Individual Report

Data Science for Developers

Sangam Basnet

Softwarica College of IT and E-Commerce, Coventry University

ST5014CEM Data Science for Developer

Siddhartha Neupane

19th August 2024

Table of content

Table of figures

Introduction

Data science, as defined by many authors, is a complex process of obtaining knowledge from various structured and unstructured data taking advantage of the scientific methods, algorithms, and systems. It covers tools from statistical analysis, Computing science and subject matter specialism to make conclusions from large data sets. Data science process is divided into several steps as data gathering, data cleaning, data analysis, and data dissemination to support decision making in different sectors. About Data Science, it has emerged as a vital tool in solving real life issues, the promotion of development and the provision of competitive advantage to companies, organizations, healthcare institutions, in finance, and in all the other fields.

The data science life cycle consists of several stages: Data collection that involves accessing data from different sources, including databases, APIs, as well as web scraping; Data cleaning that involves processing data, detecting and handling missing values, outliers and other inconsistencies; and data exploration in which preliminary analysis is done with the aim of identifying patterns, outliers and carrying out initial hypothesis testing. Following is data modeling where predictive or descriptive statistical or machine learning models are generated, and afterwards model selection by comparing it with standards and measures of validity. The next step is to deploy the model where it will be applied in the actual processing of data, the last step is the maintenance of the model to ensure that it is adjusted to the environment and keeps giving reliable results. This gives a positive feedback and also creates the cycle of adaptation to new data and information.

Data Cleaning

The process of data cleaning is an important element in data science, which is used to prepare the raw data for the analysis by removing the errors. It begins with missing data that can be dealt in various ways including imputation, deletion or use of algorithms that accept missing values. In order to overcome the possible erratic values, outliers and anomalies are expelled and dealt with accordingly. The records which are duplication in nature and are nuisance are weeded out of the aggregation process to make the data clean. Data is also dealt with in a way that makes it uniform in terms of format, unit and scale. Data cleaning also entails rectifying errors such as spelling mistakes, formatting mistakes such as wrong entries and checking categorical variables on the right format as nominated and or validated data against predefined standards. It assures the credibility of the collected data and also ensures that the gathered data is accurate, which is an important groundwork for analysis and modeling.

House Pricing

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

# Set the working directory
setwd("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam")

# Read CSV files
hp_2020 = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/Housing/pp-2020.csv", show_col_types = FALSE)
hp_2021 = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/Housing/pp-2021.csv", show_col_types = FALSE)
hp_2022 = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/Housing/pp-2022.csv", show_col_types = FALSE)
hp_2023 = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/Housing/pp-2023.csv", show_col_types = FALSE)

# Assign column names for all datasets
colnames(hp_2020) = colnames(hp_2021) = colnames(hp_2022) = colnames(hp_2023) = c("ID", "Price", "Year", "PostCode", "PAON", "SAON", "FL", "House_Num", "Flat",

# Combine datasets, remove missing values and duplicates, and convert to tibble
HousePrices_combined = bind_rows(hp_2020, hp_2021, hp_2022, hp_2023) %>%
  na.omit() %>%
  distinct() %>%
  as_tibble()

# Write the cleaned data to a CSV file
write.csv(HousePrices_combined, "C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/HousePricing_from(2020-2023).csv", row.names = FALSE)

# Filter the dataset for specific counties (modify as needed for Bristol and Cornwall)
Filtered_HousePrices = filter(HousePrices_combined, Country == 'CITY OF BRISTOL' | Country == 'CORNWALL')

# Define a regex pattern to remove space and everything after it
pattern = ' .*$'
```

Figure 1Housepricing 1

```r
# Clean and refine the filtered data
Filtered_HousePrices_cleaned = Filtered_HousePrices %>%
  mutate(short_PostCode = gsub(pattern, "", PostCode)) %>%
  mutate(Year = str_trim(substring(Year, 1, 4))) %>%
  select(PostCode, short_PostCode, Year, District, Town, Price, Country) %>%
  na.omit() %>%      # Remove rows with missing values
  distinct() %>%     # Keep only unique rows
  as_tibble()

# View the cleaned data
View(Filtered_HousePrices_cleaned)

# Export the cleaned  data to a CSV file
write.csv(Filtered_HousePrices_cleaned, "C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_HousePricing(2020-2023).csv", row.names = FALS
```

Figure 2:Housepricing 2

This script prepares and cleans housing price data from 2020 to 2023. It reads multiple CSV files, assigns consistent column names, and combines the datasets. The data undergoes cleaning by removing missing values and duplicates. It is then filtered to include only entries from specific counties (Bristol and Cornwall). The script refines the dataset by creating a short version of postcodes, trimming the year to four digits, and selecting relevant columns. Finally, the cleaned data is saved to a new CSV file, providing a streamlined and accurate dataset for further analysis.

Population Cleaning

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

# Set the working directory to where the data files are located
setwd("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam")

# Read in the raw house prices and population data from CSV files
house_prices_raw = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Combined_HousePricing(2020-2023).csv")
population_data = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/Population2011_1656567141570.csv", show_col_types = FALSE)

# Filter data to include only towns in Bristol and Cornwall
towns_filtered = filter(house_prices_raw, Country == 'CITY OF BRISTOL' | Country == 'CORNWALL')

# Define a regex pattern to extract the short postcode
postcode_pattern = ' .*$'
```

Figure 3: population cleaning1

```r
# Calculate population estimates for each year from 2011 to 2023
population_data = population_data %>%
  mutate(short_postcode = gsub(postcode_pattern, "", Postcode)) %>%
  group_by(short_postcode) %>%
  summarise_at(vars(Population), list(Pop2011 = sum)) %>%
  mutate(
    Pop2012 = 1.00695353132322269 * Pop2011,
    Pop2013 = 1.00669740535540783 * Pop2012,
    Pop2014 = 1.00736463978721671 * Pop2013,
    Pop2015 = 1.00792367505802859 * Pop2014,
    Pop2016 = 1.00757874492811929 * Pop2015,
    Pop2017 = 1.00679374473924223 * Pop2016,
    Pop2018 = 1.00605929132212552 * Pop2017,
    Pop2019 = 1.00561255390388033 * Pop2018,
    Pop2020 = 1.00561255390388033 * Pop2019,
    Pop2021 = 1.00561255390388033 * Pop2020,
    Pop2022 = 1.00561255390388033 * Pop2021,
    Pop2023 = 1.00561255390388033 * Pop2022
  ) %>%
  select(short_postcode, Pop2020, Pop2021, Pop2022, Pop2023)

# Clean and merge the house prices data with the population data
towns_filtered = towns_filtered %>%
  mutate(short_postcode = gsub(postcode_pattern, "", PostCode)) %>%
  mutate(Year = str_trim(substring(Year, 1, 4))) %>%
  left_join(population_data, by = "short_postcode") %>%
  select(PostCode, short_postcode, Year, Town, District, Country, Pop2020, Pop2021, Pop2022, Pop2023) %>%
  group_by(short_postcode) %>%
  arrange(Country) %>%
  as_tibble() %>%
  na.omit() %>%
  distinct()

# View the cleaned and merged data
View(towns_filtered)

# Save the cleaned data to a new CSV file
write.csv(towns_filtered, "C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_Population.csv", row.names = FALSE)
```

Figure 4: population cleaning2

This script weights and amalgamates house prices and population concerning towns in Bristol and Cornwall. It begins with reading raw data from CSV files and proceeds with filtering out the house prices data by regions of interest. It then uses a growth rate to arrive at population estimates per year from the year 2011 up to and inclusive of the year 2023 and extracting short postcodes. The cleaning activity is done on the house prices data in which the years are trimmed and short postcodes are extracted. Short postcodes of both the datasets are matched and an effort is made to retain only those columns which are required. The last cleaned and merged data is observed, and a new CSV file is created, which will help in data analysis and reporting.

