

## Calculating percent proficiency and its percent change in New York State education assessment data

After downloading the [2015, 2016 and 2017 data from the New York State Education Department \(https://data.nysed.gov/downloads.php\)](https://data.nysed.gov/downloads.php) and renaming the files to 3-8\_ELA\_MATH\_{{ year }}.csv, this program calculates the percent proficient, scoring at Level 3 or 4, of each school. The percent proficient is calculated by adding each raw proficient count together for available grades and then dividing the total test takers. The 2013 and 2014 files were first converted from .mdb file format.

```
In [1]: import agate
import csv
```

Due to missing school values and changes in data columns, I needed two different typetesters to force the columns into their correct formatting.

```

In [13]: tester_15 = agate.TypeTester(limit=100,force={
    'NRC_DESC': agate.Text(),
    'NRC_CODE': agate.Text(),
    'BEDSCODE': agate.Text(),
    'L1_COUNT': agate.Text(),
    'L1_PCT': agate.Text(),
    'L2_COUNT': agate.Text(),
    'L2_PCT': agate.Text(),
    'L3_COUNT': agate.Text(),
    'L3_PCT': agate.Text(),
    'L4_COUNT': agate.Text(),
    'L4_PCT': agate.Text(),
    'L2-L4_PCT': agate.Text(),
    'L3-L4_PCT': agate.Text(),
    'MEAN_SCALE_SCORE': agate.Text(),
    'COUNTY_DESC': agate.Text(),
    'TOTAL_TESTED': agate.Text(),
    'COUNTY_CODE': agate.Text(),
    'Sum_Of_SCALE_SCORE': agate.Text()
})
tester_16 = agate.TypeTester(limit=100,force={
    'NRC_DESC': agate.Text(),
    'NRC_CODE': agate.Text(),
    'BEDSCODE': agate.Text(),
    'L1_COUNT': agate.Text(),
    'L1_PCT': agate.Text(),
    'L2_COUNT': agate.Text(),
    'L2_PCT': agate.Text(),
    'L3_COUNT': agate.Text(),
    'L3_PCT': agate.Text(),
    'L4_COUNT': agate.Text(),
    'L4_PCT': agate.Text(),
    'L2-L4_PCT': agate.Text(),
    'L3-L4_PCT': agate.Text(),
    'MEAN_SCALE_SCORE': agate.Text(),
    'COUNTY_DESC': agate.Text(),
    'TOTAL_TESTED': agate.Text(),
    'COUNTY_CODE': agate.Text(),
    'SY_END_DATE': agate.Text(),
    'SUBGROUP_CODE': agate.Text()
})
tester_12 = agate.TypeTester(limit=100,force={
    'BEDSCODE': agate.Text(),
    'L1_COUNT': agate.Text(),
    'L2_COUNT': agate.Text(),
    'L3_COUNT': agate.Text(),
    'L4_COUNT': agate.Text(),
    'MEAN_SCALE_SCORE': agate.Text(),
    'COUNTY_DESC': agate.Text(),
    'TOTAL_TESTED': agate.Text(),
    'COUNTY_CODE': agate.Text()
})

```

This program takes in the csv of a specific year's assesment data and its tester and returns a dictionary of BEDS (state id codes) and test data.

```

In [14]: def get_percent_dictionary(file_name, tester):
    schools = agate.Table.from_csv(file_name, column_types=tester)
    #Grab just Erie and Niagara schools, filtering out available subgroups and tota
    l county stats
    erie_niagara = schools.where(lambda row: row['COUNTY_DESC'] in ['ERIE', 'NIAGAR
A']).where(lambda row: row['SUBGROUP_NAME'] in ['All Students'])
    print(len(erie_niagara.rows))
    #Data doesn't include district information, but the first six digits of a schoo
l's BEDS codes includes that info.
    beds_code = {}
    for row in erie_niagara.rows:
        if 'SCHOOL DISTRICT' in row['NAME']:
            beds_district = row['BEDSCODE'][:6]
            #Creates a lookup table of ids for the district name
            beds_code[str(beds_district)] = row['NAME']
    def calculate_prof_counts(row):
        #Some schools don't have test takers for certain tests, so fill in with zer
os.
        try:
            l3_counts = int(row['L3_COUNT'])
        except ValueError:
            l3_counts = 0
        try:
            l4_counts = int(row['L4_COUNT'])
        except ValueError:
            l4_counts = 0
        proficient = l3_counts + l4_counts
        return proficient
    erie_calcs = erie_niagara.compute([
        ('proficient_count', agate.Formula(agate.Number(), lambda row: calculate_pr
of_counts(row)))
    ])
    # Form the data structure for each school with beds code as a key
    schools_info = {}
    for row in erie_calcs.rows:
        if row['NAME'] == 'NIAGARA CHARTER SCHOOL':
            district = 'NIAGARA-WHEATFIELD CENTRAL SCHOOL DISTRICT'
        elif row['NAME'] == 'ERIE COUNTY' or row['NAME'] == 'NIAGARA COUNTY':
            district = row['NAME']
        else:
            beds_district = row['BEDSCODE'][:6]
            district = beds_code[beds_district]
            """name = School name
            math/ela dictionary has totals = total test takers one per applicable clas
s; proficient = proficient_count values; classes = grades taking the test
            district = district name (looked up using the beds_code dict)
            county = county"""
            schools_info[row['BEDSCODE']] = {'name': row['NAME'], 'math': {'totals':
[], 'proficient': [], 'classes': []}, 'ela': {'totals': [], 'proficient': [], 'clas
ses': []}, 'district': district, 'county': row['COUNTY_DESC']}
    # After creating the correct dict format, now I run through the schools again t
o fill the score values.
    for row in erie_calcs.rows:
        if row['ITEM_SUBJECT_AREA'] == 'ELA':
            data_dict = schools_info[row['BEDSCODE']]['ela']
        else:
            data_dict = schools_info[row['BEDSCODE']]['math']
        try:
            data_dict['totals'].append(int(row['TOTAL_TESTED']))
        except ValueError:
            data_dict['totals'].append(0)
        try:
            data_dict['classes'].append(row['ITEM_DESC'])
        except ValueError:

```

```
In [15]: year_16 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2016.csv', tester_16)
1861

In [9]: year_17 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2017.csv', tester_16)
1839

In [16]: year_15 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2015.csv', tester_15)
1876

In [17]: year_14 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2014.csv', tester_15)
1883

In [18]: year_13 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2013.csv', tester_15)
1912

In [19]: year_12 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2012.csv', tester_12)
1820

In [20]: year_11 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2011.csv', tester_12)
1786

In [21]: year_10 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2010.csv', tester_12)
1780

In [22]: year_09 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2009.csv', tester_12)
1776

In [23]: year_08 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2008.csv', tester_12)
1772

In [24]: year_07 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2007.csv', tester_12)
1726

In [25]: year_06 = get_percent_dictionary('raw/3-8_ELA_AND_MATH_2006.csv', tester_12)
1722
```

Example of each year returning data for a specific school

```
In [26]: print(year_16['140000000000'])
print('$$$$$$$$$$$$$$$$$$$$')
print(year_12['140000000000'])
```

```
{'math': {'total_percent': 35.0, 'proficient': [2821, 2687, 2365, 2122, 1589, 70
8], 'classes': ['Grade 3 Math', 'Grade 4 Math', 'Grade 5 Math', 'Grade 6 Math',
'Grade 7 Math', 'Grade 8 Math'], 'totals': [6995, 6509, 6211, 5829, 5356, 420
7]}, 'ela': {'total_percent': 33.9, 'proficient': [2740, 2345, 1865, 1962, 1737,
1904], 'classes': ['Grade 3 ELA', 'Grade 4 ELA', 'Grade 5 ELA', 'Grade 6 ELA', '
Grade 7 ELA', 'Grade 8 ELA'], 'totals': [7026, 6577, 6329, 6070, 5706, 5368]}, '
name': 'ERIE COUNTY', 'district': 'ERIE COUNTY', 'county': 'ERIE'}
$$$$$$$$$$$$$$$$$$$$
{'math': {'total_percent': 63.7, 'proficient': [5674, 6490, 6191, 6421, 6413, 60
59], 'classes': ['Grade 3 Math', 'Grade 4 Math', 'Grade 5 Math', 'Grade 6 Math',
'Grade 7 Math', 'Grade 8 Math'], 'totals': [9460, 9637, 9805, 9914, 9855, 976
3]}, 'ela': {'total_percent': 55.8, 'proficient': [5281, 5629, 5515, 5765, 5114,
5127], 'classes': ['Grade 3 ELA', 'Grade 4 ELA', 'Grade 5 ELA', 'Grade 6 ELA', '
Grade 7 ELA', 'Grade 8 ELA'], 'totals': [9403, 9583, 9738, 9862, 9798, 9725]}, '
name': 'ERIE COUNTY', 'district': 'ERIE COUNTY', 'county': 'ERIE'}
```

```
In [27]: print(year_16['400400010001'])
print(year_15['400400010001'])
print(year_17['400400010001'])
print('****')
print(year_17['140101060007'])
```

```
{'math': {'total_percent': 41.3, 'proficient': [29, 21], 'classes': ['Grade 3 Ma
th', 'Grade 4 Math'], 'totals': [67, 54]}, 'ela': {'total_percent': 25.2, 'profi
cient': [22, 11], 'classes': ['Grade 3 ELA', 'Grade 4 ELA'], 'totals': [69, 6
2]}, 'name': 'ANNA MERRITT ELEMENTARY SCHOOL', 'district': 'LOCKPORT CITY SCHOOL
DISTRICT', 'county': 'NIAGARA'}
{'math': {'total_percent': 37.0, 'proficient': [15, 29], 'classes': ['Grade 3 Ma
th', 'Grade 4 Math'], 'totals': [57, 62]}, 'ela': {'total_percent': 17.8, 'profi
cient': [11, 12], 'classes': ['Grade 3 ELA', 'Grade 4 ELA'], 'totals': [60, 6
9]}, 'name': 'ANNA MERRITT ELEMENTARY SCHOOL', 'district': 'LOCKPORT CITY SCHOOL
DISTRICT', 'county': 'NIAGARA'}
{'math': {'total_percent': 36.8, 'proficient': [23, 19], 'classes': ['Grade 3 Ma
th', 'Grade 4 Math'], 'totals': [52, 62]}, 'ela': {'total_percent': 22.9, 'profi
cient': [11, 16], 'classes': ['Grade 3 ELA', 'Grade 4 ELA'], 'totals': [53, 6
5]}, 'name': 'ANNA MERRITT ELEMENTARY SCHOOL', 'district': 'LOCKPORT CITY SCHOOL
DISTRICT', 'county': 'NIAGARA'}
****
{'math': {'total_percent': 57.3, 'proficient': [62, 40, 28], 'classes': ['Grade
3 Math', 'Grade 4 Math', 'Grade 5 Math'], 'totals': [81, 73, 73]}, 'ela': {'tota
l_percent': 53.7, 'proficient': [54, 41, 21], 'classes': ['Grade 3 ELA', 'Grade
4 ELA', 'Grade 5 ELA'], 'totals': [77, 69, 70]}, 'name': 'ALDEN INTERMEDIATE SCH
OOL', 'district': 'ALDEN CENTRAL SCHOOL DISTRICT', 'county': 'ERIE'}
```

