P4DS Summative Assignment 2

Data Analysis Project

Developing Education Equity: Analysing Positive Outlier Schools' Performance at Keystage 4 for Disadvantaged Pupils in the UK - 2022/23

Student ID: 201901718

Name: Saqib Safdar

Project Plan

### 1.1 Sources of the dataset

#### a) Department for Education (DfE)

The multiple datasets are sourced from the Department for Education's (DfE) website [1][2]. The academic year 2022-23 is the most recent data and published on 1st February 2024. Five datasets from the DfE website were used in this analysis. For each of the data sets a separate file containing the metadata is also provided. The data sets were merged based on the Unique Reference Number (URN) column for each school. Progress 8 scores are used to evaluate school performance; this is a measure of the value-added by each school based on the progress made across 8 qualifications of each pupil, using their key stage 2 results from year 6 as a baseline. The attainment 8 score (total points across 8 subjects) of each pupil is similar key stage 2 results, is compared to the national average attainment 8; the difference indicates a level of progress. A progress 8 score of 1, would indicate the student has done better by 1 grade than the national average etc. Subjects included in progress 8 include:

- English and Mathematics - both double weighted due to importance
- EBacc Subjects - three slots from subjects such as sciences, computer science, history, geography and languages
- Open Group - remaining three from other academic, arts of vocational subjects

  The DfE has data of the progress 8 score and funding for disadvantaged and non-disadvantaged students, which makes its very convenient to analyse.

[1] Department for Education. (n.d.). Explore education statistics: Data tables. Retrieved November 1, 2024, from https://explore-education-statistics.service.gov.uk/data-tables

[2] Department for Education. (n.d.). Compare the performance of schools and colleges in England. Retrieved November 10, 2024, from https://www.gov.uk/school-performance-tables

b) Index of Multiple Deprivation (IMD)

In addition to the four data sets from the DfE, rather than use funding for schools or number of disadvantaged pupils, the deprivation index for each area in the UK was downloaded from the Ministry of Housing, Communities and Local Government (MHCLG) website [3] and merged with the school information data set using the school postcode. This allows for a more detailed analysis of the relationship between school performance and socioeconomic factors which may affect the performance of disadvantaged students.

[3] Ministry of Housing, Communities & Local Government. (2019). English Indices of Deprivation 2019: Postcode Lookup. Retrieved from https://imd-by-postcode.opendatacommunities.org/imd/2019

1.2 Accuracy and Reliability of Data

The data is sourced from the Department for Education's (DfE) website and the Ministry of Housing, Communities and Local Government website. The data is accurate and reliable as it is sourced from official government sources. For the DfE, provisional and final KS4 results are provided. The key differences are the final results are quality assured for:

a) Completeness of data: results are verified

b) Accuracy of data: results are corrected for any errors or omissions

c) Usage: results are approved for use in official publications and are publicly available.

The categorise each school's socioeconomic status, the Index of Multiple Deprivation Decile (IMD) is used, which ranks each postcode in England between 1 and 10. The IMD is a composite measure of deprivation based on several other domains of deprivation including income, employment, education and health. The data is from an official government source and is therefore accurate and reliable.

1.3 Data quality, usability, and presentation

Considerations:

1. The IMD data is from 2019 and is the nearest year to the academic year 2022-23 of school performance data. When evaluating the relationship between school performance and socioeconomic factors, the socioeconomic factors may have changed in some cases since 2019. However, I will treat these are negligible changes as it the three-year period between 2019 and 2022 is relatively short.
2. As the analysis in based on school performance on a national level, including thousands of schools, I will use 'inner' joins to merge the datasets to ensure the analysis is not affected by schools which are not recognised. I will also drop any rows with missing values in key columns used for analysis.

Project Aim and Objectives

2.1 Context and motivation

Context

I have been working in education for two decades now. More recently, I have worked in MATs that are high performing and data-driven. The efficiency of a school/MAT in using its funds, together with the impact of its pedagogoical framework can be seen unsing progress 8 scores. It has been shown that by five years of age, only 57% of disadvantaged pupils achieve a good level of development compared to 74% from better off households[4]. The gap continues throughout education; in 2022 -2023, 29% of free-school mean (FSM) pupils went to university which 49.8% of non FSM pupils progressed to university. [5].

Motivation

Several motivations underpin this analysis:

1. In a recent letter from the secretary of state for education, five priuorities were set out for higher education proviers, to top of which is : "Play a stronger role in expanding access and improving outcomes for disadvantaged students. The gap in outcomes from higher education between disadvantaged students and others is unacceptably large and is widening, with participation from disadvantaged students in decline for the first time in two decades." [6]

2. Enhancing Education Practice: Some secondary schools are able to close the gap and give students from disadvantaged backgrounds better opportunities to progress to university. This data science investigations aim to identify outlier schools who outperform what is expected from them.

3. Justifying School Funding: Given the various avenues of funding data available, e.g. pupil premium for disadvantaged pupils, school-led tutoring funding, and the results for FSM and non FSM students, progress 8 and Eng - Maths, the efficiency of schools in using their funds can be evaluated. I can also examine if their is a correlation between progress 8 of disadvantaged and the level of funding schools receive to support them.

4. Understand demographic factos: Analysis of school demographics, e.g. gender, school type, local authority, can help to undertand their influence on school performance.

5. Socioeconomic factors: The relationship between school performance and socioeconomic factors such as deprivation can be explored by merging the school performance data with the deprivation index for each area in the UK. Other factors such percentage of disadvantaged students, percentage of non-disadvantaged students, pupil premium funding, percentage of disadvantaged students achieving grades 9-5 in English and Maths, can also be explored.

6. Impact of MAT: Group level management, collaboration and performance, particularly on outlier schools, can be explored to determine if their is a correlation between school performance and the type of MAT they belong to.

[4] Institute for Fiscal Studies. (2024, May). *The past and future of UK health spending*. Retrieved from https://www.ifs.org.uk/publications/health-spending-report

[5] Busby, E. (2024, October 24). Gap between private and state school pupils going to top universities widens. *The Independent*. Retrieved from https://www.independent.co.uk/news/uk/gap-england-department-for-education-government-data-b2634966.html

[6] Phillipson, B. (2024, November 4). *Letter from the Secretary of State for Education*. Department for Education.

2.2 Specific Objective(s)

1. Evaluate National Disparities in Educational Performance Between Advantaged and Disadvantaged Pupils

Using comprehensive datasets from the Department for Education (DfE) and the Ministry of Housing, Communities, and Local Government (MHCLG), conduct a detailed national-level analysis of the performance gap in key metrics, including Progress 8, Attainment 8, and English and Mathematics scores. This objective will involve merging, cleaning and validating data, before statistical analysis is conducted to determine the level of gap between disadvantaged and advantaged pupils

2. Identify and analyse outlier schools nationally for progress 8 scores for disadvantaged pupils and investigate contributing factors.

This objective will conduct more in depth statistical analysis to identify positive outlier schools with progress-8 scores for disadvantaged pupils. Further analysis on quantitative and categorical factors will be conducted to determine the influence of socio-economic indicators, such as the Index of Multiple Deprivation and demographics of the school.

3. Identify and evaluate the top performing multi-academy trusts in supporting disadvantaged pupils.

This objective will conduct statistical analysis to identify top performing multi-academy trusts and their success in closing the disadvantage gap. Hypothesis testing and regression analysis will be conducted to determine the level of impact of potential factors.

System Design

Architecture

Key Components: Descriptions, Purpose and Challenges

The following data sets will be downloaded and used from the DfE website.

1. DfE data set 1: KS4 school performance 2022-23

- Purpose: This provides information on the academic performance of each school and provides categories relating to advantage and disadvantage pupils in progress 8, attainment-8 and in EBACC subjects English and Mathematics. The description of each field is given below.

- Key fields used for analysis:

```
  • URN (Unique Reference Number)
  • Average Attainment 8 score
  • Average Progress 8 score
  • Percentage of disadvantaged students
  • Percentage of non-disadvantaged students
  • Percentage of disadvantaged students achieving grades 9-5 in
English and Maths
  • Percentage of non-disadvantaged students achieving grades 9-5
in English and Maths
  • Attainment 8 score for non-disadvantaged students
  • Progress 8 score for non-disadvantaged students
  • Attainment 8 score for disadvantaged students
  • Progress 8 score for disadvantaged students
  • Progress 8 score in Maths for disadvantaged students
  • Progress 8 score in English for disadvantaged students
  • Progress 8 score in Maths for non-disadvantaged students
  • Progress 8 score in English for non-disadvantaged students
```

2. Data set 2: School information - provides information on the demographics of each school.

Purpose: The purpose of this data set it to determine school demographics such as gender, Ofsted rating etc, and other such categorical columns which can be used to determine potential impact on students' progress.

Key fields used in analysis:

```
    - URN - Unique Reference Number for the school
    - Local Authority Name (LANAME) - Name of the local authority the
school belongs to
    - Local Authority Code (LA) - Numeric code identifying the local
authority
    - School Type - Type of school (e.g. Academy, Community School,
etc.)
    - Minor Group - More detailed classification of school type
    - Gender - Whether the school is mixed, boys only or girls only
    - Ofsted Rating - Latest Ofsted inspection rating for the school
```

3. Data set 3: School funding

Purpose: Provides information on the various types of funding for each school.

Key fields used in analysis:

```
    - School UKPRN: Unique ID number for each school provider
    - School URN: Another unique ID number for each school
    - Time Period: The academic year the funding is for
    - FSM Funding: Money given to schools for students eligible for
free school meals
    - FSM6 Funding: Money given for students who were eligible for
free school meals in the past 6 years
    - Pupil Premium: Extra funding given to help disadvantaged
students
    - Pupil Premium Pupils: Number of students who qualify for pupil
premium funding
    - School-led Tutoring Funding: Money given to schools to provide
extra tutoring
    - Total Funding: The total amount of funding received by the
school
```

4. DfE data set 4: Multi Academy Trust (MAT) performance

Purpose: provides information of performance for each Multi-Academy Trust (MAT)

Key fields used from MAT performance data:

```
    - Trust Name: Name of the Multi-Academy Trust
    - Trust UID: Unique identification number for the trust
    - Trust ID: Alternative ID code for the trust
    - Number of Institutions: Number of schools in the trust
    - Total Pupils: Total number of pupils across all schools in the
trust
    - Average Attainment 8 Score: Average attainment score across 8
subjects for the trust
    - Average Progress 8 Score: Average progress score showing value
added by trust
    - Time Period: Academic year the data is from
```

5. Data set 5: Academies membership

Purpose: provides information on which MAT each school belongs to allowing external data such as to be linked to schools through their postcode and then to URN.

Key fields used in analysis:

```
    - URN - Unique Reference Number for the school
    - Group UID - Unique identifier for school group/trust
    - Group ID - Alternative identifier for school group/trust
    - Establishment Name - Official name of the school
    - Group Name - Name of the school group/trust
    - Postcode - Postcode of the school
```

6. MHCLG Data - Index of Multiple Deprivation (IMD)

Purpose: In addition to the five data sets from the DfE, the deprivation index for each area in the UK will be downloaded from the Ministry of Housing, Communities and Local Government (MHCLG) website and merged with the school information data set using the school postcode. This allows for a more detailed analysis of the relationship between school performance and socioeconomic factors which may affect the performance of disadvantaged students, as compared to say relying solely on funding data or percentage of disadvantaged pupils.

Key columns used for analysis:

```
    - Postcode
    - Index of Multiple Deprivation Decile
```

7. Metadata

Purpose: To identify the appropriate columns for analysis from the DfE data sets, the metadata will be used. Each of the DfE data sets lists above will have a corresponding meta-data file.

8. Classes

Purpose: To optimise the processes above, functions will be organised in classes

Challenges: Key challenges will be selecting and identify the appropriate columns from the DfE data sets as the data set a very large number of fields. The meta data file will be needed to be used to identify the code and description for each field. The code used would then need to be re-written in most cases so it is clear to the non-technical reader what the field stands for, while retaining a format suitable for a data column in python. Another challenge will be ensuring data types are in the correct format for quantitative analysis. Where needed, feature engineering would need to be employed for new fields which may be required such as pupil premium funding per pupil. Another challenge will be in connecting the index of multiple deprivation IMD with each school, as the MHCLG is independent to the DfE, and will not include the school URN which is what will be used to combined the DfE data.

Pipeline and Workflow

The pipline starts by setting up necessary functions and classes for data loading, wrangling and cleaning.

- Determine necessary functions and classes needed for the project
- Data Collection: Collect 2022-23 school and MAT performance data from the Department for Education (DfE); this includes the five data sets listed above and their meta files.
- Data Collection: Collect data from the inistry of Housing, Communities and Local Government (MHCLG) website; Index of Multiple Deprivation Decile (IMD)
- Meta Data: Read the metadata for each data set to understand the data and variables. Create a dictionary of code and description.
- Using the meta-data fields extract the key columns for analysis from the data files.

- Data Integration: Merge the data sets based on the Unique Reference Number (URN) column for each school.

- Data Cleaning: Clean the data to remove any missing values and inconsistencies. Convert data to appropriate data types.

- Nomenclature: Determine new naming convention using meta-data dictionary and assign this to the data files.

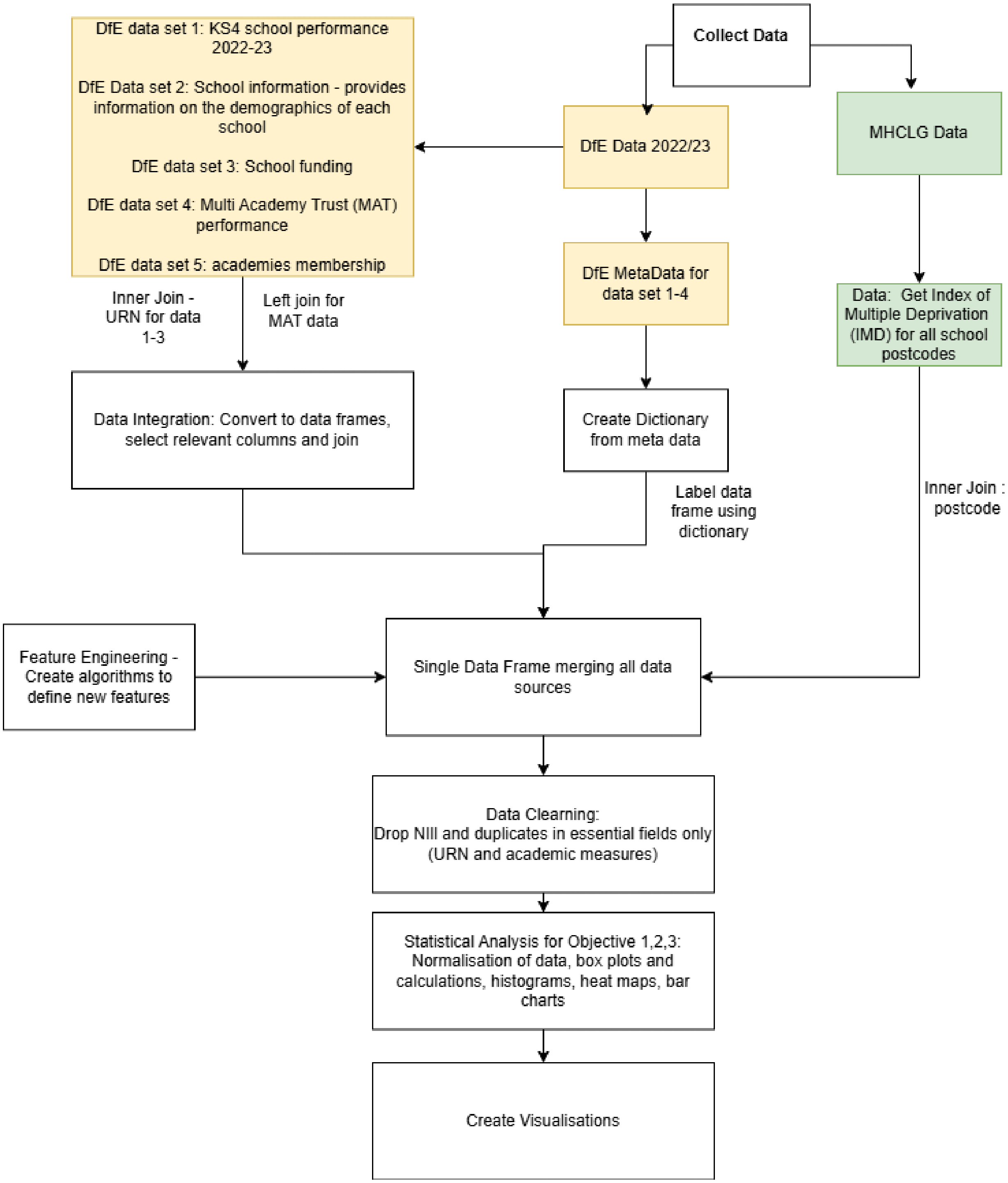
- Feature Engineering: Create algorithms to define new features e.g pupil premium funding per pupil, key stage4_maths_gaps, keystage4 English gap and progress 8 gap between advantaged and disadvantaged pupils.

- Data Integration: Socioeconomic Indexing - incorporate the Index of Multiple Deprivation Decile (IMD) for each postcode to the school information data set.

- Statistical Analysis and Modelling: Conduct statistical analysis to determine advantage - disadvantage gap, identify outlier schools and top 10 performing MATs. Evaluate the impact of socioeconomic and other factors on school performance

- Visualisation: Create visualisations to present the findings.

- Conclusion: Summarise the findings and relate them to the original objectives.


DfE data set 1: KS4 school performance 2022-23
DfE Data set 2: School information - provides information on the demographics of each school
DfE data set 3: School funding
DfE data set 4: Multi Academy Trust (MAT) performance
DfE data set 5: academies membership
Collect Data
DfE Data 2022/23
MHCLG Data
DfE MetaData for data set 1-4
Data: Get Index of Multiple Deprivation (IMD) for all school postcodes
Inner Join - URN for data 1-3
Left join for MAT data
Data Integration: Convert to data frames, select relevant columns and join
Create Dictionary from meta data
Label data frame using dictionary
Inner Join : postcode
Feature Engineering - Create algorithms to define new features
Single Data Frame merging all data sources
Data Clearning: Drop NIll and duplicates in essential fields only (URN and academic measures)
Statistical Analysis for Objective 1,2,3: Normalisation of data, box plots and calculations, histograms, heat maps, bar charts
Create Visualisations

For a more dynamics view, workflow diagram can also be viewed here

Processing Modules and Algorithms

The following modules and algrorithms will be required in a number of instances and therefore defined and written within a class:

- **Class: DataWrangler** - load data from CSV, Excel file or existing pands data frame

Methods:

- Load a csv file into a pandas dataframe using load_csv method
- Load an excel file into a pandas dataframe using load_excel method
- Create a dictionary from a dataframe using make_dictionary method
- Rename columns in a dataframe using a dictionary using column_rename method - substitute original column names with descriptive names in a dictionary or list
- Convert percentage strings in specified columns to float values using convert_percentage_columns method
- Retrieve specific columns from a given dataframe using a set of URNs using get_school_details method
- Plot boxplots, histograms, heatmaps and scatter plots to visualise the data
- Write code to generate summary statistics of the boxplots

Program Code

Libraries

I will begin by by importing the needed libraries for converting data to dataframes, conducting calculations and visualisations

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import os
from sklearn.preprocessing import StandardScaler
```

Classes

A class called Dataloader will be created to manage all core functions related to data loading and wrangling. This includes:

- load_csv
- load_excel
- make_dictionary
- column_rename
- convert_percentages_column

Details of the functions purpose, paramters and return value can be read in the doctrings below the function defintion

```
class DataWrangler:
    def __init__(self, file_path=None, dataframe=None):
        """
        Initialise the DataWrangler with a file path or an existing DataFrame.

        Parameters:
        - file_path (str): The path to the data file (CSV or Excel).
        - dataframe (pd.DataFrame): An existing DataFrame to work with.
        """
        if dataframe is not None:
            self.df = dataframe.copy()
            print("DataWrangler initialised with the provided DataFrame.")
        elif file_path is not None:
            self.file_path = file_path
            self.df = None

            if self.file_path.endswith("csv"):
                self.load_csv()
            elif self.file_path.endswith(".xlsx"):
                self.load_excel()
            else:
                raise ValueError("Unsupported file format. Please provide a CSV or Excel file.")
        else:
            raise ValueError("Either file_path or dataframe must be provided.")


    def load_csv(self):
        """
        Load a CSV file into a pandas DataFrame.
        """
        try:
            self.df = pd.read_csv(self.file_path, encoding='latin1')
            print(f"CSV file loaded successfully from {self.file_path}")
        except FileNotFoundError as e:
            print(f"Error loading CSV file: {e}")


    def load_excel(self):
        """
        Load an Excel file into a pandas DataFrame.
        """
        try:
            self.df = pd.read_excel(self.file_path)
            print(f"Excel file loaded successfully from
{self.file_path}")
        except FileNotFoundError as e:
            print(f"Error loading Excel file: {e}")
            self.df = None


    def make_dictionary(self, key_column: str, value_column: str):
        """
        Create a dictionary from two columns of the DataFrame.

        Parameters:
        - key_column (str): The column to use as the dictionary key.
        - value_column (str): The column to use as the dictionary
value.

        Returns:
        - dict: A dictionary mapping keys to values.
        """
        try:
            return dict(zip(self.df[key_column],
self.df[value_column]))
        except KeyError as e:
            print(f"Error: Key column not found in DataFrame: {e}")
            return None

    def column_rename(self, column_dict: dict):
        """
        Rename columns in the DataFrame using a provided dictionary.

        Parameters:
        - column_dict (dict): A dictionary mapping original column
names to new names.

        Returns:
        - pd.DataFrame: The DataFrame with renamed columns.
        """
        self.df = self.df.rename(columns=column_dict)
        print("Columns renamed successfully.")
        return self.df


    def convert_percentage_columns(self, columns):
        """
        Remove % sign form colums .

        Parameters:
        - columns (list): List of column names to convert.
        Returns:
        - pd.DataFrame: The DataFrame with converted columns.
        """
        for col in columns:
            # Remove '%' and convert to float
            self.df[col] = self.df[col].astype(str).str.replace('%',
'')
            print(f"Column '{col}' converted")
        return self.df


    def get_school_details(self, urn_set, columns):
        """
        Retrieve essential school details for specified URNs and
columns.

        Parameters:
        - urn_set (set): A set of URNs (Unique Reference Numbers) for
schools.
        - columns (list): List of columns to include in the output.

        Returns:
        - pd.DataFrame: A DataFrame containing the specified details.
        """
        return self.df[self.df['URN'].isin(urn_set)][columns]
```

Load Data

I will now load and examine the five data files from the DfE as pandas data frames and do a quick inspection using .head(),info(), describe(). To avoid repetition, I will do a more thorough analyse of data types and missing values later, once all the data is combined.

```
# Beginning with MAT data:
ks4_mat_performance = DataWrangler('data/2022-2023_england_ks4-mats-
performance.csv')
ks4_mat_performance.df.head()
```

```
CSV file loaded successfully from data/2022-2023_england_ks4-mats-
performance.csv

   TIME_PERIOD TIME_IDENTIFIER      TRUST_GROUP_TYPE  \
0       202223    AcademicYear  Multi-academy trusts
1       202223    AcademicYear  Multi-academy trusts
2       202223    AcademicYear  Multi-academy trusts
3       202223    AcademicYear  Multi-academy trusts
4       202223    AcademicYear  Multi-academy trusts

                          TRUST_NAME  TRUST_UID TRUST_ID  \
0  ACTIVATE LEARNING EDUCATION TRUST      15710  TR02786
1                          ACER TRUST      15720  TR01414
2             RED KITE LEARNING TRUST      15727  TR00969
3                 CONSILIUM ACADEMIES      15728  TR00082
4          BATLEY MULTI ACADEMY TRUST      15729  TR00147

   TRUST_COMPANIES_HOUSE_NUMBER  TRUST_UKPRN         TRUST_LEADREGION
\
0                       8707909     10060613               South East

1                       9591931     10060976               South East

2                       7523507     10054307  Yorkshire and the Humber

3                       9495671     10061209               North East

4                       7732537     10059240  Yorkshire and the Humber


                               INSTITUTIONS_MATPTINC  ...  \
0          139268;141111;142024;145155;145945;146375  ...
1                               142104;143984;144008  ...
2                        136497;138304;141883;146247  ...
3  138314;143059;143845;144199;144200;144937;1449...  ...
4                               137424;137487;142406  ...

                                  INSTITUTIONS_INMAT  NUMINST_INMAT  \
0          139268;141111;142024;145155;145945;146375              6
1                               142104;143984;144008              3
2                        136497;138304;141883;146247              4
3  138314;143059;143845;144199;144200;144937;1449...              8
4                               137424;137487;142406              3

   NUMINST_CONVERTER_INMAT  NUMINST_SPONSOR_INMAT  NUMINST_FREE_INMAT
\
0                        0                      2                   0

1                        3                      0                   0

2                        2                      1                   1

3                        5                      3                   0

4                        1                      1                   1


   NUMINST_STUDIO_INMAT  NUMINST_UTC_INMAT  TPUP_INMAT
PFSM6CLA1A_INMAT  \
0                     0                  4         647
21%
1                     0                  0         548
```

```
14%
2                     0                  0         810
22%
3                     0                  0        1150
40%
4                     0                  0         520
32%

   PNOTFSM6CLA1A_INMAT
0                  79%
1                  86%
2                  78%
3                  60%
4                  68%

[5 rows x 67 columns]
```

```python
# Keystage 4 school performance data:
ks4_school_performance = DataWrangler('data/2022-
2023_england_ks4final.csv')
ks4_school_performance.df.head()
```

```
CSV file loaded successfully from data/2022-2023_england_ks4final.csv

C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\3754428246.py:32:
DtypeWarning: Columns
(52,54,56,109,110,323,324,325,326,327,328,329,330,331,332,497,498,507,
508) have mixed types. Specify dtype option on import or set
low_memory=False.
  self.df = pd.read_csv(self.file_path, encoding='latin1')

   ï»¿RECTYPE    LEA   ESTAB       URN
SCHNAME  \
0            1  201.0  6007.0  100003.0            City of London
School
1            1  201.0  6005.0  100001.0  City of London School for
Girls
2            1  201.0  6000.0  100544.0               David Game
College
3            4  201.0     NaN       NaN
NaN
4            1  202.0  4285.0  100053.0           Acland Burghley
School

  SCHNAME_AC                   ADDRESS1  ADDRESS2 ADDRESS3
TOWN  ...  \
0        NaN  107 Queen Victoria Street       NaN      NaN
London  ...
1        NaN          St Giles' Terrace  Barbican      NaN
London  ...
```

```
2        NaN            31 Jewry Street    London      NaN
NaN  ...
3        NaN                        NaN       NaN      NaN
NaN  ...
4        NaN              Burghley Road       NaN      NaN
London  ...

  TAVENT_GFSM6CLA1A_PTQ_EE  TAVENT_GNFSM6CLA1A_PTQ_EE  \
0                       NP                         NP
1                       NP                         NP
2                       NP                         NP
3                      NaN                        NaN
4                      7.0                        8.1

  TAVENT_GFSM6CLA1A_21_PTQ_EE TAVENT_GNFSM6CLA1A_21_PTQ_EE  \
0                         NaN                          NaN
1                         NaN                          NaN
2                         NaN                          NaN
3                         NaN                          NaN
4                         NaN                          NaN

   TAVENT_GFSM6CLA1A_22_PTQ_EE  TAVENT_GNFSM6CLA1A_22_PTQ_EE  \
0                           NP                            NP
1                           NP                            NP
2                           NP                            NP
3                          NaN                           NaN
4                            7                           8.2

  TTOTENT_E_TOTAL_PTQ_EE TTOTENT_E_COVID_IMPACTED_PTQ_EE  \
0                  569.0                             3.0
1                  606.0                             5.0
2                   53.0                             1.0
3                    NaN                             NaN
4                 1397.0                             0.0

  PTOTENT_E_COVID_IMPACTED_PTQ_EE P8_BANDING
0                              1%        NaN
1                              1%        NaN
2                              2%        NaN
3                             NaN        NaN
4                              0%    Average

[5 rows x 515 columns]

#School demographics data:
school_demographics = DataWrangler('data/2022-
2023_england_school_information.csv')
school_demographics.df.rename(columns={'ï»¿URN': 'URN'}, inplace=True)
#correction to URN column name
school_demographics.df.head()
```

```
CSV file loaded successfully from data/2022-
2023_england_school_information.csv

      URN          LANAME   LA  ESTAB  LAESTAB  \
0  100000  City of London  201   3614  2013614
1  100001  City of London  201   6005  2016005
2  100002  City of London  201   6006  2016006
3  100003  City of London  201   6007  2016007
4  100008          Camden  202   2019  2022019

                           SCHNAME                     STREET
LOCALITY  \
0                The Aldgate School         St James's Passage  Duke's
Place
1  City of London School for Girls          St Giles' Terrace
Barbican
2       St Paul's Cathedral School               2 New Change
NaN
3            City of London School  107 Queen Victoria Street
NaN
4            Argyle Primary School           Tonbridge Street
NaN

  ADDRESS3    TOWN  ... ISPRIMARY ISSECONDARY ISPOST16 AGELOW AGEHIGH
GENDER  \
0      NaN  London  ...         1           0        0    3.0    11.0
Mixed
1      NaN  London  ...         1           1        1    7.0    18.0
Girls
2      NaN  London  ...         1           1        0    4.0    13.0
Mixed
3      NaN  London  ...         1           1        1   10.0    18.0
Boys
4      NaN  London  ...         1           0        0    3.0    11.0
Mixed

             RELCHAR          ADMPOL  OFSTEDRATING  OFSTEDLASTINSP
0  Church of England  Not applicable   Outstanding      13-06-2024
1                NaN       Selective           NaN             NaN
2  Church of England  Not applicable           NaN             NaN
3                NaN  Not applicable           NaN             NaN
4     Does not apply  Not applicable          Good      05-10-2022

[5 rows x 26 columns]

# School funding data:
school_funding =
DataWrangler('data/20230126_school_level_data_csv.csv')
school_funding.df.rename(columns={'ï»¿time_period': 'time_period'},
inplace=True) #correction to time period column
school_funding.df.head()
```

```
CSV file loaded successfully from
data/20230126_school_level_data_csv.csv

   time_period time_identifier geographic_level country_code
country_name  \
0       202223  Financial year           School    E92000001
England
1       202223  Financial year           School    E92000001
England
2       202223  Financial year           School    E92000001
England
3       202223  Financial year           School    E92000001
England
4       202223  Financial year           School    E92000001
England

   old_la_code new_la_code               la_name  school_ukprn
school_urn  \
0          301   E09000002  Barking and Dagenham      10000222
101247
1          301   E09000002  Barking and Dagenham      10000527
101241
2          301   E09000002  Barking and Dagenham      10071309
101202
3          301   E09000002  Barking and Dagenham      10071301
101231
4          301   E09000002  Barking and Dagenham      10029207
136028

   ...  allocation_per_pupil pupil_premium pupil_premium_pupils  \
0  ...           6658.346870       291560                   296
1  ...           6817.784488       492993                   501
2  ...           5524.276364       209135                   151
3  ...           5542.356295       153735                   111
4  ...           7127.415584       511215                   519

  universal_infant_free_school_meals_grant pe_&_sport_premium  \
0                                        x                  x
1                                        x                  x
2                                    44741              20720
3                                    29116              19540
4                                        x                  x

  pe_&_sport_premium_pupils  coronavirus_recovery_premium_funding  \
0                         x                                 81972
1                         x                                148598
2                       472                                 21895
```

```
3                       354                                  17927
4                         x                                 164595

   School_led_tutoring_funding  schools_supplementary_grant
total_funding
0                        49248                       249881
8542828.0
1                        84024                       389143
13420859.0
2                        24138                        80619
3439599.0
3                        18117                        62142
2633909.0
4                        91017                       288410
9836214.0

[5 rows x 57 columns]
```

```
#Academies data which connect URN code to postcode
academies_membership =
DataWrangler('data/academiesmatmembership20220901.csv')
academies_membership.df.head()
```

```
CSV file loaded successfully from
data/academiesmatmembership20220901.csv

        URN DfE Number  EstablishmentNumber  Establishment UKPRN  LA
(code)  \
0  136683.0   840/4054               4054.0           10033436.0
840.0
1  140594.0   936/2341               2341.0           10044809.0
936.0
2  136354.0   925/3510               3510.0           10032221.0
925.0
3  137036.0   381/5404               5404.0           10034739.0
381.0
4  140214.0   925/2016               2016.0           10043499.0
925.0

       LA (name)  Group UID Group ID  TypeOfEstablishment (code)  \
0  County Durham     1128.0  TR02582                        34.0
1         Surrey     1128.0  TR02582                        34.0
2   Lincolnshire     2044.0  TR00261                        34.0
3     Calderdale     2044.0  TR00261                        28.0
4   Lincolnshire     2044.0  TR00261                        28.0

  TypeOfEstablishment (name)  ...  Group Town Group County  Group
Postcode  \
0          Academy converter  ...      Harrow         99.0         HA2
9DS
```

```
1          Academy converter  ...      Harrow          99.0          HA2
9DS
2          Academy converter  ...      Bourne          99.0         PE10
9EP
3        Academy sponsor led  ...      Bourne          99.0         PE10
9EP
4        Academy sponsor led  ...      Bourne          99.0         PE10
9EP

  HeadTitle (name)  HeadFirstName HeadLastName  Ofsted Last Inspection
Date  \
0     Not recorded  Felicity Jane        Smith                   2018-
10-31
1              Mrs          Clare       Spires                   2019-
02-27
2              Mrs          Sarah        Moore                   2021-
07-08
3              Mrs      Rosalinda    Wood-Ives                   2022-
03-17
4              Mrs          Sarah        Moore                   2017-
04-27

  OfstedRating (name) Predecessor Establishment Successor
Establishment
0                 NaN                       YES
YES
1                 NaN                       YES
NO
2                 NaN                       YES
NO
3                 NaN                       YES
NO
4                 NaN                        NO
NO

[5 rows x 57 columns]
```

Load Metadata and Make Dictionaries

I will now load the meta-data for each data file. To determine what each column in the data files means, I will create a dictionary using the make_dictionary function defined as part of the DataWrangler class. The meta data is labeled after each associated data file with the addition of 'meta' at the end.