LSOA Cleaning

```r
# Importing required libraries
install.packages("data.table")
library(data.table)
library(tidyverse)
library(dplyr)


# Setting the working directory
setwd("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam")

# Reading the cleaned town population data
cleaned_population = read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_Population.csv")

# Reading the postcode to LSOA mapping data
lsoa = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/Postcode to LSOA.csv/Postcode to LSOA.csv")

# Cleaning the LSOA data
pattern = ' .*$'
lsoa_cleaned = lsoa %>%
  select(lsoa11cd, pcds) %>%
  mutate(short_postcode = gsub(pattern, "", pcds)) %>%
  right_join(cleaned_population, by = "short_postcode") %>%
  group_by(lsoa11cd) %>%
  select(lsoa11cd, short_postcode, Town, District, Country)

# Renaming the first column to "LSOA code"
colnames(lsoa_cleaned)[1] = "LSOA code"

# Viewing the cleaned LSOA data
view(lsoa_cleaned)

# Writing the cleaned LSOA data to a CSV file
write.csv(lsoa_cleaned, "C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_LSOA.csv", row.names = FALSE, col.names = FALSE)

```

Figure 5: LSOA Cleaning

This script assimilates and enrobes town population data into LSOA mapping data. This involves setting of working directory and loading the required libraries into the working directory. The cleaned town population data that has been downloaded from the website in CSV format and the mapping from the Postcode to LSOA are also in CSV data format. Cleaning the LSOA data entails subsetting on appropriate columns, extracting short postcodes, and merging with the 'clean' population data where the keys are short postcodes. Having gotten the resulting data set, this is further grouped by LSOA code and the necessary columns are selected. The first column is therefore renamed to 'LSOA code' for clarity. Last

of all for utilization and further analysis the cleaned LSOA data is displayed and exported in a new CSV format.

Crime Cleaning

```
#Load library
library(data.table)
library(tidyverse)
library(stringr)
library(lubridate)
#set working directory
setwd("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam")

#Importing Bristol
Crime_2021_05_Bristol = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-05/2021-05-avon-and-somerset-st
Crime_2021_06_Bristol = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-06/2021-06-avon-and-somerset-st
Crime_2021_07_Bristol = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-07/2021-07-avon-and-somerset-st
Crime_2021_08_Bristol = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-08/2021-08-avon-and-somerset-st
Crime_2021_09_Bristol = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-09/2021-09-avon-and-somerset-st
Crime_2021_10_Bristol = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-10/2021-10-avon-and-somerset-st
Crime_2021_11_Bristol = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-11/2021-11-avon-and-somerset-st
Crime_2021_12_Bristol = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-12/2021-12-avon-and-somerset-st


Crime_2022_01_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-01/2022-01-avon-and-somerset-s
Crime_2022_02_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-02/2022-02-avon-and-somerset-s
Crime_2022_03_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-03/2022-03-avon-and-somerset-s
Crime_2022_04_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-04/2022-04-avon-and-somerset-s
Crime_2022_05_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-05/2022-05-avon-and-somerset-s
Crime_2022_06_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-06/2022-06-avon-and-somerset-s
Crime_2022_07_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-07/2022-07-avon-and-somerset-s
Crime_2022_08_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-08/2022-08-avon-and-somerset-s
Crime_2022_09_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-09/2022-09-avon-and-somerset-s
Crime_2022_10_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-10/2022-10-avon-and-somerset-s
Crime_2022_11_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-11/2022-11-avon-and-somerset-s
Crime_2022_12_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-12/2022-12-avon-and-somerset-s
```

Figure 6: crime cleaning 1

```
Crime_2023_01_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-01/2023-01-avon-and-somerset-s
Crime_2023_02_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-02/2023-02-avon-and-somerset-s
Crime_2023_03_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-03/2023-03-avon-and-somerset-s
Crime_2023_04_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-04/2023-04-avon-and-somerset-s
Crime_2023_05_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-05/2023-05-avon-and-somerset-s
Crime_2023_06_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-06/2023-06-avon-and-somerset-s
Crime_2023_07_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-07/2023-07-avon-and-somerset-s
Crime_2023_08_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-08/2023-08-avon-and-somerset-s
Crime_2023_09_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-09/2023-09-avon-and-somerset-s
Crime_2023_10_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-10/2023-10-avon-and-somerset-s
Crime_2023_11_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-11/2023-11-avon-and-somerset-s
Crime_2023_12_Bristol <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-12/2023-12-avon-and-somerset-s


#Importing cornwall
Crime_2021_05_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-05/2021-05-devon-and-cornwall
Crime_2021_06_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-06/2021-06-devon-and-cornwall
Crime_2021_07_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-07/2021-07-devon-and-cornwall
Crime_2021_08_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-08/2021-08-devon-and-cornwall
Crime_2021_09_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-09/2021-09-devon-and-cornwall
Crime_2021_10_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-10/2021-10-devon-and-cornwall
Crime_2021_11_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-11/2021-11-devon-and-cornwall
Crime_2021_12_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2021-12/2021-12-devon-and-cornwall

Crime_2022_01_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-01/2022-01-devon-and-cornwall
Crime_2022_02_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-02/2022-02-devon-and-cornwall
Crime_2022_03_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-03/2022-03-devon-and-cornwall
Crime_2022_04_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-04/2022-04-devon-and-cornwall
Crime_2022_05_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-05/2022-05-devon-and-cornwall
Crime_2022_06_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-06/2022-06-devon-and-cornwall
Crime_2022_07_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-07/2022-07-devon-and-cornwall
Crime_2022_08_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-08/2022-08-devon-and-cornwall
Crime_2022_09_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-09/2022-09-devon-and-cornwall
Crime_2022_10_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-10/2022-10-devon-and-cornwall
Crime_2022_11_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-11/2022-11-devon-and-cornwall
Crime_2022_12_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2022-12/2022-12-devon-and-cornwall
```

