# Data

December 12, 2018

Tristan Moser

## 1 Data Combining and Cleaning

All of the relevant CSV files found in this document were provided by the Colorado Department of Education' website: https://www.cde.state.co.us/cdereval.

The goal of this file is to read in all pertinent data, gather and clean relevant variables, and then combine the resulting tables into one CSV file. To accomplish this, I rely heavily on the Python Pandas library as it allows for easy manipulation of CSV files.

```
In [1]: import pandas as pd
        pd.options.mode.chained_assignment = None
```

Before reading in these files, I made some adjustments in Microsoft Excel so that the documents were able to be read in successfully and also so that I could get an idea as to how the files were organized. Because some datasets are different across the years, I made comments outlining if one file should be treated differently than the others.

```
In [2]: #Teacher Salaries
        sal_17 = pd.read_csv('Data/2017-18 Average Salaries for Teachers (1).csv')
        sal_16 = pd.read_csv('Data/2016-17AverageTeacherSalary (1).csv')
        sal_15 = pd.read_csv('Data/15-16Average Salaries for Teachers.csv')
        sal_14 = pd.read_csv('Data/14-15Teacher FTE and Average Salary.csv')
        sal_13 = pd.read_csv('Data/13-14Average Teacher Salary Report.csv')

        #Teacher ethnicity/gender
        tcomp_17 = pd.read_csv('Data/2017-18 Teachers by Gender and Race.csv')
        tcomp_16 = pd.read_csv('Data/16-17Teachers by Gender and Race (1).csv')
        tcomp_15 = pd.read_csv('Data/15-16- Revised Count of Teachers by District, Ethnicity and
        tcomp_14 = pd.read_csv('Data/14-15- Teachers by Ethnicity,Race and Gender.csv')
        tcomp_13 = pd.read_csv('Data/13-14Count of Teachers by District, Ethnicity and Gender.cs

        #Student to Teacher Ratios
        str_17 = pd.read_csv('Data/2017-18 Student Teacher Ratios.csv')
        str_16 = pd.read_csv('Data/16-17Pupil Teacher Ratio by School (1).csv')
        str_15 = pd.read_csv('Data/15-16-Student Teacher Ratios.csv')
        str_14 = pd.read_csv('Data/2014-15StudentTeacherRatioBySchool.csv')
```

```
str_13 = pd.read_csv('Data/13-14Pupil Teacher Ratios by School.csv')

#Graduation Statistics
grad_17 = pd.read_csv('Data/2017-18-Grad-District-Race.csv')
grad_16 = pd.read_csv('Data/16_cohort4_graduates_and_completers_by_district_gender_and_r
grad_15 = pd.read_csv('Data/15_cohort4_graduates_completers_by_district_gender_and_racee
grad_14 = pd.read_csv('Data/14_cohort4_graduates_and_completers_by_district_gender_and_r
grad_13 = pd.read_csv('Data/13_cohort4_graduates_and_completers_by_district_gender_and_r

#Number of students receiving free/reduced lunch
#Drop County Total rows
lunch_17 = pd.read_csv('Data/2017-18_K12_FRL_byDistrict.csv')
lunch_16 = pd.read_csv('Data/2016-17_K_12_FRL_byCountyDistrict.csv')
lunch_15 = pd.read_csv('Data/2015_16_K_12FreeandReducedLunchEligibiltybyCountyDistrict.c
lunch_14 = pd.read_csv('Data/2014_2015_K_12FreeandReducedLunchEligibiltybyCountyDistrict
lunch_13 = pd.read_csv('Data/2013_2014_K_12FreeandReducedLunchEligibiltybyCountyDistrict

#Student ethnicity stats
scomp_17 = pd.read_csv('Data/2017-18_Membership_RaceEthnicity_PctMinority_byCountyDistri
#Sum male and female columns
scomp_16 = pd.read_csv('Data/2016-17_pupilmembershipbycountydistrictethnicityandgender.c
scomp_15 = pd.read_csv('Data/2015_16_PupilMembershipbyCountyDistrictRaceEthnicityandPctM
scomp_14 = pd.read_csv('Data/2014_15_PupilMembershipbyCountyDistrictRaceEthnicityandPctM
scomp_13 = pd.read_csv('Data/2013_14_PupilMembershipbyCountyDistrictRaceEthnicityandPctM

#Enrollment in AP classes
#Only take total rows
ap_16 = pd.read_csv('Data/2016-2017_advanced_placement_by_course_and_district.csv')
ap_15 = pd.read_csv('Data/2015-2016_advanced_placement_by_course_and_district.csv')
ap_14 = pd.read_csv('Data/2014-2015_advanced_placement_by_course_and_district.csv')
ap_13 = pd.read_csv('Data/2013-2014_advanced_placement_by_course_and_district.csv')
```

