

main

September 21, 2019

```
[4]: # Dependencies and Setup
import pandas as pd

[5]: # File to Load (Remember to Change These)
school_data_to_load = "Resources/schools_complete.csv"
student_data_to_load = "Resources/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"])

[6]: school_data_complete.columns

[6]: Index(['Student ID', 'student_name', 'gender', 'grade', 'school_name',
        'reading_score', 'math_score', 'School ID', 'type', 'size', 'budget'],
        dtype='object')

[7]: Total_Schools = len(school_data_complete["school_name"].unique())
Total_Students = school_data_complete["Student ID"].count()
Total_Budget = school_data_complete["budget"].sum()
Average_Math_Score = school_data_complete["math_score"].mean()
Average_Reading_Score = school_data_complete["reading_score"].mean()
math_score_pass =
    →school_data_complete[(school_data_complete["math_score"]>=70)].
    →count()["math_score"]
reading_score_pass =
    →school_data_complete[(school_data_complete["reading_score"]>=70)].
    →count()["reading_score"]
math_pass_percent = (math_score_pass/Total_Students)*100
reading_pass_percent = (reading_score_pass/Total_Students)*100
overall_passing_rate = (Average_Math_Score + Average_Reading_Score)/2

[8]: district_summary = pd.DataFrame({"Total Schools":[Total_Schools],
        "Total Students":[Total_Students],
        "Total Budget":[Total_Budget],
```

```

        "Average Math Score": [Average_Math_Score],
        "Average Reading Score":
→ [Average_Reading_Score],
        "% Passing Math": [math_pass_percent],
        "% Passing Reading": [reading_pass_percent],
        "% Overall Passing Rate":
→ [overall_passing_rate]})
district_summary

```

```

[8]: Total Schools  Total Students  Total Budget  Average Math Score  \
0          15          39170    82932329558          78.985371

      Average Reading Score  % Passing Math  % Passing Reading  \
0          81.87784          74.980853          85.805463

      % Overall Passing Rate
0          80.431606

```

```

[10]: school_data_complete.groupby(["school_name"]).mean()

```

```

[10]:
      school_name  Student ID  reading_score  math_score  School ID  \
Bailey High School      20358.5      81.033963      77.048432      7.0
Cabrera High School      16941.5      83.975780      83.061895      6.0
Figueroa High School      4391.0      81.158020      76.711767      1.0
Ford High School      36165.0      80.746258      77.102592     13.0
Griffin High School      12995.5      83.816757      83.351499      4.0
Hernandez High School      9944.0      80.934412      77.289752      3.0
Holden High School      23060.0      83.814988      83.803279      8.0
Huang High School      1458.0      81.182722      76.629414      0.0
Johnson High School      32415.0      80.966394      77.072464     12.0
Pena High School      23754.5      84.044699      83.839917      9.0
Rodriguez High School      28035.0      80.744686      76.842711     11.0
Shelton High School      6746.0      83.725724      83.359455      2.0
Thomas High School      38352.0      83.848930      83.418349     14.0
Wilson High School      14871.0      83.989488      83.274201      5.0
Wright High School      25135.5      83.955000      83.682222     10.0

```

```

      size  budget
school_name
Bailey High School      4976.0  3124928.0
Cabrera High School      1858.0  1081356.0
Figueroa High School      2949.0  1884411.0
Ford High School      2739.0  1763916.0
Griffin High School      1468.0   917500.0
Hernandez High School      4635.0  3022020.0
Holden High School      427.0   248087.0
Huang High School      2917.0  1910635.0

```

Johnson High School	4761.0	3094650.0
Pena High School	962.0	585858.0
Rodriguez High School	3999.0	2547363.0
Shelton High School	1761.0	1056600.0
Thomas High School	1635.0	1043130.0
Wilson High School	2283.0	1319574.0
Wright High School	1800.0	1049400.0

```
[ ]: school_name_groupby = pd.DataFrame(school_data_complete.
    ↳groupby(["school_name"]))
school_math_pass = school_name_groupby.
    ↳loc[school_name_groupby["math_score"]>=70].count()
school_groupby_grade = school_name_groupby.(["grade"]).mean()
```

```
[34]: per_student_budget = school_data_complete.groupby(["school_name"]).
    ↳mean()["budget"] / school_data_complete.groupby(["school_name"]).
    ↳mean()["budget"]
```

```
[24]: school_summary = pd.DataFrame({ "School Type":school_data_complete.
    ↳groupby(["school_name"])["type"],
                                     "Total Students":school_data_complete.
    ↳groupby(["school_name"]).mean()["size"],
                                     "Total School Budget":school_data_complete.
    ↳groupby(["school_name"]).mean()["budget"],
                                     "Per Student Budget":[per_student_budget],
                                     "Average Math Score":school_data_complete.
    ↳groupby(["school_name"]).mean()["math_score"],
                                     "Average Reading Score":school_data_complete.
    ↳groupby(["school_name"]).mean()["reading_score"],
                                     "% Passing Math":
    ↳school_groupby["math_pass_percent"],
                                     "% Passing Reading":
    ↳school_groupby["reading_pass_percent"],
                                     "% Overall Passing Rate":
    ↳school_groupby["overall_passing_rate"]})
school_summary
```

```
↳
↳-----
ValueError                                Traceback (most recent call↳
↳last)

<ipython-input-24-97b7b4f5d40c> in <module>
      4                                "Per Student Budget":
↳[per_student_budget],
```

```

5                                     "Average Math Score":
↳ school_data_complete.groupby(["school_name"]).mean()["math_score"],
    ----> 6                                     "Average Reading Score":
↳ school_data_complete.groupby(["school_name"]).mean()["reading_score"]})
7 school_summary

```

```

~\Anaconda3\envs\PythonData\lib\site-packages\pandas\core\frame.py in
↳ __init__(self, data, index, columns, dtype, copy)
390                                     dtype=dtype, copy=copy)
391     elif isinstance(data, dict):
--> 392         mgr = init_dict(data, index, columns, dtype=dtype)
393     elif isinstance(data, ma.MaskedArray):
394         import numpy.ma.mrecords as mrecords

```

```

↳ ~\Anaconda3\envs\PythonData\lib\site-packages\pandas\core\internals\construction.
↳ py in init_dict(data, index, columns, dtype)
210         arrays = [data[k] for k in keys]
211
--> 212     return arrays_to_mgr(arrays, data_names, index, columns,
↳ dtype=dtype)
213
214

```

```

↳ ~\Anaconda3\envs\PythonData\lib\site-packages\pandas\core\internals\construction.
↳ py in arrays_to_mgr(arrays, arr_names, index, columns, dtype)
49     # figure out the index, if necessary
50     if index is None:
---> 51         index = extract_index(arrays)
52     else:
53         index = ensure_index(index)

```

```

↳ ~\Anaconda3\envs\PythonData\lib\site-packages\pandas\core\internals\construction.
↳ py in extract_index(data)
326                                     'length {idx_len}'
327                                     .format(length=lengths[0],
↳ idx_len=len(index)))
--> 328                                     raise ValueError(msg)
329     else:
330         index = ibase.default_index(lengths[0])

```

ValueError: array length 1 does not match index length 39185

```
[ ]: schools_passing_rate = school_summary.  
      ↪sort_values(["overall_passing_rate"],ascending=False)  
schools_passing_rate.head()  
schools_passing_rate.tail()
```

[14]:

```
File "<ipython-input-14-416d131ef50f>", line 1  
school_groupby_grade = school_name_groupby.(["grade"]).mean()  
                        ^
```

SyntaxError: invalid syntax