```
ks4_mat_performance_meta = DataWrangler('data/ks4-mats-
performance_meta.csv')
ks4_mat_performance_dict =
DataWrangler.make_dictionary(ks4_mat_performance_meta, 'Metafile
heading','Metafile description')
```

```
ks4_mat_performance_dict
```

```
CSV file loaded successfully from data/ks4-mats-performance_meta.csv
```

```
{'TIME_PERIOD': nan,
 'TIME_IDENTIFIER': nan,
 'TRUST_GROUP_TYPE': 'Trust type',
 'TRUST_NAME': 'Trust name',
 'TRUST_UID': 'Trust Unique identifier',
 'TRUST_ID': 'Trust Identifier',
 'TRUST_COMPANIES_HOUSE_NUMBER': 'Trust companies house number',
 'TRUST_UKPRN': 'Trust UK provider reference number',
 'TRUST_LEADREGION': 'Trust lead region',
 'INSTITUTIONS_MATPTINC': 'URNs, included in performance measures',
 'NUMINST_MATPTINC': 'Number of academies in the trust, included in
performance measures',
 'NUMINST_CONVERTER_MATPTINC': 'Number of converter academies,
included in performance measures',
 'NUMINST_SPONSOR_MATPTINC': 'Number of sponsor-led academies,
included in performance measures',
 'NUMINST_FREE_MATPTINC': 'Number of free school - mainstream
academies, included in performance measures',
 'NUMINST_STUDIO_MATPTINC': 'Number of free school - studio schools,
included in performance measures',
 'NUMINST_UTC_MATPTINC': 'Number of free school - UTCs, included in
performance measures',
 'NUMINST_FSM6CLA1A_MATPTINC': 'Number of academies with disadvantaged
pupils, included in performance measures',
 'NUMINST_3_MATPTINC': 'Number of academies that have been in the
trust for 3 years, included in performance measures',
 'NUMINST_4_MATPTINC': 'Number of academies that have been in the
trust for 4 years, included in performance measures',
 'NUMINST_5PLUS_MATPTINC': 'Number of academies that have been in the
trust for 5 years or more, included in performance measures',
 'TPUP_MATPTINC': 'Number of pupils at the end of ks4, included in
performance measures',
 'KS2ASS_MATPTINC': 'KS4 cohort average KS2 Scaled Score (average of
English reading and maths), included in performance measures',
 'PFSM6CLA1A_MATPTINC': '% of pupils at the end of ks4 who are
disadvantaged, included in performance measures',
 'PNOTFSM6CLA1A_MATPTINC': '% of pupils at the end of ks4 who are not
disadvantaged, included in performance measures',
 'PEALGRP2_MATPTINC': '% of pupils at the end of ks4 with English as
additional language (EAL), included in performance measures',
 'PSEN_ALL4_MATPTINC': '% of pupils at the end of ks4 with special
educational needs (SEN) including those with or without an Education,
health and care (EHC) plan, included in performance measures',
 'ATT8SCR_WGTAVG': 'Average Attainment 8 score per pupil at the end of
```

```
KS4, weighted average',
 'P8MEACOV': '% of pupils at the end of ks4 included in Progress 8
measure',
 'P8MEA_WGTAVG': 'Progress 8 measure after adjustment for extreme
scores, weighted average',
 'P8CILOW': 'Progress 8 lower 95% confidence interval for adjusted
average',
 'P8CIUPP': 'Progress 8 upper 95% confidence interval for adjusted
average',
 'PTL2BASICS_95_WGTAVG': '% of pupils at the end of KS4 achieving
strong 9-5 passes in both English and mathematics GCSEs , weighted
average',
 'EBACCAPS_WGTAVG': 'Average EBacc APS score per pupil at the end of
KS4, weighted average',
 'PTEBACC_95_WGTAVG': '% of pupils at the end of KS4 achieving the
English Baccalaureate with 9-5 passes, weighted average',
 'PTEBACC_94_WGTAVG': '% of pupils at the end of KS4 achieving the
English Baccalaureate with 9-4 passes, weighted average',
 'PTEBACC_E_PTQ_EE_WGTAVG': '% of pupils at the end of KS4 with
entries in all English Baccalaureate subject areas, weighted average',
 'ATT8SCR_WGTAVG_FSM6CLA1A': 'Average Attainment 8 score per
disadvantaged pupil at the end of KS4, weighted average',
 'P8MEACOV_FSM6CLA1A': '% of disadvantaged pupils at the end of ks4
included in Progress 8 measure',
 'P8MEA_WGTAVG_FSM6CLA1A': 'Progress 8 measure after adjustment for
extreme scores for disadvantaged pupils, weighted average',
 'P8CILOW_FSM6CLA1A': 'Progress 8 lower 95% confidence interval for
adjusted average for disadvantaged pupils',
 'P8CIUPP_FSM6CLA1A': 'Progress 8 upper 95% confidence interval for
adjusted average for disadvantaged pupils',
 'PTL2BASICS_95_WGTAVG_FSM6CLA1A': '% of disadvantaged pupils at the
end of KS4 achieving strong 9-5 passes in both English and mathematics
GCSEs , weighted average',
 'EBACCAPS_WGTAVG_FSM6CLA1A': 'Average EBacc APS score per
disadvantaged pupil at the end of KS4, weighted average',
 'PTEBACC_95_WGTAVG_FSM6CLA1A': '% of disadvantaged pupils at the end
of KS4 achieving the English Baccalaureate with 9-5 passes, weighted
average',
 'PTEBACC_94_WGTAVG_FSM6CLA1A': '% of disadvantaged pupils at the end
of KS4 achieving the English Baccalaureate with 9-4 passes, weighted
average',
 'PTEBACC_E_PTQ_EE_WGTAVG_FSM6CLA1A': '% of disadvantaged pupils at
the end of KS4 with entries in all English Baccalaureate subject
areas, weighted average',
 'ATT8SCR_WGTAVG_NFSM6CLA1A': 'Average Attainment 8 score per non-
disadvantaged pupil at the end of KS4, weighted average',
 'P8MEACOV_NFSM6CLA1A': '% of non-disadvantaged pupils at the end of
ks4 included in Progress 8 measure',
 'P8MEA_WGTAVG_NFSM6CLA1A': 'Progress 8 measure after adjustment for
```

```
extreme scores for non-disadvantaged pupils, weighted average',
 'P8CILOW_NFSM6CLA1A': 'Progress 8 lower 95% confidence interval for
adjusted average for non-disadvantaged pupils',
 'P8CIUPP_NFSM6CLA1A': 'Progress 8 upper 95% confidence interval for
adjusted average for non-disadvantaged pupils',
 'PTL2BASICS_95_WGTAVG_NFSM6CLA1A': '% of non-disadvantaged pupils at
the end of KS4 achieving strong 9-5 passes in both English and
mathematics GCSEs , weighted average',
 'EBACCAPS_WGTAVG_NFSM6CLA1A': 'Average EBacc APS score per non-
disadvantaged pupil at the end of KS4, weighted average',
 'PTEBACC_95_WGTAVG_NFSM6CLA1A': '% of non-disadvantaged pupils at the
end of KS4 achieving the English Baccalaureate with 9-5 passes,
weighted average',
 'PTEBACC_94_WGTAVG_NFSM6CLA1A': '% of non-disadvantaged pupils at the
end of KS4 achieving the English Baccalaureate with 9-4 passes,
weighted average',
 'PTEBACC_E_PTQ_EE_WGTAVG_NFSM6CLA1A': '% of non-disadvantaged pupils
at the end of KS4 with entries in all English Baccalaureate subject
areas, weighted average',
 'P8_BANDING': 'Progress 8 banding shown on performance tables
website',
 'INSTITUTIONS_INMAT': 'URNs, including mainstream academies not in
performance measures',
 'NUMINST_INMAT': 'Number of academies in the trust, including those
not in performance measures',
 'NUMINST_CONVERTER_INMAT': 'Number of converter academies, including
those not in performance measures',
 'NUMINST_SPONSOR_INMAT': 'Number of sponsor-led academies, including
those not in performance measures',
 'NUMINST_FREE_INMAT': 'Number of free school - mainstream academies,
including those not in performance measures',
 'NUMINST_STUDIO_INMAT': 'Number of free school - studio schools,
including those not in performance measures',
 'NUMINST_UTC_INMAT': 'Number of free school - UTCs, including those
not in performance measures',
 'TPUP_INMAT': 'Number of pupils at the end of KS4, including those
not in performance measures',
 'PFSM6CLA1A_INMAT': '% of pupils at the end of KS4 who are
disadvantaged, including those not in performance measures',
 'PNOTFSM6CLA1A_INMAT': '% of pupils at the end of KS4 who are not
disadvantaged, including those not in performance measures'}
```

```
school_demographics_meta = DataWrangler('data\
school_information_meta.csv')
school_demographics_dict =
DataWrangler.make_dictionary(school_demographics_meta,'Field Name',
'Description')
school_demographics_dict
```

```
CSV file loaded successfully from data\school_information_meta.csv
```

```
<>:1: SyntaxWarning: invalid escape sequence '\s'
<>:1: SyntaxWarning: invalid escape sequence '\s'
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\3609157132.py:1:
SyntaxWarning: invalid escape sequence '\s'
  school_demographics_meta = DataWrangler('data\
school_information_meta.csv')
```

```
{'URN': 'School unique reference number',
 'LANAME': 'Local authority name',
 'LA': 'Local authority number',
 'ESTAB': 'Establishment number',
 'LAESTAB': 'DfE number',
 'SCHNAME': 'School name',
 'STREET': 'School address (1)',
 'LOCALITY': 'School address (2)',
 'ADDRESS3': 'School address (3)',
 'TOWN': 'School town',
 'POSTCODE': 'School postcode',
 'SCHSTATUS': 'School open / closed status',
 'OPENDATE': 'Open date of school (if opened on or after 1st September
2022)',
 'CLOSEDATE': 'Date the school closed',
 'MINORGROUP': 'Type of school / college eg maintained school',
 'SCHOOLTYPE': 'School Type eg Voluntary Aided school',
 'ISPRIMARY': 'Does the school provide primary education? ( 0 = No, 1
= Yes)',
 'ISSECONDARY': 'Does the school provide secondary education? ( 0 =
No, 1 = Yes)',
 'ISPOST16': 'Does the school provide post 16 education? (  0 = No, 1
= Yes)',
 'AGELOW': 'Lowest age of entry',
 'AGEHIGH': 'Highest age of entry',
 'GENDER': "Indicates whether it's a mixed or single sex school",
 'RELCHAR': 'Religious character',
 'ADMPOL': 'Admissions Policy',
 'OFSTEDRATING': 'Ofsted rating',
 'OFSTEDLASTINSP': 'Ofsted last inspection date'}
```

```python
ks4_school_performance_meta = DataWrangler('data/ks4_meta.xlsx') #
this is originally in .xlsx format
school_performance_dict =
DataWrangler.make_dictionary(ks4_school_performance_meta, 'Metafile
heading','Metafile description')
#school_performance_dict['URN'] = 'URN'  # keep the URN column as it
is as this will be used to merge the dataframes
school_performance_dict
```

```
Excel file loaded successfully from data/ks4_meta.xlsx
```

```
{'RECTYPE': 'Record type (1=mainstream school; 2=special school;
4=local authority; 5=National (all schools); 7=National (maintained
schools))',
 'LEA': 'Local authority code (see separate list of local authorities
and their codes)',
 'ESTAB': 'Establishment number',
 'URN': 'School Unique Reference Number',
 'SCHNAME': 'School name',
 'SCHNAME_AC': 'School now known as (used if the school has converted
to an academy on or after 12 Sept 2022)',
 'ADDRESS1': 'School address (1)',
 'ADDRESS2': 'School address (2)',
 'ADDRESS3': 'School address (3)',
 'TOWN': 'School town',
 'PCODE': 'School postcode',
 'TELNUM': 'School telephone number',
 'PCON_CODE': 'Parliamentary constituency code',
 'PCON_NAME': 'Parliamentary constituency name',
 'CONTFLAG': "Contingency flag - school results 'significantly
affected'. This field is zero for all schools.",
 'ICLOSE': 'Closed school flag (0=open; 1=closed; 2=pending closure)',
 'NFTYPE': 'School type (see separate list of abbreviations used in
the tables)',
 'RELDENOM': 'School religious character',
 'ADMPOL': 'School admissions policy (self-declared by schools on
Edubase)',
 'ADMPOL_PT': 'School admissions policy - new definition from 2019',
 'EGENDER': 'School gender of entry',
 'FEEDER': 'Indicates whether school is a feeder school for sixth form
centre/consortia (0=No; 1=Yes)',
 'TABKS2': 'Indicates whether school is published in the primary
school (key stage 2) performance tables (0=No; 1=Yes)',
 'TAB1618': 'Indicates whether school is published in the school and
college (16-18) performance tables (0=No; 1=Yes)',
 'AGERANGE': 'Age range',
 'TOTPUPS': 'Number of pupils on roll (all ages)',
 'NUMBOYS': 'Total boys on roll (including part-time pupils)',
 'NUMGIRLS': 'Total girls on roll (including part-time pupils)',
 'TPUP': 'Number of pupils at the end of key stage 4',
 'BPUP': 'Number of boys at the end of key stage 4',
 'PBPUP': '% of pupils at the end of key stage 4 who are boys',
 'GPUP': 'Number of girls at the end of key stage 4',
 'PGPUP': '% of pupils at the end key stage 4 who are girls',
 'KS2ASS': 'KS4 cohort average KS2 Scaled Score (average of English
reading and maths)',
 'TPRIORLO': 'Number of pupils at the end of key stage 4 with low
prior attainment at the end of key stage 2',
 'PTPRIORLO': '% of pupils at the end of key stage 4 with low prior
attainment at the end of key stage 2',
 'TPRIORAV': 'Number of pupils at the end of key stage 4 with middle
```

```
prior attainment at the end of key stage 2',
 'PTPRIORAV': '% of pupils at the end of key stage 4 with middle prior
attainment at the end of key stage 2',
 'TPRIORHI': 'Number of pupils at the end of key stage 4 with high
prior attainment at the end of key stage 2',
 'PTPRIORHI': '% of pupils at the end of key stage 4 with high prior
attainment at the end of key stage 2',
 'TFSM6CLA1A': 'Number of disadvantaged pupils at the end of key stage
4',
 'PTFSM6CLA1A': '% of pupils at the end of key stage 4 who are
disadvantaged',
 'TNOTFSM6CLA1A': 'Number of non-disadvantaged pupils at the end of
key stage 4',
 'PTNOTFSM6CLA1A': '% of pupils at the end of key stage 4 who are not
disadvantaged',
 'TEALGRP2': 'Number of pupils at the end of key stage 4 with English
as additional language (EAL)',
 'PTEALGRP2': '% of pupils at the end of key stage 4 with English as
additional language (EAL)',
 'TEALGRP1': 'Number of pupils at the end of key stage 4 with English
as their first language',
 'PTEALGRP1': '% of pupils at the end of key stage 4 with English as
their first language',
 'TEALGRP3': 'Number of pupils at the end of key stage 4 whose first
language is unclassified',
 'PTEALGRP3': '% of pupils at the end of key stage 4 whose first
language is unclassified',
 'TNMOB': 'Number of pupils at the end of key stage 4 who are non-
mobile',
 'PTNMOB': '% of pupils at the end of key stage 4 who are non-mobile',
 'SENE4': 'Number of pupils at the end of key stage 4 with special
educational needs (SEN) with an Education, health and care (EHC)
plan',
 'PSENE4': '% of pupils at the end of key stage 4 with special
educational needs (SEN) with an Education, health and care (EHC)
plan',
 'SEN_ALL4': 'Number of pupils at the end of key stage 4 with special
educational needs (SEN) including those with or without an Education,
health and care (EHC) plan',
 'PSEN_ALL4': '% of pupils at the end of key stage 4 with special
educational needs (SEN) including those with or without an Education,
health and care (EHC) plan',
 'SENK4': 'Number of pupils at the end of key stage 4 with special
educational needs (SEN) without an Education, health and care (EHC)
plan',
 'PSENK4': '% of pupils at the end of key stage 4 with special
educational needs (SEN) without an Education, health and care (EHC)
plan',
 'TOTATT8': 'Total sum of Attainment 8 scores',
 'ATT8SCR': 'Average Attainment 8 score per pupil',
 'TOTATT8ENG': 'Total sum of Attainment 8 scores for English element',
 'ATT8SCRENG': 'Average Attainment 8 score per pupil for English
element',
 'TOTATT8MAT': 'Total sum of Attainment 8 scores for mathematics
element',
 'ATT8SCRMAT': 'Average Attainment 8 score per pupil for mathematics
element',
 'TOTATT8EBAC': 'Total sum of Attainment 8 scores for EBacc element',
 'ATT8SCREBAC': 'Average Attainment 8 score per pupil for EBacc
element',
 'TOTATT8OPEN': 'Total sum of Attainment 8 scores for open element',
 'ATT8SCROPEN': 'Average Attainment 8 score per pupil for open
element',
 'TOTATT8OPENG': 'Total sum of Attainment 8 scores for open element -
GCSE only',
 'ATT8SCROPENG': 'Average Attainment 8 score per pupil for open
element - GCSE only',
 'TOTATT8OPENNG': 'Total sum of Attainment 8 scores for open element -
non-GCSE only',
 'ATT8SCROPENNG': 'Average Attainment 8 score per pupil for open
element - non-GCSE only',
 'AVGEBACFILL': 'Average number of EBacc slots filled in Attainment 8
per pupil',
 'AVGOPENFILL': 'Average number of Open slots filled in Attainment 8
per pupil',
 'P8PUP': 'Number of pupils included in Progress 8 measure',
 'TP8ADJ': 'Number of pupils who have had P8 score adjusted in
average',
 'P8MEACOV': '% of pupils at the end of key stage 4 included in
Progress 8 measure',
 'P8MEA': 'Progress 8 measure after adjustment for extreme scores',
 'P8CILOW': 'Progress 8 lower 95% confidence interval for adjusted
average',
 'P8CIUPP': 'Progress 8 upper 95% confidence interval for adjusted
average',
 'P8MEA_ORIG': 'Progress 8 measure based on unadjusted pupil scores',
 'P8CILOW_ORIG': 'Progress 8 lower 95% confidence interval for
unadjusted average',
 'P8CIUPP_ORIG': 'Progress 8 upper 95% confidence interval for
unadjusted average',
 'P8MEAENG': 'Progress 8 measure for English element',
 'P8MEAENG_CILOW': 'Lower 95% confidence interval for Progress 8
English element',
 'P8MEAENG_CIUPP': 'Upper 95% confidence interval for Progress 8
English element',
 'P8MEAMAT': 'Progress 8 measure for mathematics element',
 'P8MEAMAT_CILOW': 'Lower 95% confidence interval for Progress 8 maths
element',
 'P8MEAMAT_CIUPP': 'Upper 95% confidence interval for Progress 8 maths
element',
 'P8MEAEBAC': 'Progress 8 measure for EBacc element',
 'P8MEAEBAC_CILOW': 'Lower 95% confidence interval for Progress 8
EBacc element',
 'P8MEAEBAC_CIUPP': 'Upper 95% confidence interval for Progress 8
EBacc element',
 'P8MEAOPEN': 'Progress 8 measure for open element',
 'P8MEAOPEN_CILOW': 'Lower 95% confidence interval for Progress 8 open
element',
 'P8MEAOPEN_CIUPP': 'Upper 95% confidence interval for Progress 8 open
element',
 'PTL2BASICS_94': '% of pupils achieving standard 9-4 passes in both
English and mathematics GCSEs ',
 'PTL2BASICS_95': '% of pupils achieving strong 9-5 passes in both
English and mathematics GCSEs ',
 'TOTEBACCAPS': 'Total EBacc APS score per pupil',
 'EBACCAPS': 'Average EBacc APS score per pupil',
 'EBACCAPS_FSM6CLA1A': 'Average EBacc APS score per disadvantaged
pupil',
 'EBACCAPS_NFSM6CLA1A': 'Average EBacc APS score per non-disadvantaged
pupil',
 'EBACCAPS_LO': 'Average EBacc APS score per pupil with low prior
attainment',
 'EBACCAPS_MID': 'Average EBacc APS score per pupil with middle prior
attainment',
 'EBACCAPS_HI': 'Average EBacc APS score per pupil with high prior
attainment',
 'EBACCAPS_EAL': 'Average EBacc APS score per pupil for whom English
is an additional language',
 'EBACCAPS_GIRLS': 'Average EBacc APS score per girl',
 'EBACCAPS_BOYS': 'Average EBacc APS score per boy',
 'EBACCAPS_NMOB': 'Average EBacc APS score per non-mobile pupil',
 'EBACCAPS_21': 'Average EBacc APS score per pupil in 2021',
 'EBACCAPS_FSM6CLA1A_21': 'Average EBacc APS score per disadvantaged
pupil in 2021',
 'EBACCAPS_NFSM6CLA1A_21': 'Average EBacc APS score per non-
disadvantaged pupil in 2021',
 'EBACCAPS_22': 'Average EBacc APS score per pupil in 2022',
 'EBACCAPS_FSM6CLA1A_22': 'Average EBacc APS score per disadvantaged
pupil in 2022',
 'EBACCAPS_NFSM6CLA1A_22': 'Average EBacc APS score per non-
disadvantaged pupil in 2022',
 'TEBACC_E_PTQ_EE': 'Number of key stage 4 pupils with entries in all
English Baccalaureate subject areas',
 'PTEBACC_E_PTQ_EE': '% of key stage 4 pupils with entries in all
English Baccalaureate subject areas',
 'PTEBACC_94': '% of pupils achieving the English Baccalaureate with
9-4 passes',
 'PTEBACC_95': '% of pupils achieving the English Baccalaureate with
9-5 passes',
 'TEBACENG_E_PTQ_EE': 'Number of pupils entering the English
```

```
Baccalaureate English subject area',
 'PTEBACENG_E_PTQ_EE': '% of pupils entering the English Baccalaureate
English subject area',
 'TEBACMAT_E_PTQ_EE': 'Number of pupils entering the English
Baccalaureate Maths subject area',
 'PTEBACMAT_E_PTQ_EE': '% of pupils entering the English Baccalaureate
Maths subject area',
 'TEBAC2SCI_E_PTQ_EE': 'Number of pupils entering the English
Baccalaureate Science subject area',
 'PTEBAC2SCI_E_PTQ_EE': '%  of pupils entering the English
Baccalaureate Science subject area',
 'TEBACHUM_E_PTQ_EE': 'Number of pupils entering the English
Baccalaureate Humanities subject area',
 'PTEBACHUM_E_PTQ_EE': '%  of pupils entering the English
Baccalaureate Humanities subject area',
 'TEBACLAN_E_PTQ_EE': 'Number of pupils entering the English
Baccalaureate Language subject area',
 'PTEBACLAN_E_PTQ_EE': '%  of pupils entering the English
Baccalaureate Language subject area',
 'PTEBACENG_94': '% of pupils achieving the EBacc English subject area
with a standard 9-4 pass',
 'PTEBACENG_95': '% of pupils achieving the EBacc English subject area
with a strong 9-5 pass ',
 'PTEBACMAT_94': ' % of pupils achieving the EBacc Maths subject area
with a standard 9-4 pass ',
 'PTEBACMAT_95': ' % of pupils achieving the EBacc Maths subject area
with a strong 9-5 pass ',
 'PTEBAC2SCI_94': ' % of entered pupils achieving the EBacc Science
subject area with a 9-4 pass',
 'PTEBAC2SCI_95': ' % of entered pupils achieving the EBacc Science
subject area with a 9-5 pass',
 'PTEBACHUM_94': ' % of entered pupils achieving the EBacc Humanities
subject area with a 9-4 pass',
 'PTEBACHUM_95': ' % of entered pupils achieving the EBacc Humanities
subject area with a 9-5 pass',
 'PTEBACLAN_94': ' % of entered pupils achieving the EBacc Language
subject area with a 9-4 pass',
 'PTEBACLAN_95': ' % of entered pupils achieving the EBacc Language
subject area with a 9-5 pass',
 'SCIVAPUP_PTQ_EE': 'Number of pupils included in English
Baccalaureate Science Value Added measure ',
 'SCIVACOV_PTQ_EE': 'Coverage of the English Baccalaureate Science
Value Added indicators of those who entered for science',
 'HUMVAPUP_PTQ_EE': 'Number of pupils included in English
Baccalaureate Humanities Value Added measure ',
 'HUMVACOV_PTQ_EE': 'Coverage of the English Baccalaureate Humanities
Value Added indicators of those who entered for humanities',
 'LANVAPUP_PTQ_EE': 'Number of pupils included in English
Baccalaureate Language Value Added measure ',
 'LANVACOV_PTQ_EE': 'Coverage of the English Baccalaureate Language
```

```
Value Added indicators of those who entered for languages',
 'SCIVAMEA_PTQ_EE': 'English Baccalaureate Science Value Added
measure',
 'SCIVALOW_PTQ_EE': 'English Baccalaureate Science Value Added lower
95% confidence limit',
 'SCIVAUPP_PTQ_EE': 'English Baccalaureate Science Value Added upper
95% confidence limit',
 'HUMVAMEA_PTQ_EE': 'EBacc Humanities VA measure',
 'HUMVALOW_PTQ_EE': 'English Baccalaureate Humanities Value Added
lower 95% confidence limit',
 'HUMVAUPP_PTQ_EE': 'English Baccalaureate Humanities Value Added
upper 95% confidence limit',
 'LANVAMEA_PTQ_EE': 'English Baccalaureate Languages Value Added
measure',
 'LANVALOW_PTQ_EE': 'English Baccalaureate Languages Value Added lower
95% confidence limit',
 'LANVAUPP_PTQ_EE': 'English Baccalaureate Languages Value Added upper
95% confidence limit',
 'TEBACENG_94': 'Number of pupils achieving EBacc English subject area
with a standard 9-4 pass ',
 'TEBACENG_95': 'Number of pupils achieving EBacc English subject area
with a strong 9-5 pass ',
 'TEBACMAT_94': 'Number of pupils achieving EBacc Maths subject area
with a standard 9-4 pass ',
 'TEBACMAT_95': 'Number of pupils achieving EBacc Maths subject area
with a strong 9-5 pass ',
 'TEBAC2SCI_94': 'Number of pupils achieving EBacc Science subject
area with a 9-4 pass',
 'TEBAC2SCI_95': 'Number of pupils achieving EBacc Science subject
area with a 9-5 pass',
 'TEBACHUM_94': 'Number of pupils achieving EBacc Humanities subject
area with a 9-4 pass',
 'TEBACHUM_95': 'Number of pupils achieving EBacc Humanities subject
area with a 9-5 pass',
 'TEBACLAN_94': 'Number of pupils achieving EBacc Language subject
area with a 9-4 pass',
 'TEBACLAN_95': 'Number of pupils achieving EBacc Language subject
area with a 9-5 pass',
 'TEBACC91': 'Number of pupils achieving the English Baccalaureate at
grades 9-1',
 'PTEBACC91': ' % of pupils achieving the English Baccalaureate at
grades 9-1 ',
 'TEBACENG91': 'Number of pupils achieving EBacc English subject area
at grade 9-1',
 'PTEBACENG91': '% of pupils achieving the EBacc English subject area
at grade 9-1',
 'TEBACMAT91': 'Number of pupils achieving EBacc Maths subject area at
grade 9-1',
 'PTEBACMAT91': ' % of pupils achieving the EBacc Maths subject area
at grade 9-1',
 'TEBAC2SCI91': 'Number of pupils achieving EBacc Science subject area
with grades 9-1',
 'PTEBAC2SCI91': ' % entered pupils achieving the EBacc Science
subject area with grades 9-1',
 'TEBACHUM91': 'Number of pupils achieving EBacc Humanities subject
area with grades 9-1',
 'PTEBACHUM91': ' % entered pupils achieving the EBacc Humanities
subject area with grades 9-1',
 'TEBACLAN91': 'Number of pupils achieving EBacc Language subject area
with grades 9-1',
 'PTEBACLAN91': ' % of entered pupils achieving the EBacc Language
subject area with grades 9-1',
 'ATT8SCR_FSM6CLA1A': 'Average Attainment 8 score per disadvantaged
pupil',
 'P8PUP_FSM6CLA1A': 'Number of disadvantaged pupils in Progress 8
measure',
 'TP8ADJ_FSM6CLA1A': 'Number of disadvantaged pupils in progress
measure with adjusted scores',
 'P8MEA_FSM6CLA1A': 'Adjusted Progress 8 measure - disadvantaged
pupils',
 'P8CILOW_FSM6CLA1A': 'Adjusted Progress 8 lower 95% confidence
interval - disadvantaged pupils',
 'P8CIUPP_FSM6CLA1A': 'Adjusted Progress 8 upper 95% confidence
interval - disadvantaged pupils',
 'P8MEA_FSM6CLA1A_ORIG': 'Unadjusted Progress 8 measure -
disadvantaged pupils',
 'P8CILOW_FSM6CLA1A_ORIG': 'Unadjusted Progress 8 lower 95% confidence
interval - disadvantaged pupils',
 'P8CIUPP_FSM6CLA1A_ORIG': 'Unadjusted Progress 8 upper 95% confidence
interval - disadvantaged pupils',
 'ATT8SCR_NFSM6CLA1A': 'Average Attainment 8 score per non-
disadvantaged pupil',
 'P8PUP_NFSM6CLA1A': 'Number of non-disadvantaged pupils in Progress 8
measure',
 'TP8ADJ_NFSM6CLA1A': 'Number of non-disadvantaged pupils in progress
measure with adjusted scores',
 'P8MEA_NFSM6CLA1A': 'Adjusted Progress 8 measure - non-disadvantaged
pupils',
 'P8CILOW_NFSM6CLA1A': 'Progress 8 lower 95% confidence interval -
non-disadvantaged pupils',
 'P8CIUPP_NFSM6CLA1A': 'Progress 8 upper 95% confidence interval -
non-disadvantaged pupils',
 'P8MEA_NFSM6CLA1A_ORIG': 'Unadjusted Progress 8 measure - non-
disadvantaged pupils',
 'P8CILOW_NFSM6CLA1A_ORIG': 'Unadjusted Progress 8 lower 95%
confidence interval - non-disadvantaged pupils',
 'P8CIUPP_NFSM6CLA1A_ORIG': 'Unadjusted Progress 8 upper 95%
confidence interval - non-disadvantaged pupils',
 'ATT8SCRENG_FSM6CLA1A': 'Average Attainment 8 score per disadvantaged
pupil for English element',
 'P8MEAENG_FSM6CLA1A': 'Progress 8 measure for English element -
disadvantaged pupils',
 'P8MEAENG_CILOW_FSM6CLA1A': 'Lower 95% confidence interval for
Progress 8 English element for disadvantaged pupils',
 'P8MEAENG_CIUPP_FSM6CLA1A': 'Upper 95% confidence interval for
Progress 8 English element for disadvantaged pupils',
 'ATT8SCRMAT_FSM6CLA1A': 'Average Attainment 8 score per disadvantaged
pupil for mathematics element',
 'P8MEAMAT_FSM6CLA1A': 'Progress 8 measure for maths element -
disadvantaged pupils',
 'P8MEAMAT_CILOW_FSM6CLA1A': 'Lower 95% confidence interval for
Progress 8 maths element for disadvantaged pupils',
 'P8MEAMAT_CIUPP_FSM6CLA1A': 'Upper 95% confidence interval for
Progress 8 maths element for disadvantaged pupils',
 'ATT8SCREBAC_FSM6CLA1A': 'Average Attainment 8 score per
disadvantaged pupil for EBacc element',
 'P8MEAEBAC_FSM6CLA1A': 'Progress 8 measure for EBacc element -
disadvantaged pupils',
 'P8MEAEBAC_CILOW_FSM6CLA1A': 'Lower 95% confidence interval for
Progress 8 EBacc element for disadvantaged pupils',
 'P8MEAEBAC_CIUPP_FSM6CLA1A': 'Upper 95% confidence interval for
Progress 8 EBacc element for disadvantaged pupils',
 'ATT8SCROPEN_FSM6CLA1A': 'Average Attainment 8 score per
disadvantaged pupil for open element',
 'P8MEAOPEN_FSM6CLA1A': 'Progress 8 measure for open element -
disadvantaged pupils',
 'P8MEAOPEN_CILOW_FSM6CLA1A': 'Lower 95% confidence interval for
Progress 8 open element for disadvantaged pupils',
 'P8MEAOPEN_CIUPP_FSM6CLA1A': 'Upper 95% confidence interval for
Progress 8 open element for disadvantaged pupils',
 'ATT8SCRENG_NFSM6CLA1A': 'Average Attainment 8 score per non-
disadvantaged pupil for English element',
 'P8MEAENG_NFSM6CLA1A': 'Progress 8 measure for English element - non-
disadvantaged pupils',
 'P8MEAENG_CILOW_NFSM6CLA1A': 'Lower 95% confidence interval for
Progress 8 English element for non-disadvantaged pupils',
 'P8MEAENG_CIUPP_NFSM6CLA1A': 'Upper 95% confidence interval for
Progress 8 English element for non-disadvantaged pupils',
 'ATT8SCRMAT_NFSM6CLA1A': 'Average Attainment 8 score per non-
disadvantaged pupil for mathematics element',
 'P8MEAMAT_NFSM6CLA1A': 'Progress 8 measure for maths element - non-
disadvantaged pupils',
 'P8MEAMAT_CILOW_NFSM6CLA1A': 'Lower 95% confidence interval for
Progress 8 maths element for non-disadvantaged pupils',
 'P8MEAMAT_CIUPP_NFSM6CLA1A': 'Upper 95% confidence interval for
Progress 8 maths element for non-disadvantaged pupils',
 'ATT8SCREBAC_NFSM6CLA1A': 'Average Attainment 8 score per non-
disadvantaged pupil for EBacc element',
 'P8MEAEBAC_NFSM6CLA1A': 'Progress 8 measure for EBacc element - non-
disadvantaged pupils',
 'P8MEAEBAC_CILOW_NFSM6CLA1A': 'Lower 95% confidence interval for
Progress 8 EBacc element for non-disadvantaged pupils',
 'P8MEAEBAC_CIUPP_NFSM6CLA1A': 'Upper 95% confidence interval for
Progress 8 EBacc element for non-disadvantaged pupils',
 'ATT8SCROPEN_NFSM6CLA1A': 'Average Attainment 8 score per non-
disadvantaged pupil for open element',
 'P8MEAOPEN_NFSM6CLA1A': 'Progress 8 measure for open element - non-
disadvantaged pupils',
 'P8MEAOPEN_CILOW_NFSM6CLA1A': 'Lower 95% confidence interval for
Progress 8 open element for non-disadvantaged pupils',
 'P8MEAOPEN_CIUPP_NFSM6CLA1A': 'Upper 95% confidence interval for
Progress 8 open element for non-disadvantaged pupils',
 'ATT8SCROPENG_FSM6CLA1A': 'Average Attainment 8 score per
disadvantaged pupil for open element - GCSE only',
 'ATT8SCROPENNG_FSM6CLA1A': 'Average Attainment 8 score per
disadvantaged pupil for open element - non-GCSE only',
 'ATT8SCROPENG_NFSM6CLA1A': 'Average Attainment 8 score per non-
disadvantaged pupil for open element - GCSE only',
 'ATT8SCROPENNG_NFSM6CLA1A': 'Average Attainment 8 score per non-
disadvantaged pupil for open element - non-GCSE only',
 'DIFFN_ATT8': 'Difference between Attainment 8 for disadvantaged
pupils in school/LA and non-disadvantaged pupils nationally',
 'DIFFN_P8MEA': 'Difference between Progress 8 measure for
disadvantaged pupils in school/LA and non-disadvantaged pupils
nationally',
 'ATT8SCR_LO': 'Average Attainment 8 score per pupil with low prior
attainment',
 'P8PUP_LO': 'Number of pupils with low prior attainment included in
Progress 8 measure',
 'TP8ADJ_LO': 'Number of pupils with low prior attainments in progress
measure with adjusted scores',
 'P8MEA_LO': 'Adjusted Progress 8 measure - pupils with low prior
attainments',
 'P8CILOW_LO': 'Adjusted Progress 8 lower 95% confidence interval -
pupils with low prior attainments',
 'P8CIUPP_LO': 'Adjusted Progress 8 upper 95% confidence interval -
pupils with low prior attainments',
 'P8MEA_LO_ORIG': 'Unadjusted Progress 8 measure - pupils with low
prior attainments',
 'P8CILOW_LO_ORIG': 'Unadjusted Progress 8 lower 95% confidence
interval - pupils with low prior attainments',
 'P8CIUPP_LO_ORIG': 'Unadjusted Progress 8 upper 95% confidence
interval - pupils with low prior attainments',
 'ATT8SCR_MID': 'Average Attainment 8 score per pupil with middle
prior attainment',
 'P8PUP_MID': 'Number of pupils with middle prior attainment included
in Progress 8 measure',
 'TP8ADJ_MID': 'Number of pupils with middle prior attainments in
progress measure with adjusted scores',
 'P8MEA_MID': 'Adjusted Progress 8 measure - pupils with middle prior
```

```
attainment',
 'P8CILOW_MID': 'Progress 8 lower 95% confidence interval - pupils
with middle prior attainment',
 'P8CIUPP_MID': 'Progress 8 upper 95% confidence interval - pupils
with middle prior attainment',
 'P8MEA_MID_ORIG': 'Unadjusted Progress 8 measure - pupils with middle
prior attainments',
 'P8CILOW_MID_ORIG': 'Unadjusted Progress 8 lower 95% confidence
interval - pupils with middle prior attainments',
 'P8CIUPP_MID_ORIG': 'Unadjusted Progress 8 upper 95% confidence
interval - pupils with middle prior attainments',
 'ATT8SCR_HI': 'Average Attainment 8 score per pupil with high prior
attainment',
 'P8PUP_HI': 'Number of pupils with high prior attainment included in
Progress 8 measure',
 'TP8ADJ_HI': 'Number of pupils with high prior attainments in
progress measure with adjusted scores',
 'P8MEA_HI': 'Adjusted Progress 8 measure - pupils with high prior
attainment',
 'P8CILOW_HI': 'Progress 8 lower 95% confidence interval - pupils with
high prior attainment',
 'P8CIUPP_HI': 'Progress 8 upper 95% confidence interval - pupils with
high prior attainment',
 'P8MEA_HI_ORIG': 'Unadjusted Progress 8 measure - pupils with high
prior attainments',
 'P8CILOW_HI_ORIG': 'Unadjusted Progress 8 lower 95% confidence
interval - pupils with high prior attainments',
 'P8CIUPP_HI_ORIG': 'Unadjusted Progress 8 upper 95% confidence
interval - pupils with high prior attainments',
 'ATT8SCR_EAL': 'Average Attainment 8 score per pupil for whom English
is an additional language',
 'ATT8SCRENG_EAL': 'Average Attainment 8 score per pupil for whom
English is an additional language for English element',
 'ATT8SCRMAT_EAL': 'Average Attainment 8 score per pupil for whom
English is an additional language for mathematics element',
 'ATT8SCREBAC_EAL': 'Average Attainment 8 score per pupil for whom
English is an additional language for EBacc element',
 'ATT8SCROPEN_EAL': 'Average Attainment 8 score per pupil for whom
English is an additional language for open element',
 'ATT8SCROPENG_EAL': 'Average Attainment 8 score per pupil for whom
English is an additional language - GCSE only',
 'ATT8SCROPENNG_EAL': 'Average Attainment 8 score per pupil for whom
English is an additional language - non-GCSE only',
 'P8PUP_EAL': 'Number of pupils for whom English is an additional
language included in Progress 8 measure',
 'TP8ADJ_EAL': 'Number of pupils for whom English is an additional
language in progress measure with adjusted scores',
 'P8MEA_EAL': 'Adjusted Progress 8 measure - pupils for whom English
is an additional language',
 'P8CILOW_EAL': 'Adjusted Progress 8 lower 95% confidence interval -
```

```
pupils for whom English is an additional language',
 'P8CIUPP_EAL': 'Adjusted Progress 8 upper 95% confidence interval -
pupils for whom English is an additional language',
 'P8MEA_EAL_ORIG': 'Unadjusted Progress 8 measure - pupils for whom
English is an additional language',
 'P8CILOW_EAL_ORIG': 'Unadjusted Progress 8 lower 95% confidence
interval - pupils for whom English is an additional language',
 'P8CIUPP_EAL_ORIG': 'Unadjusted Progress 8 upper 95% confidence
interval - pupils for whom English is an additional language',
 'ATT8SCR_GIRLS': 'Average Attainment 8 score per girl',
 'ATT8SCRENG_GIRLS': 'Average Attainment 8 score per girl for English
element',
 'ATT8SCRMAT_GIRLS': 'Average Attainment 8 score per girl for
mathematics element',
 'ATT8SCREBAC_GIRLS': 'Average Attainment 8 score per girl for EBacc
element',
 'ATT8SCROPEN_GIRLS': 'Average Attainment 8 score per girl for open
element',
 'ATT8SCROPENG_GIRLS': 'Average Attainment 8 score per girl - GCSE
only',
 'ATT8SCROPENNG_GIRLS': 'Average Attainment 8 score per girl - non-
GCSE only',
 'P8PUP_GIRLS': 'Number of girls included in Progress 8 measure',
 'TP8ADJ_GIRLS': 'Number of girls in progress measure with adjusted
scores',
 'P8MEA_GIRLS': 'Adjusted Progress 8 measure - girls',
 'P8CILOW_GIRLS': 'Adjusted Progress 8 lower 95% confidence interval -
girls',
 'P8CIUPP_GIRLS': 'Adjusted Progress 8 upper 95% confidence interval -
girls',
 'P8MEA_GIRLS_ORIG': 'Unadjusted Progress 8 measure - girls',
 'P8CILOW_GIRLS_ORIG': 'Unadjusted Progress 8 lower 95% confidence
interval - girls',
 'P8CIUPP_GIRLS_ORIG': 'Unadjusted Progress 8 upper 95% confidence
interval - girls',
 'ATT8SCR_BOYS': 'Average Attainment 8 score per boy',
 'ATT8SCRENG_BOYS': 'Average Attainment 8 score per boy for English
element',
 'ATT8SCRMAT_BOYS': 'Average Attainment 8 score per boy for
mathematics element',
 'ATT8SCREBAC_BOYS': 'Average Attainment 8 score per boy for EBacc
element',
 'ATT8SCROPEN_BOYS': 'Average Attainment 8 score per boy for open
element',
 'ATT8SCROPENG_BOYS': 'Average Attainment 8 score per boy - GCSE
only',
 'ATT8SCROPENNG_BOYS': 'Average Attainment 8 score per boy - non-GCSE
only',
 'P8PUP_BOYS': 'Number of boys included in Progress 8 measure',
 'TP8ADJ_BOYS': 'Number of boys in progress measure with adjusted
```

```
scores',
 'P8MEA_BOYS': 'Adjusted Progress 8 measure - boys',
 'P8CILOW_BOYS': 'Adjusted Progress 8 lower 95% confidence interval -
boys',
 'P8CIUPP_BOYS': 'Adjusted Progress 8 upper 95% confidence interval -
boys',
 'P8MEA_BOYS_ORIG': 'Unadjusted Progress 8 measure - boys',
 'P8CILOW_BOYS_ORIG': 'Unadjusted Progress 8 lower 95% confidence
interval - boys',
 'P8CIUPP_BOYS_ORIG': 'Unadjusted Progress 8 upper 95% confidence
interval - boys',
 'ATT8SCR_NMOB': 'Average Attainment 8 score per non-mobile pupil',
 'ATT8SCRENG_NMOB': 'Average Attainment 8 score per non-mobile pupil
for English element',
 'ATT8SCRMAT_NMOB': 'Average Attainment 8 score per non-mobile pupil
for mathematics element',
 'ATT8SCREBAC_NMOB': 'Average Attainment 8 score per non-mobile pupil
for EBacc element',
 'ATT8SCROPEN_NMOB': 'Average Attainment 8 score per non-mobile pupil
for open element',
 'ATT8SCROPENG_NMOB': 'Average Attainment 8 score per non-mobile pupil
- GCSE only',
 'ATT8SCROPENNG_NMOB': 'Average Attainment 8 score per non-mobile
pupil - non-GCSE only',
 'P8PUP_NMOB': 'Number of non-mobile pupils included in Progress 8
measure',
 'TP8ADJ_NMOB': 'Number of non-mobile pupils in progress measure with
adjusted scores',
 'P8MEA_NMOB': 'Adjusted Progress 8 measure - non-mobile pupils',
 'P8CILOW_NMOB': 'Adjusted Progress 8 lower 95% confidence interval -
non-mobile pupils',
 'P8CIUPP_NMOB': 'Adjusted Progress 8 upper 95% confidence interval -
non-mobile pupils',
 'P8MEA_NMOB_ORIG': 'Unadjusted Progress 8 measure - non-mobile
pupils',
 'P8CILOW_NMOB_ORIG': 'Unadjusted Progress 8 lower 95% confidence
interval - non-mobile pupils',
 'P8CIUPP_NMOB_ORIG': 'Unadjusted Progress 8 upper 95% confidence
interval - non-mobile pupils',
 'ATT8SCR_21': 'Average Attainment 8 score per pupil  - 2021',
 'P8PUP_21': 'Number of pupils in progress measure  - 2021',
 'P8MEA_21': 'Progress 8 measure  - 2021',
 'P8CILOW_21': 'Progress 8 lower 95% confidence interval  - 2021',
 'P8CIUPP_21': 'Progress 8 upper 95% confidence interval - 2021',
 'ATT8SCR_FSM6CLA1A_21': 'Average Attainment 8 score per disadvantaged
pupil - 2021',
 'P8PUP_FSM6CLA1A_21': 'Number of disadvantaged pupils in progress
measure - 2021',
 'P8MEA_FSM6CLA1A_21': 'Progress 8 measure - disadvantaged pupils -
```

```
2021',
 'P8CILOW_FSM6CLA1A_21': 'Progress 8 lower 95% confidence interval -
disadvantaged pupils - 2021',
 'P8CIUPP_FSM6CLA1A_21': 'Progress 8 upper 95% confidence interval -
disadvantaged pupils - 2021',
 'ATT8SCR_NFSM6CLA1A_21': 'Average Attainment 8 score per non-
disadvantaged pupil - 2021',
 'P8PUP_NFSM6CLA1A_21': 'Number of non-disadvantaged pupils in
progress measure - 2021',
 'P8MEA_NFSM6CLA1A_21': 'Progress 8 measure - non-disadvantaged pupils
- 2021',
 'P8CILOW_NFSM6CLA1A_21': 'Progress 8 lower 95% confidence interval -
non-disadvantaged pupils - 2021',
 'P8CIUPP_NFSM6CLA1A_21': 'Progress 8 upper 95% confidence interval -
non-disadvantaged pupils - 2021',
 'ATT8SCR_22': 'Average Attainment 8 score per pupil - 2022',
 'P8PUP_22': 'Number of pupils in progress measure - 2022',
 'P8MEA_22': 'Progress 8 measure - 2022',
 'P8CILOW_22': 'Progress 8 lower 95% confidence interval - 2022',
 'P8CIUPP_22': 'Progress 8 upper 95% confidence interva - 2022',
 'ATT8SCR_FSM6CLA1A_22': 'Average Attainment 8 score per disadvantaged
pupil - 2022',
 'P8PUP_FSM6CLA1A_22': 'Number of disadvantaged pupils in progress
measure - 2022',
 'P8MEA_FSM6CLA1A_22': 'Progress 8 measure - disadvantaged pupils -
2022',
 'P8CILOW_FSM6CLA1A_22': 'Progress 8 lower 95% confidence interval -
disadvantaged pupils - 2022',
 'P8CIUPP_FSM6CLA1A_22': 'Progress 8 upper 95% confidence interval -
disadvantaged pupils - 2022',
 'ATT8SCR_NFSM6CLA1A_22': 'Average Attainment 8 score per non-
disadvantaged pupil  - 2022',
 'P8PUP_NFSM6CLA1A_22': 'Number of non-disadvantaged pupils in
progress measure - 2022',
 'P8MEA_NFSM6CLA1A_22': 'Progress 8 measure - non-disadvantaged pupils
- 2022',
 'P8CILOW_NFSM6CLA1A_22': 'Progress 8 lower 95% confidence interval -
non-disadvantaged pupils - 2022',
 'P8CIUPP_NFSM6CLA1A_22': 'Progress 8 upper 95% confidence interval -
non-disadvantaged pupils - 2022',
 'TEBACC_ELO_PTQ_EE': 'Number of pupils in low prior attainment band
with entries in all EBacc subject areas ',
 'PTEBACC_ELO_PTQ_EE': 'EBacc entered % by low prior attainment',
 'PTEBACCLO_94': 'EBacc achieved % by low prior attainment - with
standard 9-4 passes in English and maths ',
 'PTEBACCLO_95': 'EBacc achieved % by low prior attainment - with 9-5
passes',
 'TEBACC_EAV_PTQ_EE': 'Number of pupils in middle prior attainment
band with entries in all EBacc subject areas ',
 'PTEBACC_EAV_PTQ_EE': 'EBacc entered % by middle prior attainment',
 'PTEBACCAV_94': 'EBacc achieved % by middle prior attainment - with
9-4 passes',
 'PTEBACCAV_95': 'EBacc achieved % by middle prior attainment - with
9-5 passes',
 'TEBACC_EHI_PTQ_EE': 'Number of pupils in high prior attainment band
with entries in all EBacc subject areas ',
 'PTEBACC_EHI_PTQ_EE': 'EBacc entered % by high prior attainment',
 'PTEBACCHI_94': 'EBacc achieved % by high prior attainment - with 9-4
passes',
 'PTEBACCHI_95': 'EBacc achieved % by high prior attainment - with 9-5
passes',
 'PTEBACC_EFSM6CLA1A_PTQ_EE': '% of disadvantaged pupils entering all
English Baccalaureate subject areas',
 'PTEBACC_ENFSM6CLA1A_PTQ_EE': ' % of non-disadvantaged pupils
entering all English Baccalaureate subject areas',
 'PTEBACC_94_FSM6CLA1A': ' % of disadvantaged pupils achieving the
English Baccalaureate - with 9-4 passes',
 'PTEBACC_95_FSM6CLA1A': ' % of disadvantaged pupils achieving the
English Baccalaureate - with 9-5 passes',
 'PTEBACC_94_NFSM6CLA1A': ' % of non-disadvantaged pupils achieving
the English Baccalaureate - with 9-4 passes',
 'PTEBACC_95_NFSM6CLA1A': ' % of non-disadvantaged pupils achieving
the English Baccalaureate - with 9-5 passes',
 'SCIVAMEA_LO_PTQ_EE': 'English Baccalaureate Science Value Added
measure for pupils with low prior attainment',
 'SCIVAMEA_MID_PTQ_EE': 'English Baccalaureate Science Value Added
measure for pupils with middle prior attainment',
 'SCIVAMEA_HI_PTQ_EE': 'English Baccalaureate Science Value Added
measure for pupils with high prior attainment',
 'SCIVAMEA_FSM6CLA1A_PTQ_EE': 'English Baccalaureate Science Value
Added measure for disadvantaged pupils',
 'SCIVAMEA_NFSM6CLA1A_PTQ_EE': 'English Baccalaureate Science Value
Added measure for non-disadvantaged pupils',
 'HUMVAMEA_LO_PTQ_EE': 'English Baccalaureate Humanities Value Added
measure for pupils with low prior attainment',
 'HUMVAMEA_MID_PTQ_EE': 'English Baccalaureate Humanities Value Added
measure for pupils with middle prior attainment',
 'HUMVAMEA_HI_PTQ_EE': 'English Baccalaureate Humanities Value Added
measure for pupils with high prior attainment',
 'HUMVAMEA_FSM6CLA1A_PTQ_EE': 'English Baccalaureate Humanities Value
Added measure for disadvantaged pupils',
 'HUMVAMEA_NFSM6CLA1A_PTQ_EE': 'English Baccalaureate Humanities Value
Added measure for non-disadvantaged pupils',
 'LANVAMEA_LO_PTQ_EE': 'English Baccalaureate Languages Value Added
measure for pupils with low prior attainment',
 'LANVAMEA_MID_PTQ_EE': 'English Baccalaureate Languages Value Added
measure for pupils with middle prior attainment',
 'LANVAMEA_HI_PTQ_EE': 'English Baccalaureate Languages Value Added
```

```
measure for pupils with high prior attainment',
 'LANVAMEA_FSM6CLA1A_PTQ_EE': 'English Baccalaureate Languages Value
Added measure for disadvantaged pupils',
 'LANVAMEA_NFSM6CLA1A_PTQ_EE': 'English Baccalaureate Languages Value
Added measure for non-disadvantaged pupils',
 'SCIVAUPP_FSM6CLA1A_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Science Value Added measure for disadvantaged pupils',
 'SCIVALOW_FSM6CLA1A_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Science Value Added measure for disadvantaged pupils',
 'SCIVAUPP_NFSM6CLA1A_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Science Value Added measure for non-disadvantaged
pupils',
 'SCIVALOW_NFSM6CLA1A_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Science Value Added measure for non-disadvantaged
pupils',
 'SCIVAUPP_LO_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Science Value Added measure for pupils with low prior
attainment',
 'SCIVALOW_LO_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Science Value Added measure for pupils with low prior
attainment',
 'SCIVAUPP_MID_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Science Value Added measure for pupils with middle prior
attainment',
 'SCIVALOW_MID_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Science Value Added measure for pupils with middle prior
attainment',
 'SCIVAUPP_HI_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Science Value Added measure for pupils with high prior
attainment',
 'SCIVALOW_HI_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Science Value Added measure for pupils with high prior
attainment',
 'HUMVAUPP_FSM6CLA1A_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Humanities Value Added measure for disadvantaged
pupils',
 'HUMVALOW_FSM6CLA1A_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Humanities Value Added measure for disadvantaged
pupils',
 'HUMVAUPP_NFSM6CLA1A_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Humanities Value Added measure for non-disadvantaged
pupils',
 'HUMVALOW_NFSM6CLA1A_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Humanities Value Added measure for non-disadvantaged
pupils',
 'HUMVAUPP_LO_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Humanities Value Added measure for pupils with low prior
attainment',
 'HUMVALOW_LO_PTQ_EE': 'Lower 95% confidence limit for English
```

```
Baccalaureate Humanities Value Added measure for pupils with low prior
attainment',
 'HUMVAUPP_MID_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Humanities Value Added measure for pupils with middle
prior attainment',
 'HUMVALOW_MID_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Humanities Value Added measure for pupils with middle
prior attainment',
 'HUMVAUPP_HI_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Humanities Value Added measure for pupils with high
prior attainment',
 'HUMVALOW_HI_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Humanities Value Added measure for pupils with high
prior attainment',
 'LANVAUPP_FSM6CLA1A_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Languages Value Added measure for disadvantaged pupils',
 'LANVALOW_FSM6CLA1A_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Languages Value Added measure for disadvantaged pupils',
 'LANVAUPP_NFSM6CLA1A_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Languages Value Added measure for non-disadvantaged
pupils',
 'LANVALOW_NFSM6CLA1A_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Languages Value Added measure for non-disadvantaged
pupils',
 'LANVAUPP_LO_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Languages Value Added measure for pupils with low prior
attainment',
 'LANVALOW_LO_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Languages Value Added measure for pupils with low prior
attainment',
 'LANVAUPP_MID_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Languages Value Added measure for pupils with middle
prior attainment',
 'LANVALOW_MID_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Languages Value Added measure for pupils with middle
prior attainment',
 'LANVAUPP_HI_PTQ_EE': 'Upper 95% confidence limit for English
Baccalaureate Languages Value Added measure for pupils with high prior
attainment',
 'LANVALOW_HI_PTQ_EE': 'Lower 95% confidence limit for English
Baccalaureate Languages Value Added measure for pupils with high prior
attainment',
 'PTEBACC_E_21_PTQ_EE': '% of pupils entering all English
Baccalaureate subject areas in 2021',
 'PTEBACC_94_21': '% of KS4 pupils achieving the Ebacc - with standard
9-4 passes in English and maths in 2021',
 'PTEBACC_95_21': '% of KS4 pupils achieving the Ebacc - with strong
9-5 passes in English and maths in 2021',
 'PTEBACC_E_22_PTQ_EE': '% of  pupils entering all English
```

```
Baccalaureate subject areas in 2022',
 'PTEBACC_94_22': '% of KS4 pupils achieving the Ebacc - with standard
9-4 passes in English and maths in 2022',
 'PTEBACC_95_22': '% of KS4 pupils achieving the Ebacc - with strong
9-5 passes in English and maths in 2022',
 'PBEBACC_E_PTQ_EE': '% of boys with entries in all English
Baccalaureate subject areas',
 'PBEBACC_94': '% of KS4 boys achieving the Ebacc - with 9-4 passes',
 'PBEBACC_95': '% of KS4 boys achieving the Ebacc - with 9-5 passes',
 'PGEBACC_E_PTQ_EE': '% of girls with entries in all English
Baccalaureate subject areas',
 'PGEBACC_94': '% of KS4 girls achieving the Ebacc - with 9-4 passes',
 'PGEBACC_95': '% of KS4 girls achieving the Ebacc - with 9-5 passes',
 'PTEBACC_ENMOB_PTQ_EE': '% of non-mobile pupils with entries in all
English Baccalaureate subject areas',
 'PTEBACCNMOB_94': '% of non-mobile pupils achieving the English
Baccalaureate with 9-4 passes',
 'PTEBACCNMOB_95': '% of non-mobile pupils achieving the English
Baccalaureate with 9-5 passes',
 'PTEBACC_EEAL_PTQ_EE': '% of pupils for whom English is an additional
language with entries in all English Baccalaureate subject areas',
 'PTEBACCEAL_94': '% of pupils for whom English as an additional
language achieving the English Baccalaureate with 9-4 passes',
 'PTEBACCEAL_95': '% of pupils for whom English as an additional
language achieving the English Baccalaureate with 9-5 passes',
 'PTEBACC_EFSM6CLA1A_21': '% of disadvantaged pupils entering all
English Baccalaureate subject areas in 2021',
 'PTEBACC_94_FSM6CLA1A_21': '% of disadvantaged pupils achieving the
English Baccalaureate at grades 9-4 in 2021',
 'PTEBACC_95_FSM6CLA1A_21': '% of disadvantaged pupils achieving the
English Baccalaureate at grades 9-5 in 2021',
 'PTEBACC_ENFSM6CLA1A_21': '% of non-disadvantaged pupils entering all
English Baccalaureate subject areas in 2021',
 'PTEBACC_94_NFSM6CLA1A_21': '% of non-disadvantaged pupils achieving
the English Baccalaureate at grade 9-4 in 2021',
 'PTEBACC_95_NFSM6CLA1A_21': '% of non-disadvantaged pupils achieving
the English Baccalaureate at grade 9-5 in 2021',
 'PTEBACC_EFSM6CLA1A_22': '% of disadvantaged pupils entering all
English Baccalaureate subject areas in 2022',
 'PTEBACC_94_FSM6CLA1A_22': '% of disadvantaged pupils achieving the
English Baccalaureate including 9-4 passes in English and maths in
2022',
 'PTEBACC_95_FSM6CLA1A_22': '% of disadvantaged pupils achieving the
English Baccalaureate including 9-5 passes in English and maths in
2022',
 'PTEBACC_ENFSM6CLA1A_22': '% of non-disadvantaged pupils entering all
English Baccalaureate subject areas in 2022',
 'PTEBACC_94_NFSM6CLA1A_22': '% of non-disadvantaged pupils achieving
the English Baccalaureate including 9-4 passes in English and maths in
```

```
2022',
 'PTEBACC_95_NFSM6CLA1A_22': '% of non-disadvantaged pupils achieving
the English Baccalaureate including 9-5 passes in English and maths in
2022',
 'PT5EM_94': '% of pupils achieving Level 2 threshold including
standard passes 9-4 in both English and Maths GCSEs',
 'PT5EM_94_21': '% of pupils achieving Level 2 threshold including
standard passes 9-4 in both English and Maths GCSEs in 2021',
 'PT5EM_94_22': '% of pupils achieving Level 2 threshold including
standard passes 9-4 in both English and Maths GCSEs in 2022',
 'PTANYQ_PTQ_EE': '% of pupils achieving any qualifications',
 'PTL2BASICS_94_21': '% of pupils achieving 9-4 passes in GCSE English
and maths in 2021',
 'PTL2BASICS_95_21': '% of pupils achieving 9-5 passes in GCSE English
and maths in 2021',
 'PTL2BASICS_94_22': '% of pupils achieving 9-4 passes in GCSE English
and maths in 2022',
 'PTL2BASICS_95_22': '% of pupils achieving 9-5 passes in GCSE English
and maths in 2022',
 'PTFSM6CLA1ABASICS_94': '% of disadvantaged pupils achieving standard
9-4 passes in GCSE English and maths',
 'PTNOTFSM6CLA1ABASICS_94': '% of non-disadvantaged pupils achieving
standard 9-4 passes in GCSE English and maths',
 'TBASICSLO_94': 'Number of pupils in low prior attainment band who
achieved standard 9-4 passes in English and maths',
 'PTBASICSLO_94': '% of pupils in low prior attainment band who
achieved standard 9-4 passes in English and maths',
 'TBASICSAV_94': 'Number of pupils in middle prior attainment band who
achieved standard 9-4 passes in English and maths',
 'PTBASICSAV_94': '% pupils in middle prior attainment band who
achieved standard 9-4 passes in English and maths',
 'TBASICSHI_94': 'Number of pupils in high prior attainment band who
achieved standard 9-4 passes in English and maths',
 'PTBASICSHI_94': '% pupils in high prior attainment band who achieved
standard 9-4 passes in English and maths',
 'PBL2BASICS_94': '% of boys achieving standard 9-4 passes in both
English and mathematics GCSEs ',
 'PGL2BASICS_94': '% of girls achieving standard 9-4 passes in both
English and mathematics GCSEs ',
 'PTL2BASICSEAL_94': '% of pupils achieving standard 9-4 passes in
both English and mathematics GCSEs and for whom English is an
additional language',
 'PTL2BASICSNMOB_94': '% of non-mobile pupils achieving standard 9-4
passes in both English and mathematics GCSEs',
 'PTFSM6CLA1ABASICS_95': '% of disadvantaged pupils achieving strong
9-5 passes in GCSE English and maths',
 'PTNOTFSM6CLA1ABASICS_95': '% of non-disadvantaged pupils achieving
strong 9-5 passes in GCSE English and maths',
 'TBASICSLO_95': 'Number of pupils in low prior attainment band who
```

```
achieved strong 9-5 passes in English and maths',
 'PTBASICSLO_95': '% of pupils in low prior attainment band who
achieved strong 9-5 passes in English and maths',
 'TBASICSAV_95': 'Number of pupils in middle prior attainment band who
achieved strong 9-5 passes in English and maths',
 'PTBASICSAV_95': '% pupils in middle prior attainment band who
achieved strong 9-5 passes in English and maths',
 'TBASICSHI_95': 'Number of pupils in high prior attainment band who
achieved strong 9-5 passes in English and maths',
 'PTBASICSHI_95': '% pupils in high prior attainment band who achieved
strong 9-5 passes in English and maths',
 'PBL2BASICS_95': '% of boys achieving strong 9-5 passes in both
English and mathematics GCSEs ',
 'PGL2BASICS_95': '% of girls achieving strong 9-5 passes in both
English and mathematics GCSEs ',
 'PTL2BASICSEAL_95': '% of pupils achieving strong 9-5 passes in both
English and mathematics GCSEs and for whom English is an additional
language',
 'PTL2BASICSNMOB_95': '% of non-mobile pupils achieving strong 9-5
passes in both English and mathematics GCSEs',
 'PTFSM6CLA1ABASICS_94_21': '% of disadvantaged pupils achieving 9-4
in GCSE English and maths in 2021',
 'PTFSM6CLA1ABASICS_95_21': '% of disadvantaged pupils achieving 9-4
passes in GCSE English and maths in 2021',
 'PTNOTFSM6CLA1ABASICS_94_21': '% of non-disadvantaged pupils
achieving 9-4 passes in GCSE English and maths in 2021',
 'PTNOTFSM6CLA1ABASICS_95_21': '% of non-disadvantaged pupils
achieving 9-5 passes in GCSE English and maths in 2021',
 'PTFSM6CLA1ABASICS_94_22': '% of disadvantaged pupils achieving 9-4
passes in GCSE English and maths in 2022',
 'PTFSM6CLA1ABASICS_95_22': '% of disadvantaged pupils achieving 9-5
passes in GCSE English and maths in 2022',
 'PTNOTFSM6CLA1ABASICS_94_22': '% of non-disadvantaged pupils
achieving 9-4 passes in GCSE English and maths in 2022',
 'PTNOTFSM6CLA1ABASICS_95_22': '% of non-disadvantaged pupils
achieving 9-5 passes in GCSE English and maths in 2022',
 'PTmultiLan_E': '% of pupils entering more than one language',
 'PTtripleSci_E': '% of pupils entering biology, chemistry and
physics',
 'TFSM6CLA1A_21': 'Number of disadvantaged pupils at the end of key
stage 4 in 2021',
 'PTFSM6CLA1A_21': '% of pupils at the end of key stage 4 who were
disadvantaged in 2021',
 'TNOTFSM6CLA1A_21': 'Number of non-disadvantaged pupils at the end of
key stage 4 in 2021',
 'PTNOTFSM6CLA1A_21': '% of pupils at the end of key stage 4 who were
not disadvantaged in 2021',
 'TFSM6CLA1A_22': 'Number of disadvantaged pupils in 2022',
 'PTFSM6CLA1A_22': '% of pupils who were disadvantaged in 2022',
 'TNOTFSM6CLA1A_22': 'Number of non-disadvantaged pupils in 2022',
 'PTNOTFSM6CLA1A_22': '% of pupils who were not disadvantaged in
2022',
 'TAVENT_E_3NG_PTQ_EE': 'Average number of KS4 entries per pupil',
 'TAVENT_E_3NG_LO_PTQ_EE': 'Average number of KS4 entries per pupil
with low prior attainment',
 'TAVENT_E_3NG_MID_PTQ_EE': 'Average number of KS4 entries per pupil
with middle prior attainment',
 'TAVENT_E_3NG_HI_PTQ_EE': 'Average number of KS4 entries per pupil
with high prior attainment',
 'TAVENT_E_3NG_FSM6CLA1A_PTQ_EE': 'Average number of KS4 entries per
disadvantaged pupil',
 'TAVENT_E_3NG_NFSM6CLA1A_PTQ_EE': 'Average number of KS4 entries per
non-disadvantaged pupil',
 'TAVENT_EFSM6CLA1A_21_PTQ_EE': 'Average number of KS4 entries per
disadvantaged pupil in 2021',
 'TAVENT_ENFSM6CLA1A_21_PTQ_EE': 'Average number of KS4 entries per
non-disadvantaged pupil in 2021',
 'TAVENT_EFSM6CLA1A_22_PTQ_EE': 'Average number of KS4 entries per
disadvantaged pupil in 2022',
 'TAVENT_ENFSM6CLA1A_22_PTQ_EE': 'Average number of KS4 entries per
non-disadvantaged pupil in 2022',
 'TAVENT_G_PTQ_EE': 'Average number of GCSE entries per pupil',
 'TAVENT_GLO_PTQ_EE': 'Average number of GCSE entries per pupil with
low prior attainment',
 'TAVENT_GAV_PTQ_EE': 'Average number of GCSE entries per pupil with
middle prior attainment',
 'TAVENT_GHI_PTQ_EE': 'Average number of GCSE entries per pupil with
high prior attainment',
 'TAVENT_GFSM6CLA1A_PTQ_EE': 'Average number of GCSE entries per
disadvantaged pupil',
 'TAVENT_GNFSM6CLA1A_PTQ_EE': 'Average number of GCSE entries per non-
disadvantaged pupil',
 'TAVENT_GFSM6CLA1A_21_PTQ_EE': 'Average number of GCSE entries per
disadvantaged pupil in 2021',
 'TAVENT_GNFSM6CLA1A_21_PTQ_EE': 'Average number of GCSE entries per
non-disadvantaged pupil in 2021',
 'TAVENT_GFSM6CLA1A_22_PTQ_EE': 'Average number of GCSE entries per
disadvantaged pupil in 2022',
 'TAVENT_GNFSM6CLA1A_22_PTQ_EE': 'Average number of GCSE entries per
non-disadvantaged pupil in 2022',
 'TTOTENT_E_TOTAL_PTQ_EE': 'Total volume of entries without
discounting',
 'TTOTENT_E_COVID_IMPACTED_PTQ_EE': 'Total volume of covid-impacted
entries without discounting',
 'PTOTENT_E_COVID_IMPACTED_PTQ_EE': '% of covid-impacted entries out
of total number of entries',
 'P8_BANDING': 'Progress 8 banding shown on school performance tables
website'}
```

```
school_funding_meta = DataWrangler('data/funding_meta.csv') # this is
originally in .xlsx format
school_funding_dict =
DataWrangler.make_dictionary(school_funding_meta, 'Variable
name','Variable description')
school_funding_dict
```

```
CSV file loaded successfully from data/funding_meta.csv
```

```
{'academy': 'Academy?',
 'allocation_per_pupil': 'Allocation per Pupil',
 'basic_entitlement_ks3': 'Basic Entitlement KS3',
 'basic_entitlement_ks4': 'Basic Entitlement KS4',
 'basic_entitlement_primary': 'Basic Entitlement Primary',
 'basic_entitlement_total_funding': 'Basic Entitlement Total Funding',
 'coronavirus_recovery_premium_funding': 'Coronavirus (COVID-19)
recovery premium funding',
 'deprivation_total_funding': 'Deprivation Total Funding',
 'eal_total_funding': 'EAL Total Funding',
 'exceptional_factors_total_funding': 'Exceptional Factors Total
Funding',
 'fsm_funding': 'FSM Funding',
 'fsm6_funding': 'FSM6 Funding',
 'idaci_band_a': 'IDACI Band A',
 'idaci_band_b': 'IDACI Band B',
 'idaci_band_c': 'IDACI Band C',
 'idaci_band_d': 'IDACI Band D',
 'idaci_band_e': 'IDACI Band E',
 'idaci_band_f': 'IDACI Band F',
 'idaci_funding': 'IDACI Funding',
 'lac_total_funding': 'LAC Total Funding',
 'london_fringe': 'London Fringe',
 'lump_sum_total_funding': 'Lump Sum Total Funding',
 'mfg_protection_or_capping_scaling': 'MFG protection (+ve) or
capping/scaling (-ve)',
 'minimum_per_pupil_funding': 'Minimum per pupil funding',
 'mobility_total_funding': 'Mobility Total Funding',
 'national_non_domestic_rates_funding': 'National Non Domestic Rates
Funding',
 'notional_sen': 'Notional SEN',
 'pe_&_sport_premium': 'PE & Sport Premium funding',
 'pe_&_sport_premium_pupils': 'PE & Sport Premium pupils',
 'pfi_total_funding': 'PFI Total Funding',
 'prior_attainment_total_funding': 'Prior Attainment Total Funding',
 'pupil_premium': 'Pupil Premium funding',
 'pupil_premium_pupils': 'Pupil Premium pupils',
 'School_led_tutoring_funding': 'School-led tutoring funding',
 'school_phase': 'Phase',
 'school_type': 'School type',
 'school_ukprn': 'UKPRN',
 'schools_supplementary_grant': 'Schools Supplementary Grant funding',
 'sparsity_total_funding': 'Sparsity Total Funding',
 'split_site_total_funding': 'Split Site Total Funding',
 'total_funding': 'Total funding',
 'total_number_of_pupils': 'Total Number of Pupils (rounded)',
 'total_schools_block_allocation_(post_mfg)': 'Total Schools Block
Allocation (Post MFG)',
 'total_schools_block_allocation_(pre_mfg)': 'Total Schools Block
Allocation (Pre MFG)',
 'trust': 'Trust',
 'universal_infant_free_school_meals_grant': 'Universal Infant Free
School Meals Grant funding'}
```

Select Columns from Data

Before re-labeling the columns using the defintions in the dictionaries, it will be more efficient to select the columns needed in each data file. I shall therefor re-define each dataframe according to the selected columns needed.

MAT Performance Data:

```
#only the following columns are needed

ks4_mat_performance_df =
ks4_mat_performance.df[['TRUST_NAME','TRUST_UID', 'TRUST_ID',
'NUMINST_MATPTINC','TPUP_MATPTINC','ATT8SCR_WGTAVG', 'P8MEA_WGTAVG',
'TIME_PERIOD']]
ks4_mat_performance_df.head()
```

```
                          TRUST_NAME  TRUST_UID TRUST_ID
NUMINST_MATPTINC  \
0  ACTIVATE LEARNING EDUCATION TRUST      15710  TR02786
6
1                        ACER TRUST      15720  TR01414
3
2              RED KITE LEARNING TRUST      15727  TR00969
4
3                  CONSILIUM ACADEMIES      15728  TR00082
8
4           BATLEY MULTI ACADEMY TRUST      15729  TR00147
3

   TPUP_MATPTINC  ATT8SCR_WGTAVG  P8MEA_WGTAVG  TIME_PERIOD
0            647            40.4         -0.68       202223
1            548            49.4          0.08       202223
2            810            46.9          0.00       202223
3           1150            37.5         -0.86       202223
4            520            45.7          0.16       202223
```

Keystage 4 School Performance Data:

```
#following columns are key measures for school performance
ks4_school_performance_df = ks4_school_performance.df[['URN',
    'ATT8SCR',
    'P8MEA',
    'PTFSM6CLA1A_22',
    'PTNOTFSM6CLA1A_22',
    'PTFSM6CLA1ABASICS_95', #462
    'PTNOTFSM6CLA1ABASICS_95',
    'ATT8SCR_NFSM6CLA1A_22',
    'P8MEA_NFSM6CLA1A_22',
'ATT8SCR_FSM6CLA1A_22',
    'P8MEA_FSM6CLA1A_22',
    'P8MEAMAT_FSM6CLA1A',
    'P8MEAENG_FSM6CLA1A',
    'P8MEAMAT_NFSM6CLA1A',
    'P8MEAENG_NFSM6CLA1A'
    ]]


ks4_school_performance_df.head()
```

```
        URN ATT8SCR  P8MEA PTFSM6CLA1A_22 PTNOTFSM6CLA1A_22  \
0  100003.0    36.8     NP             NP                NP
1  100001.0    29.4     NP             NP                NP
2  100544.0     6.8     NP             NP                NP
3       NaN     NaN    NaN            NaN               NaN
4  100053.0    50.3  -0.16            42%               58%

  PTFSM6CLA1ABASICS_95 PTNOTFSM6CLA1ABASICS_95
ATT8SCR_NFSM6CLA1A_22  \
0                   NP                      NP                     NP

1                   NP                      NP                     NP

2                   NP                      NP                     NP

3                  NaN                     NaN                    NaN

4                  28%                     74%                   59.7


  P8MEA_NFSM6CLA1A_22 ATT8SCR_FSM6CLA1A_22 P8MEA_FSM6CLA1A_22  \
0                  NP                   NP                 NP
1                  NP                   NP                 NP
2                  NP                   NP                 NP
3                 NaN                  NaN                NaN
4                0.26                   38              -0.99

  P8MEAMAT_FSM6CLA1A P8MEAENG_FSM6CLA1A P8MEAMAT_NFSM6CLA1A  \
0                 NP                 NP                  NP
1                 NP                 NP                  NP
2                 NP                 NP                  NP
3                NaN                NaN                 NaN
4              -0.82              -0.79                0.39

  P8MEAENG_NFSM6CLA1A
0                  NP
1                  NP
2                  NP
3                 NaN
4                0.35
```

School Demographics:

```
school_demographics_df =
school_demographics.df[['URN','LANAME','LA','SCHOOLTYPE','MINORGROUP',
'RELCHAR','ADMPOL','GENDER','OFSTEDRATING', 'POSTCODE']]
school_demographics_df.head()

      URN           LANAME    LA                    SCHOOLTYPE
MINORGROUP  \
0  100000  City of London  201    Voluntary aided school   Maintained
school
1  100001  City of London  201  Other independent school  Independent
school
2  100002  City of London  201  Other independent school  Independent
school
3  100003  City of London  201  Other independent school  Independent
school
4  100008          Camden  202          Community school   Maintained
school

             RELCHAR          ADMPOL GENDER OFSTEDRATING  POSTCODE
0  Church of England  Not applicable  Mixed  Outstanding  EC3A 5DE
1                NaN       Selective  Girls          NaN  EC2Y 8BB
2  Church of England  Not applicable  Mixed          NaN  EC4M 9AD
3                NaN  Not applicable   Boys          NaN  EC4V 3AL
4     Does not apply  Not applicable  Mixed         Good  WC1H 9EG
```

School Funding:

```
school_funding_df =
school_funding.df[['school_urn','fsm_funding','pupil_premium','pupil_p
remium_pupils','School_led_tutoring_funding','total_funding']]

school_funding_df.head()
```

```
   school_urn  fsm_funding pupil_premium pupil_premium_pupils  \
0      101247       118662        291560                  296
1      101241       198479        492993                  501
2      101202        75028        209135                  151
3      101231        55872        153735                  111
4      136028       211782        511215                  519

  School_led_tutoring_funding  total_funding
0                       49248      8542828.0
1                       84024     13420859.0
2                       24138      3439599.0
3                       18117      2633909.0
4                       91017      9836214.0
```

```
#change column name
school_funding_df.rename(columns={'school_urn':'URN'}, inplace=True) # change column name to match the school demographics dataframe  to allow ease of merging
school_funding_df.head()
```

```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\65584972.py:2: 
SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: 
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  school_funding_df.rename(columns={'school_urn':'URN'}, inplace=True) 
# change column name to match the school demographics dataframe  to allow ease of merging

      URN  fsm_funding pupil_premium pupil_premium_pupils  \
0  101247       118662        291560                  296
1  101241       198479        492993                  501
2  101202        75028        209135                  151
3  101231        55872        153735                  111
4  136028       211782        511215                  519

  School_led_tutoring_funding  total_funding
0                       49248      8542828.0
1                       84024     13420859.0
2                       24138      3439599.0
3                       18117      2633909.0
4                       91017      9836214.0
```

Academies Membership

- Only URN, Trust ID, School Name and Trust Name are needed

```
academies_membership_df = academies_membership.df[['URN','Group
UID','Group ID','EstablishmentName','Group Name']]
academies_membership_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12637 entries, 0 to 12636
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   URN                12618 non-null  float64
 1   Group UID          12463 non-null  float64
 2   Group ID           12463 non-null  object
 3   EstablishmentName  12618 non-null  object
 4   Group Name         12463 non-null  object
dtypes: float64(2), object(3)
memory usage: 493.8+ KB
```

Merging DfE Data

I can now begin merging the various DfE data based on school URN

Merge: school demographics and school funding

```
school_funding_df.columns
school_funding_df.rename(columns={'school_urn': 'URN'}, inplace=True)
# change name of URN column
school_funding_df.columns
```

```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\2541994176.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  school_funding_df.rename(columns={'school_urn': 'URN'},
inplace=True) # change name of URN column
```

```
Index(['URN', 'fsm_funding', 'pupil_premium', 'pupil_premium_pupils',
       'School_led_tutoring_funding', 'total_funding'],
      dtype='object')
```

```
# Merge School Information with Funding Data
merged_df = pd.merge(school_demographics_df, school_funding_df,
on='URN', how='inner')
#chosen an inner join, as having incomplete left or right fields will
not be of use
print("School Demographics and Funding data merged.")
```

```
merged_df.shape

School Demographics and Funding data merged.

