# Dependencies and Setup

import pandas as pd

from pathlib import Path

# File to Load (Remember to Change These)

school\_data\_to\_load = Path("Resources/schools\_complete.csv")

student\_data\_to\_load = Path("Resources/students\_complete.csv")

# Read School and Student Data File and store into Pandas DataFrames

school\_data = pd.read\_csv(school\_data\_to\_load)

student\_data = pd.read\_csv(student\_data\_to\_load)

# Combine the data into a single dataset.

school\_data\_complete = pd.merge(student\_data, school\_data, how="left", on=["school\_name", "school\_name"])

school\_data\_complete.head()

Student IDstudent\_namegenderyearschool\_namereading\_scoremaths\_scoreSchool IDtypesizebudget00Paul BradleyM9Huang High School96940Government2917191063511Victor SmithM12Huang High School90430Government2917191063522Kevin RodriguezM12Huang High School41760Government2917191063533Richard ScottM12Huang High School89860Government2917191063544Bonnie RayF9Huang High School87690Government29171910635

# Calculate the total  number of unique schools in the dataset

unique\_school\_name=school\_data\_complete["school\_name"].unique()

total\_school\_number=len(unique\_school\_name)

total\_school\_number

15

#Calculate the total number of students

total\_students\_number=school\_data\_complete["student\_name"].count()

total\_students\_number

39170

#Calculate the total budget

total\_budget = school\_data["budget"].sum()

total\_budget

24649428

#Calculate the average math score in the dataset

average\_math\_score = school\_data\_complete["maths\_score"].mean()

average\_math\_score

70.33819249425581

#Calculate the average reading score

average\_reading\_score = school\_data\_complete["reading\_score"].mean()

average\_reading\_score

69.98013786060761

# Calculate the percentage passing maths (the percentage of students who passed maths)

students\_passing\_maths = school\_data\_complete.loc[school\_data\_complete["maths\_score"] >= 50]

number\_students\_passing\_maths = students\_passing\_maths["Student ID"].count()

percentage\_passing\_maths = (number\_students\_passing\_maths / total\_students\_number) \* 100

percentage\_passing\_maths

**86.07863160582077**

# Calculate the percentage passing reading (the percentage of students who passed reading)

students\_passing\_reading = school\_data\_complete.loc[school\_data\_complete["reading\_score"] >= 50]

number\_students\_passing\_reading = students\_passing\_reading["Student ID"].count()

percent\_passing\_reading = (number\_students\_passing\_reading / total\_students\_number) \* 100

percent\_passing\_reading

84.42685728874139

#Calculate overall passing (the percentage of students who passed maths AND reading)

overall\_passing = school\_data\_complete[(school\_data\_complete["maths\_score"] >= 50) & (school\_data\_complete["reading\_score"] >=50)]

overall\_passing\_count = overall\_passing["Student ID"].count()

overall\_passing\_percent = (overall\_passing\_count / total\_students\_number) \* 100

overall\_passing\_percent

72.80827163645647

 display key metrics in a data frame

district\_summary = pd.DataFrame({

    "total schools": total\_school\_number,

    "total students": total\_students\_number,

    "total budget": total\_budget,

    "average math score": average\_math\_score,

    "average reading score": average\_reading\_score,

    "percent passing maths": percentage\_passing\_maths,

    "percent passing reading": percent\_passing\_reading,

    "percent overall passing": overall\_passing\_percent,

}, index=[0])

district\_summary



A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedGraphical user interface, text

Description automatically generatedText

Description automatically generatedGraphical user interface, text

Description automatically generatedGraphical user interface, text

Description automatically generatedGraphical user interface, text

Description automatically generatedA screenshot of a computer

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedText

Description automatically generatedText

Description automatically generatedGraphical user interface, text

Description automatically generatedText

Description automatically generatedGraphical user interface, text

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedText

Description automatically generated with low confidence