Unit 4 | Assignment – PyCitySchools

SMU Data Science Bootcamp

Zeinab MASSUDI

Contents

Context

In order to make **strategic decisions regarding future school budgets and priorities,** the school board and mayor expressed the need for detailed insights into current school performances within the district.

Hence, **our task was to analyze the district-wide standardized test results and showcase obvious trends in school performance**.

Objectives

The task was to aggregate the data and calculate each of the following (by school, school type and budget brackets):

1. Total Students
2. Total School Budget
3. Per Student Budget
4. Average Math Score
5. Average Reading Score
6. % Passing Math
7. % Passing Reading
8. Overall Passing Rate (Average of the above two)

Contents

2. Methodology

Our approach is made up of several logical steps using methods, functions and conditions to merge and analyse the district school data and the schools' generated math and reading exam resultsdataset.

1 Importing libraries and merging datasets

```
In [1]: # Dependencies and Setup
        import pandas as pd
        import matplotlib.pyplot as plt


        # File to Load (Remember to Change These)
        school_data_to_load = "/Users/zeinabmassudi/Desktop/schools_complete.csv"
        student_data_to_load = "/Users/zeinabmassudi/Desktop/students_complete.csv"

        # Read School and Student Data File and store into Pandas Data Frames
        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"])
```

The data was analysed using Pandas, a software library for the Python programming language, that provides high performance data structures and data analysis tools.

2 Aggregating data by School, School Size and School Type using "groupby" function

```
In [11]: # Rename dataset and group by school name
         grpby = school_data_complete.groupby(['school_name'])

         # Calculate student count per school
         #stu_tt = pd.Series(stu_total) or...
         stu_total = grpby['size'].count()

         # Define school type
         stu_type = grpby['type'].first()

         # Math average score per school
         sch_math_mean = grpby['math_score'].mean()

         # Average reading score per school
         sch_reading_mean = grpby['reading_score'].mean()

         # MATH PASSING RATE CALCULATIONS
         # Count of students that passed math exam
         math_pass_ct = school_data_complete[school_data_complete['math_score'] > 70].groupby('school_name')['math_score'] .cour

         # Percentage of total school students that passed math exam
         math_pass_rt = math_pass_ct/ stu_total*100
```

The groupby function splits the data into groups based on a particular criteria. Groups can also be formed by more than one category and condition.

3 Extracting specific grade columns using .loc method to analyse various grade performance

```
In [15]: # Create average maths score series for each grade using .loc conditions (groupby school name)
ninth_m = school_data_complete.loc[school_data_complete["grade"]=="9th"].groupby('school_name').mean()['math_score']
tenth_m = school_data_complete.loc[school_data_complete["grade"]=="10th"].groupby('school_name').mean()['math_score']
eleventh_m = school_data_complete.loc[school_data_complete["grade"]=="11th"].groupby('school_name').mean()['math_score']
twelfth_m = school_data_complete.loc[school_data_complete["grade"]=="12th"].groupby('school_name').mean()['math_score']
```

.loc is a method that allows the retrieval of rows from a dataframe using index labels and returns the row if its index exists in the caller dataframe.

school_name	9th	10th	11th	12th
Bailey High School	77.0837	76.9968	77.5156	76.4922
Cabrera High School	83.0947	83.1545	82.7656	83.2775
Figueroa High School	76.403	76.54	76.8843	77.1514
Ford High School	77.3613	77.6723	76.9181	76.18
Griffin High School	82.044	84.2291	83.8421	83.3562
Hernandez High School	77.4385	77.3374	77.136	77.1866
Holden High School	83.7874	83.4298	85	82.8554
Huang High School	77.0273	75.9087	76.4466	77.2256
Johnson High School	77.1879	76.6911	77.4917	76.8632
Pena High School	83.6255	83.372	84.3281	84.1215
Rodriguez High School	76.86	76.6125	76.3956	77.6907
Shelton High School	83.4208	82.9174	83.3835	83.779
Thomas High School	83.59	83.0879	83.4988	83.497
Wilson High School	83.0856	83.7244	83.1953	83.0358
Wright High School	83.2647	84.0103	83.8368	83.645
Schools Average	80.3516	80.3789	80.5759	80.4238

4 Creating School Size and Budget per Student Brackets using "Cut" function

```
In [20]: # Bins
size_bins = [0, 1500, 3000, 5000]
group_namess = ["Small (<1500)", "Medium (1500-3000)", "Large (3000-5000)"]
```

```
In [21]: # rename original dataframe
bin_pd1 = sch_summary_sorted

# Segment and sort data into buckets by School Size
bin_pd1["School Size"] = pd.cut(bin_pd1["Total Students"], size_bins, labels=group_namess)

#Groupby school size
bin_group1 = bin_pd1.groupby('School Size')

# Create dataframe
bin_group1[["Average Math Score",
            "Average Reading Score",
            "%Passing Math",
            "%Passing Reading",
            "%Overall Passing Rate"]].mean()
```

Out[21]:

School Size	Average Math Score	Average Reading Score	%Passing Math	%Passing Reading	%Overall Passing Rate
Small (<1500)	83.664898	83.892148	90.676736	92.778720	91.727728
Medium (1500-3000)	80.904987	82.822740	80.462303	87.605449	84.033876
Large (3000-5000)	77.063340	80.919864	64.323717	78.378664	71.351190

The cut function segments and sorts data into pre-defined bins (brackets).