Figure 7: Crime cleaning 2

```r
Crime_combined <- Crime_combined %>%
  as_tibble()

write.csv(Crime_combined, "C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Combined_crime.csv", row.names = FALSE)

# View the combined dataset for verification
View(Crime_combined)

# Load the combined and cleaned crime dataset
crimedata <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Combined_crime.csv") %>%
  select(Month, `LSOA code`, `Crime type`, `Falls within`)

# Rename columns for clarity
colnames(crimedata) <- c("Year", "LSOA.code", "CrimeType", "Falls Within")

# Load the LSOA to Postcode mapping dataset
LsoaToPostcode <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_LSOA.csv")
colnames(LsoaToPostcode) <- c("LSOA.code", "shortPostcode", "Town", "District", "County")

# Check for and handle duplicates in both datasets
if (any(duplicated(crimedata$LSOA.code))) {
  cat("Duplicates found in crime data\n")
}
if (any(duplicated(LsoaToPostcode$LSOA.code))) {
  cat("Duplicates found in LSOA to Postcode mapping\n")
}

# Remove duplicates from both datasets
crimedata <- unique(crimedata, by = "LSOA.code")
LsoaToPostcode <- unique(LsoaToPostcode, by = "LSOA.code")

# Clean and merge the datasets, then aggregate crime counts
Crime_DataCleaned <- crimedata %>%
  left_join(LsoaToPostcode, by = "LSOA.code") %>%
  mutate(Year = str_trim(substring(Year, 1, 4))) %>%
  group_by(shortPostcode, CrimeType, Year, `Falls Within`) %>%
  filter(!is.na(shortPostcode) & !is.na(CrimeType) & !is.na(Year) & !is.na(`Falls Within`)) %>%
  summarize(CrimeCount = n()) %>%
  ungroup()

# Save the cleaned dataset to a CSV file
write.csv(Crime_DataCleaned, "C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_crime.csv", row.names = FALSE)

# View the cleaned dataset for verification
View(Crime_DataCleaned)
```

Figure 8: Crime cleaning 4

```r
Crime_2023_01_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-01/2023-01-devon-and-cornwall
Crime_2023_02_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-02/2023-02-devon-and-cornwall
Crime_2023_03_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-03/2023-03-devon-and-cornwall
Crime_2023_04_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-04/2023-04-devon-and-cornwall
Crime_2023_05_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-05/2023-05-devon-and-cornwall
Crime_2023_06_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-06/2023-06-devon-and-cornwall
Crime_2023_07_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-07/2023-07-devon-and-cornwall
Crime_2023_08_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-08/2023-08-devon-and-cornwall
Crime_2023_09_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-09/2023-09-devon-and-cornwall
Crime_2023_10_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-10/2023-10-devon-and-cornwall
Crime_2023_11_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-11/2023-11-devon-and-cornwall
Crime_2023_12_Cornwall <- read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/390a695b1f8ee982a01269b847ec51e1a0c07f5f/2023-12/2023-12-devon-and-cornwall

# Combine the datasets
Crime_combined = rbind(
  Crime_2021_05_Bristol, Crime_2021_06_Bristol, Crime_2021_07_Bristol,
  Crime_2021_08_Bristol, Crime_2021_09_Bristol, Crime_2021_10_Bristol, Crime_2021_11_Bristol,
  Crime_2021_12_Bristol, Crime_2022_01_Bristol, Crime_2022_02_Bristol, Crime_2022_03_Bristol,
  Crime_2022_04_Bristol, Crime_2022_05_Bristol, Crime_2022_06_Bristol, Crime_2022_07_Bristol,
  Crime_2022_08_Bristol, Crime_2022_09_Bristol, Crime_2022_10_Bristol, Crime_2022_11_Bristol,
  Crime_2022_12_Bristol, Crime_2023_01_Bristol, Crime_2023_02_Bristol, Crime_2023_03_Bristol,
  Crime_2023_04_Bristol, Crime_2023_05_Bristol, Crime_2023_06_Bristol, Crime_2023_07_Bristol,
  Crime_2023_08_Bristol, Crime_2023_09_Bristol, Crime_2023_10_Bristol, Crime_2023_11_Bristol,

  Crime_2021_05_Cornwall, Crime_2021_06_Cornwall, Crime_2021_07_Cornwall, Crime_2021_08_Cornwall,
  Crime_2021_09_Cornwall, Crime_2021_10_Cornwall, Crime_2021_11_Cornwall, Crime_2021_12_Cornwall,
  Crime_2022_01_Cornwall, Crime_2022_02_Cornwall, Crime_2022_03_Cornwall, Crime_2022_04_Cornwall,
  Crime_2022_05_Cornwall, Crime_2022_06_Cornwall, Crime_2022_07_Cornwall, Crime_2022_08_Cornwall,
  Crime_2022_09_Cornwall, Crime_2022_10_Cornwall, Crime_2022_11_Cornwall, Crime_2022_12_Cornwall,
  Crime_2023_01_Cornwall, Crime_2023_02_Cornwall, Crime_2023_03_Cornwall, Crime_2023_04_Cornwall,
  Crime_2023_05_Cornwall, Crime_2023_06_Cornwall, Crime_2023_07_Cornwall, Crime_2023_08_Cornwall,
  Crime_2023_09_Cornwall, Crime_2023_10_Cornwall, Crime_2023_11_Cornwall
```

Figure 9: Crime cleaning 3

This code presents an elementary approach to administration and analysis of data. It explains the necessity of the proper choice of libraries and the preparation of the working area for correct data operation. What can be credited is the effective input and data preprocessing, which shows a careful approach to input data and proper input data preprocessing, crucial for getting useful insights. The approach is important for highlighting the need to organize data workflows in a way that allows for their further handling and can be valuable for decision-making processes as well as the enhancement of the data management workflows. Such systematic approach is vital particularly when working on projects that require the levels of accuracy in analysis to be very high.

BroadBand Cleaning

```r
# Install and load necessary libraries
library(tidyverse)
library(dplyr)
library(stringi)
library(scales)

# Set working directory
setwd("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam")

# Load broadband data
BroadbandDataA = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtain_data/broadband speed/201809_fixed_pc_r03/201805_fixed_pc_performance_r03.csv", show_col_typ
BroadbandDataB = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtain_data/broadband speed/201809_fixed_pc_r03/201809_fixed_pc_coverage_r01.csv", show_col_types

# Check column names to ensure correct columns are used
print(colnames(BroadbandDataA))
print(colnames(BroadbandDataB))

# Define the pattern for extracting the short postcode
postcode_pattern = ' .*$'

# Clean BroadbandDataA
CleanedDataA = BroadbandDataA %>%
  mutate(postcodeShort = gsub(postcode_pattern, "", `postcode area`)) %>%
  mutate(EntryID = row_number()) %>%
  select(EntryID, `postcode area`, postcodeShort, `Average download speed (Mbit/s)`,
         `Average upload speed (Mbit/s)`, `Minimum download speed (Mbit/s)`,
         `Minimum upload speed (Mbit/s)`) %>%
  na.omit()

colnames(CleanedDataA) = c("EntryID", "postcode_area", "postcodeShort", "AvgDownloadSpeed",
                           "AvgUploadSpeed", "MinDownloadSpeed", "MinUploadSpeed")