Backfilling new 2017 schools

[illegible]

not in 2007 GASKILL PREPARATORY SCHOOL  
not in 2006 GASKILL PREPARATORY SCHOOL  
not in 2006 ELMWOOD VILLAGE CHARTER SCHOOL  
not in 2012 HOLLAND JUNIOR/SENIOR HIGH SCHOOL  
not in 2011 HOLLAND JUNIOR/SENIOR HIGH SCHOOL  
not in 2010 HOLLAND JUNIOR/SENIOR HIGH SCHOOL  
not in 2009 HOLLAND JUNIOR/SENIOR HIGH SCHOOL  
not in 2008 HOLLAND JUNIOR/SENIOR HIGH SCHOOL  
not in 2007 HOLLAND JUNIOR/SENIOR HIGH SCHOOL  
not in 2006 HOLLAND JUNIOR/SENIOR HIGH SCHOOL  
not in 2010 TONAWANDA MIDDLE/HIGH SCHOOL  
not in 2009 TONAWANDA MIDDLE/HIGH SCHOOL  
not in 2008 TONAWANDA MIDDLE/HIGH SCHOOL  
not in 2007 TONAWANDA MIDDLE/HIGH SCHOOL  
not in 2006 TONAWANDA MIDDLE/HIGH SCHOOL  
not in 2011 ALTERNATIVE HIGH SCHOOL AT 4  
not in 2010 ALTERNATIVE HIGH SCHOOL AT 4  
not in 2009 ALTERNATIVE HIGH SCHOOL AT 4  
not in 2008 ALTERNATIVE HIGH SCHOOL AT 4  
not in 2007 ALTERNATIVE HIGH SCHOOL AT 4  
not in 2006 ALTERNATIVE HIGH SCHOOL AT 4  
not in 2007 LASALLE PREPARATORY SCHOOL  
not in 2006 LASALLE PREPARATORY SCHOOL  
not in 2016 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2015 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2014 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2013 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2012 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2011 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2010 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2009 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2008 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2007 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2006 KENMORE EAST SENIOR HIGH SCHOOL  
not in 2011 PINE HILL EDUCATION CENTER  
not in 2010 PINE HILL EDUCATION CENTER  
not in 2009 PINE HILL EDUCATION CENTER  
not in 2008 PINE HILL EDUCATION CENTER  
not in 2007 PINE HILL EDUCATION CENTER  
not in 2006 PINE HILL EDUCATION CENTER  
not in 2006 MATH SCIENCE TECHNOLOGY PREPARATORY SCHOOL AT SENECA (THE)  
not in 2006 NIAGARA CHARTER SCHOOL  
not in 2007 CATARACT ELEMENTARY SCHOOL  
not in 2006 CATARACT ELEMENTARY SCHOOL  
not in 2009 PARKDALE ELEMENTARY SCHOOL  
not in 2008 PARKDALE ELEMENTARY SCHOOL  
not in 2007 PARKDALE ELEMENTARY SCHOOL  
not in 2006 PARKDALE ELEMENTARY SCHOOL  
not in 2016 CHARTER SCHOOL OF INQUIRY  
not in 2015 CHARTER SCHOOL OF INQUIRY  
not in 2014 CHARTER SCHOOL OF INQUIRY  
not in 2013 CHARTER SCHOOL OF INQUIRY  
not in 2012 CHARTER SCHOOL OF INQUIRY  
not in 2011 CHARTER SCHOOL OF INQUIRY  
not in 2010 CHARTER SCHOOL OF INQUIRY  
not in 2009 CHARTER SCHOOL OF INQUIRY  
not in 2008 CHARTER SCHOOL OF INQUIRY  
not in 2007 CHARTER SCHOOL OF INQUIRY  
not in 2006 CHARTER SCHOOL OF INQUIRY  
not in 2016 WESTERN NEW YORK MARITIME CHARTER SCHOOL  
not in 2015 WESTERN NEW YORK MARITIME CHARTER SCHOOL  
not in 2014 WESTERN NEW YORK MARITIME CHARTER SCHOOL  
not in 2013 WESTERN NEW YORK MARITIME CHARTER SCHOOL  
not in 2012 WESTERN NEW YORK MARITIME CHARTER SCHOOL

Grab clean school/district names and which schools are charters

```
In [29]: clean_names = {}
         charters = []
         with open('../school_name_dictionary.csv') as csvfile:
             reader = csv.DictReader(csvfile)
             for row in reader:
                 clean_names[row['BEDS']] = row['clean_school']
                 if len(row['charter']) == 1:
                     charters.append(row['BEDS'])
         clean_names['140101060007']
```

```
/Users/lheinle/anaconda/envs/py35/lib/python3.5/site-packages/IPython/core/formatters.py:92: DeprecationWarning: DisplayFormatter._ipython_display_formatter_default is deprecated: use @default decorator instead.
/Users/lheinle/anaconda/envs/py35/lib/python3.5/site-packages/IPython/core/formatters.py:98: DeprecationWarning: DisplayFormatter._formatters_default is deprecated: use @default decorator instead.
/Users/lheinle/anaconda/envs/py35/lib/python3.5/site-packages/IPython/core/formatters.py:677: DeprecationWarning: PlainTextFormatter._deferred_printers_default is deprecated: use @default decorator instead.
/Users/lheinle/anaconda/envs/py35/lib/python3.5/site-packages/IPython/core/formatters.py:669: DeprecationWarning: PlainTextFormatter._singleton_printers_default is deprecated: use @default decorator instead.
/Users/lheinle/anaconda/envs/py35/lib/python3.5/site-packages/IPython/core/formatters.py:672: DeprecationWarning: PlainTextFormatter._type_printers_default is deprecated: use @default decorator instead.
/Users/lheinle/anaconda/envs/py35/lib/python3.5/site-packages/IPython/core/formatters.py:669: DeprecationWarning: PlainTextFormatter._singleton_printers_default is deprecated: use @default decorator instead.
/Users/lheinle/anaconda/envs/py35/lib/python3.5/site-packages/IPython/core/formatters.py:672: DeprecationWarning: PlainTextFormatter._type_printers_default is deprecated: use @default decorator instead.
/Users/lheinle/anaconda/envs/py35/lib/python3.5/site-packages/IPython/core/formatters.py:677: DeprecationWarning: PlainTextFormatter._deferred_printers_default is deprecated: use @default decorator instead.
```

```
Out[29]: 'Alden Intermediate'
```

Separate schools/district into county lists for print, and create dictionary with included schools and codes.



```
In [30]: erie_list = {}
niagara_list = {}
charter_list = {}
for key, value in year_17.items():
    if key not in charters:
        if 'SCHOOL DISTRICT' in value['name']:
            if value['county'] == 'ERIE':
                erie_list[key] = {'name': clean_names[key], 'schools': {}, 'dist_key': value['name']}
            else:
                niagara_list[key] = {'name': clean_names[key], 'schools': {}, 'dist_key': value['name']}
        else:
            charter_list[key] = clean_names[key]
for distict, value in erie_list.items():
    for school, info in year_17.items():
        if school not in charters:
            if value['dist_key'] == info['district'] and 'SCHOOL DISTRICT' not in info['name']:
                erie_list[distict]['schools'][school] = clean_names[school]
for distict, value in niagara_list.items():
    for school, info in year_17.items():
        if school not in charters:
            if value['dist_key'] == info['district'] and 'SCHOOL DISTRICT' not in info['name']:
                niagara_list[distict]['schools'][school] = clean_names[school]
print(niagara_list)
```

```
{'400800010000': {'schools': {'400800010022': 'Niagara Street', '400800010041': 'LaSalle Preparatory', '400800010021': 'Maple Avenue', '400800010020': 'Hyde Park', '400800010031': 'Abate Elementary', '400800010042': 'Cataract Elementary', '400800010040': 'Gaskill Preparatory', '400800010012': 'Mann', '400800010015': 'K alfas Magnet', '400800010010': 'Seventy Ninth Street'}, 'name': 'NIAGARA FALLS SCHOOL DISTRICT', 'dist_key': 'NIAGARA FALLS CITY SCHOOL DISTRICT'}, '400701060000': {'schools': {'400701060005': 'Errick Road Elementary', '400701060002': 'West Street Elementary', '400701060004': 'Colonial Village Elem.', '400701060009': 'Edward Town Middle', '400701060003': 'Tuscarora Elementary'}, 'name': 'NIAGARA WHEATFIELD SCH. DIST.', 'dist_key': 'NIAGARA-WHEATFIELD CENTRAL SCHOOL DISTRICT'}, '400400010000': {'schools': {'400400010009': 'Belknap Intermediate', '400400010002': 'Upson Elementary', '400400010010': 'North Park Junior High', '400400010001': 'Merritt Elementary', '400400010007': 'Kelley Elementary', '400400010005': 'Southard Elementary'}, 'name': 'LOCKPORT SCHOOL DISTRICT', 'dist_key': 'LOCKPORT CITY SCHOOL DISTRICT'}, '400601060000': {'schools': {'400601060008': 'Newfane Middle', '400601060002': 'Newfane Elementary'}, 'name': 'NEWFANE SCHOOL DISTRICT', 'dist_key': 'NEWFANE CENTRAL SCHOOL DISTRICT'}, '400900010000': {'schools': {'400900010003': 'Drake', '400900010007': 'Meadow', '400900010012': 'North Tonawanda Middle', '400900010008': 'Ohio Elementary', '400900010009': 'Spruce'}, 'name': 'NORTH TONAWANDA SCH. DIST.', 'dist_key': 'NORTH TONAWANDA CITY SCHOOL DISTRICT'}, '401501060000': {'schools': {'401501060003': 'Wilson High', '401501060002': 'Wilson Elementary'}, 'name': 'WILSON SCHOOL DISTRICT', 'dist_key': 'WILSON CENTRAL SCHOOL DISTRICT'}, '401201060000': {'schools': {'401201060001': 'Royalton-Hartland Elem.', '401201060004': 'Royalton-Hartland Middle'}, 'name': 'ROY-HART SCHOOL DISTRICT', 'dist_key': 'ROYALTON-HARTLAND CENTRAL SCHOOL DISTRICT'}, '400301060000': {'schools': {'400301060003': 'Intermediate Educ. Center', '400301060002': 'Primary Education Center', '400301060005': 'Lewiston Porter Middle'}, 'name': 'LEW-PORT SCHOOL DISTRICT', 'dist_key': 'LEWISTON-PORTER CENTRAL SCHOOL DISTRICT'}, '401301040000': {'schools': {'401301040002': 'Pratt Elementary', '401301040003': 'Barker Junior/Senior High'}, 'name': 'BARKER SCHOOL DISTRICT', 'dist_key': 'BARKER CENTRAL SCHOOL DISTRICT'}, '401001060000': {'schools': {'401001060002': 'Regan Intermediate', '401001060004': 'Starpoint Middle'}, 'name': 'STARPOINT SCHOOL DISTRICT', 'dist_key': 'STARPOINT CENTRAL SCHOOL DISTRICT'}}
```

```
In [31]: def percent_change(new,old):  
    try:  
        calculate = (new-old)/old  
        percent = calculate * 100  
        one_decimal = float("{0:.1f}".format(percent))  
        return one_decimal  
    except (ZeroDivisionError,TypeError):  
        return '-'
```

Exporting for print

```

In [23]: erie_districts = 0
niagara_districts = 0
charter_districts = 0
def export_county_schools(county_list, county):
    print('Begging {0}'.format(county))
    county_count = 0
    if county != 'charter':
        #Sort the county of district id's by its name value.
        sorted_county = sorted(county_list, key= lambda district: county_list[district]['name'])
        for district in sorted_county:
            county_count += 1
            #print('*** starting district {0}'.format(county_list[district]['name']))

            pc_ela_15_17 = percent_change(year_17[district]['ela']['total_percent'],year_15[district]['ela']['total_percent'])
            pc_ela_16_17 = percent_change(year_17[district]['ela']['total_percent'],year_16[district]['ela']['total_percent'])
            pc_math_15_17 = percent_change(year_17[district]['math']['total_percent'],year_15[district]['math']['total_percent'])
            pc_math_16_17 = percent_change(year_17[district]['math']['total_percent'],year_16[district]['math']['total_percent'])
            district_info = [county_list[district]['name'], clean_names[district], year_15[district]['ela']['total_percent'],year_16[district]['ela']['total_percent'],year_17[district]['ela']['total_percent'],year_15[district]['math']['total_percent'],year_16[district]['math']['total_percent'],year_17[district]['math']['total_percent'], pc_ela_15_17, pc_ela_16_17, pc_math_15_17, pc_math_16_17]
            writer.writerow(district_info)
            schools_info = county_list[district]['schools']
            sorted_schools = sorted(schools_info, key = lambda school: schools_info[school])

            #Now go through each school in the district and calculate its percent change.

            for school in sorted_schools:
                pc_school_ela_15_17 = percent_change(year_17[school]['ela']['total_percent'],year_15[school]['ela']['total_percent'])
                pc_school_ela_16_17 = percent_change(year_17[school]['ela']['total_percent'],year_16[school]['ela']['total_percent'])
                pc_school_math_15_17 = percent_change(year_17[school]['math']['total_percent'],year_15[school]['math']['total_percent'])
                pc_school_math_16_17 = percent_change(year_17[school]['math']['total_percent'],year_16[school]['math']['total_percent'])
                school_info = [county_list[district]['name'], clean_names[school], year_15[school]['ela']['total_percent'],year_16[school]['ela']['total_percent'],year_17[school]['ela']['total_percent'],year_15[school]['math']['total_percent'],year_16[school]['math']['total_percent'],year_17[school]['math']['total_percent'], pc_school_ela_15_17, pc_school_ela_16_17, pc_school_math_15_17, pc_school_math_16_17]

                writer.writerow(school_info)

            else:
                #Charter schools only have one level aka direct to school data.
                sorted_county = sorted(county_list, key= lambda district: county_list[district])

                for district in sorted_county:
                    county_count += 1
                    #print('*** starting district {0}'.format(county_list[district]))
                    pc_ela_15_17 = percent_change(year_17[district]['ela']['total_percent'],year_15[district]['ela']['total_percent'])
                    pc_ela_16_17 = percent_change(year_17[district]['ela']['total_percent'],year_16[district]['ela']['total_percent'])
                    pc_math_15_17 = percent_change(year_17[district]['math']['total_percent'],year_15[district]['math']['total_percent'])
                    pc_math_16_17 = percent_change(year_17[district]['math']['total_percent'],year_16[district]['math']['total_percent'])

```

```
Begging erie  
Begging niagara  
Begging charter  
28 erie districts and 10 niagara districts 15 charters
```

Online needs JSON in the clean\_json groups.

```
In [32]: print(erie_list['140201060000'])  
  
{'schools': {'140201060001': 'Amherst Middle', '140201060006': 'Windermere', '14  
0201060005': 'Smallwood'}, 'name': 'AMHERST', 'dist_key': 'AMHERST CENTRAL SCHOO  
L DISTRICT'}
```

```

In [33]: erie_districts = 0
niagara_districts = 0
clean_json = {'Erie': {}, 'Niagara': {}, 'Buffalo': {}, 'Charters': {}}
def export_county_schools(county_list, county):
    ordered = []
    county_count = 0
    #Sort the county of district id's by its name value.
    if county != 'Charters':
        if county == 'Erie':
            ordered.append(['140000000000', clean_names['140000000000']])
        elif county == 'Niagara':
            ordered.append(['400000000000', clean_names['400000000000']])
        sorted_county = sorted(county_list, key= lambda district: county_list[district]['name'])
        for district in sorted_county:
            #Buffalo gets thrown in its own group
            if district != '140600010000':
                county_count += 1
                #print('*** starting district {0}'.format(county_list[district]['name']))

                schools_info = county_list[district]['schools']
                ordered.append([district,clean_names[district]])
                sorted_schools = sorted(schools_info, key = lambda school: schools_info[school])

                #Now go through each school in the district and calculate its percent change.
                for school in sorted_schools:
                    try:
                        #print('writing the following {0}'.format(schools_info[school]))

                        ordered.append([school,clean_names[school]])
                    except KeyError:
                        #Beds code isn't found in one of the years for this school
                        print('Missing school in 2013, 2014, 2015, 2016 and/or 2017')

                        print(schools_info[school])
                        print(school)
                else:
                    buffalo_list = []
                    buffalo_list.append([district,clean_names[district]])
                    schools_info = county_list[district]['schools']
                    sorted_schools = sorted(schools_info, key = lambda school: schools_info[school])

                    #Now go through each school in the district and calculate its percent change.
                    for school in sorted_schools:
                        try:
                            #print('writing the following {0}'.format(schools_info[school]))

                            buffalo_list.append([school,clean_names[school]])
                        except KeyError:
                            #Beds code isn't found in one of the years for this school
                            print('Missing school in 2013, 2014, 2015, 2016 and/or 2017')

                            print(schools_info[school])
                            print(school)
                        clean_json['Buffalo']['ordered_schools'] = buffalo_list
            else:
                ordered.append(['8686868686868', 'Charters Average'])
                sorted_county = sorted(county_list, key= lambda district: county_list[district])
                for district in sorted_county:
                    county_count += 1
                    print('*** starting district {0}'.format(county_list[district]))

```

```
*** starting district Bflo. Academy of Science
*** starting district Buffalo United
*** starting district Charter Sch. for App. Tech.
*** starting district Charter School of Inquiry
*** starting district Elmwood Village
*** starting district Enterprise
*** starting district Global Concepts
*** starting district Johnson
*** starting district King Center
*** starting district Niagara Charter
*** starting district South Buffalo
*** starting district Tapestry
*** starting district WNY Maritime Charter
*** starting district West Buffalo
*** starting district Westminster
{'ordered_schools': [['400000000000', 'Niagara County average'], ['401301040000', 'BARKER SCHOOL DISTRICT'], ['401301040003', 'Barker Junior/Senior High'], ['401301040002', 'Pratt Elementary'], ['400301060000', 'LEW-PORT SCHOOL DISTRICT'], ['400301060003', 'Intermediate Educ. Center'], ['400301060005', 'Lewiston Porter Middle'], ['400301060002', 'Primary Education Center'], ['400400010000', 'LOCKPORT SCHOOL DISTRICT'], ['400400010009', 'Belknap Intermediate'], ['400400010007', 'Kelley Elementary'], ['400400010001', 'Merritt Elementary'], ['400400010010', 'North Park Junior High'], ['400400010005', 'Southard Elementary'], ['400400010002', 'Upton Elementary'], ['400601060000', 'NEWFANE SCHOOL DISTRICT'], ['400601060002', 'Newfane Elementary'], ['400601060008', 'Newfane Middle'], ['400800010000', 'NIAGARA FALLS SCHOOL DISTRICT'], ['400800010031', 'Abate Elementary'], ['400800010042', 'Cataract Elementary'], ['400800010040', 'Gaskill Preparatory'], ['400800010020', 'Hyde Park'], ['400800010015', 'Kalfas Magnet'], ['400800010041', 'LaSalle Preparatory'], ['400800010012', 'Mann'], ['400800010021', 'Maple Avenue'], ['400800010022', 'Niagara Street'], ['400800010010', 'Seventy Ninth Street'], ['400701060000', 'NIAGARA WHEATFIELD SCH. DIST.'], ['400701060004', 'Colonia 1 Village Elem.'], ['400701060009', 'Edward Town Middle'], ['400701060005', 'Errick Road Elementary'], ['400701060003', 'Tuscarora Elementary'], ['400701060002', 'West Street Elementary'], ['400900010000', 'NORTH TONAWANDA SCH. DIST.'], ['400900010003', 'Drake'], ['400900010007', 'Meadow'], ['400900010012', 'North Tonawanda Middle'], ['400900010008', 'Ohio Elementary'], ['400900010009', 'Spruce'], ['401201060000', 'ROY-HART SCHOOL DISTRICT'], ['401201060001', 'Royalton-Hartland Elem.'], ['401201060004', 'Royalton-Hartland Middle'], ['401001060000', 'STARPOINT SCHOOL DISTRICT'], ['401001060002', 'Regan Intermediate'], ['401001060004', 'Starpoint Middle'], ['401501060000', 'WILSON SCHOOL DISTRICT'], ['401501060002', 'Wilson Elementary'], ['401501060003', 'Wilson High']]]
```

```
In [34]: def year_build(school_dict, year_dict, year):
    if school[0] in year_dict:
        #Only output dictionary if the school has test scores for that year.
        if len(year_dict[school[0]]['math']['classes']) != 0 or len(year_dict[school[0]]['ela']['classes']) != 0:
            year_info = year_dict[school[0]]
            school_info['district'] = year_info['district']
            school_info['name'] = clean_names[school[0]]
            math_year = {}
            math_year['total'] = str(year_info['math']['total_percent'])
            for i, item in enumerate(year_info['math']['classes']):
                clean_class = year_info['math']['classes'][i][0:7]
                try:
                    percent_profficient = (year_info['math']['proficient'][i] / year_info['math']['totals'][i]) * 100
                    math_year[clean_class] = str(float("{0:.1f}".format(percent_profficient)))
                except ZeroDivisionError:
                    math_year[clean_class] = '-'
            school_dict['math'][year] = math_year
            ela_year = {}
            ela_year['total'] = str(year_info['ela']['total_percent'])
            for i, item in enumerate(year_info['ela']['classes']):
                clean_class = year_info['ela']['classes'][i][0:7]
                try:
                    percent_profficient = (year_info['ela']['proficient'][i] / year_info['ela']['totals'][i]) * 100
                    ela_year[clean_class] = str(float("{0:.1f}".format(percent_profficient)))
                except ZeroDivisionError:
                    ela_year[clean_class] = '-'
            school_dict['ela'][year] = ela_year
        else:
            print('Missing {0} - {2} in {1}'.format(school[0], year, year_17[school[0]]['name']))
```

```
In [35]: for group, value in clean_json.items():
print ('**Starting {0}'.format(group))
#print('values {0}'.format(value))
schools = {}
for school in value['ordered_schools']:
    school_info = {}
    school_info['math'] = {}
    school_info['ela'] = {}
    if school[0] == '8686868686868':
        #Charters
        print('hit charter average')
    else:
        year_build(school_info, year_17, '2017')
        year_build(school_info, year_16, '2016')
        year_build(school_info, year_15, '2015')
        year_build(school_info, year_14, '2014')
        year_build(school_info, year_13, '2013')
        year_build(school_info, year_12, '2012')
        year_build(school_info, year_11, '2011')
        year_build(school_info, year_10, '2010')
        year_build(school_info, year_09, '2009')
        year_build(school_info, year_08, '2008')
        year_build(school_info, year_07, '2007')
        year_build(school_info, year_06, '2006')
    schools[school[0]] = school_info
clean_json[group]['schools'] = schools
```

```
**Starting Buffalo
**Starting Erie
**Starting Charters
hit charter average
**Starting Niagara
```



```

In [36]: print(clean_json['Buffalo']['schools']['140600010000']['ela']['2017']['total'])
print(clean_json['Buffalo']['schools']['140600010000']['math']['2017']['total'])
print('*****')
print(clean_json['Buffalo']['schools']['140600010000']['ela']['2016']['total'])
print(clean_json['Buffalo']['schools']['140600010000']['math']['2016']['total'])
print('*****')
print('*****')
print(clean_json['Buffalo']['schools']['140600010000']['ela']['2015']['total'])
print(clean_json['Buffalo']['schools']['140600010000']['math']['2015']['total'])
print(clean_json['Buffalo']['schools']['140600010000']['ela']['2015'])
print('*****')
print('*****')
print(clean_json['Buffalo']['schools']['140600010000']['ela']['2014']['total'])
print(clean_json['Buffalo']['schools']['140600010000']['math']['2014']['total'])
print(clean_json['Buffalo']['schools']['140600010000']['ela']['2014'])
print('*****')
"""for school in clean_json['Buffalo']['schools']:
    print("{0}:{1}".format(clean_json['Buffalo']['schools'][school]['name'], schoo
1)) """
print(year_11['140600010000'])

17.8
17.2
*****
16.4
16.1
*****
*****
11.9
15.1
{'Grade 5': '10.8', 'Grade 8': '13.6', 'total': '11.9', 'Grade 6': '11.0', 'Grade 4': '13.6', 'Grade 3': '12.0', 'Grade 7': '10.2'}
*****
*****
11.9
13.1
{'Grade 5': '9.3', 'Grade 8': '14.0', 'total': '11.9', 'Grade 6': '9.9', 'Grade 4': '14.7', 'Grade 3': '13.2', 'Grade 7': '10.5'}
*****
{'math': {'total_percent': 31.1, 'proficient': [685, 872, 760, 790, 779, 664], 'classes': ['Grade 3 Math', 'Grade 4 Math', 'Grade 5 Math', 'Grade 6 Math', 'Grade 7 Math', 'Grade 8 Math'], 'totals': [2472, 2481, 2445, 2434, 2416, 2391]}, 'ela': {'total_percent': 27.0, 'proficient': [635, 680, 596, 843, 567, 534], 'classes': ['Grade 3 ELA', 'Grade 4 ELA', 'Grade 5 ELA', 'Grade 6 ELA', 'Grade 7 ELA', 'Grade 8 ELA'], 'totals': [2406, 2415, 2391, 2393, 2377, 2306]}, 'name': 'BUFFALO CITY SCHOOL DISTRICT', 'district': 'BUFFALO CITY SCHOOL DISTRICT', 'county': 'ERIE'}

```

```

In [37]: def calculate_charter_average():
    """from statistics import mean
    l = [15, 18, 2, 36, 12, 78, 5, 6, 9]
    mean(l)"""
    charter_average = {'math': {}, 'ela': {}}
    check = 0
    #Get all values added to the correct subject/year/grade list
    for charter, charter_info in clean_json['Charters']['schools'].items():
        if charter != '8686868686868':
            print(charter)
            print(charter_info)
            print('*****')
            print(charter_average)
            for subject in charter_average.keys():
                subject_years = charter_info[subject].keys()
                for year in subject_years:
                    grades = charter_info[subject][year].keys()
                    for grade in grades:
                        if grade != 'total':
                            if grade == 'Grade 7' and year == '2013' and subject ==
'ela':
                                check += 1
                                if year not in charter_average[subject]:
                                    charter_average[subject][year] = {grade: [float(char
ter_info[subject][year][grade])]}
                                elif grade not in charter_average[subject][year]:
                                    charter_average[subject][year][grade] = [float(char
ter_info[subject][year][grade])]
                                else:
                                    charter_average[subject][year][grade].append(float
(charter_info[subject][year][grade]))
                                print(len(charter_average['ela']['2013']['Grade 7']))
                                print(check)
                                #Start calculating the correct mean
                                from statistics import mean
                                for subject in charter_average:
                                    for year in charter_average[subject]:
                                        mean_grades = []
                                        for grade in charter_average[subject][year]:
                                            print(charter_average[subject][year][grade])
                                            mean_score = "{0:.1f}".format(mean(charter_average[subject][year][g
rade]))
                                            print(mean_score)
                                            charter_average[subject][year][grade] = mean_score
                                            mean_grades.append(float(mean_score))
                                        charter_average[subject][year]['total'] = mean(mean_grades)

                                clean_json['Charters']['schools']['8686868686868'] = charter_average
                                clean_json['Charters']['schools']['8686868686868']['name'] = 'Charter Average'

    calculate_charter_average()

