Academy of PY

```
In [1]: # Panda's HW - Academy of Py by Verna Orsatti
```

```
In [2]: # OBSERVED TREND 1 Narrative
        # OBSERVED TREND 2 Narrative
        # OBSERVED TREND 3 Narrative
```

```
In [3]: import pandas as pd
        import numpy as np
```

```
In [4]: schools_csv = "schools_complete.csv"
        students_csv = "students_complete.csv"
```

```
In [5]: # Useful
        # Remember that == is for boolean and = is to set a value
        # table_name = pd.DataFrame({"":[],"":[],"":[],"":[],"":[],"":[],"":[]
        ,"":[],"":[],"":[],"":[],"":[]})
```

```
In [6]: # STUDENTS - read csv as pandas dataframe
        df_stu = pd.read_csv(students_csv)
```

```
In [7]: # Re-name header labels
        df_stu.columns = ['Student ID', 'Name', 'Gender', 'Grade', 'School', '
        Reading_Score', 'Math_Score']
```

```
In [8]: # SCHOOLS - read csv as pandas dataframe
        df_sch = pd.read_csv(schools_csv)
```

```
In [9]: # Re-name Header labels
        df_sch.columns = ['School ID', 'School', 'Type', 'Size', 'Budget']
```

District Summary

```
In [10]: # District Summary  - Get data
         # THIS IS A ONE LINE SUMMARY DATA FRAME:
         # One line of data for:
         # Create a high level snapshot (in table form) of the district's key m
```

```
etrics, including:

# Total Schools
tot_sch = df_sch.School.count()

# Total Students in Schools
tot_stu = df_sch.Size.sum()

# Total Budget in Schools
tot_bud = df_sch.Budget.sum()

# Average Math Score from Students # returns boolean for all (True and
False)
ave_mathscores_all = df_stu.Math_Score.mean()

# Average Reading Score
ave_readscores_all = df_stu.Reading_Score.mean()

# % Passing Math - Based on 70% or better; how many passed/total stude
n
passing_math = df_stu.Math_Score.between(70,100, inclusive=True) # Ret
urns boolean of all in selected,
df_stu[passing_math]

dfmp = df_stu[passing_math]
perc_pass_math = (dfmp.Math_Score.count()/tot_stu) * 100

# % Passing Reading - Base on 70% or better
passing_reading = df_stu.Reading_Score.between(70,100, inclusive=True)
# Returns boolean of all in selected,

dfrp = df_stu[passing_reading]
perc_pass_reading = (dfrp.Reading_Score.count()/tot_stu) * 100

#   * Overall Passing Rate (Average of the above two)  - average of th
ose s
overall_pass_rate = (perc_pass_math + perc_pass_reading)/2

# Create Table - District Summary# DISTRICT SUMMARY
district_summary = pd.DataFrame({"Total Schools":[tot_sch],"Total Stud
ents":[tot_stu],"Total Budget":[tot_bud],
                                 "Average Math Score":[ave_mathscores_
all],"Average Reading Score":[ave_readscores_all],
                                 "% Passing Math":[perc_pass_math],"%
Passing Reading":[perc_pass_reading],"Overall Passing Rate":[overall_p
ass_rate]})

# Create Table - District Summary
district_summary = district_summary[["Total Schools","Total Students",
"Total Budget","Average Math Score","Average Reading Score","% Passing
Math","% Passing Reading","Overall Passing Rate"]]
# Need to format $ for budget
district_summary["Total Budget"] = district_summary["Total Budget"].ma
p("${:,.2f}".format)
district_summary
```

Out[10]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Ov Pas
0	15	39170	$24,649,428.00	78.985371	81.87784	74.980853	85.805463	80.39

In [11]:

```
## merge data sets df_stu & df_sch
merge_data = pd.merge(df_sch, df_stu, on=('School'))
```

School Summary

In [12]:

```
# School Summary - Get data from
# school is df_sch
# student is df_stu
# merge_data us combined data files

# Get Name school Type values
sch_type = df_sch.set_index(["School"])["Type"]

# Get student count per school
tot_sch_stu = merge_data["School"].value_counts()

# Get Total per School Budget
tot_sch_bud = merge_data.groupby(["School"]).mean()["Budget"]

# Compute per Student Budget
stu_bud = tot_sch_bud/tot_sch_stu

# Get Average Math and Reading scores per school
ave_math_score = merge_data.groupby(["School"]).mean()["Math_Score"]
ave_reading_score = merge_data.groupby(["School"]).mean()["Reading_Sco
re"]

# Compute Percent Passing for Math and Reading
sch_perc_pass_math = merge_data[merge_data["Math_Score"] >= 70].groupb
y("School").count()["Name"] #/tot_stu) * 100
sch_perc_pass_math =(sch_perc_pass_math/tot_sch_stu) * 100
sch_perc_pass_reading = merge_data[merge_data["Reading_Score"] >= 70].
groupby("School").count()["Name"]
```

```
sch_perc_pass_reading = (sch_perc_pass_reading/tot_sch_stu) * 100

# Overall Passing Rate (Average of the above two)  - average of those
s
over_pass_rate = (sch_perc_pass_math + sch_perc_pass_reading)/2

# Create Table - School Summary
school_summary = pd.DataFrame({"School Type": sch_type, "Total Student
s": tot_sch_stu, "Total School Budget": tot_sch_bud, "Per Student Budg
et": stu_bud, "Average Math Score": ave_math_score, "Average Reading S
core": ave_reading_score, "% Passing Math": sch_perc_pass_math, "% Pas
sing Reading": sch_perc_pass_reading, "Overall Passing Rate": over_pas
s_rate})
school_summary = school_summary[["School Type","Total Students","Total
School Budget","Per Student Budget", "Average Math Score","Average Rea
ding Score","% Passing Math","% Passing Reading","Overall Passing Rate
"]]

# Format columns where needed
school_summary["Total School Budget"] = school_summary["Total School B
udget"].map("${:,.2f}".format)
school_summary["Per Student Budget"] = school_summary["Per Student Bud
get"].map("${:,.2f}".format)
school_summary
```

Out[12]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math
Bailey High School	District	4976	$3,124,928.00	$628.00	77.048432	81.033963	66.680064
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471
Ford High School	District	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.309602
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371
Hernandez							

High School	District	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.752967
Holden High School	Charter	427	$248,087.00	$581.00	83.803279	83.814988	92.505855
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.594595
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366592
Shelton High School	Charter	1761	$1,056,600.00	$600.00	83.359455	83.725724	93.867121
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718
Wright High School	Charter	1800	$1,049,400.00	$583.00	83.682222	83.955000	93.333333

Top Performing Schools (By Passing Rate)

```
In [13]: # **Top Performing Schools (By Passing Rate)**

         # Resort and take 5
         # Create Table - Top Performing Schools (By Passing Rate)
         top_schools = school_summary.sort_values(["Overall Passing Rate"], asc
         ending = False).head(5)
         top_schools
```

Out[13]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477	9
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171	9
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.594595	9
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371	9
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718	9

Bottom Performing Schools (By Passing Rate)

In [14]:

```
# * Create a table that highlights the bottom 5 performing schools bas
ed on Overall Passing Rate. Include all of the same metrics as above.
# Re-sort and take 5

# Create Table - Bottom Performing Schools (By Passing Rate)
bottom_schools = school_summary.sort_values(["Overall Passing Rate"],
ascending = True).head(5)
bottom_schools
```

Out[14]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366592
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551
Ford High School	District	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.309602

Math Scores by Grade

In [15]:

```
# **Math Scores by Grade**

# Get Average values per grade for all students, grouped by school
m_nineth_score = merge_data[merge_data["Grade"] == "9th"].groupby("School").mean()["Math_Score"]
m_tenth_score = merge_data[merge_data["Grade"] == "10th"].groupby("School").mean()["Math_Score"]
m_eleventh_score = merge_data[merge_data["Grade"] == "11th"].groupby("School").mean()["Math_Score"]
m_twelfth_score = merge_data[merge_data["Grade"] == "12th"].groupby("School").mean()["Math_Score"]

# Create Table - Math Scores by Grade
df_ms_by_grade = pd.DataFrame({"9th":m_nineth_score,"10th":m_tenth_score,"11th":m_eleventh_score,"12th":m_twelfth_score})
ms_by_grade = df_ms_by_grade[["9th","10th","11th","12th"]]
ms_by_grade
```

