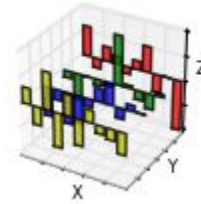
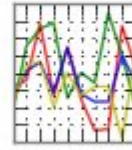
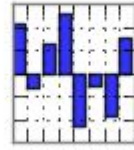


pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



# ***Pandas 2-minutes overview***

*T.C, Jan 2015, Pynxton meeting*

<http://pandas.pydata.org/>

# What is Pandas

- *pandas is an open source library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. (Pandas web site)*
- *Python is great for data munging and preparation, but less so for data analysis and modeling. pandas helps fill this gap, enabling you to carry out your entire data analysis workflow in Python without having to switch to a more domain specific language like R. (Pandas web site)*

# What Pandas does not

- *“pandas does not implement significant modeling functionality outside of linear and panel regression; for this, look to statsmodels and scikit-learn. More work is still needed to make Python a first class statistical modeling environment, but we are well on our way toward that goal.”* (Pandas web site)

# Some features

- Provide a **DataFrame** (see video) like in R
- Reading/Writing data (e.g., CSV reader, Excel , SQL, **HDF5**)
- Data **alignement**
- Integrate **missing values**
- Reshaping
- Aggregating or transformation of data using e.g., '**group by**'
- Integrated plotting
- Hiearchical axis indexing
- And so on

# Installation

- Available on pip
- If numpy and matplotlib are installed, no problem to be installed.

# Example

## DataFrame creation

```
In [8]: df = pd.DataFrame(np.random.randn(6,4), index=dates, columns=
```

```
In [9]: df
```

```
Out[9]:
```

	A	B	C	D
2013-01-01	0.469112	-0.282863	-1.509059	-1.135632
2013-01-02	1.212112	-0.173215	0.119209	-1.044236
2013-01-03	-0.861849	-2.104569	-0.494929	1.071804
2013-01-04	0.721555	-0.706771	-1.039575	0.271860
2013-01-05	-0.424972	0.567020	0.276232	-1.087401
2013-01-06	-0.673690	0.113648	-1.478427	0.524988

## Access to a column

```
In [23]: df['A']
```

```
Out[23]:
```

2013-01-01	0.469112
2013-01-02	1.212112
2013-01-03	-0.861849
2013-01-04	0.721555
2013-01-05	-0.424972
2013-01-06	-0.673690

Freq: D, Name: A, dtype: float64

## Easy stats

```
In [61]: df.mean()
```

```
Out[61]:
```

A	-0.004474
B	-0.383981
C	-0.687758
D	5.000000
F	3.000000

dtype: float64

# Example

```
In [20]: plt.figure();
```

```
In [21]: df4.plot(kind='hist', stacked=True, bins=20)
```

```
Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0xae5286c>
```