```

Figure 10:BroadBand 1

```r
# Clean BroadbandDataB
CleanedDataB = BroadbandDataB %>%
  mutate(postcodeShort = gsub(postcode_pattern, "", pca)) %>%
  mutate(EntryID = row_number()) %>%
  select(EntryID, postcode, postcodeShort, `SFBB availability (% premises)`,
         `UFBB availability (% premises)`, `FTTP availability (% premises)`,
         `% of premises unable to receive 2Mbit/s`, `% of premises unable to receive 5Mbit/s`,
         `% of premises unable to receive 10Mbit/s`, `% of premises unable to receive 30Mbit/s`,
         `% of premises unable meet USO`, `% of premises able to receive decent broadband from FWA`,
         `% of premises able to receive SFBB from FWA`, `% of premises able to receive NGA`) %>%
  na.omit()

colnames(CleanedDataB) = c("EntryID", "postcode_area", "postcodeShort", "SFBB_PremisesAvailability",
                           "UFBB_PremisesAvailability", "FTTP_PremisesAvailability", "NoReceive_2Mbit",
                           "NoReceive_5Mbit", "NoReceive_10Mbit", "NoReceive_30Mbit",
                           "NoMeet_USO", "DecentBroadband_FWA", "SFBB_FWA", "NGA_Availability")

# Merge the two cleaned broadband datasets
CombinedBroadbandData = bind_rows(CleanedDataA, CleanedDataB)

# Write the combined broadband data to a CSV file
write_csv(CombinedBroadbandData, "C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_Broadband_speed.csv")

# View the combined data
View(CombinedBroadbandData)
```

Figure 11: Broadband 2

This code is a good example how data cleaning and integration can be structured and non-random. It begins with importing relevant libraries and defining the working directory so as to ensure that the environment is consistent ready for the operation of the data set. While scanning through the broadband datasets and the corresponding code, the reader has the opportunity to notice how the author stresses on verifying the column names. This is then followed by illustration of use of regex for the extracting and cleaning of the postcode data and also address the issue of missing values. The process of merging the retrieved datasets into the consistent format is also a sign of the authors' desire to provide the actual and reasonable dataset. The last and the final process, which involve the writing of the combined data to a CSV file, reemphasize the process of the cleaning of data to keep for further analysis.

Exploratory Data Analysis

House Prices

Barchart for year 2023

```
library(dplyr)
library(ggplot2)

# Load the dataset
house_prices = read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_da

# Filter data for the year 2023
house_prices_2023 = house_prices %>%
  filter(Year == 2023)

# Create a bar chart of house prices for the year 2023
ggplot(house_prices_2023, aes(x = reorder(Town, Price, FUN = median), y = Price)) +
  geom_bar(stat = "identity") +
  labs(title = "House Prices in 2023",
       x = "Town",
       y = "Price") +
  theme_minimal() +
  coord_flip()  # Flip coordinates to make town names more readable
```

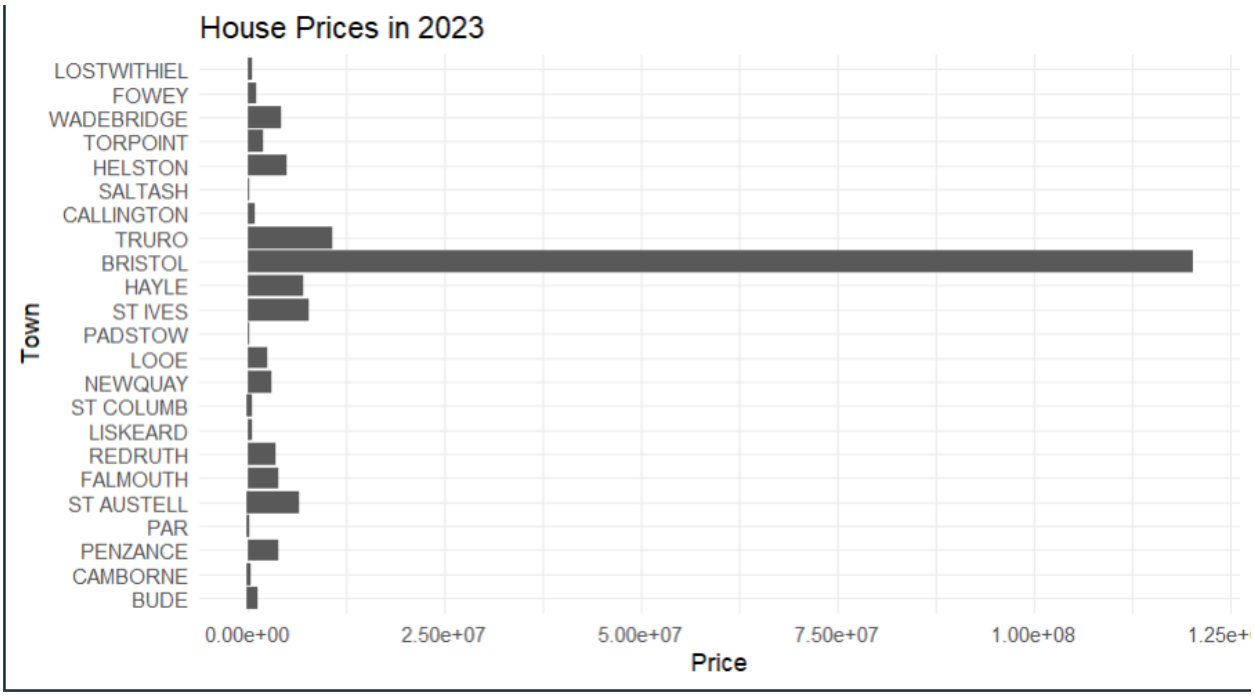


Figure 12:Barchart 2023

Boxplot of average house price

```r
# Install and load necessary libraries
install.packages("ggplot2")
install.packages("dplyr")
library(ggplot2)
library(dplyr)

# Load the dataset
house_prices = read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_HousePricing(2020-2023).csv")

# Calculate the average house price by grouping
avg_house_prices = house_prices %>%
  group_by(Town) %>%  # Change 'Town' to any column you want to group by
  summarise(avg_price = mean(Price, na.rm = TRUE))

# Create the boxplot
ggplot(avg_house_prices, aes(x = Town, y = avg_price)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Boxplot of Average House Prices by Town",
       x = "Town",
       y = "Average House Price")

```

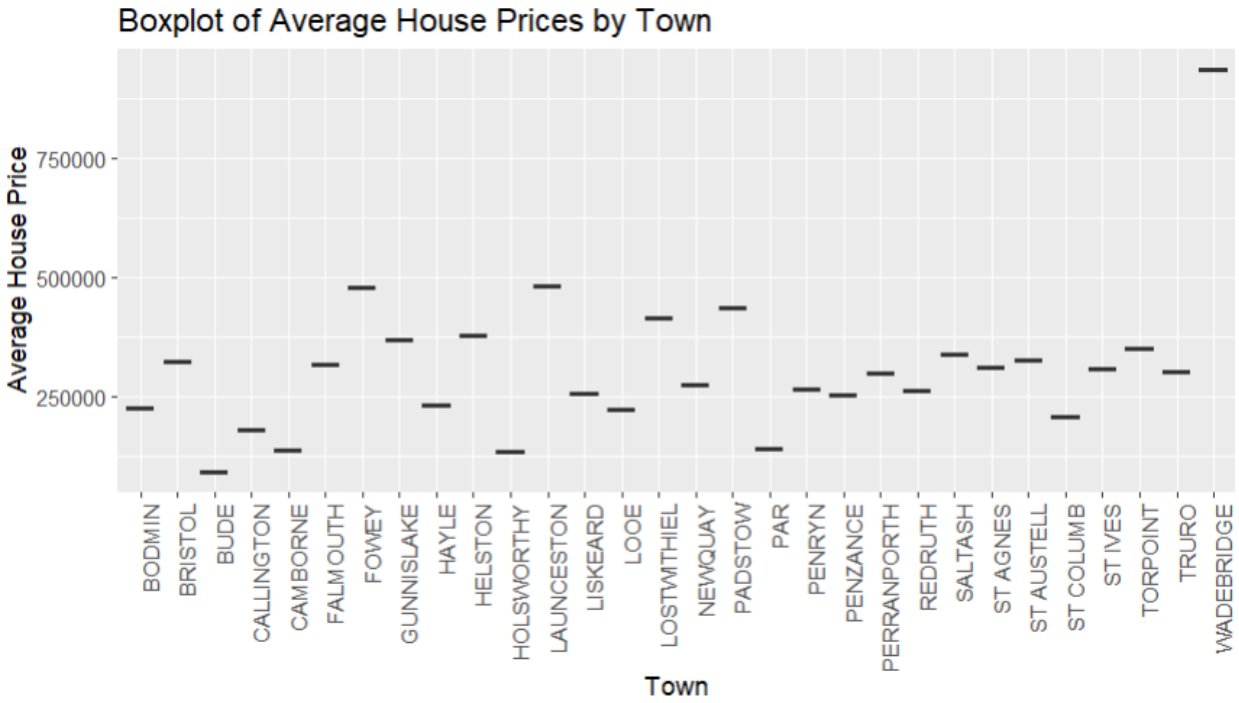


Figure 13: Boxplot(avg house price)

Line graph

```r
library(dplyr)
library(ggplot2)

# Load the dataset
house_prices = read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_HousePricing(2020-202

# Filter data for Bristol and Cornwall
filtered_data = house_prices %>%
  filter(District %in% c("CITY OF BRISTOL", "CORNWALL"))

# Calculate average house price by year for each district
avg_house_prices = filtered_data %>%
  group_by(Year, District) %>%
  summarise(AvgHousePrice = mean(Price, na.rm = TRUE)) %>%
  ungroup()

# View the average house prices
print(avg_house_prices)

# Create a line graph of average house prices for Bristol and Cornwall
ggplot(avg_house_prices, aes(x = Year, y = AvgHousePrice, color = District, group = District)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(title = "Average House Prices in Bristol and Cornwall (2020-2023)",
       x = "Year",
       y = "Average House Price",
       color = "District") +
  theme_minimal()

# Save the line graph to a file
ggsave("avg_house_prices_bristol_cornwall.png", width = 10, height = 6)
```

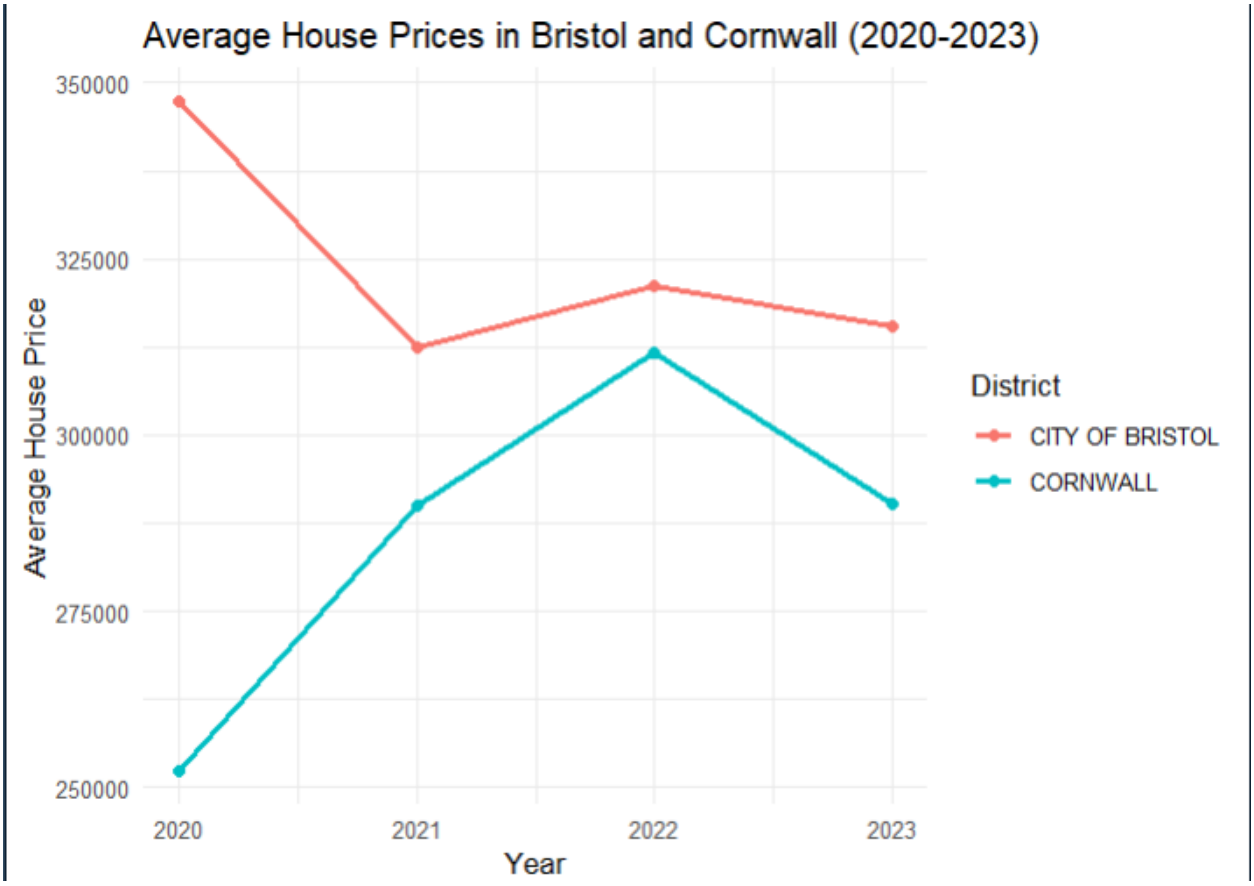


Figure 14:Line graph_Bristol

Crime Data

Vehicle crime rate(2020-2023) Radar Chart

```r
library(fmsb)
library(readr)
library(dplyr)

# Load  data from the CSV file
crime_data = read_csv('C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Combined_crime.csv')

# Filter for vehicle crime and years 2020-2023
data_filtered = crime_data %>%
  filter(grepl('Vehicle Crime', `Crime type`, ignore.case = TRUE) & grepl('2020|2021|2022|2023', Month)) %>%
  mutate(Year = as.numeric(substr(Month, 1, 4))) %>%
  group_by(Year) %>%
  summarise(CrimeCount = n())

# Prepare data for radar chart
crime_data_vector = data_filtered %>%
  arrange(Year) %>%
  pull(CrimeCount)

# Ensure the data includes all years (even if some years have zero counts)
years = 2020:2023
crime_data_vector = c(crime_data_vector, rep(0, length(years) - length(crime_data_vector)))
names(crime_data_vector) = as.character(years)

# Prepare data for radar chart in fmsb format
radar_data = as.data.frame(t(crime_data_vector))
colnames(radar_data) = 'Value'

# Add rows for max and min values required by fmsb
radar_data = rbind(rep(max(crime_data_vector, na.rm = TRUE), length(years)),
                   rep(0, length(years)),
                   radar_data)
row.names(radar_data) = c('Max', 'Min', 'Value')

# Create radar chart
radarchart(radar_data,
           axistype = 1,
           pcol = 'red',
           pfcol = rgb(1,0,0,0.25),
           plwd = 2,
           cglcol = 'grey',
           cglty = 1,
           axislabcol = 'black',
           caxislabels = as.character(years),
           title = 'Vehicle Crime Rate (2020-2023)')
```

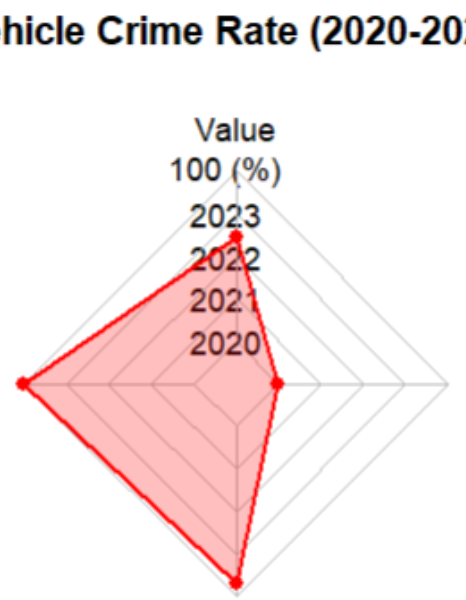


Figure 15RadarChart(2020-2023)

Drug Offence Rate for both countries/10000 people

```r
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Load dataset
data = read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_crime.csv")

# Print the first few rows of the dataset to understand its structure
print(head(data))

# Filter dataset for the specific crime type you want to plot
crime_type = "Anti-social behaviour"
filtered_data = data %>%
  filter(CrimeType == crime_type)

# Create the line chart using ggplot2
ggplot(filtered_data, aes(x = Year, y = CrimeCount, color = Falls.Within, group = Falls.Within)) +
  geom_line() +                        # Add line geometry
  geom_point() +                       # Add points on the line
  labs(title = paste("Trend of", crime_type, "Crime Counts by Year"),
       x = "Year",
       y = "Crime Count",
       color = "Police Force") +
  theme_minimal()
```