In [3]:
```
#Teacher Salary datasets

#Gather only relevent columns
sal_cols = ['Organization Code','Orgnazation Name','Total FTE','Average Salary']

#Correct column name to match the other years
sal_13['Average Salary'] = sal_13['Average Teacher Salary']

#Add Year column for each dataset
sal_17 = sal_17[sal_cols]
sal_17['Year'] = '17'
sal_16 = sal_16[sal_cols]
sal_16['Year'] = '16'
sal_15 = sal_15[sal_cols]
sal_15['Year'] = '15'
sal_14 = sal_14[sal_cols]
```

```
sal_14['Year'] = '14'
sal_13 = sal_13[sal_cols]
sal_13['Year'] = '13'

#Combine yearly data sets into one Salary DataFrame
sal_data = sal_17.append(sal_16, ignore_index=True)
sal_data = sal_data.append(sal_15, ignore_index=True)
sal_data = sal_data.append(sal_14, ignore_index=True)
sal_data = sal_data.append(sal_13, ignore_index=True)

#Correct datatypes of columns
#These were recorded as strings, but they need to be numeric values
sal_data['Average Salary'] = sal_data['Average Salary'].replace({'\$':"",",":""},regex=T
sal_data['Total FTE'] = sal_data['Total FTE'].replace({'\$':"",",":""},regex=True)
sal_data['Average Salary'] = sal_data['Average Salary'].astype(float)
sal_data['Total FTE'] = sal_data['Total FTE'].astype(float)

#Drop all rows missing any values
sal_data = sal_data.dropna()

sal_data['Organization Code'] = sal_data['Organization Code'].astype(int)
```

In [4]:
```
#Review Salary DataFrame
sal_data.head()
```

Out[4]:

| | Organization Code | Orgnazation Name | Total FTE | Average Salary | \ |
|---|---|---|---|---|---|
| 0 | 10 | MAPLETON 1 | 358.47 | 56410.0 | |
| 1 | 20 | ADAMS 12 FIVE STAR SCHOOLS | 1969.88 | 59511.0 | |
| 2 | 30 | ADAMS COUNTY 14 | 376.35 | 57394.0 | |
| 3 | 40 | SCHOOL DISTRICT 27J | 793.94 | 49488.0 | |
| 4 | 50 | BENNETT 29J | 62.22 | 39148.0 | |