(19973, 15)

merged_df.columns

Index(['URN', 'LANAME', 'LA', 'SCHOOLTYPE', 'MINORGROUP', 'RELCHAR',
'ADMPOL',
       'GENDER', 'OFSTEDRATING', 'POSTCODE', 'fsm_funding',
'pupil_premium',
       'pupil_premium_pupils', 'School_led_tutoring_funding',
'total_funding'],
      dtype='object')
```

Merge MAT info

```
 # Merge with MAT Performance
merged_df = pd.merge(merged_df, academies_membership_df, on= ['URN'],
how='left')
# some schools may not have an academy therefore a left join
print("Merged with MAT Performance data.")

Merged with MAT Performance data.

merged_df['URN'].nunique() # count how many unique schools exist and
therefore if some are duplicated

19973
```

Some of the URNs may be duplicates and will need to be dropped later on when I conduct data cleaning.

merge school performance

```
merged_df = pd.merge(merged_df, ks4_school_performance_df, on='URN',
how='inner')
# inner join essential as having only 'right' or 'left' data wouldnt
be of much use
print("School KS4 performance data merged.")
merged_df.head()

School KS4 performance data merged.

      URN  LANAME   LA         SCHOOLTYPE         MINORGROUP
RELCHAR  \
0  100049  Camden  202  Community school  Maintained school  Does not
apply
1  100050  Camden  202  Community school  Maintained school  Does not
apply
```

```
2  100051  Camden  202  Community school  Maintained school  Does not
apply
3  100052  Camden  202  Community school  Maintained school  Does not
apply
4  100053  Camden  202  Community school  Maintained school  Does not
apply

          ADMPOL GENDER OFSTEDRATING POSTCODE  ...
PTFSM6CLA1ABASICS_95  \
0  Non-selective  Mixed         Good  NW3 2BQ  ...
40%
1  Non-selective  Girls  Outstanding  NW5 1RL  ...
53%
2  Non-selective  Mixed         Good  NW1 1RX  ...
35%
3  Non-selective  Mixed         Good  NW2 3RT  ...
31%
4  Non-selective  Mixed         Good  NW5 1UJ  ...
28%

  PTNOTFSM6CLA1ABASICS_95 ATT8SCR_NFSM6CLA1A_22 P8MEA_NFSM6CLA1A_22  \
0                     48%                  54.9                0.15
1                     76%                  71.9                1.28
2                     48%                  54.5                0.66
3                     53%                  50.2                0.22
4                     74%                  59.7                0.26

   ATT8SCR_FSM6CLA1A_22  P8MEA_FSM6CLA1A_22 P8MEAMAT_FSM6CLA1A  \
0                    47               -0.12              -0.16
1                  52.3                0.36              -0.09
2                  49.4                0.29              -0.01
3                  43.2                0.05              -0.05
4                    38               -0.99              -0.82

  P8MEAENG_FSM6CLA1A P8MEAMAT_NFSM6CLA1A P8MEAENG_NFSM6CLA1A
0              -0.57                0.30                0.01
1               0.12                0.74                0.93
2              -0.20                0.77                0.10
3              -0.01                0.38                0.19
4              -0.79                0.39                0.35

[5 rows x 33 columns]
```

```
merged_df['URN'].nunique()
```

```
3281
```

Observation: The number of unique schools has dropped from 19k to 3k, when the keystage 4 data was merged on an inner join. This is expected as reportedly 3444 state-funded secondary schools in England, with private schools included it is approximately 4175 [7]. Of these a number by newly opened and not have delivered GCSE in 2022/23

[7] Tes. (2024, January 17). How many schools are there in the UK? Retrieved from https://www.tes.com/magazine/analysis/general/how-many-schools-in-the-uk

merged MAT performance data

```
merged_df = pd.merge(merged_df, ks4_mat_performance_df,
left_on=['Group ID'], right_on=['TRUST_ID'], how='left')
print("Merged with MAT Performance data.")
```

```
Merged with MAT Performance data.
```

```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3866 entries, 0 to 3865
Data columns (total 41 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   URN                           3866 non-null   int64  
 1   LANAME                        3866 non-null   object 
 2   LA                            3866 non-null   int64  
 3   SCHOOLTYPE                    3866 non-null   object 
 4   MINORGROUP                    3866 non-null   object 
 5   RELCHAR                       2399 non-null   object 
 6   ADMPOL                        3601 non-null   object 
 7   GENDER                        3866 non-null   object 
 8   OFSTEDRATING                  3822 non-null   object 
 9   POSTCODE                      3866 non-null   object 
 10  fsm_funding                   3866 non-null   int64  
 11  pupil_premium                 3866 non-null   object 
 12  pupil_premium_pupils          3866 non-null   object 
 13  School_led_tutoring_funding   3866 non-null   object 
 14  total_funding                 3866 non-null   float64
 15  Group UID                     3214 non-null   float64
 16  Group ID                      3214 non-null   object 
 17  EstablishmentName             3214 non-null   object 
 18  Group Name                    3214 non-null   object 
 19  ATT8SCR                       3807 non-null   object 
 20  P8MEA                         3807 non-null   object 
 21  PTFSM6CLA1A_22                3770 non-null   object 
 22  PTNOTFSM6CLA1A_22             3770 non-null   object 
 23  PTFSM6CLA1ABASICS_95          3807 non-null   object 
 24  PTNOTFSM6CLA1ABASICS_95       3807 non-null   object 
 25  ATT8SCR_NFSM6CLA1A_22         3770 non-null   object 
 26  P8MEA_NFSM6CLA1A_22           3770 non-null   object 
 27  ATT8SCR_FSM6CLA1A_22          3770 non-null   object 
 28  P8MEA_FSM6CLA1A_22            3770 non-null   object 
 29  P8MEAMAT_FSM6CLA1A            3807 non-null   object 
 30  P8MEAENG_FSM6CLA1A           3807 non-null   object
 31  P8MEAMAT_NFSM6CLA1A          3807 non-null   object
 32  P8MEAENG_NFSM6CLA1A          3807 non-null   object
 33  TRUST_NAME                   1346 non-null   object
 34  TRUST_UID                    1346 non-null   float64
 35  TRUST_ID                     1346 non-null   object
 36  NUMINST_MATPTINC             1346 non-null   float64
 37  TPUP_MATPTINC                1346 non-null   float64
 38  ATT8SCR_WGTAVG               1346 non-null   float64
 39  P8MEA_WGTAVG                 1346 non-null   float64
 40  TIME_PERIOD                  1346 non-null   float64
dtypes: float64(8), int64(3), object(30)
memory usage: 1.2+ MB
```

Data Cleaning

Now that the merging is complete, I can now remove rows which are note needed

Remove NaN values

```
merged_df.isna().sum()
```

```
URN                              0
LANAME                           0
LA                               0
SCHOOLTYPE                       0
MINORGROUP                       0
RELCHAR                       1467
ADMPOL                         265
GENDER                           0
OFSTEDRATING                    44
POSTCODE                         0
fsm_funding                      0
pupil_premium                    0
pupil_premium_pupils             0
School_led_tutoring_funding      0
total_funding                    0
Group UID                      652
Group ID                       652
EstablishmentName              652
Group Name                     652
ATT8SCR                         59
P8MEA                           59
PTFSM6CLA1A_22                  96
PTNOTFSM6CLA1A_22               96
PTFSM6CLA1ABASICS_95            59
PTNOTFSM6CLA1ABASICS_95         59
ATT8SCR_NFSM6CLA1A_22           96
P8MEA_NFSM6CLA1A_22             96
```

```
ATT8SCR_FSM6CLA1A_22             96
P8MEA_FSM6CLA1A_22               96
P8MEAMAT_FSM6CLA1A               59
P8MEAENG_FSM6CLA1A               59
P8MEAMAT_NFSM6CLA1A              59
P8MEAENG_NFSM6CLA1A              59
TRUST_NAME                     2520
TRUST_UID                      2520
TRUST_ID                       2520
NUMINST_MATPTINC               2520
TPUP_MATPTINC                  2520
ATT8SCR_WGTAVG                 2520
P8MEA_WGTAVG                   2520
TIME_PERIOD                    2520
dtype: int64
```

There may not be a need to drop all NaN values in every variable, as some schools may not have an OFSTED rating nor be part of a Trust in 2022/23, and I wouldnt want to discard the rest of their data from analysis

```
merged_df = merged_df.dropna(subset=['P8MEA_FSM6CLA1A_22'])
# Im not dropping all NaN values, as some schools may not have an OFSTED rating nor be part of a Trust in 2022/23
merged_df.isna().sum()
```

```
URN                                0
LANAME                             0
LA                                 0
SCHOOLTYPE                         0
MINORGROUP                         0
RELCHAR                         1403
ADMPOL                           248
GENDER                             0
OFSTEDRATING                      33
POSTCODE                           0
fsm_funding                        0
pupil_premium                      0
pupil_premium_pupils               0
School_led_tutoring_funding        0
total_funding                      0
Group UID                        650
Group ID                         650
EstablishmentName                650
Group Name                       650
ATT8SCR                            0
P8MEA                              0
PTFSM6CLA1A_22                     0
PTNOTFSM6CLA1A_22                  0
PTFSM6CLA1ABASICS_95               0
```

```
PTNOTFSM6CLA1ABASICS_95           0
ATT8SCR_NFSM6CLA1A_22             0
P8MEA_NFSM6CLA1A_22               0
ATT8SCR_FSM6CLA1A_22              0
P8MEA_FSM6CLA1A_22                0
P8MEAMAT_FSM6CLA1A                0
P8MEAENG_FSM6CLA1A                0
P8MEAMAT_NFSM6CLA1A               0
P8MEAENG_NFSM6CLA1A               0
TRUST_NAME                     2474
TRUST_UID                      2474
TRUST_ID                       2474
NUMINST_MATPTINC               2474
TPUP_MATPTINC                  2474
ATT8SCR_WGTAVG                 2474
P8MEA_WGTAVG                   2474
TIME_PERIOD                    2474
dtype: int64
```

Remove Duplicates

We can now check for duplicates and remove them

```
duplicate_urns = merged_df[merged_df.duplicated('URN', keep=False)]
#keep= False will mark all duplicates as True regardless of position
duplicate_urns.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1114 entries, 600 to 3819
Data columns (total 41 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   URN                           1114 non-null   int64
 1   LANAME                        1114 non-null   object
 2   LA                            1114 non-null   int64
 3   SCHOOLTYPE                    1114 non-null   object
 4   MINORGROUP                    1114 non-null   object
 5   RELCHAR                       654 non-null    object
 6   ADMPOL                        1031 non-null   object
 7   GENDER                        1114 non-null   object
 8   OFSTEDRATING                  1106 non-null   object
 9   POSTCODE                      1114 non-null   object
 10  fsm_funding                   1114 non-null   int64
 11  pupil_premium                 1114 non-null   object
 12  pupil_premium_pupils          1114 non-null   object
 13  School_led_tutoring_funding   1114 non-null   object
 14  total_funding                 1114 non-null   float64
 15  Group UID                     1114 non-null   float64
 16  Group ID                      1114 non-null   object
 17  EstablishmentName              1114 non-null   object
 18  Group Name                     1114 non-null   object
 19  ATT8SCR                        1114 non-null   object
 20  P8MEA                          1114 non-null   object
 21  PTFSM6CLA1A_22                 1114 non-null   object
 22  PTNOTFSM6CLA1A_22              1114 non-null   object
 23  PTFSM6CLA1ABASICS_95           1114 non-null   object
 24  PTNOTFSM6CLA1ABASICS_95        1114 non-null   object
 25  ATT8SCR_NFSM6CLA1A_22          1114 non-null   object
 26  P8MEA_NFSM6CLA1A_22            1114 non-null   object
 27  ATT8SCR_FSM6CLA1A_22           1114 non-null   object
 28  P8MEA_FSM6CLA1A_22             1114 non-null   object
 29  P8MEAMAT_FSM6CLA1A             1114 non-null   object
 30  P8MEAENG_FSM6CLA1A             1114 non-null   object
 31  P8MEAMAT_NFSM6CLA1A            1114 non-null   object
 32  P8MEAENG_NFSM6CLA1A            1114 non-null   object
 33  TRUST_NAME                     386 non-null    object
 34  TRUST_UID                      386 non-null    float64
 35  TRUST_ID                       386 non-null    object
 36  NUMINST_MATPTINC               386 non-null    float64
 37  TPUP_MATPTINC                  386 non-null    float64
 38  ATT8SCR_WGTAVG                 386 non-null    float64
 39  P8MEA_WGTAVG                   386 non-null    float64
 40  TIME_PERIOD                    386 non-null    float64
dtypes: float64(8), int64(3), object(30)
memory usage: 365.5+ KB
```

```
#drop duplicates
merged_df = merged_df.drop_duplicates(subset='URN')
#confirm duplicates are removed
duplicate_urns = merged_df[merged_df.duplicated('URN', keep=False)]
duplicate_urns.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 0 entries
Data columns (total 41 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   URN                            0 non-null      int64
 1   LANAME                         0 non-null      object
 2   LA                             0 non-null      int64
 3   SCHOOLTYPE                     0 non-null      object
 4   MINORGROUP                     0 non-null      object
 5   RELCHAR                        0 non-null      object
 6   ADMPOL                         0 non-null      object
 7   GENDER                         0 non-null      object
 8   OFSTEDRATING                   0 non-null      object
 9   POSTCODE                       0 non-null      object
 10  fsm_funding                    0 non-null      int64
 11  pupil_premium                  0 non-null      object
 12  pupil_premium_pupils         0 non-null      object
 13  School_led_tutoring_funding  0 non-null      object
 14  total_funding                0 non-null      float64
 15  Group UID                    0 non-null      float64
 16  Group ID                     0 non-null      object
 17  EstablishmentName            0 non-null      object
 18  Group Name                   0 non-null      object
 19  ATT8SCR                      0 non-null      object
 20  P8MEA                        0 non-null      object
 21  PTFSM6CLA1A_22               0 non-null      object
 22  PTNOTFSM6CLA1A_22            0 non-null      object
 23  PTFSM6CLA1ABASICS_95         0 non-null      object
 24  PTNOTFSM6CLA1ABASICS_95      0 non-null      object
 25  ATT8SCR_NFSM6CLA1A_22        0 non-null      object
 26  P8MEA_NFSM6CLA1A_22          0 non-null      object
 27  ATT8SCR_FSM6CLA1A_22         0 non-null      object
 28  P8MEA_FSM6CLA1A_22           0 non-null      object
 29  P8MEAMAT_FSM6CLA1A           0 non-null      object
 30  P8MEAENG_FSM6CLA1A           0 non-null      object
 31  P8MEAMAT_NFSM6CLA1A          0 non-null      object
 32  P8MEAENG_NFSM6CLA1A          0 non-null      object
 33  TRUST_NAME                   0 non-null      object
 34  TRUST_UID                    0 non-null      float64
 35  TRUST_ID                     0 non-null      object
 36  NUMINST_MATPTINC             0 non-null      float64
 37  TPUP_MATPTINC                0 non-null      float64
 38  ATT8SCR_WGTAVG               0 non-null      float64
 39  P8MEA_WGTAVG                 0 non-null      float64
 40  TIME_PERIOD                  0 non-null      float64
dtypes: float64(8), int64(3), object(30)
memory usage: 0.0+ bytes
```

Correct Data Types

I will not proceed to check the data is in the format needed, particularly for numerical analysis.

```
merged_df.dtypes
```

```
URN                              int64
LANAME                          object
LA                               int64
SCHOOLTYPE                      object
MINORGROUP                      object
RELCHAR                         object
ADMPOL                          object
GENDER                          object
OFSTEDRATING                    object
POSTCODE                        object
fsm_funding                      int64
```

```
pupil_premium                   object
pupil_premium_pupils            object
School_led_tutoring_funding     object
total_funding                  float64
Group UID                      float64
Group ID                        object
EstablishmentName               object
Group Name                      object
ATT8SCR                         object
P8MEA                           object
PTFSM6CLA1A_22                  object
PTNOTFSM6CLA1A_22               object
PTFSM6CLA1ABASICS_95            object
PTNOTFSM6CLA1ABASICS_95         object
ATT8SCR_NFSM6CLA1A_22           object
P8MEA_NFSM6CLA1A_22             object
ATT8SCR_FSM6CLA1A_22            object
P8MEA_FSM6CLA1A_22              object
P8MEAMAT_FSM6CLA1A              object
P8MEAENG_FSM6CLA1A              object
P8MEAMAT_NFSM6CLA1A             object
P8MEAENG_NFSM6CLA1A             object
TRUST_NAME                      object
TRUST_UID                      float64
TRUST_ID                        object
NUMINST_MATPTINC               float64
TPUP_MATPTINC                  float64
ATT8SCR_WGTAVG                 float64
P8MEA_WGTAVG                   float64
TIME_PERIOD                    float64
dtype: object
```

A number of the numerical columns are listed as objects and will need to be changed to a numerical type (integer or float). However, before that, we would need to identify and remove any signs in the data e.g. £ or %

Identified columns which have a % in their data and would need removing

```
percentage_columns = [
    'PTFSM6CLA1A_22',
    'PTNOTFSM6CLA1A_22',
    'PTFSM6CLA1ABASICS_95',
    'PTNOTFSM6CLA1ABASICS_95'
]

merged_df[percentage_columns].head()

  PTFSM6CLA1A_22 PTNOTFSM6CLA1A_22 PTFSM6CLA1ABASICS_95  \
0            63%               38%                  40%
```

```
1            39%               61%                  53%
2            72%               28%                  35%
3            45%               55%                  31%
4            42%               58%                  28%

  PTNOTFSM6CLA1ABASICS_95
0                     48%
1                     76%
2                     48%
3                     53%
4                     74%
```

```
data_loader = DataWrangler(dataframe=merged_df)

# Convert percentage columns
merged_df = data_loader.convert_percentage_columns(percentage_columns)


print("\nAfter removing '%' signs and converting to float:")
merged_df[percentage_columns].head()
```

```
DataWrangler initialised with the provided DataFrame.
Column 'PTFSM6CLA1A_22' converted
Column 'PTNOTFSM6CLA1A_22' converted
Column 'PTFSM6CLA1ABASICS_95' converted
Column 'PTNOTFSM6CLA1ABASICS_95' converted

After removing '%' signs and converting to float:

  PTFSM6CLA1A_22 PTNOTFSM6CLA1A_22 PTFSM6CLA1ABASICS_95  \
0             63                38                   40
1             39                61                   53
2             72                28                   35
3             45                55                   31
4             42                58                   28

  PTNOTFSM6CLA1ABASICS_95
0                      48
1                      76
2                      48
3                      53
4                      74
```

I can now convert all 'numerical columns' to their correct data type

```
columns_to_convert_numeric = [
    'fsm_funding',
    'pupil_premium',
    'pupil_premium_pupils',
    'School_led_tutoring_funding',
    'ATT8SCR',
    'P8MEA',
    'PTFSM6CLA1A_22',
    'PTNOTFSM6CLA1A_22',
    'PTFSM6CLA1ABASICS_95',
    'PTNOTFSM6CLA1ABASICS_95',
    'ATT8SCR_NFSM6CLA1A_22',
    'P8MEA_NFSM6CLA1A_22',
    'ATT8SCR_FSM6CLA1A_22',
    'P8MEA_FSM6CLA1A_22',
    'P8MEAMAT_FSM6CLA1A',
    'P8MEAENG_FSM6CLA1A',
    'P8MEAMAT_NFSM6CLA1A',
    'P8MEAENG_NFSM6CLA1A',
]

# Convert specified columns to numeric, coercing errors to NaN
merged_df[columns_to_convert_numeric] =
merged_df[columns_to_convert_numeric].apply(pd.to_numeric,
errors='coerce')

print("Data types after conversion:")
merged_df.dtypes
```

```
Data types after conversion:
```

```
URN                                  int64
LANAME                              object
LA                                   int64
SCHOOLTYPE                          object
MINORGROUP                          object
RELCHAR                             object
ADMPOL                              object
GENDER                              object
OFSTEDRATING                        object
POSTCODE                            object
fsm_funding                          int64
pupil_premium                        int64
pupil_premium_pupils                 int64
School_led_tutoring_funding        float64
total_funding                      float64
Group UID                          float64
Group ID                            object
EstablishmentName                   object
Group Name                          object
ATT8SCR                            float64
P8MEA                              float64
PTFSM6CLA1A_22                       int64
PTNOTFSM6CLA1A_22                    int64
```

```
PTFSM6CLA1ABASICS_95           float64
PTNOTFSM6CLA1ABASICS_95        float64
ATT8SCR_NFSM6CLA1A_22          float64
P8MEA_NFSM6CLA1A_22            float64
ATT8SCR_FSM6CLA1A_22           float64
P8MEA_FSM6CLA1A_22             float64
P8MEAMAT_FSM6CLA1A             float64
P8MEAENG_FSM6CLA1A             float64
P8MEAMAT_NFSM6CLA1A            float64
P8MEAENG_NFSM6CLA1A            float64
TRUST_NAME                      object
TRUST_UID                      float64
TRUST_ID                        object
NUMINST_MATPTINC               float64
TPUP_MATPTINC                  float64
ATT8SCR_WGTAVG                 float64
P8MEA_WGTAVG                   float64
TIME_PERIOD                    float64
dtype: object
```

Nomenclature

Using the dictionaries created earlier from the meta data, I can run the column rename function to only rename the columns available in merged_df

```
data_loader = DataWrangler(dataframe=merged_df)

#rename columns based on dictionary
merged_df = data_loader.column_rename(school_performance_dict)
merged_df = data_loader.column_rename(ks4_mat_performance_dict)
merged_df = data_loader.column_rename(school_demographics_dict)
merged_df = data_loader.column_rename(school_funding_dict)
merged_df.info()
```

```
DataWrangler initialised with the provided DataFrame.
Columns renamed successfully.
Columns renamed successfully.
Columns renamed successfully.
Columns renamed successfully.
<class 'pandas.core.frame.DataFrame'>
Index: 3190 entries, 0 to 3862
Data columns (total 41 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   School Unique Reference Number
3190 non-null   int64
```

```
 1   Local authority name
3190 non-null   object
 2   Local authority number
3190 non-null   int64
 3   School Type eg Voluntary Aided school
3190 non-null   object
 4   Type of school / college eg maintained school
3190 non-null   object
 5   Religious character
2028 non-null   object
 6   School admissions policy (self-declared by schools on Edubase)
2985 non-null   object
 7   Indicates whether it's a mixed or single sex school
3190 non-null   object
 8   Ofsted rating
3161 non-null   object
 9   School postcode
3190 non-null   object
 10  FSM Funding
3190 non-null   int64
 11  Pupil Premium funding
3190 non-null   int64
 12  Pupil Premium pupils
3190 non-null   int64
 13  School-led tutoring funding
3188 non-null   float64
 14  Total funding
3190 non-null   float64
 15  Group UID
2540 non-null   float64
 16  Group ID
2540 non-null   object
 17  EstablishmentName
2540 non-null   object
 18  Group Name
2540 non-null   object
 19  Average Attainment 8 score per pupil
3190 non-null   float64
 20  Progress 8 measure after adjustment for extreme scores
3187 non-null   float64
 21  % of pupils who were disadvantaged in 2022
3190 non-null   int64
 22  % of pupils who were not disadvantaged in 2022
3190 non-null   int64
 23  % of disadvantaged pupils achieving strong 9-5 passes in GCSE
English and maths      3145 non-null   float64
 24  % of non-disadvantaged pupils achieving strong 9-5 passes in GCSE
English and maths  3147 non-null   float64
 25  Average Attainment 8 score per non-disadvantaged pupil  - 2022
```

```
3149 non-null   float64
 26  Progress 8 measure - non-disadvantaged pupils - 2022
3147 non-null   float64
 27  Average Attainment 8 score per disadvantaged pupil - 2022
3147 non-null   float64
 28  Progress 8 measure - disadvantaged pupils - 2022
3139 non-null   float64
 29  Progress 8 measure for maths element - disadvantaged pupils
3134 non-null   float64
 30  Progress 8 measure for English element - disadvantaged pupils
3134 non-null   float64
 31  Progress 8 measure for maths element - non-disadvantaged pupils
3142 non-null   float64
 32  Progress 8 measure for English element - non-disadvantaged pupils
3142 non-null   float64
 33  Trust name
1077 non-null   object
 34  Trust Unique identifier
1077 non-null   float64
 35  Trust Identifier
1077 non-null   object
 36  Number of academies in the trust, included in performance
measures                   1077 non-null   float64
 37  Number of pupils at the end of ks4, included in performance
measures                 1077 non-null   float64
 38  Average Attainment 8 score per pupil at the end of KS4, weighted
average             1077 non-null   float64
 39  Progress 8 measure after adjustment for extreme scores, weighted
average             1077 non-null   float64
 40  nan
1077 non-null   float64
dtypes: float64(21), int64(7), object(13)
memory usage: 1.0+ MB
```

The columns have a new name based on a description. This can now be changed to a more column friendly format using a new dectionary:

```
# A dictionary mapping old column names to new column names
column_rename_dict = {
    'School Unique Reference Number': 'URN',
    'Local authority name': 'Local_Authority_Name',
    'Local authority number': 'Local_Authority_Number',
    'School Type eg Voluntary Aided school': 'School_Type',
    'Type of school / college eg maintained school':
'School_College_Type',
    'Religious character': 'Religious_Character',
    'School admissions policy (self-declared by schools on Edubase)':
'Admissions_Policy',
    'Indicates whether it\'s a mixed or single sex school':
```

```
'School_Gender',
    'Ofsted rating': 'Ofsted_Rating',
    'FSM Funding': 'FSM_Funding',
    'Pupil Premium funding': 'Pupil_Premium_Funding',
    'Pupil Premium pupils': 'Pupil_Premium_Pupils',
    'School-led tutoring funding': 'School_Led_Tutoring_Funding',
    'Total funding': 'Total_Funding',
    'Group UID': 'Group_UID',
    'Group ID': 'Group_ID',
    'EstablishmentName': 'School_Name',
    'Group Name': 'Trust_Name', #first option for Trust Name
    'Average Attainment 8 score per pupil': 'Attainment8',
    'Progress 8 measure after adjustment for extreme scores':
'Progress8',
    '% of pupils who were disadvantaged in 2022':
'Percent_Disadvantaged_2022',
    '% of pupils who were not disadvantaged in 2022':
'Percent_Not_Disadvantaged_2022',
    '% of disadvantaged pupils achieving strong 9-5 passes in GCSE
English and maths': 'Percent_Disadvantaged_Strong_Passes',
    '% of non-disadvantaged pupils achieving strong 9-5 passes in GCSE
English and maths': 'Percent_Not_Disadvantaged_Strong_Passes',
    'Average Attainment 8 score per non-disadvantaged pupil - 2022':
'Attainment8_NonDisadvantaged_2022',
    'Progress 8 measure - non-disadvantaged pupils - 2022':
'Progress8_NonDisadvantaged_2022',
    'Average Attainment 8 score per disadvantaged pupil - 2022':
'Attainment8_Disadvantaged_2022',
    'Progress 8 measure - disadvantaged pupils - 2022':
'Progress8_Disadvantaged_2022',
    'Progress 8 measure for maths element - disadvantaged pupils':
'Progress8_Maths_Disadvantaged',
    'Progress 8 measure for English element - disadvantaged pupils':
'Progress8_English_Disadvantaged',
    'Progress 8 measure for maths element - non-disadvantaged pupils':
'Progress8_Maths_NonDisadvantaged',
    'Progress 8 measure for English element - non-disadvantaged
pupils': 'Progress8_English_NonDisadvantaged',
    'Trust name': 'trust_name', # second option to match Trust and
quality assure data
    'Trust Unique identifier': 'Trust_UID',
    'Trust Identifier': 'Trust_ID',
    'Number of academies in the trust, included in performance
measures': 'Num_Academies_Performance',
    'Number of pupils at the end of ks4, included in performance
measures': 'Num_Pupils_KS4_Performance',
    'Average Attainment 8 score per pupil at the end of KS4, weighted
average': 'Avg_Attainment8_KS4_Weighted',
    'Progress 8 measure after adjustment for extreme scores, weighted
```

```
average': 'Progress8_Adjusted_Weighted',
    'nan': 'Time_Period'  # I can remove if not needed and time
analysis isnt conducted
}
```

This new dictionary will now be used to rename the columns to a more userfriendly format

```
# Rename the columns in the DataFrame using new dictionary
merged_df.rename(columns=column_rename_dict, inplace=True)

merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3190 entries, 0 to 3862
Data columns (total 41 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   URN
3190 non-null   int64
 1   Local_Authority_Name
3190 non-null   object
 2   Local_Authority_Number
3190 non-null   int64
 3   School_Type
3190 non-null   object
 4   School_College_Type
3190 non-null   object
 5   Religious_Character
2028 non-null   object
 6   Admissions_Policy
2985 non-null   object
 7   School_Gender
3190 non-null   object
 8   Ofsted_Rating
3161 non-null   object
 9   School postcode
3190 non-null   object
 10  FSM_Funding
3190 non-null   int64
 11  Pupil_Premium_Funding
3190 non-null   int64
 12  Pupil_Premium_Pupils
3190 non-null   int64
 13  School_Led_Tutoring_Funding
3188 non-null   float64
 14  Total_Funding
```

```
3190 non-null   float64
 15  Group_UID
2540 non-null   float64
 16  Group_ID
2540 non-null   object
 17  School_Name
2540 non-null   object
 18  Trust_Name
2540 non-null   object
 19  Attainment8
3190 non-null   float64
 20  Progress8
3187 non-null   float64
 21  Percent_Disadvantaged_2022
3190 non-null   int64
 22  Percent_Not_Disadvantaged_2022
3190 non-null   int64
 23  Percent_Disadvantaged_Strong_Passes
3145 non-null   float64
 24  Percent_Not_Disadvantaged_Strong_Passes
3147 non-null   float64
 25  Average Attainment 8 score per non-disadvantaged pupil  - 2022
3149 non-null   float64
 26  Progress8_NonDisadvantaged_2022
3147 non-null   float64
 27  Attainment8_Disadvantaged_2022
3147 non-null   float64
 28  Progress8_Disadvantaged_2022
3139 non-null   float64
 29  Progress8_Maths_Disadvantaged
3134 non-null   float64
 30  Progress8_English_Disadvantaged
3134 non-null   float64
 31  Progress8_Maths_NonDisadvantaged
3142 non-null   float64
 32  Progress8_English_NonDisadvantaged
3142 non-null   float64
 33  trust_name
1077 non-null   object
 34  Trust_UID
1077 non-null   float64
 35  Trust_ID
1077 non-null   object
 36  Num_Academies_Performance
1077 non-null   float64
 37  Num_Pupils_KS4_Performance
1077 non-null   float64
 38  Avg_Attainment8_KS4_Weighted
1077 non-null   float64
```

```
 39  Progress8_Adjusted_Weighted
1077 non-null   float64
 40  nan
1077 non-null   float64
dtypes: float64(21), int64(7), object(13)
memory usage: 1.0+ MB
```

```
merged_df.head()
```

```
      URN Local_Authority_Name  Local_Authority_Number
School_Type  \
0  100049               Camden                     202  Community
school
1  100050               Camden                     202  Community
school
2  100051               Camden                     202  Community
school
3  100052               Camden                     202  Community
school
4  100053               Camden                     202  Community
school

  School_College_Type Religious_Character Admissions_Policy
School_Gender  \
0   Maintained school      Does not apply     Non-selective
Mixed
1   Maintained school      Does not apply     Non-selective
Girls
2   Maintained school      Does not apply     Non-selective
Mixed
3   Maintained school      Does not apply     Non-selective
Mixed
4   Maintained school      Does not apply     Non-selective
Mixed

  Ofsted_Rating School postcode  ...  Progress8_Maths_NonDisadvantaged
\
0          Good         NW3 2BQ  ...                              0.30

1   Outstanding         NW5 1RL  ...                              0.74

2          Good         NW1 1RX  ...                              0.77

3          Good         NW2 3RT  ...                              0.38

4          Good         NW5 1UJ  ...                              0.39


   Progress8_English_NonDisadvantaged  trust_name  Trust_UID  Trust_ID
\
0                                0.01         NaN        NaN       NaN

1                                0.93         NaN        NaN       NaN

2                                0.10         NaN        NaN       NaN

3                                0.19         NaN        NaN       NaN

4                                0.35         NaN        NaN       NaN


   Num_Academies_Performance Num_Pupils_KS4_Performance  \
0                        NaN                        NaN
1                        NaN                        NaN
2                        NaN                        NaN
3                        NaN                        NaN
4                        NaN                        NaN

  Avg_Attainment8_KS4_Weighted Progress8_Adjusted_Weighted  NaN
0                          NaN                         NaN  NaN
1                          NaN                         NaN  NaN
2                          NaN                         NaN  NaN
3                          NaN                         NaN  NaN
4                          NaN                         NaN  NaN

[5 rows x 41 columns]
```

Add Deprivation Index

I will now add the deprivation information from the Ministry of Housing, Communities and Local Government. The website return deprivation information based on a postcode. However before that, I will create a variable from the school demographics, which has a postcode for each school's URN

```
school_demographics_df.columns

Index(['URN', 'LANAME', 'LA', 'SCHOOLTYPE', 'MINORGROUP', 'RELCHAR',
'ADMPOL',
       'GENDER', 'OFSTEDRATING', 'POSTCODE'],
      dtype='object')

urn_postcode = school_demographics_df[['URN','POSTCODE']]
urn_postcode.dropna(inplace=True)

C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\3856292752.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
```

```
returning-a-view-versus-a-copy
  urn_postcode.dropna(inplace=True)
```

```
urn_postcode.drop_duplicates(subset='URN', inplace=True) # drop
duplicates
total_postcodes = len(urn_postcode)
total_postcodes
```

```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\3769423262.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  urn_postcode.drop_duplicates(subset='URN', inplace=True) # drop
duplicates
```

```
25112
```

As the MHCLG website only allows for 10,000 postcodes to be uploaded at a time, I will need to split the urn_postcode into smaller groups, to upload to the website

```
total_postcode_1 = urn_postcode.iloc[0:9000]
total_postcode_2 = urn_postcode.iloc[9001:18000]
total_postcode_3= urn_postcode.iloc[18001:25135]

output_path1 = 'data/urn_postcode_list1.csv'
output_path2 = 'data/urn_postcode_list2.csv'
output_path3 = 'data/urn_postcode_list3.csv'
total_postcode_1['POSTCODE'].to_csv(output_path1, index=False)
total_postcode_2['POSTCODE'].to_csv(output_path2, index=False)
total_postcode_3['POSTCODE'].to_csv(output_path3, index=False)
```

Read and convertthe deprivation data download from the MHCLG website into data frames

```
deprivation_by_index1 = pd.read_csv(r'data\deprivation-by-postcode
(1).csv')
deprivation_by_index2 = pd.read_csv(r'data\deprivation-by-postcode
(2).csv')
deprivation_by_index3 = pd.read_csv(r'data\deprivation-by-postcode
(3).csv')
```

Check on the shape of the data

```
deprivation_by_index1.shape, deprivation_by_index2.shape,
deprivation_by_index3.shape
```

```
((9001, 28), (9000, 28), (7135, 28))
```

Combine the three data frames together

```
combined_deprivation = pd.concat([deprivation_by_index1,
deprivation_by_index2, deprivation_by_index3], ignore_index=True)

combined_deprivation.shape

(25136, 28)
```

Check columns and data types

```
combined_deprivation.dtypes

Postcode                                    object
Postcode Status                             object
LSOA code                                   object
LSOA Name                                   object
Index of Multiple Deprivation Rank         float64
Index of Multiple Deprivation Decile       float64
Income Rank                                float64
Income Decile                              float64
Income Score                               float64
Employment Rank                            float64
Employment Decile                          float64
Employment Score                           float64
Education and Skills Rank                  float64
Education and Skills Decile                float64
Health and Disability Rank                 float64
Health and Disability Decile               float64
Crime Rank                                 float64
Crime Decile                               float64
Barriers to Housing and Services Rank      float64
Barriers to Housing and Services Decile    float64
Living Environment Rank                    float64
Living Environment Decile                  float64
IDACI Rank                                 float64
IDACI Decile                               float64
IDACI Score                                float64
IDAOPI Rank                                float64
IDAOPI Decile                              float64
IDAOPI Score                               float64
dtype: object

urn_postcode.dtypes

URN          int64
POSTCODE    object
dtype: object
```

```
#rename postcode to match
combined_deprivation.rename(columns={'Postcode':'POSTCODE'},
inplace=True)
```

I can now combine deptivation with school URN fields

```
deprivation_urn = combined_deprivation.merge(urn_postcode,
on='POSTCODE', how='inner')
#I selected an inner join as I am only interested in deprivation data
that can be linked to a postcode and URN
deprivation_urn.shape
deprivation_urn.head()
```

```
   POSTCODE Postcode Status  LSOA code                      LSOA Name
\
0  EC3A 5DE            Live  E01032739  City of London 001F E01032739

1  EC2Y 8BB            Live  E01000002  City of London 001B E01000002

2  EC4M 9AD            Live  E01032739  City of London 001F E01032739

3  EC4V 3AL            Live  E01032739  City of London 001F E01032739

4  WC1H 9EG            Live  E01000941          Camden 025C E01000941


   Index of Multiple Deprivation Rank  Index of Multiple Deprivation
Decile  \
0                             20391.0
7.0
1                             30379.0
10.0
2                             20391.0
7.0
3                             20391.0
7.0
4                              4860.0
2.0

   Income Rank  Income Decile  Income Score  Employment Rank  ...  \
0      32638.0           10.0         0.014          32727.0  ...
1      29901.0           10.0         0.034          31190.0  ...
2      32638.0           10.0         0.014          32727.0  ...
3      32638.0           10.0         0.014          32727.0  ...
4       3178.0            1.0         0.271           4445.0  ...

