Report on an Individual Database Development Project

Utsav Prajapati

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

ST5014CEM: Data Science for Developers

Siddhartha Neupane

July 30, 2022

TABLE OF CONTENT

ASSIGNMENT BRIEFING

The scenario we were provided with was that our relatives are seeking to purchase a house in the Greater Manchester or Merseyside areas. The assignment asks us to make a recommendation for an appropriate place based on the price of the house, broadband speed, and local crimes. We were also asked to add additional elements that could influence a person's choice of place, so for this school data has been used.

The objectives of the assignment are to understand and apply the data science lifecycle to real-world data problems and analyze, design, implement, manage, and critically evaluate a solution for a commercial or scientific objective using R.

The followed approach to this assignment is cleaning the data, creating a uniform data model, using exploratory data analysis to examine the dataset, looking into statistical links between the attributes, and creating a basic recommendation system that determines a value in the range 0-10 for each of the characteristics and displaying the top three towns in order.

OBTAINING DATA

The datasets that are used in this assignment are the House prices dataset, broadband dataset, local crime dataset, and school dataset. An additional dataset that is used is population dataset. Datasets that have been released by the UK government are used. The data are downloaded in .csv format.

DATA CLEANING

Data cleaning or scrubbing is the second phase in the data science OSEMN pipeline. This is the most important part of the lifecycle and also requires more time and effort because the final results depend upon the data, we put in.

Firstly, the datasets are examined, errors along with corrupted records are identified, and then the necessary scrubbing proceeds. The downloaded data were inconsistent, the dataset was not understandable due to irregular names for columns, there were missing values, and various other forms of issues. Also, some datasets had to be merged as there were multiple datasets divided by months. The following are the process that has carried out to clean the datasets.

House Prices Data cleaning

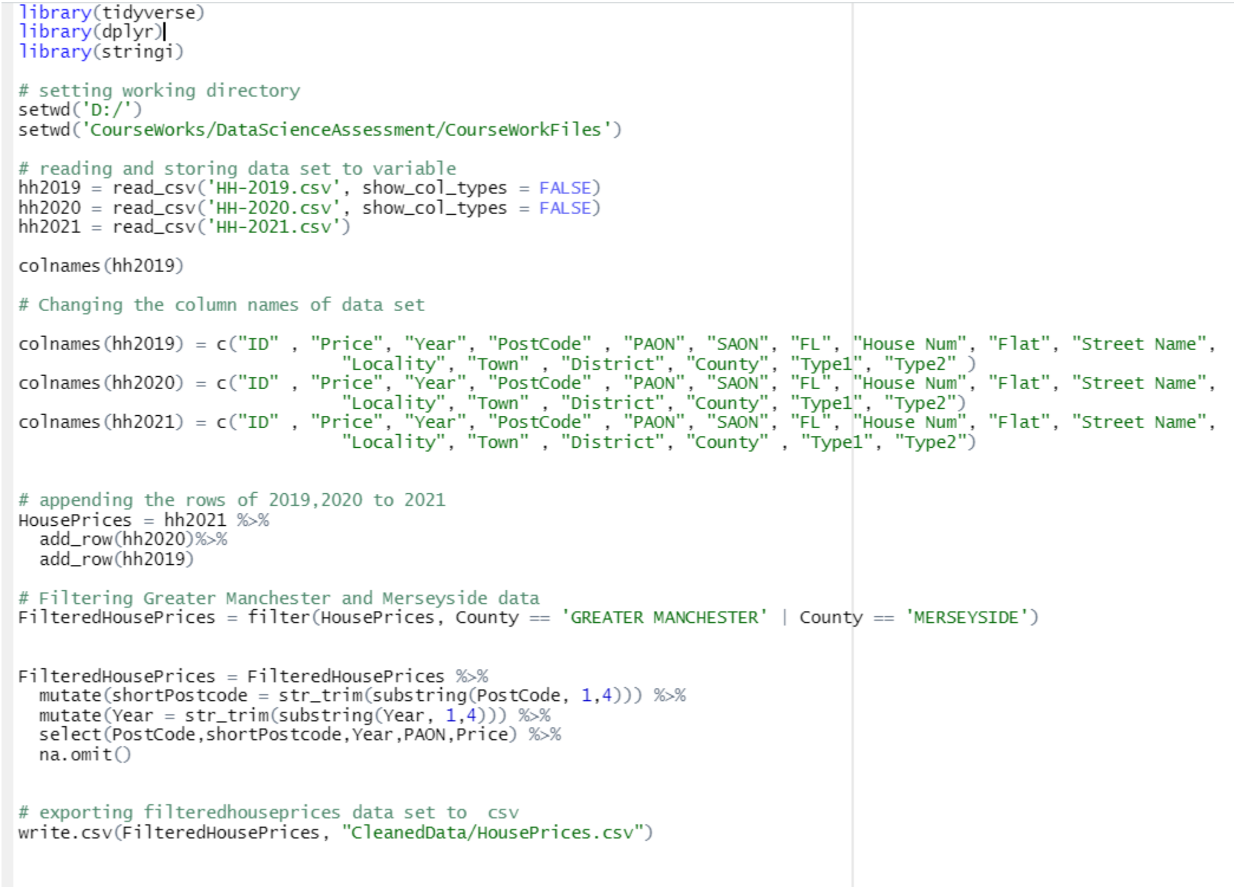
Three house price datasets for the years 2019, 2020, and 2021 were downloaded and imported. Upon examining the data, the column names were based on the data itself, so the first step was to rename the column names. This step simplified examining the data.

The three datasets were then merged into one single dataset. As per the requirement of the assignment, the data from Greater Manchester and Merseyside were filtered.

The required data were Postcode, short postcode, year, PAON, and price. The short post code is generated with the trimmed postcode. The date data contained year, month, and day so only the year was trimmed out of it. After filtering the necessary data and fixing the missing values, the dataset is exported to a ‘CleanedData’ folder with the name HousePrices.csv.

Figure 1

Code for cleaned house price



```
library(tidyverse)
library(dplyr)
library(stringi)

# setting working directory
setwd('D:/')
setwd('CourseWorks/DataScienceAssessment/CourseWorkFiles')

# reading and storing data set to variable
hh2019 = read_csv('HH-2019.csv', show_col_types = FALSE)
hh2020 = read_csv('HH-2020.csv', show_col_types = FALSE)
hh2021 = read_csv('HH-2021.csv')

colnames(hh2019)

# Changing the column names of data set

colnames(hh2019) = c("ID" , "Price", "Year", "PostCode" , "PAON", "SAON", "FL", "House Num", "Flat", "Street Name",
                              "Locality", "Town" , "District", "County", "Type1", "Type2" )
colnames(hh2020) = c("ID" , "Price", "Year", "PostCode" , "PAON", "SAON", "FL", "House Num", "Flat", "Street Name",
                              "Locality", "Town" , "District", "County", "Type1", "Type2")
colnames(hh2021) = c("ID" , "Price", "Year", "PostCode" , "PAON", "SAON", "FL", "House Num", "Flat", "Street Name",
                              "Locality", "Town" , "District", "County" , "Type1", "Type2")


# appending the rows of 2019,2020 to 2021
HousePrices = hh2021 %>%
  add_row(hh2020)%>%
  add_row(hh2019)

# Filtering Greater Manchester and Merseyside data
FilteredHousePrices = filter(HousePrices, County == 'GREATER MANCHESTER' | County == 'MERSEYSIDE')


FilteredHousePrices = FilteredHousePrices %>%
  mutate(shortPostcode = str_trim(substring(PostCode, 1,4))) %>%
  mutate(Year = str_trim(substring(Year, 1,4))) %>%
  select(PostCode,shortPostcode,Year,PAON,Price) %>%
  na.omit()


# exporting filteredhouseprices data set to  csv
write.csv(FilteredHousePrices, "CleanedData/HousePrices.csv")
```

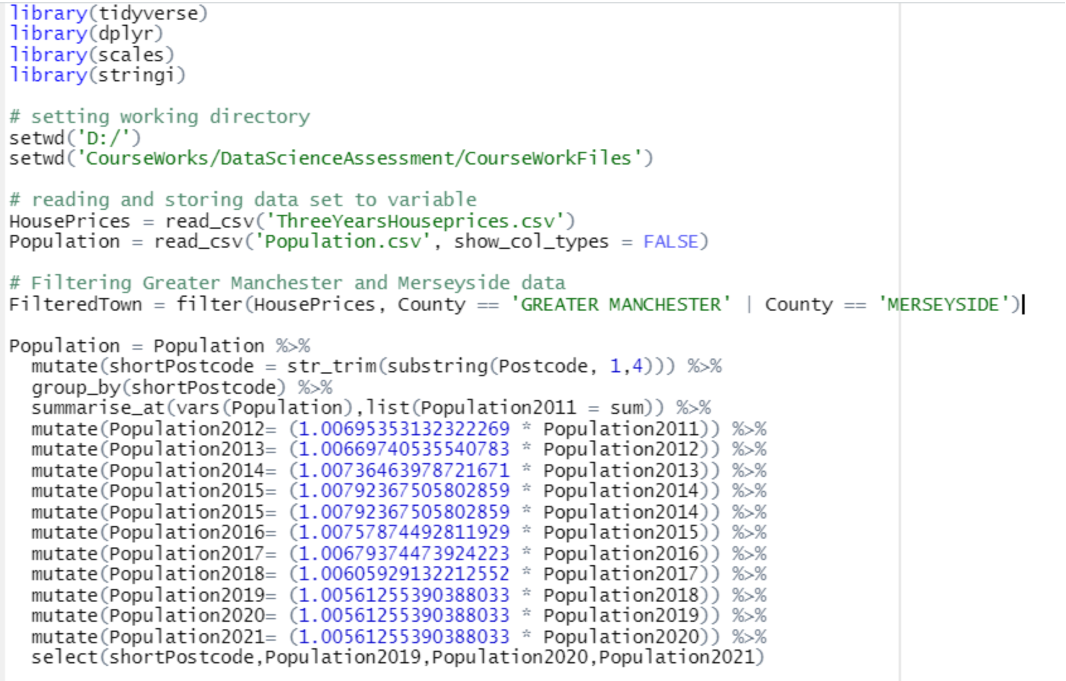
Towns Data Cleaning

As there was no town dataset specifically. The house prices dataset has been used as it contained necessary data of towns and districts in Greater Manchester and Merseyside. Our teacher provided us with population data for the year 2011 and asked us to join the population data with town data.

The first step was to filter out required data for the town dataset such as short postcode, Town, District, and County from the house prices dataset. After that, the population data was examined. As the population data contained data for the year 2011 only, the population data up to the year 2021 was generated and merged.

Figure 2

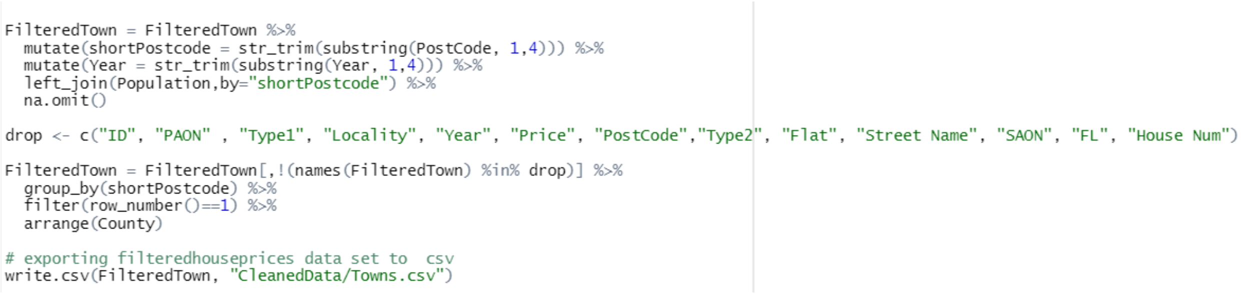
Code for filtering data and generating population data



```
library(tidyverse)
library(dplyr)
library(scales)
library(stringi)

# setting working directory
setwd('D:/')
setwd('CourseWorks/DataScienceAssessment/CourseWorkFiles')

# reading and storing data set to variable
HousePrices = read_csv('ThreeYearsHouseprices.csv')
Population = read_csv('Population.csv', show_col_types = FALSE)

# Filtering Greater Manchester and Merseyside data
FilteredTown = filter(HousePrices, County == 'GREATER MANCHESTER' | County == 'MERSEYSIDE')

Population = Population %>%
  mutate(shortPostcode = str_trim(substring(Postcode, 1,4))) %>%
  group_by(shortPostcode) %>%
  summarise_at(vars(Population),list(Population2011 = sum)) %>%
  mutate(Population2012= (1.00695353132322269 * Population2011)) %>%
  mutate(Population2013= (1.00669740535540783 * Population2012)) %>%
  mutate(Population2014= (1.00736463978721671 * Population2013)) %>%
  mutate(Population2015= (1.00792367505802859 * Population2014)) %>%
  mutate(Population2015= (1.00792367505802859 * Population2014)) %>%
  mutate(Population2016= (1.00757874492811929 * Population2015)) %>%
  mutate(Population2017= (1.00679374473924223 * Population2016)) %>%
  mutate(Population2018= (1.00605929132212552 * Population2017)) %>%
  mutate(Population2019= (1.00561255390388033 * Population2018)) %>%
  mutate(Population2020= (1.00561255390388033 * Population2019)) %>%
  mutate(Population2021= (1.00561255390388033 * Population2020)) %>%
  select(shortPostcode,Population2019,Population2020,Population2021)
```

Figure 3

Code for joining the population data with town data



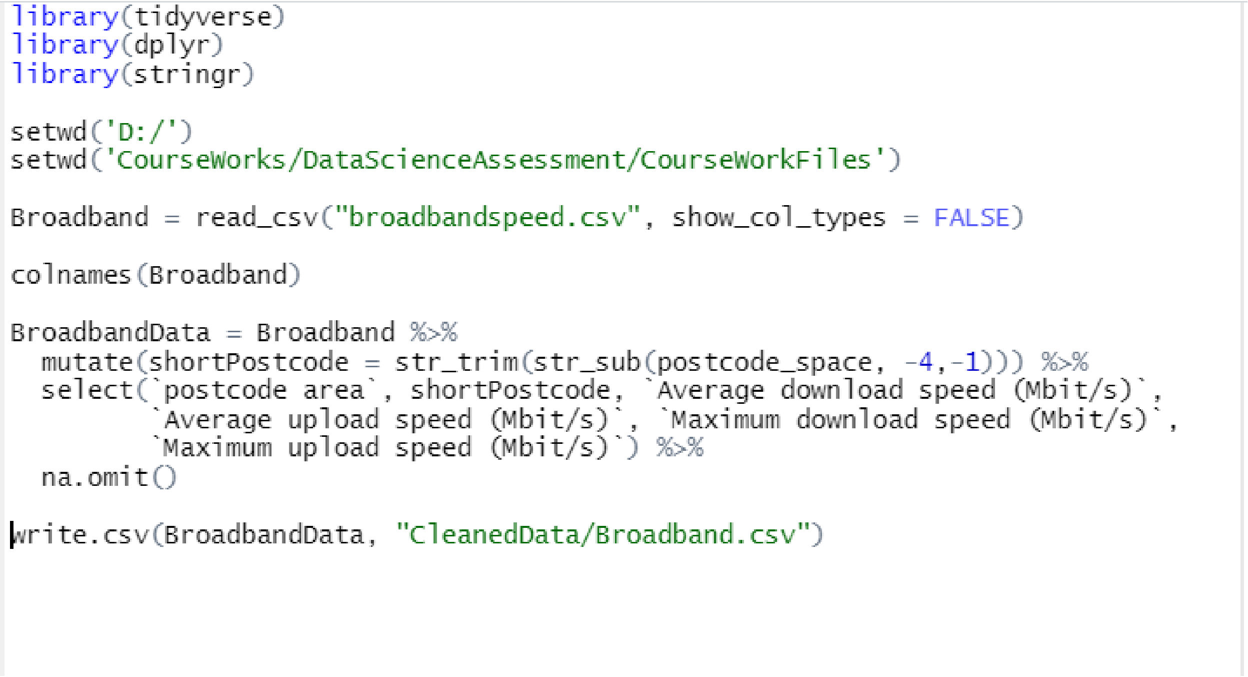
```r
FilteredTown = FilteredTown %>%
  mutate(shortPostcode = str_trim(substring(PostCode, 1,4))) %>%
  mutate(Year = str_trim(substring(Year, 1,4))) %>%
  left_join(Population,by="shortPostcode") %>%
  na.omit()

drop <- c("ID", "PAON" , "Type1", "Locality", "Year", "Price", "PostCode","Type2", "Flat", "Street Name", "SAON", "FL", "House Num")

FilteredTown = FilteredTown[,!(names(FilteredTown) %in% drop)] %>%
  group_by(shortPostcode) %>%
  filter(row_number()==1) %>%
  arrange(County)

# exporting filteredhouseprices data set to  csv
write.csv(FilteredTown, "CleanedData/Towns.csv")
```

Broadband Speed Data Cleaning

Upon examining the dataset, it contained download speed, upload speed, and data usage data. Postcode area, short postcode, average download speed, average upload speed, maximum download speed, and minimum download speed data were filtered out, then the rows with missing values were removed.

Figure 4

Code for cleaning broadband speed data



```
library(tidyverse)
library(dplyr)
library(stringr)

setwd('D:/')
setwd('CourseWorks/DataScienceAssessment/CourseWorkFiles')

Broadband = read_csv("broadbandspeed.csv", show_col_types = FALSE)

colnames(Broadband)

BroadbandData = Broadband %>%
  mutate(shortPostcode = str_trim(str_sub(postcode_space, -4,-1))) %>%
  select(`postcode area`, shortPostcode, `Average download speed (Mbit/s)`,
         `Average upload speed (Mbit/s)`, `Maximum download speed (Mbit/s)`,
         `Maximum upload speed (Mbit/s)`) %>%
  na.omit()

write.csv(BroadbandData, "CleanedData/Broadband.csv")
```

School Data Cleaning

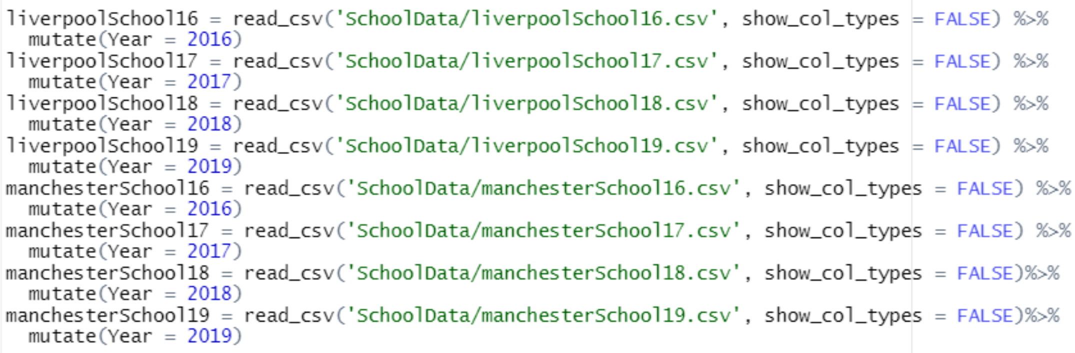
School data for the year 2016-2021 of Manchester and Liverpool were downloaded. The downloaded zip file for each district contained multiple csv files. The required data were Postcode, short postcode Year, school Name, and Attainment8score.

Upon examining the data, the files that contained attainment8score was ‘ks4final.csv’ from 2016-2019 Manchester school and Liverpool school data. As the 2020-2021 Manchester and Liverpool data did not contain attainment8score data, the datasets that were not used. Before importing the dataset to R, the files were renamed to ‘manchesterSchool.csv’ and ‘liverpoolSchool.csv’ respectively.

As there were two files for a different location, they had to be merged, while doing so there occurred a problem with an unconcerned column. That’s why only required columns are selected at the beginning. After that, the two datasets were merged.

Figure 4.1

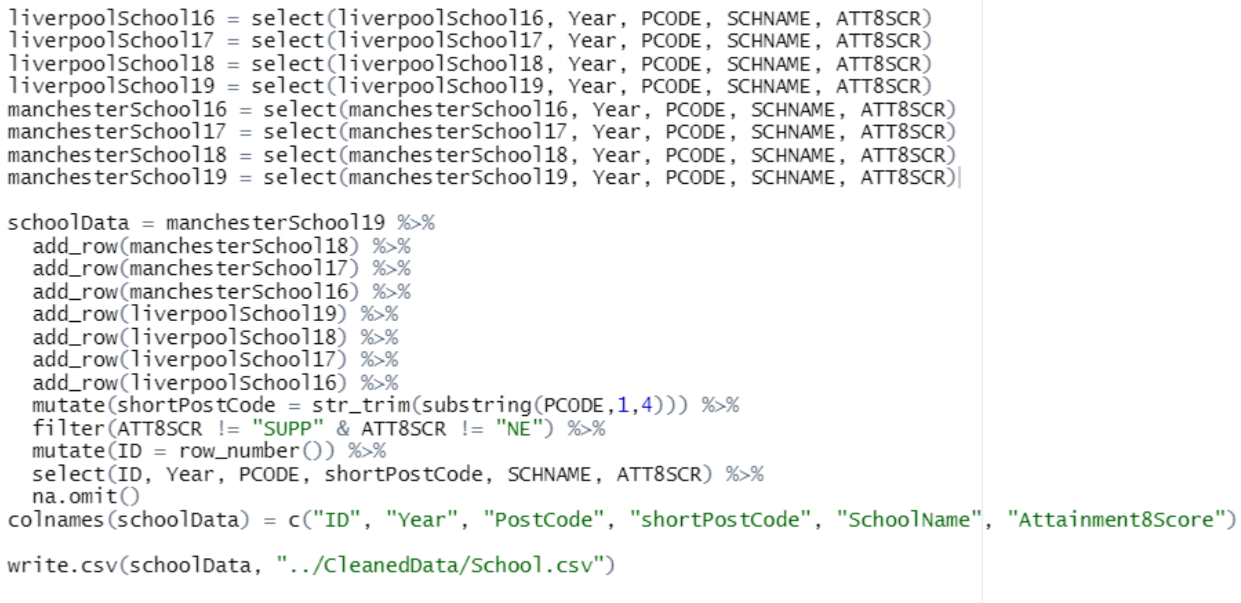
Code for cleaning school data



```
liverpoolSchool16 = read_csv('SchoolData/liverpoolSchool16.csv', show_col_types = FALSE) %>%
  mutate(Year = 2016)
liverpoolSchool17 = read_csv('SchoolData/liverpoolSchool17.csv', show_col_types = FALSE) %>%
  mutate(Year = 2017)
liverpoolSchool18 = read_csv('SchoolData/liverpoolSchool18.csv', show_col_types = FALSE) %>%
  mutate(Year = 2018)
liverpoolSchool19 = read_csv('SchoolData/liverpoolSchool19.csv', show_col_types = FALSE) %>%
  mutate(Year = 2019)
manchesterSchool16 = read_csv('SchoolData/manchesterSchool16.csv', show_col_types = FALSE) %>%
  mutate(Year = 2016)
manchesterSchool17 = read_csv('SchoolData/manchesterSchool17.csv', show_col_types = FALSE) %>%
  mutate(Year = 2017)
manchesterSchool18 = read_csv('SchoolData/manchesterSchool18.csv', show_col_types = FALSE)%>%
  mutate(Year = 2018)
manchesterSchool19 = read_csv('SchoolData/manchesterSchool19.csv', show_col_types = FALSE)%>%
  mutate(Year = 2019)
```

Figure 4.2

Code for cleaning school data



```
liverpoolSchool16 = select(liverpoolSchool16, Year, PCODE, SCHNAME, ATT8SCR)
liverpoolSchool17 = select(liverpoolSchool17, Year, PCODE, SCHNAME, ATT8SCR)
liverpoolSchool18 = select(liverpoolSchool18, Year, PCODE, SCHNAME, ATT8SCR)
liverpoolSchool19 = select(liverpoolSchool19, Year, PCODE, SCHNAME, ATT8SCR)
manchesterSchool16 = select(manchesterSchool16, Year, PCODE, SCHNAME, ATT8SCR)
manchesterSchool17 = select(manchesterSchool17, Year, PCODE, SCHNAME, ATT8SCR)
manchesterSchool18 = select(manchesterSchool18, Year, PCODE, SCHNAME, ATT8SCR)
manchesterSchool19 = select(manchesterSchool19, Year, PCODE, SCHNAME, ATT8SCR)

schoolData = manchesterSchool19 %>%
  add_row(manchesterSchool18) %>%
  add_row(manchesterSchool17) %>%
  add_row(manchesterSchool16) %>%
  add_row(liverpoolSchool19) %>%
  add_row(liverpoolSchool18) %>%
  add_row(liverpoolSchool17) %>%
  add_row(liverpoolSchool16) %>%
  mutate(shortPostCode = str_trim(substring(PCODE,1,4))) %>%
  filter(ATT8SCR != "SUPP" & ATT8SCR != "NE") %>%
  mutate(ID = row_number()) %>%
  select(ID, Year, PCODE, shortPostCode, SCHNAME, ATT8SCR) %>%
  na.omit()
colnames(schoolData) = c("ID", "Year", "PostCode", "shortPostCode", "SchoolName", "Attainment8Score")

write.csv(schoolData, "../CleanedData/School.csv")
```

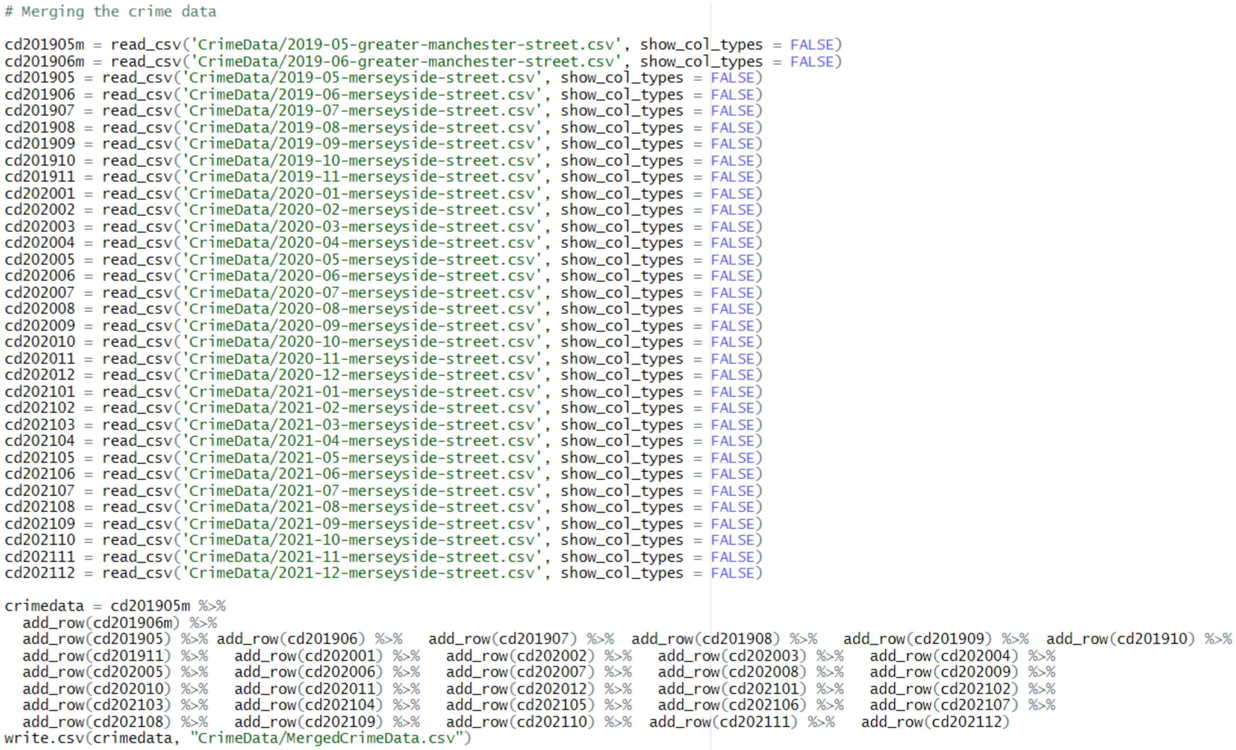
Crime Data Cleaning

The downloaded crime data zip file for the years 2019, 2020, and 2021 contained numerous subfolder that contained csv files divided by months. The zip file contained only two csv files of Greater Manchester for the year 2019 and none for the year 2020, and 2021. The first step was to organize all of the csv files into a single folder to make importing all of the csv files into R easier.

The datasets were then imported and merged into a single dataset.

Figure 5.1

Code for merging all the crime data into one dataset

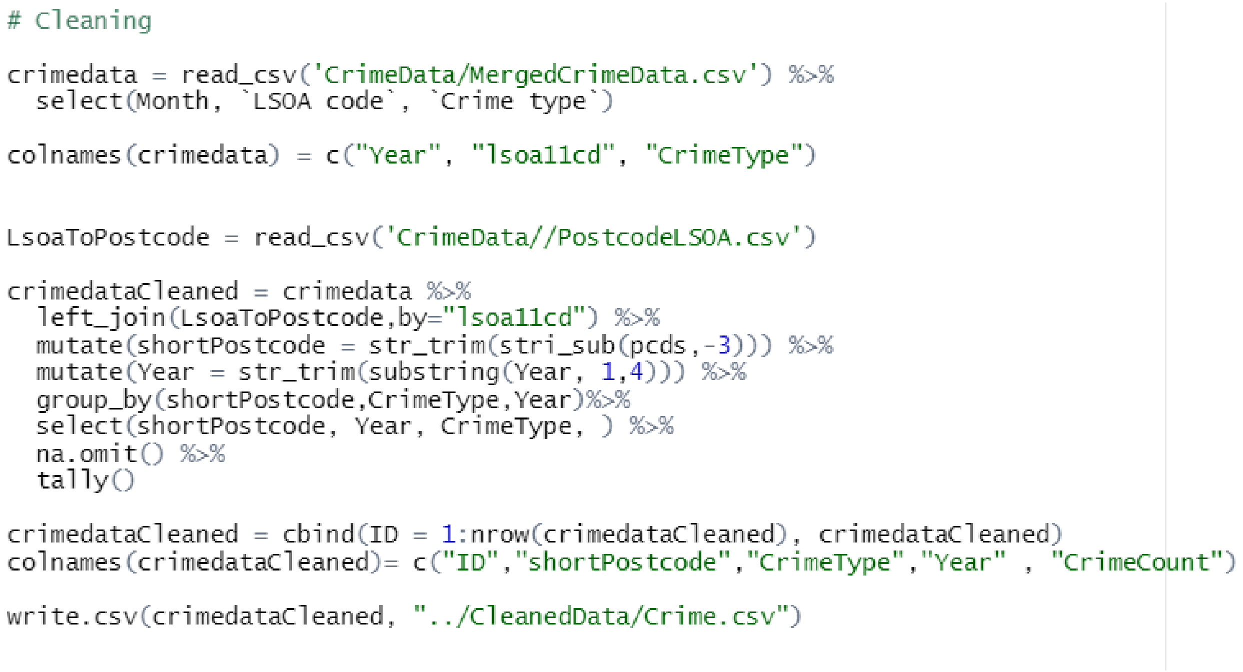


```r
# Merging the crime data

cd201905m = read_csv('CrimeData/2019-05-greater-manchester-street.csv', show_col_types = FALSE)
cd201906m = read_csv('CrimeData/2019-06-greater-manchester-street.csv', show_col_types = FALSE)
cd201905 = read_csv('CrimeData/2019-05-merseyside-street.csv', show_col_types = FALSE)
cd201906 = read_csv('CrimeData/2019-06-merseyside-street.csv', show_col_types = FALSE)
cd201907 = read_csv('CrimeData/2019-07-merseyside-street.csv', show_col_types = FALSE)
cd201908 = read_csv('CrimeData/2019-08-merseyside-street.csv', show_col_types = FALSE)
cd201909 = read_csv('CrimeData/2019-09-merseyside-street.csv', show_col_types = FALSE)
cd201910 = read_csv('CrimeData/2019-10-merseyside-street.csv', show_col_types = FALSE)
cd201911 = read_csv('CrimeData/2019-11-merseyside-street.csv', show_col_types = FALSE)
cd202001 = read_csv('CrimeData/2020-01-merseyside-street.csv', show_col_types = FALSE)
cd202002 = read_csv('CrimeData/2020-02-merseyside-street.csv', show_col_types = FALSE)
cd202003 = read_csv('CrimeData/2020-03-merseyside-street.csv', show_col_types = FALSE)
cd202004 = read_csv('CrimeData/2020-04-merseyside-street.csv', show_col_types = FALSE)
cd202005 = read_csv('CrimeData/2020-05-merseyside-street.csv', show_col_types = FALSE)
cd202006 = read_csv('CrimeData/2020-06-merseyside-street.csv', show_col_types = FALSE)
cd202007 = read_csv('CrimeData/2020-07-merseyside-street.csv', show_col_types = FALSE)
cd202008 = read_csv('CrimeData/2020-08-merseyside-street.csv', show_col_types = FALSE)
cd202009 = read_csv('CrimeData/2020-09-merseyside-street.csv', show_col_types = FALSE)
cd202010 = read_csv('CrimeData/2020-10-merseyside-street.csv', show_col_types = FALSE)
cd202011 = read_csv('CrimeData/2020-11-merseyside-street.csv', show_col_types = FALSE)
cd202012 = read_csv('CrimeData/2020-12-merseyside-street.csv', show_col_types = FALSE)
cd202101 = read_csv('CrimeData/2021-01-merseyside-street.csv', show_col_types = FALSE)
cd202102 = read_csv('CrimeData/2021-02-merseyside-street.csv', show_col_types = FALSE)
cd202103 = read_csv('CrimeData/2021-03-merseyside-street.csv', show_col_types = FALSE)
cd202104 = read_csv('CrimeData/2021-04-merseyside-street.csv', show_col_types = FALSE)
cd202105 = read_csv('CrimeData/2021-05-merseyside-street.csv', show_col_types = FALSE)
cd202106 = read_csv('CrimeData/2021-06-merseyside-street.csv', show_col_types = FALSE)
cd202107 = read_csv('CrimeData/2021-07-merseyside-street.csv', show_col_types = FALSE)
cd202108 = read_csv('CrimeData/2021-08-merseyside-street.csv', show_col_types = FALSE)
cd202109 = read_csv('CrimeData/2021-09-merseyside-street.csv', show_col_types = FALSE)
cd202110 = read_csv('CrimeData/2021-10-merseyside-street.csv', show_col_types = FALSE)
cd202111 = read_csv('CrimeData/2021-11-merseyside-street.csv', show_col_types = FALSE)
cd202112 = read_csv('CrimeData/2021-12-merseyside-street.csv', show_col_types = FALSE)

crimedata = cd201905m %>%
  add_row(cd201906m) %>%
  add_row(cd201905) %>% add_row(cd201906) %>%   add_row(cd201907) %>%  add_row(cd201908) %>%   add_row(cd201909) %>%  add_row(cd201910) %>%
  add_row(cd201911) %>%   add_row(cd202001) %>%   add_row(cd202002) %>%   add_row(cd202003) %>%   add_row(cd202004) %>%
  add_row(cd202005) %>%   add_row(cd202006) %>%   add_row(cd202007) %>%   add_row(cd202008) %>%   add_row(cd202009) %>%
  add_row(cd202010) %>%   add_row(cd202011) %>%   add_row(cd202012) %>%   add_row(cd202101) %>%   add_row(cd202102) %>%
  add_row(cd202103) %>%   add_row(cd202104) %>%   add_row(cd202105) %>%   add_row(cd202106) %>%   add_row(cd202107) %>%
  add_row(cd202108) %>%   add_row(cd202109) %>%   add_row(cd202110) %>%  add_row(cd202111) %>%   add_row(cd202112)
write.csv(crimedata, "CrimeData/MergedCrimeData.csv")
```

The merged crime dataset was imported, and the process continued. The dataset did not include postcode data, but rather LSOA codes that had to be converted to postcodes. Another dataset containing LSOA, and postcode was used to obtain postcodes from LSOA. The crime data is then joined with the LSOA postcode data. Finally, the required data is filtered out, such as postcode, short postcode, crime type, year, and crime type count, and the cleaned dataset is exported as 'Crime.csv'.

Figure 5.2

Code for cleaning crime data into



```
# Cleaning

crimedata = read_csv('CrimeData/MergedCrimeData.csv') %>%
  select(Month, `LSOA code`, `Crime type`)

colnames(crimedata) = c("Year", "lsoa11cd", "CrimeType")


LsoaToPostcode = read_csv('CrimeData//PostcodeLSOA.csv')

crimedataCleaned = crimedata %>%
  left_join(LsoaToPostcode,by="lsoa11cd") %>%
  mutate(shortPostcode = str_trim(stri_sub(pcds,-3))) %>%
  mutate(Year = str_trim(substring(Year, 1,4))) %>%
  group_by(shortPostcode,CrimeType,Year)%>%
  select(shortPostcode, Year, CrimeType, ) %>%
  na.omit() %>%
  tally()

crimedataCleaned = cbind(ID = 1:nrow(crimedataCleaned), crimedataCleaned)
colnames(crimedataCleaned)= c("ID","shortPostcode","CrimeType","Year" , "CrimeCount")

write.csv(crimedataCleaned, "../CleanedData/Crime.csv")
```

EXPLORATORY DATA ANALYSIS – DATA VISUALIZATION

Exploratory data analysis is the process of describing the data by means of visualization and statistical techniques in order to bring important aspects of data for further analysis. This phase involves inspecting the cleaned dataset from many angles and describing it without assumption.

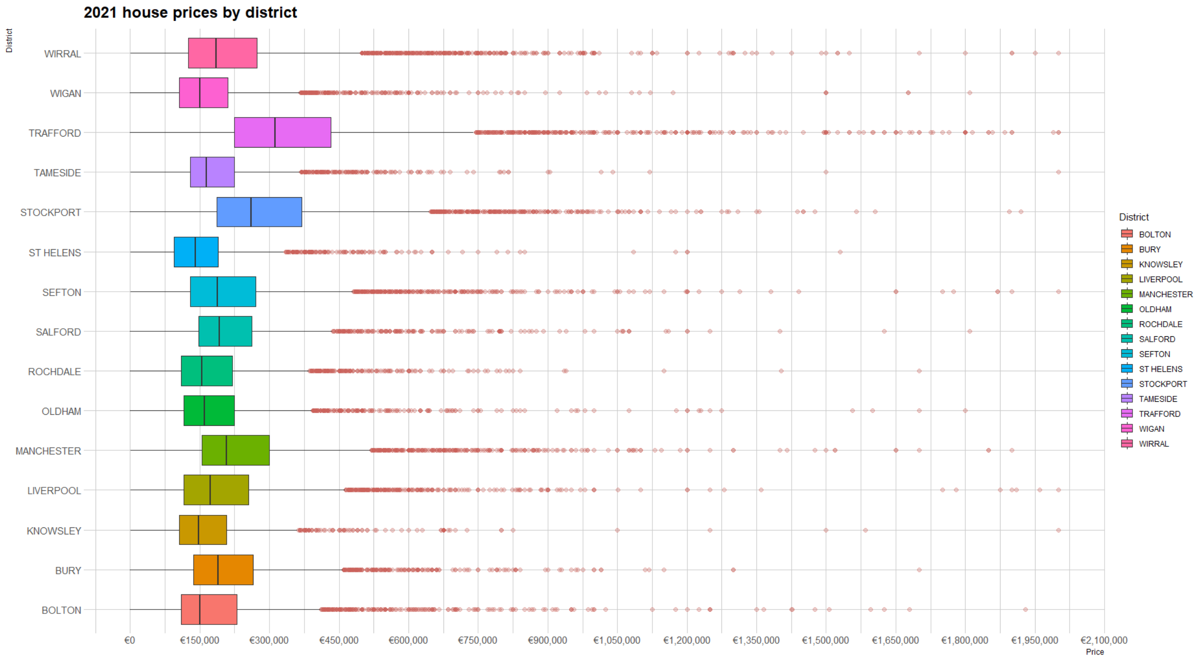
After the data preparation, visualization and analysis is done in this phase. The following are the visualization and observation of datasets.

House Prices Visualization

The average house price data was first visualized using a boxplot. The boxplot shows the data in the minimum, median, upper quartile, and maximum values at the 25th percentile, 50th percentile, and 75th percentile, respectively (*Understanding Boxplots* 2022).

Figure 6

Box plot of 2021 House prices by district

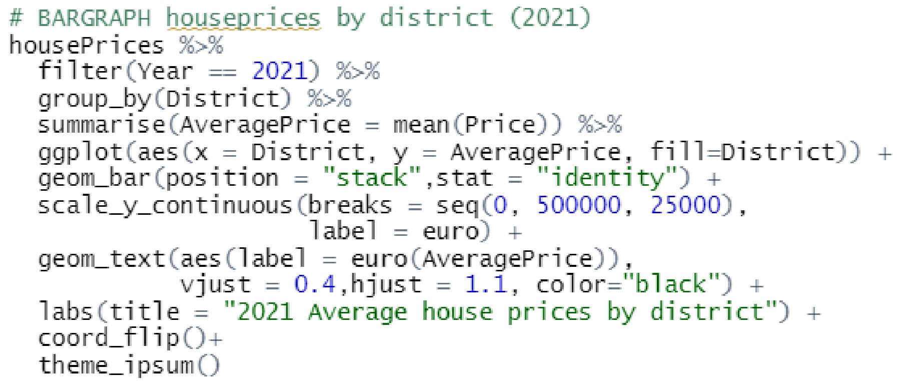


Greater Manchester and Merseyside have housing costs ranging from $100 to $127937135. Summarizing the data, we find that the minimum value is €100, the 25th percentile price is €118000, the median is €170000 with an average value of €228443, the 75th percentile price is €250000, and the maximum value is €127937183. The price has been set at 2 million euros because the maximum price of over 100 million euros caused problems with visualization. Basic observations that could be drawn from the boxplot include:

St Helens has the lowest median price among the districts at €140,000, and Trafford has the highest median price at €315,000. Trafford median price exceeds €300,000 which is the not even the upper quartile prices for most of the other districts except for Manchester and Stockport, whose upper quartile are €300,000 and €370,000.