| | Year |
|---|---|
| 0 | 17 |
| 1 | 17 |
| 2 | 17 |
| 3 | 17 |
| 4 | 17 |

In [5]:
```
#Teacher Ethnicity datasets

#The data file for 2017-2018 has all of the columns that are relevant so it was chosen a
tcomp_cols = list(tcomp_17.columns)

#Correct column name differences
tcomp_14['Organization Code'] = tcomp_14['District Code']
tcomp_14['Orgnazation Name'] = tcomp_14['District Name']
tcomp_13['Organization Code'] = tcomp_13['District Code']
```

```
tcomp_13['Orgnazation Name'] = tcomp_13['District Name']

#Add Year column for each dataset
tcomp_17 = tcomp_17[tcomp_cols]
tcomp_17['Year'] = '17'
tcomp_16 = tcomp_16[tcomp_cols]
tcomp_16['Year'] = '16'
tcomp_15 = tcomp_15[tcomp_cols]
tcomp_15['Year'] = '15'
tcomp_14 = tcomp_14[tcomp_cols]
tcomp_14['Year'] = '14'
tcomp_13 = tcomp_13[tcomp_cols]
tcomp_13['Year'] = '13'

#Combine yearly data sets into one Teacher Ethnicity DataFrame
tcomp_data = tcomp_17.append(tcomp_16, ignore_index=True)
tcomp_data = tcomp_data.append(tcomp_15, ignore_index=True)
tcomp_data = tcomp_data.append(tcomp_14, ignore_index=True)
tcomp_data = tcomp_data.append(tcomp_13, ignore_index=True)

#Drop all rows missing any values
tcomp_data = tcomp_data.dropna()

#Correct datatypes of columns
#These were recorded as strings, but they need to be numeric values
tcomp_data['T_Total_Count'] = tcomp_data['T_Total_Count'].replace({'\,':''},regex=True).
tcomp_data['T_Whi_T'] = tcomp_data['T_Whi_T'].replace({'\,':''},regex=True).astype(int)
tcomp_data['T_Whi_M'] = tcomp_data['T_Whi_M'].replace({'\,':''},regex=True).astype(int)
tcomp_data['T_Whi_F'] = tcomp_data['T_Whi_F'].replace({'\,':''},regex=True).astype(int)
tcomp_data['T_Pac_M'] = tcomp_data['T_Pac_M'].replace({'\,':''},regex=True).astype(int)
tcomp_data['T_Pac_F'] = tcomp_data['T_Pac_F'].replace({'\,':''},regex=True).astype(int)
tcomp_data['T_Pac_T'] = tcomp_data['T_Pac_T'].replace({'\,':''},regex=True).astype(int)


tcomp_data['Organization Code'] = tcomp_data['Organization Code'].astype(int)

tcomp_data = tcomp_data.reset_index().drop(['index'],axis=1)

#Set New Minority Percentage variable
tcomp_data['Teacher Pct Minority'] = 0.0
tcomp_data['Teacher Count'] = tcomp_data['T_Total_Count']

#Calculate minority percentages
for ii in range(len(tcomp_data)):
    tcomp_data['Teacher Pct Minority'][ii] = 1 - (tcomp_data['T_Whi_T'][ii]/tcomp_data['

#Only take new, updated variables
tcomp_data = tcomp_data[['Organization Code','Orgnazation Name','Teacher Pct Minority','
```

In [6]:
```
#Review Teacher Ethnicity DataFrame
tcomp_data.head()
```

Out[6]:

| | Organization Code | Orgnazation Name | Teacher Pct Minority | \ |
|---|---|---|---|---|
| 0 | 10 | MAPLETON 1 | 0.162162 | |
| 1 | 20 | ADAMS 12 FIVE STAR SCHOOLS | 0.105957 | |
| 2 | 30 | ADAMS COUNTY 14 | 0.243108 | |
| 3 | 40 | SCHOOL DISTRICT 27J | 0.036675 | |
| 4 | 50 | BENNETT 29J | 0.063492 | |

| | Teacher Count | Year |
|---|---|---|
| 0 | 370 | 17 |
| 1 | 2048 | 17 |
| 2 | 399 | 17 |
| 3 | 818 | 17 |
| 4 | 63 | 17 |

In [7]:
```
# Student to Teacher Ratio datasets

#Create Uniform Column names
str_17['Organization Code'] = str_17['Distr Code']
str_17['Orgnazation Name'] = str_17['Distr Name']
str_17['Pupil/Teacher Ratio'] = str_17['Pupil/ Teacher FTE Ratio']
str_16['Organization Code'] = str_16['District Code']
str_16['Orgnazation Name'] = str_16['District Name']
str_16['Pupil/Teacher Ratio'] = str_16['Pupil/Teacher FTE Ratio']
str_15['Organization Code'] = str_15['District Code']
str_15['Orgnazation Name'] = str_15['District Name']
str_14['Organization Code'] = str_14['District Code']
str_14['Orgnazation Name'] = str_14['District Name']
str_13['Organization Code'] = str_13['DISTRICT CODE']
str_13['Orgnazation Name'] = str_13['DISTRICT NAME']
str_13['Pupil/Teacher Ratio'] = str_13['PUPIL/TEACHER RATIO']

#These datasets were organized by schools within the districts
#I performed summing operations so that the data was organized by school district
dataframes = [str_13,str_14,str_15,str_16,str_17]
str_data = pd.DataFrame(data={'Orgnazation Name':['test'],'Pupil/Teacher Ratio':['test']
for dataframe in range(len(dataframes)):
    districts = []
    totals = []
    year = []
    for district in range(len(dataframes[dataframe]['Orgnazation Name'].unique())):
        year.append(str(dataframe + 13))
        totals.append((dataframes[dataframe][dataframes[dataframe]['Orgnazation Name']==
        districts.append(dataframes[dataframe]['Orgnazation Name'].unique()[district])
    str_data = str_data.append(pd.DataFrame(data={'Orgnazation Name': districts, 'Pupil/
```

```
        #Drop Null entries
        str_data = str_data.dropna()
```

```
/Applications/Anaconda/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:28: RuntimeWar
/Applications/Anaconda/anaconda/lib/python3.6/site-packages/pandas/core/frame.py:6211: FutureWar
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

  sort=sort)
```

In [8]:
```
        #Review Student to Teacher Ratios DataFrame
        str_data.head()
```

Out[8]:

| | Orgnazation Name | Pupil/Teacher Ratio | Year |
|---|---|---|---|
| 1 | MAPLETON 1 | 19.566 | 13 |
| 2 | ADAMS 12 FIVE STAR SCHOOLS | 19.6706 | 13 |
| 3 | ADAMS COUNTY 14 | 17.0036 | 13 |
| 4 | BRIGHTON 27J | 20.5754 | 13 |
| 5 | BENNETT 29J | 18.4067 | 13 |

In [9]:
```
        #Graduation datasets

        #This dataset contained previous years as well
        #Isolate only the 17 graduation rate
        grad_17 = grad_17[grad_17['Ant_Year_of_Grad']=='2016-2017']

        #All relevant columns
        grad_cols = ['Organization Code','Organization Name','All Students Graduation Rate',
                     'Female Graduation Rate','Male Graduation Rate']

        #Columns that need Percentage datatype adjustments
        pct_cols = ['All Students Graduation Rate','Female Graduation Rate','Male Graduation Rat

        #Add Year values
        #Correct percent variables from string to floats
        grad_17 = grad_17[grad_cols].replace('\%','',regex=True)
        grad_17[pct_cols] = grad_17[pct_cols].astype(float)/100
        grad_17['Year'] = '17'
        grad_16 = grad_16[grad_cols].replace('\%','',regex=True)
        grad_16[pct_cols] = grad_16[pct_cols].astype(float)/100
        grad_16['Year'] = '16'
        grad_15 = grad_15[grad_cols].replace('\%','',regex=True)
        grad_15[pct_cols] = grad_15[pct_cols].astype(float)/100
        grad_15['Year'] = '15'
        grad_14 = grad_14[grad_cols].replace('\%','',regex=True)
```

```
grad_14[pct_cols] = grad_14[pct_cols].astype(float)/100
grad_14['Year'] = '14'
grad_13 = grad_13[grad_cols].replace('\%','',regex=True)
grad_13[pct_cols] = grad_13[pct_cols].astype(float)/100
grad_13['Year'] = '13'

#Combine grad datasets to one big DataFrame
grad_data = grad_17.append(grad_16, ignore_index=True)
grad_data = grad_data.append(grad_15, ignore_index=True)
grad_data = grad_data.append(grad_14, ignore_index=True)
grad_data = grad_data.append(grad_13, ignore_index=True)

#Drop all null values
grad_data = grad_data.dropna()

#Match convention from other DataFrames
grad_data['Organization Code'] = grad_data['Organization Code'].astype(int)
grad_data['Orgnazation Name'] = grad_data['Organization Name']
grad_data = grad_data.drop(['Organization Name'],axis=1)

#Only take graduation rates
grad_data = grad_data[['Organization Code','Orgnazation Name','All Students Graduation R
```

In [10]:
```
#Review Graduation DataFrame
grad_data.head()
```

Out[10]:

| | Organization Code | Orgnazation Name | All Students Graduation Rate | Female Graduation Rate | Male Graduation Rate | Year |
|---|---|---|---|---|---|---|
| 0 | 10 | MAPLETON 1 | 0.590 | 0.625 | 0.555 | 17 |
| 1 | 20 | ADAMS 12 FIVE STAR SCHOOLS | 0.836 | 0.889 | 0.784 | 17 |
| 2 | 30 | ADAMS COUNTY 14 | 0.656 | 0.702 | 0.613 | 17 |
| 3 | 40 | SCHOOL DISTRICT 27J | 0.774 | 0.833 | 0.724 | 17 |
| 4 | 50 | BENNETT 29J | 0.886 | 0.941 | 0.852 | 17 |

In [11]:
```
#Free and Reduced Lunch data
```

```
#Correct wrong naming convention
for ii in [lunch_17,lunch_16,lunch_15,lunch_14,lunch_13]:
    ii['Organization Code'] = ii['DISTRICT CODE']
    try:
        ii['Orgnazation Name'] = ii['DISTRICT NAME']
    except:
        ii['Orgnazation Name'] = ii['DISTRICT NAME\t']
lunch_17['K-12 COUNT'] = lunch_17['PK-12 COUNT']

#Isolate relevant columns
lunch_cols = ['Organization Code','Orgnazation Name','% FREE AND REDUCED','K-12 COUNT']

#Add Year column
#Correct percentages from string to floats
lunch_17 = lunch_17[lunch_cols].replace('\%','',regex=True)
lunch_17['Year'] = '17'
lunch_17['% FREE AND REDUCED'] = lunch_17['% FREE AND REDUCED'].astype(float)/100
lunch_16 = lunch_16[lunch_cols].replace('\%','',regex=True)
lunch_16['Year'] = '16'
lunch_16['% FREE AND REDUCED'] = lunch_16['% FREE AND REDUCED'].astype(float)/100
lunch_15 = lunch_15[lunch_cols].replace('\%','',regex=True)
lunch_15['Year'] = '15'
lunch_15['% FREE AND REDUCED'] = lunch_15['% FREE AND REDUCED'].astype(float)/100
lunch_14 = lunch_14[lunch_cols].replace('\%','',regex=True)
lunch_14['Year'] = '14'
lunch_14['% FREE AND REDUCED'] = lunch_14['% FREE AND REDUCED'].astype(float)/100
lunch_13 = lunch_13[lunch_cols].replace('\%','',regex=True)
lunch_13['Year'] = '13'
lunch_13['% FREE AND REDUCED'] = lunch_13['% FREE AND REDUCED'].astype(float)/100

#Combine datasets to make one big DataFrame
lunch_data = lunch_17.append(lunch_16, ignore_index=True)
lunch_data = lunch_data.append(lunch_15, ignore_index=True)
lunch_data = lunch_data.append(lunch_14, ignore_index=True)
lunch_data = lunch_data.append(lunch_13, ignore_index=True)

#Drop Null values
lunch_data = lunch_data.dropna()

lunch_data['Organization Code'] = lunch_data['Organization Code'].astype(int)
lunch_data['K-12 COUNT'] = lunch_data['K-12 COUNT'].replace({',':''},regex=True).astype
```

In [12]:
```
#Review Lunch DataFrame
lunch_data.tail()
```

Out[12]:

| | Organization Code | Orgnazation Name | \ |
|---|---|---|---|
| 1151 | 9130 | EXPEDITIONARY BOCES | |
| 1152 | 9170 | COLORADO DIGITAL BOCES | |

```
1154               8001                CHARTER SCHOOL INSTITUTE
1155               9000  Colorado School for the Deaf and Blind
1157               9999         COLORADO DETENTION CENTER TOTAL

      % FREE AND REDUCED  K-12 COUNT Year
1151              0.0000         387   13
1152              0.0000         358   13
1154              0.4647       10359   13
1155              0.6788         193   13
1157              0.2763         152   13
```

In [13]:
```
#Student Ethnicity Datasets

#Combine Male and Female entries to be one full value for each district
#This matches what the other years have
scomp_16 = scomp_16.dropna()
scomp_16 = scomp_16.replace({',':''},regex=True)
scomp_16['American Indian or Alaskan Native'] = scomp_16['F_American Indian or Alaskan
scomp_16['Asian'] = scomp_16['F_Asian'].astype(int)+scomp_16['M_Asian'].astype(int)
scomp_16['Black or African American'] = scomp_16['F_Black or African American'].astype(
scomp_16['Hispanic or Latino'] = scomp_16['F_Hispanic or Latino'].astype(int)+scomp_16[
scomp_16['White'] = scomp_16['F_White'].astype(int)+scomp_16['M_White'].astype(int)
scomp_16['Native Hawaiian or Other Pacific Islander'] = scomp_16['F_Native Hawaiian or
scomp_16['Two or More Races'] = scomp_16['F_Two or More Races'].astype(int)+scomp_16['M
scomp_16['Percent Minority'] = scomp_16['F_Percent Female'].replace({'\%':''},regex=Tru


sdataframes = [scomp_13,scomp_14,scomp_15,scomp_16,scomp_17]

#Isolate ethnic columns that need correcting
eth_cols = ['American Indian or Alaskan Native','Asian','Black or African American', 'H
            'White','Native Hawaiian or Other Pacific Islander', 'Two or More Races']

#Add Year values and correct wrong names
for ii in range(len(sdataframes)):
    sdataframes[ii]['Organization Code'] = sdataframes[ii]['Org. Code']
    sdataframes[ii]['Orgnazation Name'] = sdataframes[ii]['Organization Name']
    sdataframes[ii]['Year'] = str(ii+13)

#Isolate only relevant columns
scomp_cols = ['Organization Code','Orgnazation Name','American Indian or Alaskan Native
            'White','Native Hawaiian or Other Pacific Islander', 'Two or More Races',

#Combine datasets to one big DataFrame
scomp_data = scomp_17[scomp_cols].append(scomp_16[scomp_cols],ignore_index=True)
scomp_data = scomp_data.append(scomp_15[scomp_cols],ignore_index=True)
scomp_data = scomp_data.append(scomp_14[scomp_cols],ignore_index=True)
scomp_data = scomp_data.append(scomp_13[scomp_cols],ignore_index=True)
```

```
#Remove State Totals row
scomp_data = scomp_data[scomp_data['Organization Code']!='TOTALS']

#Remove null values
scomp_data = scomp_data.dropna()


#Correct wrong dataypes
scomp_data['Student Pct Minority'] = scomp_data['Percent Minority'].replace({'\%':''},r
scomp_data[eth_cols] = scomp_data[eth_cols].replace({',':''},regex=True).astype(int)
scomp_data['Organization Code'] = scomp_data['Organization Code'].astype(int)

scomp_data = scomp_data[['Organization Code','Orgnazation Name','Student Pct Minority',
```

In [14]:
```
#Review Student Ethnicity data
scomp_data.head()
```

Out[14]:

| | Organization Code | Orgnazation Name | Student Pct Minority | Year |
|---|---|---|---|---|
| 0 | 10 | MAPLETON 1 | 0.718 | 17 |
| 1 | 20 | ADAMS 12 FIVE STAR SCHOOLS | 0.512 | 17 |
| 2 | 30 | ADAMS COUNTY 14 | 0.893 | 17 |
| 3 | 40 | SCHOOL DISTRICT 27J | 0.538 | 17 |
| 4 | 50 | BENNETT 29J | 0.316 | 17 |

In [15]:
```
#AP Class data

#All relevant columns
ap_cols = ['Organization Code','Organization Name','American Indian or Alaska Native Co
           'Black or African American Count', 'Hispanic or Latino Count','White Count',
            'Native Hawaiian or Other Pacific Islander Count','Two Or More Races Count',
            'Female Count','Total Student Count','Year']

#Numeric columns
count_cols = ['American Indian or Alaska Native Count', 'Asian Count',
           'Black or African American Count', 'Hispanic or Latino Count','White Count',
            'Native Hawaiian or Other Pacific Islander Count','Two Or More Races Count',
            'Female Count','Total Student Count']

#New names for columns as there are already ones with the same names from other dataset
ap_count_cols = ['ap_American Indian or Alaska Native Count', 'ap_Asian Count',
           'ap_Black or African American Count', 'ap_Hispanic or Latino Count','ap_White
            'ap_Native Hawaiian or Other Pacific Islander Count','ap_Two Or More Races C
            'Female AP Enrollment','Total AP Enrollment']

#Only take the total AP count
#Add Year values
ap_16 = ap_16[ap_16['Course Code']=='TOTAL']
```

```
ap_16['Year'] = '16'
ap_15 = ap_15[ap_15['Course Code']=='TOTAL']
ap_15['Year'] = '15'
ap_14 = ap_14[ap_14['Course Code']=='TOTAL']
ap_14['Year'] = '14'
ap_13 = ap_13[ap_13['Course Code']=='TOTAL']
ap_13['Year'] = '13'

#Combine datasets
ap_data = ap_16[ap_cols].append(ap_15[ap_cols],ignore_index=True)
ap_data = ap_data.append(ap_14[ap_cols],ignore_index=True)
ap_data = ap_data.append(ap_14[ap_cols],ignore_index=True)

#Drop null values
ap_data = ap_data.dropna()


ap_data['Organization Code'] = ap_data['Organization Code'].astype(float)
ap_data['Orgnazation Name'] = ap_data['Organization Name']
ap_data = ap_data.drop(['Organization Name'],axis=1)

#Apply new column names
for name in range(len(count_cols)):
    ap_data[ap_count_cols[name]] = ap_data[count_cols[name]].astype(int)

#Only take the total enrollment variables along with merge columns
ap_data = ap_data[['Organization Code','Orgnazation Name','Year','Female AP Enrollment'
```

