In [1]:
```
# -*- coding: utf-8 -*-
%matplotlib inline

import pandas as pd
import numpy as np
import math
import matplotlib as mplstyle
import matplotlib.pyplot as plt
import locale
from locale import atof
from dateutil import parser
from datetime import datetime

plt.style.use('fivethirtyeight')
```

Import data

In [2]:
```
df_vouchersBySchool = pd.read_csv('school_choice_data/imported_data/awards_by_school_16.csv')
df_specialEdEnrollment = pd.read_csv('school_choice_data/gov_data/EnrollmentEnglishLearnerSpecEdSchls_ALL.csv')
df_missingSpecialEdEnrollment = pd.read_csv('school_choice_data/imported_data/updatedEnrollment.csv')
df_vouchersByOgSchoolDistrict = pd.read_csv('school_choice_data/imported_data/voucher-students-by-original-school-district.csv')
df_publicSpecialEdEnrollment = pd.read_csv('school_choice_data/gov_data/PublicEnrollmentEnglishLearnerSpecEd.csv')
df_schoolGrades2015 = pd.read_csv('school_choice_data/imported_data/final-2016-af-school-grades.csv')
```

Functions

In [3]:
```
# replace symbols
# use 0 for no school choice students and 9 for schools with less than 10 stud
ents
def replaceSymbols(id):
    if id == '*':
        return 9
    elif id == '-':
        return 0
    else:
        return id

# normalize numbers
# if is not a number, return None. if is a number, turn it into an integer
def normalizeNums(num):
    if (math.isnan(num)):
        return None
    else:
        num = int(math.floor(num))
        return num

# multiply by 100 to convert from percent
def convertPercent(num):
    return num * 100

# normalize percent from x% to just a number
def normalizePerc(num):
    try:
        float(num.strip('%'))
    except AttributeError:
        return num
    return float(num.strip('%'))

# get average award amount
# divide award amount by award count if they are both numbers
def averageAward(awardCount, awardAmount):
    if math.isnan(awardAmount) or math.isnan(awardCount):
        return None
    return round((awardAmount/awardCount),2)

# Convert NaN to zeros
# the zeros do not always indicate that no students
# were in the program. schools with less than ten
# students do not have to report
def convertInt(num):
    if math.isnan(num):
        return 0
    else:
        return int(num)

# calculate percent
def calcPerc(num, denom):
    if num != 0:
        percent = (float(num/denom))*100
        return round(percent,2)
    else:
        return None
```

```
# calculate percent change
def percentChange(oldVal, newVal):
    val1 = float(oldVal)
    val2 = float(newVal)
    change = (val2-val1)/val1
    return float(change)

# fill in missing values
# compares two values, takes the first if is not null. if it is, takes second.
def fillInfo(choiceCell, backupCell):
    if pd.isnull(choiceCell):
        return backupCell
    elif math.isnan(choiceCell):
        return backupCell
    else:
        return choiceCell

# replace with None
def fillNone(num):
    if num == 1:
        return None
    else:
        return num
```

Voucher enrollment data

Apply function to replace symbols. Drop data from all but most recent school year. Filter out schools with no voucher enrollment.

In [4]:
```
df_vouchersBySchool2016_filtered = df_vouchersBySchool
df_vouchersBySchool2016_filtered['Voucher enrollment 2016'] = df_vouchersBySch
ool2016_filtered['2016-2017'].apply(replaceSymbols)

# Trim out data from 2011-2015.
df_vouchersBySchool2016_trimmed = df_vouchersBySchool.drop(['2011-2012','2012-
2013','2013-2014','2014-2015','2015-2016','2016-2017'], axis=1)

# Filter out schools with no voucher enrollment in 2016
# In the data, 1 == "-", and 9 == "Less than 10 voucher students"
# We want to filter out the "-" ones because they're basically zero.
df_vouchersBySchool2016_trimmed = df_vouchersBySchool2016_trimmed[df_vouchersB
ySchool2016_trimmed['Voucher enrollment 2016'] > 0]
```

Indiana school enrollment (all)

In [5]:
```
# INDIANA SCHOOL ENROLLMENT (ALL)

# Filter out the public schools.
# Public school IDs are numbers-only. Private school IDs have letters + number
s.
def filterPrivate(id):
    if pd.isnull(id):
        return True
    else:
        try:
            float(id)
        except ValueError:
            return False
        return True

df_specialEdEnrollment_filtered = df_specialEdEnrollment
df_specialEdEnrollment_filtered['filter'] = df_specialEdEnrollment_filtered['S
CHL_NAME_CODE'].apply(filterPrivate)
df_specialEdEnrollment_filtered = df_specialEdEnrollment_filtered[df_specialEd
Enrollment_filtered['filter'] == False]
df_specialEdEnrollment_filtered = df_specialEdEnrollment_filtered.reset_index
()

# Tidy and rename columns
df_specialEdEnrollment_trimmed = df_specialEdEnrollment_filtered.drop(['index'
,'CORP','CORP_NAME','ENR_ELL_N','ENR_ELL_RATIO', 'filter'], axis=1)
df_specialEdEnrollment_trimmed = df_specialEdEnrollment_trimmed.rename(columns
= {'SCHL_NAME_CODE':'School No.','SCHL_NAME':'School Name','ENR_SPEC_ED_N':'Sp
ecial ed enrollment 2016','ENR_SPEC_ED_RATIO':'Special ed ratio 2016','ENROLLM
ENT_N':'Total enrollment 2016'})
df_specialEdEnrollment_trimmed['Special ed ratio 2016'] = df_specialEdEnrollme
nt_trimmed['Special ed ratio 2016'].apply(normalizePerc)
df_specialEdEnrollment_trimmed

# Tidy and rename updated enrollment columns
df_missingSpecialEdEnrollment = df_missingSpecialEdEnrollment.rename(columns =
{'School ID':'School No.','SpecEd 2016':'Special ed enrollment 2016','SpecEd P
erc':'Special ed ratio 2016','Enrollment 2016':'Total enrollment 2016'})
df_missingSpecialEdEnrollment['Special ed ratio 2016'] = df_missingSpecialEdEn
rollment['Special ed ratio 2016'].apply(normalizePerc)
df_missingSpecialEdEnrollment

# Merge w/ other special education data
df_specialEdEnrollment_merged = df_specialEdEnrollment_trimmed.merge(df_missin
gSpecialEdEnrollment,on='School No.', how='outer')
df_specialEdEnrollment_merged['Special ed enrollment 2016'] = np.vectorize(fil
lInfo)(df_specialEdEnrollment_merged['Special ed enrollment 2016_x'],df_specia
lEdEnrollment_merged['Special ed enrollment 2016_y'])
df_specialEdEnrollment_merged['Special ed ratio 2016'] = np.vectorize(fillInfo
)(df_specialEdEnrollment_merged['Special ed ratio 2016_x'],df_specialEdEnrollm
ent_merged['Special ed ratio 2016_y'])
df_specialEdEnrollment_merged['Total enrollment 2016'] = np.vectorize(fillInfo
)(df_specialEdEnrollment_merged['Total enrollment 2016_x'],df_specialEdEnrollm
ent_merged['Total enrollment 2016_y'])
df_specialEdEnrollment_merged = df_specialEdEnrollment_merged.drop(['Special e
d enrollment 2016_x','Special ed enrollment 2016_y','Special ed ratio 2016_x',
```

```
'Special ed ratio 2016_y','Total enrollment 2016_x','Total enrollment 2016_y'
], axis=1)
df_specialEdEnrollment_merged
# df_specialEdEnrollment_merged.to_csv('school_choice_data/exported_data/df_sp
ecialEdEnrollment_merged.csv')

# Merge w/ voucher school data
df_vouchersSpecialEd_merged = df_vouchersBySchool2016_trimmed.merge(df_special
EdEnrollment_merged,on='School No.', how='outer')
df_vouchersSpecialEd_merged = df_vouchersSpecialEd_merged.drop(['School Name_
y'], axis=1)
df_vouchersSpecialEd_merged = df_vouchersSpecialEd_merged.rename(columns = {'S
chool Name_x':'School Name'})
# df_vouchersSpecialEd_merged
```

