Unfortunately, due to the number of schools, it's hard to interpret the map we made on the last screen. It looks like uptown Manhattan and parts of Queens have a higher ell\_percent, but we can't be sure. One way to make very granular statistics easier to read is to aggregate them. In this case, we can aggregate by district, which will enable us to plot ell\_percent district-by-district instead of school-by-school.

In the last mission, we used the [pandas.DataFrame.groupby()](http://pandas.pydata.org/pandas-docs/stable/groupby.html" \t "_blank) followed by the [agg()](http://pandas.pydata.org/pandas-docs/stable/groupby.html" \l "aggregation" \t "_blank) method on the resulting object to find the mean class size for each unique DBN. The principle is exactly the same, except that here we'd find the mean of each column for each unique value in school\_dist.

Instructions

* Find the average values for each column for each school\_dist in combined.
  + Use the [pandas.DataFrame.groupby()](http://pandas.pydata.org/pandas-docs/stable/groupby.html" \t "_blank) method to group combined by school\_dist.
  + Use the [agg()](http://pandas.pydata.org/pandas-docs/stable/groupby.html" \l "aggregation" \t "_blank) method, along with the numpy.mean function as an argument, to calculate the average of each group.
  + Assign the result to the variable districts.
* Reset the index of districts, making school\_dist a column again.
  + Use the [pandas.DataFrame.reset\_index()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.reset_index.html" \t "_blank) method with the keyword argument inplace=True.
* Display the first few rows of districts to verify that everything went okay.

Answer

# import numpy to calculate the mean

import numpy

# group dataframe compined by "school\_dist" column and calculate the mean of the groups

districts = combined.groupby("school\_dist").agg(numpy.mean)

# reset the dataframe so that the DBN column to remain a column and not the index

districts.reset\_index(inplace = True)

# print 5 lines

print(districts.head())