Data Research on Enrollment by School Year (Schools) From 2017-2018 in the State of Massachusetts

By: Edwin Asitimbay, Daniela Bautista, John Lee

ABOUT

The data set records the data of enrollment in the state of Massachusetts during the 2017-2018 academic year. It highlights each city (municipality) and each school within a city. The data frame consists of not just the total number of enrollments, but also the number of enrollments base on grade level, special classes, nationality/race/sex, language profienciency, disability, income standing, and high needs. Along with numbers, the data set also captured the percentage. However, despite having the columns related to income standing, no data has been collected on those.

What Is The Purpose of This Project

- We want to see the amount of enrollments, in both the highest and lowest out of all the schools in the state of Massachusetts during the 2017 - 2018 academic school year.
- As well as to find the overall average

In This Presentation, We'll Present

- Data Loading
- Data Cleaning
- Data Analysis Using Descriptive Statistics
- Data Visualization

Data Loading

Loading Data

Data Loading

```
#Import and data loading
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv('tabular.educ_enrollment_by_year_schools_2017-18.csv')

pd.options.display.max_columns = 60

df
```

	seq_id	schid	name	municipal	schoolyear	enrolled	grade_pk	grade_k	grade_1	grade_2	grade_3	grade_4	grade_5	grade_6	grade_7	grad
0	22265	4450105	Abby Kelley Foster Charter Public School	Worcester	2017-18	1425	0	117	116	113	117	122	118	115	123	
1	22266	10001	Abington Early Education Program	Abington	2017-18	84	84	0	0	0	0	0	0	0	0	
2	22267	10505	Abington High	Abington	2017-18	520	0	0	0	0	0	0	0	0	0	
3	22268	10405	Abington Middle School	Abington	2017-18	687	0	0	0	0	0	0	154	188	178	
4	22269	10020	Beaver Brook Elementary	Abington	2017-18	424	0	137	155	132	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1845	24110	3480605	Worcester Technical High	Worcester	2017-18	1389	0	0	0	0	0	0	0	0	0	
1846	24111	3490010	R. H. Conwell	Worthington	2017-18	62	14	13	9	9	7	5	3	2	0	
1847	24112	3500010	Charles E Roderick	Wrentham	2017-18	466	0	0	0	0	0	161	149	156	0	
1848	24113	3500003	Delaney	Wrentham	2017-18	580	83	130	122	113	132	0	0	0	0	
1849	24114	0	State Totals	State Totals	2017-18	954034	30522	66014	68039	68249	70066	72164	72487	71262	70928	71

1850 rows × 55 columns

We first imported pandas as pd in order to use the **read_csv()** function to load the data in as a dataframe. We also have to make sure the directory to the ipynb file is correct.

Additionally, we **imported numpy and matplotlib** to be used for the other techniques later on.

Data Cleaning

Data Cleaning

Data Loading

```
2]: data = pd.read_csv("Enrollment by Year School.csv")
    data
```

2]:

	seq_id	schid	name	municipal	schoolyear	enrolled	grade_pk	grade_k	grade_1	grade_2	grade_3	grade_4	grade_5	grade_6	grade_7	grac
0	22265	4450105	Abby Kelley Foster Charter Public School	Worcester	2017-18	1425	0	117	116	113	117	122	118	115	123	
			Abington													

	Sequence ID	School ID	School Name	Municipal	School Year	Total Enrolled	Pre-Kindergarden Enrollment	Kindergarden Enrollment	1st Grade Enrollment	2nd Grade Enrollment	3rd Grade Enrollment	4th Grade Enrollment	5th Grade Enrollment	I
0	22265	4450105	Abby Kelley Foster Charter Public School	Worcester	2017-18	1425	0	117	116	113	117	122	118	
1	22266	10001	Abington Early Education Program	Abington	2017-18	84	84	0	0	0	0	0	0	
2	22267	10505	Abington	Abington	2017-	520	0	0	0	0	0	0	0	

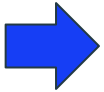
Using...

df.rename(columns = {...})

...function, we changed all column names to be more recognizable, as there are some with names that are not clear in what they are unless we compare the data set from the original resource website.

Data Cleaning

Students with Disabilities Enrollment	%Students with Disabilities Enrollment	Low-income Enrollment	%Low-income Enrollment
171	12.0	NaN	NaN
32	38.1	NaN	NaN
51	9.8	NaN	NaN
92	13.4	NaN	NaN
62	14.6	NaN	NaN
...	...	...	...
158	11.4	NaN	NaN
15	24.2	NaN	NaN
62	13.3	NaN	NaN
93	16.0	NaN	NaN
171061	17.7	NaN	NaN

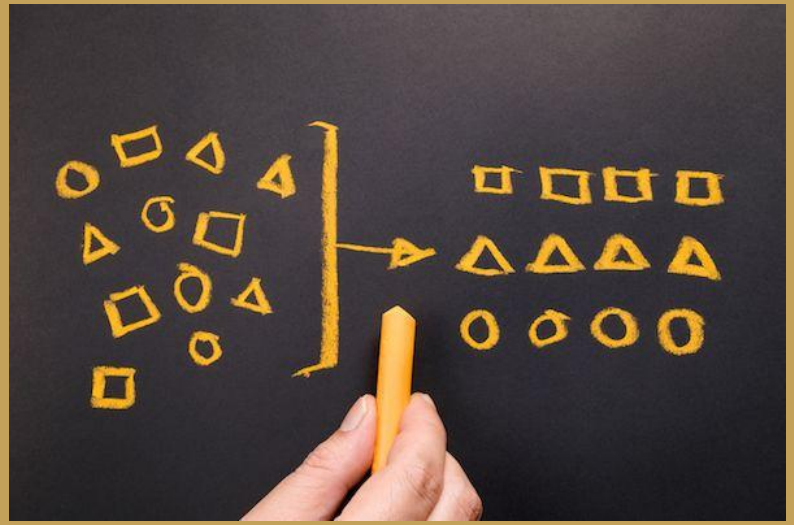


Removing Unnecessary Data

```
#Drop these columns as they are useless. Most of these are empty and were
#never recorded. Percentage is pointless as the important values are mostly
#recorded as numbers. 'ed_num' and 'ed_pct' are dropped because we have no
#information on what they represents, so removing it will avoid confusion.
#This is why the values of 'Enrolled' must be updated with the difference
#of 'Enrolled' and 'ed_num'.
df.drop(['Seq ID',
        '%African American or Black Enrollment',
        '%Asian Enrollment',
        '%Hispanic or Latino Enrollment',
        '%White Enrollment',
        '%Native American Enrollment',
        '%Native Hawaiian or Other Pacific Islander Enrollment',
        '%Multi-Race Non-Hispanic Enrollment',
        '%Male Enrollment',
        '%Female Enrollment',
        '%Limited English Proficient/First Language Not English Enrollment',
        '%First Language not English Enrollment',
        '%Students with Disabilities Enrollment',
        'Low-income Enrollment',
        '%Low-income Enrollment',
        'Student Eligible for Free Lunch Enrollment',
        '%Student Eligible for Free Lunch Enrollment',
        'Student Eligible for Reduced Price Lunch Enrollment',
        '%Student Eligible for Reduced Price Lunch Enrollment',
        '%High Needs Enrollment',
        'ed_num',
        'ed_pct'], axis = 1, inplace = True)
df
```

Using **df.drop()** function, we remove unwanted columns and data due to them not attributing to our study or that they were columns with missing data, such as the percentage and columns with NaN.

Organizing



Sorting, Set Index, and Drop

Sorting the Dataset

```
#Set the index to be on 'School Name' and sort the values of 'School Name' in
#ascending order.
df.set_index('School Name', inplace = True)
df.sort_values('School Name', inplace = True)
df
```

School Name	School ID	Municipal	School Year	Enrolled	Enrollment in Pre-Kindergarden	Enrollment in Kindergarden	Enrollment in Grade 1	Enrollment in Grade 2
1 LT Charles W. Whitcomb School	1700045	Marlborough	2017-18	1308	0	0	0	0
21st Century Skills Academy	3320515	West Springfield	2017-18	8	0	0	0	0
A Drewicz Elementary	1630016	Lynn	2017-18	492	0	71	68	74
A E Angier	2070005	Newton	2017-18	467	0	65	92	78
A F Maloney	6220015	Blackstone	2017-18	299	0	0	0	0
...	...	...	...	...	...	...	...	...
Worcester East Middle	3480420	Worcester	2017-18	821	0	0	0	0
Worcester Technical High	3480605	Worcester	2017-18	1389	0	0	0	0
Wyman	3470060	Woburn	2017-18	182	0	28	29	34
Young Achievers	350380	Boston	2017-18	565	44	54	61	60
Zervas	2070130	Newton	2017-18	407	0	58	84	64

1850 rows × 32 columns

Making a different dataframe in order to derive stats

Removed the Final Total from df1 but not from the original dataframe

```
df1 = df
df1 = df1.drop('State Totals')
df1
```

We sort the data by “School Name” in ascending order, while making it the index, so that the data is more readable.

And before moving to the Data Analysis Using Descriptive Statistics, we had to remove the row “State Totals”, as it carries the sum of each columns and would therefore get in the way our project objective.

Data Analysis Using Descriptive Statistics

Mean Formula

$$\text{Mean} = \frac{\text{Sum of All Data Points}}{\text{Number of Data Points}}$$

Descriptive Statistics

N	Minimum	Maximum	Mean	Std. Deviation
431	59.83	101.95	82.7265	6.82982
435	55.11	103.62	82.0394	7.63745
435	35.32	93.78	65.4512	8.29165
435	64.06	93.01	79.5392	5.50151
431				

Descriptive Statistics

Data Analysis Using Descriptive Statistics

Total Enrolled

In [9]:
```
mean = Data1['Total Enrolled'].mean()
median = Data1['Total Enrolled'].median()
dmax = Data1['Total Enrolled'].max()
dmin = Data1['Total Enrolled'].min()
std = Data1['Total Enrolled'].std()
print('Mean: ', mean)
print('Median: ', median)
print('Max: ', dmax)
print('Min: ', dmin)
print('Std: ', std)
```

```
Mean:  515.9724175229854
Median:  436.0
Max:  4123
Min:  1
Std:  369.95465352846537
```

In this first example, we find the mean, median, max, min, and standard deviation for the column “Enrolled”. We also do the same for the rest of the other columns. What’s important for the “Enrolled” is that it answered our project objective. Out of all schools, which one has the highest and lowest enrollment, and what is the overall average?

Descriptive Statistics

```
#The school(s) with the lowest enrollment.

df1[df1['Enrolled'] == 1]
```

School Name	School ID	Municipal	School Year	Enrolled	Enrollm Kinderg
Fall River Gateway to College @ BCC	950515	Fall River	2017-18	1	
Landmark School	300920	Rockport	2017-18	1	
Solstice School	2540810	Rockport	2017-18	1	

3 rows × 32 columns

```
#The school(s) with the highest enrollment.

df1[df1['Enrolled'] == 4123]
```

School Name	School ID	Municipal	School Year	Enrolled	Enrollment Pr Kindergard
Brockton High	440505	Brockton	2017-18	4123	

1 rows × 32 columns

Data Visualization

Data Visualization

Data Visualization

Number of Students Enrolled vs School Names by a Municipal

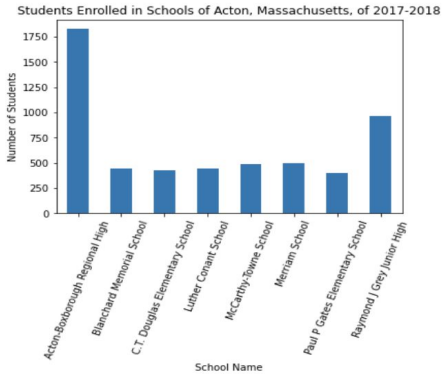
```
#Compare all schools' enrollment that are from Acton.

df2 = df1[df1['Municipal'] == 'Acton']['Enrolled']
df2
```

```
School Name
Acton-Boxborough Regional High    1827
Blanchard Memorial School          447
C.T. Douglas Elementary School     427
Luther Conant School               442
McCarthy-Towne School              487
Merriam School                     498
Paul P Gates Elementary School     402
Raymond J Grey Junior High         964
Name: Enrolled, dtype: int64
```

```
df2.plot.bar(title = "Students Enrolled in Schools of Acton, Massachusetts, of 2017-2018", rot = 70)
plt.ylabel("Number of Students")
```

```
Text(0, 0.5, 'Number of Students')
```



Having nearly 2,000 rows of schools, we decided the most optimal way of utilizing Data Visualization technique is to compare each schools that are from a specific municipal/city of how many students they have enrolled in theirs. We decided to go with Acton.

Conclusion

We can conclude that our objective was meant in answering which school has the highest and lowest enrollment, along with what was the average enrollment. However, we did not expect our result to have more than one school of lowest enrollment, nor did we expect to have no school with absolute 0 enrollment. It also seems that the schools who teach a higher grade such as High Schools tend to have a much higher enrollment rate since there are more subjects/classes to be taught and attended

Reference

Enrollment by School Year (Schools) Data

https://datacommon.mapc.org/browser/datasets/321

Now Let’s Run the Code