Exercise Session – Dataframes (exam 07/07/2017)

Danilo Ardagna

Politecnico di Milano

danilo.ardagna@polimi.it

DataFrame

- 2-dimensional labeled data structure with columns of the same type

  - You can think of it like a spreadsheet, a table, or a Python NamedTuple (with the constraint to have column data of the same type!)

- You can access a DataFrame columns by name and apply to every column functions like mean, max, etc.

weather_conditions_dataframe

temperatures	humidity
26.3	0.8
31.4	0.9
25.4	0.8
22.1	0.7

DataFrame

- Implement a DataFrame to store **double**s and provide:
  - A constructor that receives one single string as parameter. Using spaces as word separators, it will initialize the column names with the words contained in the argument
  - A `set_column` method that, given a vector of values of type **double** and a column name, replaces the entire column content
  - The method `set_element_at` that, given a **column name**, an **index i**, and a **value** of type **double**, updates the i-th value of the column
  - `get_mean`, which returns the mean of a given column
  - `select_equal` that, given a column name and a value, returns a new DataFrame including only the set of rows for which the column equals the value
    - For instance, `select_equal ("temperatures", 31.4)` called on the DataFrame of the previous slide would yield a new one with both columns, but only the second row

DataFrame

- Take particular care to error conditions, for example:

  - access to a wrong column or element index out of range

  - check that all the columns have the same number of rows (which will be known only when creating the first column)

- For the constructor implementation you can rely on the split() function:

```
vector<string> split(const string & s, char d)
```