Calculating graduation rates and its percent change in New York State education 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), 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 [45]:  tester_16 = agate.TypeTester(limit=200,force={
              'NRC_DESC': agate.Text(),
              'NRC_CODE': agate.Number(),
                  'LEA_BEDS': agate.Text(),
                  'COUNTY_CODE': agate.Text(),
                  'AGGREGATION_CODE': agate.Text(),
                  'AGGREGATION_INDEX': agate.Text(),
                  'COUNTY_NAME': agate.Text(),
                  'BOCES_CODE': agate.Text(),
                  'BOCES_NAME': agate.Text(),
                  'LEA_NAME': agate.Text(),
                  'MEMBERSHIP_DESC': 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 [49]: def get_percent_dictionary(file_name, tester):
             year = file_name[-8:-4]
             print(year)
             year4 = int(year) - 4
             year_stat = "{0} Total Cohort - 4 Year Outcome".format(year4)
             print(year_stat)
             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_NAME'] in ['ERIE', 'NIAGAR
         A']).where(lambda row: row['SUBGROUP_NAME'] in ['All Students']).where(lambda row:
         row['AGGREGATION_TYPE'] in ['School']).where(lambda row: row['MEMBERSHIP_DESC'] in
         [year_stat])
             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.
             # Form the data structure for each school with beds code as a key
             schools_info = {}
             for row in erie_niagara.rows:
                 """name = School name
                 grad_pct = graduation rate for 4 year cohort
                 district = district name
                 county = county"""
                 schools_info[row['AGGREGATION_CODE']] = {'name': row['AGGREGATION_NAME'], '
         grad_pct': row['REG_PCT'][:-1], 'district': row['LEA_NAME'], 'county': row['COUNTY_
         NAME'] }
             return schools_info
```

```
In [58]: year_17 = get_percent_dictionary('GRAD_RATE_AND_OUTCOMES_2016.csv', tester_16)
         #Note just changing so I don't have to rechange everything
```

```
2016
2012 Total Cohort - 4 Year Outcome
68
```

```
In [59]: year_16 = get_percent_dictionary('GRAD_RATE_AND_OUTCOMES_2015.csv',tester_16)
```

```
2015
2011 Total Cohort - 4 Year Outcome
68
```

```
In [7]: year_14 = get_percent_dictionary('3-8_ELA_AND_MATH_2014.csv',tester_15)
```

```
1859
```

```
In [8]: year_13 = get_percent_dictionary('3-8_ELA_AND_MATH_2013.csv',tester_15)
```

```
1888
```

Example of each year returning data for a specific school

```
In [60]: print(year_17['141800860044'])
         print(year_16['141800860044'])
```

```
{'name': 'GLOBAL CONCEPTS CHARTER SCHOOL', 'county': 'ERIE', 'grad_pct': '79', '
district': 'GLOBAL CONCEPTS CHARTER SCHOOL'}
{'name': 'GLOBAL CONCEPTS CHARTER SCHOOL', 'county': 'ERIE', 'grad_pct': '56', '
district': 'GLOBAL CONCEPTS CHARTER SCHOOL'}
```

Backfilling new 2017 schools

```
In [55]:  for school in year_16:
              if school not in year_15:
                  print('not in 2015 {0}'.format(year_16[school]['name']))
                  year_15[school] = {'name': year_16[school]['name'], 'county': year_16[schoo
          l]['county'], 'district': year_16[school]['district'], 'grad_pct': '-'}
```

```
not in 2015 NEWFANE SENIOR HIGH SCHOOL
not in 2015 NORTH TONAWANDA HIGH SCHOOL
not in 2015 LOCKPORT HIGH SCHOOL
not in 2015 LEWISTON PORTER SENIOR HIGH SCHOOL
not in 2015 BARKER JR/SR HIGH SCHOOL
not in 2015 NIAGARA FALLS HIGH SCHOOL
not in 2015 STARPOINT HIGH SCHOOL
not in 2015 NIAGARA-WHEATFIELD SR HIGH SCHOOL
not in 2015 WILSON HIGH SCHOOL
not in 2015 ROYALTON-HARTLAND HIGH SCHOOL
```

```python
In [10]:  for school in year_17:
              if school not in year_16:
                  print('not in 2016 {0}'.format(year_17[school]['name']))
                  year_16[school] = {'name': year_17[school]['name'], 'county': year_17[schoo
          l]['county'], 'district': year_17[school]['district'], 'grad_pct': '-'}
              if school not in year_15:
                  print('not in 2015 {0}'.format(year_17[school]['name']))
                  year_15[school] = {'name': year_17[school]['name'], 'county': year_17[schoo
          l]['county'], 'district': year_17[school]['district'], 'math': {'totals': [], 'prof
          icient': [], 'classes': [], 'total_percent': '-'}, 'ela': {'totals': [], 'proficien
          t': [], 'classes': [], 'total_percent': '-'}}
              if school not in year_14:
                  print('not in 2014 {0}'.format(year_17[school]['name']))
                  year_14[school] = {'name': year_17[school]['name'], 'county': year_17[schoo
          l]['county'], 'district': year_17[school]['district'], 'math': {'totals': [], 'prof
          icient': [], 'classes': [], 'total_percent': '-'}, 'ela': {'totals': [], 'proficien
          t': [], 'classes': [], 'total_percent': '-'}}
              if school not in year_13:
                  print('not in 2013 {0}'.format(year_17[school]['name']))
                  year_13[school] = {'name': year_17[school]['name'], 'county': year_17[schoo
          l]['county'], 'district': year_17[school]['district'], 'math': {'totals': [], 'prof
          icient': [], 'classes': [], 'total_percent': '-'}, 'ela': {'totals': [], 'proficien
          t': [], 'classes': [], 'total_percent': '-'}}
          for school in year_16:
              if school not in year_17:
                  print ('Not in 2017 {0} {1}'.format(school, year_16[school]['name']))
              if school not in year_15:
                  print ('Not in 2015 {0} {1}'.format(school, year_16[school]['name']))
          print('checking 2015')
          for school in year_15:
              if school not in year_17:
                  print ('Not in 2017 {0} {1}'.format(school, year_15[school]['name']))
              if school not in year_16:
                  print ('Not in 2016 {0} {1}'.format(school, year_15[school]['name']))
```

```
        not in 2016 NEWCOMER ACADEMY AT LAFAYETTE
        not in 2015 NEWCOMER ACADEMY AT LAFAYETTE
        not in 2014 NEWCOMER ACADEMY AT LAFAYETTE
        not in 2013 NEWCOMER ACADEMY AT LAFAYETTE
        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 2013 WEST BUFFALO CHARTER SCHOOL
        not in 2016 KENMORE WEST SENIOR HIGH SCHOOL
        not in 2015 KENMORE WEST SENIOR HIGH SCHOOL
        not in 2014 KENMORE WEST SENIOR HIGH SCHOOL
        not in 2013 KENMORE WEST SENIOR HIGH 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 2016 ALDEN INTERMEDIATE SCHOOL
        not in 2015 ALDEN INTERMEDIATE SCHOOL
        not in 2014 ALDEN INTERMEDIATE SCHOOL
        not in 2013 ALDEN INTERMEDIATE SCHOOL
        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 2017 142601030022 KENMORE MIDDLE SCHOOL
        Not in 2017 140101060003 ALDEN PRIMARY AT TOWNLINE
        Not in 2017 140600010039 DR MARTIN LUTHER KING, JR MULTICULTURAL INSTITUTE
        Not in 2017 142601030002 ALEXANDER HAMILTON ELEMENTARY SCHOOL
        Not in 2017 140600010107 LAFAYETTE HIGH SCHOOL
        Not in 2017 142601030019 THEODORE ROOSEVELT ELEMENTARY SCHOOL
        checking 2015
        Not in 2017 142601030022 KENMORE MIDDLE SCHOOL
        Not in 2017 140101060003 ALDEN PRIMARY AT TOWNLINE
        Not in 2017 140600010039 DR MARTIN LUTHER KING, JR MULTICULTURAL INSTITUTE
        Not in 2017 142601030002 ALEXANDER HAMILTON ELEMENTARY SCHOOL
        Not in 2017 140600010107 LAFAYETTE HIGH SCHOOL
        Not in 2017 400301060006 LEWISTON PORTER SENIOR HIGH SCHOOL
        Not in 2016 400301060006 LEWISTON PORTER SENIOR HIGH SCHOOL
        Not in 2017 142801060001 POTTERS ROAD SCHOOL
        Not in 2016 142801060001 POTTERS ROAD SCHOOL
        Not in 2017 142601030019 THEODORE ROOSEVELT ELEMENTARY SCHOOL
```

Grab clean school/district names and which schools are charters

```python
In [57]: 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']
```

Out[57]: 'Alden Intermediate'

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

```
In [68]: erie_list = {}
         niagara_list = {}
         charter_list = {}
         missing_list = []
         for key, value in year_17.items():
             try:
                 if key not in charters:
                     if value['county'] == 'ERIE':

                         erie_list[key] = {'name': clean_names[key],'dist_key': value['name
         ']}
                     else:
                         niagara_list[key] = {'name': clean_names[key],'dist_key': value['na
         me']}
                 else:
                     charter_list[key] = clean_names[key]
             except KeyError:
                 """print('$$$$Missing school in school_name_dictionary$$$')
                 print (value['name'])
                 print(key)"""
                 missing_list.append({'BEDS': key, 'school': value['name'], 'clean_school':
         '', 'charter': ''})
         print(niagara_list)
         print(missing_list)
         """with open('../school_name_dictionary.csv', 'a') as csvfile:
             fieldnames = ['BEDS', 'school', 'clean_school', 'charter']
             writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
             for row in missing_list:
                 writer.writerow(row)"""
```

{}
[{'charter': '', 'school': 'LEONARDO DA VINCI HIGH SCHOOL', 'clean_school': '',
'BEDS': '140600010128'}, {'charter': '', 'school': 'NEWFANE SENIOR HIGH SCHOOL',
'clean_school': '', 'BEDS': '4.00601E+11'}, {'charter': '', 'school': 'GRAND ISL
AND SENIOR HIGH SCHOOL', 'clean_school': '', 'BEDS': '141501060004'}, {'charter
': '', 'school': 'ORCHARD PARK HIGH SCHOOL', 'clean_school': '', 'BEDS': '142301
060006'}, {'charter': '', 'school': 'DEPEW HIGH SCHOOL', 'clean_school': '', 'BE
DS': '140707030003'}, {'charter': '', 'school': 'LAFAYETTE HIGH SCHOOL', 'clean_
school': '', 'BEDS': '140600010107'}, {'charter': '', 'school': 'NORTH TONAWANDA
HIGH SCHOOL', 'clean_school': '', 'BEDS': '4.009E+11'}, {'charter': '', 'school
': 'WILLIAMSVILLE SOUTH HIGH SCHOOL', 'clean_school': '', 'BEDS': '140203060004
'}, {'charter': '', 'school': 'LOCKPORT HIGH SCHOOL', 'clean_school': '', 'BEDS
': '4.004E+11'}, {'charter': '', 'school': 'JOHN F KENNEDY SENIOR HIGH SCHOOL',
'clean_school': '', 'BEDS': '140709030004'}, {'charter': '', 'school': 'WEST SEN
ECA EAST SENIOR HIGH SCHOOL', 'clean_school': '', 'BEDS': '142801060016'}, {'cha
rter': '', 'school': 'HUTCHINSON CENTRAL TECH HIGH SCHOOL', 'clean_school': '',
'BEDS': '140600010105'}, {'charter': '', 'school': 'WEST SENECA WEST SENIOR HIGH
SCHOOL', 'clean_school': '', 'BEDS': '142801060010'}, {'charter': '', 'school':
'LEWISTON PORTER SENIOR HIGH SCHOOL', 'clean_school': '', 'BEDS': '4.00301E+11
'}, {'charter': '', 'school': 'CHEEKTOWAGA HIGH SCHOOL', 'clean_school': '', 'BE
DS': '140701060006'}, {'charter': '', 'school': 'MIDDLE EARLY COLLEGE HIGH SCHOO
L', 'clean_school': '', 'BEDS': '140600010135'}, {'charter': '', 'school': 'SOUT
H PARK HIGH SCHOOL', 'clean_school': '', 'BEDS': '140600010110'}, {'charter':
'', 'school': 'RIVERSIDE INSTITUTE OF TECHNOLOGY', 'clean_school': '', 'BEDS': '
140600010108'}, {'charter': '', 'school': 'HEALTH SCIENCES CHARTER SCHOOL', 'cle
an_school': '', 'BEDS': '140600860961'}, {'charter': '', 'school': 'WILLIAMSVILL
E EAST HIGH SCHOOL', 'clean_school': '', 'BEDS': '140203060013'}, {'charter':
