

MSiA 422 – Fall 2019

Project # 2

DUE: 12/10/2019

Project 2 - MyPandas in Java.

In this project, you are going to implement a simple version of Pandas in Java. You can assume that you will only work on the baby name by state dataset which is the same one you worked on in project 1. There are 5 columns: state (String), gender (String), year (Integer), name (String), count (Integer). There is no limitation on the number of rows.

Implement the following functionalities and write at least one test for each method. Write a document showing the output of your tests.

Implement a MyPandas class and a MyDataFrame class. The MyPandas class should offer the following interfaces:

1. MyDataFrame **readCSV**(String path)

Read a file and store it into a MyDataFrame object. Possible data types: Integer and String. You can assume that the first row of the .csv file is a header.

2. void **writeCSV**(MyDataFrame data, String path)

Write a MyDataFrame object to file specified by path.

3. MyDataFrame **concat**(MyDataFrame df1, MyDataFrame df2)

Concatenate two MyDataFrame object along rows. Returns the concatenated MyDataFrame.

MyDataFrame class should offer the following interfaces:

1. Head and Tail.

MyDataFrame **head**(int n)

Returns the first n rows of the data.

MyDataFrame **tail**(int n)

Returns the last n rows of the data.

2. dType

String **dType**(int index)

Returns the type of the column specified by index. If the type is not uniform, return 'String'.

String **dType**(String name)

Returns the type of the column specified by name. If the type is not uniform, return 'String'.

3. Slicing

MyDataFrame **slice**(int index)
Returns the column specified by index.

MyDataFrame **slice**(String name)
Returns the column specified by name.

MyDataFrame **slice**(int[] indexArr)
Returns the columns specified by an index array.

MyDataFrame **slice**(String[] nameArr)
Returns the columns specified by a name array.

4. Filtering

MyDataFrame **filter**(String col, String op, Object o)
Returns data filtered by applying “col op o” on MyDataFrame object, e.g. “count > 10”, “state = ‘IL’”.

5. Indexing

MyDataFrame **loc**(int index)
Returns the rows starting from index.

MyDataFrame **loc**(String label)
Returns the rows starting from label.

MyDataFrame **loc**(int from, int to)
Returns the rows between from and to (including from and to).

MyDataFrame **loc**(String from, String to)
Returns the rows between from and to (including from and to).

6. Sorting

MyDataFrame **sort**(int index)
Returns the data sorted by the column specified by index.

MyDataFrame **sort**(String name)
Returns the data sorted by the column specified by name.

7. Aggregation

Object **getMin**(int index)
Returns the minimum element of the column specified by index.

Object **getMin**(String label)
Returns the minimum element of the column specified by label.

Object **getMax**(int index)
Returns the maximum element of the column specified by index.

Object **getMax**(String label)

Returns the maximum element of the column specified by label.