   Barriers to Housing and Services Decile  Living Environment Rank  \
0                                      1.0                   2040.0
1                                      4.0                  13070.0
2                                      1.0                   2040.0
```

```
3                                        1.0                     2040.0
4                                        5.0                     6394.0

   Living Environment Decile  IDACI Rank  IDACI Decile  IDACI Score  \
0                        1.0     32644.0          10.0        0.010
1                        4.0     29682.0          10.0        0.037
2                        1.0     32644.0          10.0        0.010
3                        1.0     32644.0          10.0        0.010
4                        2.0      5293.0           2.0        0.281

   IDAOPI Rank  IDAOPI Decile  IDAOPI Score     URN
0      31389.0           10.0         0.035  100000
1      31938.0           10.0         0.030  100001
2      31389.0           10.0         0.035  100002
3      31389.0           10.0         0.035  100003
4       1139.0            1.0         0.446  100008

[5 rows x 29 columns]
```

Of the various columns available from the MHCLG, I will use Index of Multiple Deprivation Decile as this gives a values between 1-10 for each postcode or small geographic areas know as LSOA (lower-layer super output areas); with decile 1 represting the most 10% of deprived areas and a decile of 10 representing the least deprived areas. The multiple deprivation index is calculated from several domains including income, employment, education, health, crime, barriers to housing and services and living environment. The LSOA Name is also selected to verify against merged_df columns when combined later, as a data integrity measure.

```
deprivation_urn= deprivation_urn[['Index of Multiple Deprivation
Decile','LSOA Name','POSTCODE', 'URN']
]

merged_df = merged_df.merge(deprivation_urn, on= 'URN', how ='inner')
# i will do an inner join here as deprivation is an essential criteria
for analysis

merged_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3685 entries, 0 to 3684
Data columns (total 44 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   URN
3685 non-null   int64
 1   Local_Authority_Name
3685 non-null   object
 2   Local_Authority_Number
3685 non-null   int64
```

```
 3   School_Type
3685 non-null   object
 4   School_College_Type
3685 non-null   object
 5   Religious_Character
2353 non-null   object
 6   Admissions_Policy
3440 non-null   object
 7   School_Gender
3685 non-null   object
 8   Ofsted_Rating
3653 non-null   object
 9   School postcode
3685 non-null   object
 10  FSM_Funding
3685 non-null   int64
 11  Pupil_Premium_Funding
3685 non-null   int64
 12  Pupil_Premium_Pupils
3685 non-null   int64
 13  School_Led_Tutoring_Funding
3681 non-null   float64
 14  Total_Funding
3685 non-null   float64
 15  Group_UID
2893 non-null   float64
 16  Group_ID
2893 non-null   object
 17  School_Name
2893 non-null   object
 18  Trust_Name
2893 non-null   object
 19  Attainment8
3685 non-null   float64
 20  Progress8
3682 non-null   float64
 21  Percent_Disadvantaged_2022
3685 non-null   int64
 22  Percent_Not_Disadvantaged_2022
3685 non-null   int64
 23  Percent_Disadvantaged_Strong_Passes
3636 non-null   float64
 24  Percent_Not_Disadvantaged_Strong_Passes
3639 non-null   float64
 25  Average Attainment 8 score per non-disadvantaged pupil  - 2022
3642 non-null   float64
 26  Progress8_NonDisadvantaged_2022
3640 non-null   float64
 27  Attainment8_Disadvantaged_2022
```

```
3640 non-null   float64
 28  Progress8_Disadvantaged_2022
3629 non-null   float64
 29  Progress8_Maths_Disadvantaged
3624 non-null   float64
 30  Progress8_English_Disadvantaged
3624 non-null   float64
 31  Progress8_Maths_NonDisadvantaged
3632 non-null   float64
 32  Progress8_English_NonDisadvantaged
3632 non-null   float64
 33  trust_name
1253 non-null   object
 34  Trust_UID
1253 non-null   float64
 35  Trust_ID
1253 non-null   object
 36  Num_Academies_Performance
1253 non-null   float64
 37  Num_Pupils_KS4_Performance
1253 non-null   float64
 38  Avg_Attainment8_KS4_Weighted
1253 non-null   float64
 39  Progress8_Adjusted_Weighted
1253 non-null   float64
 40  nan
1253 non-null   float64
 41  Index of Multiple Deprivation Decile
3682 non-null   float64
 42  LSOA Name
3682 non-null   object
 43  POSTCODE
3685 non-null   object
dtypes: float64(22), int64(7), object(15)
memory usage: 1.2+ MB
```

Feature Engineering - Gaps

A number of features are needed as part of the analysis which include gaps between disadavantaged and advantaged puppils, and pupil premium funding per pupil. These will therfore be feature engineered

```
# Gap between progress 8 scores
merged_df['progress8_gap'] =
merged_df['Progress8_NonDisadvantaged_2022']-
merged_df['Progress8_Disadvantaged_2022']

#Gap between attainment 8
merged_df['attainment8_gap'] = merged_df['Average Attainment 8 score
per non-disadvantaged pupil  - 2022']-
merged_df['Attainment8_Disadvantaged_2022']


#Gap between maths scores
merged_df['maths_gap'] =
merged_df['Progress8_Maths_NonDisadvantaged']-
merged_df['Progress8_Maths_Disadvantaged']

#Gap between English scores
merged_df['english_gap'] =
merged_df['Progress8_English_NonDisadvantaged']-
merged_df['Progress8_English_Disadvantaged']

#Gap between 5 GCSE strong pass percentages
merged_df['5_GCSE_gap'] =
merged_df['Percent_Not_Disadvantaged_Strong_Passes'] -
merged_df['Percent_Disadvantaged_Strong_Passes']

#Pupil premium per pupil calculation
merged_df['pupilpremium_per_pupil'] =
merged_df['Pupil_Premium_Funding'] / merged_df['Pupil_Premium_Pupils']
```

Another check for NaN values and duplicates given earlier data wrangling work

```
merged_df.isna().sum()
```

```
URN                                                                  0
Local_Authority_Name                                                 0
Local_Authority_Number                                               0
School_Type                                                          0
School_College_Type                                                  0
Religious_Character                                               1332
Admissions_Policy                                                  245
School_Gender                                                        0
Ofsted_Rating                                                       32
School postcode                                                      0
FSM_Funding                                                          0
Pupil_Premium_Funding                                                0
Pupil_Premium_Pupils                                                 0
School_Led_Tutoring_Funding                                          4
Total_Funding                                                        0
Group_UID                                                          792
Group_ID                                                           792
School_Name                                                        792
Trust_Name                                                         792
Attainment8                                                          0
Progress8                                                            3
Percent_Disadvantaged_2022                                           0
Percent_Not_Disadvantaged_2022                                       0
```

```
Percent_Disadvantaged_Strong_Passes                                 49
Percent_Not_Disadvantaged_Strong_Passes                             46
Average Attainment 8 score per non-disadvantaged pupil  - 2022      43
Progress8_NonDisadvantaged_2022                                     45
Attainment8_Disadvantaged_2022                                      45
Progress8_Disadvantaged_2022                                        56
Progress8_Maths_Disadvantaged                                       61
Progress8_English_Disadvantaged                                     61
Progress8_Maths_NonDisadvantaged                                    53
Progress8_English_NonDisadvantaged                                  53
trust_name                                                        2432
Trust_UID                                                         2432
Trust_ID                                                          2432
Num_Academies_Performance                                         2432
Num_Pupils_KS4_Performance                                        2432
Avg_Attainment8_KS4_Weighted                                      2432
Progress8_Adjusted_Weighted                                       2432
NaN                                                               2432
Index of Multiple Deprivation Decile                                 3
LSOA Name                                                            3
POSTCODE                                                             0
progress8_gap                                                       57
attainment8_gap                                                     45
maths_gap                                                           65
english_gap                                                         65
5_GCSE_gap                                                          49
pupilpremium_per_pupil                                               0
dtype: int64
```

Not all NaN rows need to be dropped. Essential ones are URN and 'Progress8_Maths_Disadvantaged','Index of Multiple Deprivation Decile','Progress8_Maths_NonDisadvantaged','Progress8_Disadvantaged_2022','Progress8_NonDisadvantaged_2022','School_Name', which will impact analysis.

```
merged_df.isna().sum()
merged_df =
merged_df.dropna(subset=['Progress8_Maths_Disadvantaged','Index of
Multiple Deprivation
Decile','Progress8_Maths_NonDisadvantaged','Progress8_Disadvantaged_20
22','Progress8_NonDisadvantaged_2022','School_Name'])
merged_df.isna().sum()
```

```
URN                                                                  0
Local_Authority_Name                                                 0
Local_Authority_Number                                               0
School_Type                                                          0
School_College_Type                                                  0
Religious_Character                                               1126
Admissions_Policy                                                  235
```

```
School_Gender                                                          0
Ofsted_Rating                                                         32
School postcode                                                        0
FSM_Funding                                                            0
Pupil_Premium_Funding                                                  0
Pupil_Premium_Pupils                                                   0
School_Led_Tutoring_Funding                                            4
Total_Funding                                                          0
Group_UID                                                              0
Group_ID                                                               0
School_Name                                                            0
Trust_Name                                                             0
Attainment8                                                            0
Progress8                                                              0
Percent_Disadvantaged_2022                                             0
Percent_Not_Disadvantaged_2022                                         0
Percent_Disadvantaged_Strong_Passes                                    0
Percent_Not_Disadvantaged_Strong_Passes                                0
Average Attainment 8 score per non-disadvantaged pupil  - 2022         0
Progress8_NonDisadvantaged_2022                                        0
Attainment8_Disadvantaged_2022                                         0
Progress8_Disadvantaged_2022                                           0
Progress8_Maths_Disadvantaged                                          0
Progress8_English_Disadvantaged                                        0
Progress8_Maths_NonDisadvantaged                                       0
Progress8_English_NonDisadvantaged                                     0
trust_name                                                          1579
Trust_UID                                                           1579
Trust_ID                                                            1579
Num_Academies_Performance                                           1579
Num_Pupils_KS4_Performance                                          1579
Avg_Attainment8_KS4_Weighted                                        1579
Progress8_Adjusted_Weighted                                         1579
NaN                                                                 1579
Index of Multiple Deprivation Decile                                   0
LSOA Name                                                              0
POSTCODE                                                               0
progress8_gap                                                          0
attainment8_gap                                                        0
maths_gap                                                              0
english_gap                                                            0
5_GCSE_gap                                                             0
pupilpremium_per_pupil                                                 0
dtype: int64
```

Remaining NaN values:

- Religious_Character 1126
- Admissions_Policy 235

- Ofsted_Rating 32

These cant be filled with a mean, median, mode or 0, and as they are categorical variables, I am not too concerned for now. To check data integrity, I will need to ensure, each row for each school as a unique URN

Also check and remove duplicates based on URN so each school has only 1 row

```
duplicate_urns = merged_df[merged_df.duplicated('URN', keep=False)]
#keep= False will mark all duplicates as True regardless of position
duplicate_urns.info()

<class 'pandas.core.frame.DataFrame'>
Index: 644 entries, 754 to 3664
Data columns (total 50 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   URN
644 non-null    int64
 1   Local_Authority_Name
644 non-null    object
 2   Local_Authority_Number
644 non-null    int64
 3   School_Type
644 non-null    object
 4   School_College_Type
644 non-null    object
 5   Religious_Character
378 non-null    object
 6   Admissions_Policy
575 non-null    object
 7   School_Gender
644 non-null    object
 8   Ofsted_Rating
638 non-null    object
 9   School postcode
644 non-null    object
 10  FSM_Funding
644 non-null    int64
 11  Pupil_Premium_Funding
644 non-null    int64
 12  Pupil_Premium_Pupils
644 non-null    int64
 13  School_Led_Tutoring_Funding
640 non-null    float64
 14  Total_Funding
644 non-null    float64
 15  Group_UID
```

```
644 non-null    float64
 16  Group_ID
644 non-null    object
 17  School_Name
644 non-null    object
 18  Trust_Name
644 non-null    object
 19  Attainment8
644 non-null    float64
 20  Progress8
644 non-null    float64
 21  Percent_Disadvantaged_2022
644 non-null    int64
 22  Percent_Not_Disadvantaged_2022
644 non-null    int64
 23  Percent_Disadvantaged_Strong_Passes
644 non-null    float64
 24  Percent_Not_Disadvantaged_Strong_Passes
644 non-null    float64
 25  Average Attainment 8 score per non-disadvantaged pupil  - 2022
644 non-null    float64
 26  Progress8_NonDisadvantaged_2022
644 non-null    float64
 27  Attainment8_Disadvantaged_2022
644 non-null    float64
 28  Progress8_Disadvantaged_2022
644 non-null    float64
 29  Progress8_Maths_Disadvantaged
644 non-null    float64
 30  Progress8_English_Disadvantaged
644 non-null    float64
 31  Progress8_Maths_NonDisadvantaged
644 non-null    float64
 32  Progress8_English_NonDisadvantaged
644 non-null    float64
 33  trust_name
316 non-null    object
 34  Trust_UID
316 non-null    float64
 35  Trust_ID
316 non-null    object
 36  Num_Academies_Performance
316 non-null    float64
 37  Num_Pupils_KS4_Performance
316 non-null    float64
 38  Avg_Attainment8_KS4_Weighted
316 non-null    float64
 39  Progress8_Adjusted_Weighted
316 non-null    float64
```

```
 40  nan
316 non-null    float64
 41  Index of Multiple Deprivation Decile
644 non-null    float64
 42  LSOA Name
644 non-null    object
 43  POSTCODE
644 non-null    object
 44  progress8_gap
644 non-null    float64
 45  attainment8_gap
644 non-null    float64
 46  maths_gap
644 non-null    float64
 47  english_gap
644 non-null    float64
 48  5_GCSE_gap
644 non-null    float64
 49  pupilpremium_per_pupil
644 non-null    float64
dtypes: float64(28), int64(7), object(15)
memory usage: 256.6+ KB
```

```
#drop duplicates
merged_df = merged_df.drop_duplicates(subset='URN')
duplicate_urns = merged_df[merged_df.duplicated('URN', keep=False)]
#keep= False will mark all duplicates as True regardless of position
duplicate_urns.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 0 entries
Data columns (total 50 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   URN                                                              0
non-null      int64
 1   Local_Authority_Name                                             0
non-null      object
 2   Local_Authority_Number                                           0
non-null      int64
 3   School_Type                                                      0
non-null      object
 4   School_College_Type                                              0
non-null      object
 5   Religious_Character                                              0
non-null      object
 6   Admissions_Policy                                                0
non-null      object
```

```
 7   School_Gender                                                  0
non-null      object
 8   Ofsted_Rating                                                  0
non-null      object
 9   School postcode                                                0
non-null      object
 10  FSM_Funding                                                    0
non-null      int64
 11  Pupil_Premium_Funding                                          0
non-null      int64
 12  Pupil_Premium_Pupils                                           0
non-null      int64
 13  School_Led_Tutoring_Funding                                    0
non-null      float64
 14  Total_Funding                                                  0
non-null      float64
 15  Group_UID                                                      0
non-null      float64
 16  Group_ID                                                       0
non-null      object
 17  School_Name                                                    0
non-null      object
 18  Trust_Name                                                     0
non-null      object
 19  Attainment8                                                    0
non-null      float64
 20  Progress8                                                      0
non-null      float64
 21  Percent_Disadvantaged_2022                                     0
non-null      int64
 22  Percent_Not_Disadvantaged_2022                                 0
non-null      int64
 23  Percent_Disadvantaged_Strong_Passes                            0
non-null      float64
 24  Percent_Not_Disadvantaged_Strong_Passes                        0
non-null      float64
 25  Average Attainment 8 score per non-disadvantaged pupil  - 2022  0
non-null      float64
 26  Progress8_NonDisadvantaged_2022                                0
non-null      float64
 27  Attainment8_Disadvantaged_2022                                 0
non-null      float64
 28  Progress8_Disadvantaged_2022                                   0
non-null      float64
 29  Progress8_Maths_Disadvantaged                                  0
non-null      float64
 30  Progress8_English_Disadvantaged                                0
non-null      float64
 31  Progress8_Maths_NonDisadvantaged                               0
```

```
non-null      float64
 32  Progress8_English_NonDisadvantaged                              0
non-null      float64
 33  trust_name                                                      0
non-null      object
 34  Trust_UID                                                       0
non-null      float64
 35  Trust_ID                                                        0
non-null      object
 36  Num_Academies_Performance                                       0
non-null      float64
 37  Num_Pupils_KS4_Performance                                      0
non-null      float64
 38  Avg_Attainment8_KS4_Weighted                                    0
non-null      float64
 39  Progress8_Adjusted_Weighted                                     0
non-null      float64
 40  nan                                                             0
non-null      float64
 41  Index of Multiple Deprivation Decile                            0
non-null      float64
 42  LSOA Name                                                       0
non-null      object
 43  POSTCODE                                                        0
non-null      object
 44  progress8_gap                                                   0
non-null      float64
 45  attainment8_gap                                                 0
non-null      float64
 46  maths_gap                                                       0
non-null      float64
 47  english_gap                                                     0
non-null      float64
 48  5_GCSE_gap                                                      0
non-null      float64
 49  pupilpremium_per_pupil                                          0
non-null      float64
dtypes: float64(28), int64(7), object(15)
memory usage: 0.0+ bytes
```

Descriptive Statistics

Before delving into investigating the merged_df data, I will conduct some basic descript statistics to get a feel of the distribution and spread of the data

```
#distribution of deprivation decile
merged_df['Index of Multiple Deprivation Decile'].value_counts()
```

```
Index of Multiple Deprivation Decile
9.0     275
4.0     258
3.0     255
2.0     253
7.0     249
10.0    248
8.0     246
5.0     245
6.0     236
1.0     204
Name: count, dtype: int64
```

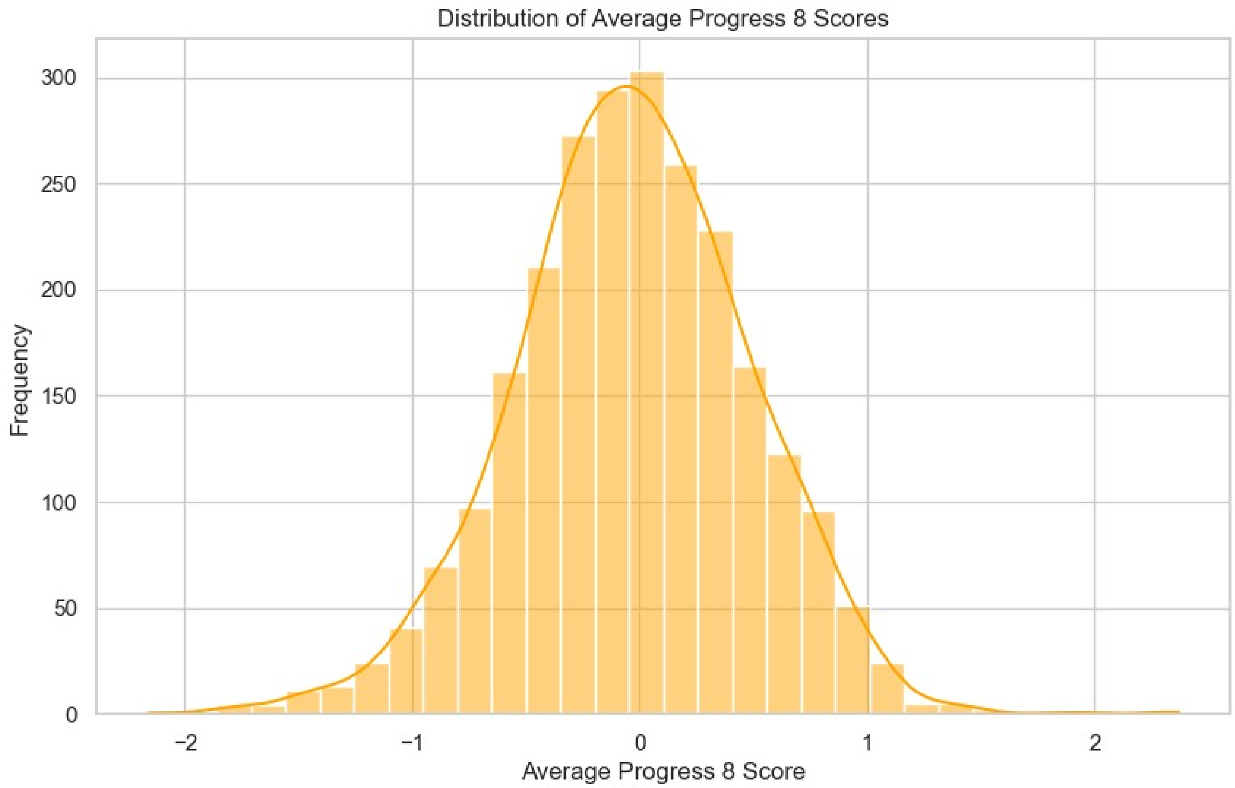
```
## Descriptive Statistics
descriptive_stats = merged_df[['Progress8', 'Attainment8',
                                 'Total_Funding',
'Pupil_Premium_Funding', 'School_Led_Tutoring_Funding','Index of
Multiple Deprivation Decile' ]].describe()
print(descriptive_stats)
```

```
         Progress8  Attainment8  Total_Funding  Pupil_Premium_Funding
\
count  2469.000000  2469.000000   2.469000e+03           2.469000e+03

mean     -0.033925    46.391009   6.360708e+06           2.538846e+05

std       0.516867     8.851594   2.156262e+06           1.533941e+05

min      -2.160000    12.000000   4.100040e+05           1.379000e+04

25%      -0.360000    40.700000   4.946905e+06           1.408550e+05

50%      -0.030000    45.600000   6.317928e+06           2.260580e+05

75%       0.310000    51.100000   7.629306e+06           3.329300e+05

max       2.370000    86.400000   1.770485e+07           1.098175e+06


       School_Led_Tutoring_Funding  Index of Multiple Deprivation
Decile
count                  2467.000000
2469.000000
mean                  43365.633563
5.594978
std                   25025.220705
2.846063
min                    2430.000000
1.000000
25%                   24916.500000
3.000000
```

```
50%                        39042.000000
6.000000
75%                        56587.500000
8.000000
max                       162450.000000
10.000000
```

View the distribution of progress 8 scores nationally

```
# Histogram for Progress 8 scores
plt.figure(figsize=(10, 6))
sns.histplot(merged_df['Progress8'], bins=30, kde=True,
color='orange')
plt.title('Distribution of Average Progress 8 Scores')
plt.xlabel('Average Progress 8 Score')
plt.ylabel('Frequency')


images_dir = 'images'
image_path =
os.path.join(images_dir,'Obj1_progress8_distribution_nationally.png' )
plt.savefig(image_path)

plt.show()
```



Objective 1 Code: Analysing Gaps Between Disadvantaged and Advantaged Pupils

I'll start by finding the mean and gaps of key performance indicators

```
# Calculate mean scores
mean_scores = {
    'Attainment 8 Disadvantaged':
merged_df['Attainment8_Disadvantaged_2022'].mean(),
    'Attainment 8 Non-Disadvantaged': merged_df['Average Attainment 8
score per non-disadvantaged pupil  - 2022'].mean(),
    'Progress 8 Disadvantaged':
merged_df['Progress8_Disadvantaged_2022'].mean(),
    'Progress 8 Non-Disadvantaged':
merged_df['Progress8_NonDisadvantaged_2022'].mean(),
    'Maths Disadvantaged':
merged_df['Progress8_Maths_Disadvantaged'].mean(),
    'Maths Non-Disadvantaged':
merged_df['Progress8_Maths_NonDisadvantaged'].mean(),
    'English Disadvantaged':
merged_df['Progress8_English_Disadvantaged'].mean(),
    'English Non-Disadvantaged':
merged_df['Progress8_English_NonDisadvantaged'].mean(),
    'Percentage Disadvanted EngMaths_95':
merged_df['Percent_Disadvantaged_Strong_Passes'].mean(),
    'Percentage Nondisadv Student EngMaths_95':
merged_df['Percent_Not_Disadvantaged_Strong_Passes'].mean()
}


gaps = {
    'Attainment 8 Gap': merged_df['attainment8_gap'].mean(),
    'Progress 8 Gap': merged_df['progress8_gap'].mean(),
    'Maths Gap':  merged_df['maths_gap'].mean(),
    'English Gap':  merged_df['english_gap'].mean(),
    'percentage_95':  merged_df['5_GCSE_gap'].mean()
}

print("\nMean Scores:")
for key, value in mean_scores.items():
    print(f"{key}: {value:.2f}") #round to 2 decimal places

print("\nGaps Between Groups:")
for key, value in gaps.items():
    print(f"{key}: {value:.2f}") #round to 2 decimal places
```

```
Mean Scores:
Attainment 8 Disadvantaged: 40.22
Attainment 8 Non-Disadvantaged: 51.83
```

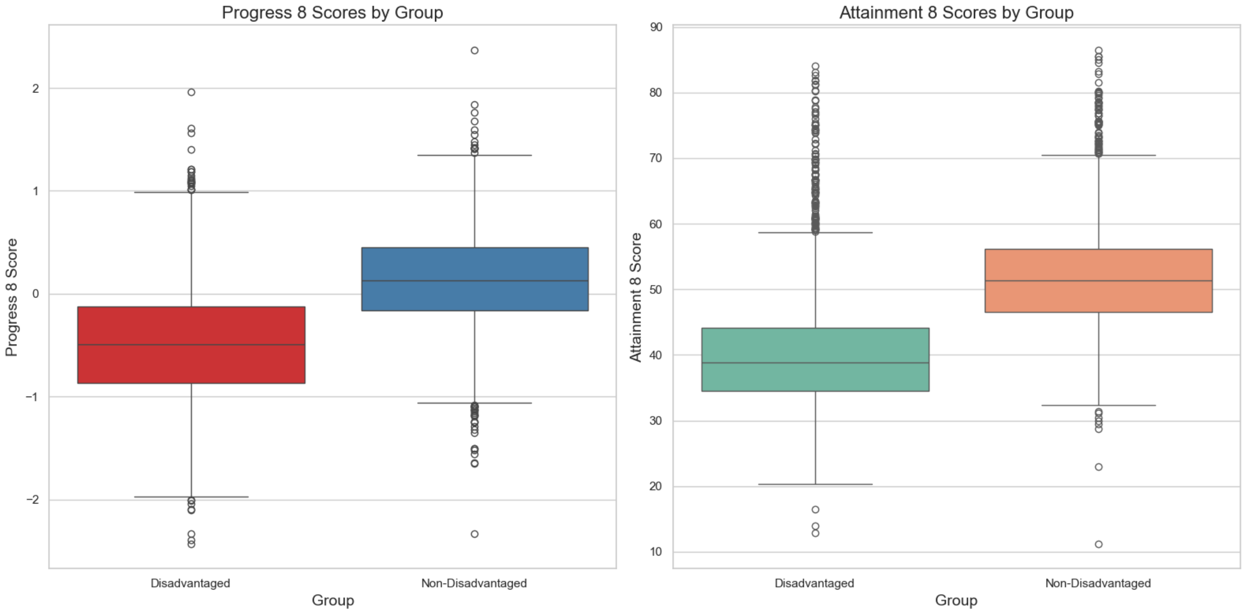
```
Progress 8 Disadvantaged: -0.47
Progress 8 Non-Disadvantaged: 0.13
Maths Disadvantaged: -0.44
Maths Non-Disadvantaged: 0.11
English Disadvantaged: -0.46
English Non-Disadvantaged: 0.12
Percentage Disadvanted EngMaths_95: 28.09
Percentage Nondisadv Student EngMaths_95: 50.01

Gaps Between Groups:
Attainment 8 Gap: 11.61
Progress 8 Gap: 0.60
Maths Gap: 0.55
English Gap: 0.58
percentage_95: 21.92
```

So as to avoid repition, I will interpret the results when discussing objectives later in this notebook. For now, it is worth noting, all the gaps are positive, suggesting disadvantaged pupils are on average are underperforming in every area compared to non-disadvantaged pupils

```
# DataFrames for Progress 8 and Attainment 8

#style
sns.set(style="whitegrid")

# Progress 8 Performance Data
progress8_data = merged_df[['Progress8_Disadvantaged_2022',
'Progress8_NonDisadvantaged_2022']].copy()
progress8_melted = progress8_data.melt(var_name='Group',
value_name='Progress 8 Score')
progress8_melted['Group'] = progress8_melted['Group'].map({
    'Progress8_Disadvantaged_2022': 'Disadvantaged',
    'Progress8_NonDisadvantaged_2022': 'Non-Disadvantaged'
})

# Attainment 8 Performance Data
attainment8_data = merged_df[['Attainment8_Disadvantaged_2022',
'Average Attainment 8 score per non-disadvantaged pupil  -
2022']].copy()
attainment8_melted = attainment8_data.melt(var_name='Group',
value_name='Attainment 8 Score')
attainment8_melted['Group'] = attainment8_melted['Group'].map({
    'Attainment8_Disadvantaged_2022': 'Disadvantaged',
    'Average Attainment 8 score per non-disadvantaged pupil  - 2022':
'Non-Disadvantaged'
})


fig, axes = plt.subplots(1, 2, figsize=(16, 8))
```

```python
# Box Plot for Progress 8 Scores
sns.boxplot(
    x='Group',
    y='Progress 8 Score',
    data=progress8_melted,
    palette="Set1",
    ax=axes[0]
)
axes[0].set_title('Progress 8 Scores by Group', fontsize=16)
axes[0].set_xlabel('Group', fontsize=14)
axes[0].set_ylabel('Progress 8 Score', fontsize=14)

# Box Plot for Attainment 8 Scores
sns.boxplot(
    x='Group',
    y='Attainment 8 Score',
    data=attainment8_melted,
    palette="Set2",
    ax=axes[1]
)
axes[1].set_title('Attainment 8 Scores by Group', fontsize=16)
axes[1].set_xlabel('Group', fontsize=14)
axes[1].set_ylabel('Attainment 8 Score', fontsize=14)
plt.tight_layout() #adjust plot for better fit

#save the file to the images folder
images_dir = 'images'
image_path =
os.path.join(images_dir,'obj1_progress8_attainment8_boxplot.png' )
plt.savefig(image_path)

plt.show()
```

```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\2793708818.py:26:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\2793708818.py:38:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(
```

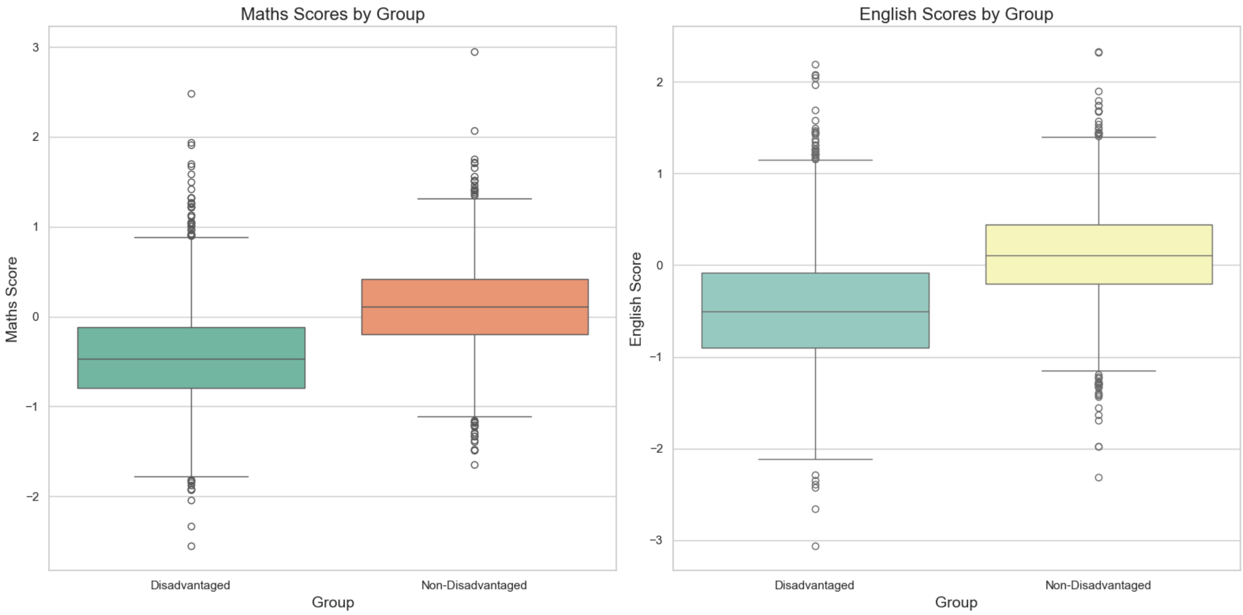


```python
# Calculate summary statistics for Attainment 8 Scores
attainment8_grouped = attainment8_melted.groupby('Group')['Attainment 8 Score']

# Calculate statistics for Attainment 8
attainment8_stats = attainment8_grouped.describe()
attainment8_q1 = attainment8_grouped.quantile(0.25)
attainment8_q3 = attainment8_grouped.quantile(0.75)
attainment8_iqr = attainment8_q3 - attainment8_q1
attainment8_range = attainment8_grouped.max() - attainment8_grouped.min()
attainment8_median = attainment8_grouped.median()
attainment8_min = attainment8_grouped.min()
attainment8_max = attainment8_grouped.max()

#
attainment8_summary = pd.DataFrame({
    'Median': attainment8_median,
    'Q1 (25%)': attainment8_q1,
    'Q3 (75%)': attainment8_q3,
    'IQR': attainment8_iqr,
    'Min': attainment8_min,
    'Max': attainment8_max,
    'Range': attainment8_range
})

print("\nAttainment 8 Scores Summary:")
print(attainment8_summary)


# Calculate summary statistics for Progress 8 Scores
```

```
progress8_grouped = progress8_melted.groupby('Group')['Progress 8
Score']

# Calculate statistics for Progress 8
progress8_stats = progress8_grouped.describe()
progress8_q1 = progress8_grouped.quantile(0.25)
progress8_q3 = progress8_grouped.quantile(0.75)
progress8_iqr = progress8_q3 - progress8_q1
progress8_range = progress8_grouped.max() - progress8_grouped.min()
progress8_median = progress8_grouped.median()
progress8_min = progress8_grouped.min()
progress8_max = progress8_grouped.max()


progress8_summary = pd.DataFrame({
    'Median': progress8_median,
    'Q1 (25%)': progress8_q1,
    'Q3 (75%)': progress8_q3,
    'IQR': progress8_iqr,
    'Min': progress8_min,
    'Max': progress8_max,
    'Range': progress8_range
})


print("\nProgress 8 Scores Summary:")
print(progress8_summary)


Attainment 8 Scores Summary:
                   Median  Q1 (25%)  Q3 (75%)  IQR   Min   Max  Range
Group
Disadvantaged        38.8      34.5      44.2  9.7  12.9  84.1   71.2
Non-Disadvantaged    51.3      46.6      56.2  9.6  11.2  86.5   75.3

Progress 8 Scores Summary:
                   Median  Q1 (25%)  Q3 (75%)   IQR   Min   Max  Range
Group
Disadvantaged       -0.49     -0.87     -0.12  0.75 -2.43  1.96   4.39
Non-Disadvantaged    0.13     -0.16      0.45  0.61 -2.33  2.37   4.70

# Dataframes for Maths and English

# Style
sns.set(style="whitegrid")

# Maths Performance Data
maths_data = merged_df[['Progress8_Maths_Disadvantaged',
'Progress8_Maths_NonDisadvantaged']].copy()
```

```
maths_melted = maths_data.melt(var_name='Group', value_name='Maths
Score')
maths_melted['Group'] = maths_melted['Group'].map({
    'Progress8_Maths_Disadvantaged': 'Disadvantaged',
    'Progress8_Maths_NonDisadvantaged': 'Non-Disadvantaged'
})

# English Performance Data
english_data = merged_df[['Progress8_English_Disadvantaged',
'Progress8_English_NonDisadvantaged']].copy()
english_melted = english_data.melt(var_name='Group',
value_name='English Score')
english_melted['Group'] = english_melted['Group'].map({
    'Progress8_English_Disadvantaged': 'Disadvantaged',
    'Progress8_English_NonDisadvantaged': 'Non-Disadvantaged'
})


fig, axes = plt.subplots(1, 2, figsize=(16, 8))