'', 'school': 'BARKER JR/SR HIGH SCHOOL', 'clean_school': '', 'BEDS': '4.01301E+
11'}, {'charter': '', 'school': 'MARYVALE HIGH SCHOOL', 'clean_school': '', 'BED
S': '140702030006'}, {'charter': '', 'school': 'MCKINLEY VOC HIGH SCHOOL', 'clea
n_school': '', 'BEDS': '140600010098'}, {'charter': '', 'school': 'FRONTIER SENI
OR HIGH SCHOOL', 'clean_school': '', 'BEDS': '141604060008'}, {'charter': '', 's
chool': 'ORACLE CHARTER SCHOOL', 'clean_school': '', 'BEDS': '140600860868'}, {'
charter': '', 'school': 'NIAGARA FALLS HIGH SCHOOL', 'clean_school': '', 'BEDS':
'4.008E+11'}, {'charter': '', 'school': 'HAMBURG HIGH SCHOOL', 'clean_school':
'', 'BEDS': '141601060007'}, {'charter': '', 'school': 'STARPOINT HIGH SCHOOL',
'clean_school': '', 'BEDS': '4.01001E+11'}, {'charter': '', 'school': 'NIAGARA-W
HEATFIELD SR HIGH SCHOOL', 'clean_school': '', 'BEDS': '4.00701E+11'}, {'charter
': '', 'school': 'IROQUOIS SENIOR HIGH SCHOOL', 'clean_school': '', 'BEDS': '141
301060006'}, {'charter': '', 'school': 'EAST AURORA HIGH SCHOOL', 'clean_school
': '', 'BEDS': '140301030005'}, {'charter': '', 'school': 'ALDEN SENIOR HIGH SCH
OOL', 'clean_school': '', 'BEDS': '140101060006'}, {'charter': '', 'school': 'AM
HERST CENTRAL HIGH SCHOOL', 'clean_school': '', 'BEDS': '140201060002'}, {'chart
er': '', 'school': 'PS 42 OCCUPATIONAL TRAINING CTR', 'clean_school': '', 'BEDS
': '140600010042'}, {'charter': '', 'school': 'SWEET HOME SENIOR HIGH SCHOOL', '
clean_school': '', 'BEDS': '140207060006'}, {'charter': '', 'school': 'AKRON HIG
H SCHOOL', 'clean_school': '', 'BEDS': '142101040002'}, {'charter': '', 'school
': 'CLEVELAND HILL HIGH SCHOOL', 'clean_school': '', 'BEDS': '140703020003'}, {'
charter': '', 'school': 'WILLIAMSVILLE NORTH HIGH SCHOOL', 'clean_school': '', '
BEDS': '140203060010'}, {'charter': '', 'school': 'CLARENCE SENIOR HIGH SCHOOL',
'clean_school': '', 'BEDS': '140801060006'}, {'charter': '', 'school': 'BURGARD
VOC HIGH SCHOOL', 'clean_school': '', 'BEDS': '140600010101'}, {'charter': '', '
school': 'LAKE SHORE SENIOR HIGH SCHOOL', 'clean_school': '', 'BEDS': '141401060
003'}, {'charter': '', 'school': 'LACKAWANNA HIGH SCHOOL', 'clean_school': '', '
BEDS': '141800010008'}, {'charter': '', 'school': 'WILSON HIGH SCHOOL', 'clean_s
chool': '', 'BEDS': '4.01501E+11'}, {'charter': '', 'school': 'EMERSON SCHOOL OF
HOSPITALITY', 'clean_school': '', 'BEDS': '140600010104'}, {'charter': '', 'scho
ol': 'BENNETT HIGH SCHOOL', 'clean_school': '', 'BEDS': '140600010099'}, {'chart
er': '', 'school': 'PS 84', 'clean_school': '', 'BEDS': '140600010084'}, {'chart
er': '', 'school': 'GRIFFITH INST HIGH SCHOOL', 'clean_school': '', 'BEDS': '141
101060001'}, {'charter': '', 'school': 'LANCASTER HIGH SCHOOL', 'clean_school':
'', 'BEDS': '141901060008'}, {'charter': '', 'school': 'ROYALTON-HARTLAND HIGH S
CHOOL', 'clean_school': '', 'BEDS': '4.01201E+11'}, {'charter': '', 'school': 'E
AST HIGH SCHOOL', 'clean_school': '', 'BEDS': '140600010307'}]

Out[68]: "with open('../school_name_dictionary.csv', 'a') as csvfile:\n   fieldnames =
         ['BEDS', 'school', 'clean_school', 'charter']\n   writer = csv.DictWriter(csvfi
         le, fieldnames=fieldnames)\n   for row in missing_list:\n   writer.writero
         w(row)"

In [18]:
```python
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

Out[68]: "with open('../school_name_dictionary.csv', 'a') as csvfile:\n   fieldnames =
         ['BEDS', 'school', 'clean_school', 'charter']\n   writer = csv.DictWriter(csvfi
         le, fieldnames=fieldnames)\n   for row in missing_list:\n   writer.writero
         w(row)"

```
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[distr
          ict]['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_perce
          nt'],year_16[district]['math']['total_percent'],year_17[district]['math']['total_pe
          rcent'], 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 c
          hange.
                      for school in sorted_schools:
                          pc_school_ela_15_17 = percent_change(year_17[school]['ela']['to
          tal_percent'],year_15[school]['ela']['total_percent'])
                          pc_school_ela_16_17 = percent_change(year_17[school]['ela']['to
          tal_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[schoo
          l], 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[distr
          ict])
                  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.

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

```
In [24]:  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':
                  sorted_county = sorted(county_list, key= lambda district: county_list[distr
          ict]['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]['na
          me']))
                          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 perce
          nt change.
                          for school in sorted_schools:
                              try:
                                  #print('writing the following {0}'.format(schools_info[scho
          ol]))
                                  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 perce
          nt change.
                          for school in sorted_schools:
                              try:
                                  #print('writing the following {0}'.format(schools_info[scho
          ol]))
                                  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:
                  sorted_county = sorted(county_list, key= lambda district: county_list[distr
          ict])
                  for district in sorted_county:
                      county_count += 1
                      print('*** starting district {0}'.format(county_list[district]))
                      ordered.append([district,county_list[district]])
              clean_json[county]['ordered_schools'] = ordered
          eried_list = export_county_schools(erie_list, 'Erie')
          niagarad_list = export_county_schools(niagara_list, 'Niagara')
          chartered_list = export_county_schools(charter_list, 'Charters')
```

```
*** 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
```

```python
In [25]: 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_proffficient = (year_info['math']['proficient'][i] / year_info['math']['totals'][i]) * 100
                    math_year[clean_class] = str(float("{0:.1f}".format(percent_proffficient)))
                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_proffficient = (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']))
```

```python
In [27]: 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'] = {}
                 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')
                 schools[school[0]] = school_info
             clean_json[group]['schools'] = schools