```

142601860031

```
{'math': {'2012': {'Grade 5': '65.7', 'Grade 8': '48.4', 'total': '61.6', 'Grade 6': '65.4', 'Grade 4': '85.0', 'Grade 3': '50.0', 'Grade 7': '54.3'}, '2009': {'Grade 5': '93.8', 'Grade 8': '89.0', 'total': '90.6', 'Grade 6': '92.4', 'Grade 4': '80.2', 'Grade 3': '98.4', 'Grade 7': '89.8'}, '2006': {'Grade 5': '51.5', 'Grade 8': '35.7', 'total': '44.8', 'Grade 6': '38.1', 'Grade 4': '58.4', 'Grade 3': '66.4', 'Grade 7': '14.5'}, '2007': {'Grade 5': '79.7', 'Grade 8': '57.9', 'total': '75.2', 'Grade 6': '76.6', 'Grade 4': '90.0', 'Grade 3': '89.8', 'Grade 7': '58.1'}, '2008': {'Grade 5': '84.0', 'Grade 8': '81.2', 'total': '85.3', 'Grade 6': '76.8', 'Grade 4': '94.6', 'Grade 3': '96.2', 'Grade 7': '78.9'}, '2016': {'Grade 5': '38.6', 'Grade 8': '7.4', 'total': '19.4', 'Grade 6': '13.6', 'Grade 4': '27.8', 'Grade 3': '34.8', 'Grade 7': '4.9'}, '2013': {'Grade 5': '24.6', 'Grade 8': '18.8', 'total': '22.5', 'Grade 6': '21.8', 'Grade 4': '39.1', 'Grade 3': '14.8', 'Grade 7': '15.6'}, '2015': {'Grade 5': '33.0', 'Grade 8': '4.0', 'total': '20.5', 'Grade 6': '9.6', 'Grade 4': '44.1', 'Grade 3': '28.4', 'Grade 7': '7.4'}, '2014': {'Grade 5': '18.9', 'Grade 8': '16.3', 'total': '22.5', 'Grade 6': '10.7', 'Grade 4': '35.4', 'Grade 3': '30.7', 'Grade 7': '22.8'}, '2010': {'Grade 5': '57.8', 'Grade 8': '59.1', 'total': '62.4', 'Grade 6': '70.1', 'Grade 4': '73.9', 'Grade 3': '43.3', 'Grade 7': '70.5'}, '2017': {'Grade 5': '31.1', 'Grade 8': '8.5', 'total': '24.3', 'Grade 6': '14.6', 'Grade 4': '30.1', 'Grade 3': '48.9', 'Grade 7': '21.9'}, '2011': {'Grade 5': '65.9', 'Grade 8': '64.6', 'total': '62.8', 'Grade 6': '71.4', 'Grade 4': '67.7', 'Grade 3': '48.9', 'Grade 7': '58.4'}}}, 'ela': {'2012': {'Grade 5': '47.8', 'Grade 8': '33.6', 'total': '42.6', 'Grade 6': '36.8', 'Grade 4': '55.6', 'Grade 3': '36.3', 'Grade 7': '45.0'}, '2009': {'Grade 5': '77.7', 'Grade 8': '73.6', 'total': '76.6', 'Grade 6': '78.5', 'Grade 4': '72.5', 'Grade 3': '82.0', 'Grade 7': '75.2'}, '2006': {'Grade 5': '53.7', 'Grade 8': '29.8', 'total': '41.0', 'Grade 6': '36.0', 'Grade 4': '44.5', 'Grade 3': '50.8', 'Grade 7': '28.3'}, '2007': {'Grade 5': '62.8', 'Grade 8': '33.8', 'total': '49.7', 'Grade 6': '43.9', 'Grade 4': '55.0', 'Grade 3': '66.4', 'Grade 7': '37.5'}, '2008': {'Grade 5': '75.9', 'Grade 8': '32.4', 'total': '61.9', 'Grade 6': '50.8', 'Grade 4': '71.8', 'Grade 3': '73.3', 'Grade 7': '66.4'}, '2016': {'Grade 5': '18.8', 'Grade 8': '16.5', 'total': '18.0', 'Grade 6': '16.4', 'Grade 4': '15.7', 'Grade 3': '25.9', 'Grade 7': '16.4'}, '2013': {'Grade 5': '17.0', 'Grade 8': '12.3', 'total': '15.1', 'Grade 6': '20.3', 'Grade 4': '16.7', 'Grade 3': '14.1', 'Grade 7': '10.0'}, '2015': {'Grade 5': '14.2', 'Grade 8': '6.8', 'total': '14.0', 'Grade 6': '10.1', 'Grade 4': '31.5', 'Grade 3': '17.9', 'Grade 7': '6.5'}, '2014': {'Grade 5': '16.8', 'Grade 8': '16.2', 'total': '17.2', 'Grade 6': '11.3', 'Grade 4': '26.5', 'Grade 3': '21.7', 'Grade 7': '10.4'}, '2010': {'Grade 5': '41.5', 'Grade 8': '40.8', 'total': '46.1', 'Grade 6': '44.0', 'Grade 4': '56.7', 'Grade 3': '48.1', 'Grade 7': '45.0'}, '2017': {'Grade 5': '15.8', 'Grade 8': '23.5', 'total': '23.9', 'Grade 6': '22.2', 'Grade 4': '25.0', 'Grade 3': '28.9', 'Grade 7': '27.5'}, '2011': {'Grade 5': '59.3', 'Grade 8': '54.3', 'total': '48.9', 'Grade 6': '51.5', 'Grade 4': '42.5', 'Grade 3': '46.6', 'Grade 7': '38.9'}}}, 'district': 'KENMORE-TONAWANDA UNION FREE SCHOOL DISTRICT', 'name': 'Charter Sch. for App. Tech.'}
```

\*\*\*\*\*

{'math': {}, 'ela': {}}

140600860863

```
{'math': {'2017': {'total': '2.6', 'Grade 7': '2.6'}}}, 'ela': {'2017': {'total': '21.1', 'Grade 7': '21.1'}}}, 'district': 'BUFFALO CITY SCHOOL DISTRICT', 'name': 'WNY Maritime Charter'}
```

\*\*\*\*\*

```
{'math': {'2012': {'Grade 5': [65.7], 'Grade 8': [48.4], 'Grade 6': [65.4], 'Grade 4': [85.0], 'Grade 3': [50.0], 'Grade 7': [54.3]}, '2009': {'Grade 5': [93.8], 'Grade 8': [89.0], 'Grade 6': [92.4], 'Grade 4': [80.2], 'Grade 3': [98.4], 'Grade 7': [89.8]}, '2006': {'Grade 5': [51.5], 'Grade 8': [35.7], 'Grade 6': [38.1], 'Grade 4': [58.4], 'Grade 3': [66.4], 'Grade 7': [14.5]}, '2007': {'Grade 5': [79.7], 'Grade 8': [57.9], 'Grade 6': [76.6], 'Grade 4': [90.0], 'Grade 3': [89.8], 'Grade 7': [58.1]}, '2008': {'Grade 5': [84.0], 'Grade 8': [81.2], 'Grade 6': [76.8], 'Grade 4': [94.6], 'Grade 3': [96.2], 'Grade 7': [78.9]}, '2016': {'Grade 5': [38.6], 'Grade 8': [7.4], 'Grade 6': [13.6], 'Grade 4': [27.8], 'Grade 3': [34.8], 'Grade 7': [4.9]}, '2013': {'Grade 5': [24.6], 'Grade 8': [18.8], 'Grade 6': [21.8], 'Grade 4': [39.1], 'Grade 3': [14.8], 'Grade 7': [15.6]}, '2015': {'Grade 5': [33.0], 'Grade 8': [4.0], 'Grade 6': [9.6], 'Grade 4': [44.1],
```

```
In [38]: print(clean_json['Charters']['schools']['8686868686868'])  
         print(clean_json['Niagara']['schools']['400000000000'])
```

```
{'math': {'2012': {'Grade 5': '62.1', 'Grade 8': '57.6', 'Grade 6': '57.9', 'total': 59.266666666666666, 'Grade 4': '67.7', 'Grade 3': '46.7', 'Grade 7': '63.6'}, '2009': {'Grade 5': '88.2', 'Grade 8': '78.6', 'Grade 6': '87.1', 'total': 87.516666666666667, 'Grade 4': '88.6', 'Grade 3': '95.8', 'Grade 7': '86.8'}, '2006': {'Grade 5': '55.0', 'Grade 8': '27.4', 'Grade 6': '53.5', 'total': 52.4, 'Grade 4': '72.4', 'Grade 3': '74.2', 'Grade 7': '31.9'}, '2007': {'Grade 5': '67.3', 'Grade 8': '52.5', 'Grade 6': '78.7', 'total': 70.216666666666667, 'Grade 4': '71.6', 'Grade 3': '85.9', 'Grade 7': '65.3'}, '2008': {'Grade 5': '79.8', 'Grade 8': '73.4', 'Grade 6': '71.3', 'total': 79.116666666666667, 'Grade 4': '82.4', 'Grade 3': '89.6', 'Grade 7': '78.2'}, '2016': {'Grade 5': '27.8', 'Grade 8': '18.1', 'Grade 6': '30.1', 'total': 26.45, 'Grade 4': '33.0', 'Grade 3': '35.4', 'Grade 7': '14.3'}, '2013': {'Grade 5': '22.3', 'Grade 8': '19.3', 'Grade 6': '24.5', 'total': 22.116666666666667, 'Grade 4': '31.2', 'Grade 3': '23.3', 'Grade 7': '12.1'}, '2015': {'Grade 5': '27.4', 'Grade 8': '18.0', 'Grade 6': '24.5', 'total': 26.266666666666666, 'Grade 4': '36.0', 'Grade 3': '34.4', 'Grade 7': '17.3'}, '2014': {'Grade 5': '24.2', 'Grade 8': '16.0', 'Grade 6': '27.8', 'total': 25.016666666666666, 'Grade 4': '31.3', 'Grade 3': '30.0', 'Grade 7': '20.8'}, '2010': {'Grade 5': '63.9', 'Grade 8': '39.8', 'Grade 6': '63.2', 'total': 54.65, 'Grade 4': '59.6', 'Grade 3': '52.9', 'Grade 7': '48.5'}, '2017': {'Grade 5': '27.8', 'Grade 8': '14.8', 'Grade 6': '33.7', 'total': 27.566666666666666, 'Grade 4': '32.3', 'Grade 3': '39.1', 'Grade 7': '17.7'}, '2011': {'Grade 5': '52.7', 'Grade 8': '50.5', 'Grade 6': '61.7', 'total': 55.583333333333336, 'Grade 4': '65.2', 'Grade 3': '54.3', 'Grade 7': '49.1'}}}, {'ela': {'2012': {'Grade 5': '46.9', 'Grade 8': '37.1', 'Grade 6': '39.9', 'total': 43.466666666666667, 'Grade 4': '54.7', 'Grade 3': '42.4', 'Grade 7': '39.8'}, '2009': {'Grade 5': '76.3', 'Grade 8': '60.7', 'Grade 6': '77.7', 'total': 73.233333333333333, 'Grade 4': '76.0', 'Grade 3': '74.6', 'Grade 7': '74.1'}, '2006': {'Grade 5': '58.3', 'Grade 8': '37.2', 'Grade 6': '46.5', 'total': 48.25, 'Grade 4': '54.0', 'Grade 3': '48.9', 'Grade 7': '44.6'}, '2007': {'Grade 5': '59.7', 'Grade 8': '45.0', 'Grade 6': '50.7', 'total': 52.583333333333336, 'Grade 4': '58.5', 'Grade 3': '57.6', 'Grade 7': '44.0'}, '2008': {'Grade 5': '72.4', 'Grade 8': '51.6', 'Grade 6': '61.0', 'total': 63.75, 'Grade 4': '63.6', 'Grade 3': '70.3', 'Grade 7': '63.6'}, '2016': {'Grade 5': '20.5', 'Grade 8': '25.9', 'Grade 6': '24.6', 'total': 25.85, 'Grade 4': '30.2', 'Grade 3': '33.9', 'Grade 7': '20.0'}, '2013': {'Grade 5': '17.7', 'Grade 8': '17.2', 'Grade 6': '19.1', 'total': 17.95, 'Grade 4': '20.7', 'Grade 3': '18.1', 'Grade 7': '14.9'}, '2015': {'Grade 5': '21.2', 'Grade 8': '19.5', 'Grade 6': '15.3', 'total': 20.233333333333334, 'Grade 4': '26.2', 'Grade 3': '24.6', 'Grade 7': '14.6'}, '2014': {'Grade 5': '15.3', 'Grade 8': '19.1', 'Grade 6': '14.6', 'total': 19.133333333333333, 'Grade 4': '25.3', 'Grade 3': '23.2', 'Grade 7': '17.3'}, '2010': {'Grade 5': '46.3', 'Grade 8': '36.3', 'Grade 6': '48.7', 'total': 43.6, 'Grade 4': '48.3', 'Grade 3': '48.8', 'Grade 7': '33.2'}, '2017': {'Grade 5': '22.2', 'Grade 8': '30.8', 'Grade 6': '22.4', 'total': 26.533333333333333, 'Grade 4': '29.5', 'Grade 3': '31.5', 'Grade 7': '22.8'}, '2011': {'Grade 5': '39.4', 'Grade 8': '32.4', 'Grade 6': '55.4', 'total': 40.6, 'Grade 4': '47.5', 'Grade 3': '44.3', 'Grade 7': '24.6'}}}, {'name': 'Charter Average'}], {'math': {'2012': {'Grade 5': '63.6', 'Grade 8': '58.3', 'total': '64.0', 'Grade 6': '65.8', 'Grade 4': '67.5', 'Grade 3': '61.1', 'Grade 7': '67.7'}, '2009': {'Grade 5': '90.2', 'Grade 8': '84.3', 'total': '90.1', 'Grade 6': '86.3', 'Grade 4': '90.9', 'Grade 3': '94.8', 'Grade 7': '94.2'}, '2006': {'Grade 5': '76.0', 'Grade 8': '66.3', 'total': '73.2', 'Grade 6': '67.0', 'Grade 4': '84.2', 'Grade 3': '87.7', 'Grade 7': '61.6'}, '2007': {'Grade 5': '80.1', 'Grade 8': '64.3', 'total': '77.8', 'Grade 6': '76.0', 'Grade 4': '85.3', 'Grade 3': '90.3', 'Grade 7': '73.9'}, '2008': {'Grade 5': '87.3', 'Grade 8': '76.3', 'total': '85.8', 'Grade 6': '85.1', 'Grade 4': '88.2', 'Grade 3': '92.5', 'Grade 7': '86.3'}, '2016': {'Grade 5': '39.0', 'Grade 8': '10.7', 'total': '37.1', 'Grade 6': '37.9', 'Grade 4': '44.8', 'Grade 3': '48.9', 'Grade 7': '27.5'}, '2013': {'Grade 5': '23.1', 'Grade 8': '18.8', 'total': '25.0', 'Grade 6': '25.3', 'Grade 4': '32.5', 'Grade 3': '28.2', 'Grade 7': '22.7'}, '2015': {'Grade 5': '42.2', 'Grade 8': '14.4', 'total': '35.2', 'Grade 6': '37.5', 'Grade 4': '44.9', 'Grade 3': '38.8', 'Grade 7': '25.5'}, '2014': {'Grade 5': '34.7', 'Grade 8': '13.8', 'total': '31.2', 'Grade 6': '33.7', 'Grade 4': '37.2', 'Grade 3': '39.1', 'Grade 7': '26.3'}, '2010': {'Grade 5': '67.7', 'Grade 8': '52.5', 'total': '62.9', 'Grade 6': '65.4', 'Grade 4': '67.3', 'Grade 3': '59.2', 'Grade 7': '66.2'}, '2017': {'Grade 5': '39.9', 'Grade 8': '9.7', 'total': '37.6', 'Grade 6': '37.6', 'Grade 4': '45.8',
```

```
In [39]: for group in clean_json:
group_max = {'mathmaxValue': 0, 'elamaxValue': 0}
group_min = {'mathminValue': 100, 'elaminValue': 100}
for school in clean_json[group]['schools']:
    #print(school)
    try:
        math_school_value = float(clean_json[group]['schools'][school]['math
']] ['2017'] ['total'])
    except ValueError:
        math_school_value = '-'
    #print(math_school_value)
    #print(group_max)
    #print(group_min)
    if math_school_value != '-':
        if math_school_value > group_max['mathmaxValue']:
            group_max['mathmaxValue'] = math_school_value
            group_max['mathschool'] = school
        elif math_school_value < group_min['mathminValue']:
            group_min['mathminValue'] = math_school_value
            group_min['mathschool'] = school

    try:
        ela_school_value = float(clean_json[group]['schools'][school]['ela'] ['2
017'] ['total'])
    except ValueError:
        ela_school_value = '-'
    #print(ela_school_value)
    if ela_school_value != '-':
        if ela_school_value > group_max['elamaxValue']:
            group_max['elamaxValue'] = ela_school_value
            group_max['elaschool'] = school
        elif ela_school_value < group_min['elaminValue']:
            group_min['elaminValue'] = ela_school_value
            group_min['elaschool'] = school

print('&&&&&group&&&&')
print(group)
print('Highest schools')
print(group_max)
print('Lowest schools')
print(group_min)
clean_json[group]['max'] = group_max['elaschool']
clean_json[group]['min'] = group_min['elaschool']
```

~~~~~group~~~~~

Buffalo

Highest schools

{'elamaxValue': 83.3, 'mathschool': '140600010102', 'mathmaxValue': 79.7, 'elasc  
hool': '140600010102'}

Lowest schools

{'elaminValue': 0.0, 'elaschool': '140600010133', 'mathschool': '140600010032',  
'mathminValue': 1.6}