# Box Plot for Maths Scores
sns.boxplot(
    x='Group',
    y='Maths Score',
    data=maths_melted,
    palette="Set2",
    ax=axes[0]
)
axes[0].set_title('Maths Scores by Group', fontsize=16)
axes[0].set_xlabel('Group', fontsize=14)
axes[0].set_ylabel('Maths Score', fontsize=14)


# Box Plot for English Scores
sns.boxplot(
    x='Group',
    y='English Score',
    data=english_melted,
    palette="Set3",
    ax=axes[1]
)
axes[1].set_title('English Scores by Group', fontsize=16)
axes[1].set_xlabel('Group', fontsize=14)
axes[1].set_ylabel('English Score', fontsize=14)
plt.tight_layout() # Adjust layout for better fit

# Save the combined box plots as a PNG file in images folder

images_dir = 'images'
image_path =
os.path.join(images_dir,'obj1_maths_english_scores_boxplot.png')
plt.savefig(image_path)


plt.show()
```

```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\367155346.py:26:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\367155346.py:39:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(
```



```
# Calculate summary statistics for Maths Scores

maths_grouped = maths_melted.groupby('Group')['Maths Score']

# Calculate statistics for Maths
maths_stats = maths_grouped.describe()
maths_q1 = maths_grouped.quantile(0.25)
maths_q3 = maths_grouped.quantile(0.75)
```

```
maths_iqr = maths_q3 - maths_q1
maths_range = maths_grouped.max() - maths_grouped.min()
maths_median = maths_grouped.median()
maths_min = maths_grouped.min()
maths_max = maths_grouped.max()


maths_summary = pd.DataFrame({
    'Median': maths_median,
    'Q1 (25%)': maths_q1,
    'Q3 (75%)': maths_q3,
    'IQR': maths_iqr,
    'Min': maths_min,
    'Max': maths_max,
    'Range': maths_range
})


print("\nMaths Scores Summary:")
print(maths_summary)


# Calculate summary statistics for English Scores


english_grouped = english_melted.groupby('Group')['English Score']

english_stats = english_grouped.describe()
english_q1 = english_grouped.quantile(0.25)
english_q3 = english_grouped.quantile(0.75)
english_iqr = english_q3 - english_q1
english_range = english_grouped.max() - english_grouped.min()
english_median = english_grouped.median()
english_min = english_grouped.min()
english_max = english_grouped.max()


english_summary = pd.DataFrame({
    'Median': english_median,
    'Q1 (25%)': english_q1,
    'Q3 (75%)': english_q3,
    'IQR': english_iqr,
    'Min': english_min,
    'Max': english_max,
    'Range': english_range
})


print("\nEnglish Scores Summary:")
print(english_summary)
```

```
Maths Scores Summary:
                   Median  Q1 (25%)  Q3 (75%)   IQR   Min   Max  Range
Group
Disadvantaged       -0.47     -0.79     -0.12  0.67 -2.55  2.48   5.03
Non-Disadvantaged    0.11     -0.20      0.42  0.62 -1.65  2.95   4.60

English Scores Summary:
                   Median  Q1 (25%)  Q3 (75%)   IQR   Min   Max  Range
Group
Disadvantaged       -0.50      -0.9     -0.08  0.82 -3.06  2.19   5.25
Non-Disadvantaged    0.11      -0.2      0.44  0.64 -2.31  2.33   4.64
```

Objective 2 Code: Identify and Analyse Outlier Schools in Positive Progress 8 of Disadavantaged Pupils

For simplicity, I have chosen to use the Interquarticle Range approach to identify outliers, rather than Z-score. This also allows for easier visualisation using a boxplot. I will begin by establishing quartiles for a box plot to see the distribtion of progress-8 disadvantaged students and then determine outliers using standard approach of interquartile range. As I am interested in high performing schools, I will only take the positive outlier schools

```
merged_df_2= merged_df.copy() # copy of merged_df is used for data
integrity
merged_df_2.head()
```

```
        URN Local_Authority_Name  Local_Authority_Number
School_Type  \
207  105135           Greenwich                     203   Academy
sponsor led
718  129342            Solihull                     334   Academy
sponsor led
722  130247             Reading                     870   Academy
sponsor led
723  130908       Middlesbrough                     806   Academy
sponsor led
724  130909            Bradford                     380   Academy
sponsor led

    School_College_Type Religious_Character Admissions_Policy
School_Gender  \
207             Academy      Roman Catholic     Non-selective
Mixed
718             Academy      Does not apply     Non-selective
Mixed
722             Academy      Does not apply     Non-selective
Mixed
723             Academy      Does not apply     Non-selective
Mixed
```

```
724             Academy      Does not apply     Non-selective
Mixed

            Ofsted_Rating School postcode  ...        NaN  \
207  Requires improvement         SE2 9PX  ...        NaN
718                  Good         B37 5JS  ...        NaN
722    Serious Weaknesses         RG2 8AF  ...        NaN
723                  Good         TS5 4AG  ...        NaN
724           Outstanding         BD5 7RR  ...   202223.0

     Index of Multiple Deprivation Decile                     LSOA
Name  \
207                                   2.0      Greenwich 003E
E01001579
718                                   3.0       Solihull 007D
E01010144
722                                   2.0        Reading 017D
E01016438
723                                   2.0  Middlesbrough 008D
E01012014
724                                   1.0       Bradford 048B
E01010732

     POSTCODE  progress8_gap  attainment8_gap maths_gap english_gap  \
207   SE2 9PX           0.48              6.9      0.51        0.69
718   B37 5JS           0.59              6.3      0.40        0.91
722   RG2 8AF           0.90             15.3      0.50        0.41
723   TS5 4AG           0.62             10.9      0.78        0.62
724   BD5 7RR           0.49              8.4      0.25        0.40

    5_GCSE_gap  pupilpremium_per_pupil
207       12.0                   985.0
718       30.0                   985.0
722       12.0                   985.0
723       32.0                   985.0
724       10.0                   985.0

[5 rows x 50 columns]

#  Outlier detection of schools in progress 8 performance of
disadavantaged pupils

Q1 = merged_df_2['Progress8_Disadvantaged_2022'].quantile(0.25)
Q3 = merged_df_2['Progress8_Disadvantaged_2022'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
 # only upper bound is taken as we are interested in high-performing
schools
outliers_p8_disadv =
merged_df_2[(merged_df_2['Progress8_Disadvantaged_2022'] >
upper_bound)]
outliers_p8_disadv[['School_Name','Trust_Name','Progress8_Disadvantage
d_2022']].sort_values('Progress8_Disadvantaged_2022', ascending=False)
```

```
                                            School_Name  \
2699                          Michaela Community School
3545                         St Peter's Catholic School
2826                 Tauheedul Islam Girls' High School
3528        Eden Girls'  Leadership Academy, Birmingham
2058                          St Mark's Catholic School
2368                       Sacred Heart Catholic School
2830                             The Hurlingham Academy
2981                          Ealing Fields High School
2884                         Eden Boys' School, Preston
3530  St Francis Xavier School - a Joint Catholic an...
2950                         Bolton Muslim Girls School
1475                        Birmingham Ormiston Academy
3158                  Dartford Grammar School for Girls
1120                    Lancaster Girls' Grammar School
833                         Ashcroft Technology Academy
2162                            Ark Bolingbroke Academy
1279                                    Wilson's School
1919                           Featherstone High School
849                               Wren Academy Finchley
1645                           Bentley Wood High School
2264                                Nishkam High School

                                         Trust_Name  \
2699               MICHAELA COMMUNITY SCHOOLS TRUST
3545                XAVIER CATHOLIC EDUCATION TRUST
2826                                 STAR ACADEMIES
3528                                 STAR ACADEMIES
2058       THE DIOCESE OF WESTMINSTER ACADEMY TRUST
2368                   SACRED HEART CATHOLIC SCHOOL
2830                          UNITED LEARNING TRUST
2981      TWYFORD CHURCH OF ENGLAND ACADEMIES TRUST
2884                                 STAR ACADEMIES
3530       NICHOLAS POSTGATE CATHOLIC ACADEMY TRUST
2950                    PROSPER MULTI ACADEMY TRUST
1475                    BIRMINGHAM ORMISTON ACADEMY
3158                                THE ARETÉ TRUST
1120                LANCASTER GIRLS' GRAMMAR SCHOOL
833   PROSPECT EDUCATION (TECHNOLOGY) TRUST LIMITED
2162                                    ARK SCHOOLS
1279                                WILSON'S SCHOOL
1919                GRAND UNION MULTI ACADEMY TRUST
849                            WREN ACADEMIES TRUST
1645                          THE BENTLEY WOOD TRUST
2264                           NISHKAM SCHOOL TRUST
```

```
      Progress8_Disadvantaged_2022
2699                          1.96
3545                          1.61
2826                          1.56
3528                          1.40
2058                          1.21
2368                          1.21
2830                          1.19
2981                          1.19
2884                          1.15
3530                          1.13
2950                          1.11
1475                          1.10
3158                          1.09
1120                          1.09
833                           1.09
2162                          1.07
1279                          1.07
1919                          1.05
849                           1.02
1645                          1.01
2264                          1.01
```

Similarly, I will repear the process for non-disadvantaged pupils' progress-8 score

```
Q1 = merged_df_2['Progress8_NonDisadvantaged_2022'].quantile(0.25)
Q3 = merged_df_2['Progress8_NonDisadvantaged_2022'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers_p8_adv =
merged_df_2[(merged_df_2['Progress8_NonDisadvantaged_2022'] >
upper_bound)]  # only upper bound is taken as we are interested in
high-performing schools
outliers_p8_adv[['School_Name','Trust_Name','Progress8_NonDisadvantage
d_2022']].sort_values('Progress8_NonDisadvantaged_2022',
ascending=False)
```

```
                                      School_Name  \
2699                     Michaela Community School
3528  Eden Girls'  Leadership Academy, Birmingham
2826           Tauheedul Islam Girls' High School
1748       Hillcrest School and Sixth Form Centre
2779                       Levenshulme High School
815                       Ark King Solomon Academy
1645                      Bentley Wood High School
2883                    Eden Girls' School, Slough
787                            Northampton Academy
```

```
2128                     Avonbourne Girls Academy
2587                             Glenmoor Academy
2830                       The Hurlingham Academy
2722                  Eden Girls' School Coventry
2981                    Ealing Fields High School
3545                   St Peter's Catholic School

                                             Trust_Name  \
2699                   MICHAELA COMMUNITY SCHOOLS TRUST
3528                                     STAR ACADEMIES
2826                                     STAR ACADEMIES
1748             HILLCREST SCHOOL AND SIXTH FORM CENTRE
2779                     EDUCATION AND LEADERSHIP TRUST
815                                         ARK SCHOOLS
1645                             THE BENTLEY WOOD TRUST
2883                                     STAR ACADEMIES
787                               UNITED LEARNING TRUST
2128  AVONBOURNE INTERNATIONAL BUSINESS AND ENTERPRI...
2587                              UNITED LEARNING TRUST
2830                              UNITED LEARNING TRUST
2722                                     STAR ACADEMIES
2981          TWYFORD CHURCH OF ENGLAND ACADEMIES TRUST
3545                    XAVIER CATHOLIC EDUCATION TRUST

      Progress8_NonDisadvantaged_2022
2699                             2.37
3528                             1.84
2826                             1.76
1748                             1.68
2779                             1.59
815                              1.55
1645                             1.48
2883                             1.45
787                              1.42
2128                             1.42
2587                             1.42
2830                             1.42
2722                             1.41
2981                             1.41
3545                             1.37
```

Categorical Variables

To investigate the impact of demographics and socioeconomic influence on outlier schools for Progress 8 - disadvantaged pupils, I will use Religious character, Ofsted Rating, Free School Meal Funding,
School_Led_Tutoring_Funding,pupilpremium_per_pupil,Percent_Not_Disadvantaged_2022,
Percent_Disadvantaged_2022, Index of Multiple Deprivation Decile

```
demographics_columns =
['Religious_Character','Ofsted_Rating','FSM_Funding','School_Led_Tutor
ing_Funding','pupilpremium_per_pupil','Percent_Not_Disadvantaged_2022'
,'Percent_Disadvantaged_2022','Index of Multiple Deprivation Decile']

descriptive_stats_disadv =
outliers_p8_disadv[demographics_columns].describe()
descriptive_stats_disadv.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8 entries, count to max
Data columns (total 6 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   FSM_Funding                           8 non-null      float64
 1   School_Led_Tutoring_Funding           8 non-null      float64
 2   pupilpremium_per_pupil                8 non-null      float64
 3   Percent_Not_Disadvantaged_2022        8 non-null      float64
 4   Percent_Disadvantaged_2022            8 non-null      float64
 5   Index of Multiple Deprivation Decile  8 non-null      float64
dtypes: float64(6)
memory usage: 448.0+ bytes
```

```
descriptive_stats_adv =
outliers_p8_adv[demographics_columns].describe()
descriptive_stats_adv
```

```
         FSM_Funding  School_Led_Tutoring_Funding
pupilpremium_per_pupil  \
count      15.000000                    15.000000
15.000000
mean    86227.333333                 43651.800000
999.404370
std     50555.511646                 21600.087911
58.855940
min         0.000000                 12150.000000
977.869565
25%     50749.000000                 29889.000000
985.000000
50%     82107.000000                 40662.000000
985.000000
75%    122693.500000                 52717.500000
985.000000
max    182830.000000                 86994.000000
1212.046263

       Percent_Not_Disadvantaged_2022  Percent_Disadvantaged_2022  \
count                       15.000000                   15.000000
mean                        66.333333                   33.666667
```

```
std                           14.110111                   14.110111
min                           37.000000                    5.000000
25%                           57.500000                   27.000000
50%                           69.000000                   31.000000
75%                           73.000000                   42.500000
max                           95.000000                   63.000000

       Index of Multiple Deprivation Decile
count                             15.000000
mean                               4.800000
std                                2.980892
min                                1.000000
25%                                2.000000
50%                                4.000000
75%                                7.500000
max                               10.000000
```

Analysis may not be conclusive of the above descriptive statistics as some schools maybe in both groups of outliers: progress 8 outliers for disadvantaged pupils and progress 8 outliers for non-disadvantaged pupils. To better understand the differences, we should differentiate between schools which are

a) only progress 8 outliers for diadavtanged pupils

b) only for advatanged

c) those which are outliers for both.

To differentiate the schools, I will select and split based on their URN numbers

```
#URN list for non-disadvantaged ouliers
nondisadv_outliers= set(outliers_p8_adv['URN'])
nondisadv_outliers
```

```
{134814,
 135242,
 137178,
 137346,
 138193,
 140008,
 140862,
 140958,
 141196,
 141565,
 141617,
 141970,
 142654,
 147201,
 147430}
```

```
#URN list for disadvantaged ouliers

disadv_outliers = set(outliers_p8_disadv['URN'])
disadv_outliers
```

```
{135316,
 135507,
 136381,
 136621,
 136944,
 137178,
 137729,
 137995,
 138267,
 138586,
 138960,
 140862,
 141565,
 141617,
 141971,
 142340,
 142654,
 144100,
 147201,
 147213,
 147430}
```

```
# Define outlier sets
only_disadvp8_outliers = disadv_outliers - nondisadv_outliers
only_nondisadvp8_outliers = nondisadv_outliers - disadv_outliers
both_p8_outliers = disadv_outliers & nondisadv_outliers


merged_df_2['Outlier_Category'] = 'None'
merged_df_2.loc[merged_df['URN'].isin(both_p8_outliers),
'Outlier_Category'] = 'Both'
merged_df_2.loc[merged_df['URN'].isin(only_disadvp8_outliers),
'Outlier_Category'] = 'Only_Disadv'
merged_df_2.loc[merged_df['URN'].isin(only_nondisadvp8_outliers),
'Outlier_Category'] = 'Only_NonDisadv'


category_counts = merged_df_2['Outlier_Category'].value_counts()
print("Distribution of Outlier Categories:")
print(category_counts)
```

```
Distribution of Outlier Categories:
Outlier_Category
None              2440
Only_Disadv         14
Only_NonDisadv       8
Both                 7
Name: count, dtype: int64
```

I will now use the categories to reate an outlier dataframe which can be used for analysing just the progress 8 school outliers against each other

```
outlier_df = merged_df_2[merged_df_2['URN'].isin(both_p8_outliers|
only_nondisadvp8_outliers|only_disadvp8_outliers)]


category_counts = outlier_df['Outlier_Category'].value_counts()
print("Distribution of Outlier Categories:")
print(category_counts)
```

```
Distribution of Outlier Categories:
Outlier_Category
Only_Disadv       14
Only_NonDisadv     8
Both               7
Name: count, dtype: int64
```

Check for null values

```
merged_df_2.isnull().sum()
```

```
URN                                                                  0
Local_Authority_Name                                                 0
Local_Authority_Number                                               0
School_Type                                                          0
School_College_Type                                                  0
Religious_Character                                                984
Admissions_Policy                                                  197
School_Gender                                                        0
Ofsted_Rating                                                       29
School postcode                                                      0
FSM_Funding                                                          0
Pupil_Premium_Funding                                                0
Pupil_Premium_Pupils                                                 0
School_Led_Tutoring_Funding                                          2
Total_Funding                                                        0
Group_UID                                                            0
Group_ID                                                             0
School_Name                                                          0
Trust_Name                                                           0
```

```
Attainment8                                                          0
Progress8                                                            0
Percent_Disadvantaged_2022                                           0
Percent_Not_Disadvantaged_2022                                       0
Percent_Disadvantaged_Strong_Passes                                  0
Percent_Not_Disadvantaged_Strong_Passes                              0
Average Attainment 8 score per non-disadvantaged pupil  - 2022       0
Progress8_NonDisadvantaged_2022                                      0
Attainment8_Disadvantaged_2022                                       0
Progress8_Disadvantaged_2022                                         0
Progress8_Maths_Disadvantaged                                        0
Progress8_English_Disadvantaged                                      0
Progress8_Maths_NonDisadvantaged                                     0
Progress8_English_NonDisadvantaged                                   0
trust_name                                                        1405
Trust_UID                                                         1405
Trust_ID                                                          1405
Num_Academies_Performance                                         1405
Num_Pupils_KS4_Performance                                        1405
Avg_Attainment8_KS4_Weighted                                      1405
Progress8_Adjusted_Weighted                                       1405
NaN                                                               1405
Index of Multiple Deprivation Decile                                 0
LSOA Name                                                            0
POSTCODE                                                             0
progress8_gap                                                        0
attainment8_gap                                                      0
maths_gap                                                            0
english_gap                                                          0
5_GCSE_gap                                                           0
pupilpremium_per_pupil                                               0
Outlier_Category                                                     0
dtype: int64
```

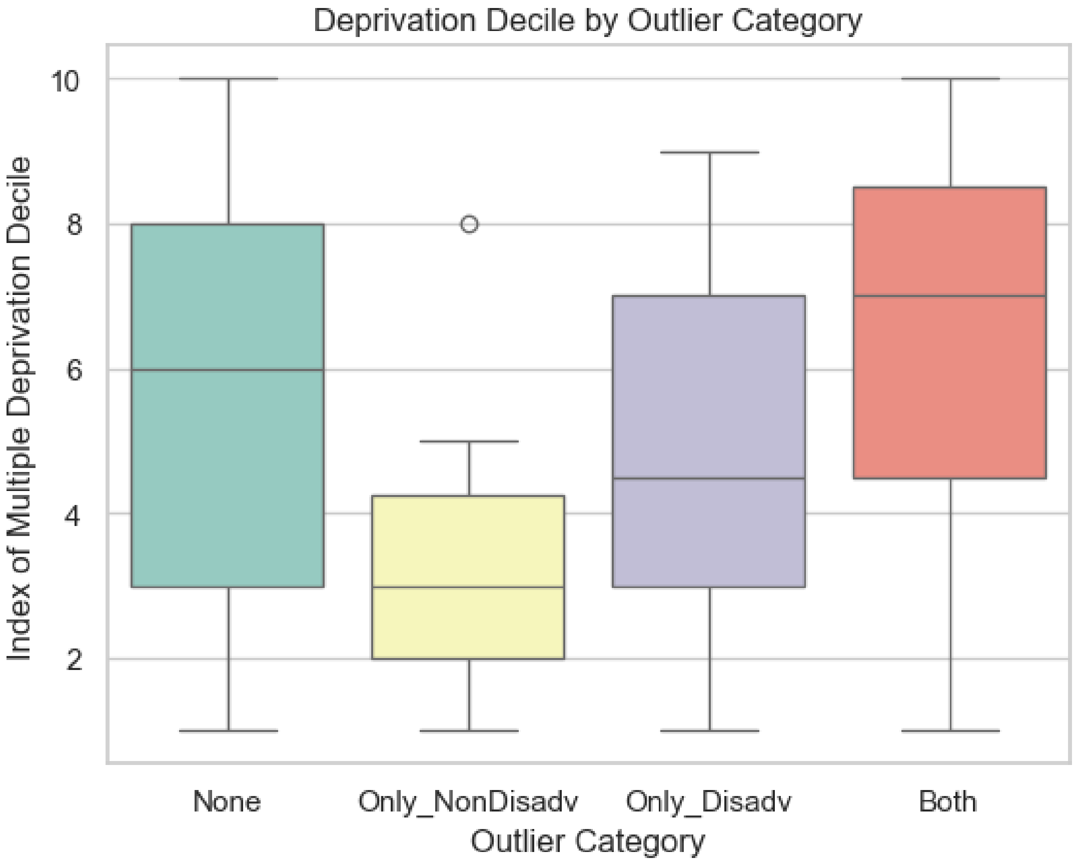
View mean Deprivation Index spread across across all categories of schools

```
outlier_performance_index = merged_df_2.groupby('Outlier_Category')
[['Index of Multiple Deprivation Decile']].mean().reset_index()
outlier_performance_index

  Outlier_Category  Index of Multiple Deprivation Decile
0             Both                              6.285714
1             None                              5.603279
2      Only_Disadv                              5.000000
3   Only_NonDisadv                              3.500000
```

To analyse various fields of the outlier schools, based on their categories, I will plot some graphs and generate summary statistics

```
#represent deprivation index against outlier category

sns.boxplot(
    x='Outlier_Category',
    y='Index of Multiple Deprivation Decile',
    data=merged_df_2,
    palette='Set3'
)
plt.title('Deprivation Decile by Outlier Category')
plt.xlabel('Outlier Category')
plt.ylabel('Index of Multiple Deprivation Decile')

images_dir = 'images'
image_path =
os.path.join(images_dir,'Obj2_Deprivation_by_Outlier_Category.png')
plt.savefig(image_path)

plt.show()
```

```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\1565168097.py:3:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(
```



```
#  calculate descriptive statistics for outlier categories and index
of multiple deprivation

# groupby method used to group merged_df_2 by outlier category, then
select IMDC column for further analysis
grouped = merged_df_2.groupby('Outlier_Category')['Index of Multiple
Deprivation Decile']


median = grouped.median()
q1 = grouped.quantile(0.25)
q3 = grouped.quantile(0.75)
iqr = q3 - q1
minimum = grouped.min()
maximum = grouped.max()
range_ = maximum - minimum


summary = pd.DataFrame({
    'Median': median,
    'Q1 (25%)': q1,
    'Q3 (75%)': q3,
    'IQR': iqr,
    'Min': minimum,
    'Max': maximum,
    'Range': range_
})

#
print("Summary Statistics for 'Index of Multiple Deprivation Decile'
by 'Outlier_Category':")
print(summary)
```

```
Summary Statistics for 'Index of Multiple Deprivation Decile' by
'Outlier_Category':
                  Median  Q1 (25%)  Q3 (75%)   IQR  Min   Max  Range
Outlier_Category
Both                 7.0       4.5      8.50  4.00  1.0  10.0    9.0
None                 6.0       3.0      8.00  5.00  1.0  10.0    9.0
Only_Disadv          4.5       3.0      7.00  4.00  1.0   9.0    8.0
Only_NonDisadv       3.0       2.0      4.25  2.25  1.0   8.0    7.0
```

```
outlier_performance = merged_df_2.groupby('Outlier_Category')
[['Progress8','Progress8_Disadvantaged_2022','Progress8_NonDisadvantag
ed_2022']].mean().reset_index()
outlier_performance
```

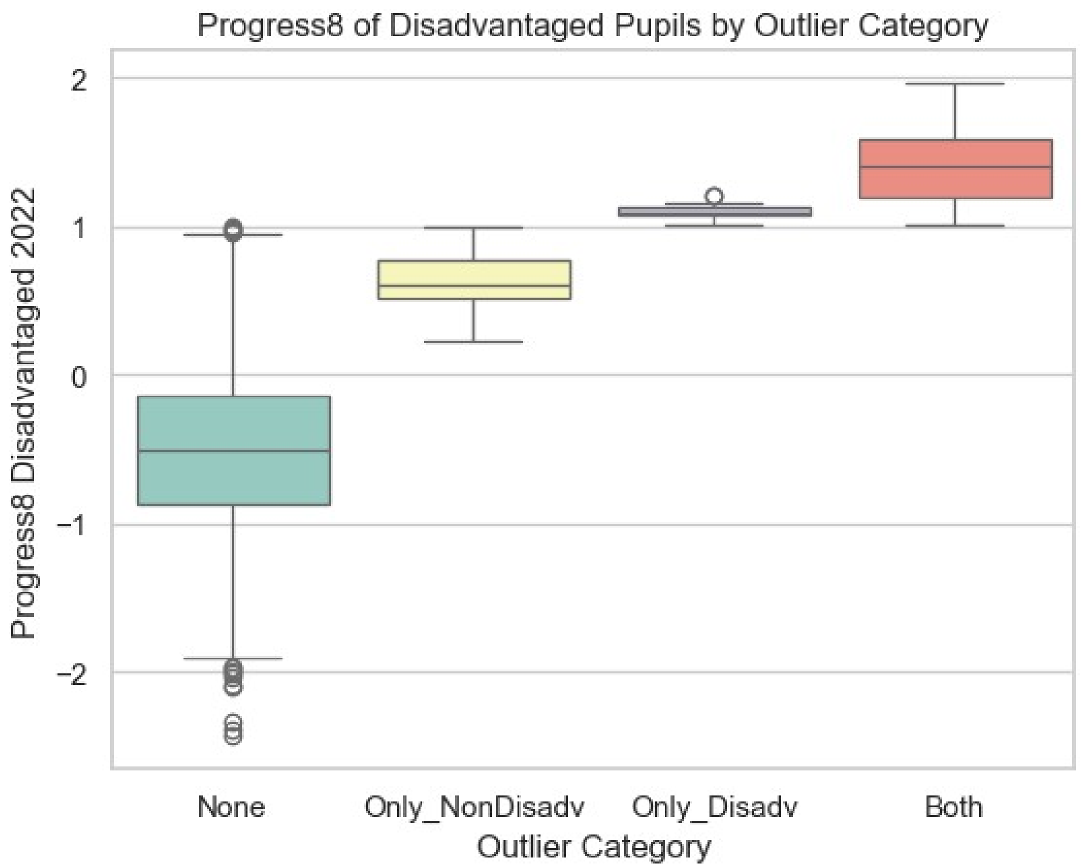
```
  Outlier_Category  Progress8  Progress8_Disadvantaged_2022  \
0             Both   1.614286                      1.417143
1             None  -0.047791                     -0.486266
2      Only_Disadv   0.946429                      1.100000
3   Only_NonDisadv   1.037500                      0.626250

   Progress8_NonDisadvantaged_2022
0                         1.664286
1                         0.118369
2                         0.950714
3                         1.492500
```

Plot box plot of progress 8 of disadvantaed pupils by outlier category

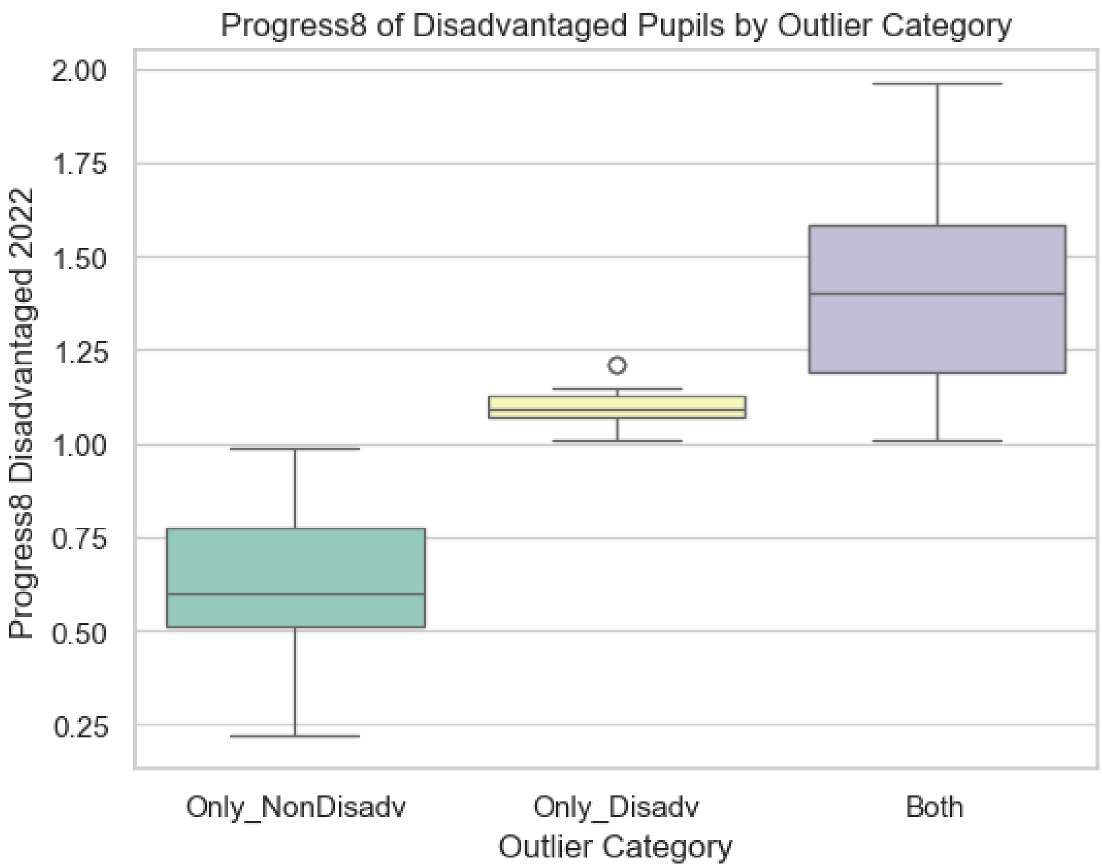
```
sns.boxplot(
    x='Outlier_Category',
    y='Progress8_Disadvantaged_2022',
    data= merged_df_2,
    palette='Set3'
)
plt.title('Progress8 of Disadvantaged Pupils by Outlier Category')
plt.xlabel('Outlier Category')
plt.ylabel('Progress8 Disadvantaged 2022')

plt.show()
```

```
sns.boxplot(
    x='Outlier_Category',
    y='Progress8_Disadvantaged_2022',
    data= outlier_df,
    palette='Set3'
)
plt.title('Progress8 of Disadvantaged Pupils by Outlier Category')
plt.xlabel('Outlier Category')
plt.ylabel('Progress8 Disadvantaged 2022')

images_dir = 'images'
image_path = os.path.join(images_dir,'Obj2_Progress8 of Disadvantaged
Pupils by Outlier Category.png')
plt.savefig(image_path)

plt.show()
```

```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\3617201507.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(
```



```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\3617201507.py:14:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(
```



```
#calculate summary statistics for the box plot above
grouped = merged_df_2.groupby('Outlier_Category')
['Progress8_Disadvantaged_2022']


median = grouped.median()
q1 = grouped.quantile(0.25)
q3 = grouped.quantile(0.75)
iqr = q3 - q1
minimum = grouped.min()
maximum = grouped.max()
range_ = maximum - minimum


summary = pd.DataFrame({
    'Median': median,
    'Q1 (25%)': q1,
    'Q3 (75%)': q3,
    'IQR': iqr,
    'Min': minimum,
    'Max': maximum,
    'Range': range_
})
```

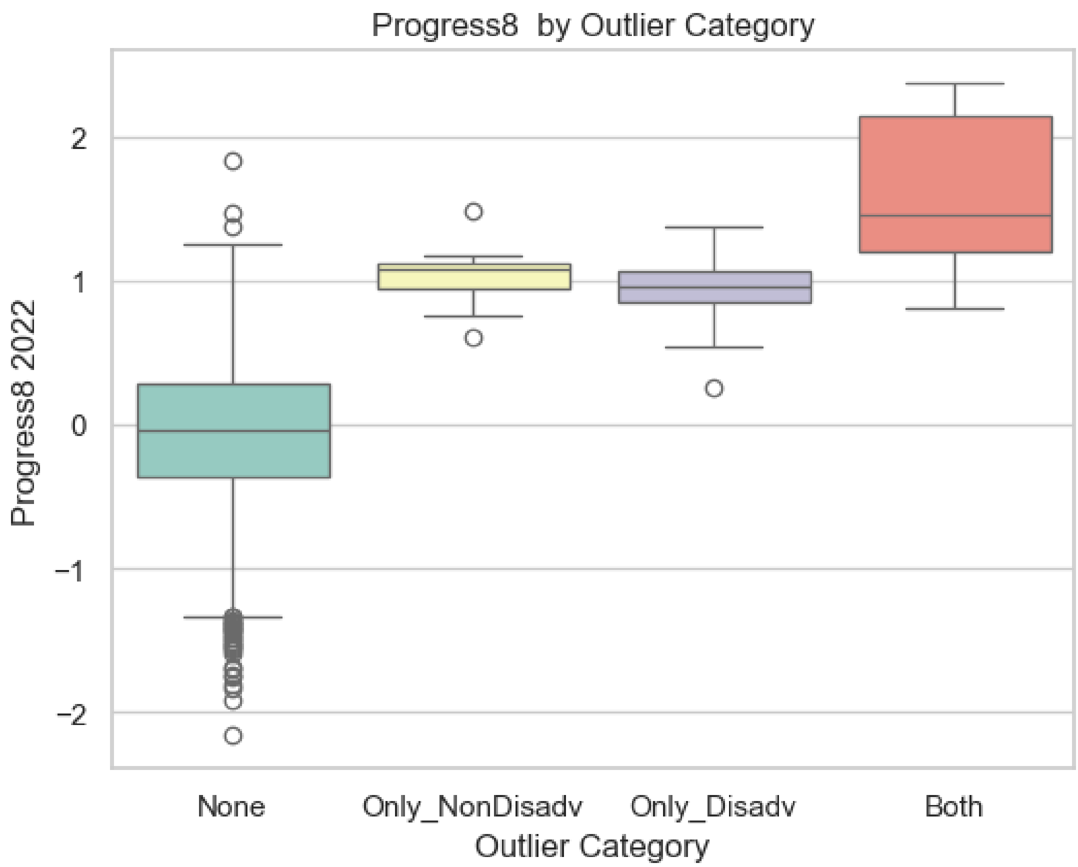
```
print("Summary Statistics for 'Progress8_Disadvantaged_2022' by
'Outlier_Category':")
print(summary)
```

```
Summary Statistics for 'Progress8_Disadvantaged_2022' by
'Outlier_Category':
                  Median  Q1 (25%)  Q3 (75%)    IQR   Min   Max  Range
Outlier_Category
Both                1.40      1.19     1.585  0.395  1.01  1.96   0.95
None               -0.50     -0.87    -0.140  0.730 -2.43  0.99   3.42
Only_Disadv         1.09      1.07     1.125  0.055  1.01  1.21   0.20
Only_NonDisadv      0.60      0.51     0.775  0.265  0.22  0.99   0.77
```

Plot a box plot to show progress 8 scores for all outlier category types

```
sns.boxplot(
    x='Outlier_Category',
    y='Progress8',
    data= merged_df_2,
    palette='Set3'
)
plt.title('Progress8  by Outlier Category')
plt.xlabel('Outlier Category')
plt.ylabel('Progress8 2022')


images_dir = 'images'
image_path = os.path.join(images_dir,'Obj2_Progress8 by Outlier
Category.png')
plt.savefig(image_path)

plt.show()
```

```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\3243744368.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(
```



```
# Calculate summary statistics
grouped = merged_df_2.groupby('Outlier_Category')['Progress8']


median = grouped.median()
q1 = grouped.quantile(0.25)
q3 = grouped.quantile(0.75)
iqr = q3 - q1
minimum = grouped.min()
maximum = grouped.max()
range_ = maximum - minimum


summary = pd.DataFrame({
    'Median': median,
    'Q1 (25%)': q1,
    'Q3 (75%)': q3,
    'IQR': iqr,
    'Min': minimum,
    'Max': maximum,
    'Range': range_
})
```

```
print("Summary Statistics for 'Progress8' by 'Outlier_Category':")
print(summary)
```

```
Summary Statistics for 'Progress8' by 'Outlier_Category':
                  Median  Q1 (25%)  Q3 (75%)     IQR   Min   Max
Range
Outlier_Category

Both               1.450    1.1950     2.140  0.9450  0.81  2.37
1.56
None              -0.040   -0.3600     0.290  0.6500 -2.16  1.83
3.99
Only_Disadv        0.955    0.8500     1.070  0.2200  0.25  1.38
1.13
Only_NonDisadv     1.085    0.9375     1.125  0.1875  0.61  1.49
0.88
```

Using the get_school_details function defined earlier, I can extract schoool details based on a URN list

```
columns = ['School_Name',
 'Trust_Name', 'Percent_Disadvantaged_2022', 'Progress8',
 'Progress8_NonDisadvantaged_2022', 'Progress8_Disadvantaged_2022',
  'Percent_Not_Disadvantaged_2022',
 'Religious_Character',
'Admissions_Policy',
'School_Gender',
'Ofsted_Rating',]

data_loader = DataWrangler(dataframe=outlier_df)
# Schools only in outliers_disadvantaged
schools_only_disdv_outliers =
data_loader.get_school_details(only_disadvp8_outliers, columns)
print("schools_only_disdv_outliers:")
print(schools_only_disdv_outliers.to_string(index=False), "\n")
```

```
DataWrangler initialised with the provided DataFrame.
schools_only_disdv_outliers:

School_Name                                    Trust_Name
Percent_Disadvantaged_2022  Progress8  Progress8_NonDisadvantaged_2022
Progress8_Disadvantaged_2022  Percent_Not_Disadvantaged_2022
Religious_Character Admissions_Policy School_Gender
Ofsted_Rating
                                                        Ashcroft
Technology Academy PROSPECT EDUCATION (TECHNOLOGY) TRUST LIMITED
34        1.09                             1.17
1.09                                 66                     Does not apply
```

```
Non-selective         Mixed          Outstanding
                                                              Wren
Academy Finchley                          WREN ACADEMIES TRUST
16       1.00                             0.76
1.02                              84                Church of England
Non-selective         Mixed          Outstanding
                                                    Lancaster Girls'
Grammar School               LANCASTER GIRLS' GRAMMAR SCHOOL
6       0.54                             0.68
1.09                              94                              NaN
Selective         Girls                 Good

Wilson's School                               WILSON'S SCHOOL
5       1.33                             1.06
1.07                              95                Church of England
Selective          Boys          Outstanding
                                                        Birmingham
Ormiston Academy                   BIRMINGHAM ORMISTON ACADEMY
8       0.25                             0.58
1.10                              92                   Does not apply
Non-selective         Mixed                 Good

Featherstone High School               GRAND UNION MULTI ACADEMY TRUST
31       0.84                             1.15
1.05                              69                   Does not apply
Non-selective         Mixed          Outstanding
                                                          St Mark's
Catholic School      THE DIOCESE OF WESTMINSTER ACADEMY TRUST
10       1.30                             1.15
1.21                              90                   Roman Catholic
Non-selective         Mixed          Outstanding
                                                            Ark
Bolingbroke Academy                                   ARK SCHOOLS
17       0.88                             0.80
1.07                              83                              NaN
Non-selective         Mixed                 Good

Nishkam High School                          NISHKAM SCHOOL TRUST
32       0.94                             1.11
1.01                              68                             Sikh
Non-selective         Mixed          Outstanding
                                                       Sacred Heart
Catholic School                  SACRED HEART CATHOLIC SCHOOL
49       1.38                             1.35
1.21                              51                   Roman Catholic
Non-selective         Mixed          Outstanding
                                                         Eden Boys'
School, Preston                                STAR ACADEMIES
21       0.97                             0.74
1.15                              79                           Muslim
```

```
Non-selective          Boys          Outstanding
                                                         Bolton Muslim
Girls School                   PROSPER MULTI ACADEMY TRUST
19       1.01                            0.98
1.11                              81                             Muslim
Non-selective         Girls Requires improvement
                                                  Dartford Grammar
School for Girls                               THE ARETÉ TRUST
17       0.93                            0.99
1.09                              83                                NaN
Selective         Girls          Outstanding
St Francis Xavier School - a Joint Catholic and Church of England
Voluntary Academy      NICHOLAS POSTGATE CATHOLIC ACADEMY TRUST
8       0.79                            0.79
1.13                              92 Roman Catholic/Church of England
Non-selective         Mixed                 NaN
```

```
# Schools only in outliers_not disadvantaged
data_loader = DataWrangler(dataframe=outlier_df)
schools_only_nondisadv_outliers =
data_loader.get_school_details(only_nondisadvp8_outliers, columns)
print("schools_only_nondisadv_outliers:")
print(schools_only_nondisadv_outliers.to_string(index=False), "\n")
```

```
DataWrangler initialised with the provided DataFrame.
schools_only_nondisadv_outliers:
                           School_Name
Trust_Name  Percent_Disadvantaged_2022  Progress8
Progress8_NonDisadvantaged_2022  Progress8_Disadvantaged_2022
Percent_Not_Disadvantaged_2022 Religious_Character Admissions_Policy
School_Gender Ofsted_Rating
                   Northampton Academy
UNITED LEARNING TRUST                          31       1.00
1.42                          0.22                              69
Christian     Non-selective         Mixed   Outstanding
              Ark King Solomon Academy
ARK SCHOOLS                          49       1.09
1.55                          0.91                              51
NaN     Non-selective         Mixed   Outstanding
Hillcrest School and Sixth Form Centre
HILLCREST SCHOOL AND SIXTH FORM CENTRE                          63
0.61                             1.68                          0.42
37      Does not apply     Non-selective         Girls          Good
              Avonbourne Girls Academy AVONBOURNE INTERNATIONAL
BUSINESS AND ENTERPRISE ACADEMY TRUST                          28
0.75                             1.42                          0.54
72                 NaN     Non-selective         Mixed          Good
```

```
                      Glenmoor Academy
UNITED LEARNING TRUST                          22       1.11
1.42                          0.73                              78
NaN     Non-selective         Girls   Outstanding
           Eden Girls' School Coventry
STAR ACADEMIES                          41       1.49
1.41                          0.99                              59
Muslim               NaN         Girls   Outstanding
               Levenshulme High School
EDUCATION AND LEADERSHIP TRUST                          44       1.08
1.59                          0.61                              56
Does not apply    Not applicable         Girls   Outstanding
            Eden Girls' School, Slough
STAR ACADEMIES                          31       1.17
1.45                          0.59                              69
Muslim     Non-selective         Girls   Outstanding
```

```
# Schools in both outliers_disadvantaged only and outliers_not
disadvantaged

data_loader = DataWrangler(dataframe=outlier_df)

schools_both = data_loader.get_school_details(both_p8_outliers,
columns)
print("Schools in both disadv and nondisadv outliers:")
print(schools_both.to_string(index=False))
```

```
DataWrangler initialised with the provided DataFrame.
Schools in both disadv and nondisadv outliers:
                                School_Name
Trust_Name  Percent_Disadvantaged_2022  Progress8
Progress8_NonDisadvantaged_2022  Progress8_Disadvantaged_2022
Percent_Not_Disadvantaged_2022 Religious_Character Admissions_Policy
School_Gender Ofsted_Rating
                   Bentley Wood High School                    THE
BENTLEY WOOD TRUST                          28       1.05
1.48                          1.01                              72
Does not apply     Non-selective         Girls   Outstanding
                  Michaela Community School          MICHAELA
COMMUNITY SCHOOLS TRUST                          26       2.37
2.37                          1.96                              74
NaN     Non-selective         Mixed   Outstanding
         Tauheedul Islam Girls' High School
STAR ACADEMIES                          17       2.30
1.76                          1.56                              83
Muslim     Non-selective         Girls   Outstanding
                     The Hurlingham Academy                     UNITED
LEARNING TRUST                          38       0.81
```

```
1.42                          1.19                              62
Does not apply     Non-selective        Mixed   Outstanding
                  Ealing Fields High School TWYFORD CHURCH OF ENGLAND
ACADEMIES TRUST                          37       1.34
1.41                          1.19                              63
NaN               NaN         Mixed          Good
Eden Girls'  Leadership Academy, Birmingham
STAR ACADEMIES                          45       1.98
1.84                          1.40                              55
Muslim     Non-selective        Girls   Outstanding
                 St Peter's Catholic School           XAVIER CATHOLIC
EDUCATION TRUST                           5       1.45
1.37                          1.61                              95
Roman Catholic     Non-selective        Mixed   Outstanding
```

I will also evaluate the categorical columns in the outlier schools

```
categorical_columns= outlier_df[['School_Type','School_College_Type',
'Religious_Character',
'Admissions_Policy',
 'School_Gender',
'Ofsted_Rating','Trust_Name','Outlier_Category']]


numerical_variables = outlier_df[['FSM_Funding',

'Pupil_Premium_Funding',

'Pupil_Premium_Pupils',

'School_Led_Tutoring_Funding',

'Total_Funding','Attainment8',

'Progress8',

'Percent_Disadvantaged_2022',

'Percent_Not_Disadvantaged_2022',

'Percent_Disadvantaged_Strong_Passes',

'Percent_Not_Disadvantaged_Strong_Passes',
       'Average Attainment 8 score per non-disadvantaged pupil  -
2022',

'Progress8_NonDisadvantaged_2022',
                                   'Attainment8_Disadvantaged_2022',
'Progress8_Disadvantaged_2022',

'Progress8_Maths_Disadvantaged',

'Progress8_English_Disadvantaged',

'Progress8_Maths_NonDisadvantaged',

'Progress8_English_NonDisadvantaged', 'Index of Multiple Deprivation
Decile']]

numerical_variables.describe()

         FSM_Funding  Pupil_Premium_Funding  Pupil_Premium_Pupils  \
count      29.000000              29.000000             29.000000
mean   105475.413793          221438.344828            219.586207
std    138064.982769          143085.321561            131.774027
min         0.000000           39400.000000             40.000000
25%     37515.000000          129035.000000            131.000000
50%     74260.000000          183210.000000            186.000000
75%    121260.000000          279740.000000            284.000000
max    730864.000000          681170.000000            562.000000

       School_Led_Tutoring_Funding  Total_Funding  Attainment8
Progress8  \
count                    29.000000   2.900000e+01    29.000000
29.000000
mean                  36555.517241   5.670275e+06    61.465517
1.132759
std                   21375.221128   2.015004e+06     8.462956
0.466835
min                    6480.000000   2.223628e+06    50.500000
0.250000
25%                   22194.000000   4.234625e+06    56.200000
0.880000
50%                   31590.000000   5.358210e+06    59.900000
1.050000
75%                   46818.000000   6.200771e+06    64.900000
1.330000
max                   86994.000000   1.047014e+07    86.400000
2.370000

       Percent_Disadvantaged_2022  Percent_Not_Disadvantaged_2022  \
count                   29.000000                       29.000000
mean                    26.827586                       73.172414
std                     15.099832                       15.099832
min                      5.000000                       37.000000
25%                     17.000000                       63.000000
50%                     28.000000                       72.000000
75%                     37.000000                       83.000000
```

```
max                     63.000000                       95.000000

       Percent_Disadvantaged_Strong_Passes  \
count                            29.000000
mean                             63.689655
std                              17.730328
min                              29.000000
25%                              53.000000
50%                              62.000000
75%                              71.000000
max                             100.000000

       Percent_Not_Disadvantaged_Strong_Passes  \
count                                29.000000
mean                                 76.689655
std                                  12.742234
min                                  56.000000
25%                                  65.000000
50%                                  77.000000
75%                                  84.000000
max                                 100.000000

       Average Attainment 8 score per non-disadvantaged pupil  - 2022
\
count                                          29.000000

mean                                           65.731034

std                                             7.008368

min                                            54.100000

25%                                            62.900000

50%                                            65.300000

75%                                            67.300000

max                                            85.500000


       Progress8_NonDisadvantaged_2022  Attainment8_Disadvantaged_2022
\
count                        29.000000                       29.000000

mean                          1.272414                       59.410345

std                           0.397911                        8.321502

min                           0.580000                       45.700000
```

```
25%                                  0.990000                        55.000000

50%                                  1.370000                        57.800000

75%                                  1.450000                        62.100000

max                                  2.370000                        84.100000


       Progress8_Disadvantaged_2022  Progress8_Maths_Disadvantaged  \
count                     29.000000                      29.000000
mean                       1.045862                       0.749655
std                        0.354718                       0.664543
min                        0.220000                      -0.680000
25%                        0.990000                       0.360000
50%                        1.090000                       0.830000
75%                        1.190000                       0.970000
max                        1.960000                       2.480000

       Progress8_English_Disadvantaged
Progress8_Maths_NonDisadvantaged  \
count                        29.000000
29.000000
mean                          0.995172
1.103103
std                           0.570705
0.578225
min                          -0.280000
0.020000
25%                           0.660000
0.720000
50%                           0.890000
1.050000
75%                           1.270000
1.400000
max                           2.190000
2.950000

       Progress8_English_NonDisadvantaged  \
count                           29.000000
mean                             1.256207
std                              0.461793
min                              0.610000
25%                              0.970000
50%                              1.150000
75%                              1.480000
max                              2.330000

       Index of Multiple Deprivation Decile
count                             29.000000
```

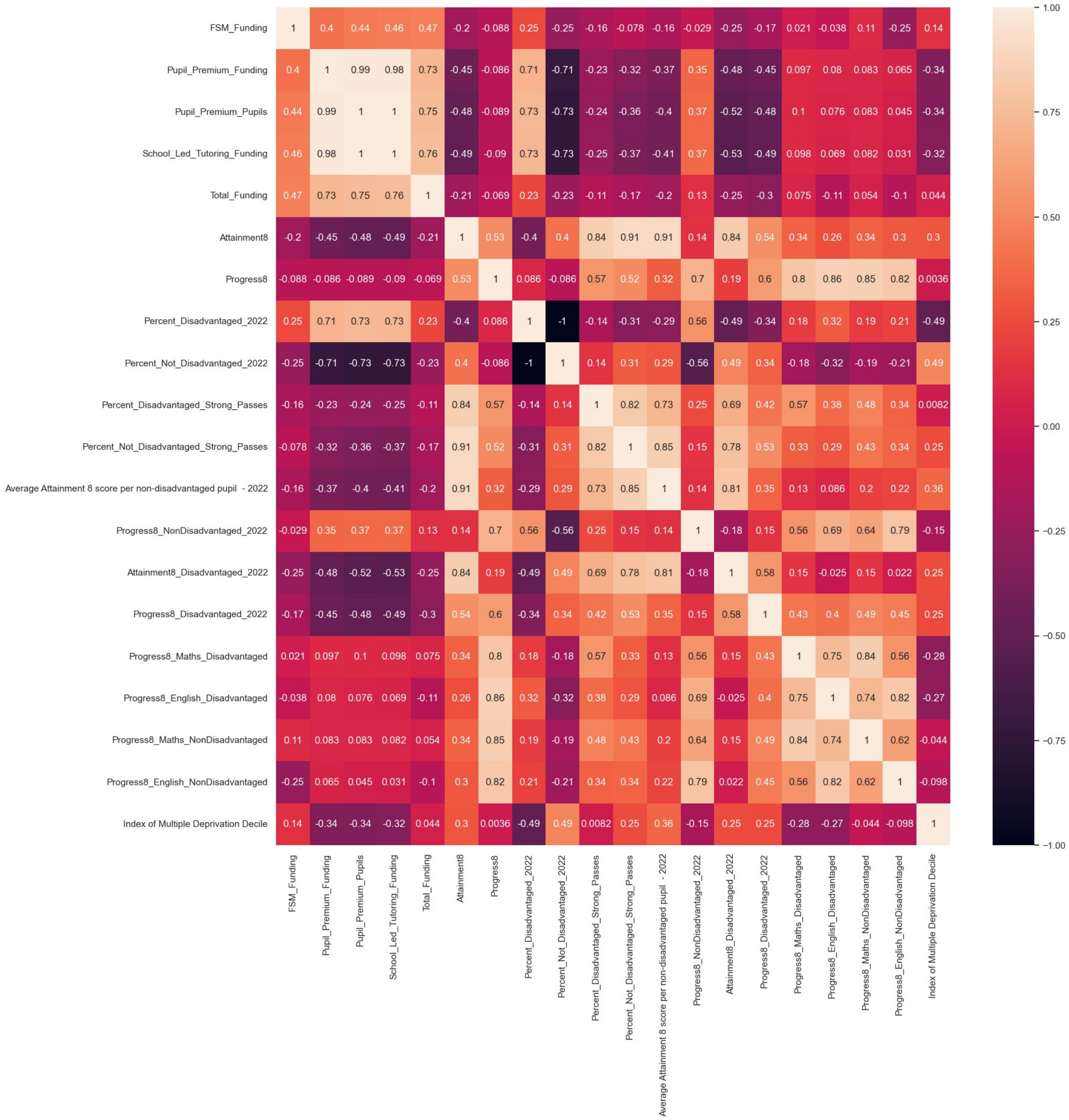
```
mean                               4.896552
std                                2.730055
min                                1.000000
25%                                3.000000
50%                                4.000000
75%                                7.000000
max                               10.000000


#Standardise the numerical values to ensure accurate corrlation

scaler = StandardScaler() # use standard scaler which will make each
feature have 0 mean and SD=1

scaled_data = scaler.fit_transform(numerical_variables)


scaled_numerical_df = pd.DataFrame(scaled_data,
columns=numerical_variables.columns)


merged_df_2[scaled_numerical_df.columns] = scaled_numerical_df


scaled_numerical_df.head()

   FSM_Funding  Pupil_Premium_Funding  Pupil_Premium_Pupils  \
0     0.341747               1.458545              1.648210   
1     0.199734               3.269861              2.644487   
2     4.609846               0.901583              1.038088   
3    -0.213244               0.050084             -0.089481   
4    -0.652757              -1.273738             -1.363788   

   School_Led_Tutoring_Funding  Total_Funding  Attainment8  Progress8
\
0                     1.768970       2.044712    -1.005983  -0.289414

1                     2.401438       1.378092    -0.392690  -0.093214

2                     1.136931       1.618857     0.220603  -0.093214

3                    -0.105293       1.051198     0.220603  -0.289414

4                    -1.401080      -0.725091     1.218708  -1.292214


   Percent_Disadvantaged_2022  Percent_Not_Disadvantaged_2022  \
0                    0.281213                       -0.281213   
1                    1.494379                       -1.494379   
2                    0.483407                       -0.483407   
3                   -0.729759                        0.729759   
```

```
4                   -1.403740                        1.403740

   Percent_Disadvantaged_Strong_Passes  \
0                            -1.072765   
1                            -0.326580   
2                             0.304808   
3                             0.419605   
4                             1.108392   

   Percent_Not_Disadvantaged_Strong_Passes  \
0                                -0.933633   
1                                 0.184523   
2                                 0.583865   
3                                 0.503996   
4                                 1.462416   

   Average Attainment 8 score per non-disadvantaged pupil  - 2022  \
0                                          -0.977428   
1                                          -0.178761   
2                                           0.184269   
3                                           0.082621   
4                                           1.679955   

   Progress8_NonDisadvantaged_2022  Attainment8_Disadvantaged_2022  \
0                         0.377468                       -1.676744   
1                         0.709957                       -0.258090   
2                        -0.261934                       -0.025725   
3                        -1.310553                        0.328939   
4                        -1.515162                        1.808741   

   Progress8_Disadvantaged_2022  Progress8_Maths_Disadvantaged  \
0                     -2.369430                       0.092414   
1                     -0.389793                       0.245557   
2                      0.126633                       0.414014   
3                     -0.074199                      -0.336386   
4                      0.126633                      -1.699358   

   Progress8_English_Disadvantaged  
Progress8_Maths_NonDisadvantaged  \
0                        -0.330206                         -0.163866   

1                         0.490082                          0.205743   

2                        -0.098385                          0.575352   

3                        -0.740349                          0.205743   

4                        -2.273930                         -1.431098   
```

```
   Progress8_English_NonDisadvantaged  Index of Multiple Deprivation
Decile
0                            -0.564629                              -
1.079766
1                             0.911918                              -
1.079766
2                            -1.093541
0.784116
3                            -0.718895
0.784116
4                            -1.424112
0.038563
```

As part of the analysis, I will create a heatmap of numerical variables

```
# Correlation matrix
corr_matrix = numerical_variables.corr()
plt.figure(figsize=(18,18))

images_dir = 'images'
image_path = os.path.join(images_dir,'Obj2_Heatmap Outlier
Schools.png')
plt.savefig(image_path)


# Heatmap
sns.heatmap(corr_matrix, annot=True)
plt.show()
```



Objective 3 Code: Identify and evaluate the top performing multi-academy trusts in supporting disadvantaged pupils

```
merged_df_3=merged_df.copy() # make another copy of merged_df

# group schools by MAT for analysis
MAT_performance = merged_df_3.groupby('Trust_Name')
[['Progress8','Progress8_Disadvantaged_2022','progress8_gap',

'attainment8_gap',
'maths_gap',

'english_gap',

'5_GCSE_gap',

'pupilpremium_per_pupil','Progress8_NonDisadvantaged_2022','Index of
Multiple Deprivation Decile']].mean().reset_index()
MAT_performance.head()
```

```
                             Trust_Name  Progress8  \
0                   5 DIMENSIONS TRUST      -0.13
1                ABBEY ACADEMIES TRUST      -0.43
2                ABBEY COLLEGE, RAMSEY      -0.10
3            ABBEY MULTI ACADEMY TRUST       0.07
4  ABBS CROSS ACADEMY AND ARTS COLLEGE       0.04

   Progress8_Disadvantaged_2022  progress8_gap  attainment8_gap
maths_gap  \
0                         -0.89          0.930            14.40
0.43
1                         -1.11          0.930            10.50
0.59
2                         -0.55          0.690            14.60
0.25
3                         -0.05          0.305             7.75
0.41
4                         -1.17          0.990            16.00
0.35

   english_gap  5_GCSE_gap  pupilpremium_per_pupil  \
0         0.93        18.0              983.164179
1         0.48        26.0              985.000000
2         0.26        14.0              985.000000
3         0.44        17.0              984.078652
4         0.31        17.0              985.000000

   Progress8_NonDisadvantaged_2022  Index of Multiple Deprivation
Decile
0                            0.040
6.0
1                           -0.180
7.0
2                            0.140
6.0
3                            0.255
1.0
4                           -0.180
8.0
```

```
MAT_performance_sorted =
MAT_performance.sort_values(by='Progress8_Disadvantaged_2022',
ascending=False)
#we can now sort by progress 8 score of disadvantaged pupils
MAT_performance_sorted.head()


                           Trust_Name  Progress8  \
654  MICHAELA COMMUNITY SCHOOLS TRUST       2.37
833      SACRED HEART CATHOLIC SCHOOL       1.38
780       PROSPER MULTI ACADEMY TRUST       1.01
111       BIRMINGHAM ORMISTON ACADEMY       0.25
587   LANCASTER GIRLS' GRAMMAR SCHOOL       0.54

     Progress8_Disadvantaged_2022  progress8_gap  attainment8_gap
maths_gap  \
654                          1.96           0.41              6.3
0.47
833                          1.21           0.14              3.2
0.49
780                          1.11          -0.13              7.6
-0.36
111                          1.10          -0.52             -2.9
0.70
587                          1.09          -0.41              3.1
0.65

     english_gap  5_GCSE_gap  pupilpremium_per_pupil  \
654         0.25         5.0                   985.0
833         0.15        15.0                   985.0
780         0.59        15.0                   985.0
111         0.12        36.0                   985.0
587         0.89        12.0                   985.0

     Progress8_NonDisadvantaged_2022  Index of Multiple Deprivation
Decile
654                             2.37
5.0
833                             1.35
4.0
780                             0.98
3.0
111                             0.58
4.0
587                             0.68
5.0

# Group by Trust and calculate mean scores along with the count of
schools
MAT_performance = merged_df_3.groupby('Trust_Name').agg(
    avg_progress8_score=('Progress8', 'mean'),
    prog8_score_disadv=('Progress8_Disadvantaged_2022', 'mean'),
    prog8_score_nondisadv=('Progress8_NonDisadvantaged_2022', 'mean'),
    progress8_gap=('progress8_gap', 'mean'),
    attainment8_gap=('attainment8_gap', 'mean'),
    maths_gap=('maths_gap', 'mean'),
    english_gap=('english_gap', 'mean'),
    FiveGCSE_gap=('5_GCSE_gap', 'mean'),
 deprivation_index= ('Index of Multiple Deprivation Decile','mean'),
    school_count=('URN', 'count')  # Counting the number of schools 
per Group Name
).reset_index()

# Sort the MAT_performance DataFrame by 'avg_progress8_score' in 
descending order
MAT_performance_sorted = 
MAT_performance.sort_values(by='prog8_score_disadv', ascending=False)

MAT_performance_sorted.head()
```

```
                           Trust_Name  avg_progress8_score  \
654  MICHAELA COMMUNITY SCHOOLS TRUST                 2.37   
833      SACRED HEART CATHOLIC SCHOOL                 1.38   
780       PROSPER MULTI ACADEMY TRUST                 1.01   
111       BIRMINGHAM ORMISTON ACADEMY                 0.25   
587   LANCASTER GIRLS' GRAMMAR SCHOOL                 0.54   

     prog8_score_disadv  prog8_score_nondisadv  progress8_gap  \
654                1.96                   2.37           0.41   
833                1.21                   1.35           0.14   
780                1.11                   0.98          -0.13   
111                1.10                   0.58          -0.52   
587                1.09                   0.68          -0.41   

     attainment8_gap  maths_gap  english_gap  FiveGCSE_gap  
deprivation_index  \
654              6.3       0.47         0.25           5.0   
5.0   
833              3.2       0.49         0.15          15.0   
4.0   
780              7.6      -0.36         0.59          15.0   
3.0   
111             -2.9       0.70         0.12          36.0   
4.0   
587              3.1       0.65         0.89          12.0   
5.0   

     school_count  
654             1  
833             1  
```

```
780             1
111             1
587             1
```

A number of MATs have 1 or 2 schools, so I will filter for those with at least 4 schools as I want to explore organisational impact of Trusts working with mutiple schools

```
# Filter MATs with school_count >= 4
MAT_performance_filtered = 
MAT_performance[MAT_performance['school_count'] >= 4]

# top 10 MATs with the highest average Progress 8 scores disadvantaged 
and at least 4 schools
MAT_performance_sorted = 
MAT_performance_filtered.sort_values(by='prog8_score_disadv', 
ascending=False)


Top_10MAT = MAT_performance_sorted.head(10)
print(Top_10MAT)
```

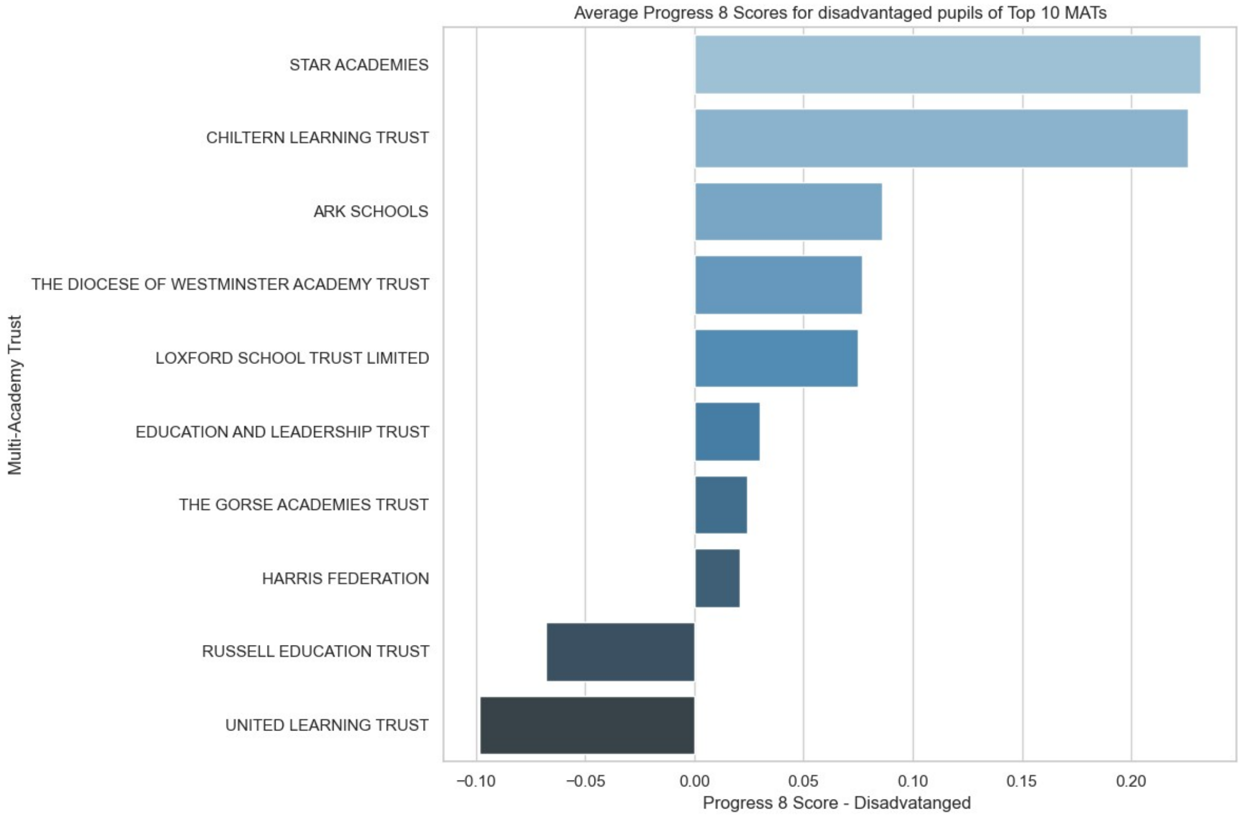
```
                                    Trust_Name  avg_progress8_score  \
975                             STAR ACADEMIES             0.640526   
219                    CHILTERN LEARNING TRUST             0.424000   
57                                 ARK SCHOOLS             0.208421   
1096  THE DIOCESE OF WESTMINSTER ACADEMY TRUST             0.645000   
628               LOXFORD SCHOOL TRUST LIMITED             0.337500   
335             EDUCATION AND LEADERSHIP TRUST             0.260000   
1123                 THE GORSE ACADEMIES TRUST             0.435714   
451                          HARRIS FEDERATION             0.265652   
827                    RUSSELL EDUCATION TRUST             0.466000   
1323                     UNITED LEARNING TRUST             0.146757   

      prog8_score_disadv  prog8_score_nondisadv  progress8_gap  \
975             0.231579               0.495789       0.264211   
219             0.226000               0.634000       0.408000   
57              0.085789               0.435789       0.350000   
1096            0.076667               0.691667       0.615000   
628             0.075000               0.422500       0.347500   
335             0.030000               0.762500       0.732500   
1123            0.024286               0.645714       0.621429   
451             0.020870               0.644348       0.623478   
827            -0.068000               0.534000       0.602000   
1323           -0.098378               0.487297       0.585676   

      attainment8_gap  maths_gap  english_gap  FiveGCSE_gap  \
975          6.247368   0.294737     0.244211     10.210526   
219          7.680000   0.456000     0.348000     17.000000   
57           6.131579   0.437368     0.290526     16.263158   
```

```
1096         10.700000   0.303333     0.161667     20.833333
628           5.075000   0.312500     0.147500     15.250000
335          10.450000   0.575000     0.527500     18.500000
1123         12.842857   0.675714     0.810000     25.571429
451          10.065217   0.574783     0.400435     19.826087
827          15.380000   0.730000     0.836000     34.000000
1323          9.802703   0.518108     0.507027     18.297297

      deprivation_index  school_count
975            2.421053            19
219            4.600000             5
57             3.157895            19
1096           5.833333             6
628            5.000000             4
335            3.000000             4
1123           5.714286             7
451            4.956522            23
827            5.200000             5
1323           4.297297            37
```

```
#represent data on a graphs

# Set the style
sns.set(style="whitegrid")

# Pconfigure the barplot
plt.figure(figsize=(12, 8))
sns.barplot(
    x='prog8_score_disadv',
    y='Trust_Name',
    data=MAT_performance_sorted.head(10),
    palette='Blues_d'
)
plt.title('Average Progress 8 Scores for disadvantaged pupils of Top 
10 MATs')
plt.xlabel('Progress 8 Score - Disadvatanged ')
plt.ylabel('Multi-Academy Trust')
plt.tight_layout()


#save image in data folder
images_dir = 'images'
image_path = os.path.join(images_dir,'Obj3_Progress8 disadvatanged top 
10 MATs.png')
plt.savefig(image_path)

plt.show()
```

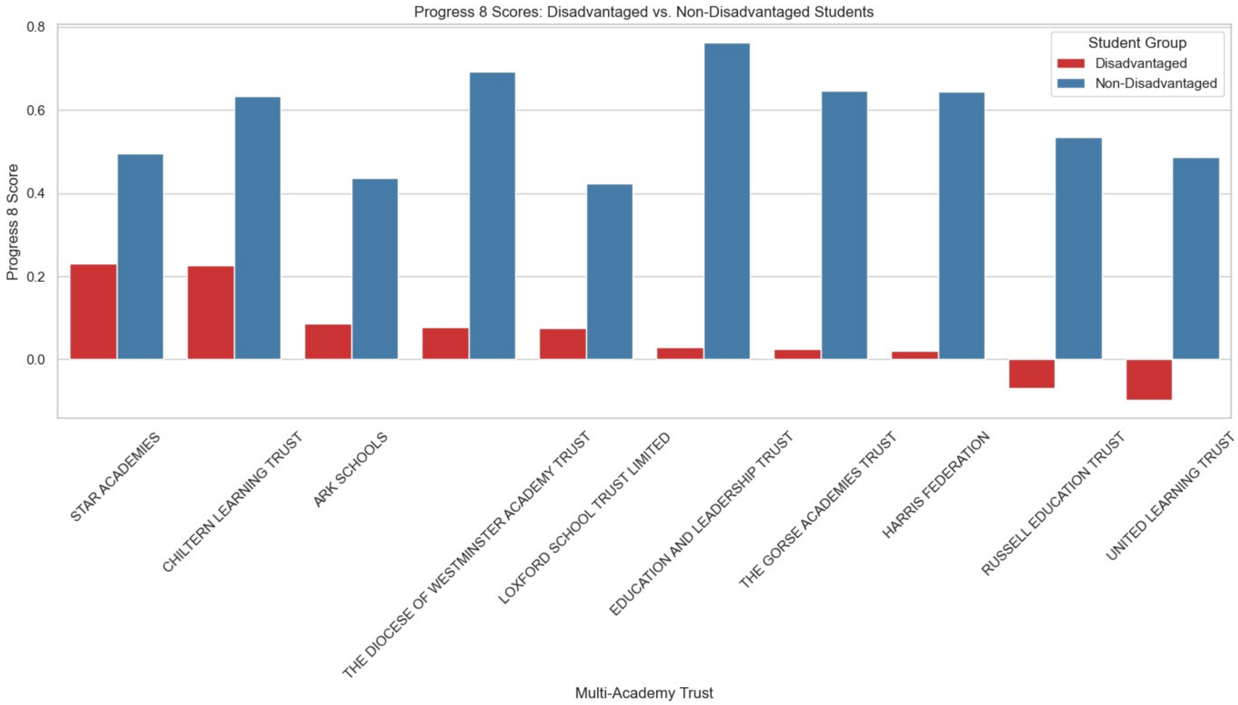
```
C:\Users\saqib\AppData\Local\Temp\ipykernel_34844\2443691115.py:8: 
FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(
```



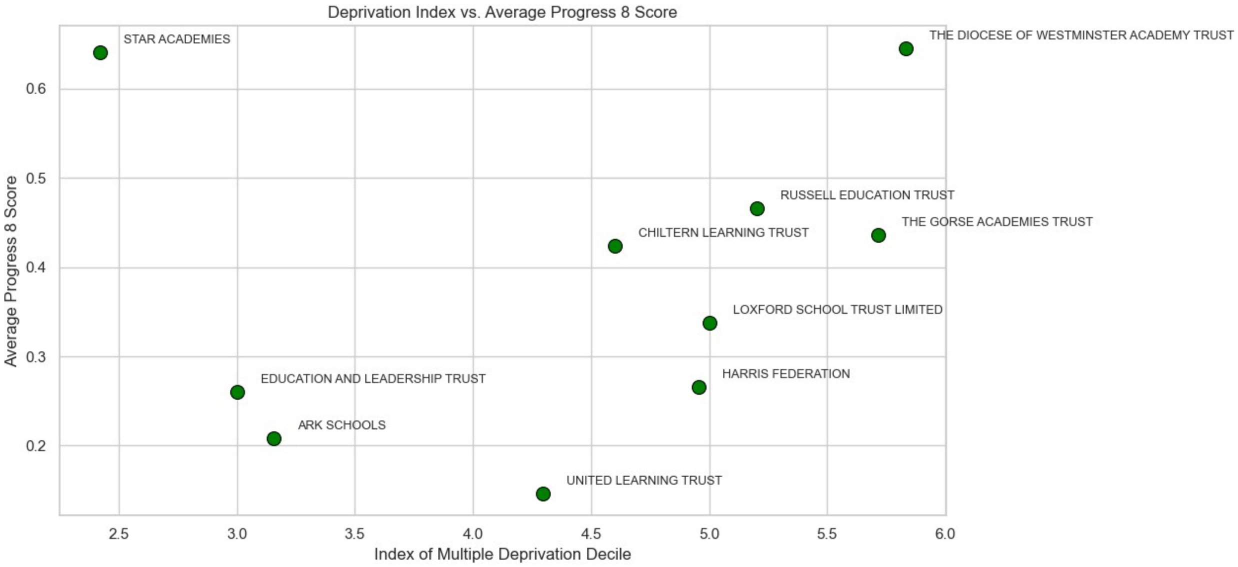
```python
# Melt the DataFrame for easier plotting
top_10_MATs= MAT_performance_sorted.head(10)
prog8_melted = top_10_MATs.melt(
    id_vars='Trust_Name',
    value_vars=['prog8_score_disadv', 'prog8_score_nondisadv'],
    var_name='Group',
    value_name='Progress8_Score'
)

# Replace group names for clarity
prog8_melted['Group'] = prog8_melted['Group'].map({
    'prog8_score_disadv': 'Disadvantaged',
    'prog8_score_nondisadv': 'Non-Disadvantaged'
})

