Pandas tutorial

Pandas is a Python package designed to make working with tabular (Excel-like) data easier. A Pandas DataFrame is similar to a Matlab DataMatrix. In addition to numerical matrix, it stores and manages row (the "index") and column names. See online documentation and examples at [http://pandas.pydata.org/pandas-docs/stable/10min.html (http://pandas.pydata.org/pandas-docs/stable/10min.html)](http://pandas.pydata.org/pandas-docs/stable/10min.html)

In [1]:

```
import pandas as pd
import numpy as np

# Read the disease correlation information from the Excel file
d=pd.read_excel('gse7307disease.xlsx', 'corr');
# Jupyter will show a "nice" table printout of this variable.
d
```

Out[1]:

	Adenom	BPH	Breast	Endome	Melano	Neurom	Normal	Parkin	
Adenom	1.000000	0.821606	0.691835	0.739842	0.693322	0.522752	0.857857	0.706505	0.8
BPH	0.821606	1.000000	0.763077	0.769374	0.754738	0.538281	0.844531	0.678420	0.9
Breast	0.691835	0.763077	1.000000	0.764342	0.860399	0.359795	0.801246	0.574269	0.8
Endome	0.739842	0.769374	0.764342	1.000000	0.740085	0.456253	0.785574	0.636274	0.7
Melano	0.693322	0.754738	0.860399	0.740085	1.000000	0.460511	0.838409	0.664676	0.8
Neurom	0.522752	0.538281	0.359795	0.456253	0.460511	1.000000	0.578325	0.534358	0.5
Normal	0.857857	0.844531	0.801246	0.785574	0.838409	0.578325	1.000000	0.885944	0.8
Parkin	0.706505	0.678420	0.574269	0.636274	0.664676	0.534358	0.885944	1.000000	0.6
Prosta	0.803844	0.915653	0.828230	0.772001	0.818134	0.538620	0.875217	0.691486	1.0
Rheuma	0.603496	0.715907	0.852794	0.740907	0.822432	0.378162	0.680101	0.485894	0.7
Uterin	0.963967	0.815426	0.716472	0.754143	0.701922	0.525012	0.867346	0.698339	0.8

In [2]:

```
d.index #rownames
```

Out[2]:

```
Index(['Adenom', 'BPH', 'Breast', 'Endome', 'Melano', 'Neurom', 'Norma
l',
       'Parkin', 'Prosta', 'Rheuma', 'Uterin'],
      dtype='object')
```

In [3]:

```
d.columns #columnnames
```

Out[3]:

```
Index(['Adenom', 'BPH', 'Breast', 'Endome', 'Melano', 'Neurom', 'Norma
l',
       'Parkin', 'Prosta', 'Rheuma', 'Uterin'],
      dtype='object')
```

In [4]:
```
d.values #the actual data, stored as a numpy array.
```

Out[4]:
```
array([[ 1.        ,  0.82160643,  0.69183514,  0.7398424 ,  0.69332242,
         0.52275248,  0.857857  ,  0.70650532,  0.80384427,  0.60349566,
         0.96396697],
       [ 0.82160643,  1.        ,  0.7630769 ,  0.76937438,  0.75473753,
         0.53828065,  0.84453145,  0.67842038,  0.91565263,  0.71590704,
         0.81542623],
       [ 0.69183514,  0.7630769 ,  1.        ,  0.76434231,  0.86039855,
         0.35979507,  0.80124647,  0.57426929,  0.82822996,  0.85279433,
         0.71647211],
       [ 0.7398424 ,  0.76937438,  0.76434231,  1.        ,  0.74008459,
         0.45625252,  0.78557422,  0.63627418,  0.77200143,  0.74090677,
         0.75414337],
       [ 0.69332242,  0.75473753,  0.86039855,  0.74008459,  1.        ,
         0.46051085,  0.83840866,  0.66467624,  0.81813432,  0.82243183,
         0.70192225],
       [ 0.52275248,  0.53828065,  0.35979507,  0.45625252,  0.46051085,
         1.        ,  0.57832544,  0.53435783,  0.53862035,  0.3781616 ,
         0.5250118 ],
       [ 0.857857  ,  0.84453145,  0.80124647,  0.78557422,  0.83840866,
         0.57832544,  1.        ,  0.88594409,  0.87521701,  0.68010115,
         0.86734623],
       [ 0.70650532,  0.67842038,  0.57426929,  0.63627418,  0.66467624,
         0.53435783,  0.88594409,  1.        ,  0.69148606,  0.48589356,
         0.69833868],
       [ 0.80384427,  0.91565263,  0.82822996,  0.77200143,  0.81813432,
         0.53862035,  0.87521701,  0.69148606,  1.        ,  0.73953737,
         0.80527085],
       [ 0.60349566,  0.71590704,  0.85279433,  0.74090677,  0.82243183,
         0.3781616 ,  0.68010115,  0.48589356,  0.73953737,  1.        ,
         0.60736336],
       [ 0.96396697,  0.81542623,  0.71647211,  0.75414337,  0.70192225,
         0.5250118 ,  0.86734623,  0.69833868,  0.80527085,  0.60736336,
         1.        ]])
```

In [5]:
```
# get a single column using d['A'] or d.A
d.Adenom
```

Out[5]:
```
Adenom    1.000000
BPH       0.821606
Breast    0.691835
Endome    0.739842
Melano    0.693322
Neurom    0.522752
Normal    0.857857
Parkin    0.706505
Prosta    0.803844
Rheuma    0.603496
Uterin    0.963967
Name: Adenom, dtype: float64
```

In [11]:

```python
# Use a list of column names to select multiple columns
d[['Adenom','BPH']]
```

Out[11]:

	Adenom	BPH
Adenom	1.000000	0.821606
BPH	0.821606	1.000000
Breast	0.691835	0.763077
Endome	0.739842	0.769374
Melano	0.693322	0.754738
Neurom	0.522752	0.538281
Normal	0.857857	0.844531
Parkin	0.706505	0.678420
Prosta	0.803844	0.915653
Rheuma	0.603496	0.715907
Uterin	0.963967	0.815426

In [ ]:

```python

```

In [7]:

```python
# Use a list of numbers to select one or more rows.
d[0:1]
```

Out[7]:

	Adenom	BPH	Breast	Endome	Melano	Neurom	Normal	Parkin	F
Adenom	1.0	0.821606	0.691835	0.739842	0.693322	0.522752	0.857857	0.706505	0.8

In [8]:

```python
d[0:3]
```

Out[8]:

	Adenom	BPH	Breast	Endome	Melano	Neurom	Normal	Parkin	
Adenom	1.000000	0.821606	0.691835	0.739842	0.693322	0.522752	0.857857	0.706505	0.8
BPH	0.821606	1.000000	0.763077	0.769374	0.754738	0.538281	0.844531	0.678420	0.9
Breast	0.691835	0.763077	1.000000	0.764342	0.860399	0.359795	0.801246	0.574269	0.8

In [9]:

```
# I recommend using d.loc or d.iloc, to be more explicit in telling row vs
# loc uses row/column names. whereas iloc uses numbers for indexing.
d.loc[['Breast','Prosta'],['Adenom','BPH']]
```

Out[9]:

	Adenom	BPH
Breast	0.691835	0.763077
Prosta	0.803844	0.915653

In [10]:

```
d.iloc[0:3,2:4]
```

Out[10]:

	Breast	Endome
Adenom	0.691835	0.739842
BPH	0.763077	0.769374
Breast	1.000000	0.764342

In [ ]: