# Lecture 13, 20 September 2022

## Pandas (Python and data analysis)

- Built on top of numpy

## Series and data frames

- Numpy defines homogeneous n-dimensional arrays
- Data science works with tables: 2-dimensional arrays
- Pandas has two fundamental data structures
  - Series : A column of data
  - Data Frame : A table of data

## Key difference

- Numpy indices are always `[0..n-1]` in each dimension
- Pandas allows more flexible “named” indices for rows and columns
  - Dictionary vs list

## Load pandas

- Don't need to import numpy unless one is separately using numpy arrays

In [1]:
```
import pandas as pd
```

## Create a series

- Convert a sequence into a series (column)

In [2]:
```
h = ('AA', '2012-02-01', 100, 10.2)
s = pd.Series(h)
type(s)
```

Out[2]:
```
pandas.core.series.Series
```

In [3]:
```
s
```

Out[3]:
```
0            AA
1    2012-02-01
2           100
3          10.2
dtype: object
```

- Note that the underlying type is `object` , array with variable types

## Convert a dictionary to a series

- Keys become "row indices"

In [4]:
```
d = {'name' : 'IBM', 'date' :'2010-09-08', 'shares' : 100, 'price' : 10.2}
ds = pd.Series(d)
type(ds)
```

Out[4]:
```
pandas.core.series.Series
```

In [5]:
```
ds
```

Out[5]:
```
name             IBM
date      2010-09-08
shares           100
price           10.2
dtype: object
```

## Creating an index

- Add an index separately when creating the series

In [6]:
```
f = ['FB', '2001-08-02', 90, 3.2]
fs = pd.Series(f, index = ['name','date', 'shares', 'price'])
```

In [7]:
```
fs
```

Out[7]:
```
name              FB
date      2001-08-02
shares            90
price            3.2
dtype: object
```

### Accessing elements

- Use named index, or position
- Use slices, sublists

In [8]:
```
fs['shares']
```

Out[8]:
```
90
```

In [9]:
```
fs[0]
```

Out[9]:
```
'FB'
```

In [10]:
```
fs[0:2]
```

Out[10]:
```
name            FB
date    2001-08-02
dtype: object
```

- Can extract arbitrary subset of columns, in any order

In [11]:
```
fs[[0,2]]
```

Out[11]:
```
name      FB
shares    90
dtype: object
```

In [12]:
```
fs[['price','name']]
```

Out[12]:
```
price    3.2
name      FB
dtype: object
```

- Slice using index labels includes endpoint, unlike positional slice

In [13]:
```
fs['name':'price']
```

Out[13]:
```
name              FB
date      2001-08-02
shares            90
price            3.2
dtype: object
```

In [14]:
```
fs[0:3]
```

Out[14]:
```
name              FB
date      2001-08-02
shares            90
dtype: object
```

### Data frames

- A table is a sequence of columns
- A data frame is a sequence of series

In [15]:
```
data1 = {'name' : ['AA', 'IBM', 'GOOG'],
        'date' : ['2001-12-01', '2012-02-10', '2010-04-09'],
        'shares' : [100, 30, 90],
        'price' : [12.3, 10.3, 32.2]
}
df1 = pd.DataFrame(data1)
df1
```

Out[15]:

| | name | date | shares | price |
|---|---|---|---|---|
| **0** | AA | 2001-12-01 | 100 | 12.3 |
| **1** | IBM | 2012-02-10 | 30 | 10.3 |
| **2** | GOOG | 2010-04-09 | 90 | 32.2 |

- If you create a data frame without column labels, the sequences are interpreted as rows!

In [16]:
```
data2 = (['AA', 'IBM', 'GOOG'],
         ['2001-12-01', '2012-02-10', '2010-04-09'],
         [100, 30, 90],
         [12.3, 10.3, 32.2])
df2 = pd.DataFrame(data2)
df2
```

Out[16]:

| | 0 | 1 | 2 |
|---|---|---|---|
| **0** | AA | IBM | GOOG |
| **1** | 2001-12-01 | 2012-02-10 | 2010-04-09 |
| **2** | 100 | 30 | 90 |
| **3** | 12.3 | 10.3 | 32.2 |