Figure 7

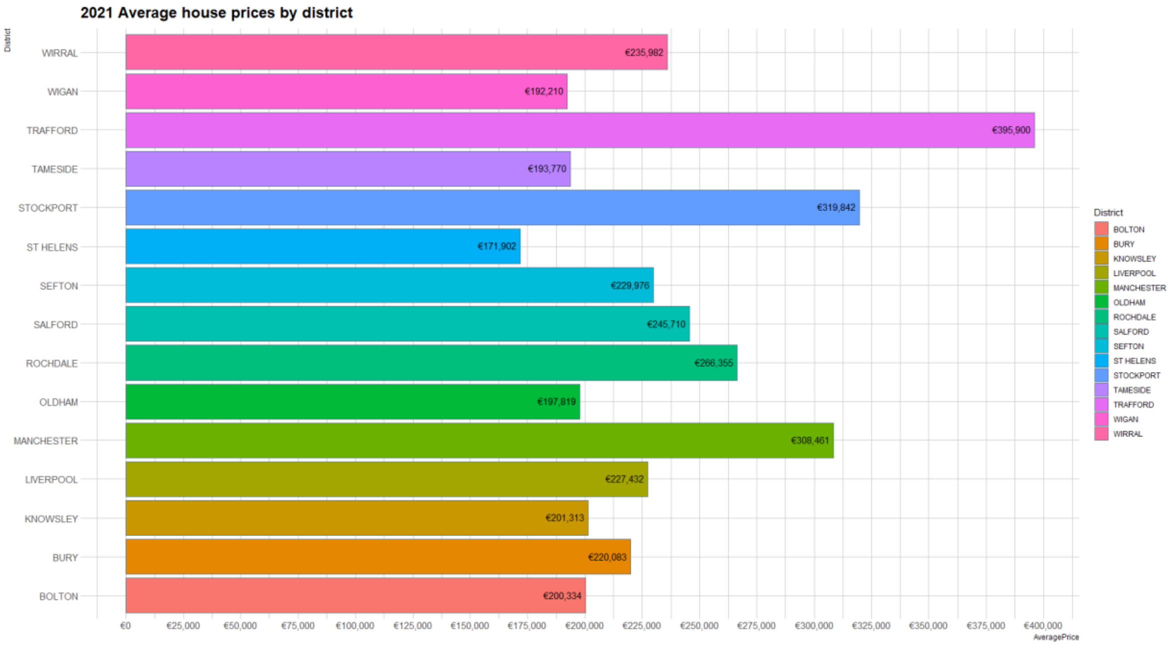
Code for *Bar chart of 2021 House prices by district*



```
# BARGRAPH houseprices by district (2021)
housePrices %>%
  filter(Year == 2021) %>%
  group_by(District) %>%
  summarise(AveragePrice = mean(Price)) %>%
  ggplot(aes(x = District, y = AveragePrice, fill=District)) +
  geom_bar(position = "stack",stat = "identity") +
  scale_y_continuous(breaks = seq(0, 500000, 25000),
                     label = euro) +
  geom_text(aes(label = euro(AveragePrice)),
            vjust = 0.4,hjust = 1.1, color="black") +
  labs(title = "2021 Average house prices by district") +
  coord_flip()+
  theme_ipsum()
```

Figure 8

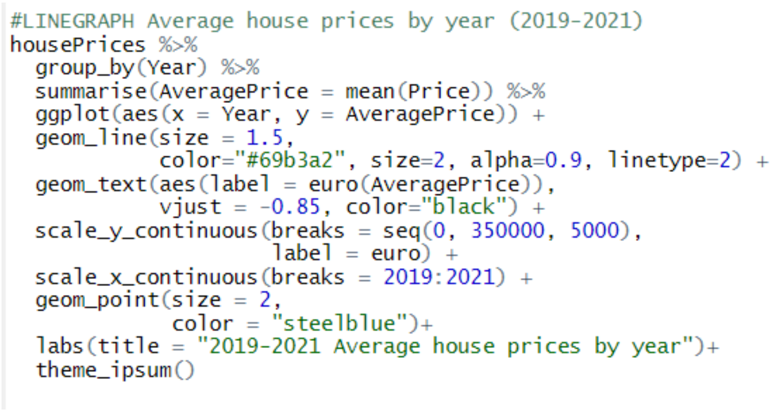
Bar chart of 2021 House prices by district



The above bar graph represents the average house prices for districts in 2021. According to the bar chart observations, Trafford has the highest average house price of €395,900 and St Helens has the lowest average house price of €171,902 in the chart. Most of the districts' average house prices are higher than €200,000. St Helens would be an ideal choice for affordable housing.

Figure 9

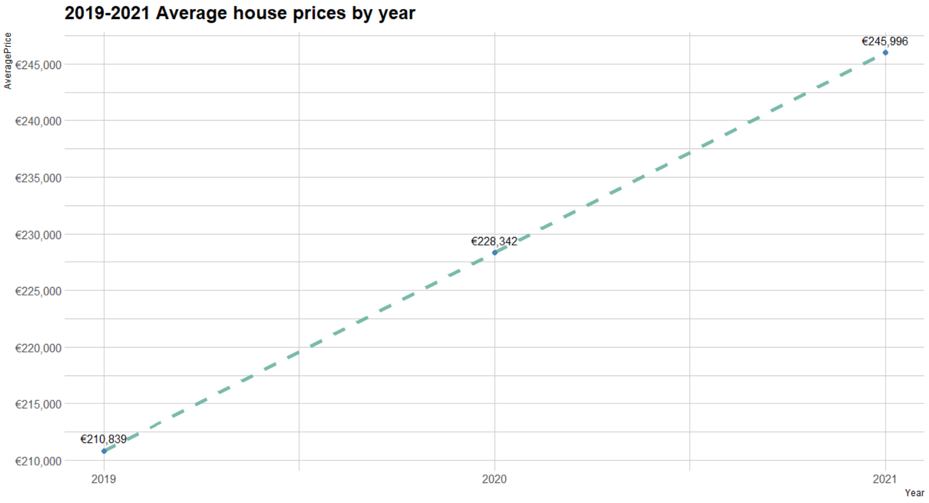
Code for *Line graph of average house prices for the year 2019-2021*



```
#LINEGRAPH Average house prices by year (2019-2021)
housePrices %>%
  group_by(Year) %>%
  summarise(AveragePrice = mean(Price)) %>%
  ggplot(aes(x = Year, y = AveragePrice)) +
  geom_line(size = 1.5,
            color="#69b3a2", size=2, alpha=0.9, linetype=2) +
  geom_text(aes(label = euro(AveragePrice)),
            vjust = -0.85, color="black") +
  scale_y_continuous(breaks = seq(0, 350000, 5000),
                     label = euro) +
  scale_x_continuous(breaks = 2019:2021) +
  geom_point(size = 2,
             color = "steelblue")+
  labs(title = "2019-2021 Average house prices by year")+
  theme_ipsum()
```

Figure 10

Line graph of average house prices for the year 2019-2021



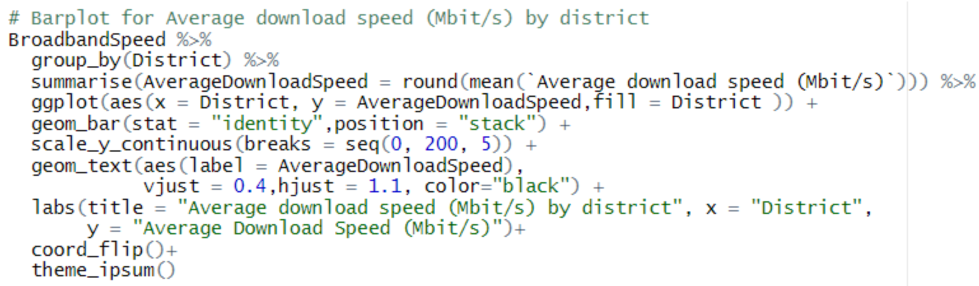
In 2019, 2020, and 2021, the average house price was €210,839, €228,342, and €245,996 respectively. The average price of a house increased gradually from 2019 to 2021 by about 8%.

Broadband Speed Visualization

Summarizing the average download speed data, the minimum speed is 1.10 Mbit/s, the lower quartile speed is 30.20 Mbit/s, the median is 48 Mbit/s, the average is 49.23 Mbit/s, the upper quartile is 67.80 Mbit/s, and the maximum value is 159.50 Mbit/s.

Figure 11

Code for boxplot of average broadband speed by district



```
# Barplot for Average download speed (Mbit/s) by district
BroadbandSpeed %>%
  group_by(District) %>%
  summarise(AverageDownloadSpeed = round(mean(`Average download speed (Mbit/s)`))) %>%
  ggplot(aes(x = District, y = AverageDownloadSpeed,fill = District )) +
  geom_bar(stat = "identity",position = "stack") +
  scale_y_continuous(breaks = seq(0, 200, 5)) +
  geom_text(aes(label = AverageDownloadSpeed),
            vjust = 0.4,hjust = 1.1, color="black") +
  labs(title = "Average download speed (Mbit/s) by district", x = "District",
       y = "Average Download Speed (Mbit/s)")+
  coord_flip()+
  theme_ipsum()
```

Figure 12

Boxplot of average download speed by district

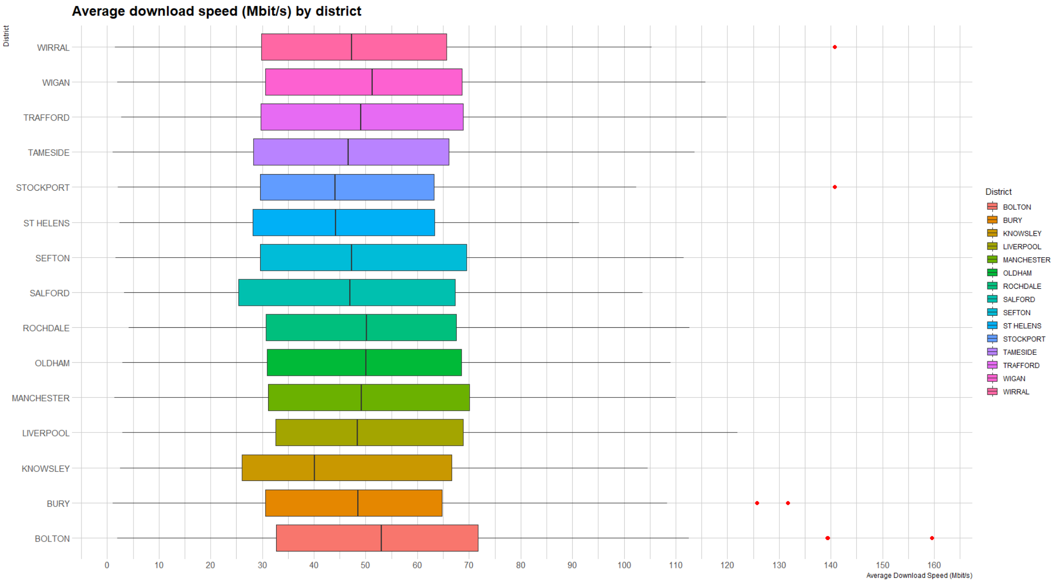
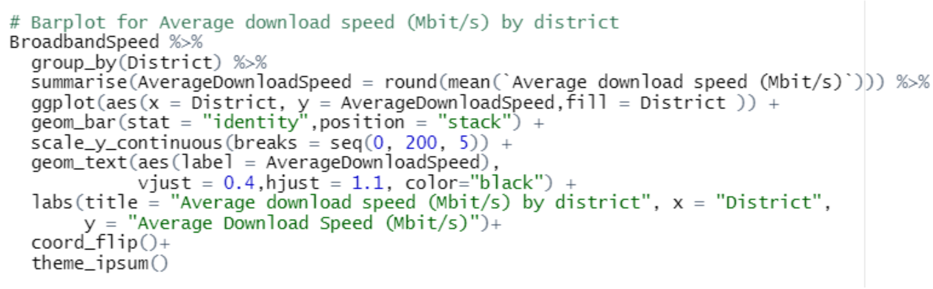


Figure 13

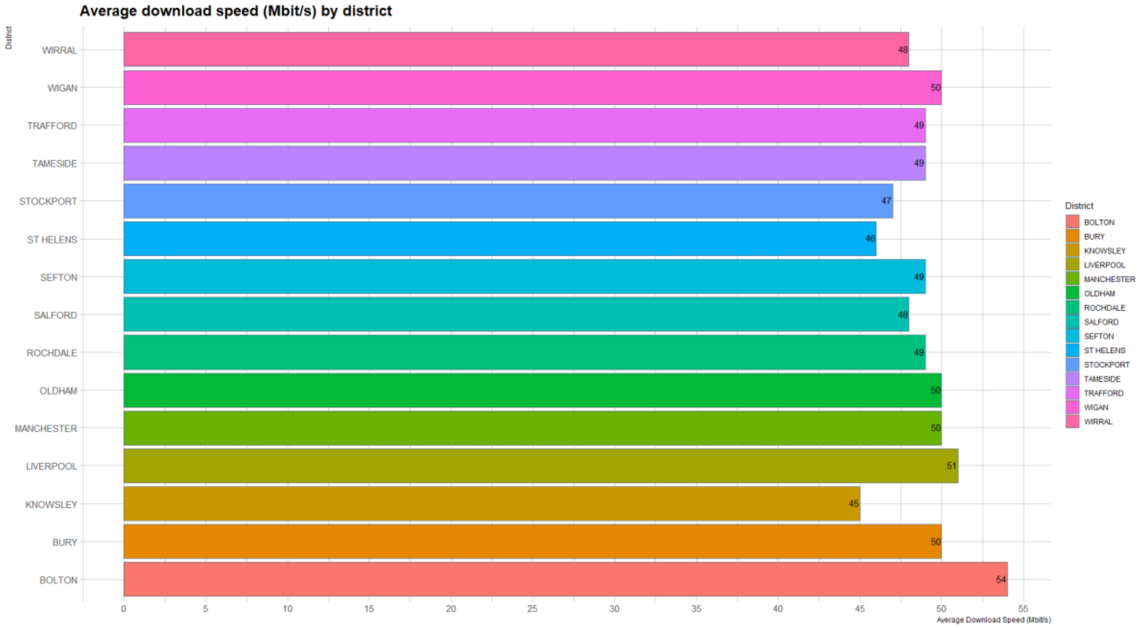
Code for bar graph of average broadband speed by district



```
# Barplot for Average download speed (Mbit/s) by district
BroadbandSpeed %>%
  group_by(District) %>%
  summarise(AverageDownloadSpeed = round(mean(`Average download speed (Mbit/s)`))) %>%
  ggplot(aes(x = District, y = AverageDownloadSpeed,fill = District )) +
  geom_bar(stat = "identity",position = "stack") +
  scale_y_continuous(breaks = seq(0, 200, 5)) +
  geom_text(aes(label = AverageDownloadSpeed),
            vjust = 0.4,hjust = 1.1, color="black") +
  labs(title = "Average download speed (Mbit/s) by district", x = "District",
       y = "Average Download Speed (Mbit/s)")+
  coord_flip()+
  theme_ipsum()
```

Figure 14

Bar graph of average broadband speed by district



Basic observations that could be drawn from the graphs include:

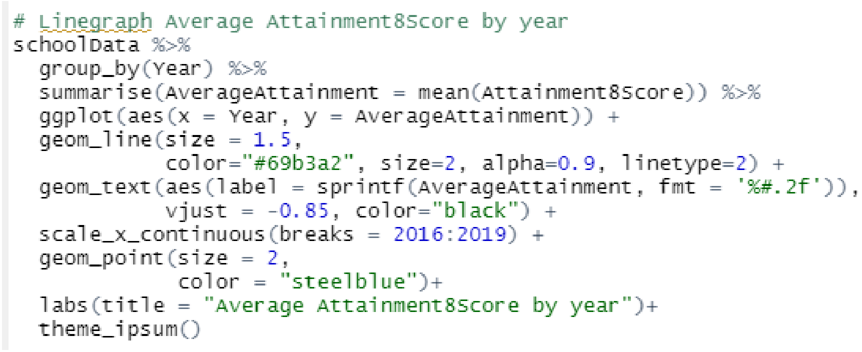
Bolton has the highest upper quartile value of 72 Mbit/s and the highest average download speed of 54 Mbit/s, while Knowsley has the lowest average download speed of 45 Mbit/s. The upper quartile value for the majority of districts is above 65 Mbit/s.

School Data Attainment8score Visualization

Attainment8Score has been visualized using Boxplot and Line graph

Figure 15

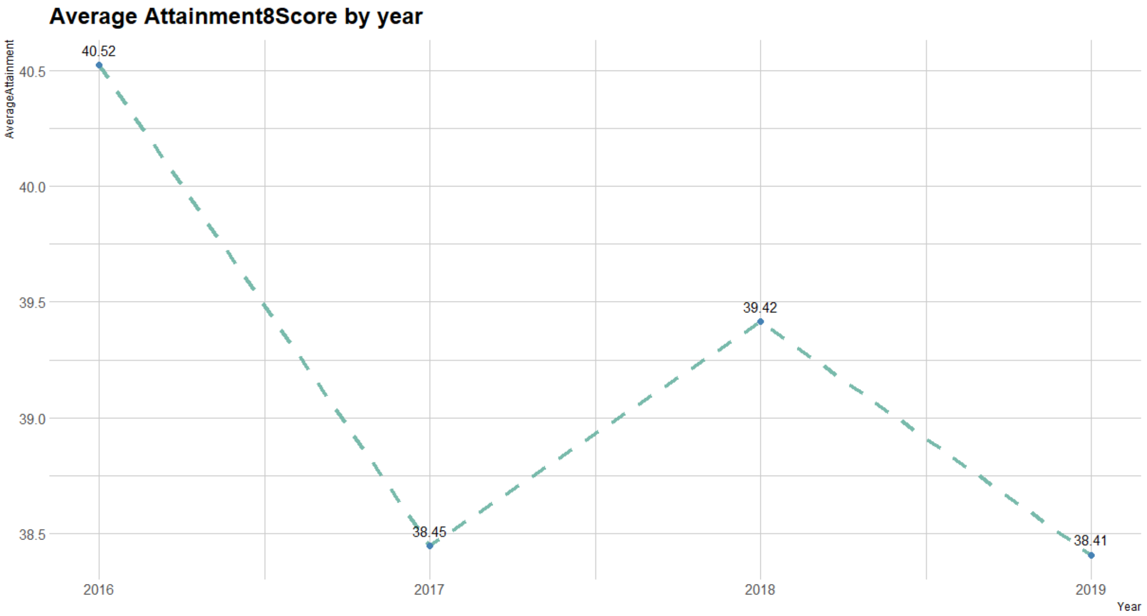
Code for line graph of average attainment8score by year



```
# Linegraph Average Attainment8Score by year
schoolData %>%
  group_by(Year) %>%
  summarise(AverageAttainment = mean(Attainment8Score)) %>%
  ggplot(aes(x = Year, y = AverageAttainment)) +
  geom_line(size = 1.5,
            color="#69b3a2", size=2, alpha=0.9, linetype=2) +
  geom_text(aes(label = sprintf(AverageAttainment, fmt = '%#.2f')),
            vjust = -0.85, color="black") +
  scale_x_continuous(breaks = 2016:2019) +
  geom_point(size = 2,
             color = "steelblue")+
  labs(title = "Average Attainment8Score by year")+
  theme_ipsum()
```

Figure 16

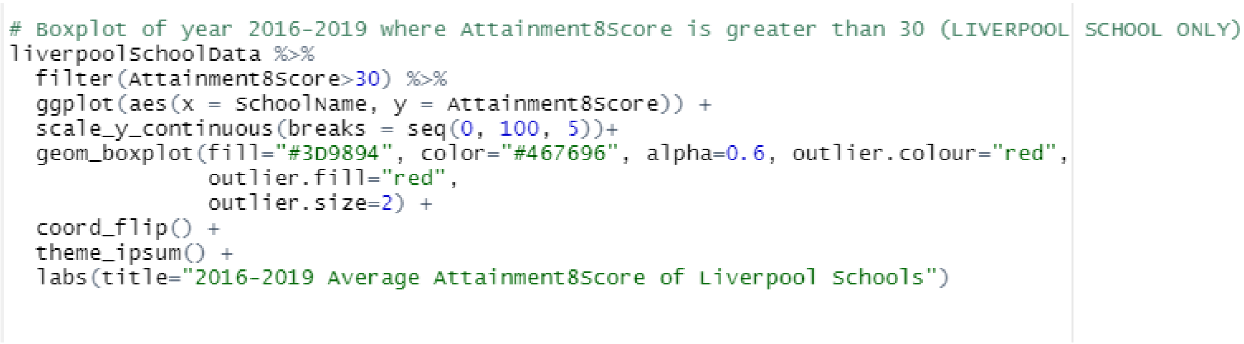
Line graph of average attainment8score by year



According to the line graph, the average attainment8score was 40.52 in 2016, fell to 38.45 in 2017, increased to 39.40 in 2018, and then fell back to 38.41 in 2019. The average score has varied by one or two points over the years.

Figure 17

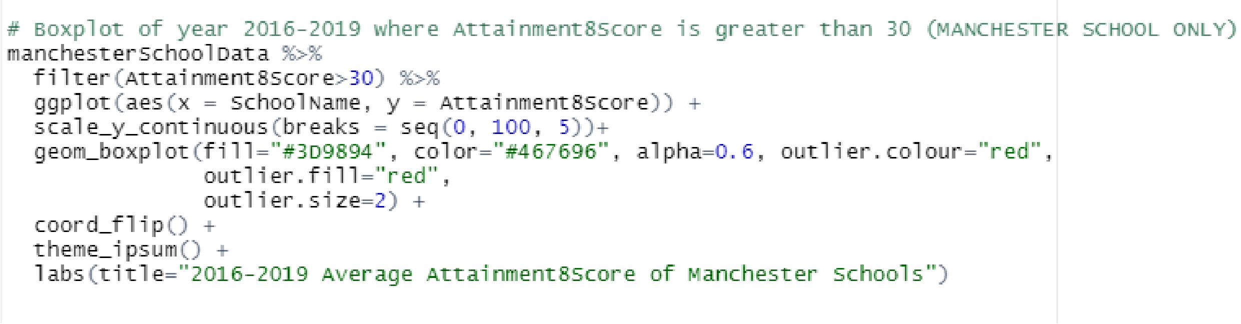
Code for boxplot of 2016-2019 average attainment8score of Liverpool Schools



```
# Boxplot of year 2016-2019 where Attainment8Score is greater than 30 (LIVERPOOL SCHOOL ONLY)
liverpoolSchoolData %>%
  filter(Attainment8Score>30) %>%
  ggplot(aes(x = SchoolName, y = Attainment8Score)) +
  scale_y_continuous(breaks = seq(0, 100, 5))+
  geom_boxplot(fill="#3D9894", color="#467696", alpha=0.6, outlier.colour="red",
               outlier.fill="red",
               outlier.size=2) +
  coord_flip() +
  theme_ipsum() +
  labs(title="2016-2019 Average Attainment8Score of Liverpool Schools")
```

Figure 18

Code for boxplot of 2016-2019 average attainment8score of Manchester Schools



```
# Boxplot of year 2016-2019 where Attainment8Score is greater than 30 (MANCHESTER SCHOOL ONLY)
manchesterSchoolData %>%
  filter(Attainment8Score>30) %>%
  ggplot(aes(x = SchoolName, y = Attainment8Score)) +
  scale_y_continuous(breaks = seq(0, 100, 5))+
  geom_boxplot(fill="#3D9894", color="#467696", alpha=0.6, outlier.colour="red",
               outlier.fill="red",
               outlier.size=2) +
  coord_flip() +
  theme_ipsum() +
  labs(title="2016-2019 Average Attainment8Score of Manchester Schools")
```

It was challenging to understand the boxplot because there were so many schools. Therefore, the boxplot has been constructed separately for Liverpool and Manchester Schools and the attainment8score greater than 30 filter has been applied.

Figure 19

Boxplot of 2016-2019 average attainment8score of Liverpool Schools

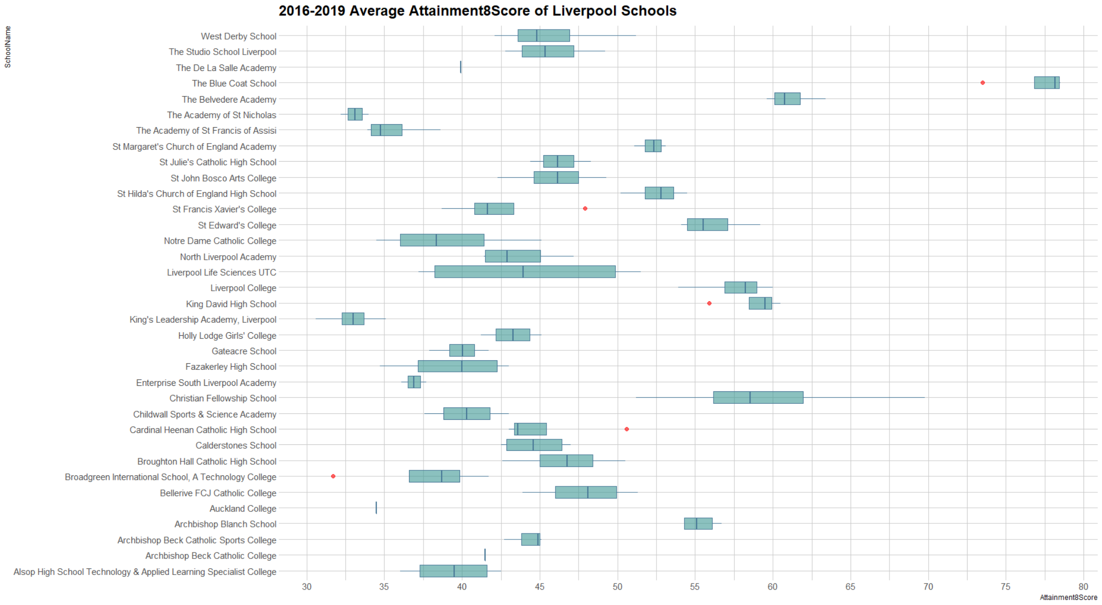
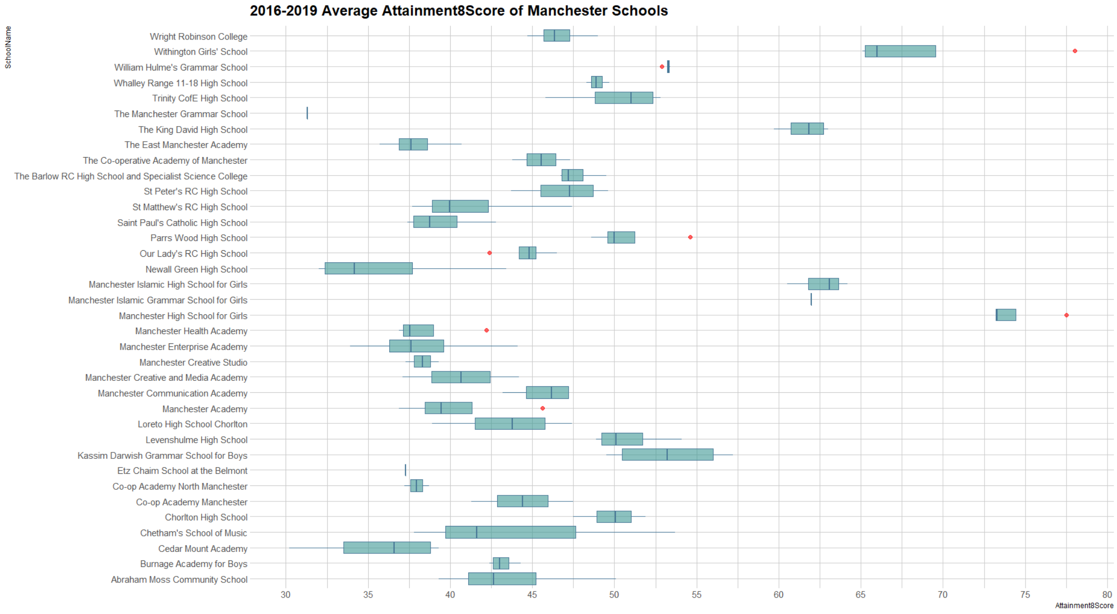


Figure 20

Boxplot of 2016-2019 average attainment8score of Liverpool Schools



Basic observations that could be drawn from the graphs include:

The majority of Liverpool's schools have an average attainment8score that falls between 35 and 50. Only The Blue Coat School and The Belvedere Academy have average scores that are higher than 60. The Blue Coat School is the top-scoring school in Manchester and Liverpool with an average attainment8score of 77.

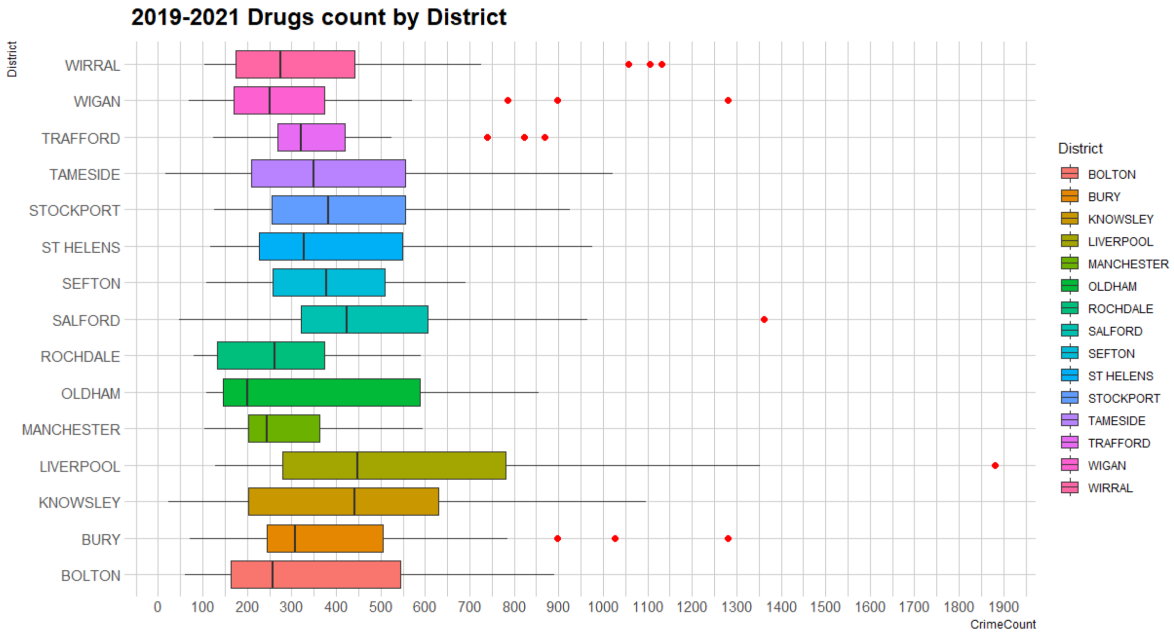
Four schools in Manchester have scores more than 60, and three of those four are girls' schools. The highest-scoring school in Manchester is Manchester High School for Girls, with an average score of 74.

Crime Data Visualization

In case of crime data, Boxplot for drug offence rate, Pie chart for Robbery and Radar chart for Vehicle crime was used.

Figure 21

Boxplot for 2019-2021 Drugs count by District

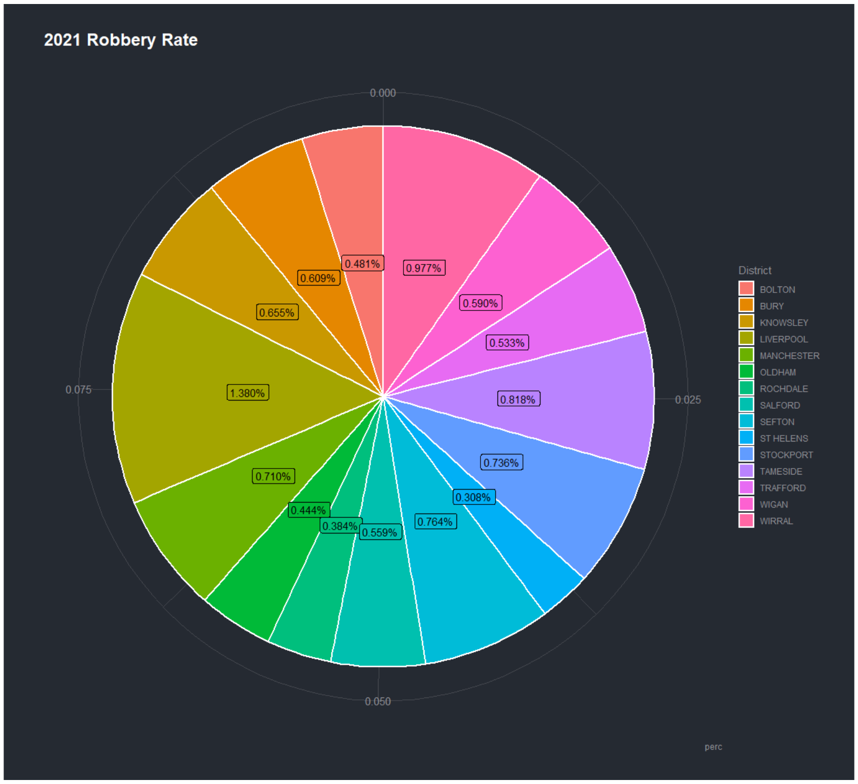


Basic observations that could be drawn from the graphs include:

Liverpool has the highest crime rate for drugs, whereas Oldham has the lowest. The average drug offence rate ranges is 337. Most of the districts have higher median rate than the overall average drug rate.

Figure 22

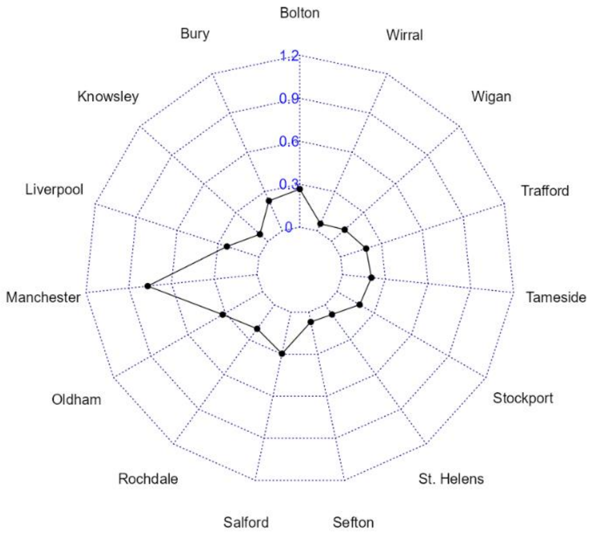
Pie chart for 2021 Robbery Rate



The Crime count was firstly converted in terms of percentage to make pie chart. As per the pie chart, it is clear that Liverpool has the highest robbery rate of 1.380% and St Helens has the lowest of 0.308% with 1.072% difference between the two. Most of the districts has robbery rate higher than 0.50% which is which is above average percentage of robbery rate.

Figure 23

Radar chart for vehicle crime rate



The radar chart indicates the vehicle crime rate; it is obvious that the Manchester district has a high rate of robbery while the other districts have a substantially lower rate of robbery compared to Manchester's. Sefton, along with Liverpool and Wigan has lowest vehicle crime rate.

LINEAR REGRESSION

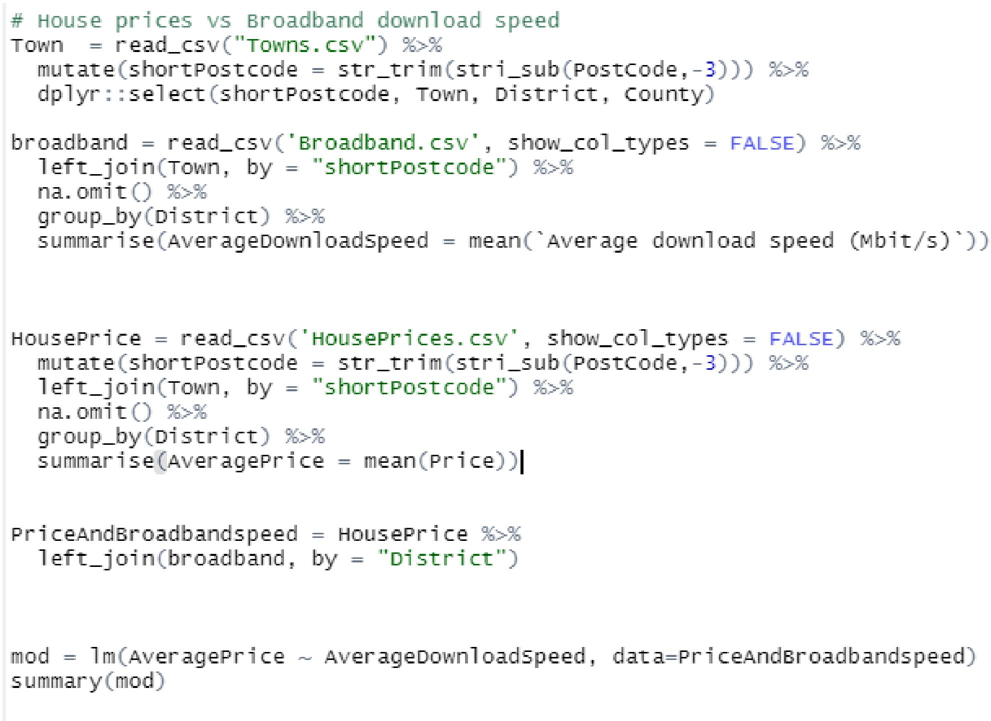
Linear regression is a commonly used predictive analysis method. The regression is done to examine the relationship between variables. Summary shows the residual which is a vertical distance between a data point and regression line. If the residuals are positive, they are above regression line, if negative they are below, if it is zero the regression line passes through the point (Regression? 2022).

Linear modeling has been done for the following variables:

House Prices and Broadband download Speed

Figure 24

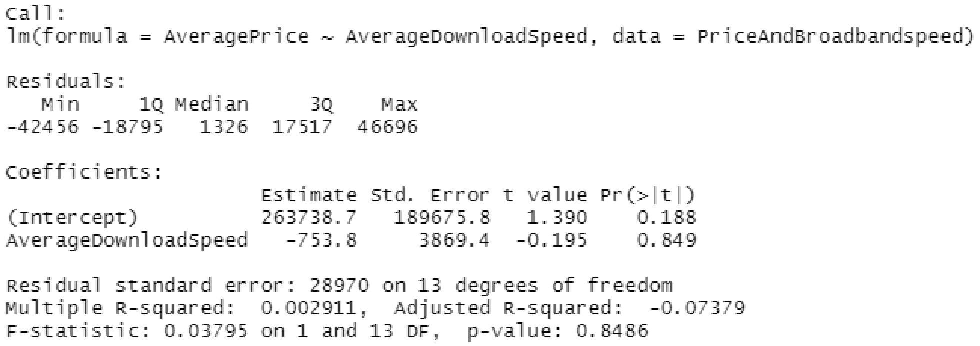
Code for linear regression of house price vs download speed



```
# House prices vs Broadband download speed
Town  = read_csv("Towns.csv") %>%
  mutate(shortPostcode = str_trim(stri_sub(PostCode,-3))) %>%
  dplyr::select(shortPostcode, Town, District, County)

broadband = read_csv('Broadband.csv', show_col_types = FALSE) %>%
  left_join(Town, by = "shortPostcode") %>%
  na.omit() %>%
  group_by(District) %>%
  summarise(AverageDownloadSpeed = mean(`Average download speed (Mbit/s)`))


HousePrice = read_csv('HousePrices.csv', show_col_types = FALSE) %>%
  mutate(shortPostcode = str_trim(stri_sub(PostCode,-3))) %>%
  left_join(Town, by = "shortPostcode") %>%
  na.omit() %>%
  group_by(District) %>%
  summarise(AveragePrice = mean(Price))


PriceAndBroadbandspeed = HousePrice %>%
  left_join(broadband, by = "District")


mod = lm(AveragePrice ~ AverageDownloadSpeed, data=PriceAndBroadbandspeed)
summary(mod)
```

Figure 25

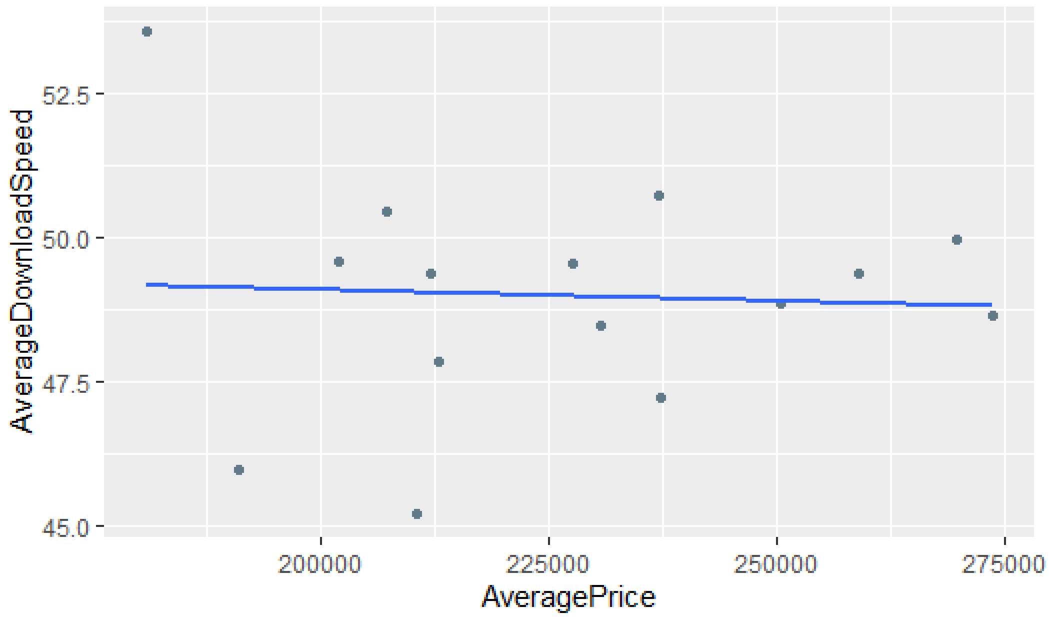
Summary for house price vs download speed



```
Call:
lm(formula = AveragePrice ~ AverageDownloadSpeed, data = PriceAndBroadbandspeed)

Residuals:
   Min     1Q Median     3Q    Max 
-42456 -18795   1326  17517  46696 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          263738.7   189675.8   1.390    0.188
AverageDownloadSpeed   -753.8     3869.4  -0.195    0.849

Residual standard error: 28970 on 13 degrees of freedom
Multiple R-squared:  0.002911,	Adjusted R-squared:  -0.07379 
F-statistic: 0.03795 on 1 and 13 DF,  p-value: 0.8486
```

Figure 26

House price vs download speed plot

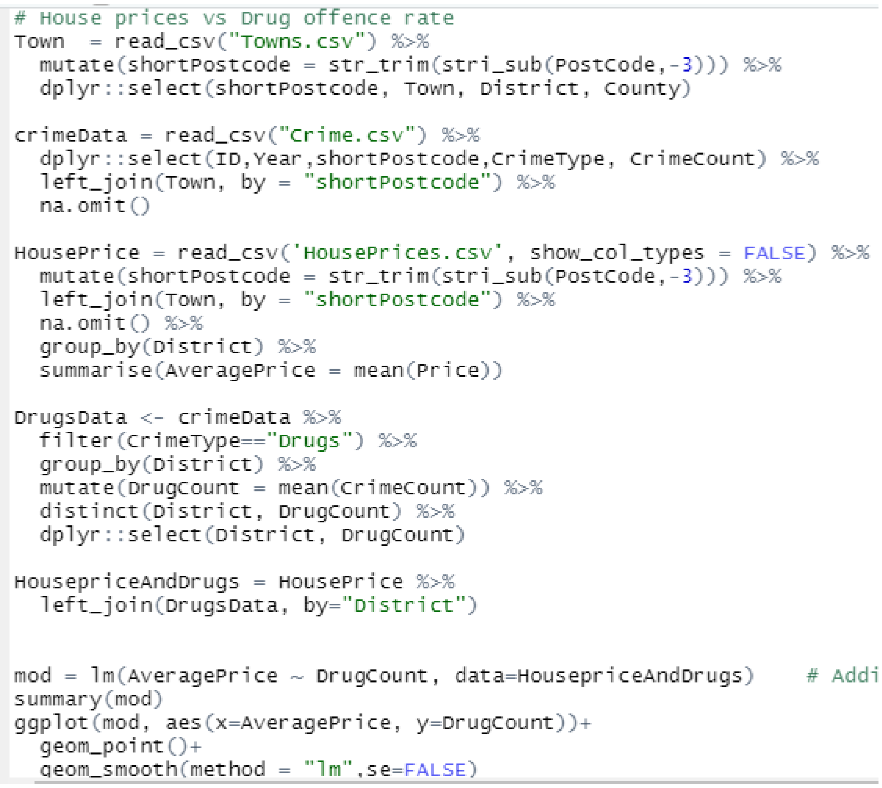


The summary shows that the R-squared value is 0.0029 which means the relation between the variables is weak and the variables have small effect to corresponding variable. The median, upper quartile and max value are above the regression line and minimum and lower quartile value are below.

House Prices and Drug offence rate

Figure 27

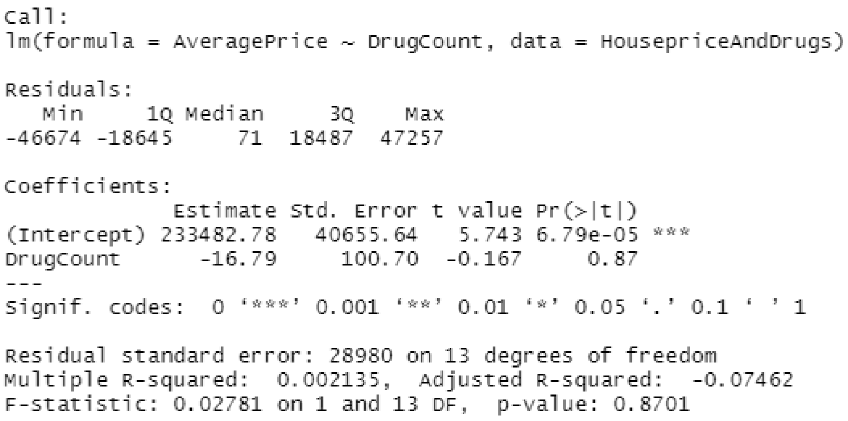
Code for House Prices vs Drug offence rate



```
# House prices vs Drug offence rate
Town  = read_csv("Towns.csv") %>%
  mutate(shortPostcode = str_trim(stri_sub(PostCode,-3))) %>%
  dplyr::select(shortPostcode, Town, District, County)

crimeData = read_csv("Crime.csv") %>%
  dplyr::select(ID,Year,shortPostcode,CrimeType, CrimeCount) %>%
  left_join(Town, by = "shortPostcode") %>%
  na.omit()

HousePrice = read_csv('HousePrices.csv', show_col_types = FALSE) %>%
  mutate(shortPostcode = str_trim(stri_sub(PostCode,-3))) %>%
  left_join(Town, by = "shortPostcode") %>%
  na.omit() %>%
  group_by(District) %>%
  summarise(AveragePrice = mean(Price))

DrugsData <- crimeData %>%
  filter(CrimeType=="Drugs") %>%
  group_by(District) %>%
  mutate(DrugCount = mean(CrimeCount)) %>%
  distinct(District, DrugCount) %>%
  dplyr::select(District, DrugCount)

HousepriceAndDrugs = HousePrice %>%
  left_join(DrugsData, by="District")


mod = lm(AveragePrice ~ DrugCount, data=HousepriceAndDrugs)    # Addi
summary(mod)
ggplot(mod, aes(x=AveragePrice, y=DrugCount))+
  geom_point()+
  geom_smooth(method = "lm",se=FALSE)
```

Figure 28

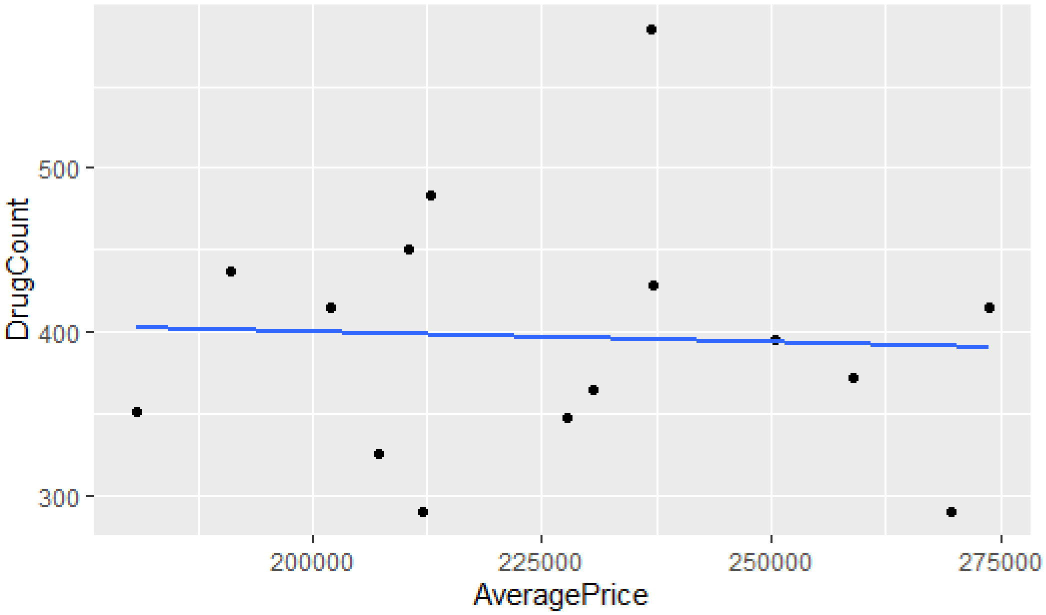
Summary for House Prices vs Drug offence rate



```
Call:
lm(formula = AveragePrice ~ DrugCount, data = HousepriceAndDrugs)

Residuals:
   Min     1Q Median     3Q    Max 
-46674 -18645     71  18487  47257 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 233482.78   40655.64   5.743 6.79e-05 ***
DrugCount      -16.79     100.70  -0.167     0.87    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 28980 on 13 degrees of freedom
Multiple R-squared:  0.002135,	Adjusted R-squared:  -0.07462 
F-statistic: 0.02781 on 1 and 13 DF,  p-value: 0.8701
```

Figure 29

House Prices vs Drug offence rate plot

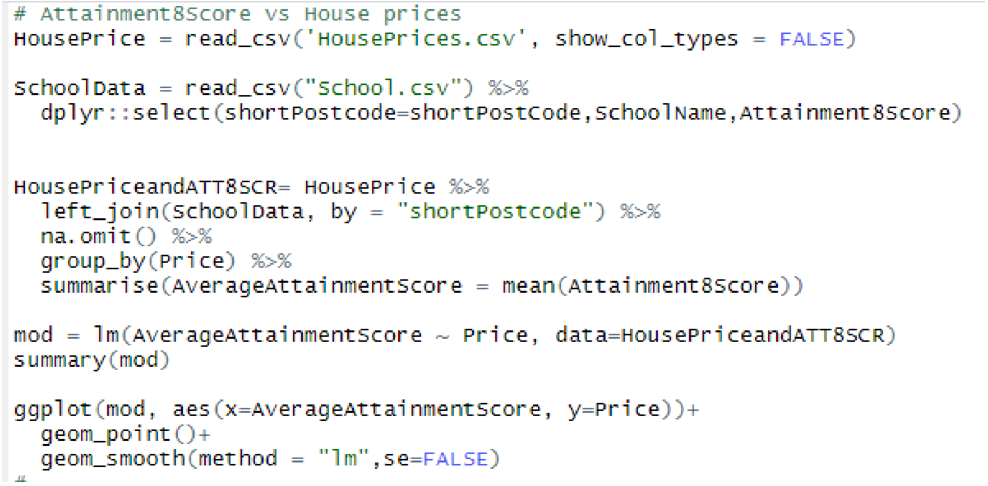


The model summarizes that the R-squared value is 0.0021 which means weak relation between the variable. The residuals median, upper quartile and max value are above the regression line and minimum and lower quartile value are below.

Attainment8score and House prices

Figure 30

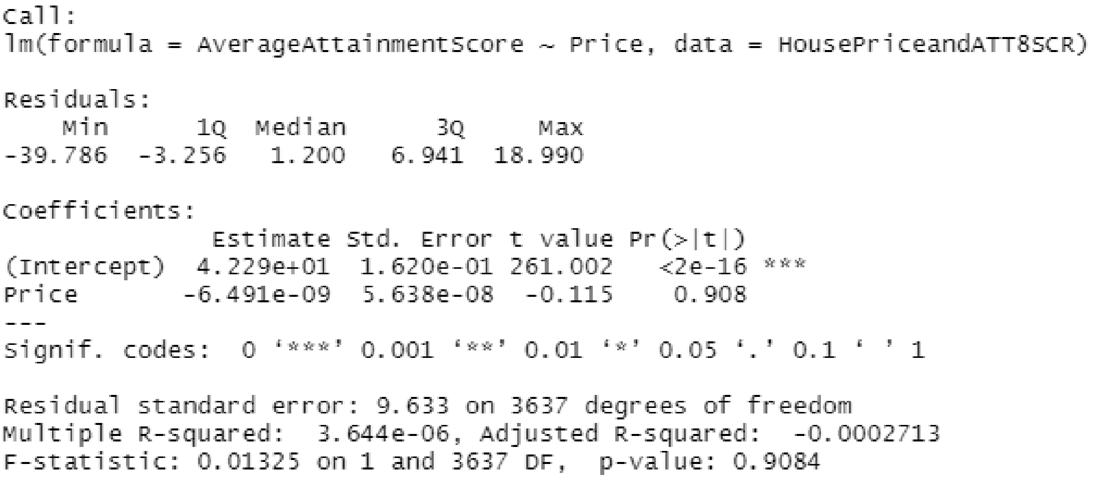
Code for Attainment8score vs House prices



```
# Attainment8Score vs House prices
HousePrice = read_csv('HousePrices.csv', show_col_types = FALSE)

SchoolData = read_csv("School.csv") %>%
  dplyr::select(shortPostcode=shortPostCode,SchoolName,Attainment8Score)


HousePriceandATT8SCR= HousePrice %>%
  left_join(SchoolData, by = "shortPostcode") %>%
  na.omit() %>%
  group_by(Price) %>%
  summarise(AverageAttainmentScore = mean(Attainment8Score))

mod = lm(AverageAttainmentScore ~ Price, data=HousePriceandATT8SCR)
summary(mod)

ggplot(mod, aes(x=AverageAttainmentScore, y=Price))+
  geom_point()+
  geom_smooth(method = "lm",se=FALSE)
```

Figure 31

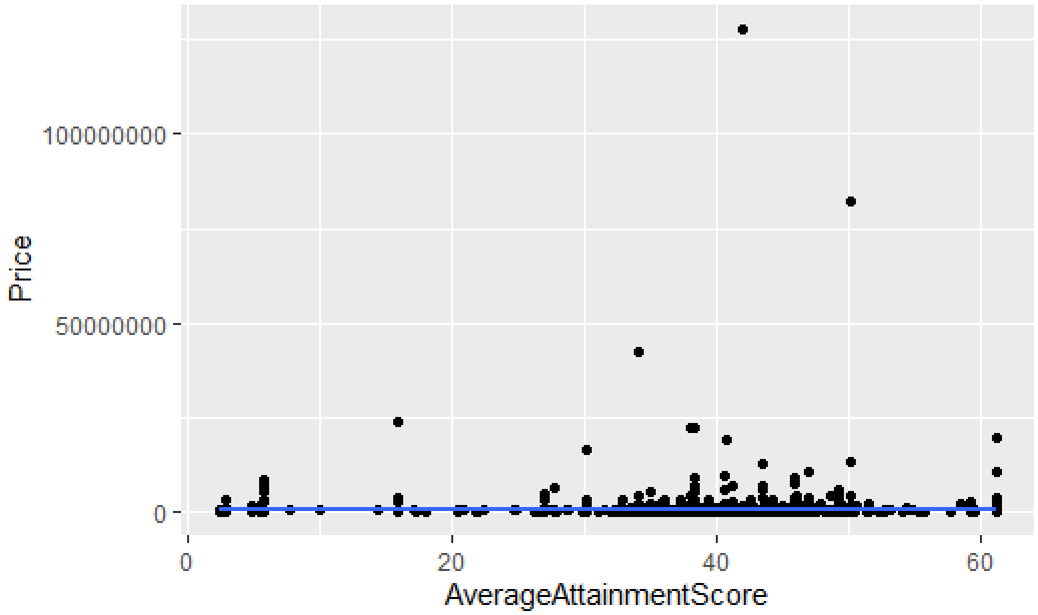
Summary for Attainment8score vs House prices



```
Call:
lm(formula = AverageAttainmentScore ~ Price, data = HousePriceandATT8SCR)

Residuals:
    Min      1Q  Median      3Q     Max 
-39.786  -3.256   1.200   6.941  18.990 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.229e+01  1.620e-01 261.002   <2e-16 ***
Price       -6.491e-09  5.638e-08  -0.115    0.908    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.633 on 3637 degrees of freedom
Multiple R-squared:  3.644e-06, Adjusted R-squared:  -0.0002713 
F-statistic: 0.01325 on 1 and 3637 DF,  p-value: 0.9084
```

Figure 32

Attainment8score vs House prices plot

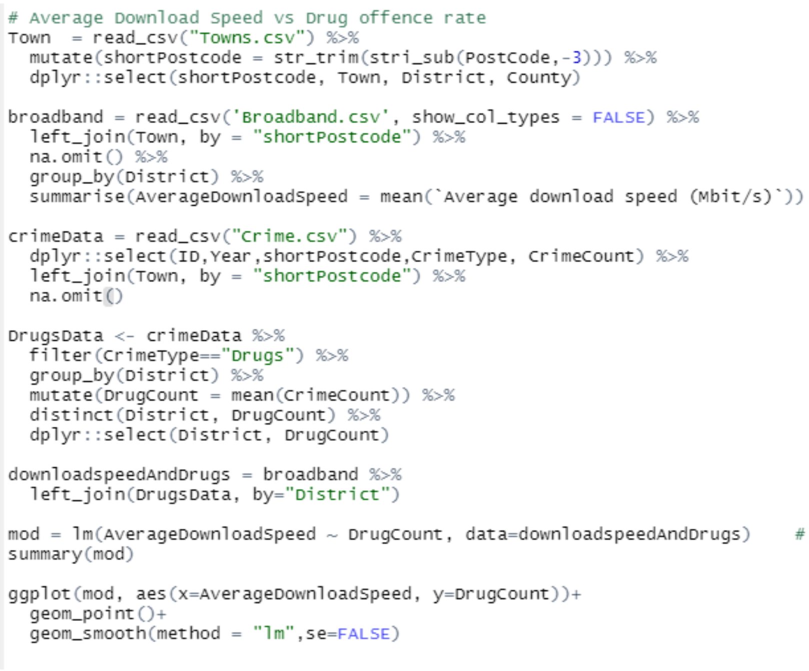


The model summarizes that the R-squared value is 0.00003 which means very weak relation between the variable.

Average Download Speed and drug offence rate

Figure 33

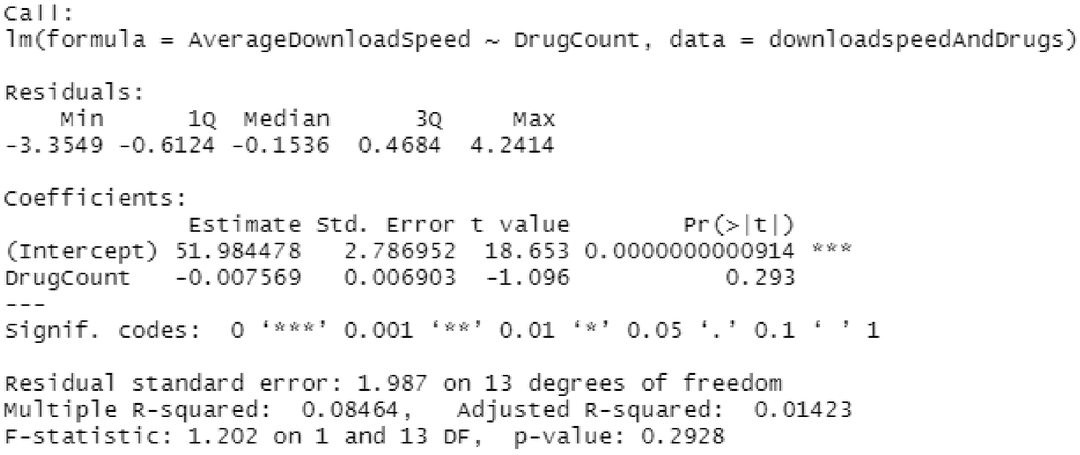
Code for average Download Speed vs drug offence rate



```
# Average Download Speed vs Drug offence rate
Town  = read_csv("Towns.csv") %>%
  mutate(shortPostcode = str_trim(stri_sub(PostCode,-3))) %>%
  dplyr::select(shortPostcode, Town, District, County)

broadband = read_csv('Broadband.csv', show_col_types = FALSE) %>%
  left_join(Town, by = "shortPostcode") %>%
  na.omit() %>%
  group_by(District) %>%
  summarise(AverageDownloadSpeed = mean(`Average download speed (Mbit/s)`))

crimeData = read_csv("Crime.csv") %>%
  dplyr::select(ID,Year,shortPostcode,CrimeType, CrimeCount) %>%
  left_join(Town, by = "shortPostcode") %>%
  na.omit()

DrugsData <- crimeData %>%
  filter(CrimeType=="Drugs") %>%
  group_by(District) %>%
  mutate(DrugCount = mean(CrimeCount)) %>%
  distinct(District, DrugCount) %>%
  dplyr::select(District, DrugCount)

downloadspeedAndDrugs = broadband %>%
  left_join(DrugsData, by="District")

mod = lm(AverageDownloadSpeed ~ DrugCount, data=downloadspeedAndDrugs)    #
summary(mod)

ggplot(mod, aes(x=AverageDownloadSpeed, y=DrugCount))+
  geom_point()+
  geom_smooth(method = "lm",se=FALSE)
```

Figure 34

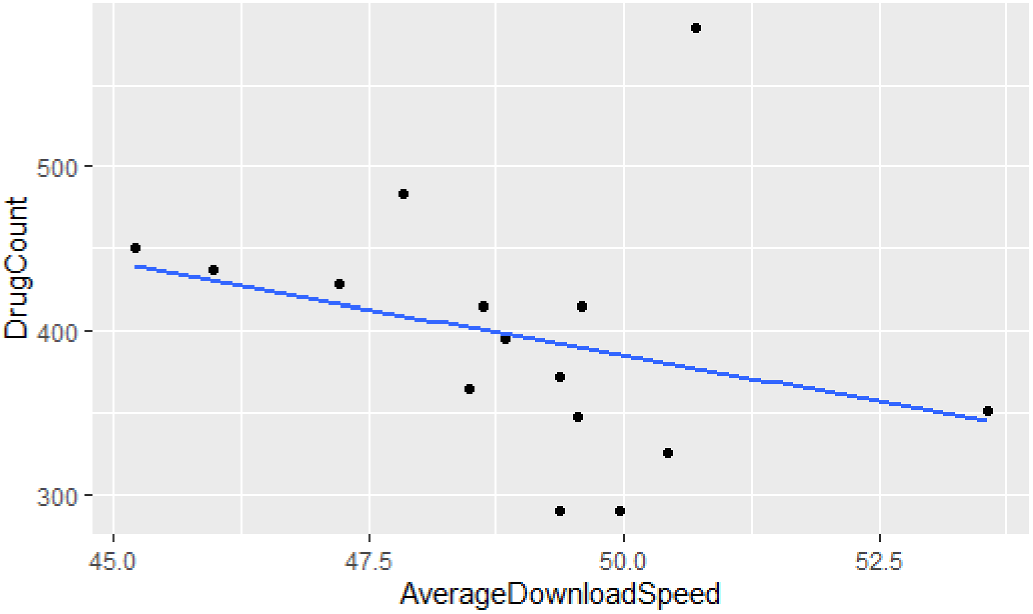
Summary for average Download Speed vs drug offence rate



```
Call:
lm(formula = AverageDownloadSpeed ~ DrugCount, data = downloadspeedAndDrugs)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3549 -0.6124 -0.1536  0.4684  4.2414 

Coefficients:
             Estimate Std. Error t value        Pr(>|t|)    
(Intercept) 51.984478   2.786952  18.653 0.0000000000914 ***
DrugCount   -0.007569   0.006903  -1.096           0.293    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.987 on 13 degrees of freedom
Multiple R-squared:  0.08464,	Adjusted R-squared:  0.01423 
F-statistic: 1.202 on 1 and 13 DF,  p-value: 0.2928
```

Figure 35

Average Download Speed vs drug offence rate plot

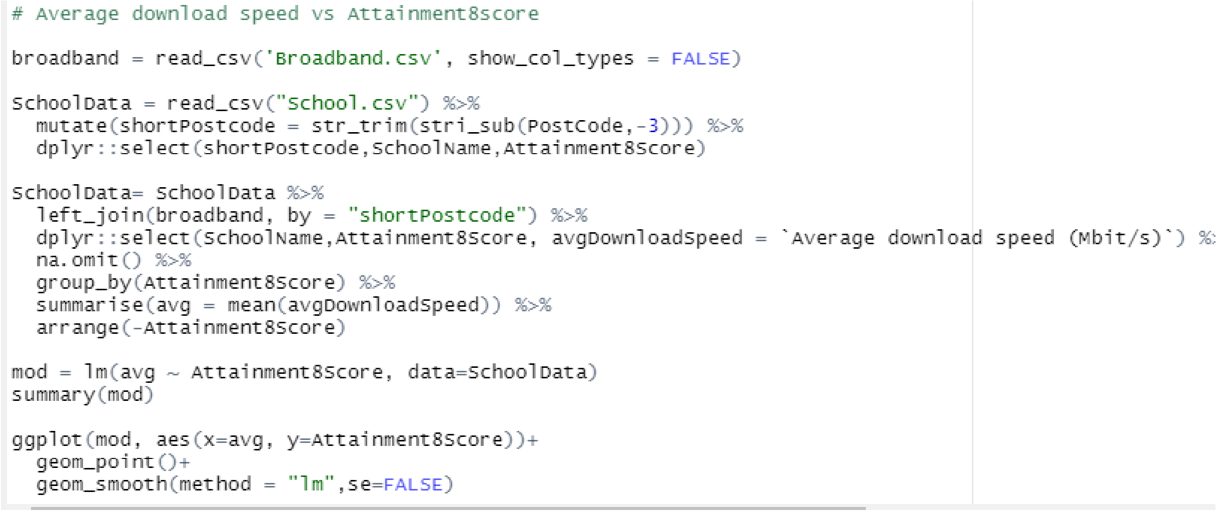


The R-squared value for average download speed and drug rate is 0.084 which represents weak relation between the variable and model doesn’t explain much of variation of data. The residuals upper quartile and max value are above the regression line and median, minimum, and lower quartile value are below.

Average download speed and Attainment8score

Figure 36

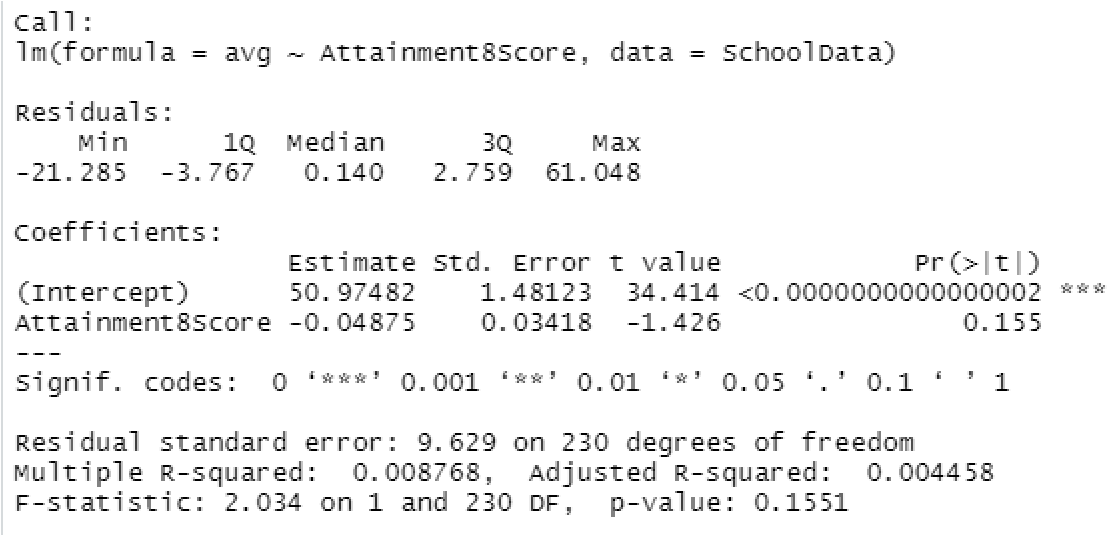
Code for Average download speed vs Attainment8score



```
# Average download speed vs Attainment8score

broadband = read_csv('Broadband.csv', show_col_types = FALSE)

SchoolData = read_csv("School.csv") %>%
  mutate(shortPostcode = str_trim(stri_sub(PostCode,-3))) %>%
  dplyr::select(shortPostcode,SchoolName,Attainment8Score)

SchoolData= SchoolData %>%
  left_join(broadband, by = "shortPostcode") %>%
  dplyr::select(SchoolName,Attainment8Score, avgDownloadSpeed = `Average download speed (Mbit/s)`) %
  na.omit() %>%
  group_by(Attainment8Score) %>%
  summarise(avg = mean(avgDownloadSpeed)) %>%
  arrange(-Attainment8Score)

mod = lm(avg ~ Attainment8Score, data=SchoolData)
summary(mod)

ggplot(mod, aes(x=avg, y=Attainment8Score))+
  geom_point()+
  geom_smooth(method = "lm",se=FALSE)
```

Figure 37

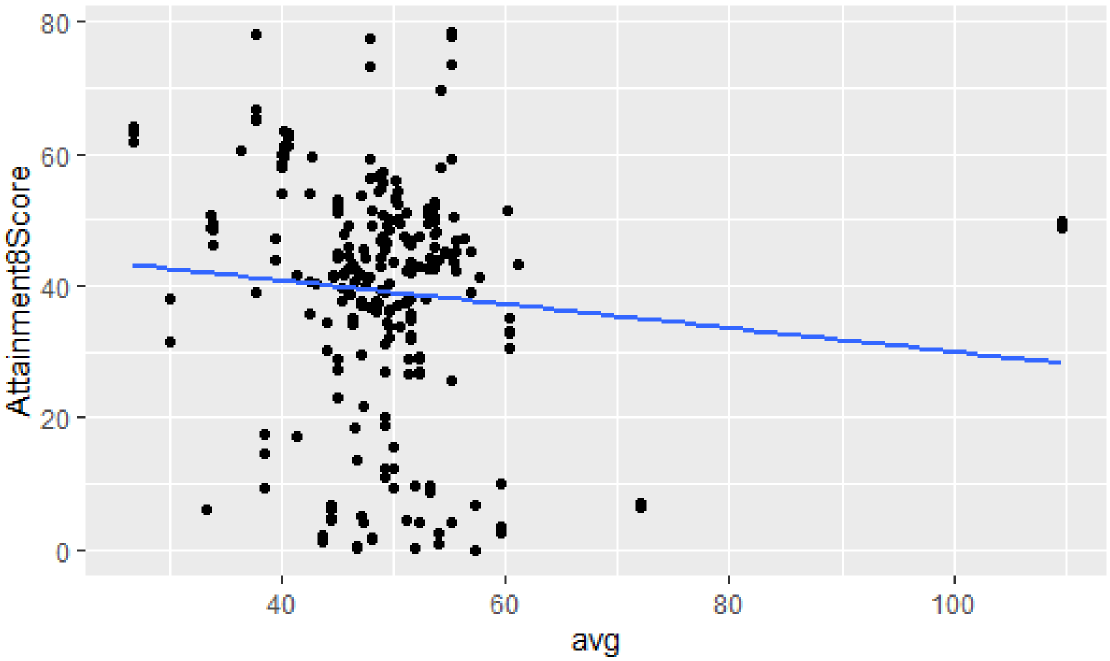
Summary for Average download speed vs Attainment8score



```
Call:
lm(formula = avg ~ Attainment8Score, data = SchoolData)

Residuals:
    Min      1Q  Median      3Q     Max 
-21.285  -3.767   0.140   2.759  61.048 

Coefficients:
                 Estimate Std. Error t value             Pr(>|t|)    
(Intercept)      50.97482    1.48123  34.414 <0.0000000000000002 ***
Attainment8Score -0.04875    0.03418  -1.426               0.155    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.629 on 230 degrees of freedom
Multiple R-squared:  0.008768,	Adjusted R-squared:  0.004458 
F-statistic: 2.034 on 1 and 230 DF,  p-value: 0.1551
```

Figure 38

Average download speed vs Attainment8score plot



The R-squared value for average download speed and Attainment8score is 0.0087 which represents weak relation between the variable and model doesn't explain much of variation of data. The residuals median, upper quartile and max value are above the regression line and minimum and lower quartile value are below.

RECOMMENDATION SYSTEM

Ranking System

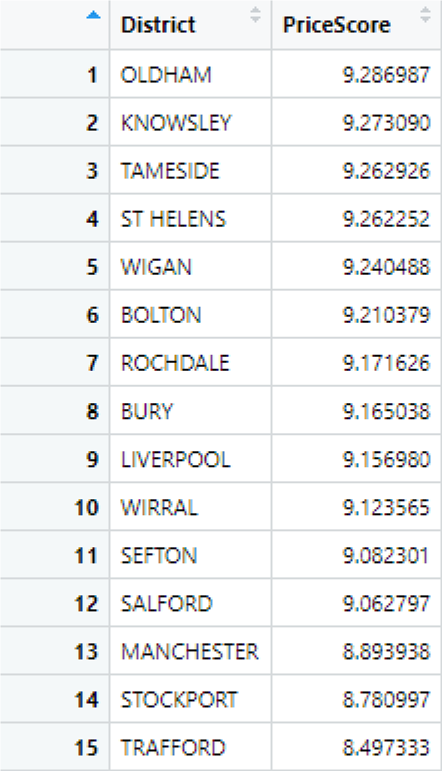
Firstly, the ranking for each component has been done separately in terms of average value grouped by district. The following are the ranking for each components:

House Price Ranking

The average housing prices in each district are used to calculate the scores. The districts with lower house prices scored higher than the districts with high prices. Oldham has an affordable average price of €174,530 compared to other districts, and other four top-ranked districts have average house prices under €190,000.

Figure 39

House price ranking with price score



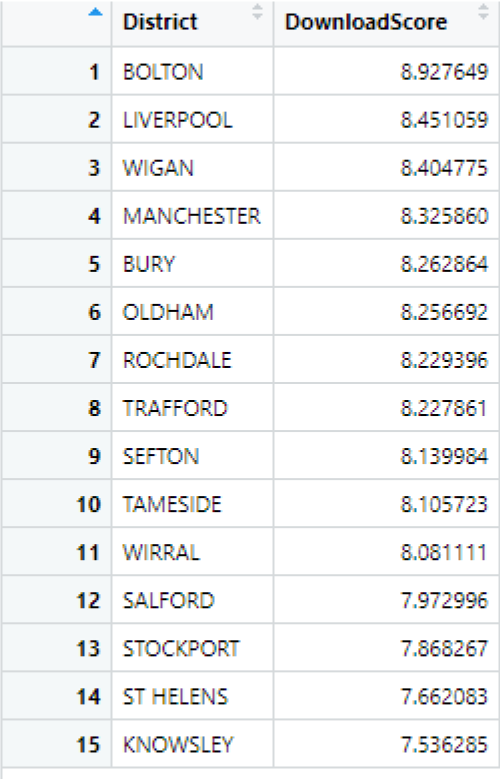
	District	PriceScore
1	OLDHAM	9.286987
2	KNOWSLEY	9.273090
3	TAMESIDE	9.262926
4	ST HELENS	9.262252
5	WIGAN	9.240488
6	BOLTON	9.210379
7	ROCHDALE	9.171626
8	BURY	9.165038
9	LIVERPOOL	9.156980
10	WIRRAL	9.123565
11	SEFTON	9.082301
12	SALFORD	9.062797
13	MANCHESTER	8.893938
14	STOCKPORT	8.780997
15	TRAFFORD	8.497333

Broadband Download Speed Ranking

Based on the average download speed, the download scores are calculated. The download score increases with the average download speed. Bolton, which is ranked first, has an average download speed of 53.6 Mbit/s, while the other two districts: Liverpool and Wigan have download speeds above 50 Mbit/s.

Figure 40

Broadband download speed with download score



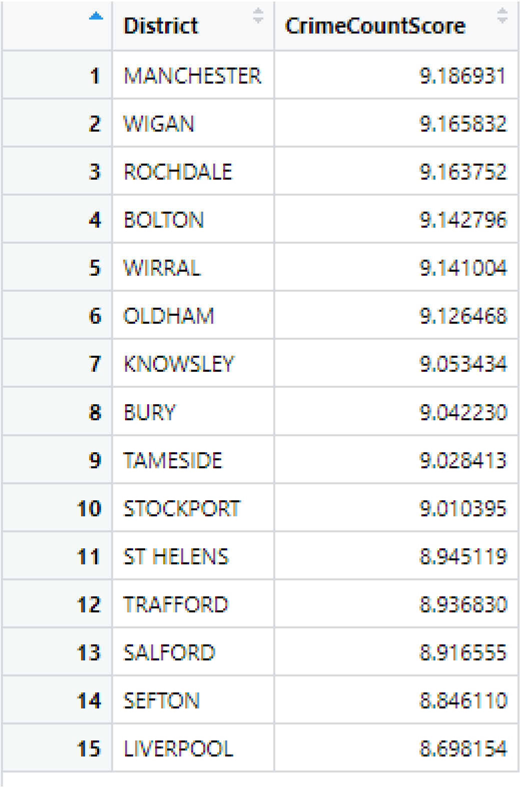
	District	DownloadScore
1	BOLTON	8.927649
2	LIVERPOOL	8.451059
3	WIGAN	8.404775
4	MANCHESTER	8.325860
5	BURY	8.262864
6	OLDHAM	8.256692
7	ROCHDALE	8.229396
8	TRAFFORD	8.227861
9	SEFTON	8.139984
10	TAMESIDE	8.105723
11	WIRRAL	8.081111
12	SALFORD	7.972996
13	STOCKPORT	7.868267
14	ST HELENS	7.662083
15	KNOWSLEY	7.536285

Crime Rate Ranking

The scores are calculated using the average crime rate. Districts that have lower average crime rates are given a higher ranking. Here, Manchester ranks top with a 9.18 score and Liverpool ranks last with an 8.69 score due to its lowest average crime count of 317 and highest average crime count of 509 respectively.

Figure 41

Crime rate ranking with crime score



	District	CrimeCountScore
1	MANCHESTER	9.186931
2	WIGAN	9.165832
3	ROCHDALE	9.163752
4	BOLTON	9.142796
5	WIRRAL	9.141004
6	OLDHAM	9.126468
7	KNOWSLEY	9.053434
8	BURY	9.042230
9	TAMESIDE	9.028413
10	STOCKPORT	9.010395
11	ST HELENS	8.945119
12	TRAFFORD	8.936830
13	SALFORD	8.916555
14	SEFTON	8.846110
15	LIVERPOOL	8.698154

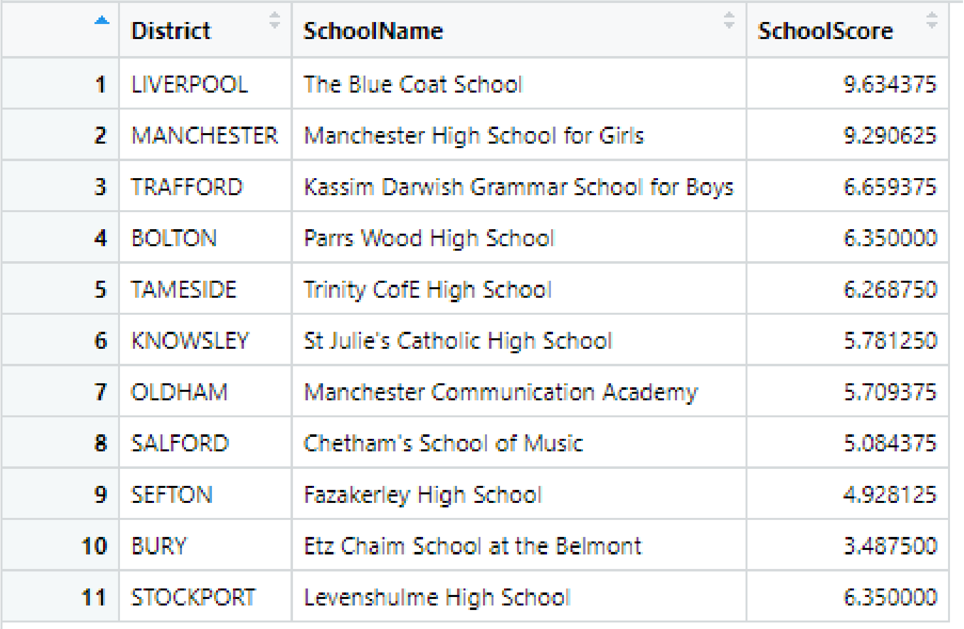
School Ranking

The average attainment8score of each school is used to determine the scores. There are several schools with varying scores within a district; the school with the highest score within each district is filtered and ranked. From the year 2016 to 2019, The Blue Coat School in Liverpool had an average attainment8score of 77 which makes it ranked first.

Due to some issues in short postcode of some district, the school located in a district was shown in another. To solve this issue, the district with issue and redundant school data was manually removed.

Figure 42

Crime rate ranking with crime score



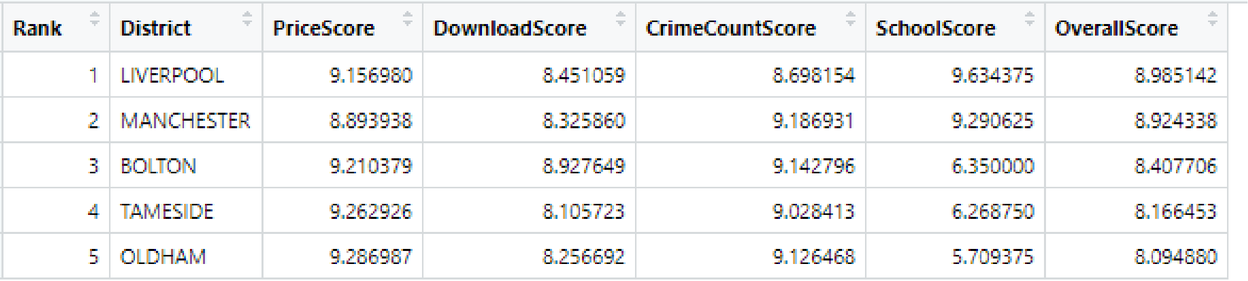
	District	SchoolName	SchoolScore
1	LIVERPOOL	The Blue Coat School	9.634375
2	MANCHESTER	Manchester High School for Girls	9.290625
3	TRAFFORD	Kassim Darwish Grammar School for Boys	6.659375
4	BOLTON	Parrs Wood High School	6.350000
5	TAMESIDE	Trinity CofE High School	6.268750
6	KNOWSLEY	St Julie's Catholic High School	5.781250
7	OLDHAM	Manchester Communication Academy	5.709375
8	SALFORD	Chetham's School of Music	5.084375
9	SEFTON	Fazakerley High School	4.928125
10	BURY	Etz Chaim School at the Belmont	3.487500
11	STOCKPORT	Levenshulme High School	6.350000

Overall Ranking Recommendation System

Rather than analyzing each aspect individually to discover the optimal location, an overall ranking recommendation system is developed that combines all of the different ranking datasets into a single ranking dataset and calculates the mean of all individual data scores for the overall score.

Figure 43

Overall ranking with overall score



Rank	District	PriceScore	DownloadScore	CrimeCountScore	SchoolScore	OverallScore
1	LIVERPOOL	9.156980	8.451059	8.698154	9.634375	8.985142
2	MANCHESTER	8.893938	8.325860	9.186931	9.290625	8.924338
3	BOLTON	9.210379	8.927649	9.142796	6.350000	8.407706
4	TAMESIDE	9.262926	8.105723	9.028413	6.268750	8.166453
5	OLDHAM	9.286987	8.256692	9.126468	5.709375	8.094880

Explanation of Ranking

The top five ranked districts do not have a substantial score difference because the average scores for each component do not differ much, and they have good scores in comparison to other low-ranking districts. Liverpool received an overall grade of 8.98, with individual component grades of 9.15 for house price, 8.45 for download speed, 8.69 for crime rate, and 9.63 for school. The difference in scores between Liverpool and Oldham is around 0.90, which is significantly higher than the difference in scores between other districts and Liverpool. This is likely due to the significantly lower school score of 5.70, than the scores of Liverpool and Manchester, both of which are higher than 9. Oldham is ranked among the top five because it has the lowest average housing price resulting in highest score of 9.28 and among all districts.

Although Liverpool ranks last with the lowest score due to having the greatest crime rate, other aspects allowed it to move up the overall rankings. For example, Liverpool ranked first in the school rankings with a score of 9.63, which is significantly higher than the majority of schools in other districts, second in broadband speed ranking with a score of 8.45, which is behind Bolton, who ranked first with a score of 8.92, and third in house price with a score of 9.15.

Similarly, Manchester came in second with an overall score of 8.29, Bolton came in third with 8.40, Tameside came in fourth with 8.16, and Oldham came in fifth with 8.09. Apart from the house price, Manchester has a very good overall score, making it an excellent option to Liverpool, which has the highest crime rate. All of the ranked districts are a great choice for accommodation in terms of house pricing, internet download speed, crime rate, school score.

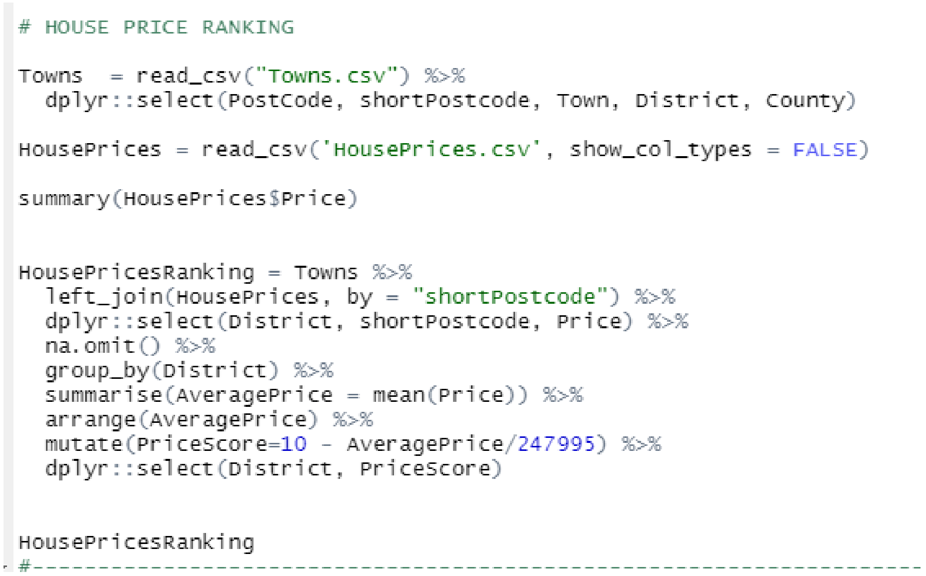
The tasks performed in this assignment certainly achieves its goal as per the scenario which is to make recommendations for appropriate places based on housing pricing, broadband speed, and local crime.

ANNEX 1

Code Screenshot

Figure 44

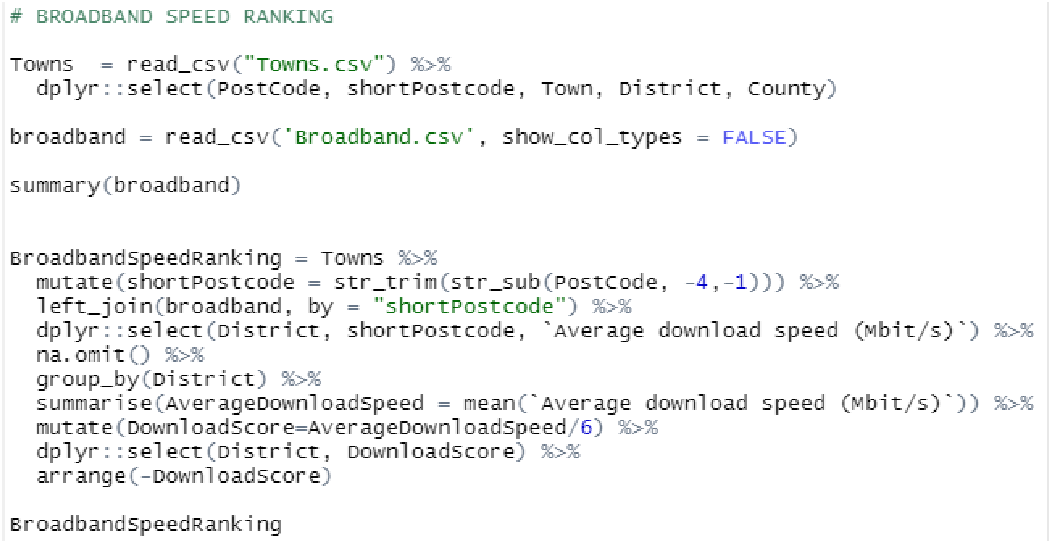
Code for house prices ranking



```
# HOUSE PRICE RANKING

Towns  = read_csv("Towns.csv") %>%
  dplyr::select(PostCode, shortPostcode, Town, District, County)

HousePrices = read_csv('HousePrices.csv', show_col_types = FALSE)

summary(HousePrices$Price)


HousePricesRanking = Towns %>%
  left_join(HousePrices, by = "shortPostcode") %>%
  dplyr::select(District, shortPostcode, Price) %>%
  na.omit() %>%
  group_by(District) %>%
  summarise(AveragePrice = mean(Price)) %>%
  arrange(AveragePrice) %>%
  mutate(PriceScore=10 - AveragePrice/247995) %>%
  dplyr::select(District, PriceScore)


HousePricesRanking
#-------------------------------------------------------------------
```

Figure 45

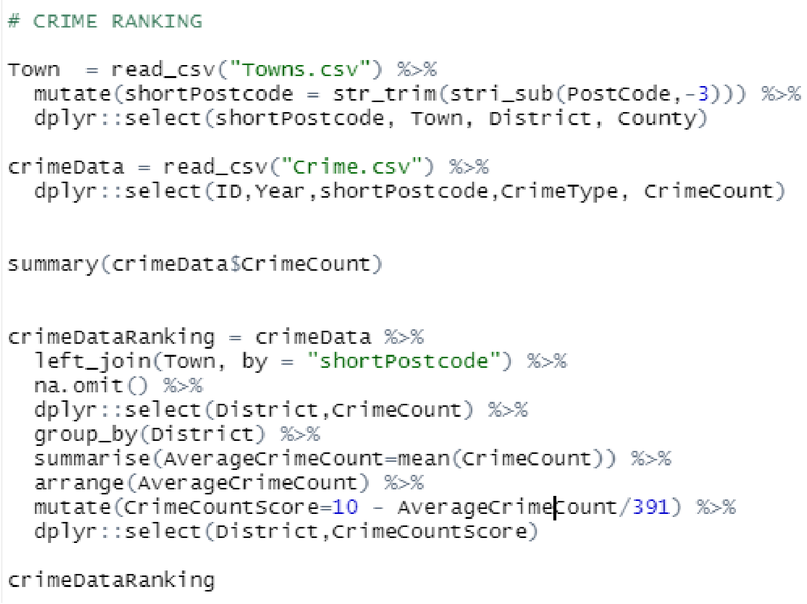
Code for Broadband speed ranking



```
# BROADBAND SPEED RANKING

Towns  = read_csv("Towns.csv") %>%
  dplyr::select(PostCode, shortPostcode, Town, District, County)

broadband = read_csv('Broadband.csv', show_col_types = FALSE)

summary(broadband)


BroadbandSpeedRanking = Towns %>%
  mutate(shortPostcode = str_trim(str_sub(PostCode, -4,-1))) %>%
  left_join(broadband, by = "shortPostcode") %>%
  dplyr::select(District, shortPostcode, `Average download speed (Mbit/s)`) %>%
  na.omit() %>%
  group_by(District) %>%
  summarise(AverageDownloadSpeed = mean(`Average download speed (Mbit/s)`)) %>%
  mutate(DownloadScore=AverageDownloadSpeed/6) %>%
  dplyr::select(District, DownloadScore) %>%
  arrange(-DownloadScore)

BroadbandSpeedRanking
```

Figure 46

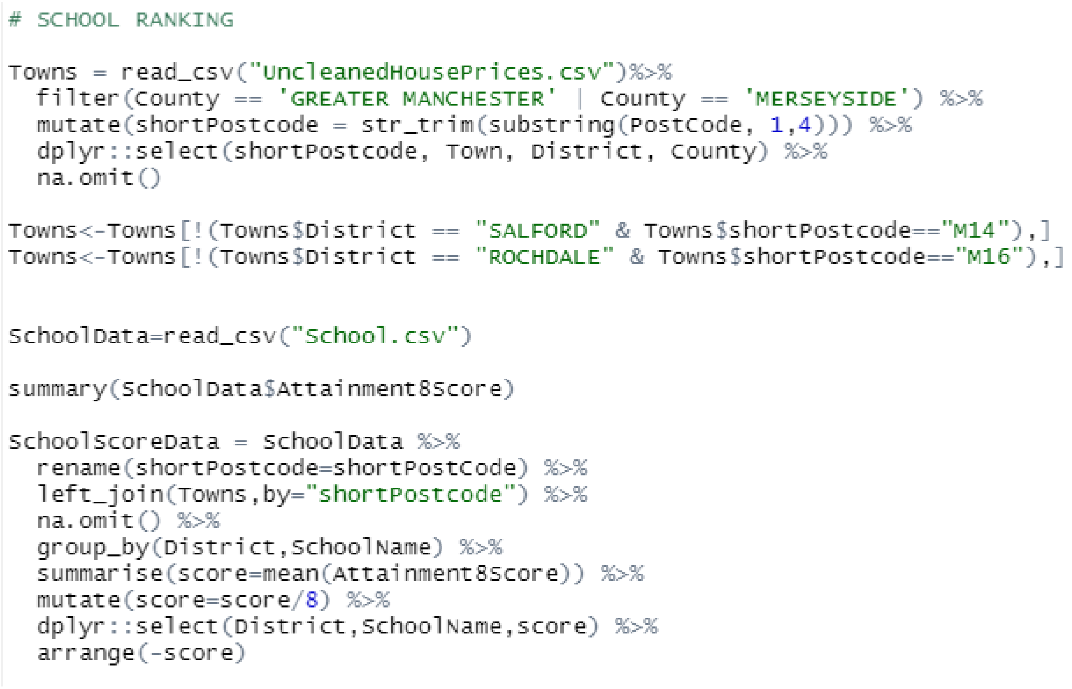
Code for Crime rate ranking



```
# CRIME RANKING

Town  = read_csv("Towns.csv") %>%
  mutate(shortPostcode = str_trim(stri_sub(PostCode,-3))) %>%
  dplyr::select(shortPostcode, Town, District, County)

crimeData = read_csv("Crime.csv") %>%
  dplyr::select(ID,Year,shortPostcode,CrimeType, CrimeCount)


summary(crimeData$CrimeCount)


crimeDataRanking = crimeData %>%
  left_join(Town, by = "shortPostcode") %>%
  na.omit() %>%
  dplyr::select(District,CrimeCount) %>%
  group_by(District) %>%
  summarise(AverageCrimeCount=mean(CrimeCount)) %>%
  arrange(AverageCrimeCount) %>%
  mutate(CrimeCountScore=10 - AverageCrimeCount/391) %>%
  dplyr::select(District,CrimeCountScore)

crimeDataRanking
```

Figure 47.1

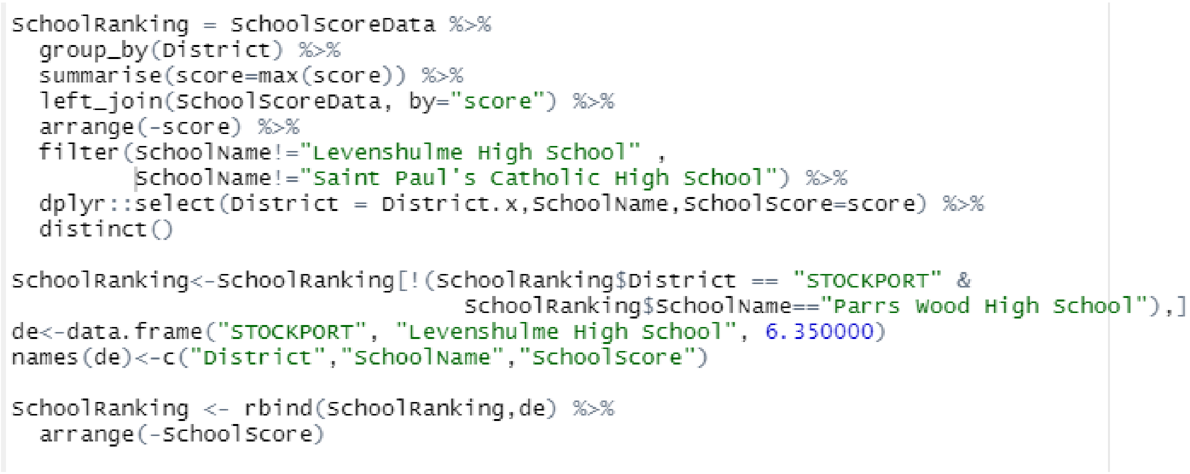
Code for Crime rate ranking



```
# SCHOOL RANKING

Towns = read_csv("UncleanedHousePrices.csv")%>%
  filter(County == 'GREATER MANCHESTER' | County == 'MERSEYSIDE') %>%
  mutate(shortPostcode = str_trim(substring(PostCode, 1,4))) %>%
  dplyr::select(shortPostcode, Town, District, County) %>%
  na.omit()

Towns<-Towns[!(Towns$District == "SALFORD" & Towns$shortPostcode=="M14"),]
Towns<-Towns[!(Towns$District == "ROCHDALE" & Towns$shortPostcode=="M16"),]


SchoolData=read_csv("School.csv")

summary(SchoolData$Attainment8Score)

SchoolScoreData = SchoolData %>%
  rename(shortPostcode=shortPostCode) %>%
  left_join(Towns,by="shortPostcode") %>%
  na.omit() %>%
  group_by(District,SchoolName) %>%
  summarise(score=mean(Attainment8Score)) %>%
  mutate(score=score/8) %>%
  dplyr::select(District,SchoolName,score) %>%
  arrange(-score)
```

Figure 47.2

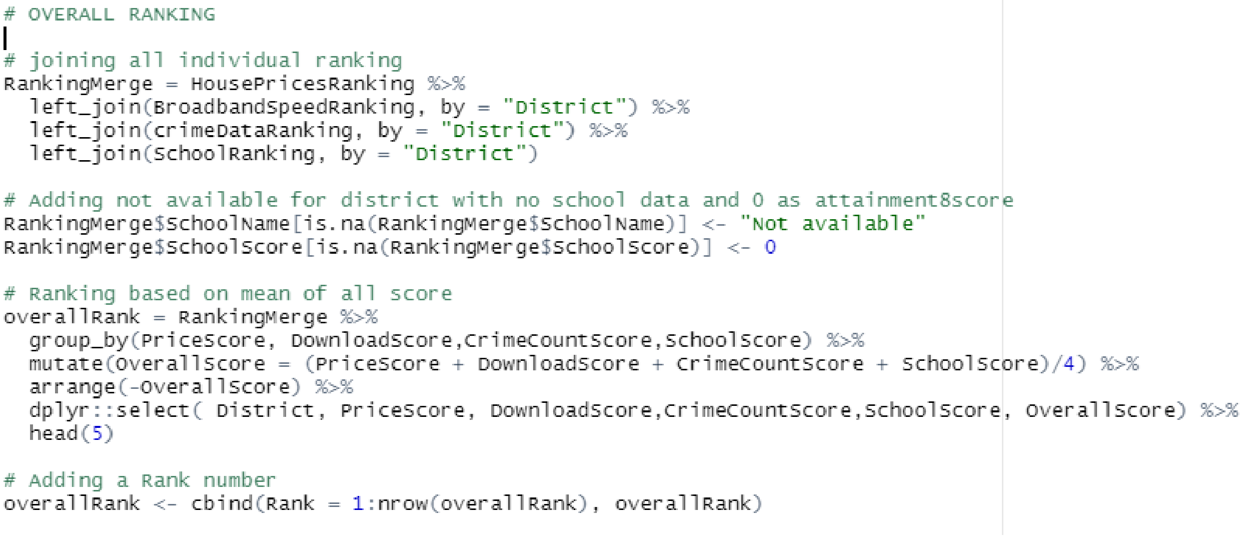
Code for Crime rate ranking



```r
SchoolRanking = SchoolScoreData %>%
  group_by(District) %>%
  summarise(score=max(score)) %>%
  left_join(SchoolScoreData, by="score") %>%
  arrange(-score) %>%
  filter(SchoolName!="Levenshulme High School" ,
         SchoolName!="Saint Paul's Catholic High School") %>%
  dplyr::select(District = District.x,SchoolName,SchoolScore=score) %>%
  distinct()

SchoolRanking<-SchoolRanking[!(SchoolRanking$District == "STOCKPORT" &
                                 SchoolRanking$SchoolName=="Parrs Wood High School"),]
de<-data.frame("STOCKPORT", "Levenshulme High School", 6.350000)
names(de)<-c("District","SchoolName","SchoolScore")

SchoolRanking <- rbind(SchoolRanking,de) %>%
  arrange(-SchoolScore)
```

Figure 48

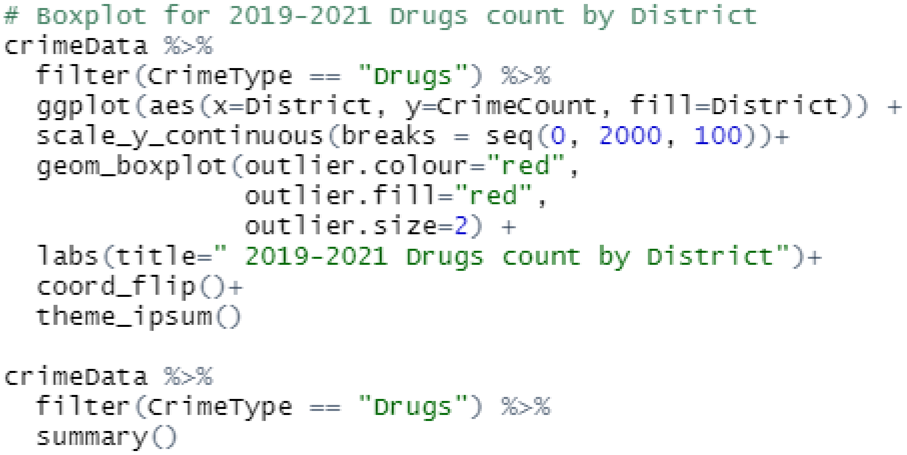
Code for Ranking recommendation system



```r
# OVERALL RANKING

# joining all individual ranking
RankingMerge = HousePricesRanking %>%
  left_join(BroadbandSpeedRanking, by = "District") %>%
  left_join(crimeDataRanking, by = "District") %>%
  left_join(SchoolRanking, by = "District")

# Adding not available for district with no school data and 0 as attainment8score
RankingMerge$SchoolName[is.na(RankingMerge$SchoolName)] <- "Not available"
RankingMerge$SchoolScore[is.na(RankingMerge$SchoolScore)] <- 0

# Ranking based on mean of all score
overallRank = RankingMerge %>%
  group_by(PriceScore, DownloadScore,CrimeCountScore,SchoolScore) %>%
  mutate(overallScore = (PriceScore + DownloadScore + CrimeCountScore + SchoolScore)/4) %>%
  arrange(-overallScore) %>%
  dplyr::select( District, PriceScore, DownloadScore,CrimeCountScore,SchoolScore, overallScore) %>%
  head(5)

# Adding a Rank number
overallRank <- cbind(Rank = 1:nrow(overallRank), overallRank)
```

Figure 49

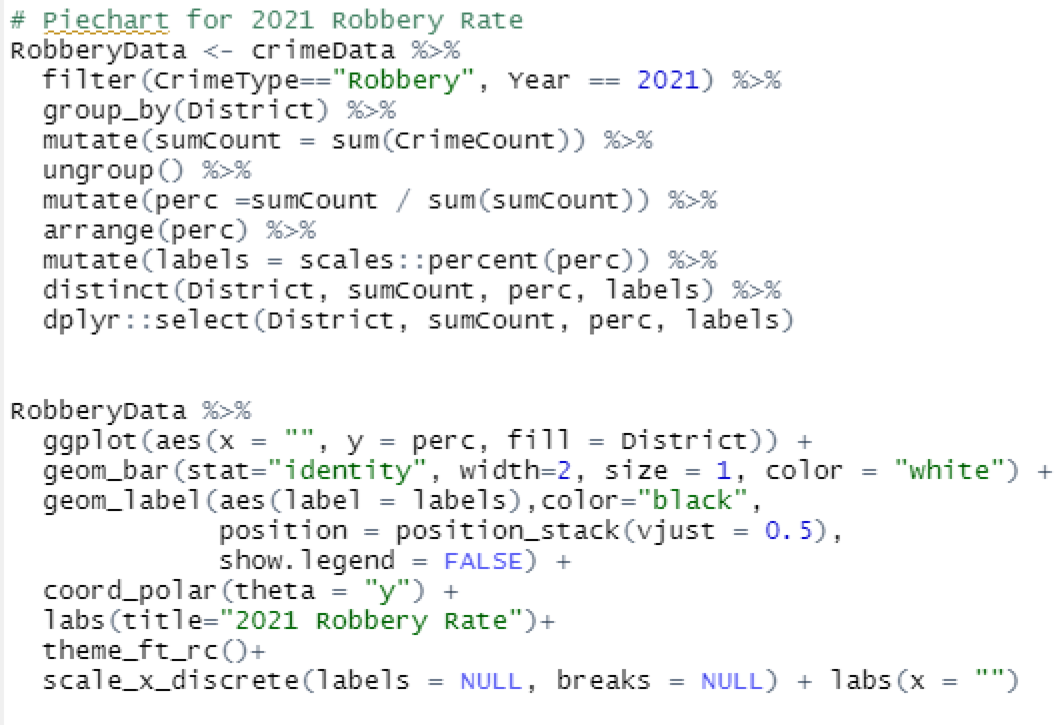
Boxplot for Drugs count



```
# Boxplot for 2019-2021 Drugs count by District
crimeData %>%
  filter(CrimeType == "Drugs") %>%
  ggplot(aes(x=District, y=CrimeCount, fill=District)) +
  scale_y_continuous(breaks = seq(0, 2000, 100))+
  geom_boxplot(outlier.colour="red",
               outlier.fill="red",
               outlier.size=2) +
  labs(title=" 2019-2021 Drugs count by District")+
  coord_flip()+
  theme_ipsum()

crimeData %>%
  filter(CrimeType == "Drugs") %>%
  summary()
```

Figure 50

Pie chart for 2021 Robbery rate



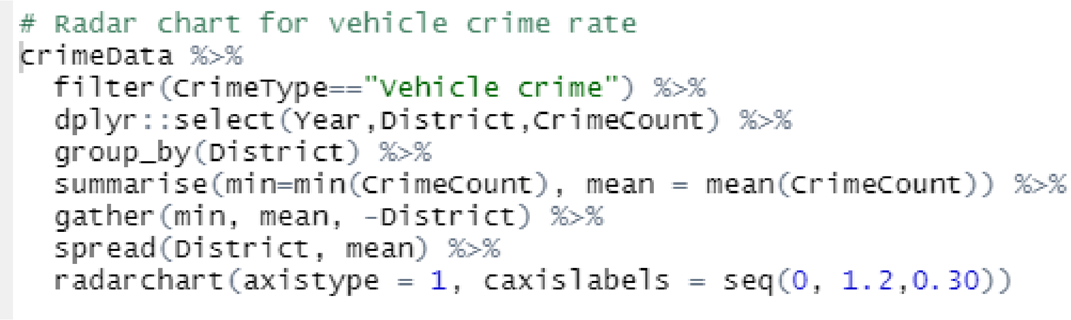
```
# Piechart for 2021 Robbery Rate
RobberyData <- crimeData %>%
  filter(CrimeType=="Robbery", Year == 2021) %>%
  group_by(District) %>%
  mutate(sumCount = sum(CrimeCount)) %>%
  ungroup() %>%
  mutate(perc =sumCount / sum(sumCount)) %>%
  arrange(perc) %>%
  mutate(labels = scales::percent(perc)) %>%
  distinct(District, sumCount, perc, labels) %>%
  dplyr::select(District, sumCount, perc, labels)


RobberyData %>%
  ggplot(aes(x = "", y = perc, fill = District)) +
  geom_bar(stat="identity", width=2, size = 1, color = "white") +
  geom_label(aes(label = labels),color="black",
             position = position_stack(vjust = 0.5),
             show.legend = FALSE) +
  coord_polar(theta = "y") +
  labs(title="2021 Robbery Rate")+
  theme_ft_rc()+
  scale_x_discrete(labels = NULL, breaks = NULL) + labs(x = "")
```

```
# Piechart for 2021 Robbery Rate of MERSEYSIDE
RobberyData <- crimeData %>%
  filter(CrimeType=="Robbery", Year == 2021, County == "MERSEYSIDE") %>%
  group_by(District) %>%
  mutate(sumCount = sum(CrimeCount)) %>%
  ungroup() %>%
  mutate(perc =sumCount / sum(sumCount)) %>%
  arrange(perc) %>%
  mutate(labels = scales::percent(perc)) %>%
  distinct(District, sumCount, perc, labels) %>%
  select(District, sumCount, perc, labels)

RobberyData %>%
  ggplot(aes(x = "", y = perc, fill = District)) +
  geom_bar(stat="identity", width=2, size = 1, color = "white") +
  geom_label(aes(label = labels),color="black",
             position = position_stack(vjust = 0.5),
             show.legend = FALSE) +
  coord_polar(theta = "y") +
  labs(title="2021 Robbery Rate of MERSEYSIDE")+
  theme_ft_rc()+
  scale_x_discrete(labels = NULL, breaks = NULL) + labs(x = "")
```

```
# Piechart for 2021 Robbery Rate of GREATER MANCHESTER
RobberyData <- crimeData %>%
  filter(CrimeType=="Robbery", Year == 2021, County == "GREATER MANCHESTER") %>%
  group_by(District) %>%
  mutate(sumCount = sum(CrimeCount)) %>%
  ungroup() %>%
  mutate(perc =sumCount / sum(sumCount)) %>%
  arrange(perc) %>%
  mutate(labels = scales::percent(perc)) %>%
  distinct(District, sumCount, perc, labels) %>%
  select(District, sumCount, perc, labels)

RobberyData %>%
  ggplot(aes(x = "", y = perc, fill = District)) +
  geom_bar(stat="identity", width=2, size = 1, color = "white") +
  geom_label(aes(label = labels),color="black",
             position = position_stack(vjust = 0.5),
             show.legend = FALSE) +
  coord_polar(theta = "y") +
  labs(title="2021 Robbery Rate of GREATER MANCHESTER")+
  theme_ft_rc()+
  scale_x_discrete(labels = NULL, breaks = NULL) + labs(x = "")
```

Figure 51

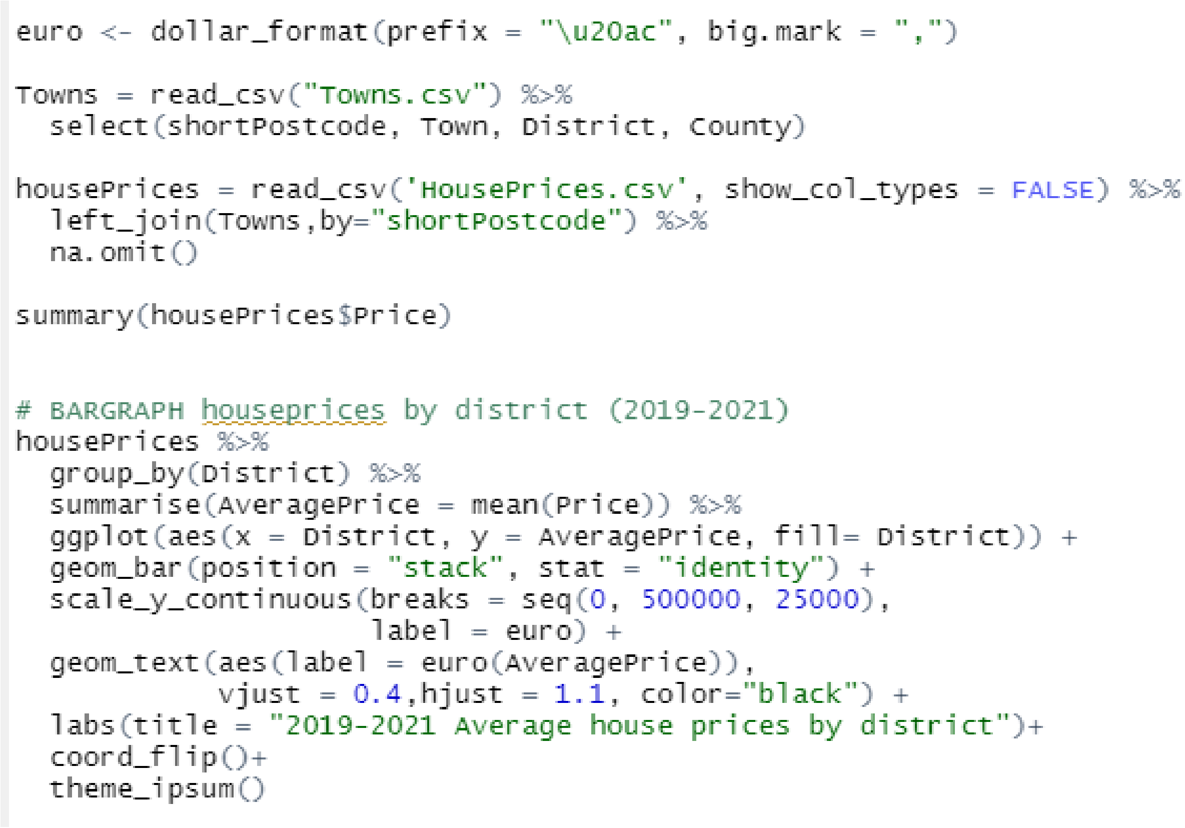
Pie chart for 2021 Robbery rate



```
# Radar chart for vehicle crime rate
crimeData %>%
  filter(CrimeType=="Vehicle crime") %>%
  dplyr::select(Year,District,CrimeCount) %>%
  group_by(District) %>%
  summarise(min=min(CrimeCount), mean = mean(CrimeCount)) %>%
  gather(min, mean, -District) %>%
  spread(District, mean) %>%
  radarchart(axistype = 1, caxislabels = seq(0, 1.2,0.30))
```

Figure 52

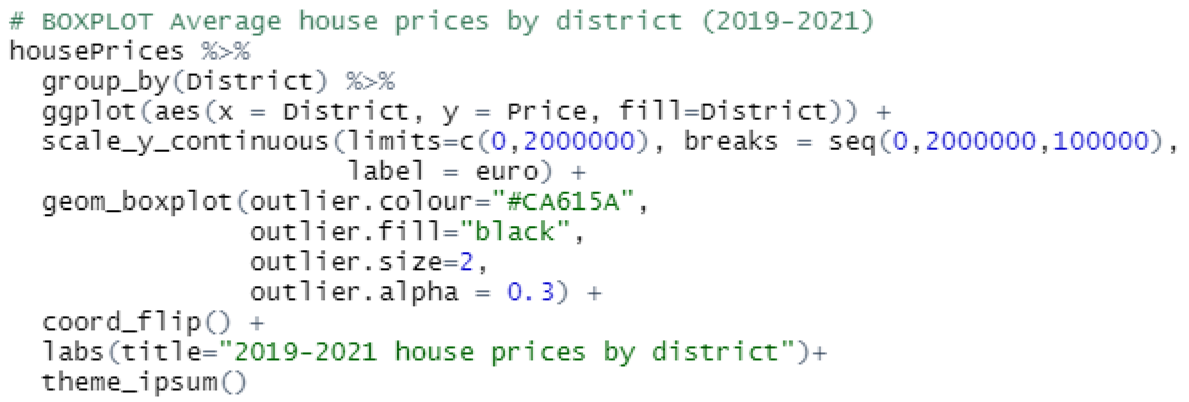
Bar graph of house price from 2019-2021



```
euro <- dollar_format(prefix = "\u20ac", big.mark = ",")

Towns = read_csv("Towns.csv") %>%
  select(shortPostcode, Town, District, County)

housePrices = read_csv('HousePrices.csv', show_col_types = FALSE) %>%
  left_join(Towns,by="shortPostcode") %>%
  na.omit()

summary(housePrices$Price)


# BARGRAPH houseprices by district (2019-2021)
housePrices %>%
  group_by(District) %>%
  summarise(AveragePrice = mean(Price)) %>%
  ggplot(aes(x = District, y = AveragePrice, fill= District)) +
  geom_bar(position = "stack", stat = "identity") +
  scale_y_continuous(breaks = seq(0, 500000, 25000),
                     label = euro) +
  geom_text(aes(label = euro(AveragePrice)),
            vjust = 0.4,hjust = 1.1, color="black") +
  labs(title = "2019-2021 Average house prices by district")+
  coord_flip()+
  theme_ipsum()
```

Figure 53

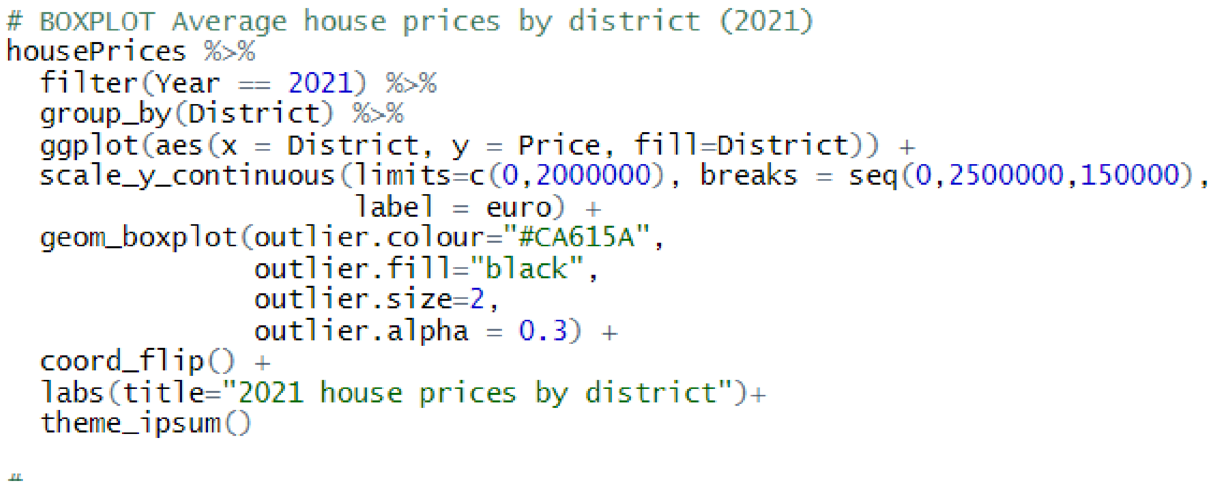
Boxplot average house prices by district for 2019-2021



```
# BOXPLOT Average house prices by district (2019-2021)
housePrices %>%
  group_by(District) %>%
  ggplot(aes(x = District, y = Price, fill=District)) +
  scale_y_continuous(limits=c(0,2000000), breaks = seq(0,2000000,100000),
                     label = euro) +
  geom_boxplot(outlier.colour="#CA615A",
               outlier.fill="black",
               outlier.size=2,
               outlier.alpha = 0.3) +
  coord_flip() +
  labs(title="2019-2021 house prices by district")+
  theme_ipsum()
```

Figure 54

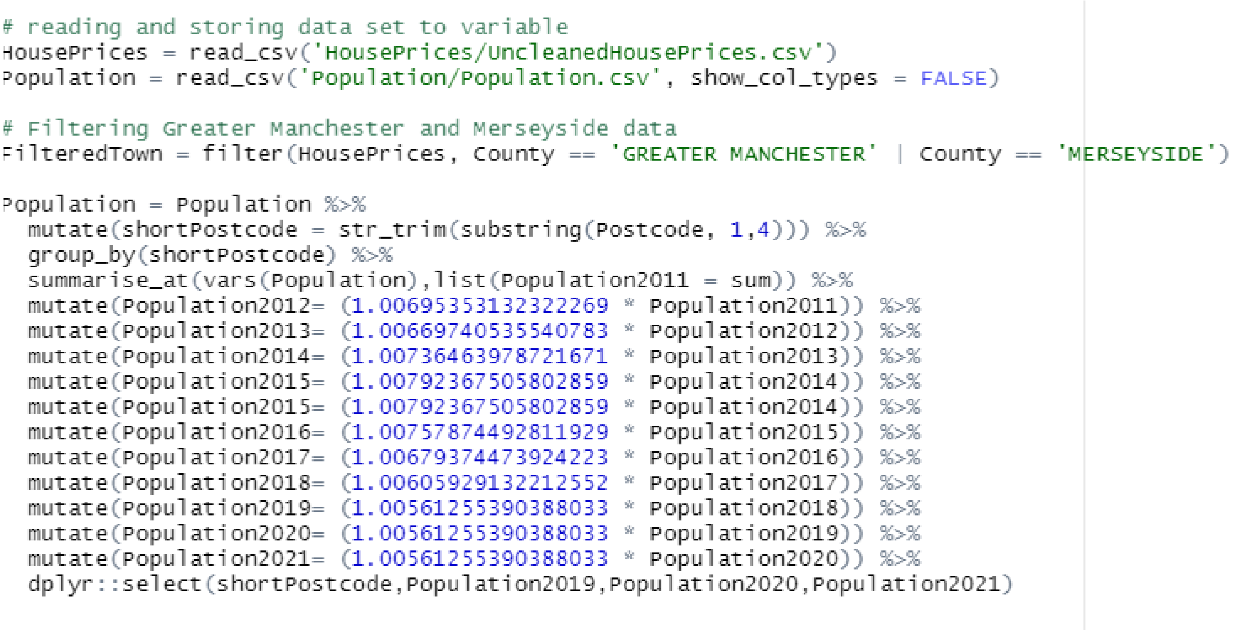
Code for *Box plot of 2021 House prices by district*



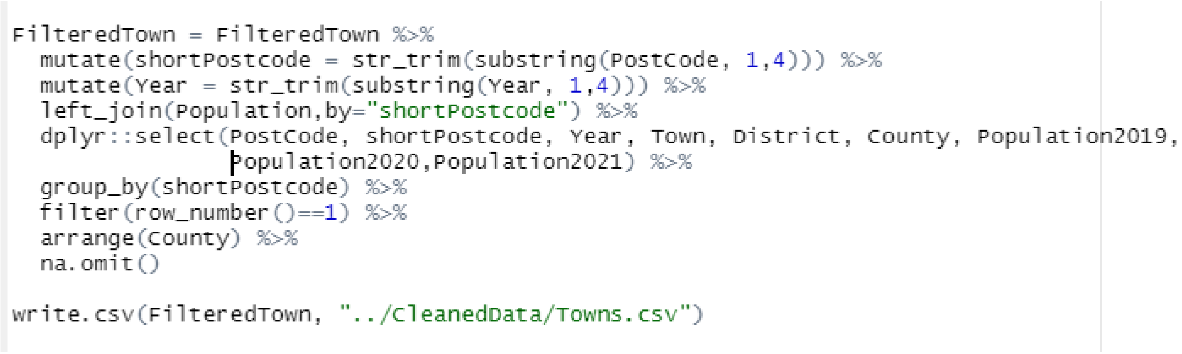
```
# BOXPLOT Average house prices by district (2021)
housePrices %>%
  filter(Year == 2021) %>%
  group_by(District) %>%
  ggplot(aes(x = District, y = Price, fill=District)) +
  scale_y_continuous(limits=c(0,2000000), breaks = seq(0,2500000,150000),
                     label = euro) +
  geom_boxplot(outlier.colour="#CA615A",
               outlier.fill="black",
               outlier.size=2,
               outlier.alpha = 0.3) +
  coord_flip() +
  labs(title="2021 house prices by district")+
  theme_ipsum()
```

Figure 55

Code for Population and Town Cleaning



```
# reading and storing data set to variable
HousePrices = read_csv('HousePrices/UncleanedHousePrices.csv')
Population = read_csv('Population/Population.csv', show_col_types = FALSE)

# Filtering Greater Manchester and Merseyside data
FilteredTown = filter(HousePrices, County == 'GREATER MANCHESTER' | County == 'MERSEYSIDE')

Population = Population %>%
  mutate(shortPostcode = str_trim(substring(Postcode, 1,4))) %>%
  group_by(shortPostcode) %>%
  summarise_at(vars(Population),list(Population2011 = sum)) %>%
  mutate(Population2012= (1.00695353132322269 * Population2011)) %>%
  mutate(Population2013= (1.00669740535540783 * Population2012)) %>%
  mutate(Population2014= (1.00736463978721671 * Population2013)) %>%
  mutate(Population2015= (1.00792367505802859 * Population2014)) %>%
  mutate(Population2015= (1.00792367505802859 * Population2014)) %>%
  mutate(Population2016= (1.00757874492811929 * Population2015)) %>%
  mutate(Population2017= (1.00679374473924223 * Population2016)) %>%
  mutate(Population2018= (1.00605929132212552 * Population2017)) %>%
  mutate(Population2019= (1.00561255390388033 * Population2018)) %>%
  mutate(Population2020= (1.00561255390388033 * Population2019)) %>%
  mutate(Population2021= (1.00561255390388033 * Population2020)) %>%
  dplyr::select(shortPostcode,Population2019,Population2020,Population2021)
```



```
FilteredTown = FilteredTown %>%
  mutate(shortPostcode = str_trim(substring(PostCode, 1,4))) %>%
  mutate(Year = str_trim(substring(Year, 1,4))) %>%
  left_join(Population,by="shortPostcode") %>%
  dplyr::select(PostCode, shortPostcode, Year, Town, District, County, Population2019,
                Population2020,Population2021) %>%
  group_by(shortPostcode) %>%
  filter(row_number()==1) %>%
  arrange(County) %>%
  na.omit()

write.csv(FilteredTown, "../CleanedData/Towns.csv")
```

R FILES LINK

The Clean.R, Graph.R, LinearModel.R, Rank.R files along with the dataset used are stored in the OneDrive and can be accessed with the link below:

https://softwaricacollege-my.sharepoint.com/:f:/g/personal/200261_softwarica_edu_np/EpdWg0-ld4xIjX_Fb1P-UvEB5mo2iLcrxwJ3mhsA6ZV-Aw?e=PFiTzT

REFERENCES

Price Paid Data (2022) available from <https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads> [30 July 2022

(2022) available from <https://www.ofcom.org.uk/research-and-data/multi-sector-research/infrastructure-research/connected-nations-2018/data-downloads> [30 July 2022]

Data Downloads | Data.Police.Uk (2022) available from <https://data.police.uk/data/> [30 July 2022]

Population.Xlsx (2022) available from <https://softwaricacollege-my.sharepoint.com/:x:/g/personal/200261_softwarica_edu_np/ERcKwIRTkWlFh-UV1xeJp9QBel3glJ4tEUMiCQqnH8fvQw?e=A6aUSX> [30 July 2022]

Download Data - GOV.UK - Find And Compare Schools In England (2022) available from <https://www.compare-school-performance.service.gov.uk/download-data?currentstep=datatypes&regiontype=all&la=0&downloadYear=2018-2019&datatypes=ks5> [30 July 2022]

Regression?, W. (2022) *Residual Values (Residuals) In Regression Analysis* [online] available from <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/residual/> [30 July 2022]

Understanding Boxplots (2022) available from <https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51> [30 July 2022]