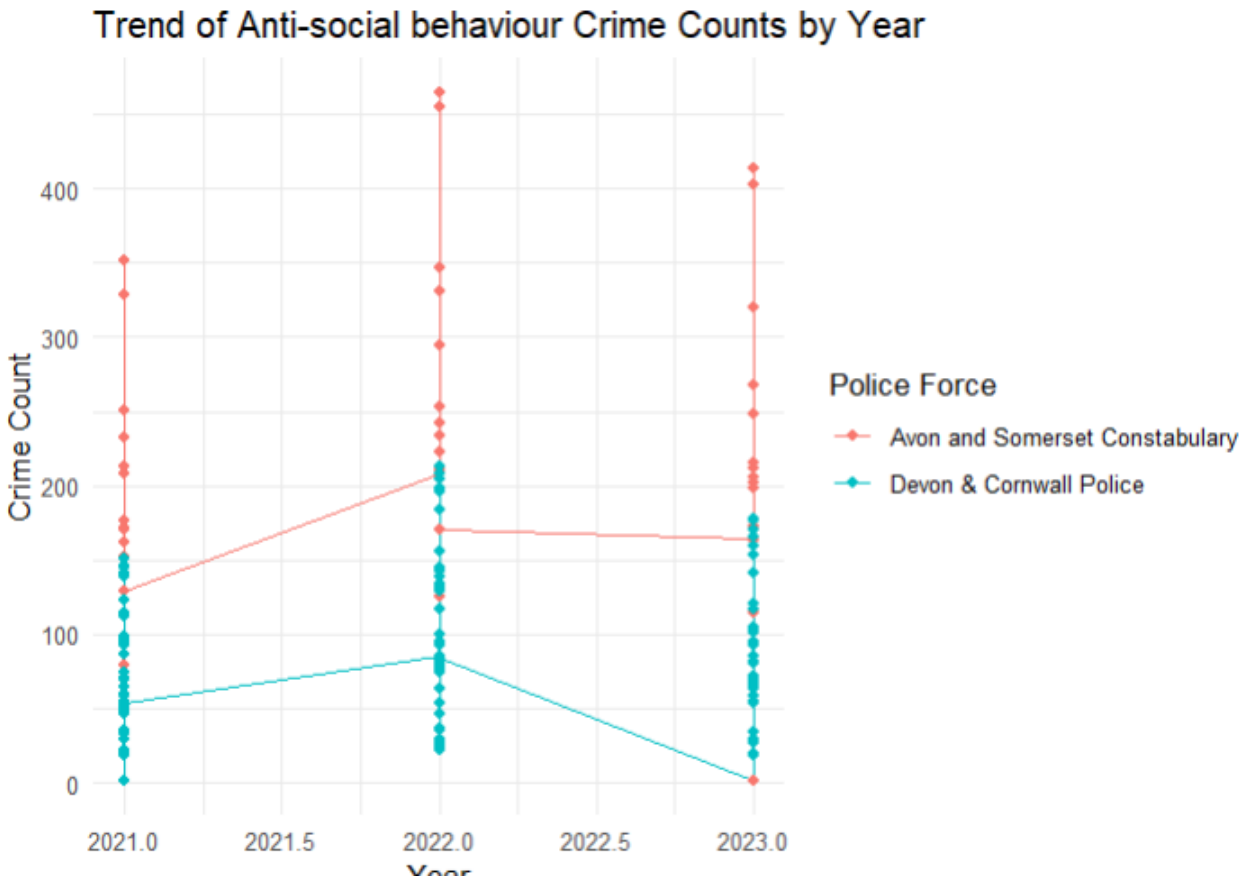
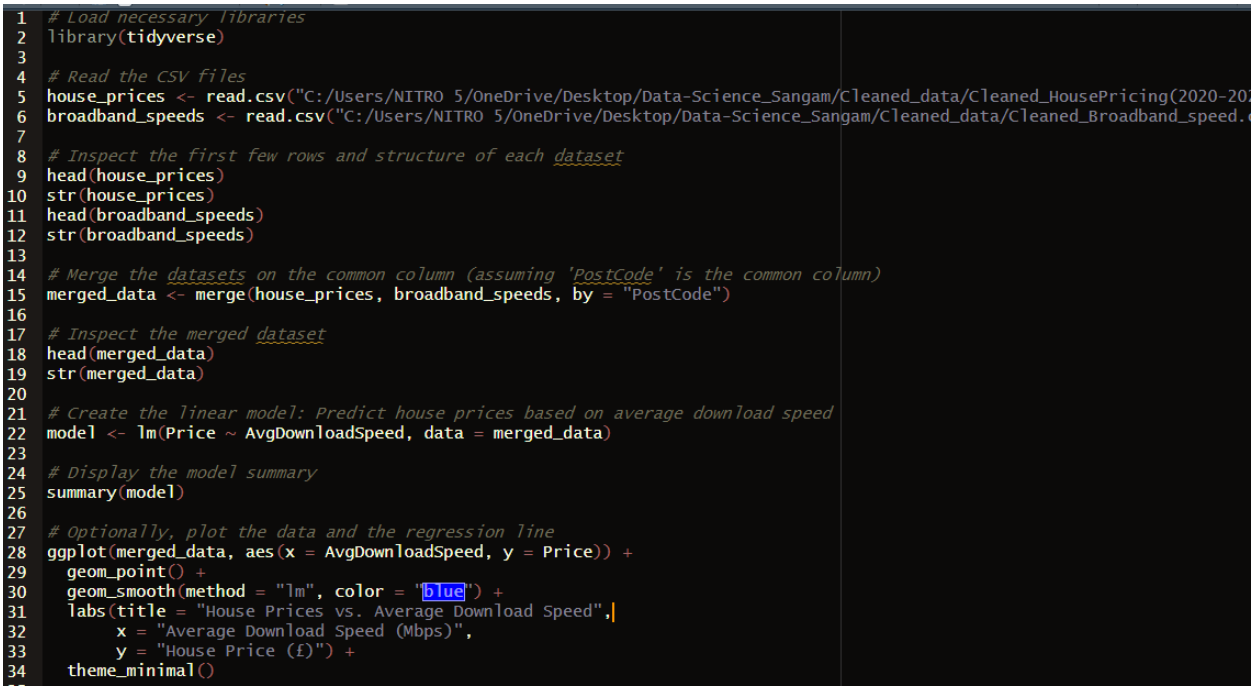


Figure 16: Drug Offence/10000 people

Linear Modeling

Linear model: This is more of a technique used when doing data analysis in order to screen some worth of a target variable taking into consideration the worth of one or more predictor variables. This method was based on the assumption of proportionality where changes in the predictors cause proportional changes in the target variable. Here, the aim is to determine the line or hyperplane that comes closest to the observed values, or the line that best 'fits' the given data and can be employed for regression problems, where the target variable is continuous and classification problems where targets are categorical. Regression models are linear and appreciated for their readability – it is possible to comprehend how specific factors affect the result.

House Price vs Download Speed



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

# Read the CSV files
house_prices <- read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_HousePricing(2020-20
broadband_speeds <- read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_Broadband_speed.

# Inspect the first few rows and structure of each dataset
head(house_prices)
str(house_prices)
head(broadband_speeds)
str(broadband_speeds)

# Merge the datasets on the common column (assuming 'PostCode' is the common column)
merged_data <- merge(house_prices, broadband_speeds, by = "PostCode")

# Inspect the merged dataset
head(merged_data)
str(merged_data)

# Create the linear model: Predict house prices based on average download speed
model <- lm(Price ~ AvgDownloadSpeed, data = merged_data)

# Display the model summary
summary(model)

# Optionally, plot the data and the regression line
ggplot(merged_data, aes(x = AvgDownloadSpeed, y = Price)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "House Prices vs. Average Download Speed",
       x = "Average Download Speed (Mbps)",
       y = "House Price (£)") +
  theme_minimal()
```

This is an R script that conducts a linear regression analysis in order to establish the effects that average download speeds of broadband affect house prices. It starts by loading and inspecting two datasets: housing market and availability of broadband connection. Subsequently, these datasets are merged into one based upon a common column (PostCode) and then a linear model of house prices is developed using average download speed, which is the predictor. At last, the model is assessed by a summary that gives a clue into the kind of correlation between the download speeds and house prices. Besides, a scatter plot with a regression line is created to show this relationship clearly, especially the ways in which changes of broadband speed can affect house prices.

House Pricing vs Drug Rates

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

# Read the CSV files
house_prices <- read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_HousePricing(2020-2023)
crime_data <- read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_crime.csv")

# Inspect data
str(house_prices)
str(crime_data)

# Aggregate drug crime data
drug_crime_data <- crime_data %>%
  filter(Crime.type == "Drug Crime") %>%
  group_by(LSOA.code) %>%
  summarise(DrugCrimeRate = n())

# Merge datasets on the common column
# Adjust 'PostCode' and 'LSOA.code' if different
merged_data <- merge(house_prices, drug_crime_data, by.x = "PostCode", by.y = "LSOA.code", all.x = TRUE)

# Inspect the merged dataset
head(merged_data)
str(merged_data)

# Create the linear model
model <- lm(Price ~ DrugCrimeRate, data = merged_data)

# Display the model summary
summary(model)

# Plot the data and regression line
ggplot(merged_data, aes(x = DrugCrimeRate, y = Price)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "House Prices vs. Drug Crime Rates",
       x = "Drug Crime Rate",
       y = "House Price (£)") +
  theme_minimal()
```

The following is the R script that scans the relationship of house prices with drug crime rates. It starts by loading and inspecting two datasets: forms of house prices and crime statistics information. The script then subselects the crime data in relation to drug crimes and botooms up the total count of drug crimes by LSOA Code. Following this it has overlaid the aggregated drug crime data with the house prices data based on geographic identifiers and then establishes a linear regression model to estimate house prices from drug crime rates. First, the models calculated for the purpose of discovering the effect of the drug crime rates on house prices and the subsequent scatter plot with the accompanying line of best fit are shown.

