PyCitySchools

January 11, 2018

1 Data Analysis - PyCitySchools

2 Observed Trend 1

- When analysing average math and reading scores, both stay consistent across grade level when grouped by school. We don't see major improvement in scores from any school.
- Observing Math passing rates, are always consistently lower across every metric, but the difference between math and reading passing rates is greater amoung lower performing schools, large schools, and higher spending per student which all seem to correlate.

3 Observed Trend 2

- Top 5 schools are all charter schools while the bottom 5 all district schools.
- General observation (one exception), per student spending is higher in bottom performing schools than top performing.

4 Observed Trend 3

- Looking at schools under 2000 students, have much higher passing rates than those with student populations above 2000. A comparision of 95 to 75%. The same trend is seen with high and low per student spending brackets and district versus charter schools.

In [1]:
```
#Dependencies
import pandas as pd
import numpy as np
import os

# define file path
schools_file = os.path.join('Resources','schools_complete.csv')
students_file = os.path.join('Resources', 'students_complete.csv')

# read schools file
schools_df = pd.read_csv(schools_file)

#read student file
students_df = pd.read_csv(students_file)
```

```
#renames for merge
schools_df.rename(columns = {'name': 'school'}, inplace = True)

merged_df = students_df.merge(schools_df, how = 'left', on = 'school')
```

4.1 District Summary

In [2]:
```
#create array of unique school names
unique_school_names = schools_df['school'].unique()

#gives the length of unique school names to give us how many schools
school_count = len(unique_school_names)

#district student count
dist_student_count = schools_df['size'].sum()

#student count from student file (to verify with district student count)
total_student_rec = students_df['name'].count()

#total budget
total_budget = schools_df['budget'].sum()

#calculations for number and % passing reading
num_passing_reading = students_df.loc[students_df['reading_score'] >= 70]['reading_sco
perc_pass_reading = num_passing_reading/total_student_rec

#calculations for number and % passing math
num_passing_math = students_df.loc[students_df['math_score'] >= 70]['math_score'].coun
perc_pass_math = num_passing_math/total_student_rec

#average math score calculation
avg_math_score = students_df['math_score'].mean()

#average reading score calculation
avg_reading_score = students_df['reading_score'].mean()

#Overall Passing Rate Calculations
overall_pass = students_df[(students_df['math_score'] >= 70) & (students_df['reading_s

# district dataframe from dictionary

district_summary = pd.DataFrame({

    "Total Schools": [school_count],
    "Total Students": [dist_student_count],
    "Total Budget": [total_budget],
    "Average Reading Score": [avg_reading_score],
```

```
        "Average Math Score": [avg_math_score],
        "% Passing Reading":[perc_pass_reading],
        "% Passing Math": [perc_pass_math],
        "Overall Passing Rate": [overall_pass]

    })

    #store as different df to change order
    dist_sum = district_summary[["Total Schools",
                                 "Total Students",
                                 "Total Budget",
                                 "Average Reading Score",
                                 "Average Math Score",
                                 '% Passing Reading',
                                 '% Passing Math',
                                 'Overall Passing Rate']]

    #format cells
    dist_sum.style.format({"Total Budget": "${:,.2f}",
                           "Average Reading Score": "{:.1f}",
                           "Average Math Score": "{:.1f}",
                           "% Passing Math": "{:.1%}",
                           "% Passing Reading": "{:.1%}",
                           "Overall Passing Rate": "{:.1%}"})
```

Out[2]: <pandas.io.formats.style.Styler at 0x1a04136d908>

4.2 School Summary

```
In [3]: #groups by school
        by_school = merged_df.set_index('school').groupby(['school'])

        #school types
        sch_types = schools_df.set_index('school')['type']

        # total students by school
        stu_per_sch = by_school['Student ID'].count()

        # school budget
        sch_budget = schools_df.set_index('school')['budget']

        #per student budget
        stu_budget = schools_df.set_index('school')['budget']/schools_df.set_index('school')['

        #avg scores by school
        avg_math = by_school['math_score'].mean()
        avg_read = by_school['reading_score'].mean()
```

```
        # % passing scores
        pass_math = merged_df[merged_df['math_score'] >= 70].groupby('school')['Student ID'].co
        pass_read = merged_df[merged_df['reading_score'] >= 70].groupby('school')['Student ID']
        overall = merged_df[(merged_df['reading_score'] >= 70) & (merged_df['math_score'] >= 70

        sch_summary = pd.DataFrame({
            "School Type": sch_types,
            "Total Students": stu_per_sch,
            "Per Student Budget": stu_budget,
            "Total School Budget": sch_budget,
            "Average Math Score": avg_math,
            "Average Reading Score": avg_read,
            '% Passing Math': pass_math,
            '% Passing Reading': pass_read,
            "Overall Passing Rate": overall
        })




        #munging
        sch_summary = sch_summary[['School Type',
                                   'Total Students',
                                   'Total School Budget',
                                   'Per Student Budget',
                                   'Average Math Score',
                                   'Average Reading Score',
                                   '% Passing Math',
                                   '% Passing Reading',
                                   'Overall Passing Rate']]

        #formatting
        sch_summary.style.format({'Total Students': '{:,}',
                                   "Total School Budget": "${:,}",
                                   "Per Student Budget": "${:.0f}",
                                   'Average Math Score': "{:.1f}",
                                   'Average Reading Score': "{:.1f}",
                                   "% Passing Math": "{:.1%}",
                                   "% Passing Reading": "{:.1%}",
                                   "Overall Passing Rate": "{:.1%}"})
```

Out[3]: <pandas.io.formats.style.Styler at 0x1a042f0f828>

4.3 Top Performing Schools by Passing Rate

```
In [4]: # sort values by passing rate and then only print top 5
        top_5 = sch_summary.sort_values("Overall Passing Rate", ascending = False)
        top_5.head().style.format({'Total Students': '{:,}',
```

```
                                "Total School Budget": "${:,}",
                                "Per Student Budget": "${:.0f}",
                                "% Passing Math": "{:.1%}",
                                "% Passing Reading": "{:.1%}",
                                "Overall Passing Rate": "{:.1%}"})
```

Out[4]: <pandas.io.formats.style.Styler at 0x1a04136d048>

4.4 Bottom Performing Schools by Passing Rate

```
In [5]: #bottom 5 schools from worse to best
        #take tail of top5 sort and re-sort from worst to best
        bottom_5 = top_5.tail()
        bottom_5 = bottom_5.sort_values('Overall Passing Rate')
        bottom_5.style.format({'Total Students': '{:,}',
                               "Total School Budget": "${:,}",
                               "Per Student Budget": "${:.0f}",
                               "% Passing Math": "{:.1%}",
                               "% Passing Reading": "{:.1%}",
                               "Overall Passing Rate": "{:.1%}"})
```

Out[5]: <pandas.io.formats.style.Styler at 0x1a04136d860>

4.5 Math Scores by Grade

```
In [6]: #creates grade level average math scores for each school
        ninth_math = students_df.loc[students_df['grade'] == '9th'].groupby('school')["math_sc
        tenth_math = students_df.loc[students_df['grade'] == '10th'].groupby('school')["math_s
        eleventh_math = students_df.loc[students_df['grade'] == '11th'].groupby('school')["mat
        twelfth_math = students_df.loc[students_df['grade'] == '12th'].groupby('school')["math_

        math_scores = pd.DataFrame({
                "9th": ninth_math,
                "10th": tenth_math,
                "11th": eleventh_math,
                "12th": twelfth_math
        })
        math_scores = math_scores[['9th', '10th', '11th', '12th']]
        math_scores.index.name = "School"

        #show and format
        math_scores.style.format({'9th': '{:.1f}',
                                  "10th": '{:.1f}',
                                  "11th": "{:.1f}",
                                  "12th": "{:.1f}"})
```

Out[6]: <pandas.io.formats.style.Styler at 0x1a0435f2f60>

4.6 Reading Scores by Grade

In [7]:
```
#creates grade level average reading scores for each school
ninth_reading = students_df.loc[students_df['grade'] == '9th'].groupby('school')["readi
tenth_reading = students_df.loc[students_df['grade'] == '10th'].groupby('school')["read
eleventh_reading = students_df.loc[students_df['grade'] == '11th'].groupby('school')["
twelfth_reading = students_df.loc[students_df['grade'] == '12th'].groupby('school')["re

#merges the reading score averages by school and grade together
reading_scores = pd.DataFrame({
        "9th": ninth_reading,
        "10th": tenth_reading,
        "11th": eleventh_reading,
        "12th": twelfth_reading
})
reading_scores = reading_scores[['9th', '10th', '11th', '12th']]
reading_scores.index.name = "School"

#format
reading_scores.style.format({'9th': '{:.1f}',
                             "10th": '{:.1f}',
                             "11th": "{:.1f}",
                             "12th": "{:.1f}"})
```

Out[7]: <pandas.io.formats.style.Styler at 0x1a042ee1be0>

4.7 Scores by School Spending

In [8]:
```
# create spending bins
bins = [0, 584.999, 614.999, 644.999, 999999]
group_name = ['< $585', "$585 - 614", "$615 - 644", "> $644"]
merged_df['spending_bins'] = pd.cut(merged_df['budget']/merged_df['size'], bins, labels

#group by spending
by_spending = merged_df.groupby('spending_bins')

#calculations
avg_math = by_spending['math_score'].mean()
avg_read = by_spending['reading_score'].mean()
pass_math = merged_df[merged_df['math_score'] >= 70].groupby('spending_bins')['Student
pass_read = merged_df[merged_df['reading_score'] >= 70].groupby('spending_bins')['Stude
overall = merged_df[(merged_df['reading_score'] >= 70) & (merged_df['math_score'] >= 7



# df build
scores_by_spend = pd.DataFrame({
    "Average Math Score": avg_math,
    "Average Reading Score": avg_read,
    '% Passing Math': pass_math,
```

```
        '% Passing Reading': pass_read,
        "Overall Passing Rate": overall

    })

    #reorder columns
    scores_by_spend = scores_by_spend[[
        "Average Math Score",
        "Average Reading Score",
        '% Passing Math',
        '% Passing Reading',
        "Overall Passing Rate"
    ]]

    scores_by_spend.index.name = "Per Student Budget"
    scores_by_spend = scores_by_spend.reindex(group_name)

    #formating
    scores_by_spend.style.format({'Average Math Score': '{:.1f}',
                                  'Average Reading Score': '{:.1f}',
                                  '% Passing Math': '{:.1%}',
                                  '% Passing Reading':'{:.1%}',
                                  'Overall Passing Rate': '{:.1%}'})
```

Out[8]: <pandas.io.formats.style.Styler at 0x1a0435f2048>

4.8 Scores by School Size

```
In [9]: # create size bins
        bins = [0, 999, 1999, 99999999999]
        group_name = ["Small (<1000)", "Medium (1000-2000)" , "Large (>2000)"]
        merged_df['size_bins'] = pd.cut(merged_df['size'], bins, labels = group_name)

        #group by spending
        by_size = merged_df.groupby('size_bins')

        #calculations
        avg_math = by_size['math_score'].mean()
        avg_read = by_size['math_score'].mean()
        pass_math = merged_df[merged_df['math_score'] >= 70].groupby('size_bins')['Student ID']
        pass_read = merged_df[merged_df['reading_score'] >= 70].groupby('size_bins')['Student
        overall = merged_df[(merged_df['reading_score'] >= 70) & (merged_df['math_score'] >= 7


        # df build
        scores_by_size = pd.DataFrame({
            "Average Math Score": avg_math,
            "Average Reading Score": avg_read,
```

```
        '% Passing Math': pass_math,
        '% Passing Reading': pass_read,
        "Overall Passing Rate": overall

    })

    #reorder columns
    scores_by_size = scores_by_size[[
        "Average Math Score",
        "Average Reading Score",
        '% Passing Math',
        '% Passing Reading',
        "Overall Passing Rate"
    ]]

    scores_by_size.index.name = "Total Students"
    scores_by_size = scores_by_size.reindex(group_name)

    #formating
    scores_by_size.style.format({'Average Math Score': '{:.1f}',
                                 'Average Reading Score': '{:.1f}',
                                 '% Passing Math': '{:.1%}',
                                 '% Passing Reading':'{:.1%}',
                                 'Overall Passing Rate': '{:.1%}'})
```

Out[9]: <pandas.io.formats.style.Styler at 0x1a042f0fc50>

4.9 Scores by School Type

```
In [10]: # group by type of school
         by_type = merged_df.groupby("type")

         #calculations
         avg_math = by_type['math_score'].mean()
         avg_read = by_type['math_score'].mean()
         pass_math = merged_df[merged_df['math_score'] >= 70].groupby('type')['Student ID'].cou
         pass_read = merged_df[merged_df['reading_score'] >= 70].groupby('type')['Student ID']
         overall = merged_df[(merged_df['reading_score'] >= 70) & (merged_df['math_score'] >=

         # df build
         scores_by_type = pd.DataFrame({
             "Average Math Score": avg_math,
             "Average Reading Score": avg_read,
             '% Passing Math': pass_math,
             '% Passing Reading': pass_read,
             "Overall Passing Rate": overall})

         #reorder columns
```

```
scores_by_type = scores_by_type[[
    "Average Math Score",
    "Average Reading Score",
    '% Passing Math',
    '% Passing Reading',
    "Overall Passing Rate"
]]
scores_by_type.index.name = "Type of School"


#formating
scores_by_type.style.format({'Average Math Score': '{:.1f}',
                             'Average Reading Score': '{:.1f}',
                             '% Passing Math': '{:.1%}',
                             '% Passing Reading':'{:.1%}',
                             'Overall Passing Rate': '{:.1%}'})
```

Out[10]: <pandas.io.formats.style.Styler at 0x1a042f0f6d8>