~~~~~group~~~~~

Erie

Highest schools

{'elamaxValue': 72.0, 'mathschool': '140801060003', 'mathmaxValue': 87.6, 'elasc  
hool': '140203060002'}

Lowest schools

{'elaminValue': 6.5, 'elaschool': '141800010005', 'mathschool': '141800010005',  
'mathminValue': 5.1}

~~~~~group~~~~~

Charters

Highest schools

{'elamaxValue': 46.6, 'mathschool': '400701860890', 'mathmaxValue': 52.9, 'elasc  
hool': '140600860896'}

Lowest schools

{'elaminValue': 9.8, 'elaschool': '140600860856', 'mathschool': '140600860863',  
'mathminValue': 2.6}

~~~~~group~~~~~

Niagara

Highest schools

{'elamaxValue': 81.2, 'mathschool': '400701060005', 'mathmaxValue': 73.1, 'elasc  
hool': '400301060002'}

Lowest schools

{'elaminValue': 14.1, 'elaschool': '400800010015', 'mathschool': '400800010040',  
'mathminValue': 8.7}

```
In [83]: clean_json['Erie']['schools']['140101060000']
```



```
Out[83]: {'district': 'ALDEN CENTRAL SCHOOL DISTRICT',
          'ela': {'2006': {'Grade 3': '78.0',
                           'Grade 4': '77.0',
                           'Grade 5': '71.4',
                           'Grade 6': '74.6',
                           'Grade 7': '78.5',
                           'Grade 8': '75.9',
                           'total': '76.0'},
                  '2007': {'Grade 3': '76.5',
                           'Grade 4': '81.2',
                           'Grade 5': '79.7',
                           'Grade 6': '73.0',
                           'Grade 7': '69.1',
                           'Grade 8': '77.1',
                           'total': '76.0'},
                  '2008': {'Grade 3': '83.7',
                           'Grade 4': '80.5',
                           'Grade 5': '84.3',
                           'Grade 6': '81.0',
                           'Grade 7': '87.5',
                           'Grade 8': '62.3',
                           'total': '79.6'},
                  '2009': {'Grade 3': '85.9',
                           'Grade 4': '92.2',
                           'Grade 5': '87.9',
                           'Grade 6': '86.3',
                           'Grade 7': '91.3',
                           'Grade 8': '85.6',
                           'total': '88.2'},
                  '2010': {'Grade 3': '69.9',
                           'Grade 4': '70.6',
                           'Grade 5': '64.8',
                           'Grade 6': '68.5',
                           'Grade 7': '63.0',
                           'Grade 8': '69.3',
                           'total': '67.7'},
                  '2011': {'Grade 3': '66.1',
                           'Grade 4': '68.0',
                           'Grade 5': '55.8',
                           'Grade 6': '74.6',
                           'Grade 7': '54.5',
                           'Grade 8': '64.3',
                           'total': '63.6'},
                  '2012': {'Grade 3': '68.3',
                           'Grade 4': '71.7',
                           'Grade 5': '62.4',
                           'Grade 6': '62.4',
                           'Grade 7': '62.8',
                           'Grade 8': '52.3',
                           'total': '63.1'},
                  '2013': {'Grade 3': '33.9',
                           'Grade 4': '37.6',
                           'Grade 5': '33.9',
                           'Grade 6': '24.4',
                           'Grade 7': '38.5',
                           'Grade 8': '48.4',
                           'total': '36.3'},
                  '2014': {'Grade 3': '45.2',
                           'Grade 4': '37.4',
                           'Grade 5': '27.1',
                           'Grade 6': '31.1',
                           'Grade 7': '22.1',
                           'Grade 8': '26.5',
                           'total': '31.7'}}
```

```

In [81]: clean_csv = []
clean_csv.append(['Group', 'District', 'School', 'GEOID', 'Grade', 'Test_type', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017'])
years = ['2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017']
grades = ['Grade 5', 'Grade 8', 'total', 'Grade 6', 'Grade 4', 'Grade 3', 'Grade 7']

for group, schools in clean_json.items():
    print('group = {}'.format(group))
    for school, school_info in schools['schools'].items():
        print('school = {}'.format(school))
        for grade in grades:
            foundNum = False
            for year in years:
                if year in school_info['math']:
                    if grade in school_info['math'][year]:
                        foundNum = True
                        break
                if year in school_info['ela']:
                    if grade in school_info['ela'][year]:
                        foundNum = True
                        break
            if foundNum:
                print(school_info)
                if school == '8686868686868':
                    grade_info_math = [group, 'charters', school_info['name'], school, grade, 'math']
                    grade_info_ela = [group, 'charters', school_info['name'], school, grade, 'ela']
                else:
                    grade_info_math = [group, school_info['district'], school_info['name'], school, grade, 'math']
                    grade_info_ela = [group, school_info['district'], school_info['name'], school, grade, 'ela']
                for year in years:
                    try:
                        grade_info_math.append(school_info['math'][year][grade])
                    except KeyError:
                        print('#####')
                        print(grade)
                        try:
                            print(school_info['math'][year])
                        except KeyError:
                            print(year)
                            print(school_info['math'])
                        print('#####')
                        grade_info_math.append('-')
                    try:
                        grade_info_ela.append(school_info['ela'][year][grade])
                    except KeyError:
                        print('#####')
                        print(grade)
                        try:
                            print(school_info['ela'][year])
                        except KeyError:
                            print(year)
                            print(school_info['ela'])
                        print('#####')
                        grade_info_ela.append('-')
                #print(grade_info_math)
                clean_csv.append(grade_info_math)
                clean_csv.append(grade_info_ela)
print(len(clean_csv))
print(clean_csv[0])

