Pandas data analysis functions

You now know how to load CSV data into Python as pandas dataframes and you also know how to manipulate a dataframe. Let's now see what data analysis methods we can apply to the pandas dataframes.

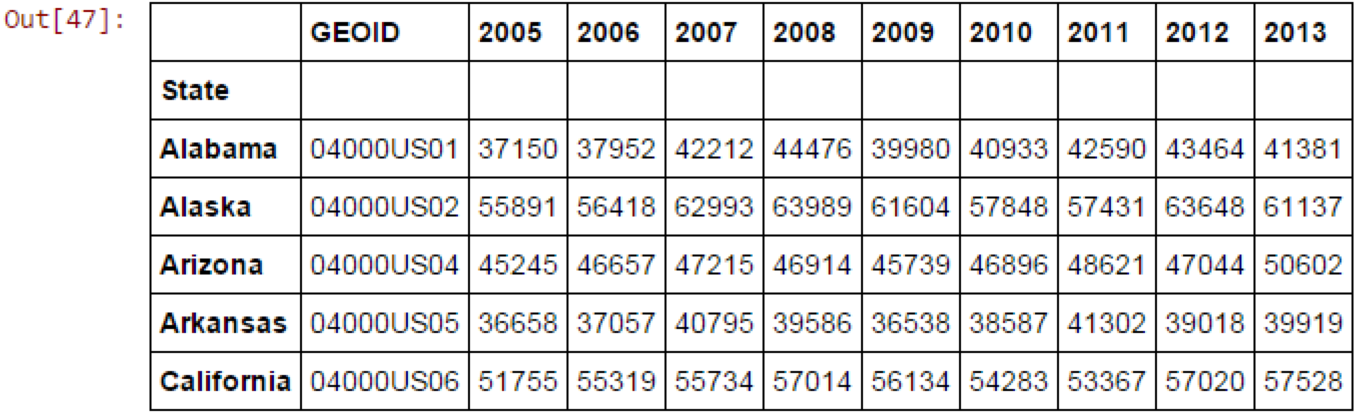
You know that the dataframe is the main pandas object. So, if you have some data loaded in dataframe *df*, you could apply methods to analyze those data. For instance, here is how you apply the mean method to the dataframe we have been working on:

```
df.mean()
```

And you would get:

```
2005 45339.8
2006 46680.6
2007 49789.8
2008 50395.8
2009 47999.0
2010 47709.4
2011 48662.2
2012 50038.8
2013 50113.4
dtype: float64
```

So, these are the mean values for each of the dataframe columns. Just to remind you, we generated the dataframe in the previous lessons of this tutorial. The dataframe looks like this:



Out[47]:

	GEOID	2005	2006	2007	2008	2009	2010	2011	2012	2013
State										
Alabama	04000US01	37150	37952	42212	44476	39980	40933	42590	43464	41381
Alaska	04000US02	55891	56418	62993	63989	61604	57848	57431	63648	61137
Arizona	04000US04	45245	46657	47215	46914	45739	46896	48621	47044	50602
Arkansas	04000US05	36658	37057	40795	39586	36538	38587	41302	39018	39919
California	04000US06	51755	55319	55734	57014	56134	54283	53367	57020	57528

You can get a list of available DataFrame methods using the Python *dir* function:

```
dir(pd.DataFrame)
```

And you can get the description of each method using help:

```
help(pd.DataFrame.mean)
```

You can also apply methods to columns of the dataframe:

```
df2.loc[:,"2005"].mean()
```

Note though that in this case you are not applying the mean method to a pandas dataframe, but to a pandas series object:

```
type(d2.loc[:,"2005"])
```

So, checking the type of the object would give the type of the object:

```
pandas.core.series.Series
```

And again you can pass the Series object to the dir method to get a list of available methods.

Adding columns to a DataFrame is quite straightforward:

```
df2["2014"]=[4000,6000,4000,4000,6000]
```

That would add a new column with label "2014" and the values of the Python list. You can also add a column containing the average income for each state:

```
df2["Mean"]=df2.mean(axis=1)
```

And you would get this:

The axis parameter tells Python to compute the mean along axis 1 which means along the columns. Axis set to 0 would go along the rows. Let's see how to calculate the mean for each year and add them as a new row:

```
df2.loc["MEAN"]=df2.mean(axis=0)
```

This would add a new row with index "MEAN":

Instead of throwing an error, pandas generated a NaN datatype for the GEOID column which is a good thing because operations won't break when the dataframe has non-numeric values.