Intro-Pandas-PBCII

February 14, 2017

1 Introduction to data manipulation and analysis with *Pandas*

The objective of this notebook is to make a brief overview of the pandas module for data analysis and manipulation. To present the different pandas tools,w e use as an example the hydrometeorological data from the famous sistema Cantareira reservoir (São Paulo/Brazil). The data files are located in the data directory of this repository.

Have fun :)

1.1 What is Pandas and why use it?

Pandas provides easy to use and intuitive data structures and high performance tools for data analysis. It has been designed for practical, real-world analysis.

The community have the objective to build the most powerful and flexible open source data analysis / manipulation tool available in any language.

You will find everything you need to know about Pandas at: http://pandas.pydata.org/

1.2 Topics covered

- Object Creation: Series and Dataframe
- Data Visualisation
- Data Selection
- Read data from file
- Dealing with missing data
- Setting in data structure
- Basic operation and apply function
- Merge/concatenate/join/append data structures
- Resample
- Grouping : split-apply-combine
- Pivot Table
- Writing to file
- Basic plots

This presentation is inspired from the amazing "10 minutes to Pandas" tutorial (http://pandas.pydata.org/pandas-docs/stable/10min.html).

For further details look at the Pandas documentation: http://pandas.pydata.org/pandas-docs/stable/index.html

1.3 Let's fire up Pandas !

We will need numpy and matplotlib.pyplot as well.

```
In [2]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

1.4 Object Creation

Pandas supports three data structures: * Series (1 axis). * Dataframe (2 axis) * Panel (3 - N axis)

They are labeled arrays which can contain any kind of data types (int,float, string, objects ...). The labels are called index in pandas.

The main difference between numpy and pandas reside

For more information about data structure, please have a look at the documentation: http://pandas.pydata.org/pandas-docs/stable/dsintro.html#dsintro

1.4.1 Create a serie

We need: * an index, e.g. a list of string * an array, e.g. a numpy.ndarray

if no index is specified, pandas will by default use a list of interger.

```
In [3]: labels = ['a','b','c','d']
        values = np.arange(4)
```

```
In [4]: pd.Series(values, index=labels)
```

```
Out[4]: a    0
        b    1
        c    2
        d    3
        dtype: int64
```

1.4.2 Create a dataframe

A dataframe is a two-dimensional labeled array. There is various way to create a dataframe. for more information, you can have a look at: http://pandas.pydata.org/pandas-docs/stable/dsintro.html#dataframe

from a dictionnary of list

```
In [5]: dic = {'a':[1,2,3,4,5],
               'b':[10,20,30,40,50]
               }
        print dic
```

```
{'a': [1, 2, 3, 4, 5], 'b': [10, 20, 30, 40, 50]}
```

```
In [6]: df = pd.DataFrame(dic)
        print df
```

```
   a   b
0  1  10
1  2  20
2  3  30
3  4  40
4  5  50
```

1.5 Time Series / Date functionality

Pandas provide powerful tools to work with time series based upon numpy datetime64 and timedelta64 dtypes.

```
In [10]: rng = pd.date_range('1/1/2016', periods=100, freq='S')
```

```
In [11]: pd.period_range('1/1/2016', '1/1/2017', freq='D')
```

```
Out[11]: PeriodIndex(['2016-01-01', '2016-01-02', '2016-01-03', '2016-01-04',
                      '2016-01-05', '2016-01-06', '2016-01-07', '2016-01-08',
                      '2016-01-09', '2016-01-10',
                      ...
                      '2016-12-23', '2016-12-24', '2016-12-25', '2016-12-26',
                      '2016-12-27', '2016-12-28', '2016-12-29', '2016-12-30',
                      '2016-12-31', '2017-01-01'],
                     dtype='int64', length=367, freq='D')
```

To create a temporal serie

```
In [12]: pd.Series(range(100), index = rng)
```

```
Out[12]: 2016-01-01 00:00:00     0
         2016-01-01 00:00:01     1
         2016-01-01 00:00:02     2
         2016-01-01 00:00:03     3
         2016-01-01 00:00:04     4
         2016-01-01 00:00:05     5
         2016-01-01 00:00:06     6
         2016-01-01 00:00:07     7
         2016-01-01 00:00:08     8
         2016-01-01 00:00:09     9
         2016-01-01 00:00:10    10
         2016-01-01 00:00:11    11
         2016-01-01 00:00:12    12
         2016-01-01 00:00:13    13
         2016-01-01 00:00:14    14
         2016-01-01 00:00:15    15
         2016-01-01 00:00:16    16
         2016-01-01 00:00:17    17
         2016-01-01 00:00:18    18
```

```
2016-01-01 00:00:19    19
2016-01-01 00:00:20    20
2016-01-01 00:00:21    21
2016-01-01 00:00:22    22
2016-01-01 00:00:23    23
2016-01-01 00:00:24    24
2016-01-01 00:00:25    25
2016-01-01 00:00:26    26
2016-01-01 00:00:27    27
2016-01-01 00:00:28    28
2016-01-01 00:00:29    29
                       ..
2016-01-01 00:01:10    70
2016-01-01 00:01:11    71
2016-01-01 00:01:12    72
2016-01-01 00:01:13    73
2016-01-01 00:01:14    74
2016-01-01 00:01:15    75
2016-01-01 00:01:16    76
2016-01-01 00:01:17    77
2016-01-01 00:01:18    78
2016-01-01 00:01:19    79
2016-01-01 00:01:20    80
2016-01-01 00:01:21    81
2016-01-01 00:01:22    82
2016-01-01 00:01:23    83
2016-01-01 00:01:24    84
2016-01-01 00:01:25    85
2016-01-01 00:01:26    86
2016-01-01 00:01:27    87
2016-01-01 00:01:28    88
2016-01-01 00:01:29    89
2016-01-01 00:01:30    90
2016-01-01 00:01:31    91
2016-01-01 00:01:32    92
2016-01-01 00:01:33    93
2016-01-01 00:01:34    94
2016-01-01 00:01:35    95
2016-01-01 00:01:36    96
2016-01-01 00:01:37    97
2016-01-01 00:01:38    98
2016-01-01 00:01:39    99
Freq: S, dtype: int64
```

1.6 Read (real) data from file.

Let's explore hydrometeorological data observed in the region of the Cantareira sistema (The main reservoir of the São Paulo megacity), more precisely the accumulated rainfall (mm/day) and the volume of water in (%) (100% correspond to the maximum capacity).

Note: In this example, the index of our dataframe is going to be a datetime object, as time series are recurrently used in atmospheric sciences. However, the index could be whatever labeled object, for example the ID of the stars for applications in astronomy.