School grades

In [6]:
```
# Associate schools with their current grade by merging grades df and df_vouch
ersSpecialEd_merged

# Tidy and rename columns
df_schoolGrades2015_trimmed = df_schoolGrades2015.drop(['IDOE_CORPORATION_ID',
'CORPORATION_NAME'], axis=1)
df_schoolGrades2015_trimmed = df_schoolGrades2015_trimmed.rename(columns = {'I
DOE_SCHOOL_ID':'School No.','SCHOOL_NAME':'School Name','2015 Grade':'2015_gra
de'})
df_schoolGrades2015_trimmed

# Merge w/ voucher school data
df_vouchersSpecialEdGrades_merged = df_vouchersSpecialEd_merged.merge(df_schoo
lGrades2015_trimmed,on='School No.', how='outer')
df_vouchersSpecialEdGrades_merged = df_vouchersSpecialEdGrades_merged.drop(['S
chool Name_y'], axis=1)
df_vouchersSpecialEdGrades_merged = df_vouchersSpecialEdGrades_merged.rename(c
olumns = {'School Name_x':'School Name'})
df_vouchersSpecialEdGrades_merged

# Filter out all schools without vouchers in 2016
df_vouchersSpecialEdGrades_merged = df_vouchersSpecialEdGrades_merged[df_vouch
ersSpecialEdGrades_merged['Voucher enrollment 2016'] > 0]
# df_vouchersSpecialEdGrades_merged
```

School districts

In [7]:
```
# Assign schools to parent school districts

# Tidy and rename
df_vouchersByOgSchoolDistrict_filtered = df_vouchersByOgSchoolDistrict.rename(
columns = {'Corp. No.':'District No.','Corporation Name':'District Name'})
df_vouchersByOgSchoolDistrict_filtered = df_vouchersByOgSchoolDistrict_filtere
d.drop(['Voucher students from this district (2016-2017)'], axis=1)
# df_vouchersByOgSchoolDistrict_filtered

# Merge w/ school ID
df_vouchersSpecialEdGradesDistricts_merged = df_vouchersSpecialEdGrades_merged
.merge(df_vouchersByOgSchoolDistrict_filtered,on='School No.', how='outer')
df_vouchersSpecialEdGradesDistricts_merged = df_vouchersSpecialEdGradesDistric
ts_merged.drop(['School Name_y'], axis=1)
df_vouchersSpecialEdGradesDistricts_merged = df_vouchersSpecialEdGradesDistric
ts_merged.rename(columns = {'School Name_x':'School Name'})
df_vouchersSpecialEdGradesDistricts_merged

df_vouchersSpecialEdGradesDistricts_merged.to_csv('school_choice_data/exported
_data/df_vouchersSpecialEdGradesDistricts_merged.csv')
```

Aggregate by parent school district

In [8]:
```
# Drop columns, group by district and reset index.
df_districtVoucherSpecialEdAggregate = df_vouchersSpecialEdGradesDistricts_mer
ged.drop(['Special ed ratio 2016'], axis=1)
df_districtVoucherSpecialEdAggregate = df_districtVoucherSpecialEdAggregate.gr
oupby('District No.').sum()
df_districtVoucherSpecialEdAggregate = df_districtVoucherSpecialEdAggregate.re
set_index()
df_districtVoucherSpecialEdAggregate

# Recalculate special ed ratio
df_districtVoucherSpecialEdAggregate['Special ed ratio 2016'] = np.vectorize(c
alcPerc)(df_districtVoucherSpecialEdAggregate['Special ed enrollment 2016'],df
_districtVoucherSpecialEdAggregate['Total enrollment 2016'])
# df_districtVoucherSpecialEdAggregate
```

Public school special ed enrollment

In [9]:
```
# Merge in public school special ed enrollment data

# Tidy and clean columns
df_publicSpecialEdEnrollment_trimmed = df_publicSpecialEdEnrollment
df_publicSpecialEdEnrollment_trimmed = df_publicSpecialEdEnrollment_trimmed.dr
op(['ENR_ELL_N','ENR_ELL_RATIO'], axis=1)
df_publicSpecialEdEnrollment_trimmed = df_publicSpecialEdEnrollment_trimmed.re
name(columns = {'CORP':'District No.','CORP_NAME':'District Name','ENR_SPEC_ED
_N':'Public special ed enrollment 2016','ENR_SPEC_ED_RATIO':'Public special ed
ratio 2016','ENROLLMENT_N':'Public total enrollment 2016'})
df_publicSpecialEdEnrollment_trimmed['Public special ed ratio 2016'] = df_publ
icSpecialEdEnrollment_trimmed['Public special ed ratio 2016'].apply(normalizeP
erc)
df_publicSpecialEdEnrollment_trimmed

# Merge
df_districtVoucherSpecialEdAggregate_merged = df_publicSpecialEdEnrollment_tri
mmed.merge(df_districtVoucherSpecialEdAggregate,on='District No.', how='outer'
)
df_districtVoucherSpecialEdAggregate_merged

# df_districtVoucherSpecialEdAggregate_merged.to_csv('school_choice_data/expor
ted_data/df_districtVoucherSpecialEdAggregate_merged.csv')
```

Out[9]:

	District No.	District Name	Public special ed enrollment 2016	Public special ed ratio 2016	Public total enrollment 2016	Special ed enrollment 2016	Total enrollment 2016	Special ed ratio 2016
0	15	Adams Central Community Schools	132	10.48	1259	25.0	297.0	8.42
1	25	North Adams Community Schools	260	14.37	1809	75.0	1069.0	7.02
2	35	South Adams Schools	195	15.02	1298	114.0	1012.0	11.26
3	125	M S D Southwest Allen County Schls	565	7.86	7190	629.0	9483.0	6.63
4	225	Northwest Allen County Schools	887	12.05	7362	677.0	10389.0	6.52
5	235	Fort Wayne Community Schools	4514	15.37	29377	849.0	13046.0	6.51
6	255	East Allen County Schools	1113	11.63	9569	824.0	12617.0	6.53
7	365	Bartholomew Con School Corp	1431	12.38	11562	224.0	3058.0	7.33
8	370	Flat Rock-Hawcreek School Corp	124	14.37	863	60.0	1025.0	5.85
9	395	Benton Community School Corp	438	23.41	1871	243.0	2362.0	10.29
10	515	Blackford County Schools	285	16.69	1708	NaN	NaN	NaN
11	615	Western Boone Co Com Sch Dist	279	16.33	1709	51.0	616.0	8.28
12	630	Zionsville Community Schools	813	11.73	6931	301.0	6048.0	4.98
13	665	Lebanon Community School Corp	639	18.61	3434	265.0	3406.0	7.78
14	670	Brown County School Corporation	493	24.64	2001	254.0	3214.0	7.90
15	750	Carroll Consolidated School Corp	108	9.73	1110	195.0	1877.0	10.39