         print(clean_json['Erie']['schools']['140101060007'])
         print(clean_json['Erie']['ordered_schools'])
         import json
         with open('data.json', 'w') as output:
             json.dump(clean_json, output)
```

**Starting Buffalo
**Starting Niagara
**Starting Charters
**Starting Erie
{'ela': {'2017': {'Grade 5': '30.0', 'Grade 4': '59.4', 'Grade 3': '70.1', 'tota
l': '53.7'}}, 'district': 'ALDEN CENTRAL SCHOOL DISTRICT', 'name': 'Alden Interm
ediate', 'math': {'2017': {'Grade 5': '38.4', 'Grade 4': '54.8', 'Grade 3': '76.
5', 'total': '57.3'}}}}
[['142101040000', 'AKRON SCHOOL DISTRICT'], ['142101040001', 'Akron Elementary
'], ['142101040003', 'Akron Middle'], ['140101060000', 'ALDEN SCHOOL DISTRICT'],
['140101060007', 'Alden Intermediate'], ['140101060005', 'Alden Middle'], ['1402
01060000', 'AMHERST'], ['140201060001', 'Amherst Middle'], ['140201060005', 'Sma
llwood'], ['140201060006', 'Windermere'], ['140701060000', 'CHEEKTOWAGA SCHOOL D
ISTRICT'], ['140701060007', 'Cheektowaga Middle'], ['140701060008', 'Pine Hill
'], ['140701060004', 'Union East'], ['140801060000', 'CLARENCE SCHOOL DISTRICT
'], ['140801060007', 'Clarence Center Elem.'], ['140801060008', 'Clarence Middle
'], ['140801060002', 'Harris Hill Elementary'], ['140801060003', 'Ledgeview Elem
entary'], ['140801060005', 'Sheridan Hill Elementary'], ['140703020000', 'CLEVEL
AND HILL SCHOOL DISTRICT'], ['140703020002', 'Cleveland Hill Elem.'], ['14070302
0004', 'Cleveland Hill Middle'], ['140707030000', 'DEPEW SCHOOL DISTRICT'], ['14
0707030005', 'Cayuga Heights Elem.'], ['140707030004', 'Depew Middle'], ['140301
030000', 'EAST AURORA SCHOOL DISTRICT'], ['140301030004', 'East Aurora Middle'],
['140301030001', 'Parkdale Elementary'], ['141201060000', 'EDEN SCHOOL DISTRICT
'], ['141201060005', 'Eden Elementary'], ['141201060001', 'Eden Junior-Senior Hi
gh'], ['141604060000', 'FRONTIER SCHOOL DISTRICT'], ['141604060003', 'Big Tree E
lementary'], ['141604060004', 'Blasdell Elementary'], ['141604060005', 'Cloverba
nk Elementary'], ['141604060007', 'Frontier Middle'], ['141604060006', 'Pinehurs
t Elementary'], ['141501060000', 'GRAND ISLAND SCHOOL DISTRICT'], ['141501060005
', 'Connor Middle'], ['141501060001', 'Huth Road'], ['141501060003', 'Kaegebein
'], ['141601060000', 'HAMBURG SCHOOL DISTRICT'], ['141601060001', 'Armor Element
ary'], ['141601060002', 'Boston Valley Elem.'], ['141601060003', 'Charlotte Ave.
Elem.'], ['141601060006', 'Hamburg Middle'], ['141601060004', 'Union Pleasant El
em.'], ['141701040000', 'HOLLAND SCHOOL DISTRICT'], ['141701040002', 'Brumsted E
lementary'], ['141701040001', 'Holland Junior/Senior High'], ['141301060000', 'I
ROQUOIS SCHOOL DISTRICT'], ['141301060007', 'Elma Primary'], ['141301060002', 'I
roquois Intermediate'], ['141301060005', 'Iroquois Middle'], ['141301060003', 'M
arilla Primary'], ['141301060004', 'Wales Primary'], ['142601030000', 'KEN-TON S
CHOOL DISTRICT'], ['142601030020', 'Edison Elementary'], ['142601030023', 'Frank
lin Elementary'], ['142601030003', 'Franklin Middle'], ['142601030013', 'Holmes
Elementary'], ['142601030024', 'Hoover Elementary'], ['142601030011', 'Hoover Mi
ddle'], ['142601030025', 'Kenmore East High School'], ['142601030026', 'Kenmore
West High School'], ['142601030006', 'Lindbergh Elementary'], ['141800010000', '
LACKAWANNA SCHOOL DISTRICT'], ['141800010005', 'Lackawanna Middle'], ['141800010
011', 'Martin Road Elementary'], ['141401060000', 'LAKE SHORE SCHOOL DISTRICT'],
['141401060001', 'Highland Elementary'], ['141401060009', 'Lake Shore Middle'],
['141401060008', 'Schmidt Elementary'], ['141401060004', 'Waugh Elementary'], ['
141901060000', 'LANCASTER SCHOOL DISTRICT'], ['141901060004', 'Como Park Element
ary'], ['141901060005', 'Court Street Elementary'], ['141901060006', 'Hillview E
lementary'], ['141901060007', 'Lancaster Middle'], ['141901060001', 'Sciole Elem
entary'], ['141901060010', 'William Street'], ['140702030000', 'MARYVALE SCHOOL
DISTRICT'], ['140702030004', 'Maryvale Intermediate'], ['140702030005', 'Maryval
e Middle'], ['142201040000', 'NORTH COLLINS SCHOOL DISTRICT'], ['142201040002',
'North Collins Elementary'], ['142201040001', 'North Collins Junior-Senior'], ['
142301060000', 'ORCHARD PARK SCHOOL DISTRICT'], ['142301060003', 'Eggert Road El
ementary'], ['142301060007', 'Ellicott Road Elementary'], ['142301060002', 'Orch
ard Park Middle'], ['142301060005', 'South Davis Elementary'], ['142301060001',
'Windom Elementary'], ['140709030000', 'SLOAN SCHOOL DISTRICT'], ['140709030003
', 'JFK Middle'], ['140709030002', 'Wilson Elementary'], ['141101060000', 'SPRIN
GVILLE-GRIFFITH SCH. DIST.'], ['141101060002', 'Colden Elementary'], ['141101060
004', 'Griffith Institute Middle'], ['141101060003', 'Springville Elementary'],
['140207060000', 'SWEET HOME SCHOOL DISTRICT'], ['140207060002', 'Glendale Eleme
ntary'], ['140207060008', 'Heritage Heights Elem.'], ['140207060003', 'Maplemere
Elementary'], ['140207060005', 'Sweet Home Middle'], ['140207060007', 'Willow Ri
dge Elementary'], ['142500010000', 'TONAWANDA SCHOOL DISTRICT'], ['142500010003

In [ ]: