Introduction to Pandas

```python
#Import Python Libraries
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib as mpl
```

Reading data using pandas

```
#Read csv file
df = pd.read_csv(“iris.csv")
```

There is a number of pandas commands to read other data formats:

```
pd.read_excel(‘iris.xlsx',sheet_name='Sheet1', index_col=None, na_values=['NA'])

pd.read_stata(‘iris.dta')

pd.read_sas(‘iris.sas7bdat')

pd.read_hdf(‘iris.h5','df')
```

Dataframe attributes

df.attribute	description
dtypes	list the types of the columns
columns	list the column names
axes	list the row labels and column names
ndim	number of dimensions
size	number of elements
shape	return a tuple representing the dimensionality
values	numpy representation of the data

Dataframe methods

df.method()	description
head([n]), tail([n])	first/last n rows
describe()	generate descriptive statistics (for numeric columns only)
max(), min()	return max/min values for all numeric columns
mean(), median()	return mean/median values for all numeric columns
std()	standard deviation
sample([n])	returns a random sample of the data frame
dropna()	drop all the records with missing values

Selecting a column in a Data Frame

Method 1: Subset the data frame using column name:

df['sepal_length']

Method 2: Use the column name as an attribute:

df.sepal_length

Note: there is an attribute *rank* for pandas data frames, so to select a column with a name "rank" we should use method 1.

Data Frame: filtering

To subset the data we can apply Boolean indexing also referred to as ‘Filter’
For example if we want to subset the rows in which the ‘sepal_length’ value is greater than 4.5:

```
#Extract the rows where sepal_length is more than 4.5:
df_sub = df['sepal_length'] > 4.5
```

Any Boolean operator can be used to subset the data:

> greater;	>= greater or equal;
< less;	<= less or equal;
== equal;	!= not equal;

```
#Select only those rows that contain setosa species:
df_f = df[ df['species'] == 'setosa']
```

Data Frames: Missing values

Missing values are marked as NaN

```
# Read a dataset with missing values
iris = pd.read_csv('iris.csv')
```

```
# Select the rows that have at least one missing value
iris[iris.isnull().any(axis=1)].head()
```

df.method()	description
dropna()	Drop missing observations
dropna(how='all')	Drop observations where all cells is NA
dropna(axis=1, how='all')	Drop column if all the values are missing
dropna(thresh = 5)	Drop rows that contain less than 5 non-missing values
fillna(0)	Replace missing values with zeros
isnull()	returns True if the value is missing
notnull()	Returns True for non-missing values

Data Frames: Missing values

- When summing the data, missing values will be treated as zero
- If all values are missing, the sum will be equal to NaN
- cumsum() and cumprod() methods ignore missing values but preserve them in the resulting arrays
- Missing values in GroupBy method are excluded
- Many descriptive statistics methods have *skipna* option to control if missing data should be excluded. This value is set to *True* by default

Data Frames: Basic derivative statistics

df.method()	description
describe	Basic statistics (count, mean, std, min, quantiles, max)
min, max	Minimum and maximum values
mean, median, mode	Arithmetic average, median and mode
var, std	Variance and standard deviation
sem	Standard error of mean
skew	Sample skewness
kurt	kurtosis