	District No.	District Name	Public special ed enrollment 2016	Public special ed ratio 2016	Public total enrollment 2016	Special ed enrollment 2016	Total enrollment 2016	Special ed ratio 2016
16	755	Delphi Community School Corp	154	10.91	1411	283.0	3035.0	9.32
17	775	Pioneer Regional School Corp	107	11.75	911	NaN	NaN	NaN
18	815	Southeastern School Corp	143	10.68	1339	38.0	267.0	14.23
19	875	Logansport Community Sch Corp	499	11.73	4254	1.0	127.0	0.79
20	940	West Clark Community Schools	670	14.07	4761	208.0	3076.0	6.76
21	1000	Clarksville Community School Corp	236	16.83	1402	187.0	2733.0	6.84
22	1010	Greater Clark County Schools	1718	16.29	10544	208.0	2983.0	6.97
23	1125	Clay Community Schools	865	20.46	4228	4.0	462.0	0.87
24	1150	Clinton Central School Corporation	108	11.66	926	63.0	745.0	8.46
25	1160	Clinton Prairie School Corporation	161	13.39	1202	212.0	2333.0	9.09
26	1170	Community Schools of Frankfort	408	12.78	3193	164.0	2531.0	6.48
27	1180	Rossville Con School District	108	10.88	993	179.0	2043.0	8.76
28	1300	Crawford County Community Sch Corp	317	20.39	1555	19.0	743.0	2.56
29	1315	Barr-Reeve Community Schools Inc	87	10.61	820	16.0	172.0	9.30
...	...	...	...	...	...	...	...	...
361	9840	Excel Center - University Heights	44	11.28	390	NaN	NaN	NaN
362	9845	Xavier School of Excellence	38	17.76	214	NaN	NaN	NaN

	District No.	District Name	Public special ed enrollment 2016	Public special ed ratio 2016	Public total enrollment 2016	Special ed enrollment 2016	Total enrollment 2016	Special ed ratio 2016
363	9855	Excel Center - Noblesville	26	13.13	198	NaN	NaN	NaN
364	9865	Hoosier Acad Virtual Charter	435	13.02	3342	NaN	NaN	NaN
365	9870	Discovery Charter School	67	12.93	518	NaN	NaN	NaN
366	9875	Rock Creek Community Academy	141	27.81	507	NaN	NaN	NaN
367	9880	Career Academy High School	68	25.56	266	NaN	NaN	NaN
368	9885	Gary Middle College	20	8.44	237	NaN	NaN	NaN
369	9890	Indiana Virtual School	222	7.53	2947	NaN	NaN	NaN
370	9895	IN Math & Science Academy - North	84	15.14	555	NaN	NaN	NaN
371	9900	Excel Center - South Bend	10	2.78	360	NaN	NaN	NaN
372	9905	Indiana Connections Academy	631	15.65	4032	NaN	NaN	NaN
373	9910	Excel Center for Adult Learners	103	9.94	1036	NaN	NaN	NaN
374	9915	Marion Academy	47	27.65	170	NaN	NaN	NaN
375	9920	Damar Charter Academy	160	97.56	164	NaN	NaN	NaN
376	9925	Phalen Leadership Academy - IN Inc	23	7.42	310	NaN	NaN	NaN
377	9930	Nexus Academy of Indianapolis	22	16.92	130	NaN	NaN	NaN
378	9935	Vision Academy	74	16.86	439	NaN	NaN	NaN
379	9940	Tindley Collegiate Academy	34	12.64	269	NaN	NaN	NaN

	District No.	District Name	Public special ed enrollment 2016	Public special ed ratio 2016	Public total enrollment 2016	Special ed enrollment 2016	Total enrollment 2016	Special ed ratio 2016
380	9945	Tindley Renaissance Academy	52	9.56	544	NaN	NaN	NaN
381	9950	Dugger Union Community School Corp	42	13.73	306	NaN	NaN	NaN
382	9955	Mays Community Academy	39	21.67	180	NaN	NaN	NaN
383	9960	Success Academy Primary School	83	17.47	475	NaN	NaN	NaN
384	9965	Career Academy Middle School	83	20.80	399	NaN	NaN	NaN
385	9970	ACE Preparatory Academy	2	5.56	36	NaN	NaN	NaN
386	9975	Global Preparatory Academy	22	8.18	269	NaN	NaN	NaN
387	9980	Steel City Academy	23	15.13	152	NaN	NaN	NaN
388	9985	Seven Oaks Classical School	20	12.42	161	NaN	NaN	NaN
389	9990	Heritage Institute of Arts & Tech	18	11.76	153	NaN	NaN	NaN
390	9995	Excel Center - Shelbyville	3	3.57	84	NaN	NaN	NaN

391 rows × 8 columns

In [ ]: