Relevant Python modules: pandas

AM

Pandas

- Conceived by Wes McKinney. Quantitative analyst for hedge fund AQR. ::: {.nonincremental}

  -It is a library for processing tabular data, both numeric and time series.
  -Provides data structures (series, dataframe) and methods for data analysis.
  Python for Data Analysis. Wes McKinney :::

  ```
  1  pip install pandas
  ```

  Available by default with Anaconda.

Data Structures - Series

A one dimension object containing values, and associated labels called Index.
Unless we assign indices, P. will simply enumerate the items.

```
0    10
1    20
2    30
3    40
dtype: int64


a    10
b    20
c    30
d    40
dtype: int64


a    10
b    20
c    30
d    40
dtype: int64
```

Data Structures - Series

You can use the index to select one or more specific values.

```
10


a    10
c    30
dtype: int64
```

You can filter elements

```
a    10
b    20
dtype: int64
```

apply element-wise mathematical operations

```
a     100
b     400
c     900
d    1600
dtype: int64
```

or a combination of both

```
a    100
b    400
dtype: int64
```

Data Structures - DataFrame

In Pandas, DataFrames are 2D structures. Values are labelled by their index and column location.
Example: set up a DataFrame

	Integers
a	10
b	20
c	30
d	40

	Integers	Floats
a	10	1.5
b	20	2.5
c	30	3.5
d	40	4.5

Data Structures: DataFrame - ‘loc’

You can select specific data according to its location label.

```
Integers    30.0
Floats       3.5
Name: c, dtype: float64


a    10
b    20
c    30
d    40
Name: Integers, dtype: int64


30
```

Data Structures: DataFrame - ‘iloc’

Select a specific slice of data according to its position.

```
Integers    30.0
Floats       3.5
Name: c, dtype: float64


a    10
b    20
c    30
d    40
Name: Integers, dtype: int64


30
```

Data Structures: DataFrame - filters

Complex selection is achieved applying Boolean filters. Multiple conditions can be combined in one statement.

	Integers	Floats
b	20	2.5
c	30	3.5
d	40	4.5

	Integers	Floats
c	30	3.5
d	40	4.5

Data Structures: DataFrame - Axis

DataFrames operate on 2 dimensions.
Axis = 0 invokes functions across rows. This is the default behaviour if axis is not specified.

```
Integers    100.0
Floats       12.0
dtype: float64
```

Axis = 1 invokes functions across columns.

```
a    11.5
b    22.5
c    33.5
d    44.5
dtype: float64
```

Data Structures: DataFrame - Axis

We can mix element-wise operations with functions applied to a given axis
Example: Create a column with the sum of squares of each row.

	Integers	Floats	sumsq
a	10	1.5	102.25
b	20	2.5	406.25
c	30	3.5	912.25
d	40	4.5	1620.25

Reading a file into pandas

Pandas can read a file and turn it into a DataFrame. Several arguments are available to specify the behavior of the process.
Some examles: index_col sets the column of the csv file to be used as index of the DataFrame sep specifies the separator in the source file parse_dates indicates which column to be converted as a datetime objects

```
1  FILE = 'some_file.csv'
2
3  df_r = pd.read_csv(FILE,
4                     index_col = 0,
5                     sep = ';',
6                     parse_dates = ['date'
```

Biostats data - info()

The info() method outputs top-down information on the DataFrame

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Name          18 non-null     object
 1   Sex           18 non-null     object
 2   Age           18 non-null     int64
 3   Height(in)    18 non-null     int64
 4   Weight(lbs)   18 non-null     int64
dtypes: int64(3), object(2)
memory usage: 852.0+ bytes
```

Biostats data - head() and tail()

These convenient methods visualise respectively the first/last n rows (default = 5) in the DataFrame.

	Name	Sex	Age	Height(in)	W
0	Alex	M	41	74	17
1	Bert	M	42	68	16
2	Dave	M	32	70	15
3	Dave	M	39	72	16
4	Elly	F	30	66	12

	Name	Sex	Age	Height(in)	
13	Neil	M	36	75	1
14	Omar	M	38	70	1
15	Page	F	31	67	1
16	Luke	M	29	71	1
17	Ruth	F	28	65	1

Biostats data - index column

Selecting the index column affects the structure of the DataFrame and information retrieval. CAUTION: the index does not have to be unique. Multiple rows could have the same index name.

Name	Sex	Age	Height(in)	We
Alex	M	41	74	170
Bert	M	42	68	166
Dave	M	32	70	155
Dave	M	39	72	167
Elly	F	30	66	124

```
Sex              M
Age             42
Height(in)      68
Weight(lbs)    166
Name: Bert, dtype: object
```

Descriptive statistics - describe()

Pandas selects quantitative variables and computes descriptive statistics

	Age	Height(in)	Weight(lbs
count	18.000000	18.000000	18.000000
mean	34.666667	69.055556	146.722222
std	7.577055	3.522570	22.540958
min	23.000000	62.000000	98.000000
25%	30.000000	66.250000	132.000000
50%	32.500000	69.500000	150.000000
75%	38.750000	71.750000	165.250000
max	53.000000	75.000000	176.000000

```
count    18.000000
mean     34.666667
std       7.577055
min      23.000000
25%      30.000000
50%      32.500000
75%      38.750000
max      53.000000
Name: Age, dtype: float64
```

Descriptive statistics - categorcal variables

The value_counts() method computes the unique values and how many time they occurr.

```
Sex
M    11
F     7
Name: count, dtype: int64
```