```
In [13]: Path = './data/'
         filename = "DataCantareira.csv"
```

Lets have a look at the amazing capabilities of the "read_csv" method! http://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html

This method provide so many options that you will probably never need to perform extra programing in order to read your data! (This is really timesaving for data analysis tasks)

```
In [14]: df = pd.read_csv(Path+filename)
         print df
```

```
          dates   reservoir  volume (%)  rain (mm/dia)
0      1/1/2003  Cantareira        42.5            0.0
1      1/2/2003  Cantareira         NaN           11.1
2      1/3/2003  Cantareira        42.3            8.6
3      1/4/2003  Cantareira         NaN           14.3
4      1/5/2003  Cantareira        42.4           12.7
5      1/6/2003  Cantareira        42.4           23.8
6      1/7/2003  Cantareira        42.5            0.0
7      1/8/2003  Cantareira        42.5            0.0
8      1/9/2003  Cantareira        42.6            0.0
9     1/10/2003  Cantareira        42.3            0.0
10    1/11/2003  Cantareira        42.4           25.0
11    1/12/2003  Cantareira        42.6            8.3
12    1/13/2003  Cantareira        43.0           23.3
13    1/14/2003  Cantareira        42.9            2.8
14    1/15/2003  Cantareira        42.8            0.0
15    1/16/2003  Cantareira        42.8            0.0
16    1/17/2003  Cantareira        43.3           43.4
17    1/18/2003  Cantareira        44.6           42.6
18    1/19/2003  Cantareira        45.3            6.4
19    1/20/2003  Cantareira        46.1           30.0
20    1/21/2003  Cantareira        46.7            0.0
21    1/22/2003  Cantareira        48.1           49.6
22    1/23/2003  Cantareira        49.2           18.2
23    1/24/2003  Cantareira        50.1           22.2
24    1/25/2003  Cantareira        51.1            1.1
25    1/26/2003  Cantareira        52.1            4.1
26    1/27/2003  Cantareira        53.3           18.5
27    1/28/2003  Cantareira        54.4           16.9
28    1/29/2003  Cantareira        56.0           18.2
29    1/30/2003  Cantareira        57.5            4.3
...         ...         ...         ...            ...
4631   9/6/2015  Cantareira        15.0            0.0
```

```
4632   9/7/2015  Cantareira        15.0            0.1
4633   9/8/2015  Cantareira        15.0           17.5
4634   9/9/2015  Cantareira        15.4           45.8
4635  9/10/2015  Cantareira        15.6            0.2
4636  9/11/2015  Cantareira        15.7           23.8
4637  9/12/2015  Cantareira        16.0           18.3
4638  9/13/2015  Cantareira        16.3            1.1
4639  9/14/2015  Cantareira        16.4            0.0
4640  9/15/2015  Cantareira        16.5            0.0
4641  9/16/2015  Cantareira        16.5            0.0
4642  9/17/2015  Cantareira        16.5            0.0
4643  9/18/2015  Cantareira        16.5            0.0
4644  9/19/2015  Cantareira        16.4            0.0
4645  9/20/2015  Cantareira        16.4            0.0
4646  9/21/2015  Cantareira        16.3            0.0
4647  9/22/2015  Cantareira        16.3            0.0
4648  9/23/2015  Cantareira        16.2            0.0
4649  9/24/2015  Cantareira        16.1            0.0
4650  9/25/2015  Cantareira        16.0            0.0
4651  9/26/2015  Cantareira        16.1           29.6
4652  9/27/2015  Cantareira        16.2            2.0
4653  9/28/2015  Cantareira        16.2            9.6
4654  9/29/2015  Cantareira        16.2            0.2
4655  9/30/2015  Cantareira        16.2            4.2
4656  10/1/2015  Cantareira        16.4           28.7
4657  10/2/2015  Cantareira        16.5            0.0
4658  10/3/2015  Cantareira        16.6           11.8
4659  10/4/2015  Cantareira        16.7            0.2
4660  10/5/2015  Cantareira        16.7            0.9

[4661 rows x 4 columns]
```

In this case the read csv methods return directly a dataframe. As you can see the first column hold the date of each event. We could have directly defined the index to be a datetime object by passing the following argument to read csv.

```
In [15]: df = pd.read_csv( Path+filename, index_col =0, parse_dates=True)
         print df

             reservoir  volume (%)  rain (mm/dia)
dates
2003-01-01  Cantareira        42.5            0.0
2003-01-02  Cantareira         NaN           11.1
2003-01-03  Cantareira        42.3            8.6
2003-01-04  Cantareira         NaN           14.3
2003-01-05  Cantareira        42.4           12.7
2003-01-06  Cantareira        42.4           23.8
```

```
2003-01-07  Cantareira        42.5            0.0
2003-01-08  Cantareira        42.5            0.0
2003-01-09  Cantareira        42.6            0.0
2003-01-10  Cantareira        42.3            0.0
2003-01-11  Cantareira        42.4           25.0
2003-01-12  Cantareira        42.6            8.3
2003-01-13  Cantareira        43.0           23.3
2003-01-14  Cantareira        42.9            2.8
2003-01-15  Cantareira        42.8            0.0
2003-01-16  Cantareira        42.8            0.0
2003-01-17  Cantareira        43.3           43.4
2003-01-18  Cantareira        44.6           42.6
2003-01-19  Cantareira        45.3            6.4
2003-01-20  Cantareira        46.1           30.0
2003-01-21  Cantareira        46.7            0.0
2003-01-22  Cantareira        48.1           49.6
2003-01-23  Cantareira        49.2           18.2
2003-01-24  Cantareira        50.1           22.2
2003-01-25  Cantareira        51.1            1.1
2003-01-26  Cantareira        52.1            4.1
2003-01-27  Cantareira        53.3           18.5
2003-01-28  Cantareira        54.4           16.9
2003-01-29  Cantareira        56.0           18.2
2003-01-30  Cantareira        57.5            4.3
...                ...         ...            ...
2015-09-06  Cantareira        15.0            0.0
2015-09-07  Cantareira        15.0            0.1
2015-09-08  Cantareira        15.0           17.5
2015-09-09  Cantareira        15.4           45.8
2015-09-10  Cantareira        15.6            0.2
2015-09-11  Cantareira        15.7           23.8
2015-09-12  Cantareira        16.0           18.3
2015-09-13  Cantareira        16.3            1.1
2015-09-14  Cantareira        16.4            0.0
2015-09-15  Cantareira        16.5            0.0
2015-09-16  Cantareira        16.5            0.0
2015-09-17  Cantareira        16.5            0.0
2015-09-18  Cantareira        16.5            0.0
2015-09-19  Cantareira        16.4            0.0
2015-09-20  Cantareira        16.4            0.0
2015-09-21  Cantareira        16.3            0.0
2015-09-22  Cantareira        16.3            0.0
2015-09-23  Cantareira        16.2            0.0
2015-09-24  Cantareira        16.1            0.0
2015-09-25  Cantareira        16.0            0.0
2015-09-26  Cantareira        16.1           29.6
2015-09-27  Cantareira        16.2            2.0
2015-09-28  Cantareira        16.2            9.6
```

```
2015-09-29  Cantareira       16.2            0.2
2015-09-30  Cantareira       16.2            4.2
2015-10-01  Cantareira       16.4           28.7
2015-10-02  Cantareira       16.5            0.0
2015-10-03  Cantareira       16.6           11.8
2015-10-04  Cantareira       16.7            0.2
2015-10-05  Cantareira       16.7            0.9

[4661 rows x 3 columns]
```

We can verify that the index is a datetime index with the folowing code

```
In [16]: type(df.index)
```

```
Out[16]: pandas.tseries.index.DatetimeIndex
```

1.7 Visualising the data structure

Methods to quickly give a look at your data:

- head
- tail
- columns
- index
- values
- describe

```
In [17]: df.describe()
```

```
Out[17]:        volume (%)  rain (mm/dia)
         count  4659.000000    4661.000000
         mean     52.512578       3.787428
         std      24.497958      17.028824
         min       3.000000    -999.000000
         25%      34.600000       0.000000
         50%      51.300000       0.100000
         75%      71.700000       3.500000
         max     100.500000     102.000000
```

```
In [18]: df.columns
```

```
Out[18]: Index([u'reservoir', u'volume (%)', u'rain (mm/dia)'], dtype='object')
```

```
In [19]: df.index
```

```
Out[19]: DatetimeIndex(['2003-01-01', '2003-01-02', '2003-01-03', '2003-01-04',
                        '2003-01-05', '2003-01-06', '2003-01-07', '2003-01-08',
                        '2003-01-09', '2003-01-10',
               ...
               '2015-09-26', '2015-09-27', '2015-09-28', '2015-09-29',
               '2015-09-30', '2015-10-01', '2015-10-02', '2015-10-03',
               '2015-10-04', '2015-10-05'],
              dtype='datetime64[ns]', name=u'dates', length=4661, freq=None)
```

1.8 Selection

You can select items in your Pandas data structure in the same style than numpy array.

However, Pandas has his own optimized methods for data access:

- .at
- .iat
- .loc
- .iloc
- .ix

Opa Numpy style Get a column from the dataframe

```
In [20]: df['volume (%)']
```

```
Out[20]: dates
         2003-01-01    42.5
         2003-01-02     NaN
         2003-01-03    42.3
         2003-01-04     NaN
         2003-01-05    42.4
         2003-01-06    42.4
         2003-01-07    42.5
         2003-01-08    42.5
         2003-01-09    42.6
         2003-01-10    42.3
         2003-01-11    42.4
         2003-01-12    42.6
         2003-01-13    43.0
         2003-01-14    42.9
         2003-01-15    42.8
         2003-01-16    42.8
         2003-01-17    43.3
         2003-01-18    44.6
         2003-01-19    45.3
         2003-01-20    46.1
         2003-01-21    46.7
         2003-01-22    48.1
         2003-01-23    49.2
         2003-01-24    50.1
         2003-01-25    51.1
         2003-01-26    52.1
```

```
2003-01-27    53.3
2003-01-28    54.4
2003-01-29    56.0
2003-01-30    57.5
              ...
2015-09-06    15.0
2015-09-07    15.0
2015-09-08    15.0
2015-09-09    15.4
2015-09-10    15.6
2015-09-11    15.7
2015-09-12    16.0
2015-09-13    16.3
2015-09-14    16.4
2015-09-15    16.5
2015-09-16    16.5
2015-09-17    16.5
2015-09-18    16.5
2015-09-19    16.4
2015-09-20    16.4
2015-09-21    16.3
2015-09-22    16.3
2015-09-23    16.2
2015-09-24    16.1
2015-09-25    16.0
2015-09-26    16.1
2015-09-27    16.2
2015-09-28    16.2
2015-09-29    16.2
2015-09-30    16.2
2015-10-01    16.4
2015-10-02    16.5
2015-10-03    16.6
2015-10-04    16.7
2015-10-05    16.7
Name: volume (%), dtype: float64
```

Get some rows

```
In [ ]: df[0:3]
```

```
In [ ]: df['2003-01-01':'2004-01-01']
```

Selection by label For production code, it is recommended that you take advantage of the optimized pandas data access methods.

```
In [38]: df.loc['2003-01-01':'2004-01-01','vol (%)']
```

```
Out[38]: date
         2003-01-01    42.5
         2003-01-02     NaN
         2003-01-03    42.3
         2003-01-04     NaN
         2003-01-05    42.4
         2003-01-06    42.4
         2003-01-07    42.5
         2003-01-08    42.5
         2003-01-09    42.6
         2003-01-10    42.3
         2003-01-11    42.4
         2003-01-12    42.6
         2003-01-13    43.0
         2003-01-14    42.9
         2003-01-15    42.8
         2003-01-16    42.8
         2003-01-17    43.3
         2003-01-18    44.6
         2003-01-19    45.3
         2003-01-20    46.1
         2003-01-21    46.7
         2003-01-22    48.1
         2003-01-23    49.2
         2003-01-24    50.1
         2003-01-25    51.1
         2003-01-26    52.1
         2003-01-27    53.3
         2003-01-28    54.4
         2003-01-29    56.0
         2003-01-30    57.5
                       ...
         2003-12-03    18.7
         2003-12-04    19.0
         2003-12-05    19.2
         2003-12-06    19.4
         2003-12-07    19.5
         2003-12-08    19.6
         2003-12-09    19.4
         2003-12-10    19.6
         2003-12-11    19.5
         2003-12-12    19.4
         2003-12-13    19.3
         2003-12-14    19.2
         2003-12-15    19.0
         2003-12-16    18.8
         2003-12-17    19.0
         2003-12-18    19.1
```

```
2003-12-19    19.1
2003-12-20    19.0
2003-12-21    19.3
2003-12-22    19.6
2003-12-23    20.0
2003-12-24    20.4
2003-12-25    20.5
2003-12-26    20.6
2003-12-27    20.6
2003-12-28    20.5
2003-12-29    20.7
2003-12-30    20.2
2003-12-31    21.7
2004-01-01    20.0
Name: vol (%), dtype: float64
```

Selection by Position Select via the position of the passed integers

```
In [ ]: df.iloc[3:5,0:2]
```

Boolean Indexing Let's select the events when the Cantareira reservatory was under 5% of its volume

```
In [20]: df[df['volume (%)'] < 5]
```

```
Out[20]:            reservatorio  vol util (%)  prec (mm/dia)
         data
         2014-10-12   Cantareira           4.8            0.0
         2014-10-13   Cantareira           4.7            0.0
         2014-10-14   Cantareira           4.5            0.0
         2014-10-15   Cantareira           4.3            0.0
         2014-10-16   Cantareira           4.1            0.0
         2014-10-17   Cantareira           3.9            0.0
         2014-10-18   Cantareira           3.8            0.0
         2014-10-19   Cantareira           3.6            0.0
         2014-10-20   Cantareira           3.5           23.9
         2014-10-21   Cantareira           3.3            0.5
         2014-10-22   Cantareira           3.2            0.3
         2014-10-23   Cantareira           3.0            0.0
```

1.9 Setting

Adding a new column

```
In [23]: df['rain (m/dia)'] = df['rain (mm/dia)']/1000
         print df
```

```
           reservatorio  vol util (%)  prec (mm/dia)  prec m/dia
data
2003-01-01   Cantareira          42.5            0.0      0.0000
2003-01-02   Cantareira          42.3           11.1      0.0111
2003-01-03   Cantareira          42.3            8.6      0.0086
2003-01-04   Cantareira          42.3           14.3      0.0143
2003-01-05   Cantareira          42.4           12.7      0.0127
2003-01-06   Cantareira          42.4           23.8      0.0238
2003-01-07   Cantareira          42.5            0.0      0.0000
2003-01-08   Cantareira          42.5            0.0      0.0000
2003-01-09   Cantareira          42.6            0.0      0.0000
2003-01-10   Cantareira          42.3            0.0      0.0000
2003-01-11   Cantareira          42.4           25.0      0.0250
2003-01-12   Cantareira          42.6            8.3      0.0083
2003-01-13   Cantareira          43.0           23.3      0.0233
2003-01-14   Cantareira          42.9            2.8      0.0028
2003-01-15   Cantareira          42.8            0.0      0.0000
2003-01-16   Cantareira          42.8            0.0      0.0000
2003-01-17   Cantareira          43.3           43.4      0.0434
2003-01-18   Cantareira          44.6           42.6      0.0426
2003-01-19   Cantareira          45.3            6.4      0.0064
2003-01-20   Cantareira          46.1           30.0      0.0300
2003-01-21   Cantareira          46.7            0.0      0.0000
2003-01-22   Cantareira          48.1           49.6      0.0496
2003-01-23   Cantareira          49.2           18.2      0.0182
2003-01-24   Cantareira          50.1           22.2      0.0222
2003-01-25   Cantareira          51.1            1.1      0.0011
2003-01-26   Cantareira          52.1            4.1      0.0041
2003-01-27   Cantareira          53.3           18.5      0.0185
2003-01-28   Cantareira          54.4           16.9      0.0169
2003-01-29   Cantareira          56.0           18.2      0.0182
2003-01-30   Cantareira          57.5            4.3      0.0043
...                 ...           ...            ...         ...
2015-09-06   Cantareira          15.0            0.0      0.0000
2015-09-07   Cantareira          15.0            0.1      0.0001
2015-09-08   Cantareira          15.0           17.5      0.0175
2015-09-09   Cantareira          15.4           45.8      0.0458
2015-09-10   Cantareira          15.6            0.2      0.0002
2015-09-11   Cantareira          15.7           23.8      0.0238
2015-09-12   Cantareira          16.0           18.3      0.0183
2015-09-13   Cantareira          16.3            1.1      0.0011
2015-09-14   Cantareira          16.4            0.0      0.0000
2015-09-15   Cantareira          16.5            0.0      0.0000
2015-09-16   Cantareira          16.5            0.0      0.0000
2015-09-17   Cantareira          16.5            0.0      0.0000
2015-09-18   Cantareira          16.5            0.0      0.0000
2015-09-19   Cantareira          16.4            0.0      0.0000
2015-09-20   Cantareira          16.4            0.0      0.0000
```

```
2015-09-21   Cantareira         16.3          0.0      0.0000
2015-09-22   Cantareira         16.3          0.0      0.0000
2015-09-23   Cantareira         16.2          0.0      0.0000
2015-09-24   Cantareira         16.1          0.0      0.0000
2015-09-25   Cantareira         16.0          0.0      0.0000
2015-09-26   Cantareira         16.1         29.6      0.0296
2015-09-27   Cantareira         16.2          2.0      0.0020
2015-09-28   Cantareira         16.2          9.6      0.0096
2015-09-29   Cantareira         16.2          0.2      0.0002
2015-09-30   Cantareira         16.2          4.2      0.0042
2015-10-01   Cantareira         16.4         28.7      0.0287
2015-10-02   Cantareira         16.5          0.0      0.0000
2015-10-03   Cantareira         16.6         11.8      0.0118
2015-10-04   Cantareira         16.7          0.2      0.0002
2015-10-05   Cantareira         16.7          0.9      0.0009

[4661 rows x 4 columns]
```

1.10 Working with Missing data

Pandas uses the object np.nan to represent missing data. Important note: it is by default not included in the operations.

Methods to deal with missing data * .isnull * dropna * fillna

To drop the NaN from the dataframe

```
In [21]: df.dropna(how='any')
```

```
Out[21]:              reservoir  volume (%)  rain (mm/dia)
         dates
         2003-01-01  Cantareira        42.5            0.0
         2003-01-03  Cantareira        42.3            8.6
         2003-01-05  Cantareira        42.4           12.7
         2003-01-06  Cantareira        42.4           23.8
         2003-01-07  Cantareira        42.5            0.0
         2003-01-08  Cantareira        42.5            0.0
         2003-01-09  Cantareira        42.6            0.0
         2003-01-10  Cantareira        42.3            0.0
         2003-01-11  Cantareira        42.4           25.0
         2003-01-12  Cantareira        42.6            8.3
         2003-01-13  Cantareira        43.0           23.3
         2003-01-14  Cantareira        42.9            2.8
         2003-01-15  Cantareira        42.8            0.0
         2003-01-16  Cantareira        42.8            0.0
         2003-01-17  Cantareira        43.3           43.4
         2003-01-18  Cantareira        44.6           42.6
         2003-01-19  Cantareira        45.3            6.4
         2003-01-20  Cantareira        46.1           30.0
```

```
2003-01-21  Cantareira        46.7             0.0
2003-01-22  Cantareira        48.1            49.6
2003-01-23  Cantareira        49.2            18.2
2003-01-24  Cantareira        50.1            22.2
2003-01-25  Cantareira        51.1             1.1
2003-01-26  Cantareira        52.1             4.1
2003-01-27  Cantareira        53.3            18.5
2003-01-28  Cantareira        54.4            16.9
2003-01-29  Cantareira        56.0            18.2
2003-01-30  Cantareira        57.5             4.3
2003-01-31  Cantareira        59.1             5.1
2003-02-01  Cantareira        60.4            10.8
...                ...         ...             ...
2015-09-06  Cantareira        15.0             0.0
2015-09-07  Cantareira        15.0             0.1
2015-09-08  Cantareira        15.0            17.5
2015-09-09  Cantareira        15.4            45.8
2015-09-10  Cantareira        15.6             0.2
2015-09-11  Cantareira        15.7            23.8
2015-09-12  Cantareira        16.0            18.3
2015-09-13  Cantareira        16.3             1.1
2015-09-14  Cantareira        16.4             0.0
2015-09-15  Cantareira        16.5             0.0
2015-09-16  Cantareira        16.5             0.0
2015-09-17  Cantareira        16.5             0.0
2015-09-18  Cantareira        16.5             0.0
2015-09-19  Cantareira        16.4             0.0
2015-09-20  Cantareira        16.4             0.0
2015-09-21  Cantareira        16.3             0.0
2015-09-22  Cantareira        16.3             0.0
2015-09-23  Cantareira        16.2             0.0
2015-09-24  Cantareira        16.1             0.0
2015-09-25  Cantareira        16.0             0.0
2015-09-26  Cantareira        16.1            29.6
2015-09-27  Cantareira        16.2             2.0
2015-09-28  Cantareira        16.2             9.6
2015-09-29  Cantareira        16.2             0.2
2015-09-30  Cantareira        16.2             4.2
2015-10-01  Cantareira        16.4            28.7
2015-10-02  Cantareira        16.5             0.0
2015-10-03  Cantareira        16.6            11.8
2015-10-04  Cantareira        16.7             0.2
2015-10-05  Cantareira        16.7             0.9

[4659 rows x 3 columns]
```

To fill NaN with a value

```
In [22]: df.fillna(value=df.mean())
```

```
Out[22]:             reservoir  volume (%)  rain (mm/dia)
         dates
         2003-01-01  Cantareira   42.500000            0.0
         2003-01-02  Cantareira   52.512578           11.1
         2003-01-03  Cantareira   42.300000            8.6
         2003-01-04  Cantareira   52.512578           14.3
         2003-01-05  Cantareira   42.400000           12.7
         2003-01-06  Cantareira   42.400000           23.8
         2003-01-07  Cantareira   42.500000            0.0
         2003-01-08  Cantareira   42.500000            0.0
         2003-01-09  Cantareira   42.600000            0.0
         2003-01-10  Cantareira   42.300000            0.0
         2003-01-11  Cantareira   42.400000           25.0
         2003-01-12  Cantareira   42.600000            8.3
         2003-01-13  Cantareira   43.000000           23.3
         2003-01-14  Cantareira   42.900000            2.8
         2003-01-15  Cantareira   42.800000            0.0
         2003-01-16  Cantareira   42.800000            0.0
         2003-01-17  Cantareira   43.300000           43.4
         2003-01-18  Cantareira   44.600000           42.6
         2003-01-19  Cantareira   45.300000            6.4
         2003-01-20  Cantareira   46.100000           30.0
         2003-01-21  Cantareira   46.700000            0.0
         2003-01-22  Cantareira   48.100000           49.6
         2003-01-23  Cantareira   49.200000           18.2
         2003-01-24  Cantareira   50.100000           22.2
         2003-01-25  Cantareira   51.100000            1.1
         2003-01-26  Cantareira   52.100000            4.1
         2003-01-27  Cantareira   53.300000           18.5
         2003-01-28  Cantareira   54.400000           16.9
         2003-01-29  Cantareira   56.000000           18.2
         2003-01-30  Cantareira   57.500000            4.3
         ...                ...         ...            ...
         2015-09-06  Cantareira   15.000000            0.0
         2015-09-07  Cantareira   15.000000            0.1
         2015-09-08  Cantareira   15.000000           17.5
         2015-09-09  Cantareira   15.400000           45.8
         2015-09-10  Cantareira   15.600000            0.2
         2015-09-11  Cantareira   15.700000           23.8
         2015-09-12  Cantareira   16.000000           18.3
         2015-09-13  Cantareira   16.300000            1.1
         2015-09-14  Cantareira   16.400000            0.0
         2015-09-15  Cantareira   16.500000            0.0
         2015-09-16  Cantareira   16.500000            0.0
         2015-09-17  Cantareira   16.500000            0.0
         2015-09-18  Cantareira   16.500000            0.0
         2015-09-19  Cantareira   16.400000            0.0
         2015-09-20  Cantareira   16.400000            0.0
```

```
2015-09-21  Cantareira  16.300000         0.0
2015-09-22  Cantareira  16.300000         0.0
2015-09-23  Cantareira  16.200000         0.0
2015-09-24  Cantareira  16.100000         0.0
2015-09-25  Cantareira  16.000000         0.0
2015-09-26  Cantareira  16.100000        29.6
2015-09-27  Cantareira  16.200000         2.0
2015-09-28  Cantareira  16.200000         9.6
2015-09-29  Cantareira  16.200000         0.2
2015-09-30  Cantareira  16.200000         4.2
2015-10-01  Cantareira  16.400000        28.7
2015-10-02  Cantareira  16.500000         0.0
2015-10-03  Cantareira  16.600000        11.8
2015-10-04  Cantareira  16.700000         0.2
2015-10-05  Cantareira  16.700000         0.9

[4661 rows x 3 columns]
```

To interpolate where their is missing data

```
In [23]: df.interpolate("linear")
```

```
Out[23]:              reservoir  volume (%)  rain (mm/dia)
         dates
         2003-01-01  Cantareira       42.50           0.0
         2003-01-02  Cantareira       42.40          11.1
         2003-01-03  Cantareira       42.30           8.6
         2003-01-04  Cantareira       42.35          14.3
         2003-01-05  Cantareira       42.40          12.7
         2003-01-06  Cantareira       42.40          23.8
         2003-01-07  Cantareira       42.50           0.0
         2003-01-08  Cantareira       42.50           0.0
         2003-01-09  Cantareira       42.60           0.0
         2003-01-10  Cantareira       42.30           0.0
         2003-01-11  Cantareira       42.40          25.0
         2003-01-12  Cantareira       42.60           8.3
         2003-01-13  Cantareira       43.00          23.3
         2003-01-14  Cantareira       42.90           2.8
         2003-01-15  Cantareira       42.80           0.0
         2003-01-16  Cantareira       42.80           0.0
         2003-01-17  Cantareira       43.30          43.4
         2003-01-18  Cantareira       44.60          42.6
         2003-01-19  Cantareira       45.30           6.4
         2003-01-20  Cantareira       46.10          30.0
         2003-01-21  Cantareira       46.70           0.0
         2003-01-22  Cantareira       48.10          49.6
         2003-01-23  Cantareira       49.20          18.2
         2003-01-24  Cantareira       50.10          22.2
```

```
2003-01-25  Cantareira       51.10            1.1
2003-01-26  Cantareira       52.10            4.1
2003-01-27  Cantareira       53.30           18.5
2003-01-28  Cantareira       54.40           16.9
2003-01-29  Cantareira       56.00           18.2
2003-01-30  Cantareira       57.50            4.3
...                ...         ...            ...
2015-09-06  Cantareira       15.00            0.0
2015-09-07  Cantareira       15.00            0.1
2015-09-08  Cantareira       15.00           17.5
2015-09-09  Cantareira       15.40           45.8
2015-09-10  Cantareira       15.60            0.2
2015-09-11  Cantareira       15.70           23.8
2015-09-12  Cantareira       16.00           18.3
2015-09-13  Cantareira       16.30            1.1
2015-09-14  Cantareira       16.40            0.0
2015-09-15  Cantareira       16.50            0.0
2015-09-16  Cantareira       16.50            0.0
2015-09-17  Cantareira       16.50            0.0
2015-09-18  Cantareira       16.50            0.0
2015-09-19  Cantareira       16.40            0.0
2015-09-20  Cantareira       16.40            0.0
2015-09-21  Cantareira       16.30            0.0
2015-09-22  Cantareira       16.30            0.0
2015-09-23  Cantareira       16.20            0.0
2015-09-24  Cantareira       16.10            0.0
2015-09-25  Cantareira       16.00            0.0
2015-09-26  Cantareira       16.10           29.6
2015-09-27  Cantareira       16.20            2.0
2015-09-28  Cantareira       16.20            9.6
2015-09-29  Cantareira       16.20            0.2
2015-09-30  Cantareira       16.20            4.2
2015-10-01  Cantareira       16.40           28.7
2015-10-02  Cantareira       16.50            0.0
2015-10-03  Cantareira       16.60           11.8
2015-10-04  Cantareira       16.70            0.2
2015-10-05  Cantareira       16.70            0.9

[4661 rows x 3 columns]
```

2 Operations

Pandas includes a lot of methods to perform operations along an axis.

- count: Number of non-null observations
- sum: Sum of values
- mean: Mean of values

- mad: Mean absolute deviation
- median: Arithmetic median of values
- min: Minimum
- max: Maximum
- mode: Mode
- abs: Absolute Value
- prod: Product of values
- std: Bessel-corrected sample standard deviation
- var: Unbiased variance
- sem: Standard error of the mean
- skew: Sample skewness (3rd moment)
- kurt: Sample kurtosis (4th moment)
- quantile: Sample quantile (value at %)
- cumsum: Cumulative sum
- cumprod: Cumulative product
- cummax: Cumulative maximum
- cummin: Cumulative minimum

Perform the mean of each dataframe columns.

```
In [24]: df.mean(axis=0)
```

```
Out[24]: volume (%)       52.512578
         rain (mm/dia)     3.787428
         dtype: float64
```

Apply You can also pass a function along an axis with the apply method. This method is very efficient to iterate along the axis, much faster than a for loop for example.

```
In [ ]: df['rain (mm/dia)'].apply(np.sqrt)
```

2.1 Merge data structures

Pandas provide different methods to merge Series, Dataframe and Paneis:

- Merge
- join
- concat
- append

Let's merge our dataframe with the observed temperature from external file.

```
In [26]: Path = "./data/"
         filename = "temperature.txt"

         df_temp = pd.read_csv(Path+filename, index_col =0, parse_dates=True)
         print df_temp
```

```
                      temperature (C)
dates
2014-09-03 00:34:00             14.40
2014-09-03 00:36:00             12.14
2014-09-03 00:38:00             12.09
2014-09-03 00:40:00             12.05
2014-09-03 00:42:00             12.05
2014-09-03 00:44:00             12.00
2014-09-03 00:46:00             11.95
2014-09-03 00:48:00             11.90
2014-09-03 00:50:00             11.90
2014-09-03 00:52:00             11.90
2014-09-03 00:54:00             11.90
2014-09-03 00:56:00             11.85
2014-09-03 00:58:00             11.85
2014-09-03 01:00:00             11.81
2014-09-03 01:02:00             11.76
2014-09-03 01:04:00             11.71
2014-09-03 01:06:00             11.66
2014-09-03 01:08:00             11.61
2014-09-03 01:10:00             11.57
2014-09-03 01:12:00             11.58
2014-09-03 01:14:00             11.58
2014-09-03 01:16:00             11.47
2014-09-03 01:18:00             11.54
2014-09-03 01:20:00             11.49
2014-09-03 01:22:00             11.42
2014-09-03 01:24:00             11.37
2014-09-03 01:26:00             11.37
2014-09-03 01:28:00             11.42
2014-09-03 01:30:00             11.49
2014-09-03 01:32:00             11.44
...                               ...
2015-08-07 15:42:00             25.90
2015-08-07 15:44:00             26.00
2015-08-07 15:46:00             26.00
2015-08-07 15:48:00             26.10
2015-08-07 15:50:00             26.10
2015-08-07 15:52:00             26.10
2015-08-07 15:54:00             26.00
2015-08-07 15:56:00             25.90
2015-08-07 15:58:00             25.70
2015-08-07 16:00:00             25.70
2015-08-07 16:02:00             25.90
2015-08-07 16:04:00             25.90
2015-08-07 16:06:00             25.80
2015-08-07 16:08:00             25.60
2015-08-07 16:10:00             25.60
```

```
2015-08-07 16:12:00             25.40
2015-08-07 16:14:00             25.20
2015-08-07 16:16:00             25.10
2015-08-07 16:18:00             25.00
2015-08-07 16:20:00             25.00
2015-08-07 16:22:00             24.90
2015-08-07 16:24:00             24.90
2015-08-07 16:26:00             25.00
2015-08-07 16:28:00             25.00
2015-08-07 16:30:00             24.90
2015-08-07 16:32:00             24.50
2015-08-07 16:34:00             24.20
2015-08-07 16:36:00             23.80
2015-08-07 16:38:00             23.50
2015-08-07 16:40:00             23.30

[243844 rows x 1 columns]
```

2.1.1 Resample

df temp does not have the same time frequency. Therfore, It is necessary to resample this dataframe by day prior to merge.

```
In [28]: df_temp_day = df_temp.resample('D', how='mean')
         print df_temp_day
```

```
             temperature (C)
dates
2014-09-03         16.311963
2014-09-04         15.392722
2014-09-05         15.154472
2014-09-06         14.241111
2014-09-07         18.746944
2014-09-08         18.082222
2014-09-09         16.720417
2014-09-10         19.793889
2014-09-11         19.885694
2014-09-12         18.463056
2014-09-13         16.820972
2014-09-14         17.061944
2014-09-15         20.344444
2014-09-16         18.614861
2014-09-17         18.448542
2014-09-18         19.118472
2014-09-19         20.192778
2014-09-20         16.926861
2014-09-21         15.976819
```

```
2014-09-22          14.635500
2014-09-23          16.686528
2014-09-24          19.341944
2014-09-25          19.142222
2014-09-26          18.091319
2014-09-27          18.582514
2014-09-28          20.194736
2014-09-29          21.774167
2014-09-30          22.004583
2014-10-01          20.611417
2014-10-02          16.429278
...                       ...
2015-07-09          15.187361
2015-07-10          14.469125
2015-07-11          14.681944
2015-07-12          16.293889
2015-07-13          16.739167
2015-07-14          18.276667
2015-07-15          16.145278
2015-07-16          16.282861
2015-07-17          15.115556
2015-07-18          14.125944
2015-07-19          14.352125
2015-07-20          14.820097
2015-07-21          16.293694
2015-07-22          14.214264
2015-07-23          15.444125
2015-07-24          16.343958
2015-07-25          14.790889
2015-07-26          15.567667
2015-07-27          12.984014
2015-07-28          12.297194
2015-07-29          13.241361
2015-07-30          13.913014
2015-07-31          14.354861
2015-08-01          14.397194
2015-08-02          14.002972
2015-08-03          14.749417
2015-08-04          15.073653
2015-08-05          14.243431
2015-08-06          14.959389
2015-08-07          16.004291

[339 rows x 1 columns]
```

2.1.2 Merge

We use the argument join='inner' to keep only the index labels that are present in both dataframe. But we could have use join='outer' to keep all the labeles of each dataframes

```
In [30]: df_merged = pd.concat([df, df_temp_day], axis=1, join='inner')
         print df_merged
```

```
             reservoir  volume (%)  rain (mm/dia)   temperature (C)
dates
2014-09-03  Cantareira        10.7           22.2         16.311963
2014-09-04  Cantareira        10.6            0.2         15.392722
2014-09-05  Cantareira        10.5            0.0         15.154472
2014-09-06  Cantareira        10.4            0.0         14.241111
2014-09-07  Cantareira        10.2            0.0         18.746944
2014-09-08  Cantareira        10.1            0.0         18.082222
2014-09-09  Cantareira        10.0            0.0         16.720417
2014-09-10  Cantareira         9.8            0.0         19.793889
2014-09-11  Cantareira         9.7            0.0         19.885694
2014-09-12  Cantareira         9.5            0.0         18.463056
2014-09-13  Cantareira         9.4            0.0         16.820972
2014-09-14  Cantareira         9.2            0.0         17.061944
2014-09-15  Cantareira         9.1            0.0         20.344444
2014-09-16  Cantareira         8.9            0.0         18.614861
2014-09-17  Cantareira         8.9            0.0         18.448542
2014-09-18  Cantareira         8.6            0.0         19.118472
2014-09-19  Cantareira         8.4            0.0         20.192778
2014-09-20  Cantareira         8.2            2.2         16.926861
2014-09-21  Cantareira         8.1            6.8         15.976819
2014-09-22  Cantareira         8.0            0.9         14.635500
2014-09-23  Cantareira         7.8            0.0         16.686528
2014-09-24  Cantareira         7.6            0.0         19.341944
2014-09-25  Cantareira         7.4            0.0         19.142222
2014-09-26  Cantareira         7.2            0.2         18.091319
2014-09-27  Cantareira         7.2           22.7         18.582514
2014-09-28  Cantareira         7.1            0.1         20.194736
2014-09-29  Cantareira         7.0            0.0         21.774167
2014-09-30  Cantareira         6.9            3.1         22.004583
2014-10-01  Cantareira         6.7            0.0         20.611417
2014-10-02  Cantareira         6.6            0.3         16.429278
...                ...         ...            ...               ...
2015-07-09  Cantareira        19.6            1.6         15.187361
2015-07-10  Cantareira        19.6            0.1         14.469125
2015-07-11  Cantareira        19.6            0.1         14.681944
2015-07-12  Cantareira        19.5            1.7         16.293889
2015-07-13  Cantareira        19.5            0.1         16.739167
2015-07-14  Cantareira        19.4            0.0         18.276667
2015-07-15  Cantareira        19.4            0.0         16.145278
2015-07-16  Cantareira        19.3            0.0         16.282861
```

```
2015-07-17  Cantareira        19.3            0.0         15.115556
2015-07-18  Cantareira        19.2            0.0         14.125944
2015-07-19  Cantareira        19.2            0.1         14.352125
2015-07-20  Cantareira        19.1            0.0         14.820097
2015-07-21  Cantareira        19.0            0.0         16.293694
2015-07-22  Cantareira        19.0            0.0         14.214264
2015-07-23  Cantareira        18.9            0.0         15.444125
2015-07-24  Cantareira        18.9            1.0         16.343958
2015-07-25  Cantareira        18.8           15.1         14.790889
2015-07-26  Cantareira        18.8            5.1         15.567667
2015-07-27  Cantareira        18.9            0.3         12.984014
2015-07-28  Cantareira        18.8            0.0         12.297194
2015-07-29  Cantareira        18.8            0.2         13.241361
2015-07-30  Cantareira        18.8            0.2         13.913014
2015-07-31  Cantareira        18.7            0.0         14.354861
2015-08-01  Cantareira        18.7            0.2         14.397194
2015-08-02  Cantareira        18.6            0.1         14.002972
2015-08-03  Cantareira        18.5            0.1         14.749417
2015-08-04  Cantareira        18.4            0.1         15.073653
2015-08-05  Cantareira        18.3            0.2         14.243431
2015-08-06  Cantareira        18.2            0.0         14.959389
2015-08-07  Cantareira        18.1            0.0         16.004291

[339 rows x 4 columns]
```

2.2 Avanced manipulation

From this stage the true power of Pandas is unleashed.

2.2.1 Grouping : split-apply-combine

Grouping with the method group-by consist of a 3 step process:

- Spliting the data with some criteria
- Applying a function to each group.
- Combining the results in a data structure

In the following example, we want to perform the period average by months.

1. Create a new column with the month number
2. Pass the column month as a criteria
3. Apply the function mean() on each group

For more information please see the documentation http://pandas.pydata.org/pandas-docs/stable/groupby.html

```
In [66]: df['month'] = df.index.month
         grouped = df.groupby('month').mean()
         print grouped
         vol (%)  rain (mm/dia)
month
1      47.567332       8.996030
2      52.826975       6.413079
3      56.900993       5.256328
4      59.690513       2.775128
5      59.976675       1.907444
6      59.488462       1.894872
7      57.018362       2.052109
8      53.547891       0.665261
9      48.410769       2.272564
10     45.833156       4.024403
11     43.670278       5.612778
12     43.477957       6.574462
```

2.2.2 Pivot Table

Create pivot table spread-sheet like with multi-index.

- data: A DataFrame object
- values: a column or a list of columns to aggregate
- index: a column, Grouper, array which has the same length as data, or list of them. Keys to group by on the pivot table index. If an array is passed, it is being used as the same manner as column values.
- columns: a column, Grouper, array which has the same length as data, or list of them. Keys to group by on the pivot table column. If an array is passed, it is being used as the same manner as column values.
- aggfunc: function to use for aggregation, defaulting to numpy.mean

In the following example we want to restructure the dataframe to see the average volume and precipitation by month and year.

```
In [72]: df['month'] = df.index.month
         df['year']= df.index.year
```

```
In [76]: pt = pd.pivot_table(df, values=['volume (%)', "rain (mm/dia)"], index=['month','year'])
         print pt
```

```
            rain (mm/dia)
month year
1     2003      13.241935
      2004       5.080645
      2005       9.122581
      2006       6.780645
      2007      10.806452
      2008       7.829032
      2009       9.503226
      2010      15.667742
      2011      15.693548
      2012      10.861290
      2013       4.735484
      2014       2.845161
      2015       4.780645
2     2003       3.607143
      2004       8.813793
      2005       4.875000
      2006      10.117857
      2007       3.825000
      2008       5.817241
      2009       8.328571
      2010       5.850000
      2011       5.853571
      2012       3.293103
      2013       8.900000
      2014       2.617857
      2015      11.517857
3     2003       3.335484
      2004       3.400000
      2005       8.074194
      2006       8.606452
...                   ...
10    2010       2.958065
      2011       3.993548
      2012       3.512903
      2013       3.990323
      2014       1.367742
      2015       8.320000
11    2003       4.956667
      2004       6.800000
      2005       5.030000
      2006       6.146667
      2007       8.453333
      2008       5.070000
      2009       7.916667
      2010       5.193333
      2011       5.913333
      2012       4.136667
      2013       3.236667
      2014       4.500000
12    2003       6.380645
      2004       5.019355
      2005       6.522581
      2006       9.161290
      2007       5.341935
      2008       5.535484
      2009      13.506452
      2010       7.958065
      2011       3.806452
      2012       8.283871
      2013       2.038710
      2014       5.338710

[154 rows x 1 columns]
```

The new datastructure has a multi-index along the axis 0. Pandas object can handle multiple index on multiple axis.

```
In [77]: pt.index

Out[77]: MultiIndex(levels=[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [2003, 2004, 2005, 2006, 20
                    labels=[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
                    names=[u'month', u'year'])
```

2.3 Writing data structure to file

```
In [32]: df.to_csv('./my_results.csv')
```

3 Plots

Pandas module has some basic plot functionality built on top of matplotlib

```
In [39]: df.loc[:,"volume (%)"].resample('A', how='mean').plot(kind="bar")
         plt.show()

In [40]: df.loc[:,"volume (%)"].resample('A', how='mean').plot.area(stacked=False)
         plt.show()
```