In [16]:
```
#Review AP data
ap_data.head()
```

Out[16]:

| | Organization Code | Orgnazation Name | Year | Female AP Enrollment | \ |
|---|---|---|---|---|---|
| 0 | 9999.0 | STATE TOTALS | 16 | 42129 | |
| 1 | 10.0 | MAPLETON 1 | 16 | 0 | |
| 2 | 20.0 | ADAMS 12 FIVE STAR SCHOOLS | 16 | 1962 | |
| 3 | 30.0 | ADAMS COUNTY 14 | 16 | 167 | |
| 4 | 40.0 | SCHOOL DISTRICT 27J | 16 | 428 | |

| | Male AP Enrollment | Total AP Enrollment |
|---|---|---|
| 0 | 33264 | 75393 |
| 1 | 0 | 0 |
| 2 | 1532 | 3494 |
| 3 | 94 | 261 |
| 4 | 208 | 636 |

In [17]:
```
#Combine all datasets into one big dataset

total_data = pd.merge(scomp_data,lunch_data)
```

```
total_data = pd.merge(total_data,grad_data)
total_data = pd.merge(total_data,str_data)
total_data = pd.merge(total_data,tcomp_data)
total_data = pd.merge(total_data,sal_data)

total_ap = pd.merge(total_data,ap_data)
```

In [18]:
```
#Verify complete dataset
total_ap.head()
```

Out[18]:

| | Organization Code | Orgnazation Name | Student Pct Minority | Year |
|---|---|---|---|---|
| 0 | 10 | MAPLETON 1 | 0.01 | 16 |
| 1 | 20 | ADAMS 12 FIVE STAR SCHOOLS | 0.01 | 16 |
| 2 | 30 | ADAMS COUNTY 14 | 0.01 | 16 |
| 3 | 50 | BENNETT 29J | 0.01 | 16 |
| 4 | 60 | STRASBURG 31J | 0.01 | 16 |

| | % FREE AND REDUCED | K-12 COUNT | All Students Graduation Rate |
|---|---|---|---|
| 0 | 0.586 | 8380 | 0.646 |
| 1 | 0.396 | 37688 | 0.806 |
| 2 | 0.850 | 6876 | 0.658 |
| 3 | 0.334 | 1029 | 0.771 |
| 4 | 0.269 | 937 | 0.831 |

| | Female Graduation Rate | Male Graduation Rate | Pupil/Teacher Ratio |
|---|---|---|---|
| 0 | 0.714 | 0.573 | 31.9461 |
| 1 | 0.847 | 0.769 | 21.5446 |
| 2 | 0.732 | 0.590 | 20.5723 |
| 3 | 0.806 | 0.735 | 13.4775 |
| 4 | 0.870 | 0.791 | 15.17 |

| | Teacher Pct Minority | Teacher Count | Total FTE | Average Salary |
|---|---|---|---|---|
| 0 | 0.164921 | 382 | 373.04 | 52969.0 |
| 1 | 0.106848 | 2059 | 1980.15 | 59127.0 |
| 2 | 0.228571 | 385 | 379.65 | 53825.0 |
| 3 | 0.089286 | 56 | 56.89 | 38758.0 |
| 4 | 0.016667 | 60 | 57.07 | 43005.0 |

| | Female AP Enrollment | Male AP Enrollment | Total AP Enrollment |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 1962 | 1532 | 3494 |
| 2 | 167 | 94 | 261 |
| 3 | 0 | 0 | 0 |
| 4 | 260 | 231 | 491 |

In [19]:
```
#Save files

#All years without AP data
```

```
total_data.to_csv('Total_Data1.csv',index=False)

#AP data only has 2014-2017
total_ap.to_csv('Compiled_Data.csv',index=False)
```