Report on Town Recommendation System

Aashista Karki

B.Sc. Computing Softwarica College of IT and E-commerce

ST5014CEM Data Science for Developers

Siddhartha Neupane

Jan 24 2024

Table of Contents

Table of Figures

Introduction

This Individual Coursework endeavors to develop a town recommender system catering to the needs of international students planning a study exchange program in England. The primary goal is to assist in making informed decisions about towns in the counties of Kent and Surrey. Recommendations are based on a detailed analysis of various factors such as educational institutions, cost of living (exemplified by house prices), broadband speed, and safety ratings (illustrated by local crime statistics).

The exclusive use of datasets released by the UK government ensures the reliability of the analysis. These datasets are sourced from reputable entities, primarily accessed through the https://data.gov.uk/ website, aligning with ethical and legal considerations.

The steps involved in this project is first data gathering using official uk government data, secondly data cleaning using r language, thirdly to plot out various graphs as the need of project and lastly to recommend town based on the top scores.

The report unfolds with a comprehensive Exploratory Data Analysis (EDA) section, using graphical plots and summary statistics to comprehend the distribution of single-variable data, identify outliers, and investigate relationships between variables through plots and correlation coefficients.

The report concludes with a discussion of legal and ethical considerations associated with the data used and the recommendation system developed. Additionally, it reflects on the application of the data mining lifecycle to this problem, summarizes conclusions, and offers recommendations for future improvements or extensions.

Cleaning data

Cleaning data is the process of detecting and rectifying errors in datasets, including addressing issues like missing values, duplicates, and inconsistencies. This practice is essential for bolstering the accuracy and comprehensiveness of data, thereby improving the trustworthiness of analyses and decisionmaking. Employing techniques like exploratory analysis aids in honing the quality of data. The importance of clean data lies in its ability to predict errors and the formation of deceptive conclusions, underscoring its pivotal role in generating meaningful insights.

The obtained data will be then cleaned to be used for plotting , summary, graphs etc. Various commands such as omit, distinct, mean etc will be used to clean the data to make it duplicate free and to remove null values and make the data clean. After the data is cleaned it is written in a new csv file using write.csv command.

House pricing cleaning data

Upon acquiring the dataset, new names were assigned to the columns. Subsequently, the data for each individual year underwent filtering to eliminate any unnecessary variables. Finally, the data from all the years were merged into a unified tibble for comprehensive analysis.

Figure 1- House price cleaning1

```r

#----------2019 Dataset Cleaning----------#

#Cleaning data through the use of pipe operator
houseprices_2019 <-read_csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/housing-price/pp-2019.csv", col_names = FALSE) %>%  #Importing CSV into R
  setNames(c("Transaction unique identifier", "Price", "Date of Transfer", "Postcode", "Property Type", "Old/New", "Duration", "PAON",
             "SAON", "Street", "Locality", "Town/City", "District", "County", "PPD Category type", "Record Status")) %>% #Changing Column name
  as_tibble() %>% #Converting into tibble
  na.omit() %>% #Removing rows with null value
  select(Price, `Date of Transfer`, Postcode, `Town/City`, District, County) %>% #selecting only columns that are required
  filter(County =="KENT" | County== "SURREY") %>% #Preserving rows with Kent and Surrey as county
  mutate(`Date of Transfer` = year(as.Date(`Date of Transfer`, format = "%y/%m/%d"))) %>%  #modifying the date of transfer column to only show year
  mutate(S_No = row_number()) %>% #Adding a new serial number column
  select(S_No, everything()) #moving the serial number column at first


#----------2020 Dataset Cleaning----------#

houseprices_2020 <-read_csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/housing-price/pp-2020.csv", col_names = FALSE) %>%  #Importing CSV into R
  setNames(c("Transaction unique identifier", "Price", "Date of Transfer", "Postcode", "Property Type", "Old/New", "Duration", "PAON",
             "SAON", "Street", "Locality", "Town/City", "District", "County", "PPD Category type", "Record Status")) %>% #Changing Column name
  as_tibble() %>% #Converting into tibble
  na.omit() %>% #Removing rows with null value
  select(Price, `Date of Transfer`, Postcode, `Town/City`, District, County) %>% #selecting only columns that are required
  filter(County =="KENT" | County== "SURREY") %>% #Preserving rows with Kent and Surrey as county
  mutate(`Date of Transfer` = year(as.Date(`Date of Transfer`, format = "%y/%m/%d"))) %>%  #modifying the date of transfer column to only show year
  mutate(S_No = row_number()) %>% #Adding a new serial number column
  select(S_No, everything()) #moving the serial number column at first

#----------2021 Dataset Cleaning----------#

houseprices_2021 <-read_csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/housing-price/pp-2021.csv", col_names = FALSE) %>%  #Importing CSV into R
  setNames(c("Transaction unique identifier", "Price", "Date of Transfer", "Postcode", "Property Type", "Old/New", "Duration", "PAON",
             "SAON", "Street", "Locality", "Town/City", "District", "County", "PPD Category type", "Record Status")) %>% #Changing Column name
  as_tibble() %>% #Converting into tibble
  na.omit() %>% #Removing rows with null value
  select(Price, `Date of Transfer`, Postcode, `Town/City`, District, County) %>% #selecting only columns that are required
  filter(County =="KENT" | County== "SURREY") %>% #Preserving rows with Kent and Surrey as county
  mutate(`Date of Transfer` = year(as.Date(`Date of Transfer`, format = "%y/%m/%d"))) %>%  #modifying the date of transfer column to only show year
  mutate(S_No = row_number()) %>% #Adding a new serial number column
  select(S_No, everything()) #moving the serial number column at first
```

Figure 2-House price cleaning 2

```
#merging all the cleaned dataset into a single tibble

combined_houseprices<- bind_rows(houseprices_2019, houseprices_2020, houseprices_2021, houseprices_2022) %>%
  mutate(`Short Postcode`= substr(Postcode, 1,5)) #adding another column to the combine dataset


#defining path to save the cleaned dataset
file_path <- "C:/Users/aasis/Desktop/DataScience-Assignment/Clean-data/Cleaned House Prices.csv"


```

Population Cleaning data

After obtaining the population data it was imported into r studio. The na values as well as duplicated values were to be removed using omit and distinct commands. But the population data was already clean so it was written into a csv file.

Figure 3-Population cleaning data

```
library(tidyverse)
library(dplyr)

population_data=read.csv("Obtained-data/Population2011_1656567141570.csv")
population_data

clean_pop= population_data %>%
  na.omit() %>%
  distinct()

#no na data or duplicate data found in this pop csv file

write_csv(clean_pop, "C:/Users/aasis/Desktop/DataScience-Assignment/Clean-data/Cleaned Population.csv")

```

Broadband Speed Cleaning data

This R code begins by loading essential libraries for data manipulation and setting the working directory. It then imports a cleaned dataset linking postcodes to LSOA codes. The code proceeds to read a broadband speed dataset, selecting relevant columns, renaming them, and joining the data with the cleaned postcode-to-LSOA dataset. Additional data manipulations include selecting specific columns, handling missing values, and adding a serial number column. The cleaned broadband speed dataset is finally saved as a CSV file. Overall, the code ensures data consistency, prepares it for analysis, and stores the cleaned dataset for further use.

Figure 4-Broadband speed cleaning code

```r
# Load necessary libraries
library(tidyverse)
library(dplyr)


# Set the working directory
setwd("C:/Users/aasis/Desktop/DataScience-Assignment")
getwd()
#Importing cleaned postcode to LSOA csv into R
cleaned_postcode_to_LSOA<- read_csv("Clean-data/Cleaned Postcode To LSOA Code.csv")

#Cleaning and joining data through the use of pipe operator
broadband_speed<-read_csv("Obtained-data/broadband-data/201805_fixed_pc_performance_r03.csv") %>% #Importing broadband speed csv into R
  as_tibble() %>% #converting into tibble
  select('Average download speed (Mbit/s)', postcode_space) %>%  #only selecting columns that are required
  rename(Postcode= `postcode_space`) %>% #renaming the post_space column to Postcode
  right_join(cleaned_postcode_to_LSOA, by="Postcode") %>% #Joining with the cleaned house price dataset by matching Postcode
  select('Average download speed (Mbit/s)',Postcode, `Short Postcode`, `Town/City`, District, County,) %>% #selecting only required columns
  na.omit() %>%  #Removing rows with null value
  mutate(`Short Postcode`= substr(Postcode, 1,5)) %>% #Filling missing short code values
  mutate(S_No = row_number()) %>% #Adding a new serial number column
  select(S_No, everything()) #moving the serial number column at first


#defining path to save the cleaned dataset
file_path <- "Clean-data/Cleaned Broadband Speed Dataset.csv"


#saving the cleaned dataset
write.csv(broadband_speed,file_path, row.names = FALSE)
```

Crime rate Cleaning data

This combines crime data from Kent and Surrey, loads the datasets using the tidyverse and dplyr libraries, merges them into a single dataset (merged data), and displays the result. Subsequently, it slices values in the 'LSOA name' and 'Falls within' columns, removing excess characters. The 'Context' column is then deleted to remove any null values in this column. Finally, the cleaned dataset is saved as a CSV file named "Cleaned Crime Data.csv" at the specified path.

Figure 5-Crime data cleaning code

```r
library(tidyverse)
library(dplyr)

#load the datasets
kentcrimedata = read_csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/police-data/2023-04/2023-04-kent-street.csv")
surreycrimedata = read_csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/police-data/2023-04/2023-04-surrey-street.csv")

#Merge the data
merged_data = rbind(kentcrimedata,surreycrimedata)
merged_data

#Slicing the values in the columns LSOA and Falls
merged_data$`LSOA name` = substr(merged_data$`LSOA name`,1,nchar(merged_data$`LSOA name`)-5)
merged_data$`Falls within`= substr(merged_data$`Falls within`,1,nchar(merged_data$`Falls within`)-nchar("Police "))

#Deleting all Null Column
merged_data$Context = NULL

merged_data
write_csv(merged_data, "C:/Users/aasis/Desktop/DataScience-Assignment/Clean-data/Cleaned Crime Data.csv")
```

School Cleaning data

This R code processes school data from Kent and Surrey. It reads CSV files, each representing a specific region and academic year, loads the necessary libraries (tidyverse, dplyr), and manipulates the data. The code combines the datasets into a single tibble (**combine_data**) using **rbind**. Then, it cleans the data by handling empty strings, removing rows with any missing values, filtering out rows with "NE" or "SUPP" in the 'ATT8SCR' column, converting 'ATT8SCR' to numeric, and selecting specific columns. Finally, the cleaned school data is saved as a CSV file named "Cleaned School Data.csv" in the specified directory.

Figure 6- School data cleaning code

```r
library(tidyverse)
library(dplyr)

setwd("C:/Users/aasis/Desktop/DataScience-Assignment")

getwd()

kent2021_22= read.csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/school-data/2021-2022 kent/886_ks4provisional.csv",fill = TRUE) %>%
  mutate(Year = 2021) %>%
  select(Year, PCODE, SCHNAME, ATT8SCR) %>%
  distinct()
kent2022_23= read.csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/school-data/2022-2023 kent/886_ks4provisional.csv",fill=TRUE) %>%
  mutate(Year = 2022) %>%
  select( Year,PCODE,SCHNAME, ATT8SCR,) %>%
  na.omit() %>%
  distinct()
surrey2021_22= read.csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/school-data/2021-2022 surrey/936_ks4provisional.csv",fill=TRUE) %>%
  mutate(Year = 2021) %>%
  select( Year,PCODE,SCHNAME, ATT8SCR,) %>%
  na.omit() %>%
  distinct()
surrey2022_23= read.csv("C:/Users/aasis/Desktop/DataScience-Assignment/Obtained-data/school-data/2022-2023 surrey/936_ks4provisional.csv",fill=TRUE) %>%
  mutate(Year = 2022) %>%
  select( Year,PCODE,SCHNAME, ATT8SCR,) %>%
  na.omit() %>%
  distinct()

combine_data = rbind(kent2021_22,kent2022_23,surrey2021_22,surrey2022_23)

cleanSchooldata=combine_data %>%
  mutate_all(~ifelse(. == "", NA, .)) %>%  # Replace empty strings with NA
  filter_all(all_vars(!is.na(.))) %>%    # Remove rows with any NA values

# Remove rows where ATT8SCR contains "NE" or "SUPP"
filter(!grepl("NE|SUPP", ATT8SCR, ignore.case = TRUE)) %>%
# Convert ATT8SCR to numeric (assuming it's a numeric score column)
mutate(ATT8SCR = as.numeric(ATT8SCR)) %>%
filter(!is.na(ATT8SCR)) %>%
select(Year, PCODE, SCHNAME, ATT8SCR) %>%
distinct()


write.csv(cleanSchooldata,"Clean-data/Cleaned School Data.csv",row.names =FALSE)
```

Exploratory data analysis

Exploratory Data Analysis (EDA) is telling a story using pictures and graphs. It's about digging into data to find interesting insights without jumping to conclusions. EDA makes sure that the questions we ask about the data make sense and that the answers match what we already know.

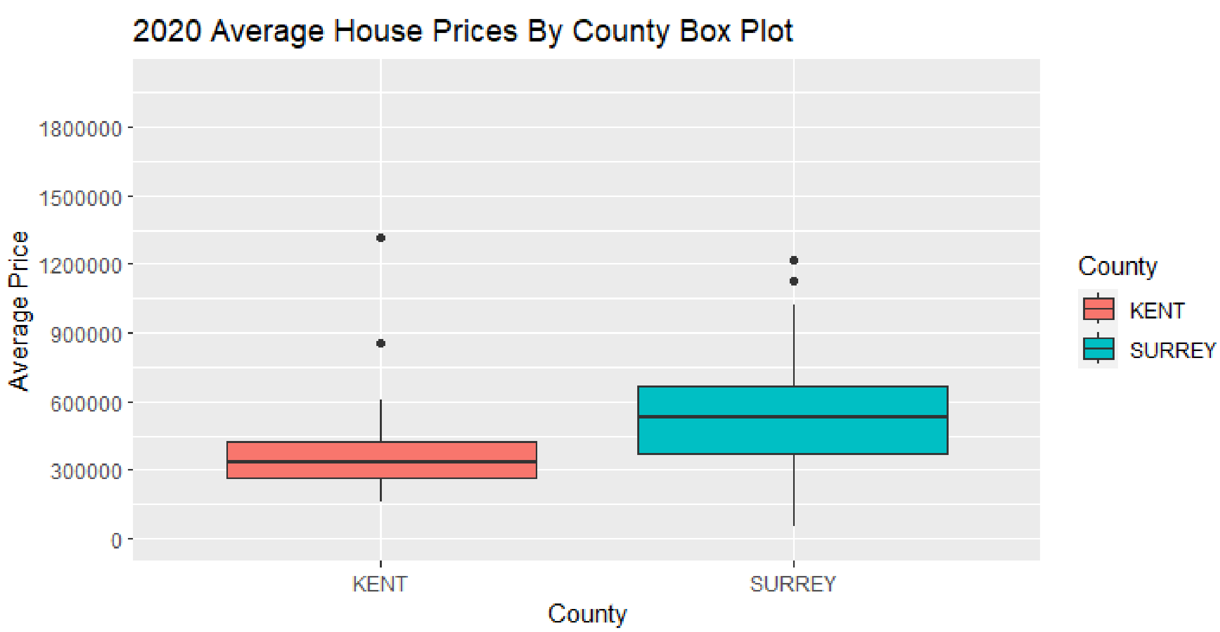
In the beginning, we decide whether to use visuals or other methods to explore the data. Then, we figure out if we want to focus on one thing or look at multiple aspects together. EDA acts like a spotlight, helping us spot unusual patterns in the data and giving us guidance on how to analyze it effectively. We start by looking at individual parts, then explore relationships between two things, and finally, we check how different factors interact with each other (Patil, What is exploratory data analysis? 2022).

While sometimes we use tables with numbers like averages, most people prefer pictures for better understanding. Different types of visuals and tools help us track and understand the data. If one way isn't clear, we try another to get a fuller picture of what the data is telling us. EDA lets us uncover hidden insights in the data, making it an exciting journey of discovery.

House price data representation

A box plot to visually compare the average house prices in Kent and Surrey for the year 2020. It starts by grouping cleaned house price data by town, district, county, and date of transfer, calculating the average price for each group. The code then filters this data to include only records from 2020. Using the ggplot library, it creates a box plot, with the x-axis representing counties (Kent and Surrey), the y-axis representing average prices, and the plot filled by county for clarity. The chart is titled "2020 Average House Prices By County Box Plot" and is customized to show prices up to 2,000,000 with breaks at intervals of 300,000 on the y-axis. The resulting visualization provides insights into the distribution of average house prices in the specified counties for the specified year.

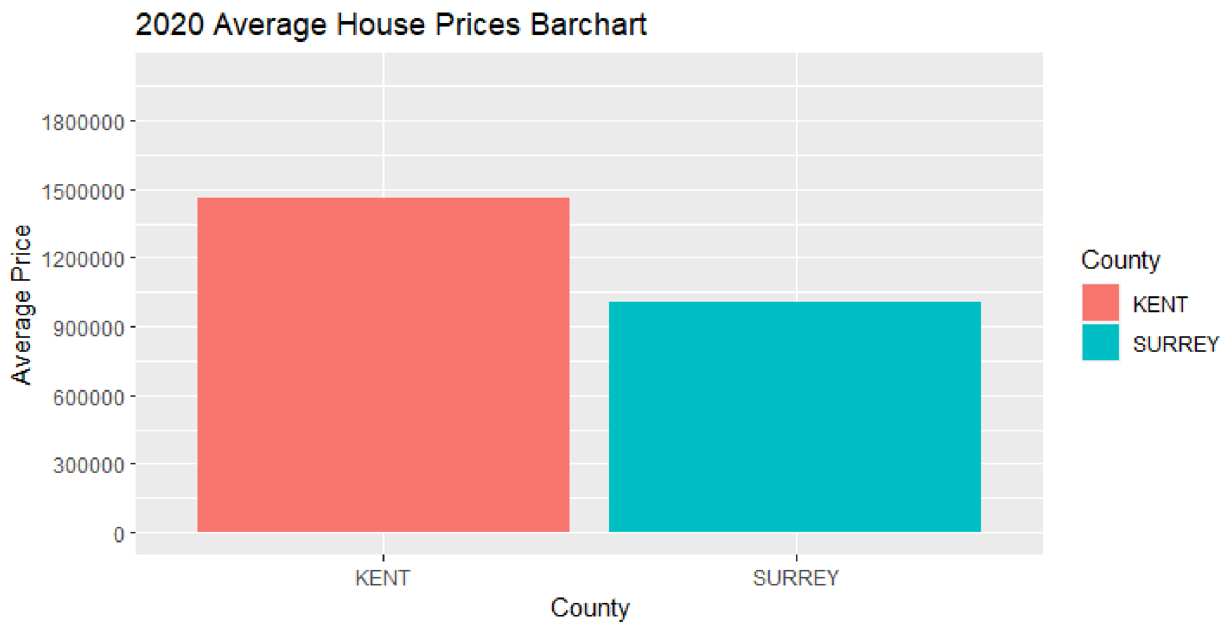
Figure 7-Average house prices by county



A bar chart to visually compare the average house prices in Kent and Surrey for the year 2020. It filters the grouped house price data to include only records from 2020, then uses ggplot to plot a bar chart with counties on the x-axis, average prices on the y-axis, and bars filled by county. The chart is titled

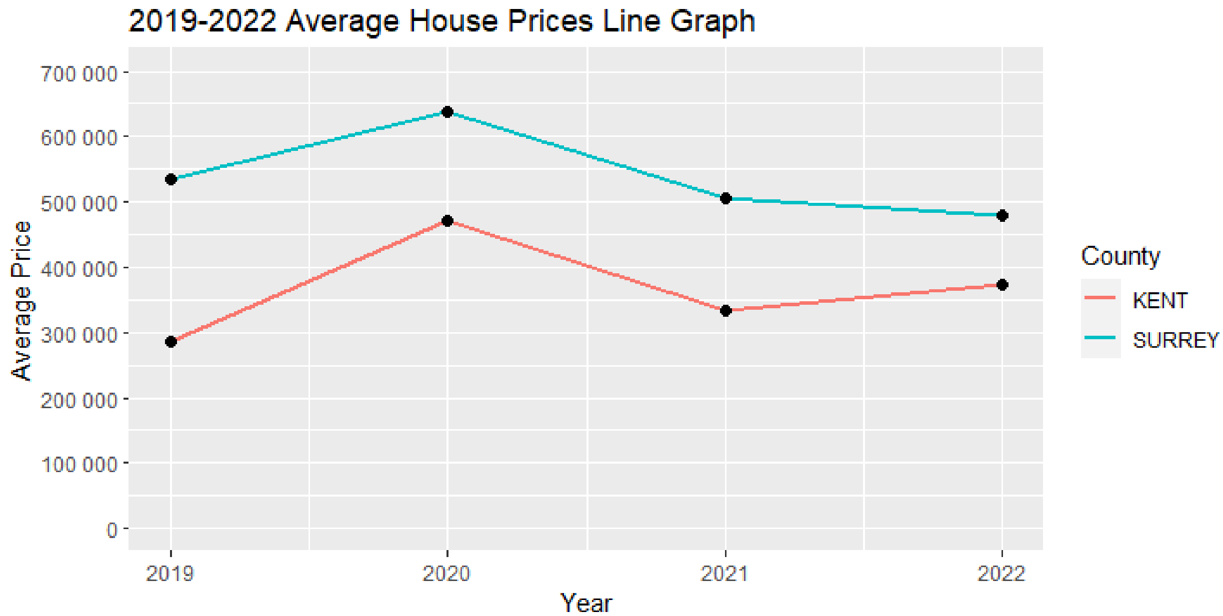
"2020 Average House Prices Barchart" and is customized to display prices up to 2,000,000 with breaks at intervals of 300,000 on the y-axis. The resulting visualization provides a clear comparison of average house prices between the two specified counties for the specified year.

Figure 8-Average house prices barchart



This graph groups cleaned house prices by county and year, calculating the average price for each group. It then creates a line graph to visualize average house prices from 2019 to 2022. The data is filtered to include only records from these years, and the graph compares county prices over this period. The resulting line graph is titled "2019-2022 Average House Prices Line Graph," with the x-axis representing years, the y-axis representing average prices, and different colors indicating different counties. Points on the lines highlight specific data points, and the graph is customized with specific limits, breaks, and labels for better clarity.

Figure 9-Average house prices line graph



Broadband Speed data representation

Ggplot was used to plot various graphs of broadband speed of both kent in surrey. At first average download speed of both county was compared in a box plot where surrey had more speed. Following that average download speed within kent was put in a bar chart. Also pie chart of robbery crimes committed is also plotted.

Figure 10-Average Download speed by COunty

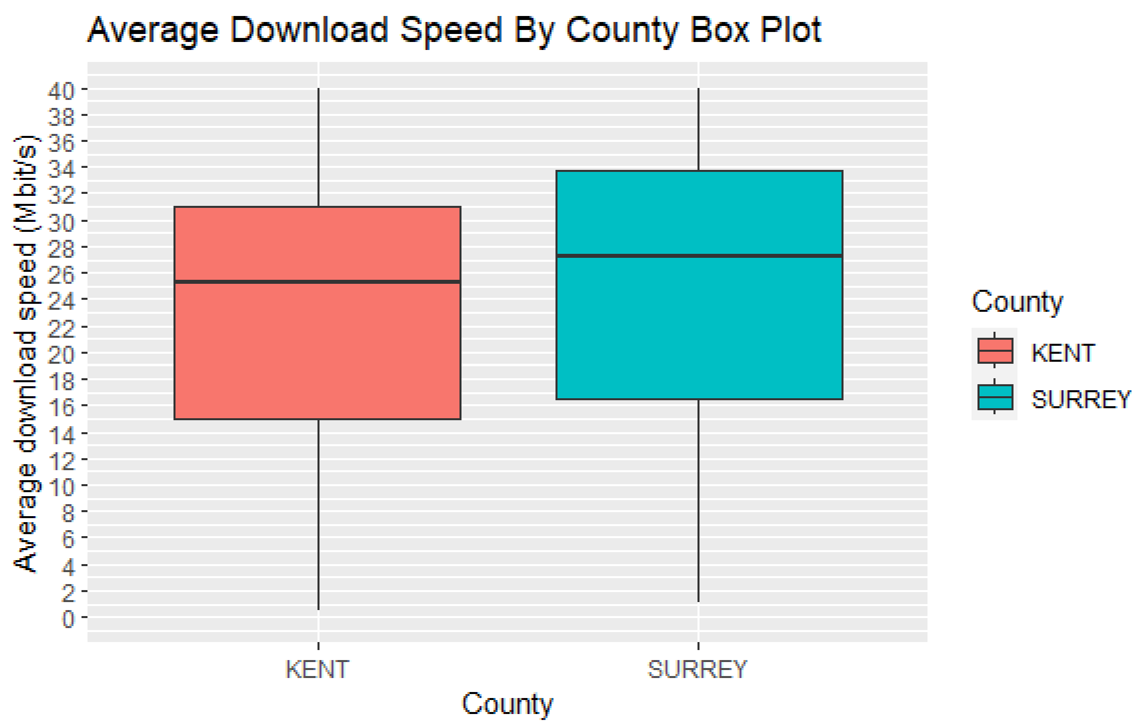


Figure 11-Average download speed within Kent

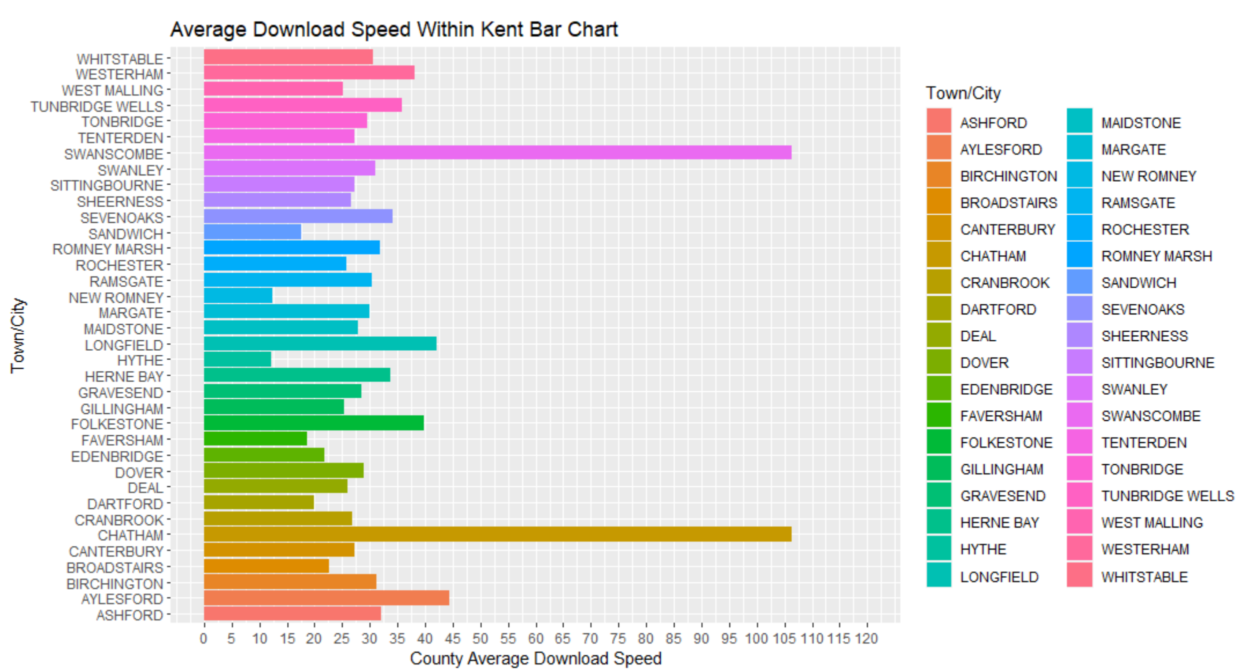
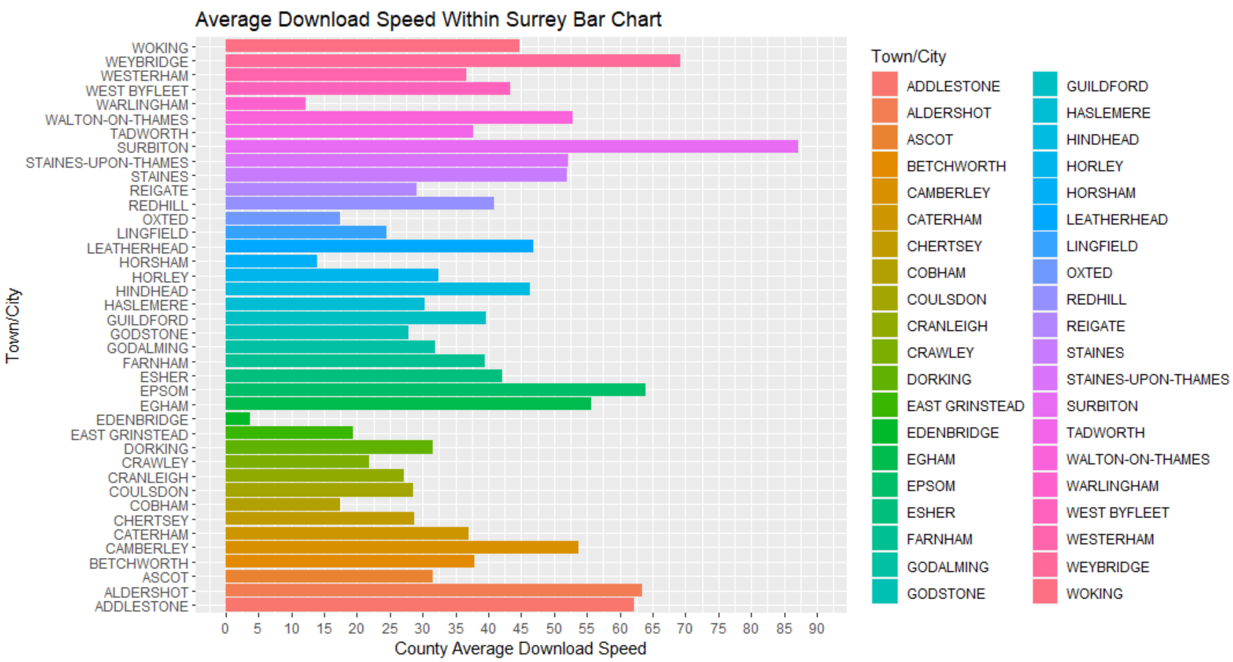


Figure 12-Average Download Speed within surrey



- Crime rate data representation

Ggplot and fmsb library was used to plot various graphs of crime rate of both Kent and surrey. At first Drug offence rate by district is plotted in a box plot. Following that a radar chart of vehicle offence rate per 10k is plotted. Also average download speed of surrey was visualized in a bar chart along with appropriate legend and colors. At last line chart of drug offence rate of both kent and surrey is plotted.

Figure 13- Drug offence rate by district

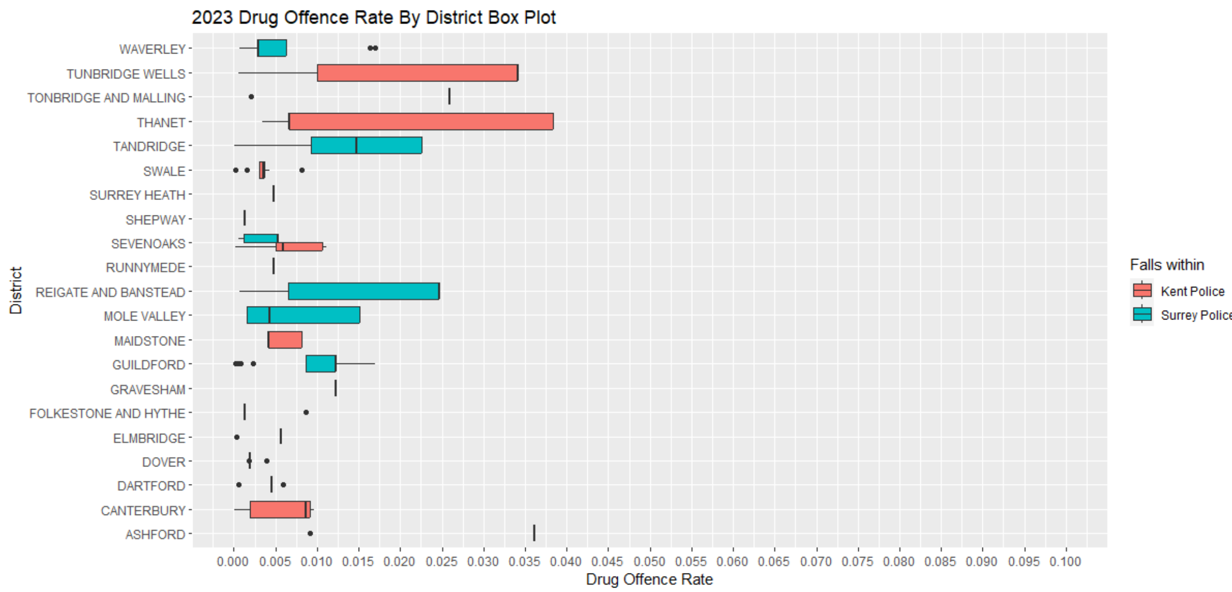


Figure 14-Radar chart

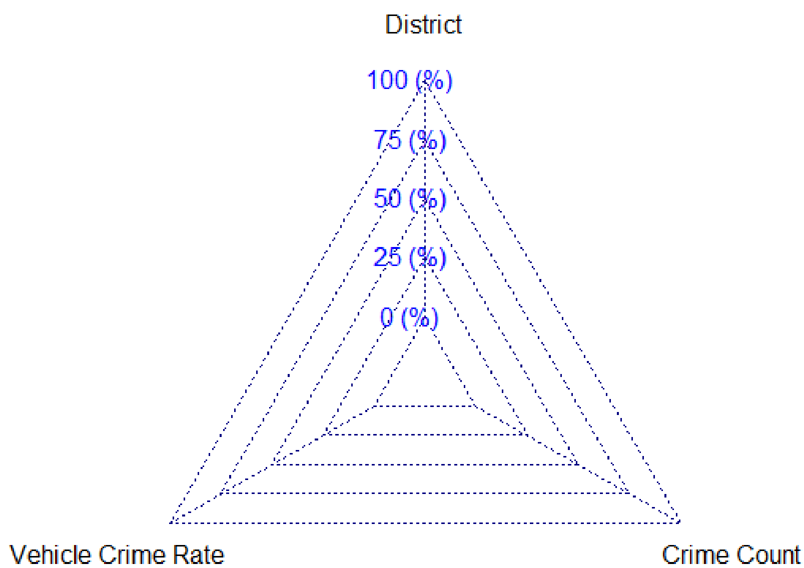


Figure 15-Robbery rate in march 2022

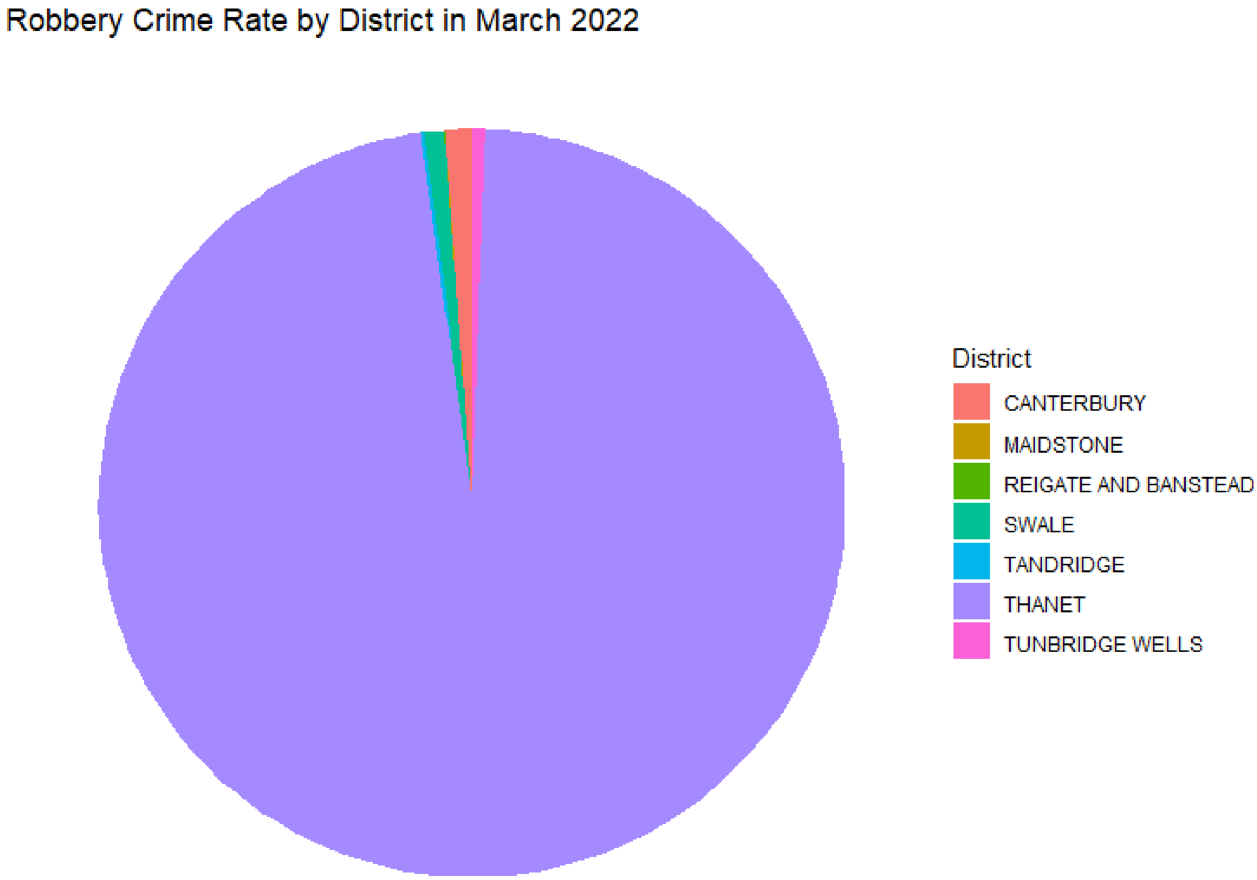
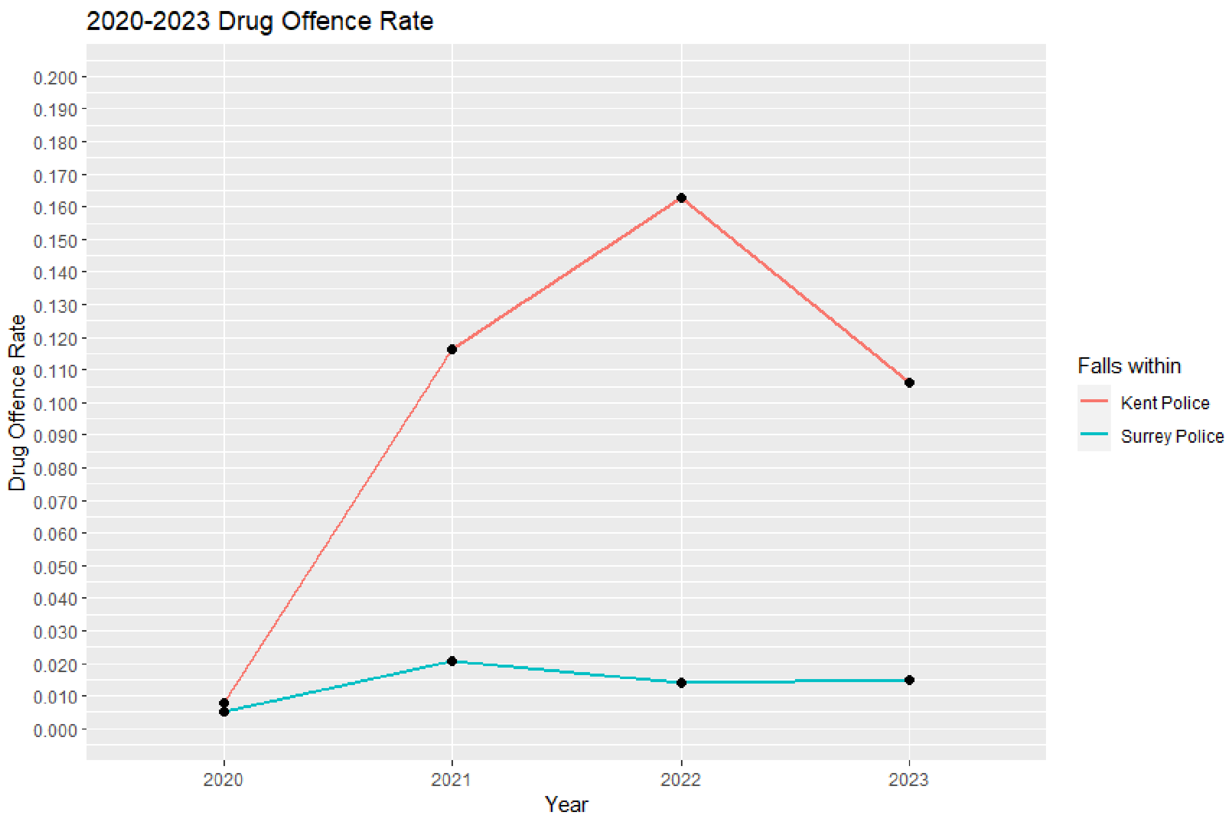


Figure 16 Drug offence rate



School data representation

In this section average attainment score is compared between both kent and surrey in a box plot. Then average attainment score line graph is drawn of kent county following that it is also drawn for surrey county.

Figure 17-Average attainment score both county

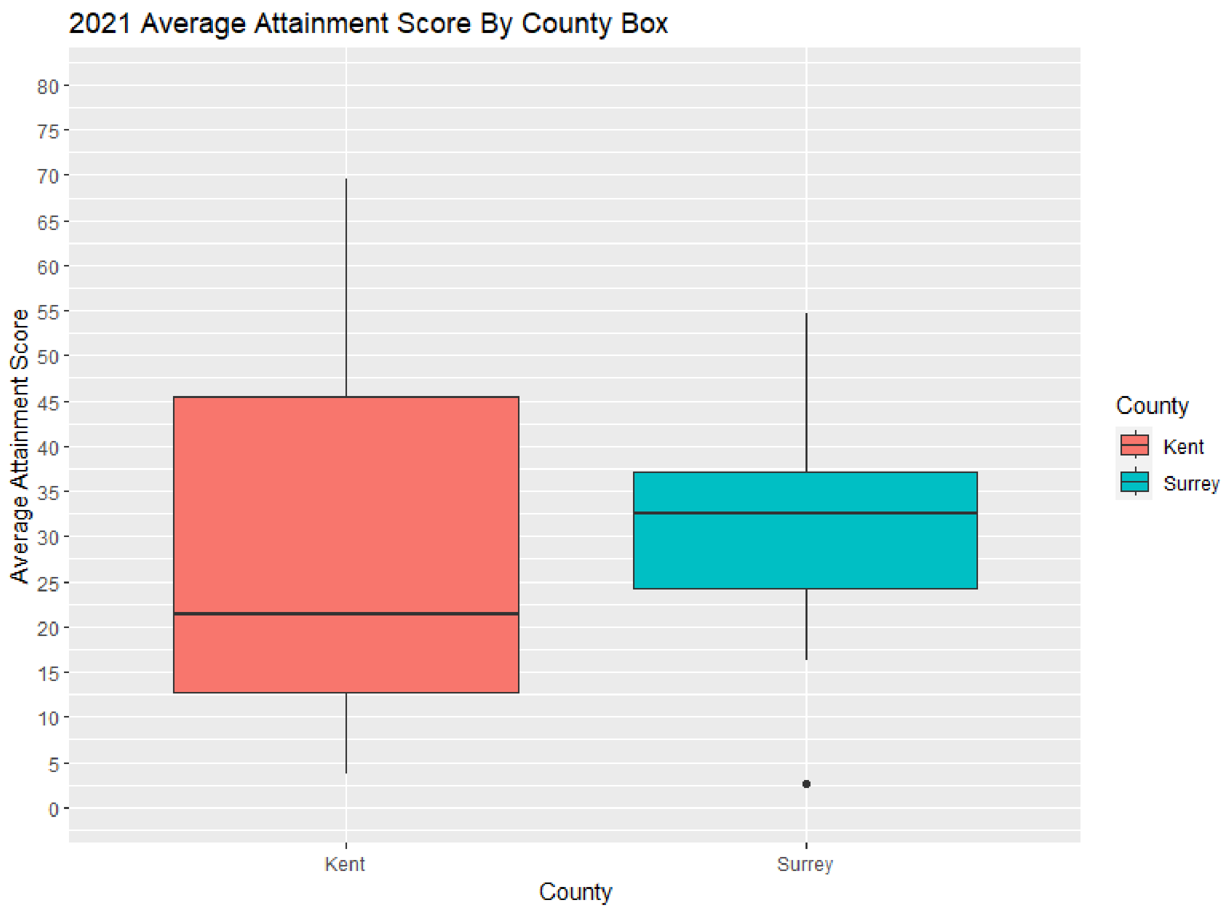


Figure 18-Average attainment score kent line graph

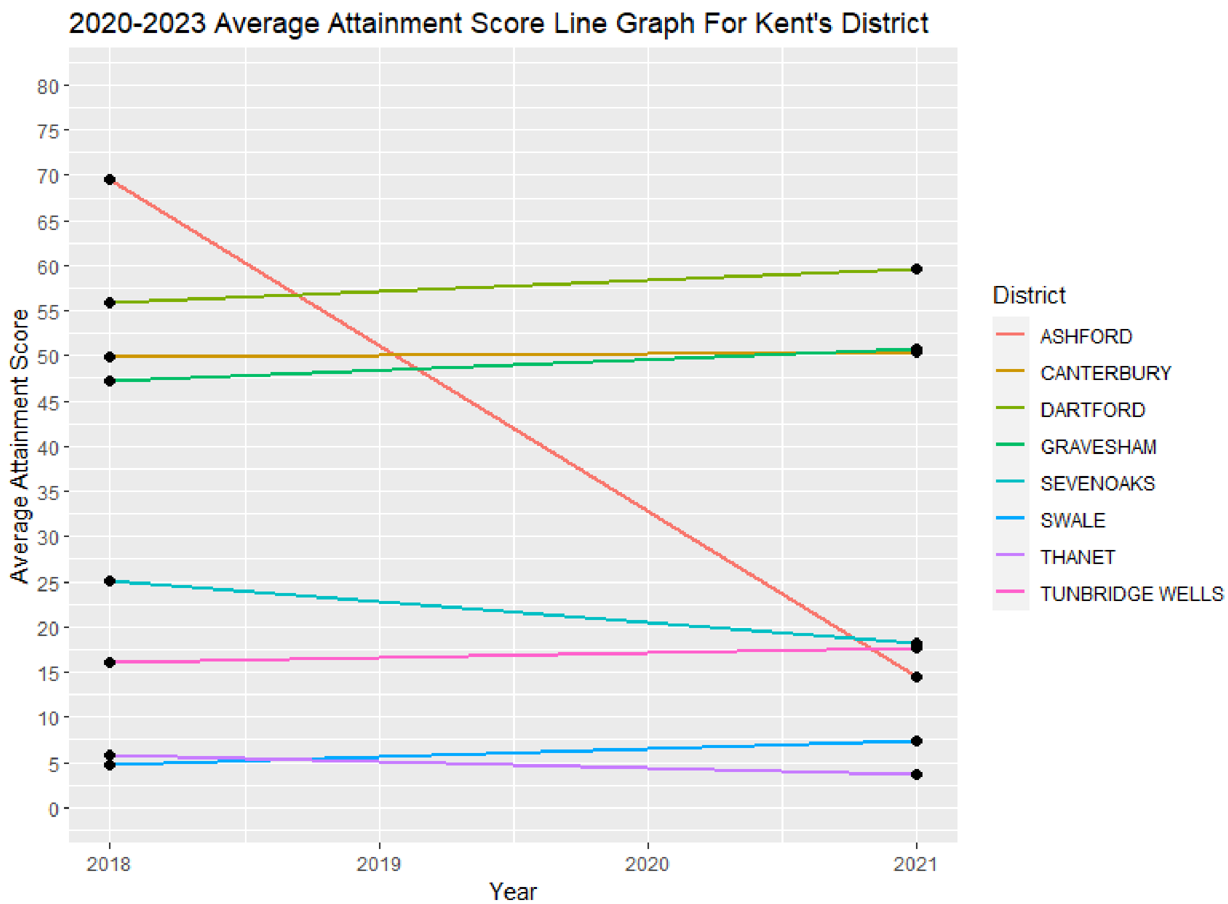
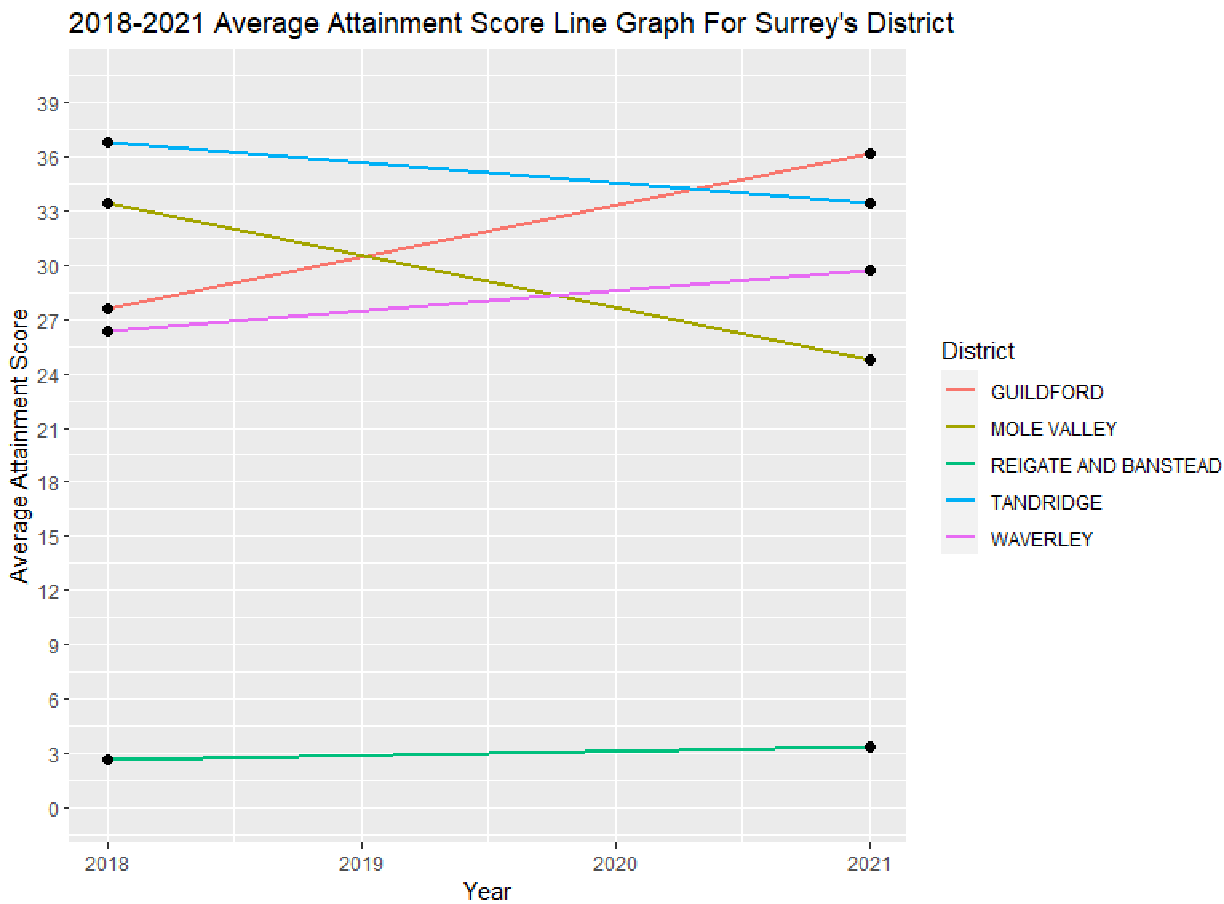


Figure 19-Average attainment score surrey line graph



Linear modeling

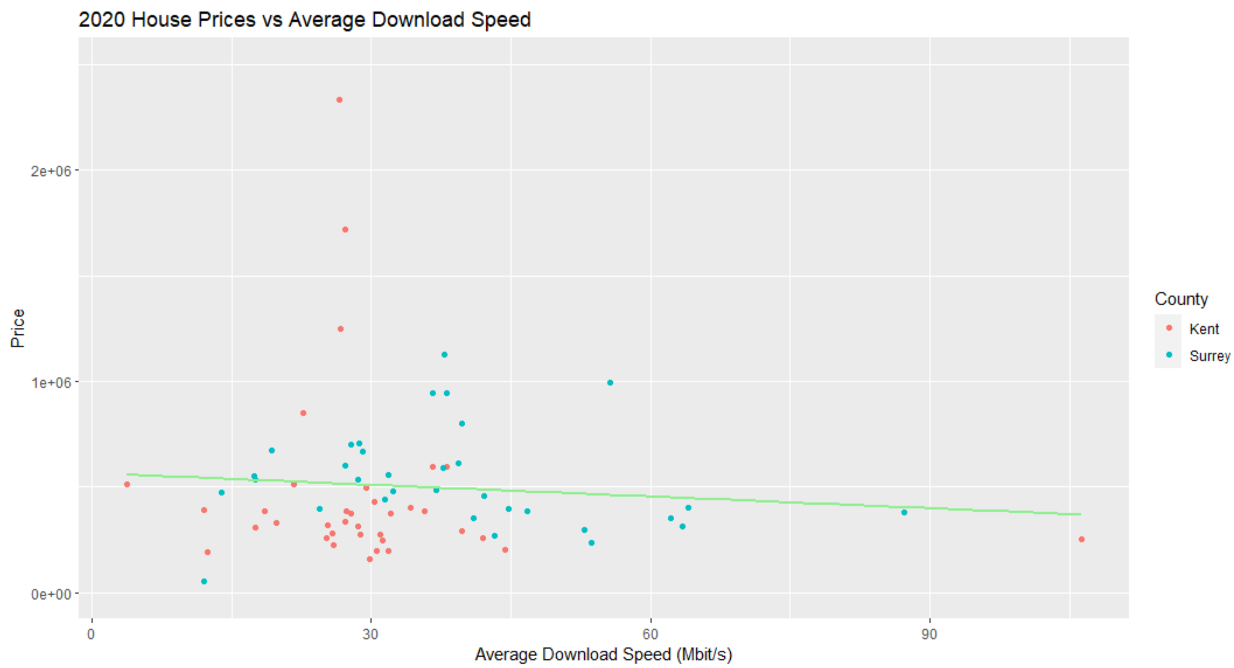
Leveraging the principles of linear regression, a statistical technique employed to identify connections between variables, streamlines the analysis of diverse datasets without repetitive efforts. In this approach, data points are linked by a straight line, and the objective is to determine regression coefficients that minimize errors, ensuring the best possible fit. Linear regression comes in two main forms: simple linear regression, dealing with a single variable, and multiple linear regression, which involves several independent variables and is more intricate (Mishra, Linear Modeling 2021).

The analysis encompasses a broad spectrum of information, including housing prices, crime rates (such as drug-related, robbery, and vehicle-related crimes), and significant educational metrics. By amalgamating these diverse datasets, a comprehensive comparison is facilitated, aiding in the identification of countries that have performed exceptionally well.

House price vs Average Download Speed

This begins by importing cleaned house prices and broadband speed datasets. It then groups the house prices and broadband speed data by town and county, calculating the average price and download speed for each group. The two datasets are joined into a single table. The code proceeds to create a linear model predicting house prices based on average download speed. The summary of the linear model is displayed, providing insights into the relationship between house prices and broadband speed. Additionally, a graphical representation of the linear model is generated using ggplot, with data points colored differently for Kent and Surrey. The resulting plot visualizes the 2020 house prices against average download speed, including a linear regression line.

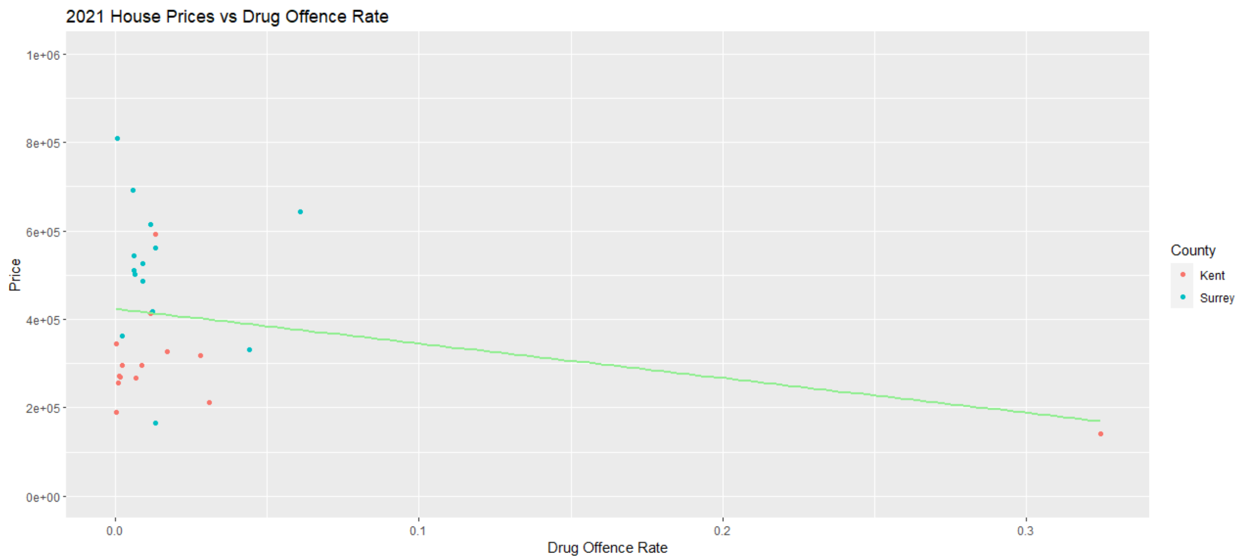
Figure 20-House price vs Average download speed



House price vs Drug rate

This R code begins by grouping cleaned house prices for the year 2021 by town and county, calculating the average price for each group. It then modifies the crime dataset to focus on drug-related offenses, creating a new dataset showing drug offense rates for each town and county in 2021. The two datasets are joined into a single table, and a linear model is created to predict house prices based on drug offense rates. The summary of the linear model is displayed, offering insights into the relationship between house prices and drug offenses. A graphical representation of the linear model is generated using ggplot, with data points colored differently for Kent and Surrey. The resulting plot visualizes 2021 house prices against drug offense rates, including a linear regression line. Finally, null values are removed from the joined dataset.

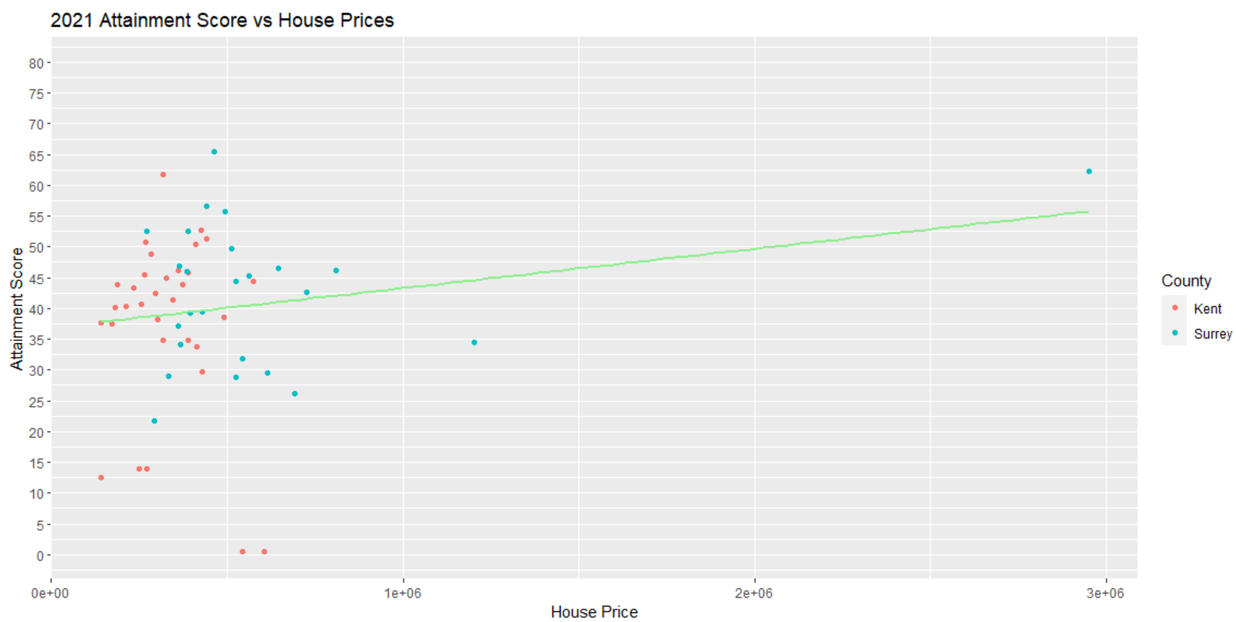
Figure 21-House price vs Drug offence rate



Attainment 8 Score vs House price

First group cleaned house prices for the year 2021 by town and county, calculating the average price for each group. It then groups school data by town and county for the same year, calculating the average attainment score for each group. The two datasets are joined into a single table based on the town, converting town names to lowercase for consistency. A linear model is created to predict average attainment scores based on average house prices. The summary of the linear model is displayed, providing insights into the relationship between attainment scores and house prices. A graphical representation of the linear model is generated using ggplot, with data points colored differently for Kent and Surrey. The resulting plot visualizes the 2021 attainment scores against house prices, including a linear regression line. Null values are removed from the joined dataset.

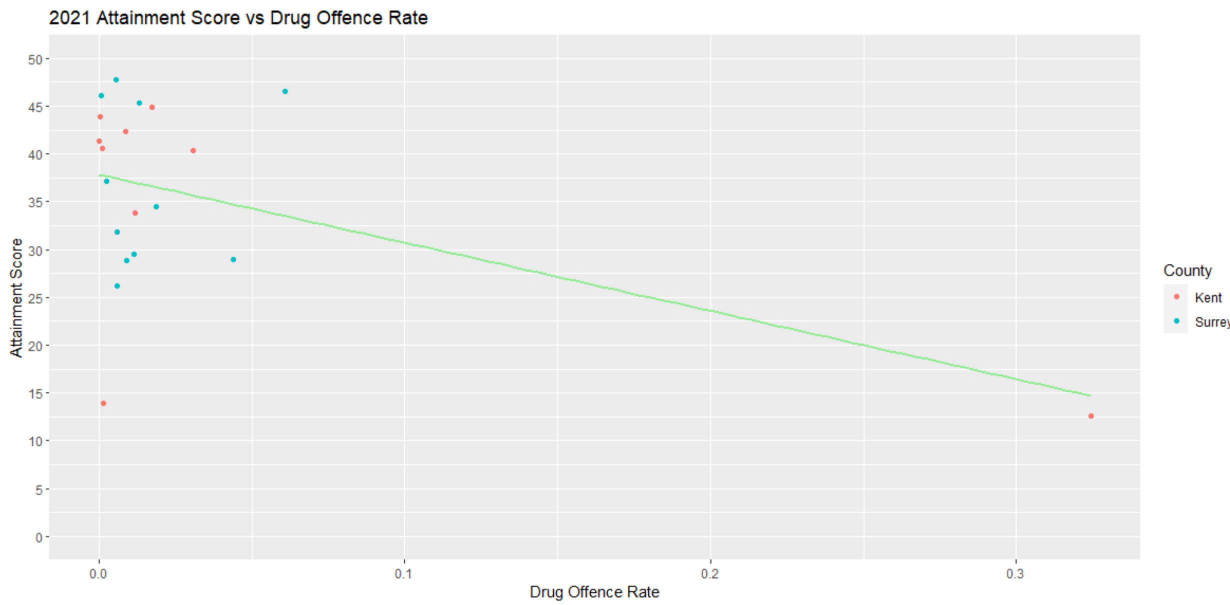
Figure 22-Attainment score vs House prices



Attainment 8 score vs drug rate

This begins by grouping school data for the year 2021 by town and county, calculating the average attainment score for each group. It then modifies the crime dataset to focus on drug-related offenses, creating a new dataset showing drug offense rates for each town and county in 2021. The two datasets are joined into a single table based on the town, converting town names to lowercase for consistency. A linear model is created to predict average attainment scores based on drug offense rates. The summary of the linear model is displayed, providing insights into the relationship between attainment scores and drug offenses. A graphical representation of the linear model is generated using ggplot, with data points colored differently for Kent and Surrey. The resulting plot visualizes the 2021 attainment scores against drug offense rates, including a linear regression line.

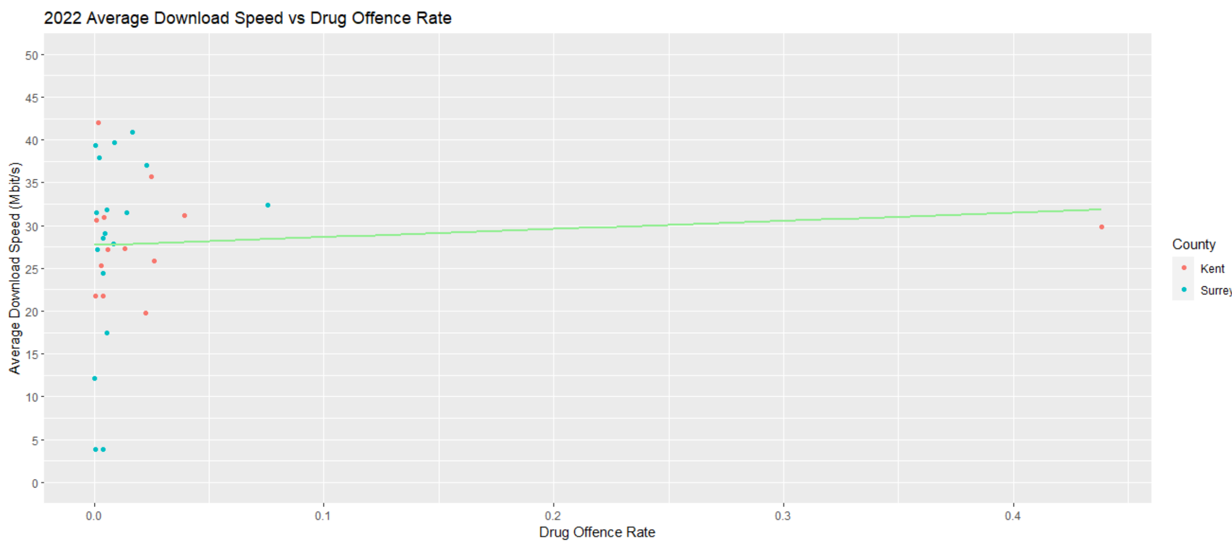
Figure 23-Attainment Score vs Drug offence rate



Average Download Speed vs Drug rate

This begins by grouping broadband speed data by town and county, calculating the average download speed for each group. It then modifies the crime dataset to focus on drug-related offenses, creating a new dataset showing drug offense rates for each town and county in 2022. The two datasets are joined into a single table based on the town. A linear model is created to predict average download speeds based on drug offense rates. The summary of the linear model is displayed, providing insights into the relationship between download speeds and drug offenses. A graphical representation of the linear model is generated using ggplot, with data points colored differently for Kent and Surrey. The resulting plot visualizes the 2022 average download speeds against drug offense rates, including a linear regression line.

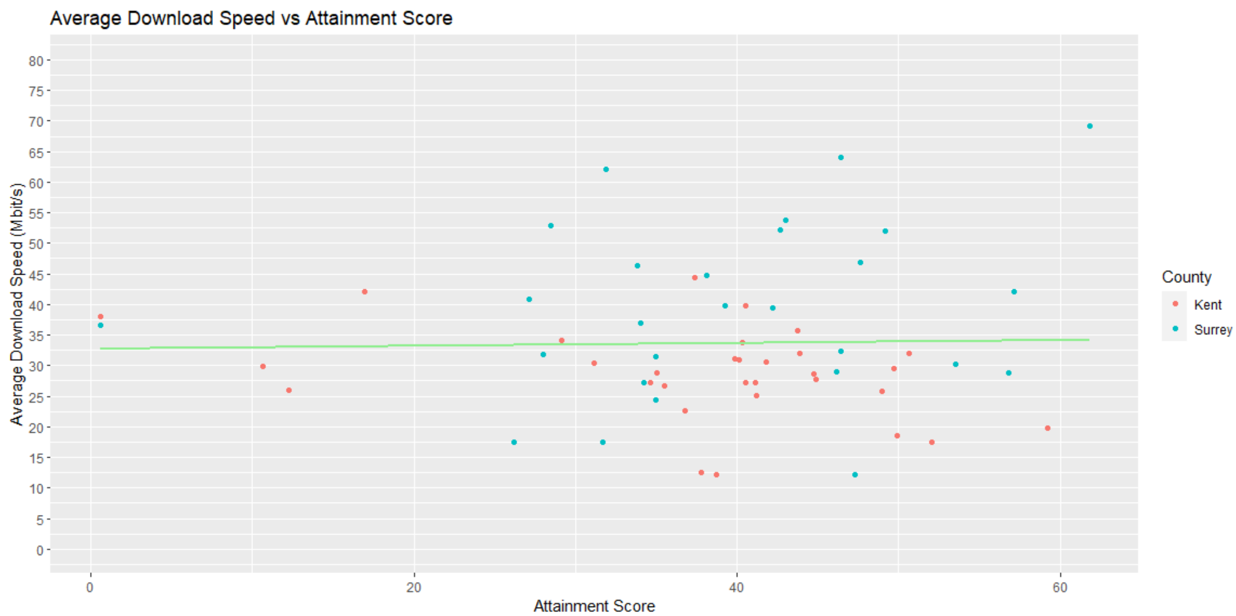
Figure 24-Average download speed vs Drug offence Rate



Average Download Speed vs Attainment Score

This begins by grouping broadband speed data by town and county, converting town names to lowercase for consistency, and calculating the average download speed for each group. It then groups school data by town and county, converting town names to lowercase, and finding the average attainment score for each group. The two datasets are joined into a single table based on the town. A linear model is created to predict average download speeds based on attainment scores. The summary of the linear model is displayed, providing insights into the relationship between download speeds and attainment scores. A graphical representation of the linear model is generated using ggplot, with data points colored differently for Kent and Surrey. The resulting plot visualizes the average download speeds against attainment scores, including a linear regression line.

Figure 25-Average Download speed vs Attainment Score



Recommendation System

House ranking

Margate came in first place from kent county with an average price of 125226. Also higher score is given to more affordable house.

Figure 26-House ranking

```r
#----------House Price Ranking----------#

# Clean the Price column (convert to numeric)
house_prices_data$Price <- as.numeric(gsub("[^0-9.]", "", house_prices_data$Price))

# Group by County and Town/City, calculate the average price for each group
grouped_data <- house_prices_data %>%
  group_by(County, `Town.City`) %>%
  summarise(`Average Price` = mean(Price))

# Assign a score to housing prices (higher score indicates affordability)
grouped_data <- grouped_data %>%
  mutate(Score = 1 - scale(`Average Price`))

# Arrange the data in descending order based on the score
sorted_data <- grouped_data[order(-grouped_data$Score), ]

# Select the top 10 entries
top_10 <- head(sorted_data, 10)

# Print the top 10 best counties and town/cities with the least house prices and their scores
print(top_10[, c("County", "Town.City", "Average Price", "Score")])

```

```
29:1    (Top Level) :

R 4.3.1 · C:/Users/aasis/Desktop/DataScience-Assignment/
# A tibble: 10 × 4
# Groups:   County [2]
   County Town.City     `Average Price` Score[,1]
   <chr>  <chr>                   <dbl>     <dbl>
 1 KENT   MARGATE               145226.      2.93
 2 KENT   SHEERNESS             163832.      2.71
 3 KENT   NEW ROMNEY            168584.      2.66
 4 KENT   WHITSTABLE            177834.      2.55
 5 KENT   GRAVESEND             220276.      2.06
 6 KENT   DEAL                  223601.      2.02
 7 KENT   BIRCHINGTON           229328.      1.95
 8 KENT   ROMNEY MARSH          233316.      1.90
 9 SURREY EDENBRIDGE            165000       1.83
10 KENT   WEST MALLING          247423.      1.74
```

Broadband Ranking

The town of Chatham from kent county came in first with the average download speed of 106 .

Figure 27-Broadband speed ranking

```r
#-----------------------------------------------------------------------------------------------------------
broadband_speed_data <- read.csv("C:/Users/aasis/Desktop/DataScience-Assignment/Clean-data/Cleaned Broadband Speed Dat
broadband_speed_data
# Group by County and Town/City, calculate the average download speed for each group
grouped_data <- broadband_speed_data %>%
  group_by(County, `Town.City`) %>%
  summarise(`Average Download Speed` = mean(`Average.download.speed..Mbit.s.`))

# Assign a score to download speeds (higher score indicates better speed)
grouped_data <- grouped_data %>%
  mutate(Score = scale(`Average Download Speed`))

# Arrange the data in descending order based on the score
sorted_data <- grouped_data[order(-grouped_data$Score), ]

# Select the top 10 entries
top_10 <- head(sorted_data, 10)

# Print the top 10 towns/cities and counties with the highest average download speed and their scores
print(top_10[, c("County", "Town.City", "Average Download Speed", "Score")])
```

```
30:39   (Untitled)                                                    R Scri
R 4.3.1 · C:/Users/aasis/Desktop/DataScience-Assignment/
# A tibble: 10 × 4
# Groups:   County [2]
   County Town.City         `Average Download Speed` Score[,1]
   <chr>  <chr>                                <dbl>     <dbl>
 1 KENT   CHATHAM                               106.     3.79
 2 KENT   SWANSCOMBE                            106.     3.79
 3 SURREY SURBITON                               87.2    2.85
 4 SURREY WEYBRIDGE                              69.2    1.80
 5 SURREY EPSOM                                  64.0    1.50
 6 SURREY ALDERSHOT                              63.4    1.46
 7 SURREY ADDLESTONE                             62.1    1.39
 8 SURREY EGHAM                                  55.6    1.01
 9 SURREY CAMBERLEY                              53.7    0.894
10 SURREY WALTON-ON-THAMES                       52.9    0.850
```

Crime ranking

Orpington town in surrey county came in 1st place with the least amount of crimes count.

```r
#------------------------------------------------------------------------------------------------------------------------
crime_data <- read.csv("C:/Users/aasis/Desktop/DataScience-Assignment/Clean-data/Cleaned Crime Dataset.csv")

grouped_data <- crime_data %>%
  group_by(Falls.within, `Town.City`) %>%
  summarise(`Total Crime Count` = n())

# Assign a score to total crime counts (higher score indicates lower crime)
grouped_data <- grouped_data %>%
  mutate(Score = rank(`Total Crime Count`))

# Arrange the data in ascending order based on the score
sorted_data <- grouped_data[order(grouped_data$Score), ]

# Select the top 10 entries
top_10 <- head(sorted_data, 10)

# Print the top 10 towns/cities and counties with the lowest total crime count and their scores
print(top_10[, c("Falls.within", "Town.City", "Total Crime Count", "Score")])
```

70:1 (Untitled) R Script

R 4.3.1 · C:/Users/aasis/Desktop/DataScience-Assignment/

```
# A tibble: 10 x 4
# Groups:   Falls.within [2]
   Falls.within  Town.City   `Total Crime Count` Score
   <chr>         <chr>                     <int> <dbl>
 1 Kent Police   CATERHAM                      2     1
 2 Surrey Police ORPINGTON                     1     1
 3 Kent Police   OXTED                         7     2
 4 Surrey Police LONGFIELD                     3     2
 5 Kent Police   CRANLEIGH                    19     3
 6 Surrey Police DARTFORD                      7     3
 7 Kent Police   LINGFIELD                    21     4
 8 Surrey Police SWANLEY                      22     4
 9 Kent Police   HERNE BAY                   293     5
10 Surrey Police EDENBRIDGE                  344     5
>
```

School ranking

Dartford town from kent came in 1st place with highest average attainment score.

```r
# Read the CSV file
school_data <- read.csv("C:/Users/aasis/Desktop/DataScience-Assignment/Clean-data/Cleaned School Dataset.csv")
school_data
# Group by County and Town/City, calculate the average attainment score for each group
grouped_data <- school_data %>%
  group_by(County, Town) %>%
  summarise(`Average Attainment Score` = mean(`Attainment.Score`))

# Assign a score to average attainment scores (higher score indicates higher attainment)
grouped_data <- grouped_data %>%
  mutate(Score = rank(`Average Attainment Score`, na.last = "keep"))

# Arrange the data in descending order based on the score
sorted_data <- grouped_data[order(-grouped_data$Score), ]

# Select the top 10 entries
top_10 <- head(sorted_data, 10)

# Print the top 10 towns/cities and counties with the highest average attainment scores and their scores
print(top_10[, c("County", "Town", "Average Attainment Score", "Score")])
```

92:1 (Untitled)

R 4.3.1 · C:/Users/aasis/Desktop/DataScience-Assignment/

```
# A tibble: 10 × 4
# Groups:   County [2]
   County Town       `Average Attainment Score` Score
   <chr>  <chr>                           <dbl> <dbl>
 1 Kent   Dartford                         59.2    39
 2 Kent   Sandwich                         52.0    38
 3 Kent   Faversham                        49.9    37
 4 Kent   Tonbridge                        49.7    36
 5 Kent   Wye                              49.2    35
 6 Kent   Rochester                        49      34
 7 Kent   Maidstone                        44.9    33
 8 Surrey Weybridge                        61.8    33
 9 Kent   Gravesend                        44.7    32
10 Surrey Esher                            57.1    32
>
```

Overall ranking

```
[1] "Recommended City:"
> print(recommended_city)
    Town/City County HouseScore Attainment.Score SchoolScore AvgDownloadSpeed BroadbandScore TotalCrimes CrimeScore TotalScore
133 WARLINGHAM SURREY      9.45                0           0             12.1       9.886171        1221   9.997334   29.33351
```

Legal and ethical issues

In the modern era, the utilization of data has resulted in a plethora of legal, ethical, and social dilemmas. One of the most crucial concerns is the protection of data. The collection, storage, and sharing of personal information can lead to infringement of individual rights. To safeguard privacy, it is imperative to implement measures against unauthorized access and misuse.

Ethical considerations are also of utmost importance. Striking a balance between utilizing data for gaining insights and respecting individuals' autonomy is a challenging task. When data is used to alter people's behavior, it exacerbates the ethical dilemma. This brings to light issues of transparency and potential manipulation.

(*5 Principles of Data Ethics for Business*, 2021)

The aspect of safety is another crucial aspect in this scenario. Data breaches can result in theft of personal information, financial harm, and damage to reputation. Ensuring data security is not only a legal requirement, but also a moral obligation to prevent harm to individuals.

Conclusion

In brief, the study aimed to assist a friend in selecting a suitable location in the UK for relocation. Key factors such as housing costs, broadband speed, school quality, and crime rates were examined to gather crucial information. The data underwent meticulous cleaning and was visually presented through graphs and charts for clarity. A specialized model was employed to explore correlations among various factors, leading to the creation of a ranking based on the analysis results.

References

Elgabry, O. (2019, March 2). *The Ultimate Guide to data cleaning*. Medium. https://towardsdatascience.com/the-ultimate-guide-to-data-cleaning-3969843991d4

R tutorial: Learn R programming language tutorial - javatpoint. www.javatpoint.com.

Police Data. (2022). Police Data Download. https://data.police.uk/data/

Registry, H. L. (2023, July 28). *Price paid data*. GOV.UK. https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads

5 Principles of Data Ethics for Business. (2021, March 16). Business Insights Blog. https://online.hbs.edu/blog/post/data-ethics

Mishra, P. (2021, December 19). *Linear Modeling*. GeeksforGeeks. https://www.geeksforgeeks.org/how-to-use-lm-function-in-r-to-fit-linear-models/

Ofcom. (2023, March 16). *Connected nations 2018: Data downloads*. Ofcom. https://www.ofcom.org.uk/research-and-data/multi-sector-research/infrastructure-research/connected-nations-2018/data-downloads

Github Link

https://github.com/aasiskrk/Datasci-assign

Appendix

```
#importing the cleaned school dataset
cleaned_school_dataset= read_csv('Clean-data/Cleaned School Dataset.csv')

#grouping broadband speed by town and county and finding average download speed for each group
grouped_broadband_speeds = cleaned_broadband_speed %>%
  group_by(`Town/City`,County) %>%
  mutate(`Town/City`= tolower(`Town/City`)) %>%  #converting the town from to all lowercase
  summarise(`Average download speed (Mbit/s)`= mean(`Average download speed (Mbit/s)`))

#grouping school data by town and county and finding average score for each group
grouped_school_dataset = cleaned_school_dataset %>%
  group_by(`Town`,County) %>%
  mutate(Town= tolower(Town)) %>%  #converting the town from to all lowercase
  summarise(`Attainment Score`=mean(`Attainment Score`))


#joining broadband data and school data in a single table
broadband_attainment_data = grouped_broadband_speeds %>%
  left_join(grouped_school_dataset,by=c("Town/City"="Town")) %>%
  na.omit #removing rows with null value

#creating a linear model
l_model = lm(data=broadband_attainment_data, `Average download speed (Mbit/s)`~`Attainment Score`) #t

#showing summary of the Linear Model
summary(l_model)


#creating the linear model graph
ggplot(broadband_attainment_data,aes(x=`Attainment Score`,y=`Average download speed (Mbit/s)`)) +
  scale_y_continuous(limits=c(0,80), breaks = seq(0,80,5))+ #setting limits and breaks
  geom_point(data = filter(broadband_attainment_data,County.x=="KENT"),aes(color=c("Red"="Kent")))+ #
  geom_point(data = filter(broadband_attainment_data,County.x=="SURREY"), aes(color=c("Blue"="Surrey"
  geom_smooth(method=lm,se=FALSE,color="lightgreen")+ #adding linear regression line and omitting err
  labs(x="Attainment Score",
       y="Average Download Speed (Mbit/s)",
       title="Average Download Speed vs Attainment Score",color="County") #setting labels
```

```r

#modifying our crime dataset to show drug offence rate and crime count
crime_dataset_drugs2 <-cleaned_crime_dataset %>%
  mutate(`Date of crime`= substr(`Date of crime`, 1, 4)) %>% #Mutating this column to only show year
  group_by(`Short Postcode`,`Crime type`,`Date of crime`, `Falls within`) %>% #Grouping to show crime count in each postcode by year
  select(`Short Postcode`,`Crime type`,`Date of crime`, `Falls within`) %>%
  na.omit() %>%
  tally() %>% #creating crime count column
  rename(`Crime Count`=n) %>%  #renaming crime count column %>%
  right_join(population_dataset, by = "Short Postcode") %>% #joining with population dataset to show district and population
  select(`Short Postcode`,`Crime type`,`Crime Count`, `Population`, `Date of crime`, `Falls within`, `Town/City`, District) %>% #select the required co
  na.omit() %>%
  filter(`Crime type`== "Drugs") %>% #filtering to show only drug crimes of 2022
  mutate(`Drug Offence Rate` = (`Crime Count` / Population)) #calculating drug offence rate

#grouping the drug crime dataset by county and town and showing the rate for each group for the year 2022
grouped_drug_crime <- crime_dataset_drugs2 %>%
  filter(`Date of crime`=="2022") %>%
  group_by(`Falls within`,`Town/City`) %>%
  summarise(`Drug Offence Rate`= mean(`Drug Offence Rate`))


#joining broadband data and drug crime rate data in a single table
broadband_crime_data = grouped_broadband_speeds %>%
  left_join(grouped_drug_crime,by="Town/City") %>%
  na.omit #removing null values


#creating a linear model
l_model = lm(data=broadband_crime_data, `Average download speed (Mbit/s)`~`Drug Offence Rate`) #this model predicts Average download speed as a functio

#showing summary of the Linear Model
summary(l_model)


#creating the linear model graph
ggplot(broadband_crime_data,aes(x=`Drug Offence Rate`,y=`Average download speed (Mbit/s)`)) +
  scale_y_continuous(limits=c(0,50), breaks = seq(0,50,5))+ #setting limits and breaks
  geom_point(data = filter(broadband_crime_data,County=="KENT"),aes(color=c("Red"="Kent")))+ #setting color as red for Kent's data point
  geom_point(data = filter(broadband_crime_data,County=="SURREY"), aes(color=c("Blue"="Surrey"))) + #setting color as blue for Surrey's data point
  geom_smooth(method=lm,se=FALSE,color="lightgreen")+ #adding linear regression line and omitting error bands
  labs(x="Drug Offence Rate",
       y="Average Download Speed (Mbit/s)",
       title="2022 Average Download Speed vs Drug Offence Rate",color="County") #setting labels

```

```r
#grouping school data by town and county and finding average score for each group
grouped_school_dataset = cleaned_school_dataset %>%
  filter(`Year`=="2021") %>%
  group_by(`Town`,County) %>%
  mutate(Town= tolower(Town)) %>%  #converting the town from to all lowercase
  summarise(`Attainment Score`=mean(`Attainment Score`))

#modifying our crime dataset to show drug offence rate and crime count
crime_dataset_drugs2 <-cleaned_crime_dataset %>%
  mutate(`Date of crime`= substr(`Date of crime`, 1, 4)) %>% #Mutating this column to only show year
  group_by(`Short Postcode`,`Crime type`,`Date of crime`, `Falls within`) %>% #Grouping to show crime count in each postcode by year
  select(`Short Postcode`,`Crime type`,`Date of crime`, `Falls within`) %>%
  na.omit() %>%
  tally() %>% #creating crime count column
  rename(`Crime Count`=n) %>%  #renaming crime count column %>%
  right_join(population_dataset, by = "Short Postcode") %>% #joining with population dataset to show district and population
  select(`Short Postcode`,`Crime type`,`Crime Count`, `Population`, `Date of crime`, `Falls within`, `Town/City`, District) %>% #select the required co
  na.omit() %>%
  filter(`Crime type`== "Drugs") %>% #filtering to show only drug crimes of 2022
  mutate(`Drug Offence Rate` = (`Crime Count` / Population)) #calculating drug offence rate

#grouping the drug crime dataset by county and town and showing the rate for each group for the year 2021
grouped_drug_crime <- crime_dataset_drugs2 %>%
  filter(`Date of crime`=="2021") %>%
  group_by(`Falls within`,`Town/City`) %>%
  mutate(`Town/City`= tolower(`Town/City`)) %>%  #converting the town from to all lowercase
  summarise(`Drug Offence Rate`= mean(`Drug Offence Rate`))

#joining school data and house price data in a single table
school_drug_data = grouped_school_dataset %>%
  left_join(grouped_drug_crime ,by=c("Town"="Town/City")) %>%
  na.omit #removing rows with null value

#creating a linear model
l_model = lm(data=school_drug_data, `Attainment Score`~`Drug Offence Rate`) #this model predicts Average attainment score as a function of Drug offence

#showing summary of the Linear Model
summary(l_model)


#creating the linear model graph
ggplot(school_drug_data,aes(x=`Drug Offence Rate`,y= `Attainment Score`)) +
  scale_y_continuous(limits=c(0,50), breaks = seq(0,50,5))+ #setting limits and breaks
  geom_point(data = filter(school_drug_data,County=="Kent"),aes(color=c("Red"="Kent")))+ #setting color as red for Kent's data point
  geom_point(data = filter(school_drug_data,County=="Surrey"), aes(color=c("Blue"="Surrey"))) + #setting color as blye for Surrey's data point
  geom_smooth(method=lm,se=FALSE,color="lightgreen")+ #adding linear regression line and omitting error bands
  labs(x="Drug Offence Rate",
       y="Attainment Score",
```

```r
#grouping house prices by town and county and finding average price for each group
grouped_house_prices = cleaned_houseprices %>%
  filter(`Date of Transfer`=="2021") %>%
  group_by(`Town/City`,County) %>%
  mutate(`Town/City` = tolower(`Town/City`)) %>% #converting the town from uppercase to all lowercase
  summarise(Price=mean(Price))


#grouping school data by town and county and finding average score for each group
grouped_school_dataset = cleaned_school_dataset %>%
  filter(`Year`=="2021") %>%
  group_by(`Town`,County) %>%
  mutate(Town= tolower(Town)) %>%  #converting the town from to all lowercase
  summarise(`Attainment Score`=mean(`Attainment Score`))


#joining school data and house price data in a single table
school_houseprice_data = grouped_school_dataset %>%
  left_join(grouped_house_prices,by=c("Town"="Town/City")) %>%
  na.omit #removing rows with null value

#creating a linear model
l_model = lm(data=school_houseprice_data, `Attainment Score`~Price) #this model predicts Average attainment score as a function of Average house prices

#showing summary of the Linear Model
summary(l_model)

#creating the linear model graph
ggplot(school_houseprice_data,aes(x=Price,y= `Attainment Score`)) +
  scale_y_continuous(limits=c(0,80), breaks = seq(0,80,5))+ #setting limits and breaks
  geom_point(data = filter(school_houseprice_data,County.x=="Kent"),aes(color=c("Red"="Kent")))+ #setting color as red for Kent's data point
  geom_point(data = filter(school_houseprice_data,County.x=="Surrey"), aes(color=c("Blue"="Surrey"))) + #setting color as blye for Surrey's data point
  geom_smooth(method=lm,se=FALSE,color="lightgreen")+ #adding linear regression line and omitting error bands
  labs(x="House Price",
       y="Attainment Score",
       title="2021 Attainment Score vs House Prices",color="County") #setting labels

```

```r
#modifying our crime dataset to show drug offence rate and crime count
crime_dataset_drugs2 <-cleaned_crime_dataset %>%
  mutate(`Date of crime`= substr(`Date of crime`, 1, 4)) %>% #Mutating this column to only show year
  group_by(`Short Postcode`,`Crime type`,`Date of crime`, `Falls within`) %>% #Grouping to show crime count in each postcode by year
  select(`Short Postcode`,`Crime type`,`Date of crime`, `Falls within`) %>%
  na.omit() %>%
  tally() %>% #creating crime count column
  rename(`Crime Count`=n) %>%  #renaming crime count column %>%
  right_join(population_dataset, by = "Short Postcode") %>% #joining with population dataset to show district and population
  select(`Short Postcode`,`Crime type`,`Crime Count`, `Population`, `Date of crime`, `Falls within`, `Town/City`, District) %>% #select the required co
  na.omit() %>%
  filter(`Crime type`== "Drugs") %>% #filtering to show only drug crimes of 2022
  mutate(`Drug Offence Rate` = (`Crime Count` / Population)) #calculating drug offence rate

#grouping the drug crime dataset by county and town and showing the rate for each group for the year 2020
grouped_drug_crime <- crime_dataset_drugs2 %>%
  filter(`Date of crime`=="2021") %>%
  group_by(`Falls within`,`Town/City`) %>%
  summarise(`Drug Offence Rate`= mean(`Drug Offence Rate`))


#joining house price data and drug crime rate data in a single table
house_price_drug_crime_data = grouped_house_prices %>%
  left_join(grouped_drug_crime,by="Town/City") %>%
  na.omit #removing null values
http://127.0.0.1:26089/graphics/5a4605a6-f469-49de-992e-e087dc02f32a.png

#creating a linear model
l_model = lm(data=house_price_drug_crime_data, Price~`Drug Offence Rate`) #this model predicts House Price as a function of Drug offence rate

#showing summary of the Linear Model
summary(l_model)

#creating the linear model graph
ggplot(house_price_drug_crime_data,aes(x=`Drug Offence Rate`,y=Price)) +
  scale_y_continuous(limits=c(0,1000000), breaks = seq(0,1000000,200000))+ #setting limits and breaks
  geom_point(data = filter(house_price_drug_crime_data,County=="KENT"),aes(color=c("Red"="Kent")))+ #setting color as red for Kent's data point
  geom_point(data = filter(house_price_drug_crime_data,County=="SURREY"), aes(color=c("Blue"="Surrey"))) + #setting color as blue for Surrey's data poi
  geom_smooth(method=lm,se=FALSE,color="lightgreen")+ #adding linear regression line and omitting error bands
  labs(x="Drug Offence Rate",
       y="Price",
       title="2021 House Prices vs Drug Offence Rate",color="County") #setting labels

```