```

```
group = Buffalo
school = 140600010037
{'math': {'2012': {'Grade 5': '6.2', 'Grade 8': '7.8', 'total': '6.9', 'Grade 6': '3.0', 'Grade 4': '16.0', 'Grade 3': '5.3', 'Grade 7': '4.3'}, '2009': {'Grade 5': '25.0', 'Grade 8': '56.6', 'total': '48.8', 'Grade 6': '19.6', 'Grade 4': '44.4', 'Grade 3': '80.6', 'Grade 7': '55.2'}, '2006': {'Grade 5': '20.6', 'Grade 8': '15.3', 'total': '24.1', 'Grade 6': '7.9', 'Grade 4': '57.6', 'Grade 3': '26.0', 'Grade 7': '21.5'}, '2007': {'Grade 5': '19.1', 'Grade 8': '12.9', 'total': '19.1', 'Grade 6': '19.0', 'Grade 4': '25.5', 'Grade 3': '27.9', 'Grade 7': '12.5'}, '2008': {'Grade 5': '21.8', 'Grade 8': '23.2', 'total': '27.3', 'Grade 6': '27.9', 'Grade 4': '18.6', 'Grade 3': '46.8', 'Grade 7': '28.6'}, '2016': {'Grade 5': '8.7', 'Grade 8': '0.0', 'total': '5.7', 'Grade 6': '0.0', 'Grade 4': '2.7', 'Grade 3': '27.6', 'Grade 7': '0.0'}, '2013': {'Grade 5': '0.0', 'Grade 8': '0.0', 'total': '0.3', 'Grade 6': '0.0', 'Grade 4': '0.0', 'Grade 3': '2.3', 'Grade 7': '0.0'}, '2015': {'Grade 5': '0.0', 'Grade 8': '0.0', 'total': '2.1', 'Grade 6': '1.6', 'Grade 4': '8.1', 'Grade 3': '7.1', 'Grade 7': '0.0'}, '2014': {'Grade 5': '0.0', 'Grade 8': '0.0', 'total': '0.0', 'Grade 6': '0.0', 'Grade 4': '0.0', 'Grade 3': '0.0', 'Grade 7': '0.0'}, '2010': {'Grade 5': '13.8', 'Grade 8': '14.3', 'total': '14.0', 'Grade 6': '13.2', 'Grade 4': '21.7', 'Grade 3': '13.6', 'Grade 7': '5.4'}, '2017': {'Grade 5': '6.1', 'Grade 8': '0.0', 'total': '4.9', 'Grade 6': '6.8', 'Grade 4': '6.9', 'Grade 3': '12.1', 'Grade 7': '0.0'}, '2011': {'Grade 5': '16.5', 'Grade 8': '9.4', 'total': '17.3', 'Grade 6': '27.1', 'Grade 4': '20.5', 'Grade 3': '24.4', 'Grade 7': '7.3'}}}, {'ela': {'2012': {'Grade 5': '21.2', 'Grade 8': '5.9', 'total': '11.7', 'Grade 6': '13.6', 'Grade 4': '8.2', 'Grade 3': '7.0', 'Grade 7': '8.7'}, '2009': {'Grade 5': '19.6', 'Grade 8': '18.5', 'total': '29.0', 'Grade 6': '44.2', 'Grade 4': '26.3', 'Grade 3': '27.9', 'Grade 7': '39.1'}, '2006': {'Grade 5': '10.9', 'Grade 8': '8.6', 'total': '15.8', 'Grade 6': '9.8', 'Grade 4': '29.3', 'Grade 3': '30.8', 'Grade 7': '11.0'}, '2007': {'Grade 5': '41.2', 'Grade 8': '14.1', 'total': '19.5', 'Grade 6': '16.9', 'Grade 4': '27.5', 'Grade 3': '9.8', 'Grade 7': '9.2'}, '2008': {'Grade 5': '38.9', 'Grade 8': '13.0', 'total': '27.6', 'Grade 6': '32.8', 'Grade 4': '15.0', 'Grade 3': '42.6', 'Grade 7': '25.9'}, '2016': {'Grade 5': '6.0', 'Grade 8': '6.2', 'total': '5.0', 'Grade 6': '2.2', 'Grade 4': '5.0', 'Grade 3': '7.1', 'Grade 7': '4.4'}, '2013': {'Grade 5': '1.4', 'Grade 8': '2.3', 'total': '1.7', 'Grade 6': '0.0', 'Grade 4': '0.0', 'Grade 3': '4.7', 'Grade 7': '3.4'}, '2015': {'Grade 5': '1.9', 'Grade 8': '0.0', 'total': '1.4', 'Grade 6': '0.0', 'Grade 4': '0.0', 'Grade 3': '6.9', 'Grade 7': '2.0'}, '2014': {'Grade 5': '3.4', 'Grade 8': '0.0', 'total': '0.6', 'Grade 6': '0.0', 'Grade 4': '0.0', 'Grade 3': '0.0', 'Grade 7': '0.0'}, '2010': {'Grade 5': '8.8', 'Grade 8': '9.5', 'total': '10.0', 'Grade 6': '5.8', 'Grade 4': '14.5', 'Grade 3': '9.2', 'Grade 7': '10.9'}, '2017': {'Grade 5': '6.5', 'Grade 8': '9.5', 'total': '5.8', 'Grade 6': '10.2', 'Grade 4': '0.0', 'Grade 3': '6.2', 'Grade 7': '0.0'}, '2011': {'Grade 5': '9.0', 'Grade 8': '3.8', 'total': '8.1', 'Grade 6': '12.5', 'Grade 4': '7.7', 'Grade 3': '17.4', 'Grade 7': '0.0'}}}, {'district': 'BUFFALO CITY SCHOOL DISTRICT', 'name': 'Daniel Futures'}
```

```
In [82]: with open('output.csv', 'w') as csvfile:
        writer = csv.writer(csvfile)
        for row in clean_csv:
            writer.writerow(row)
```

```
In [75]: print(clean_json['Charters']['schools']['8686868686868'])
```

```
{'math': {'2012': {'Grade 5': '62.1', 'Grade 8': '57.6', 'Grade 6': '57.9', 'total': 59.266666666666666, 'Grade 4': '67.7', 'Grade 3': '46.7', 'Grade 7': '63.6'}, '2009': {'Grade 5': '88.2', 'Grade 8': '78.6', 'Grade 6': '87.1', 'total': 87.516666666666667, 'Grade 4': '88.6', 'Grade 3': '95.8', 'Grade 7': '86.8'}, '2006': {'Grade 5': '55.0', 'Grade 8': '27.4', 'Grade 6': '53.5', 'total': 52.4, 'Grade 4': '72.4', 'Grade 3': '74.2', 'Grade 7': '31.9'}, '2007': {'Grade 5': '67.3', 'Grade 8': '52.5', 'Grade 6': '78.7', 'total': 70.216666666666667, 'Grade 4': '71.6', 'Grade 3': '85.9', 'Grade 7': '65.3'}, '2008': {'Grade 5': '79.8', 'Grade 8': '73.4', 'Grade 6': '71.3', 'total': 79.116666666666667, 'Grade 4': '82.4', 'Grade 3': '89.6', 'Grade 7': '78.2'}, '2016': {'Grade 5': '27.8', 'Grade 8': '18.1', 'Grade 6': '30.1', 'total': 26.45, 'Grade 4': '33.0', 'Grade 3': '35.4', 'Grade 7': '14.3'}, '2013': {'Grade 5': '22.3', 'Grade 8': '19.3', 'Grade 6': '24.5', 'total': 22.116666666666667, 'Grade 4': '31.2', 'Grade 3': '23.3', 'Grade 7': '12.1'}, '2015': {'Grade 5': '27.4', 'Grade 8': '18.0', 'Grade 6': '24.5', 'total': 26.266666666666666, 'Grade 4': '36.0', 'Grade 3': '34.4', 'Grade 7': '17.3'}, '2014': {'Grade 5': '24.2', 'Grade 8': '16.0', 'Grade 6': '27.8', 'total': 25.016666666666666, 'Grade 4': '31.3', 'Grade 3': '30.0', 'Grade 7': '20.8'}, '2010': {'Grade 5': '63.9', 'Grade 8': '39.8', 'Grade 6': '63.2', 'total': 54.65, 'Grade 4': '59.6', 'Grade 3': '52.9', 'Grade 7': '48.5'}, '2017': {'Grade 5': '27.8', 'Grade 8': '14.8', 'Grade 6': '33.7', 'total': 27.566666666666666, 'Grade 4': '32.3', 'Grade 3': '39.1', 'Grade 7': '17.7'}, '2011': {'Grade 5': '52.7', 'Grade 8': '50.5', 'Grade 6': '61.7', 'total': 55.583333333333336, 'Grade 4': '65.2', 'Grade 3': '54.3', 'Grade 7': '49.1'}}, 'ela': {'2012': {'Grade 5': '46.9', 'Grade 8': '37.1', 'Grade 6': '39.9', 'total': 43.466666666666667, 'Grade 4': '54.7', 'Grade 3': '42.4', 'Grade 7': '39.8'}, '2009': {'Grade 5': '76.3', 'Grade 8': '60.7', 'Grade 6': '77.7', 'total': 73.233333333333333, 'Grade 4': '76.0', 'Grade 3': '74.6', 'Grade 7': '74.1'}, '2006': {'Grade 5': '58.3', 'Grade 8': '37.2', 'Grade 6': '46.5', 'total': 48.25, 'Grade 4': '54.0', 'Grade 3': '48.9', 'Grade 7': '44.6'}, '2007': {'Grade 5': '59.7', 'Grade 8': '45.0', 'Grade 6': '50.7', 'total': 52.583333333333336, 'Grade 4': '58.5', 'Grade 3': '57.6', 'Grade 7': '44.0'}, '2008': {'Grade 5': '72.4', 'Grade 8': '51.6', 'Grade 6': '61.0', 'total': 63.75, 'Grade 4': '63.6', 'Grade 3': '70.3', 'Grade 7': '63.6'}, '2016': {'Grade 5': '20.5', 'Grade 8': '25.9', 'Grade 6': '24.6', 