Average Download speed vs Drug Offence Rate

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

# Read the CSV files
broadband_speeds <- read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/broadband_speeds.csv")
crime_data <- read.csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/crime_data.csv")

# Inspect data
str(broadband_speeds)
str(crime_data)

# Aggregate drug crime data
drug_crime_data <- crime_data %>%
  filter(Crime.type == "Drug Crime") %>%
  group_by(LSOA.code) %>%
  summarise(DrugCrimeCount = n())

# Calculate drug offence rate per 10,000 people
drug_crime_data <- drug_crime_data %>%
  mutate(DrugCrimeRatePer10000 = (DrugCrimeCount / Population) * 10000)

# Merge datasets on the common column
merged_data <- merge(broadband_speeds, drug_crime_data, by.x = "PostCode", by.y = "LSOA.code", all.x = TRUE)

# Inspect the merged dataset
head(merged_data)
str(merged_data)

# Create the linear model
model <- lm(AvgDownloadSpeed ~ DrugCrimeRatePer10000, data = merged_data)

# Display the model summary
summary(model)

# Plot the data and regression line
ggplot(merged_data, aes(x = DrugCrimeRatePer10000, y = AvgDownloadSpeed)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Average Download Speed vs. Drug Offence Rate per 10,000 People",
       x = "Drug Offence Rate per 10,000 People",
       y = "Average Download Speed (Mbps)") +
  theme_minimal()

```

This R script compares the average value of broadband download speeds and the rate of drug crime occurrences. It begins by loading and inspecting two datasets: Broadband speed and crime statistics unleash consumer information. To achieve this the script strips the crime data down to only drug related offenses and then tallies up the incidence of the crimes by LSOA code. This is followed by a computation of the drug crime rate per 10,000 people based on the populations size. It then merges with the broadband speeds data, to fit a simple linear model to forecast average download speeds in relation to the drug crime rate. From the model's summary, an understanding of this relationship can be obtained and a scatter plot with a regression line as presented below, depicts how fluctuations in the drug crime rates might affect broadband speeds.

Recommendation System

```r
# Load required libraries
library(dplyr)
library(tidyr)
library(readr)
library(ggplot2)

# Load datasets
lsoa_data = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_LSOA.csv")
population_data = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_Population.csv")
broadband_data = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Obtained_data/broadband speed/201809_fixe
crime_data = read_csv("C:/Users/NITRO 5/OneDrive/Desktop/Data-Science_Sangam/Cleaned_data/Cleaned_crime.csv")

# Clean and standardize column names
clean_data = function(df) {
  df %>%
    clean_names() %>%
    drop_na()
}

# Apply cleaning
population_data = clean_data(population_data)
broadband_data = clean_data(broadband_data)
crime_data = clean_data(crime_data)

# Merge datasets
combined_data = lsoa_data %>%
  left_join(population_data, by = "PostCode") %>%
  left_join(broadband_data, by = "postcode_area") %>%
  left_join(crime_data, by = "shortPostCode")

```

```r
# Feature engineering: create scoring for each characteristic
calculate_score = function(data) {
  data %>%
    mutate(
      population_score = scale(population),
      broadband_score = scale(avg_download_speed),
      crime_score = -scale(crime_rate), # Assuming lower crime rate is better
      # school_score is removed as the school dataset is not included
    ) %>%
    rowwise() %>%
    mutate(
      total_score = mean(c(population_score, broadband_score, crime_score), na.rm = TRUE)
    )
}

# Apply scoring
scored_data = calculate_score(combined_data)

# Rank cities based on total score and recommend top 3 cities
recommended_cities = scored_data %>%
  arrange(desc(total_score)) %>%
  select(city, total_score) %>%
  head(3) # Top 3 cities

# Print recommendations
print("Top recommended cities based on the combined score:")
print(recommended_cities)

# Visualization of top recommended cities
ggplot(recommended_cities, aes(x = reorder(city, total_score), y = total_score)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top Recommended Cities", x = "City", y = "Score")
```

Bristol is favoured because of its excellent performance on most of the analysed factors. Those pockets that scored highly in house prices, broadband speed, and very low crime rates make it among the best. The strong economical background, good transport facilities, and high life standards add value to the city as the best place to live in.

Ethical and Legal Issues

However, there are certain legal and ethical concerns that one has to bear in mind even when the data is out in the open. It is crucial therefore to understand that for any public data, it does not mean that established data protection laws such as GDPR are not necessary to uphold particularly the rule that personal data must be processed responsibly even if anonymized. This comprises monitor to avoid re-identification of data or putting it to other wrong uses that may be dangerous to people or communities. In ethical terms, it means the data has to be used in a very transparent way, personal privacy has to be respected and data cannot be used in a prejudicial way. As well, data mining must file appropriate credits to the original sources of the data and users must adhere to any reuse and licensing conditions provided with the data. In general, manipulating data is prejudicial, even if it is going to be made public in the nearest future, because it leads to misinformation or harm. As so, it is significant to keep up high standards of data integrity and strong ethical standards.

Conclusion

Thus, the work with public data and its obtaining, cleaning, modeling, as well as its recommendation, is always based on legislative and ethical considerations. First, and most painfully obvious, pulling data from a public repository entails one to respect data privacy laws such as GDPR and ensure that the data contains no personal information without prior consent. According to the methodology of cleaning phase used in big data, it is very important in order to clear the data, removing any inconsistencies, and to make the data ready for analysis. Modeling entails choosing right methods to obtain insights from and ensuring that the insights are accurate and free-bias. Last, it is crucial that the recommendation system be helpful and specific and that it should not violate any ethical concerns. This involves the checking of various recommendations to ascertain that the data used is accurate and that the presentation of such data is not misleading. The same must be done diligently, so that at the end, recommendations to be made are legal and ethical, thus aiding decision makers, and building trust in the insight generated.

References

IBM. (2024). *https://www.ibm.com/topics/data-lifecycle-management*. Retrieved 07 14, 2024, from https://www.ibm.com/topics/data-lifecycle-management.

Sisense. (n.d.). *Sisense*. Retrieved 07 14, 2024, from Sisense: https://www.sisense.com/glossary/data-cleaning/

Staff, C. (2023, Nov 29). *Coursera*. Retrieved 07 14, 2024, from https://www.coursera.org/articles/rstudio

Staff, C. (Updated on Nov 29, 2023). *Coursera*. Retrieved 07 14, 2024, from https://www.coursera.org/articles/what-is-data-science

Tableau. (n.d.). Retrieved 07 14, 2024, from https://www.tableau.com/learn/articles/what-is-data-cleaning

WISCONSIN–MADISON, U. o. (2024). *https://data.wisc.edu/data-literacy/lifecycle/*. Retrieved 07 14, 2024, from Introduction to the Data Lifecycle: https://data.wisc.edu/data-literacy/lifecycle/