# Plot
plt.figure(figsize=(14, 8))
```

```
sns.barplot(
    x='Trust_Name',
    y='Progress8_Score',
    hue='Group',
    data=prog8_melted,
    palette='Set1'
)
plt.title('Progress 8 Scores: Disadvantaged vs. Non-Disadvantaged Students')
plt.xlabel('Multi-Academy Trust')
plt.ylabel('Progress 8 Score')
plt.legend(title='Student Group')
plt.xticks(rotation=45)
plt.tight_layout()

#save the image in data folder
images_dir = 'images'
image_path = os.path.join(images_dir,'Obj3_Progress8 disadv vs advantaged in top 10 MATs.png')
plt.savefig(image_path)

plt.show()
```



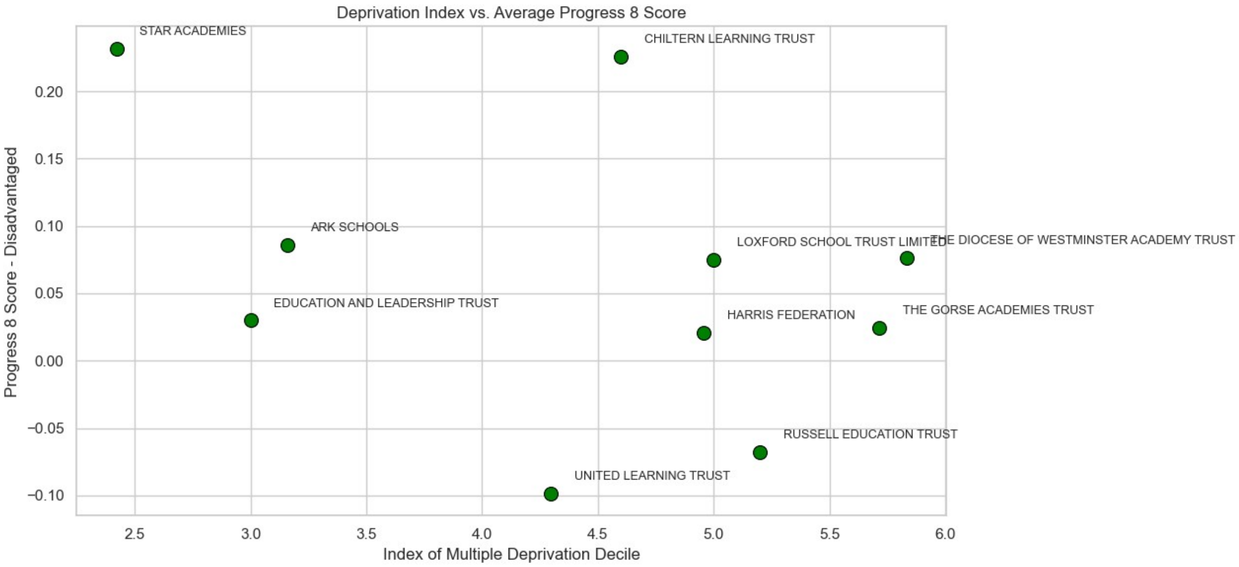
Plot a scatterplot of MATs and average progress 8

```
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x='deprivation_index',
    y='avg_progress8_score',
    data=top_10_MATs,
    s=100,
    color='green',
    edgecolor='black'
)
plt.title('Deprivation Index vs. Average Progress 8 Score')
plt.xlabel('Index of Multiple Deprivation Decile')
plt.ylabel('Average Progress 8 Score')
plt.tight_layout()


# Annotate MAT names
for idx, row in top_10_MATs.iterrows():
    plt.text(row['deprivation_index']+0.1, row['avg_progress8_score']
+0.01,
             row['Trust_Name'], fontsize=9)

images_dir = 'images'
image_path = os.path.join(images_dir,'Obj3_Deprivation Index vs P8 top
10 MATS.png')
plt.savefig(image_path)

plt.show()
```



Plot a scatter plor of MATs and progres 8 disadvantages

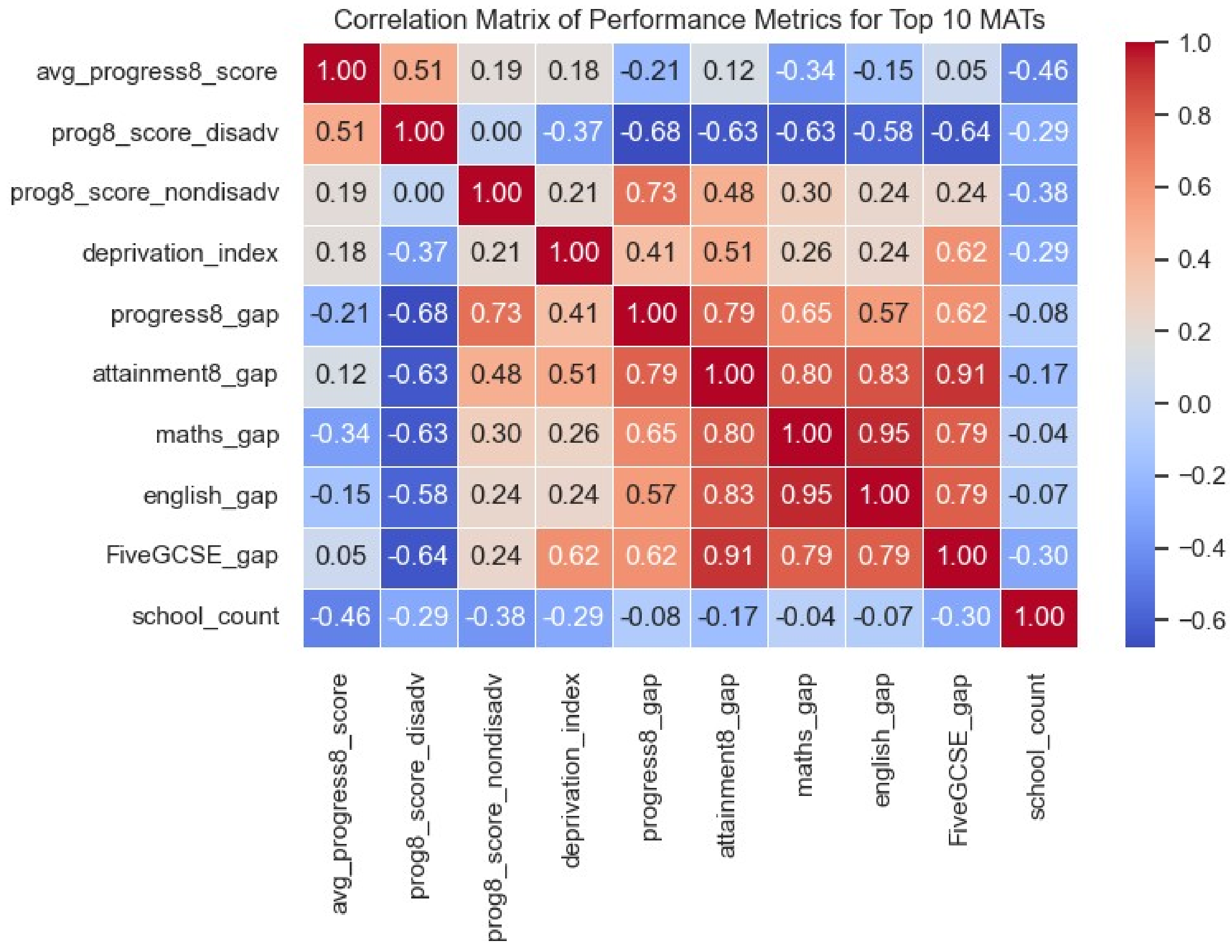
```
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x='deprivation_index',
    y='prog8_score_disadv',
    data=top_10_MATs,
    s=100,
    color='green',
    edgecolor='black'
)
plt.title('Deprivation Index vs. Average Progress 8 Score')
plt.xlabel('Index of Multiple Deprivation Decile')
plt.ylabel('Progress 8 Score - Disadvantaged')
plt.tight_layout()

# Annotate MAT names
for idx, row in top_10_MATs.iterrows():
    plt.text(row['deprivation_index']+0.1, row['prog8_score_disadv']
+0.01,
             row['Trust_Name'], fontsize=9)


images_dir = 'images'
image_path = os.path.join(images_dir,'Obj3_Progress8 Disadv vs
Deprivation Index for Top 10 MATs.png')
plt.savefig(image_path)

plt.show()
```



Before we can analyse the correlation coefficients I would need to standardise the data

```
# Columns to standardise
corr_columns = ['avg_progress8_score', 'prog8_score_disadv',
                'prog8_score_nondisadv', 'deprivation_index',
'progress8_gap',
                'attainment8_gap', 'maths_gap', 'english_gap',
'FiveGCSE_gap',
                'school_count']

scaler = StandardScaler() # this will give a mean of 0 and SD of 1

#fiter data
top_10_MATs_standardized = top_10_MATs.copy()
top_10_MATs_standardized[corr_columns] =
scaler.fit_transform(top_10_MATs[corr_columns])


print(top_10_MATs_standardized.head())

                                    Trust_Name  avg_progress8_score  \
975                             STAR ACADEMIES             1.586519
219                    CHILTERN LEARNING TRUST             0.252807
57                                 ARK SCHOOLS            -1.075069
1096  THE DIOCESE OF WESTMINSTER ACADEMY TRUST             1.614075
628               LOXFORD SCHOOL TRUST LIMITED            -0.279996

      prog8_score_disadv  prog8_score_nondisadv  progress8_gap  \
975             1.684115              -0.725355      -1.680816
219             1.629234               0.534547      -0.717045
57              0.249948              -1.272304      -1.105799
1096            0.160204               1.060226       0.670404
628             0.143809              -1.393449      -1.122556

      attainment8_gap  maths_gap  english_gap  FiveGCSE_gap  \
975         -1.044491  -1.314392    -0.789312     -1.534426
219         -0.575425  -0.216238    -0.341835     -0.421951
57          -1.082402  -0.343113    -0.589626     -0.542685
1096         0.413372  -1.255853    -1.145190      0.206152
628         -1.428343  -1.193430    -1.206268     -0.708694

      deprivation_index  school_count
975           -1.782039      0.575651
219            0.162376     -0.745515
57            -1.124507      0.575651
1096           1.262959     -0.651146
628            0.519322     -0.839884

# Selecting rthe needed columns for correlation
corr_columns = ['avg_progress8_score', 'prog8_score_disadv',
                'prog8_score_nondisadv', 'deprivation_index',
'progress8_gap', 'attainment8_gap',
       'maths_gap', 'english_gap', 'FiveGCSE_gap',
                'school_count']
```

```
# Compute the correlation matrix
corr_matrix = top_10_MATs[corr_columns].corr()

# Plot the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=0.5)
plt.title('Correlation Matrix of Performance Metrics for Top 10 MATs')
plt.tight_layout()

#save the file in the data folder
images_dir = 'images'
image_path = os.path.join(images_dir,'Obj3_Correlation Matrix top 10
MATS.png')
plt.savefig(image_path)

plt.show()
```



	avg_progress8_score	prog8_score_disadv	prog8_score_nondisadv	deprivation_index	progress8_gap	attainment8_gap	maths_gap	english_gap	FiveGCSE_gap	school_count
avg_progress8_score	1.00	0.51	0.19	0.18	-0.21	0.12	-0.34	-0.15	0.05	-0.46
prog8_score_disadv	0.51	1.00	0.00	-0.37	-0.68	-0.63	-0.63	-0.58	-0.64	-0.29
prog8_score_nondisadv	0.19	0.00	1.00	0.21	0.73	0.48	0.30	0.24	0.24	-0.38
deprivation_index	0.18	-0.37	0.21	1.00	0.41	0.51	0.26	0.24	0.62	-0.29
progress8_gap	-0.21	-0.68	0.73	0.41	1.00	0.79	0.65	0.57	0.62	-0.08
attainment8_gap	0.12	-0.63	0.48	0.51	0.79	1.00	0.80	0.83	0.91	-0.17
maths_gap	-0.34	-0.63	0.30	0.26	0.65	0.80	1.00	0.95	0.79	-0.04
english_gap	-0.15	-0.58	0.24	0.24	0.57	0.83	0.95	1.00	0.79	-0.07
FiveGCSE_gap	0.05	-0.64	0.24	0.62	0.62	0.91	0.79	0.79	1.00	-0.30
school_count	-0.46	-0.29	-0.38	-0.29	-0.08	-0.17	-0.04	-0.07	-0.30	1.00

Project Outcome

Overview of Results

Objective 1: Evaluate National Disparities in Educational Performance

There significant gaps between non-disadvantaged and disadvantaged pupils including attainment 8, progress 8, Maths, English and strong passes in both. Disadvantaged pupils lag behind by approximately 1.45 GCSE grades per subject and have an attainment 8 gap of 11.6 points. Their Progress 8 scores are 0.6 grades lower across subjects than their peers, suggesting significant performance gaps.

Objective 2: Identify and Analyse Outlier Schools in Positive Progress 8 of Disadvantaged Pupils

Schools excelling in progress 8 for disadvantaged students, tend to support all students very well and have a strong positive correlation (0.85) between overall and disadvantaged pupils. Funding has a negative correlation with Progress 8 scores for disadvantaged pupils, and could be investigated further.

Objective 3: Identify and Evaluate Top Performing Multi-Academy Trusts (MATs)

High performing MAT have shown a strong positive correlation (0.51) between progress 8 scores for disadvantaged students and overall scores. Although socio-economic factors negatively correlate (-0.37) with progress, for high performing MATs this hasn't been seen to be a barrier; Star Academies for example is one of the highest performing MATs in the country, yet faces the highest deprivation average of all MATs, suggesting a robust pedagogical strategy and governance to run its schools. Such high performing MATs are good at closing the gap (smallest is 0.264 progress 8) between disadvantaged and advantaged students, demonstrating efficient use of funding and better equity.

Objective 1: Evaluate National Disparities in Educational Performance Between Advantaged and Disadvantaged Pupils

Explaination of Results:

There are positive gaps in all categories measured between advantaged and disadvantaged pupils, confirming that nationally, isadvanteged pupils are behind in every academic measure.

Attainment 8 Gap:

- Attainment 8 Disadvantaged: 40.22
- Attainment 8 Non-Disadvantaged: 51.83
- Attainment 8 Gap: 11.61

Analysis: The attainment 8 gap of 11.6 points between disadvantaged and advantaged pupils nationally, suggest approximately 1.45 GCSE grades lower per subject for disadvantaged stidents (11.61/8 = 1.45125 - as each subject is given a point based on the GCSE grade e.g. grade 9 = 9 points).

Progress 8 Gap:

- Progress 8 Disadvantaged: -0.47
- Progress 8 Non-Disadvantaged: 0.13
- Progress 8 Gap: 0.60

Analysis: Progress 8 gap of 0.60 that disadvantaged pupils are making 0.6 grades less progress across 8 subjects between keystage 2 and keystage 4 nationally. This would amount to 0.075 grade point less in each of the 8 subjects (0.60/8=0.0.075)

Subject Specific Gaps:

- Maths Disadvantaged: -0.44
- Maths Non-Disadvantaged: 0.11
- Maths Gap: 0.55
- English Disadvantaged: -0.46
- English Non-Disadvantaged: 0.12
- English Gap: 0.58

Analysis: Maths gap of 0.55 and English gap of 0.58 suggest, nationally, disadvanted students are underperforming or making 0.55 grade less progress in maths and 0.58 less progress in English, between keystage 2 and keystage 4 nationally.

Perventage 9-5 Gap:

- Percentage Disadvanted EngMaths_95: 28.09
- Percentage Nondisadv Student EngMaths_95: 50.01
- Percentage_95 Gap: 21.92

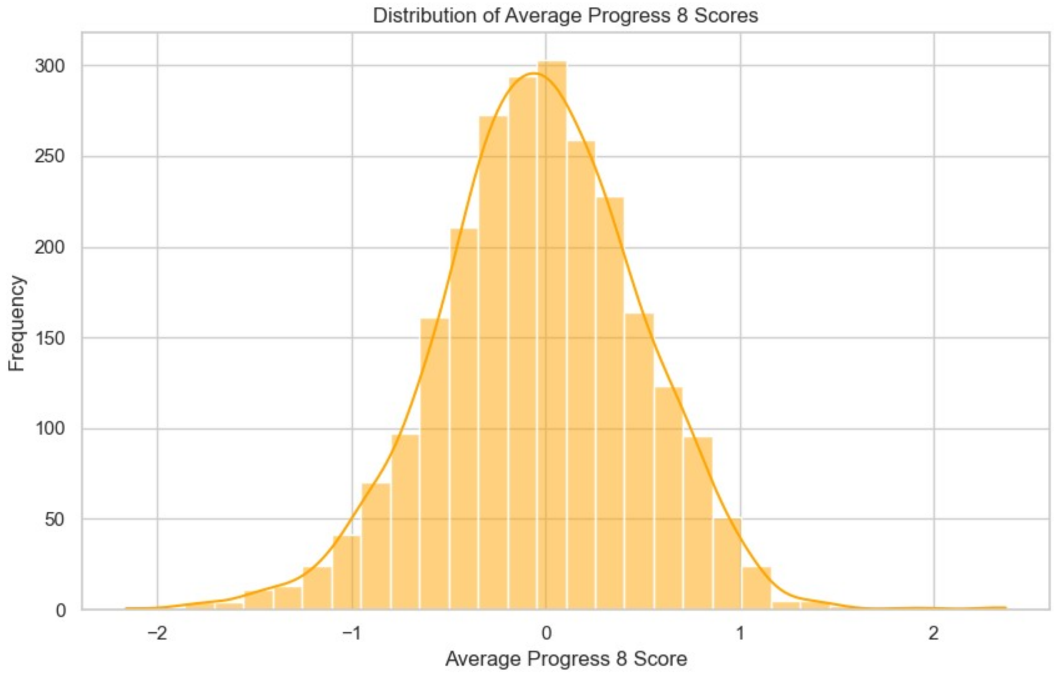
Analysis: Signficcant gap of 21.92 percentage points nationally between disadvatanged and advantaged students achieving grade 5 or above in English and Maths, suggests this needs to be addressed.

Visualisation

Distribution of Progress 8 Scores

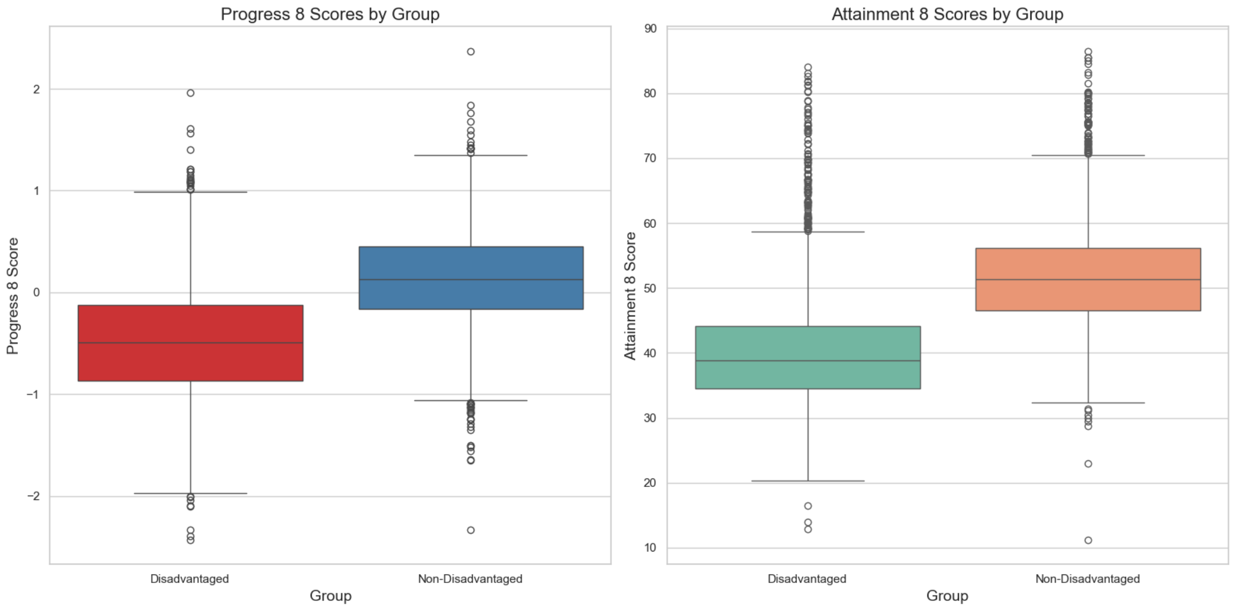
The histrogram showes an approximately normal distribution, as expected, since the results are standardised by exam boards. Most sudents would therfore have a progress 8 score of 0, with

68% of students falling within +1 or -1 standard deviations from the mean and 95% falling within +2 or -2 standard deviations from the mean.



Progess 8 and Attainment 8 Box Plots

Both box plots show disadvantaged students under performing. For progress 8, disadvatanged students have a negative progress 8 of -0.49 median score while advtanged students have a positive median score of 0.13, suggesting significant disparity. Both have a similar range and interqruatile range with a number of outliers. For attainment 8, the gap and distribution is as expected given the results.

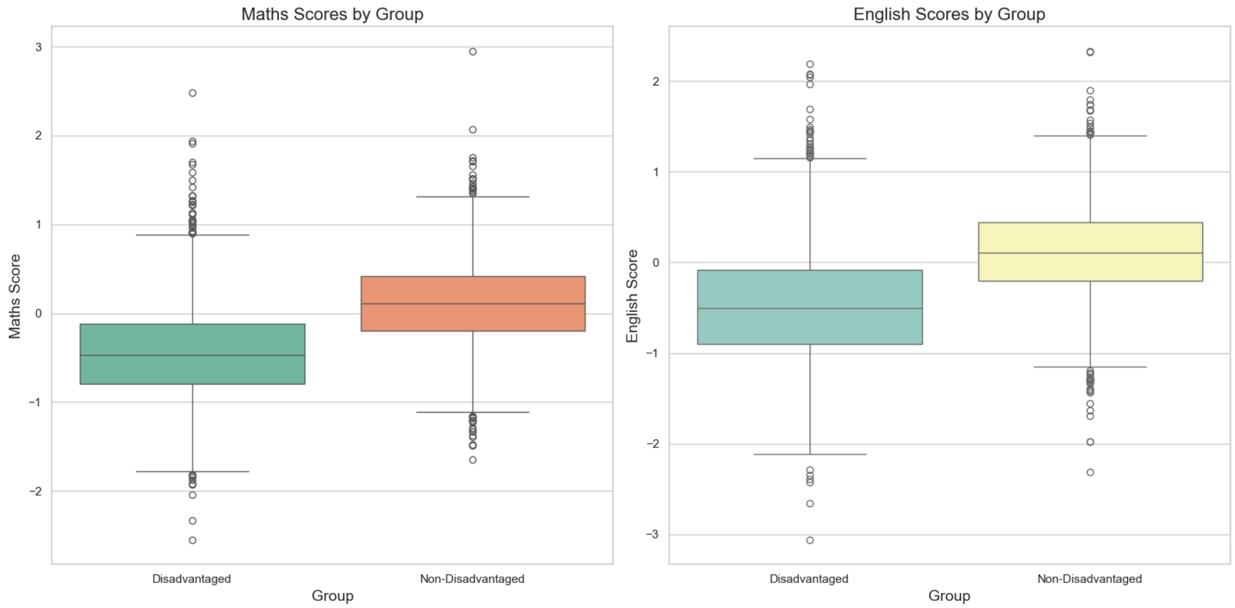


Percentage English and Mathematics Five Plus Box Plots

Maths Scores Summary	Median	Q1 (25%)	Q3 (75%)	IQR	Min	Max	Range
Disadvantaged	-0.47	-0.79	-0.12	0.67	-2.55	2.48	5.03
Non-Disadvantaged	0.11	-0.20	0.42	0.62	-1.65	2.95	4.60

English Scores Summary	Median	Q1 (25%)	Q3 (75%)	IQR	Min	Max	Range
Disadvantaged	-0.50	-0.90	-0.08	0.82	-3.06	2.19	5.25
Non-Disadvantaged	0.11	-0.20	0.44	0.64	-2.31	2.33	4.64

Both Maths and English have a negative median of -0.47 and -0.50 which is very concerning, given this is a national pattern, showing progress made by students between keystage 2 and keystage 4. English has a wider interquartile range for disadvantaged students, suggesting more variability. In both subjects, there is a greater difference between the minimum values, then between the maximum values, suggesting the disadvatnaged students will significiantly underperform than over perform.



Objective 2 Identify and Analyse Outlier Schools in Positive Progress 8 of Disadavantaged Pupils

Explanation of Results

Outlier schools for progress 8 were identified and then further categories as:

a) Schools which are outliers only for non-disadvantaged pupils

b) Schools which are outliers only for disadvantaged pupils

c) schools which are outliers for both non-disadvantaged and disadvantaged

- Overall schools which are outliers in both categories will do significantly better for disadvantaged pupils.
- There is also a higher correlation (0.85) between progress 8 disadvantaged pupils and progress 8 in general, suggesting success breeds success.
- Unexpectedly, funding (FSM(-0.45), total (-0.48) and pupil premium (-0.45)) all have negative correlation with progress 8 disadvantaged. This would need to be explored further as the range of the funding may be very small, and not being a good measure of proportionality.
- Small postitive correlation of progress 8 disadvantaged with percentage of disadvantage pupils (0.19) suggest disadvantaged pupils may do better where there are more such pupils.
- Index of multiple deprivation - has a negative correlation, suggesting lower values of the index ie. deprivation decreased, progress9 disadvantaged pupils will decline slightly, suggesting disadvantaged pupils' performance is expected to decrease when there is more deprivation.

Visualisation

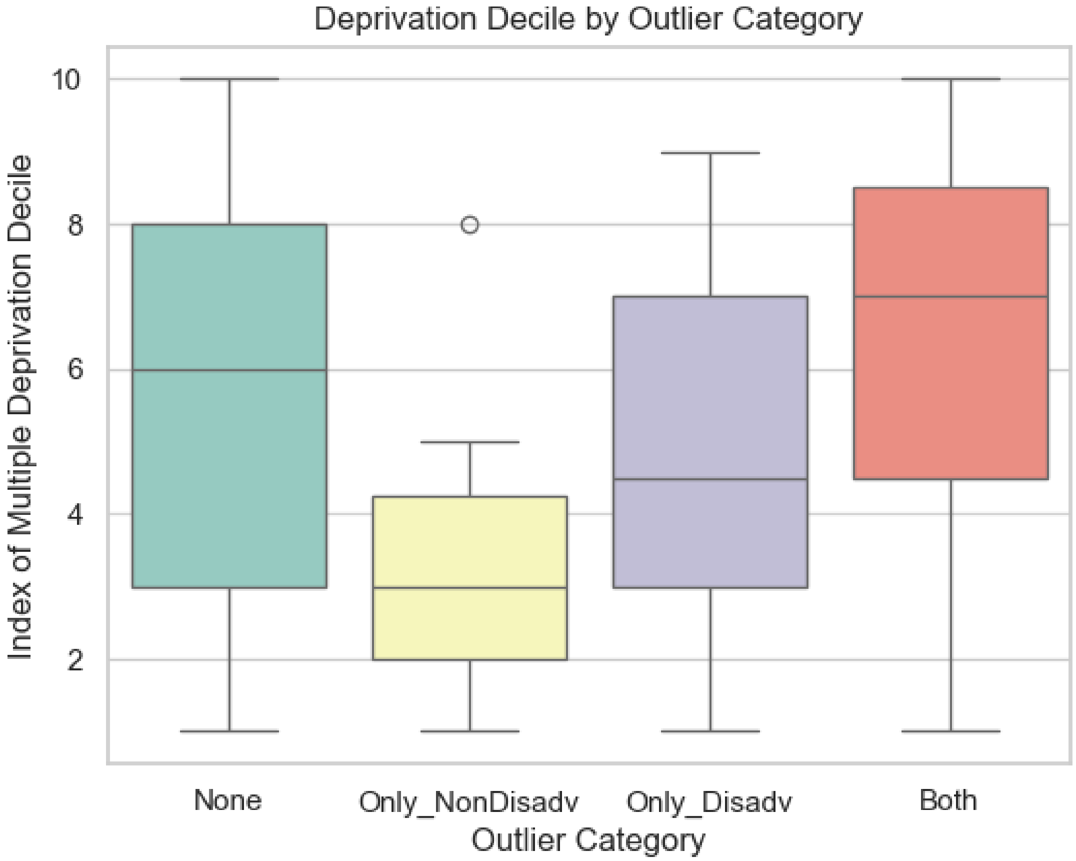
Heatmap provides insights into outlier schools in areas such as progress 8 score, funding, deprivation index etc.

obj2_heatmap of outlier schools.png

Summary Statistics for 'Index of Multiple Deprivation Decile' by 'Outlier_Category'

Outlier_Category	Median	Q1 (25%)	Q3 (75%)	IQR	Min	Max	Range
Both	7.0	4.5	8.50	4.00	1.0	10.0	9.0
None	6.0	3.0	8.00	5.00	1.0	10.0	9.0
Only_Disadv	4.5	3.0	7.00	4.00	1.0	9.0	8.0
Only_NonDisadv	3.0	2.0	4.25	2.25	1.0	8.0	7.0

Outlier schools only in progres 8 for only non-disadvantaged students, stand out as having a significanlty lower median of deprivation index, suggesting non-disadvantaged students tend to come from more deprived areas in such schools. This could be due to more focused support given they would stand out and be top of their school.



Summary Statistics for 'Progress8' by 'Outlier_Category'

Outlier_Category	Median	Q1 (25%)	Q3 (75%)	IQR	Min	Max	Range
Both	1.450	1.1950	2.140	0.9450	0.81	2.37	1.56
None	-0.040	-0.3600	0.290	0.6500	-2.16	1.83	3.99
Only_Disadv	0.955	0.8500	1.070	0.2200	0.25	1.38	1.13
Only_NonDisadv	1.085	0.9375	1.125	0.1875	0.61	1.49	0.88

Non positive outlier schools are are expected nearing 0; the minor difference may be due to negative outlier schools being included in that group. Schools which are outliers in both categories are much better performing with highest median and maximum score.

Obj2_Progress8 by Outlier Category.png

Summary Statistics for 'Progress8_Disadvantaged_2022' by 'Outlier_Category':

Outlier_Category	Median	Q1 (25%)	Q3 (75%)	IQR	Min	Max	Range
Both	1.40	1.19	1.585	0.395	1.01	1.96	0.95
None	-0.50	-0.87	-0.140	0.730	-2.43	0.99	3.42
Only_Disadv	1.09	1.07	1.125	0.055	1.01	1.21	0.20
Only_NonDisadv	0.60	0.51	0.775	0.265	0.22	0.99	0.77

Similar to before, disadvantaged pupils do better in schools which are outliers in both categories. Only disadvataged outlier schools have a very small IQR, suggesting an excellent level of consistency and low variability.

Obj2_Progress8 of Disadvantaged Pupils by Outlier Category.png

Objective 3 Identify and evaluate the top performing multi-academy trusts in supporting disadvantaged pupils

Explanation of Results

Summary:

Variable	Correlation with prog8_score_disadv
avg_progress8_score	+0.51
deprivation_index	-0.37
progress8_gap	-0.68
maths_gap	-0.63
english_gap	-0.58
attainment8_gap	-0.63
FiveGCSE_gap	-0.64
school_count	-0.29

Explaination:

- Strong positive correlation between the overall average Progress 8 score and the Progress 8 score for disadvantaged students shows, MATS that tend to perform well in progress 8 also tend to do so for disadvantaged students.

- The negative correlation between the deprivation index and the Progress 8 score for disadvantaged students suggests socio-economic factors can significantly impact student progress.
- Progress 8 score for disadvantaged students is negatively correlated with progress 8 gap; this would sugggest disadvantaged students will perform better in schools where there is a smaller progress 8 gap.
- School count in each MAT, has a negative correlation with average progress 8 (-0.46) and progres 8 for disadavtanged (-0.29) suggesting MATs with more schools may struggle with higher average progress 8 scores. This is understandable, and can be investifated further, as often free schools are set up by the MAT from the ground up will perform better, where as under performing schools which the MAT may have taken on to improve will impacts the average progress 8 result.

Visualisation

Identify Top Performing MATs based on Progress 8 Disadvantaged Students

Some MATS, although top performing for progress 8 overall, may not be top performing for disadvantaged pupils. e.g. "tar Academies and Chiltern Learning Trust have significantly higher Progress 8 scores for disadvantaged pupil, showing their strategies of support are efficient. United Learning Trust and Russell Education suggest they are making less progress with disadvantaged students.

Obj3_Progress8 disadvatanged top 10 MATs.png

Correlation Matrix for Top 10 MATS

Diagram show correlation for top 10 MATS with highest progress 8 values for disadvantaged pupils. This can be used to look at factors influencing progress 8 score for disadvantaged pupils and hence further analyse the performance of MATs.

Obj3_Correlation Matrix top 10 MATS.png

Progreess of Disadavantaged vs Advantaged Pupils

Progress 8 Gap - smaller gap between advantaged and disadvantaged poupils indicates better equity - Star Academies has the smallest gap of 0.264 followed by Chiltern Learning Trust of 0.408; While Education and Leadership Trust and Harris Federation have gaps of 0.733 and 0.623 respectively.

Obj3_Progress8 disadv vs advantaged in top 10 MATs.png

Deprivation Index vs Progress 8

Diagram shows average progress 8 scores of MATs again the multiple deprivation index. Star Academies has the highest average progress 8 score (0.64) yet the lowest deprivation index of 2.4 suggesting it is achieving very high despite have the most socio-economic challenges with deprivation index of 5.8.

Obj3_Deprivation Index vs P8 top 10 MATS.png

Deprivation Index vs Progress 8 for Disdavantaged Pupils

This diagram compares deprivation index with progress 8 performance of disadvantaged pupils. Star Academies stands out again with the highest progress 8 for disadvantaged pupils while also facing the most social economic deprivation. With a negative progress 8 and higher deprivation index, Russell Education and United Learning Trusts suggest disadvantaged pupils are making less than expected progress.

Obj3_Progress8 Disadv vs Deprivation Index for Top 10 MATs.png

Conclusion and presentation

Achievements

- I successfully managed to create a reliable data set by merging data from the he Department for Education DfE based on their Unique Reference Number URN code of schools and Multi-Academy Trusts MATs, and then linking a data from Ministry of Housing, Communities & Local Government to get the Index of Multiple Deprivation on postcode.

- The excpected gap between disadvantaged and advantaged pupils was explored. I confirmed that the gap exists in all academic variables measured which includes progress 8 (0.6), attainment 8 (11.6), Maths (0.55), English (0.58) and strong passes in both subjects (21.9 %).

- Outlier schools in progress 8 were then identified, categorised and analysed based on the groups of students they were outlier schools in i.e.

  - a) disadvantaged pupils only,
  - b) non-disadvantaged pupils only
  - c) both

  It was found expected variables such as funding, didnt have a significant correlation with disadvantage students' results, not just in progress 8, but attainment, English and Maths. The Index of Deprivation however, showed an expected impact with more deprivation leading to a drop in performance for disadvantaged students.

- Finally, top ranking MATS for progress 8 disadvantaged were identified and analysed. It was found the best MATs in supporting disadvantaged pupuls, close the gap and are able to overcome deprivation barriers with remarkable success. All the data analysis addressed and answered the objectives mentioned at the beginning of the notebook.

Limitations

Regression Analysis Further work, with time, would explore regression analysis on the data set. I would also be interested in further exploring categorical categories and their impact.

Time in Trust Also I could further filter schools which may be special-measure and hence impact the MAT progress 8 score. Another factor is the time schools have spent in the Trust; longer periods would suggest the Trust's methodology has been better understood and applied whereas younger schools may not yet be at the stage of improved progress-8 scores if they are yet to fully implement the Trust's strategy and policies.

Culture Certain things which are qualitative such as culture, may have a large influence on an organisations health and success. This can better be determined by actual school visits.

Future Work

Outlier Trusts - Strategy and Framework The outlier Trusts, should be further explored, particulary those that have managed to close the disadvantage gap.

Funding Allocation and Usage Further investgations can also be done on effective use of funding. The Ofsted report or further details from individual schools/MATs may be needed get details of strategy policy used.

Time series analysis In the future, I would like to work with a larger data set spanning back 5 or more years.

Machine Learning Models The data would make for a potential project in which I can apply machine learning models to find further trends over time. KNN models can be used to group schools for cluster analysis. Also unsupervised learning could be used to find trends which other may not be evident.

Geospational Analysis Conduct geospational analysis of MATS and evaluate schools based on clusters of proximity/ and other areas such as geospatial location and distribution of schools in MAT.

Text Analysis Another suggestion would be to do text analysis of Ofsted reports and link it to the Ofsted grade and historical trends of the school.

Video Presentation

Please submit a screen-capture video with your voiceover, providing a concise explanation of your project's design, key findings, successful aspects, and any challenges encountered. The duration of the video should be between 5 and 10 minutes in MP4 format.

References

Institute for Fiscal Studies. (2024, May). The past and future of UK health spending. https://www.ifs.org.uk/publications/health-spending-report IFS Report

Busby, E. (2024, October 24). Gap between private and state school pupils going to top universities widens. The Independent. https://www.independent.co.uk/news/uk/gap-england-department-for-education-government-data-b2634966.html

Phillipson, B. (2024, November 4). *Letter from the Secretary of State for Education*. Department for Education.

Tes. (2024, January 17). How many schools are there in the UK? Retrieved from https://www.tes.com/magazine/analysis/general/how-many-schools-in-the-uk