Contents

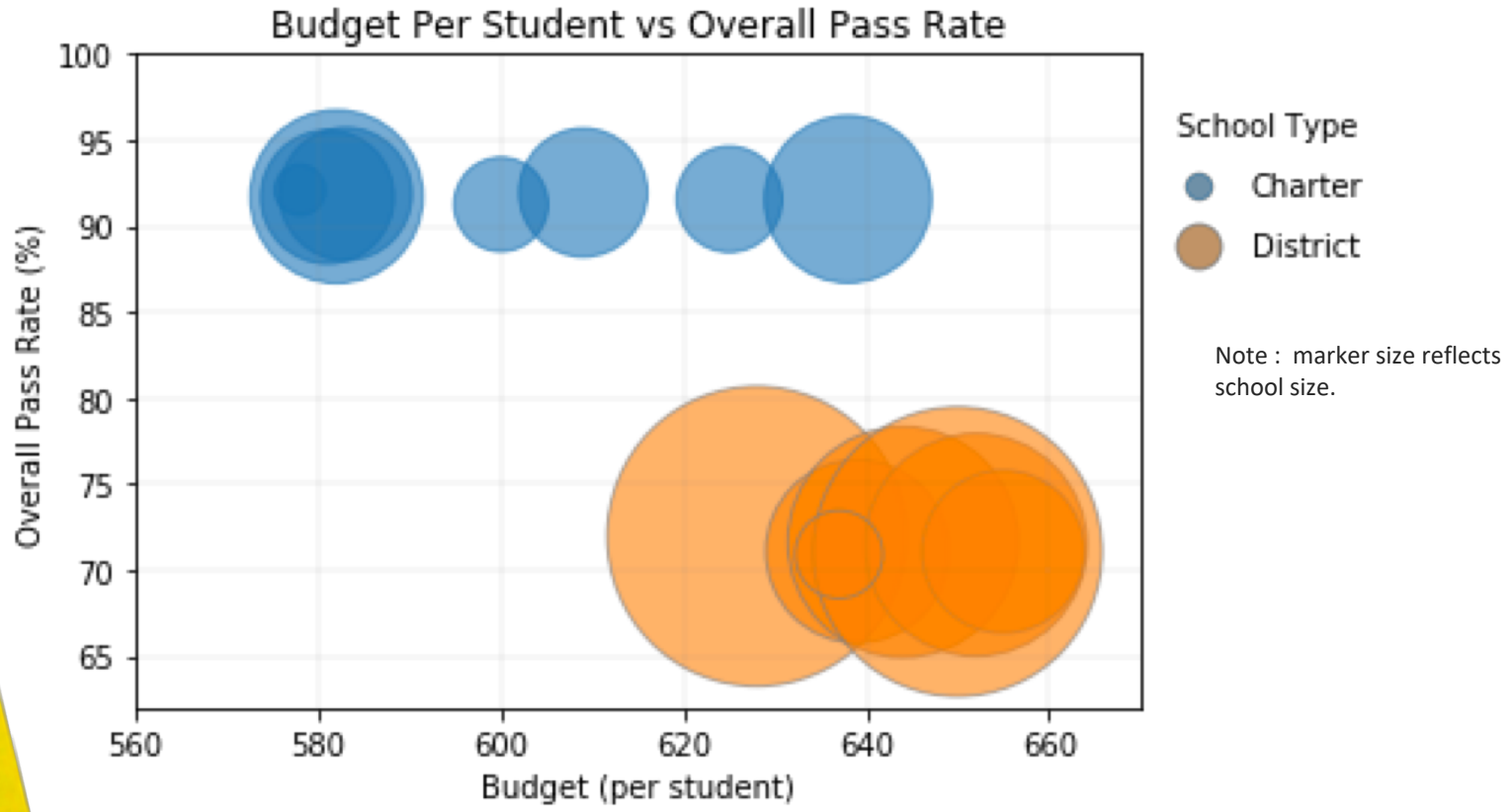
3. Results > Summary

Overall, students attending Charter Schools outperformed those of district schools despite the fact that district schools had higher budgets per student.

- Charter school yielded an average overall passing rate of 92% (20% higher than District schools).
- District schools tend to have higher budgets than Charter Schools.
- District schools are medium and large sized.
- The data also suggests that higher per student budgets doesn't ensure higher overall exam success.
- School size could have an impact on its performance.

Budget Per Student vs Overall Pass Rate

n(Total) = 15 schools, 39,170 students; | *n(Charter) = 8 schools, 12,194 students* | *n(District) = 8 schools, 26,976 students*



Note : marker size reflects school size.

Contents

4. Conclusions

As a whole, schools with higher budgets, did not yield better test results. By contrast, schools with higher spending per student ($645-675) underperformed compared to schools with smaller budgets(<$585 per student).

- **Top and Bottom Performing Schools**
  District schools students were the poorest performers especially in math with an overall passing rate average of 71%. On the other hand, Charter school yielded an average overall passing rate of 92%.

- **Larger budgets do not produce better results**
  As a whole, schools with higher budgets, did not produce better test results. By contrast, schools with higher spending per student ($645-675) underperformed compared to schools with smaller budgets(<$585 per student).

- **Impact of school size on performance**
  Overall, smaller and medium sized schools dramatically out-performed large sized schools on percentage math pass rate(92-84% passing vs 71%).

- **Other factors**
  Further analysis did not reveal any obvious trends of performance by specific gender groups or grades.

  There are no coincidences to success. Hence, a deeper dive into charter and district schools best practices, policies and teacher/student ratio could help explain the disparity between their students' performance.