'total': 25.85, 'Grade 4': '30.2', 'Grade 3': '33.9', 'Grade 7': '20.0'}, '2013': {'Grade 5': '17.7', 'Grade 8': '17.2', 'Grade 6': '19.1', 'total': 17.95, 'Grade 4': '20.7', 'Grade 3': '18.1', 'Grade 7': '14.9'}, '2015': {'Grade 5': '21.2', 'Grade 8': '19.5', 'Grade 6': '15.3', 'total': 20.233333333333334, 'Grade 4': '26.2', 'Grade 3': '24.6', 'Grade 7': '14.6'}, '2014': {'Grade 5': '15.3', 'Grade 8': '19.1', 'Grade 6': '14.6', 'total': 19.133333333333333, 'Grade 4': '25.3', 'Grade 3': '23.2', 'Grade 7': '17.3'}, '2010': {'Grade 5': '46.3', 'Grade 8': '36.3', 'Grade 6': '48.7', 'total': 43.6, 'Grade 4': '48.3', 'Grade 3': '48.8', 'Grade 7': '33.2'}, '2017': {'Grade 5': '22.2', 'Grade 8': '30.8', 'Grade 6': '22.4', 'total': 26.533333333333333, 'Grade 4': '29.5', 'Grade 3': '31.5', 'Grade 7': '22.8'}, '2011': {'Grade 5': '39.4', 'Grade 8': '32.4', 'Grade 6': '55.4', 'total': 40.6, 'Grade 4': '47.5', 'Grade 3': '44.3', 'Grade 7': '24.6'}}, 'name': 'Charter Average'}
```

```
In [58]: print(clean_json['Erie']['schools']['140000000000'])  
         print(clean_json['Niagara']['ordered_schools'])  
  
import json  
with open('data.json', 'w') as output:  
    json.dump(clean_json, output)
```

```

{'ela': {'2012': {'Grade 6': '58.5', 'Grade 5': '56.6', 'Grade 8': '52.7', 'Grade 7': '52.2', 'Grade 3': '56.2', 'Grade 4': '58.7', 'total': '55.8'}, '2013': {'Grade 6': '32.6', 'Grade 5': '28.4', 'Grade 8': '36.6', 'Grade 7': '32.3', 'Grade 3': '32.0', 'Grade 4': '30.2', 'total': '32.0'}, '2017': {'Grade 6': '29.3', 'Grade 5': '30.4', 'Grade 8': '36.3', 'Grade 7': '36.4', 'Grade 3': '37.4', 'Grade 4': '35.3', 'total': '34.2'}, '2014': {'Grade 6': '27.3', 'Grade 5': '26.4', 'Grade 8': '33.2', 'Grade 7': '26.7', 'Grade 3': '32.8', 'Grade 4': '32.7', 'total': '29.8'}, '2006': {'Grade 6': '61.6', 'Grade 5': '67.3', 'Grade 8': '48.5', 'Grade 7': '57.7', 'Grade 3': '67.3', 'Grade 4': '66.5', 'total': '60.8'}, '2015': {'Grade 6': '27.2', 'Grade 5': '28.1', 'Grade 8': '31.5', 'Grade 7': '26.3', 'Grade 3': '29.6', 'Grade 4': '31.8', 'total': '29.1'}, '2007': {'Grade 6': '64.8', 'Grade 5': '69.1', 'Grade 8': '62.0', 'Grade 7': '58.7', 'Grade 3': '68.0', 'Grade 4': '69.4', 'total': '65.1'}, '2009': {'Grade 6': '82.8', 'Grade 5': '81.4', 'Grade 8': '72.1', 'Grade 7': '82.8', 'Grade 3': '75.8', 'Grade 4': '76.8', 'total': '78.6'}, '2011': {'Grade 6': '60.4', 'Grade 5': '52.9', 'Grade 8': '51.1', 'Grade 7': '51.1', 'Grade 3': '55.7', 'Grade 4': '56.2', 'total': '54.6'}, '2016': {'Grade 6': '32.3', 'Grade 5': '29.5', 'Grade 8': '35.5', 'Grade 7': '30.4', 'Grade 3': '39.0', 'Grade 4': '35.7', 'total': '33.9'}, '2008': {'Grade 6': '70.1', 'Grade 5': '78.3', 'Grade 8': '59.8', 'Grade 7': '71.4', 'Grade 3': '72.4', 'Grade 4': '70.8', 'total': '70.3'}, '2010': {'Grade 6': '58.0', 'Grade 5': '52.5', 'Grade 8': '53.0', 'Grade 7': '52.8', 'Grade 3': '56.1', 'Grade 4': '58.0', 'total': '55.1'}}}, {'name': 'Erie County average', 'district': 'ERIE COUNTY', 'math': {'2012': {'Grade 6': '64.8', 'Grade 5': '63.1', 'Grade 8': '62.1', 'Grade 7': '65.1', 'Grade 3': '60.0', 'Grade 4': '67.3', 'total': '63.7'}, '2013': {'Grade 6': '29.0', 'Grade 5': '27.4', 'Grade 8': '28.2', 'Grade 7': '26.7', 'Grade 3': '33.4', 'Grade 4': '36.8', 'total': '30.2'}, '2017': {'Grade 6': '38.7', 'Grade 5': '40.0', 'Grade 8': '13.3', 'Grade 7': '33.1', 'Grade 3': '43.2', 'Grade 4': '39.3', 'total': '36.2'}, '2014': {'Grade 6': '33.3', 'Grade 5': '36.0', 'Grade 8': '17.3', 'Grade 7': '30.7', 'Grade 3': '40.9', 'Grade 4': '39.8', 'total': '33.6'}, '2006': {'Grade 6': '60.9', 'Grade 5': '65.5', 'Grade 8': '53.1', 'Grade 7': '57.4', 'Grade 3': '75.9', 'Grade 4': '76.6', 'total': '64.2'}, '2015': {'Grade 6': '34.0', 'Grade 5': '40.7', 'Grade 8': '16.6', 'Grade 7': '30.8', 'Grade 3': '39.3', 'Grade 4': '42.7', 'total': '35.2'}, '2007': {'Grade 6': '71.3', 'Grade 5': '72.9', 'Grade 8': '63.5', 'Grade 7': '67.4', 'Grade 3': '80.0', 'Grade 4': '77.2', 'total': '71.8'}, '2009': {'Grade 6': '82.8', 'Grade 5': '85.9', 'Grade 8': '83.5', 'Grade 7': '87.1', 'Grade 3': '91.0', 'Grade 4': '85.5', 'total': '85.9'}, '2011': {'Grade 6': '63.2', 'Grade 5': '65.1', 'Grade 8': '62.2', 'Grade 7': '66.7', 'Grade 3': '57.1', 'Grade 4': '67.2', 'total': '63.6'}, '2016': {'Grade 6': '36.4', 'Grade 5': '38.1', 'Grade 8': '16.8', 'Grade 7': '29.7', 'Grade 3': '40.3', 'Grade 4': '41.3', 'total': '35.0'}, '2008': {'Grade 6': '78.6', 'Grade 5': '80.5', 'Grade 8': '71.3', 'Grade 7': '79.8', 'Grade 3': '87.4', 'Grade 4': '81.9', 'total': '79.8'}, '2010': {'Grade 6': '63.5', 'Grade 5': '64.7', 'Grade 8': '58.5', 'Grade 7': '65.8', 'Grade 3': '58.0', 'Grade 4': '63.4', 'total': '62.3'}}}]
[['400000000000', 'Niagara County average'], ['401301040000', 'BARKER SCHOOL DISTRICT'], ['401301040003', 'Barker Junior/Senior High'], ['401301040002', 'Pratt Elementary'], ['400301060000', 'LEW-PORT SCHOOL DISTRICT'], ['400301060003', 'Intermediate Educ. Center'], ['400301060005', 'Lewiston Porter Middle'], ['400301060002', 'Primary Education Center'], ['400400010000', 'LOCKPORT SCHOOL DISTRICT'], ['400400010009', 'Belknap Intermediate'], ['400400010007', 'Kelley Elementary'], ['400400010001', 'Merritt Elementary'], ['400400010010', 'North Park Junior High'], ['400400010005', 'Southard Elementary'], ['400400010002', 'Upson Elementary'], ['400601060000', 'NEWFANE SCHOOL DISTRICT'], ['400601060002', 'Newfane Elementary'], ['400601060008', 'Newfane Middle'], ['400800010000', 'NIAGARA FALLS SCHOOL DISTRICT'], ['400800010031', 'Abate Elementary'], ['400800010042', 'Cataract Elementary'], ['400800010040', 'Gaskill Preparatory'], ['400800010020', 'Hyde Park'], ['400800010015', 'Kalfas Magnet'], ['400800010041', 'LaSalle Preparatory'], ['400800010012', 'Mann'], ['400800010021', 'Maple Avenue'], ['400800010022', 'Niagara Street'], ['400800010010', 'Seventy Ninth Street'], ['400701060000', 'NIAGARA WHEATFIELD SCH. DIST.'], ['400701060004', 'Colonial Village Elem.'], ['400701060009', 'Edward Town Middle'], ['400701060005', 'Errick Road Elementary'], ['400701060003', 'Tuscarora Elementary'], ['400701060002', 'West Street Elementary'], ['400900010000', 'NORTH TONAWANDA SCH. DIST.'], ['400900010003', 'Drake'], ['400900010007', 'Meadow'], ['400900010012', 'North Tonawanda Middle'], ['4

```

In [ ]: