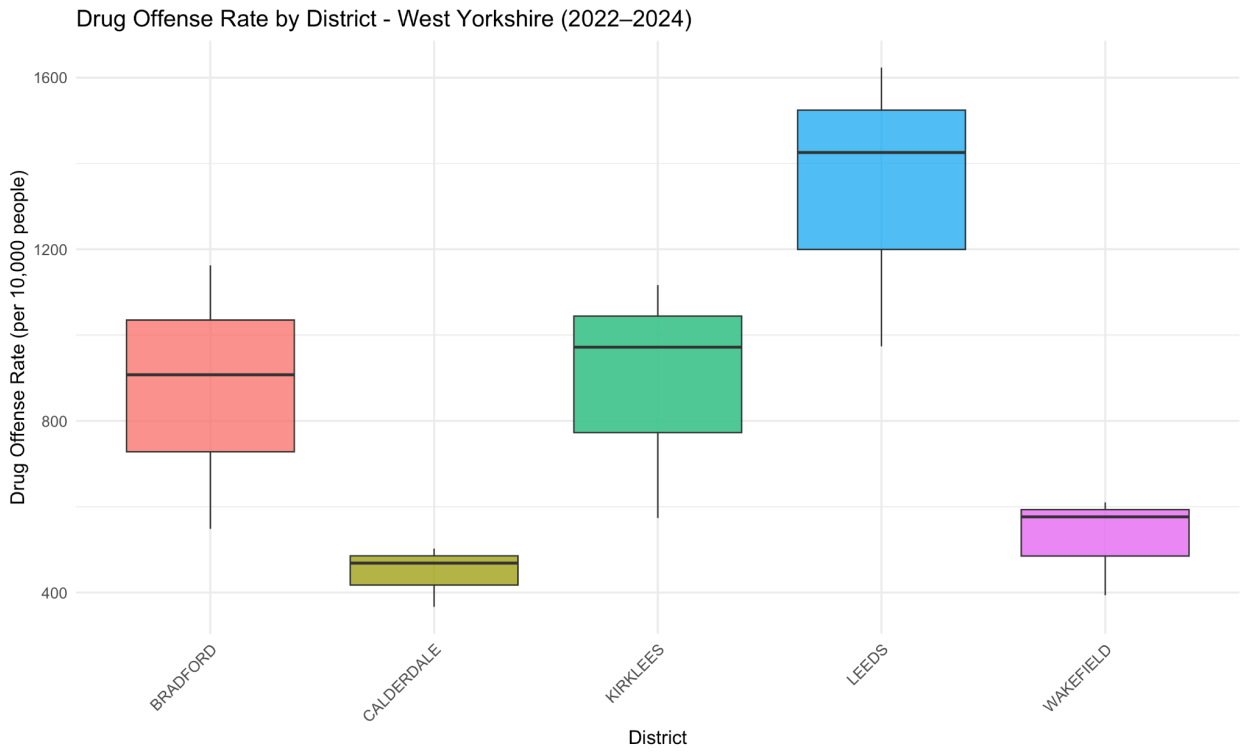
1.Boxplot

```r
# 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:

The box plot illustrates the drug offense rate per 10,000 people across four South Yorkshire districts (2022-2024). Doncaster has the highest median rate, around 600, with a wide interquartile range (IQR), indicating significant variability. Rotherham follows with a median near 350, showing a moderate IQR. Barnsley's median is around 200, with a tight IQR, suggesting consistency at a lower rate. Sheffield has the lowest median, approximately 200, with a similar tight IQR. Whiskers and outliers (e.g., in Doncaster and Rotherham) highlight occasional higher rates. The data suggests Doncaster faces the highest drug offense issues, while Sheffield and Barnsley are lower.



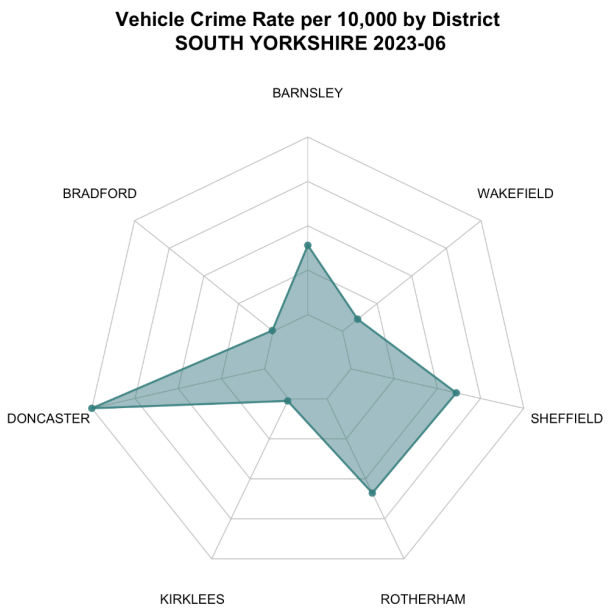
Interpretation:

The box plot displays the drug offense rate per 10,000 people across five West Yorkshire districts (2022-2024). Leeds has the highest median rate, around 1400, with a wide interquartile range (IQR), indicating significant variability. Bradford follows with a median near 900, also showing a broad IQR. Kirklees has a median around 800, with a moderate IQR. Calderdale's median is the lowest, around 400, with a tight IQR, suggesting consistency. Wakefield's median is around 500, with a narrow IQR. Outliers (e.g., in Leeds) highlight occasional spikes. The data indicates Leeds and Bradford face the highest drug offense rates, while Calderdale is 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",
           caxislabels = 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)
)
```



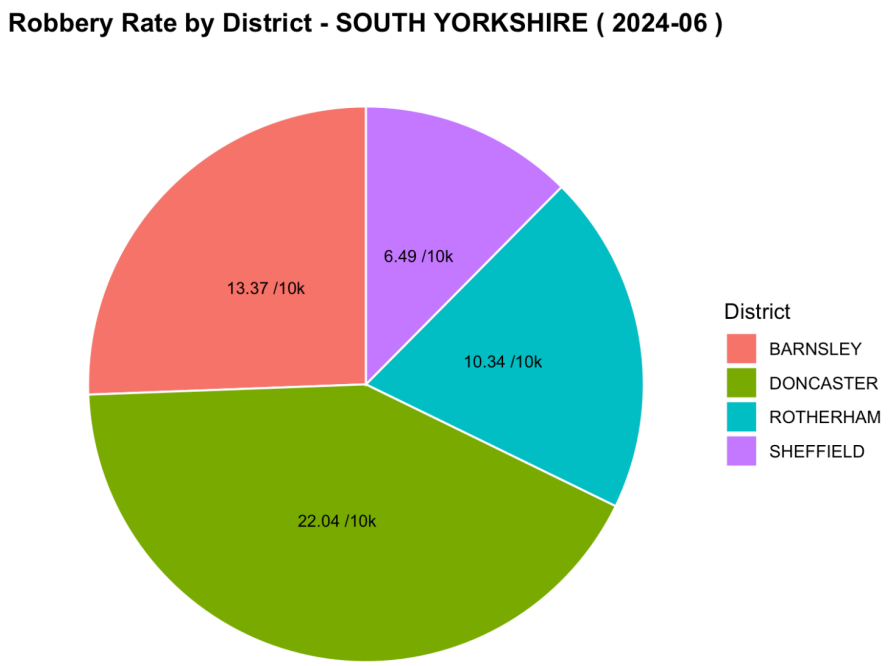
Interpretation:

The radar chart illustrates the vehicle crime rate per 10,000 people across six districts in South Yorkshire for 2023-06. Doncaster exhibits the highest rate, with a significant spike extending outward, indicating a notable issue. Barnsley and Wakefield show moderate rates, with Barnsley slightly higher. Sheffield and Kirklees have lower rates, with Sheffield being the lowest. Rotherham falls in the middle, with a balanced distribution. The chart suggests a wide variation in vehicle crime, with Doncaster facing the most challenges and Sheffield the least. The radial pattern highlights disparities, emphasizing Doncaster's outlier status among the districts.

3.Pie chart

```
# Define South Yorkshire districts
south_yorkshire_districts <- c("SHEFFIELD", "BARNSLEY", "ROTHERHAM", "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(robberyRate = (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"
  )
```

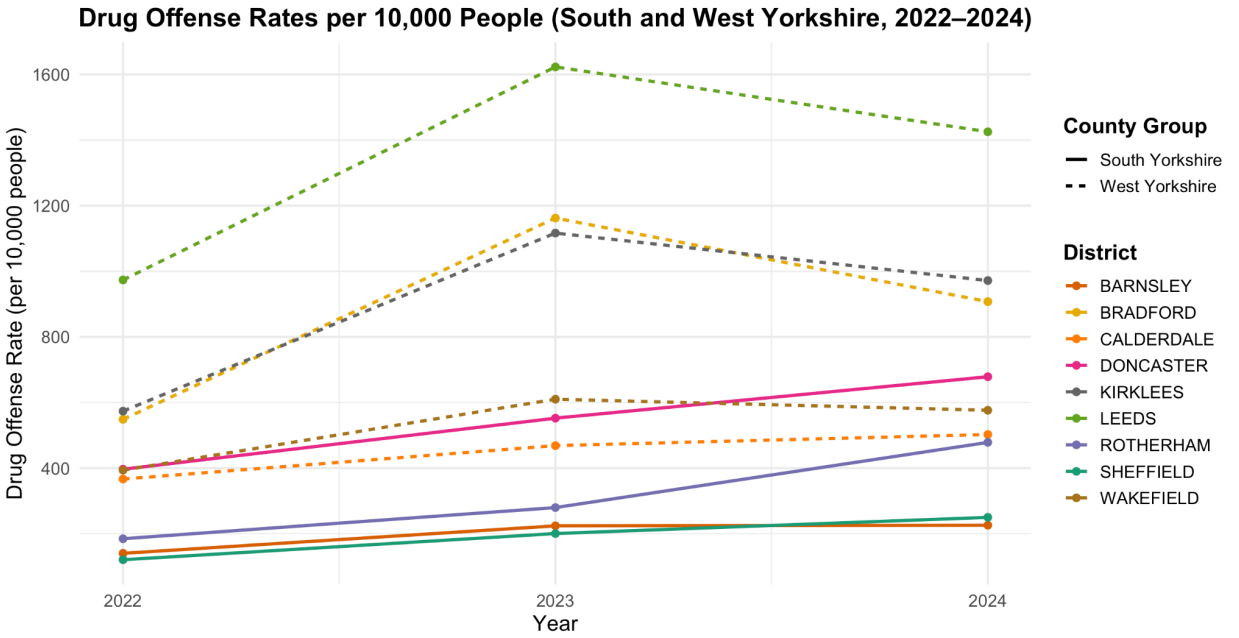


Interpretation:

The pie chart shows the robbery rate per 10,000 people across four South Yorkshire districts for 2024-06. Doncaster has the highest rate at 22.04/10k, occupying the largest segment. Barnsley follows with 13.37/10k, while Rotherham records 10.34/10k. Sheffield has the lowest rate at 6.49/10k, representing the smallest portion. The chart indicates a significant disparity, with Doncaster experiencing the highest robbery incidence, nearly double Sheffield's rate. Barnsley and Rotherham show moderate levels, with Barnsley slightly ahead. This suggests Doncaster faces the most robbery challenges, while Sheffield remains the safest among the districts.

4.Line chart

```r
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", "BARNSLEY" = "#d95f02", "ROTHERHAM" = "#7570b3", "DONCASTER" = "
    "LEEDS" = "#66a61e", "BRADFORD" = "#e6ab02", "WAKEFIELD" = "#a6761d", "KIRKLEES" = "#6666
  )) +
  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)
```



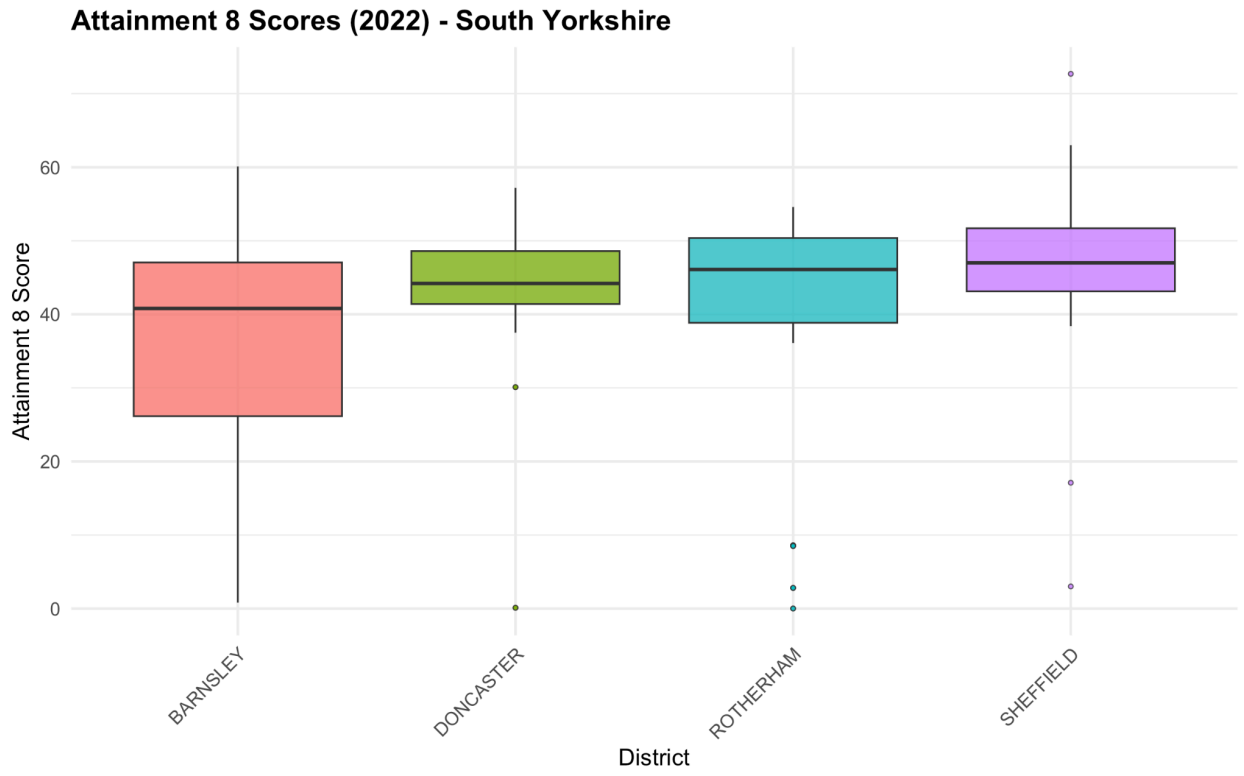
Interpretation:

The line graph tracks drug offense rates per 10,000 people in South and West Yorkshire (2022-2024). South Yorkshire (solid line) shows a peak in 2023 at 1600, declining to 1400 by 2024. West Yorkshire (dashed line) rises steadily from 1000 to 1200. Among districts, Leeds peaks at 1400 in 2023, then drops, while Bradford and Kirklees rise from 400 to 600-700. Barnsley, Calderdale, Doncaster, Rotherham, Sheffield, and Wakefield show moderate increases, ranging from 300 to 600. The data highlights a general rise in 2023, with Leeds and South Yorkshire facing the highest rates, followed by a slight 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", "BARNSLEY", "DONCASTER", "ROTHERHAM")

# 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"
  )
```

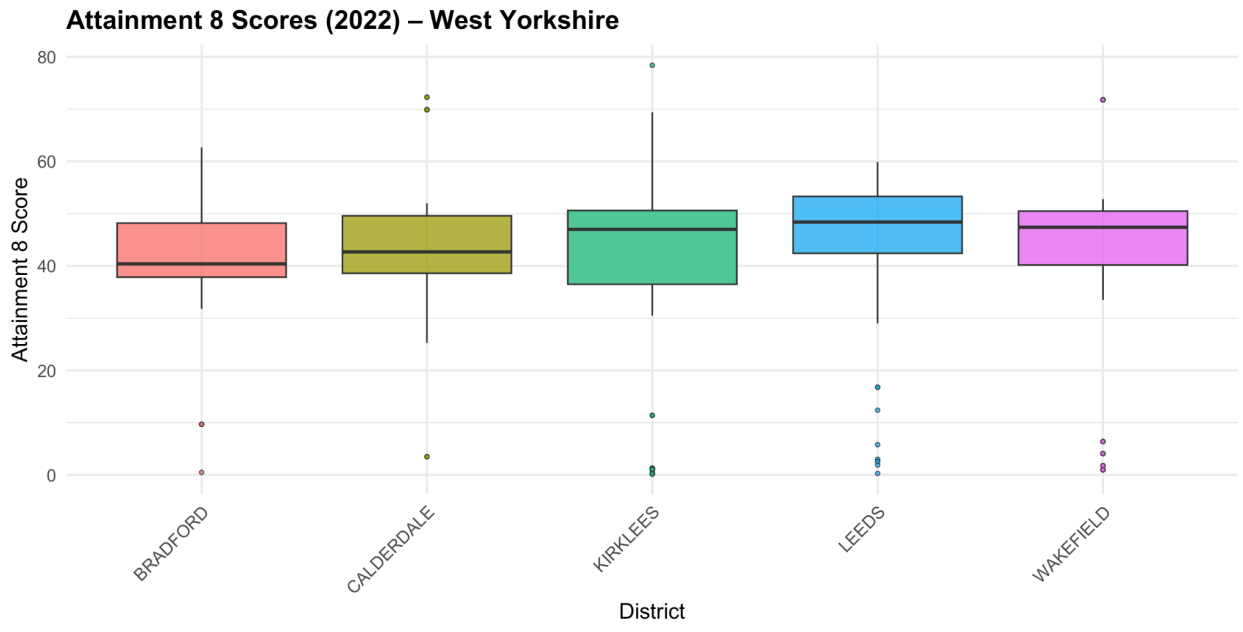


Interpretation:

The box plot displays attainment 8 scores for 2022 across four South Yorkshire districts: Barnsley, Doncaster, Rotherham, and Sheffield. Barnsley has the lowest median score, around 40, with a tight interquartile range (IQR). Doncaster's median is slightly higher, near 45, with an outlier below 20. Rotherham shows a median around 50, with a moderate IQR and an outlier near 10. Sheffield leads with a median near 50, a wide IQR, and an outlier above 60. The plot indicates Sheffield and Rotherham perform best, while Barnsley lags, with variability suggesting inconsistent school performance across 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"
  )
```

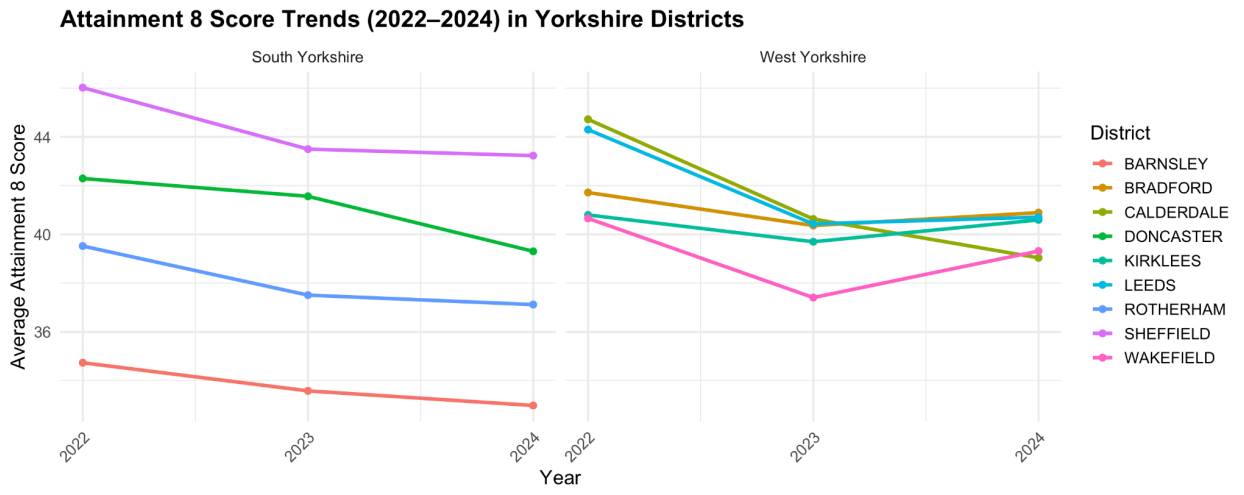


Interpretation:

The box plot shows attainment of 8 scores for 2022 across five West Yorkshire districts: Bradford, Calderdale, Kirklees, Leeds, and Wakefield. Bradford has the lowest median, around 40, with a tight interquartile range (IQR) and an outlier near 0. Calderdale's median is slightly higher, around 45, with an outlier near 70. Kirklees peaks at a median near 50, with a wide IQR. Leeds and Wakefield have medians around 50, with Wakefield showing an outlier above 70. The plot suggests Kirklees, Leeds, and Wakefield perform best, while Bradford lags, with variability indicating inconsistent educational outcomes across districts.

3.Line Graph

```r
# Filter for relevant districts and years
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"
    )
  )

# Group and summarise average Attainment 8 score by District and Year
school_summary <- school_filtered %>%
  group_by(County, District, year) %>%
  summarise(AvgAttainment8 = mean(attainment_8_score, na.rm = TRUE), .groups = "drop")

# Plot line graph
ggplot(school_summary, aes(x = year, y = AvgAttainment8, color = District, group = District)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  facet_wrap(~County) +
  scale_x_continuous(breaks = c(2022, 2023, 2024)) +
  labs(
    title = "Attainment 8 Score Trends (2022-2024) in Yorkshire Districts",
    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)
  )
```



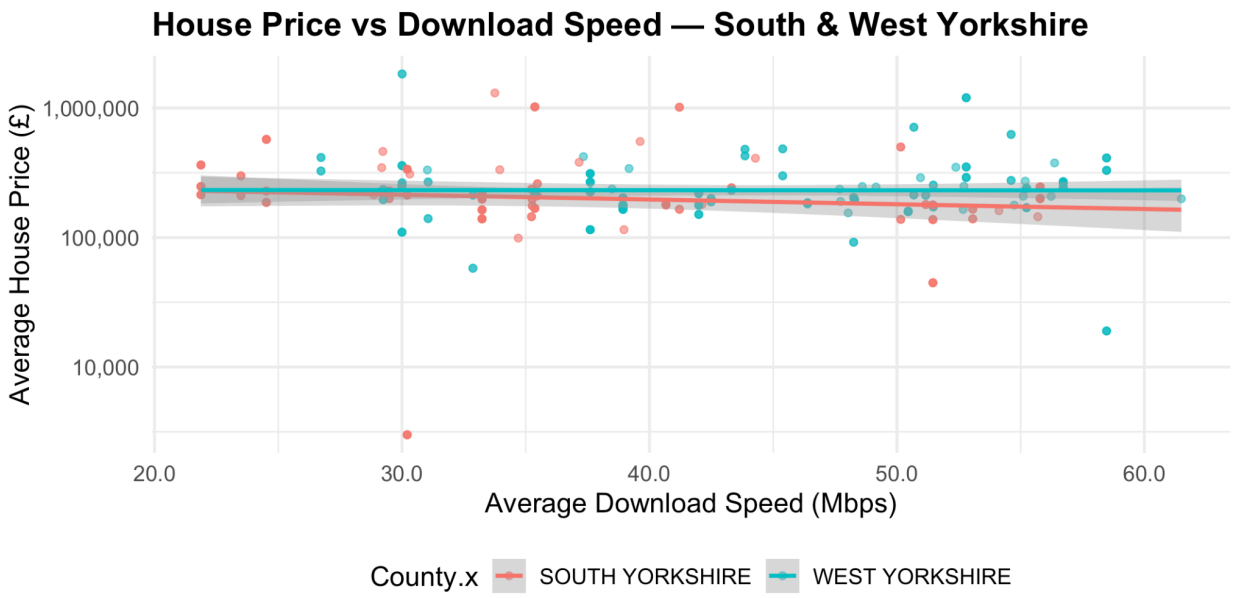
Interpretation:

The line graph tracks average attainment 8 score trends (2022-2024) across Yorkshire districts, split into South Yorkshire (left) and West Yorkshire (right). In South Yorkshire, Barnsley starts at 44, declining to 42 by 2024. Doncaster remains stable around 42, while Rotherham drops from 40 to 38. Sheffield shows a slight decline from 44 to 43. In West Yorkshire, Bradford starts at 36, dipping to 34 by 2024. Calderdale and Kirklees fluctuate around 38-40, with Kirklees peaking at 40 in 2022. Leeds holds steady at 38-39, while Wakefield rises from 36 to 38. Overall, South Yorkshire districts generally score higher (40-44) than West Yorkshire (34-40). Trends indicate a slight decline in most areas, with Wakefield as an exception, showing improvement. The data suggests educational performance varies, with South Yorkshire maintaining a lead, though all districts experience minor fluctuations, reflecting inconsistent progress over the period.

Linear Modelling

1.House Price vs Download Speed

```r
library(tidyverse)

house<-read_csv("Cleaned Data/cleanHousePrices.csv")
broadband<-read_csv("Cleaned Data/cleanBroadbandPerformance.csv")

# Step 1: Average house price per shortPostcode
avg_price <- house %>%
  group_by(shortPostcode, District, County) %>%
  summarise(AvgPrice = mean(Price, na.rm = TRUE), .groups = "drop")

avg_broadband<- broadband %>%
  group_by(shortPostcode, District,County) %>%
  summarise(AvgDownload = mean(AvgDownload, na.rm = TRUE), .groups = "drop")


# Step 2: Merge house price and broadband data
merged_df <- avg_price %>%
  inner_join(avg_broadband, by = "shortPostcode") %>%
  filter(!is.na(AvgDownload), !is.na(AvgPrice))


# Step 3: Correlation
cor_test <- cor.test(merged_df$AvgPrice, merged_df$AvgDownload)
print(cor_test)

# Step 4: Linear model
model <- lm(AvgPrice ~ AvgDownload, data = merged_df)
summary_model <- summary(model)
print(summary_model)

# Step 5: Scatter plot with regression line
ggplot(merged_df, aes(x = AvgDownload, y = AvgPrice, color = County.x)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = TRUE, fullrange=TRUE) +
  scale_x_continuous(name = "Average Download Speed (Mbps)", labels = label_number(accuracy = 0.1)) +
  scale_y_log10(name = "Average House Price (£)", labels = label_number(big.mark = ",")) +
  labs(
    title = "House Price vs Download Speed – South & West Yorkshire",
  ) +
  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

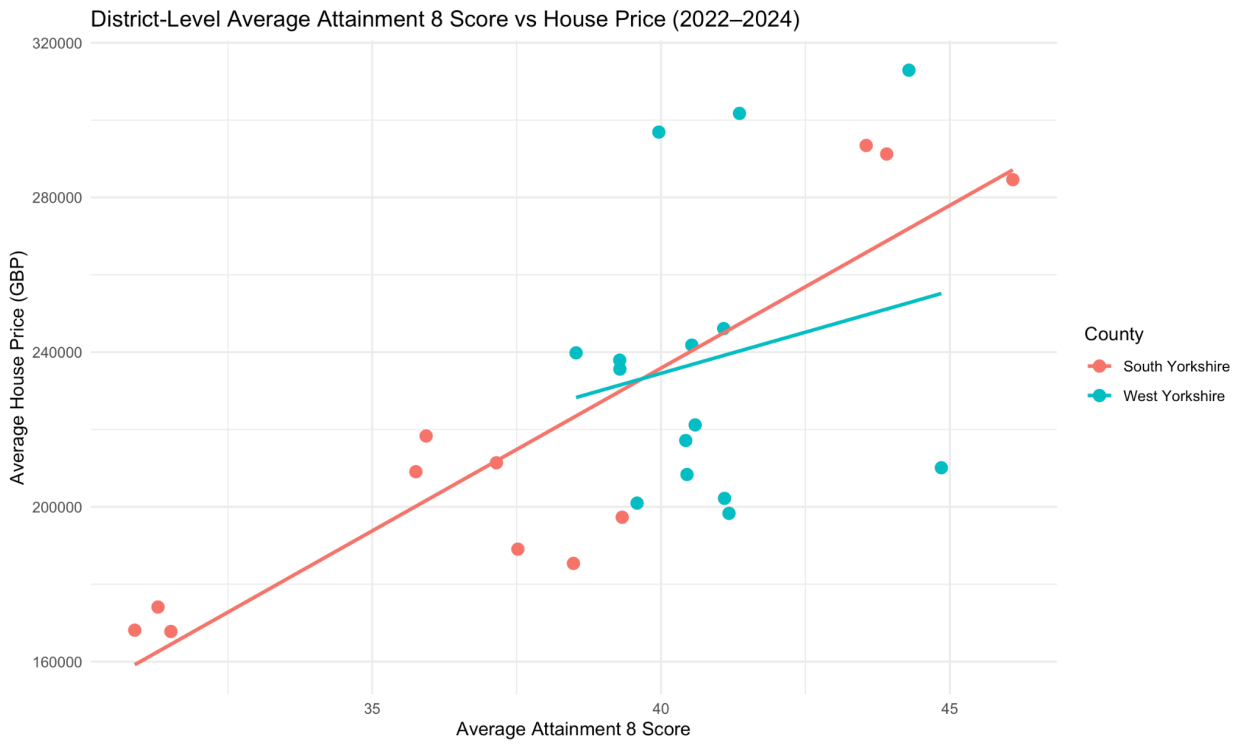
Analysis:

The scatter plot illustrates the relationship between average house prices and download speeds in South and West Yorkshire, with a linear model fitted for each county. The graph shows no clear trend, supported by a weak negative correlation (r = -0.044, p = 0.4516), indicating no significant linear relationship. The linear model summary reveals an intercept of 312,184.5, a non-significant slope of -953.6 (p = 0.452), and a low R-squared (0.001948), suggesting the model explains little variance. The p-value (0.4516) confirms the lack of predictive power, highlighting that download speed does not strongly influence house prices.

2.Attainment 8 score vs House Price

```r
# Load libraries
library(tidyverse)


south_yorkshire_districts <- c("SHEFFIELD", "BARNSLEY", "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(
```

```r
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:

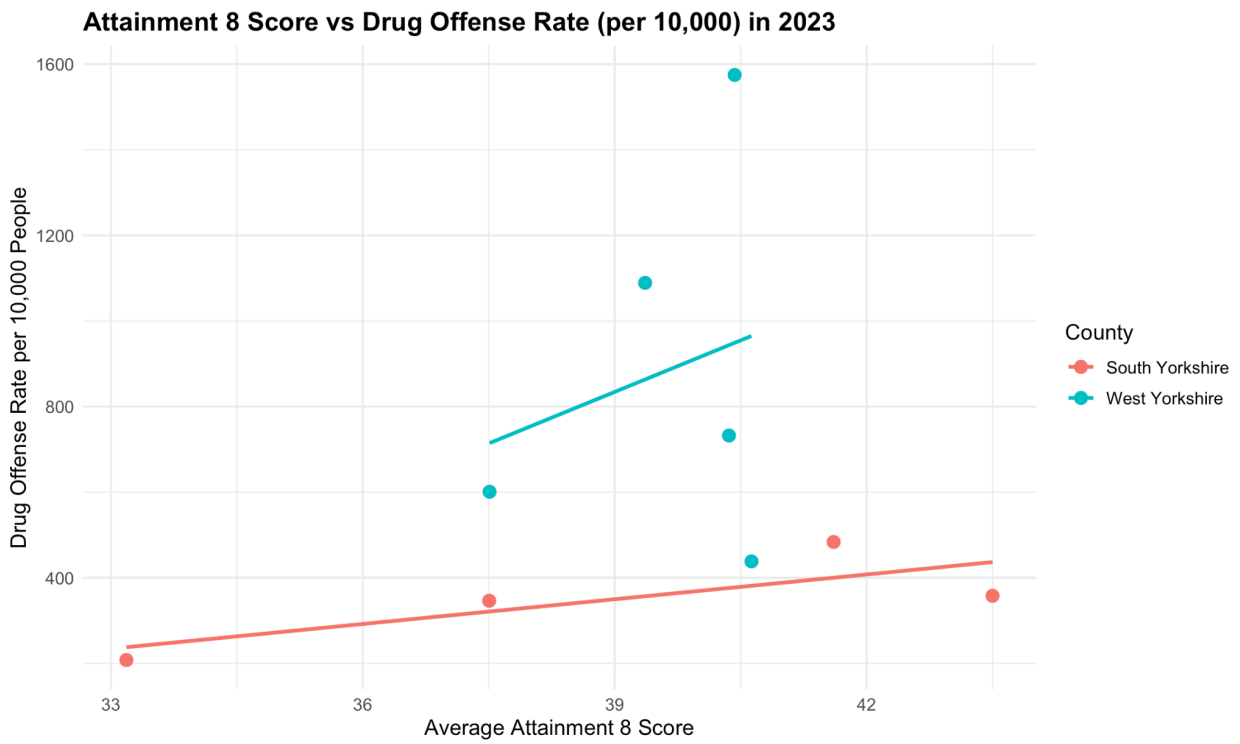
The scatter plot illustrates the relationship between district-level average Attainment 8 scores and house prices for South and West Yorkshire, with separate linear fits. A clear positive trend is observed, supported by a Pearson correlation coefficient of 0.695 (p = 5.761e-05), indicating a strong, significant positive relationship. The linear model (lm(avg_price ~ avg_attainment8)) reveals an intercept of -75,499, a slope of 7,707 (p = 5.76e-05), and an R-squared of 0.4829, explaining nearly half the variance. The low p-value confirms Attainment 8 scores are a significant predictor of house prices.

3.Attainment 8 score vs Drug (2023)

```r
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,
```

```r
south_yorkshire <- c("SHEFFIELD", "BARNSLEY", "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(attainment_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)

#    Linear Model
```

```r
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:

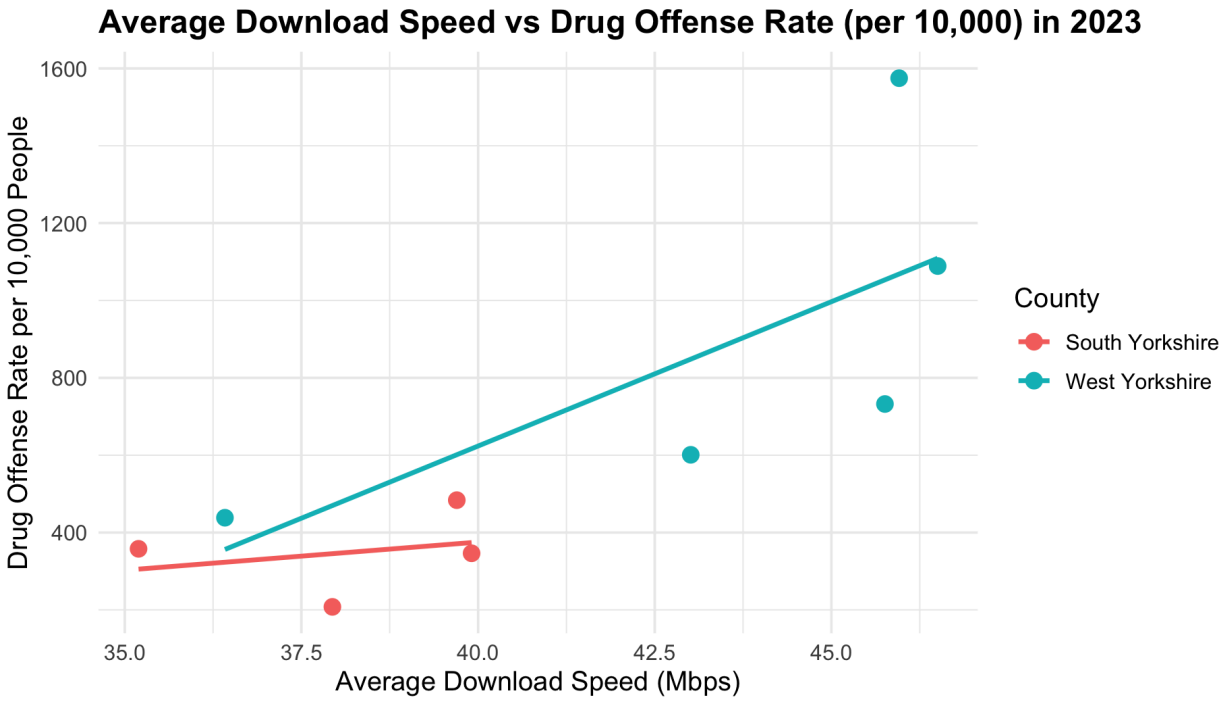
The scatter plot illustrates the relationship between average Attainment 8 scores and drug offense rates per 10,000 people in 2023 for South and West Yorkshire, with separate linear fits. A positive trend is evident, with South Yorkshire showing a strong correlation (r = 0.784) and West Yorkshire a weaker one (r = 0.230). The linear model (lm(DrugOffenseRate ~ avg_attainment8 * CountyGroup)) yields an intercept of -401.84, a non-significant slope of 19.28 (p = 0.716), and an R-squared of 0.4727, indicating moderate fit but low significance (p = 0.3234). Interaction terms with CountyGroup are also non-significant.

4.Download speed vs Drug

```r
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"
  )
```

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

# Define Yorkshire districts
south_yorkshire <- c("SHEFFIELD", "BARNSLEY", "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)
```

```r
# --- 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

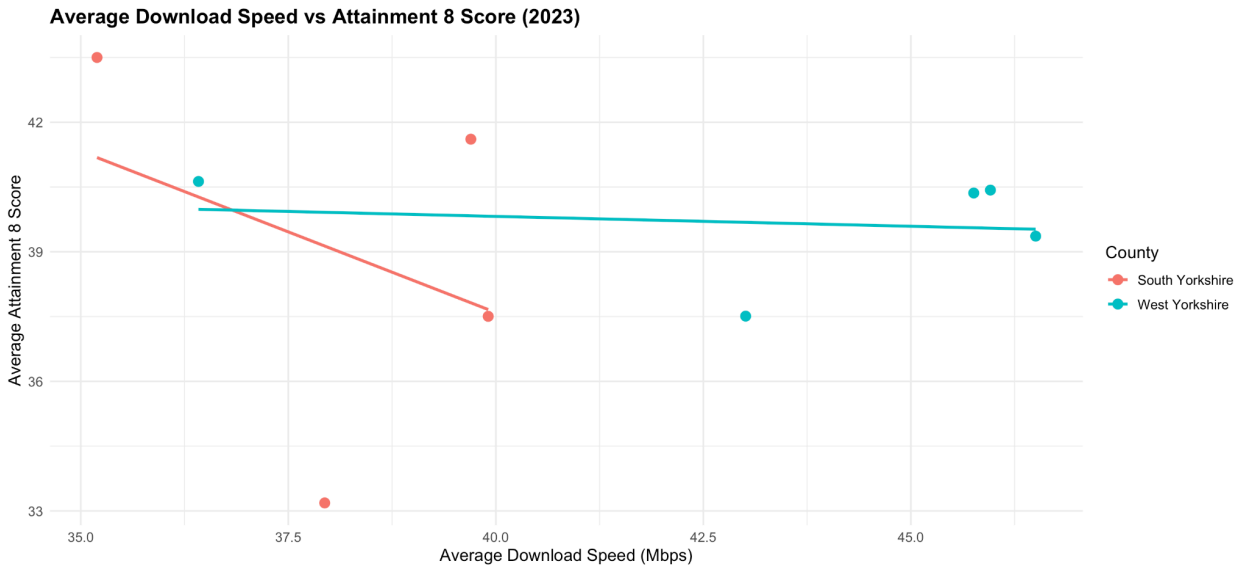
Analysis:

The scatter plot illustrates the relationship between average download speed (Mbps) and drug offense rates per 10,000 people in 2023 for South and West Yorkshire, with separate linear fits. A positive trend is evident, with West Yorkshire showing a stronger correlation (r = 0.692) and South Yorkshire a weaker one (r = 0.281). The linear model (lm(DrugOffenseRate ~ avg_download * CountyGroup)) yields 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

```r
library(tidyverse)
library(scales)


broadband <- read_csv("Cleaned Data/cleanBroadbandPerformance.csv") %>%
  mutate(District = toupper(District), County = toupper(County)) %>%
  group_by(District, County) %>%
  summarise(
    avg_download = mean(AvgDownload, na.rm = TRUE),
    .groups = "drop"
  )

# --- 2. Load and prepare school data for Attainment 8 (2023) ---
school <- read_csv("Cleaned Data/cleanSchool.csv") %>%
  mutate(
    District = toupper(District),
    County   = toupper(County),
    Year     = year
  ) %>%
  filter(Year == 2023) %>%
  group_by(District, County) %>%
  summarise(
    avg_attainment8 = mean(attainment_8_score, na.rm = TRUE),
    .groups = "drop"
  )

# --- 3. Define target districts ---
south_yorkshire <- c("SHEFFIELD", "BARNSLEY", "ROTHERHAM", "DONCASTER")
west_yorkshire  <- c("LEEDS", "BRADFORD", "WAKEFIELD", "KIRKLEES", "CALDERDALE")
yorkshire_districts <- c(south_yorkshire, west_yorkshire)

# --- 4. Filter to only those districts ---
broadband_clean <- broadband %>% filter(District %in% yorkshire_districts)
school_clean    <- school    %>% filter(District %in% yorkshire_districts)

```

```r
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 depicts the relationship between average download speed (Mbps) and Attainment 8 scores in 2023 for South and West Yorkshire, with separate linear fits. A weak negative trend is evident, with South Yorkshire showing a correlation of -0.355 and West Yorkshire -0.148, indicating low relationships. The linear model (lm(avg_attainment8 ~ avg_download * CountyGroup)) yields an intercept of 67.4609, a non-significant slope of -0.7467 (p = 0.459), and an R-squared of 0.1299, suggesting poor fit (p = 0.8592). Interaction terms with CountyGroup are also non-significant.

Recommendation System

```r
# 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")])))
```

```r
# 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
```

```r
# 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 |
```

```r
# 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
ROTHERHAM	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 towns across South and West Yorkshire based on four key factors: affordability, broadband speed, crime rate, and school performance. Each factor was normalized to a 0–10 scale, with affordability weighted highest (40%), and the others equally (20%). By integrating multiple dimensions of livability, the system identifies locations best suited for property investment.

Results

Mexborough in South Yorkshire ranked highest with a strong combination of affordability and safety. Heckmondwike and Mirfield in West Yorkshire followed closely, each demonstrating excellent broadband connectivity and low crime. The top 10 towns include a mix of both counties, reflecting the strength of West Yorkshire's connectivity and South Yorkshire's affordability. Final rankings were based on a composite score combining all four indicators.

Reflection

The system offers a balanced, data-driven perspective, though limited by missing or uneven data for some towns. Population estimates and averaging may not fully capture local nuances. Future improvements could involve more dynamic weights based on user preference or including public transport and health access. Nonetheless, the approach provides a clear, comparable way to assess multiple towns holistically.

Broadband Speeds

West Yorkshire towns generally outperformed South Yorkshire in broadband speeds. Mirfield and Heckmondwike topped the list with scores above 9. These high speeds contribute significantly to their overall desirability for remote work or streaming, helping them achieve strong composite scores in the final recommendation.

School Grades

School performance varied widely across towns, with Bingley and Mirfield leading on educational quality. Mexborough and Rotherham, despite being top-ranked overall, had relatively modest school scores. This suggests that high affordability or safety can outweigh education in the final score depending on individual priorities.

House Prices

South Yorkshire consistently offers more affordable housing, with Mexborough and Rotherham scoring highest on affordability. This aligns with investment goals focused on lower entry costs. West Yorkshire towns tended to have higher average prices, which reduced their affordability scores despite strong performance in other areas.

Crimes

Safety scores were derived from crime rate per 1,000 people, averaged from 2022 to 2024. Most top towns had crime rates low enough to score above 9 in safety, notably Mexborough, Mirfield, and Bingley. This metric heavily influenced final rankings, 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