**Other transformations with .apply**

The .apply() method when used on a groupby object performs an arbitrary function on each of the groups. These functions can be aggregations, transformations or more complex workflows. The .apply() method will then combine the results in an intelligent way.

In this exercise, you're going to analyze economic disparity within regions of the world using the Gapminder data set for 2010. To do this you'll define a function to compute the aggregate spread of per capita GDP in each region and the individual country's z-score of the regional per capita GDP. You'll then select three countries - United States, Great Britain and China - to see a summary of the regional GDP and that country's z-score against the regional mean.

The 2010 Gapminder DataFrame is provided for you as gapminder\_2010. Pandas has been imported as pd.

The following function has been defined for your use:

def disparity(gr):

# Compute the spread of gr['gdp']: s

s = gr['gdp'].max() - gr['gdp'].min()

# Compute the z-score of gr['gdp'] as (gr['gdp']-gr['gdp'].mean())/gr['gdp'].std(): z

z = (gr['gdp'] - gr['gdp'].mean())/gr['gdp'].std()

# Return a DataFrame with the inputs {'z(gdp)':z, 'regional spread(gdp)':s}

return pd.DataFrame({'z(gdp)':z , 'regional spread(gdp)':s})

**INSTRUCTIONS**

* Group gapminder\_2010 by 'region'. Save the result as regional.
* Apply the provided disparity function on regional, and save the result as reg\_disp.
* Use .loc[] to select ['United States','United Kingdom','China'] from reg\_disp and print the results.

# Group gapminder\_2010 by 'region': regional

print(gapminder\_2010.tail(20))

regional = gapminder\_2010.groupby('region')

# Apply the disparity function on regional: reg\_disp

reg\_disp = regional.apply(disparity)

# Print the disparity of 'United States', 'United Kingdom', and 'China'

print(reg\_disp.loc[['United States', 'United Kingdom', 'China']])