Out[15]:

	9th	10th	11th	12th
School				
Bailey High School	77.083676	76.996772	77.515588	76.492218
Cabrera High School	83.094697	83.154506	82.765560	83.277487
Figueroa High School	76.403037	76.539974	76.884344	77.151369
Ford High School	77.361345	77.672316	76.918058	76.179963
Griffin High School	82.044010	84.229064	83.842105	83.356164
Hernandez High School	77.438495	77.337408	77.136029	77.186567
Holden High School	83.787402	83.429825	85.000000	82.855422
Huang High School	77.027251	75.908735	76.446602	77.225641
Johnson High School	77.187857	76.691117	77.491653	76.863248
Pena High School	83.625455	83.372000	84.328125	84.121547
Rodriguez High School	76.859966	76.612500	76.395626	77.690748
Shelton High School	83.420755	82.917411	83.383495	83.778976
Thomas High School	83.590022	83.087886	83.498795	83.497041
Wilson High School	83.085578	83.724422	83.195326	83.035794
Wright High School	83.264706	84.010288	83.836782	83.644986

Reading Scores by Grade

In [16]:

```
# **Reading Scores by Grade**

# Get Average values per grade for all students, grouped by school
r_nineth_score = merge_data[merge_data["Grade"] == "9th"].groupby("Sch
ool").mean()["Reading_Score"]
r_tenth_score = merge_data[merge_data["Grade"] == "10th"].groupby("Sch
ool").mean()["Reading_Score"]
r_eleventh_score = merge_data[merge_data["Grade"] == "11th"].groupby("
School").mean()["Reading_Score"]
r_twelfth_score = merge_data[merge_data["Grade"] == "12th"].groupby("S
chool").mean()["Reading_Score"]

# Create Table - Reading Scores by Grade
df_rs_by_grade = pd.DataFrame({"9th":r_nineth_score,"10th":r_tenth_sco
re,"11th":r_eleventh_score,"12th":r_twelfth_score})
rs_by_grade = df_rs_by_grade[["9th","10th","11th","12th"]]

rs_by_grade
```

Out[16]:

	9th	10th	11th	12th
School				
Bailey High School	81.303155	80.907183	80.945643	80.912451
Cabrera High School	83.676136	84.253219	83.788382	84.287958
Figueroa High School	81.198598	81.408912	80.640339	81.384863
Ford High School	80.632653	81.262712	80.403642	80.662338
Griffin High School	83.369193	83.706897	84.288089	84.013699
Hernandez High School	80.866860	80.660147	81.396140	80.857143
Holden High School	83.677165	83.324561	83.815534	84.698795
Huang High School	81.290284	81.512386	81.417476	80.305983
Johnson High School	81.260714	80.773431	80.616027	81.227564
Pena High School	83.807273	83.612000	84.335938	84.591160
Rodriguez High School	80.993127	80.629808	80.864811	80.376426
Shelton High School	84.122642	83.441964	84.373786	82.781671
Thomas High School	83.728850	84.254157	83.585542	83.831361
Wilson High School	83.939778	84.021452	83.764608	84.317673
Wright High School	83.833333	83.812757	84.156322	84.073171

Scores by School Spending

In [17]:

```
# **Scores by School Spending**

# Create Bins as required
bin_spend = [0,585,615,645,675]
bin_spend_label = ["<585","585-615","615-645","645-675"]
spend_school_summary = school_summary
# Fill bins
spend_school_summary["Spending Ranges (Per Student)"] = pd.cut(stu_bud
, bin_spend, labels = bin_spend_label)
# Get values of average for the columns
sp_ave_mathscore = spend_school_summary.groupby(["Spending Ranges (Per
Student)"]).mean()['Average Math Score']
sp_ave_readscore = spend_school_summary.groupby(["Spending Ranges (Per
Student)"]).mean()['Average Reading Score']
sp_pass_math =  spend_school_summary.groupby(["Spending Ranges (Per St
udent)"]).mean()['% Passing Math']
sp_pass_read =  spend_school_summary.groupby(["Spending Ranges (Per St
udent)"]).mean()['% Passing Reading']
sp_over_pass =  (sp_pass_math + sp_pass_read) / 2

# Create Table - Scores by School Spending
spendscores = pd.DataFrame({"Average Math Score":sp_ave_mathscore, "Av
erage Reading Score":sp_ave_readscore,
                               "% Passing Math":sp_pass_math,"% Passin
g Reading":sp_pass_read,
                                    "Overall Passing Rate":sp_over_pas
s})
scores_by_school_spending = spendscores[["Average Math Score", "Averag
e Reading Score",
                               "% Passing Math","% Passing Reading","O
verall Passing Rate"]]
scores_by_school_spending
```

Out[17]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
Spending Ranges (Per Student)					
<585	83.455399	83.933814	93.460096	96.610877	95.035486
585-615	83.599686	83.885211	94.230858	95.900287	95.065572
615-645	79.079225	81.891436	75.668212	86.106569	80.887391
645-675	76.997210	81.027843	66.164813	81.133951	73.649382

Scores by School Size

In [18]:

```
# **Scores by School Size**

# Define bins for reasonable school spending ranges on average student
spending amounts
# Create Bins as required
bin_size = [0,1000,2000,5000]
bin_size_label = ["Small <1000","Medium (1000-2000)","Large (2000-5000
)"]

# Fill bins
school_summary["School Size"] = pd.cut(school_summary["Total Students"
], bin_size, labels = bin_size_label)

# Get values of averages for columns
sz_ave_mathscore = school_summary.groupby(["School Size"]).mean()['Ave
rage Math Score']
sz_ave_readscore = school_summary.groupby(["School Size"]).mean()['Ave
rage Reading Score']
sz_pass_math =  school_summary.groupby(["School Size"]).mean()['% Pass
ing Math']
sz_pass_read =  school_summary.groupby(["School Size"]).mean()['% Pass
ing Reading']
sz_over_pass =  (sz_pass_math + sz_pass_read) / 2

# Create Table - Scores by School Spending
spendscores = pd.DataFrame({"Average Math Score":sz_ave_mathscore, "Av
erage Reading Score":sz_ave_readscore,
                              "% Passing Math":sz_pass_math,"% Passin
g Reading":sz_pass_read,
                                   "Overall Passing Rate":sz_over_pas
s})
scores_by_school_size = spendscores[["Average Math Score", "Average Re
ading Score",
                              "% Passing Math","% Passing Reading","O
verall Passing Rate"]]
scores_by_school_size
```

Out[18]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
School Size					
Small <1000	83.821598	83.929843	93.550225	96.099437	94.824831
Medium (1000-2000)	83.374684	83.864438	93.599695	96.790680	95.195187
Large (2000-5000)	77.746417	81.344493	69.963361	82.766634	76.364998

Scores by School Type

In [19]:

```
# **Scores by School Type**

# Get values of averages for columns
st_ave_mathscore = school_summary.groupby(["School Type"]).mean()['Ave
rage Math Score']
st_ave_readscore = school_summary.groupby(["School Type"]).mean()['Ave
rage Reading Score']
st_pass_math =  school_summary.groupby(["School Type"]).mean()['% Pass
ing Math']
st_pass_read =  school_summary.groupby(["School Type"]).mean()['% Pass
ing Reading']
st_over_pass =  (st_pass_math + st_pass_read) / 2

# Create Table - Scores by School Spending
typescores = pd.DataFrame({"Average Math Score":st_ave_mathscore, "Ave
rage Reading Score":st_ave_readscore,
                               "% Passing Math":st_pass_math,"% Passin
g Reading":st_pass_read,
                                    "Overall Passing Rate":st_over_pas
s})
scores_by_school_type = typescores[["Average Math Score", "Average Rea
ding Score",
                               "% Passing Math","% Passing Reading","O
verall Passing Rate"]]
scores_by_school_type
```

Out[19]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
School Type					
Charter	83.473852	83.896421	93.620830	96.586489	95.103660
District	76.956733	80.966636	66.548453	80.799062	73.673757

In [20]:

```
# All in a day's work... give or take a few more
```