MSiA 422 Project #2

Kejin Qian kqd3834
Naomi Kaduwela nak133

Test Result

(Due to the console space limitation, instead of using the complete baby names dataset, we chose a sample of it to do the test.)

1. **Test MyPandas readCSV and store the data as MyDataFrame**

   Read test file 1

```
MyPandas pd = new MyPandas();
MyDataFrame df = pd.readCSV("/Users/qiankejin/Desktop/Project2/test.txt");
df.print();
```

   Output:

```
state, gender, year, name, count
-----------------------------------
AK,F,1910,Margaret,8
AK,F,1910,Helen,7
AK,F,1911,Mary,12
AK,F,1911,Annie,6
AK,F,1911,Elizabeth,6
AK,F,1911,Helen,6
AK,F,1912,Mary,9
AK,M,1947,Jack,12
AK,M,1947,Patrick,12
AK,M,1947,Dale,11
FL,F,1910,Annie,101
FL,F,1910,Ethel,82
FL,F,1910,Willie,71
FL,F,1910,Louise,70
FL,F,1910,Lillie,69
FL,F,1910,Ruby,65
FL,F,1910,Ruth,56
LA,F,1910,Mary,586
LA,F,1910,Annie,169
LA,F,1910,Louise,157
LA,F,1910,Marie,152
LA,F,1910,Lillian,116
LA,F,1910,Helen,113
LA,F,1910,Elizabeth,100
LA,F,1910,Rose,90
WI,F,1910,Marie,145
WI,F,1910,Florence,134
WI,F,1910,Esther,121
WI,F,1910,Irene,113
WI,F,1910,Rose,105
NC,M,1920,Ira,46
NC,M,1920,Isaac,46
NC,M,1920,Jimmie,46
NC,M,1920,Tom,46
NC,M,1920,Garland,45
NC,M,1920,Billy,44
DC,F,1910,Thelma,15
DC,F,1910,Viola,15
DC,F,1910,Edna,14
DC,F,1910,Clara,13
-----------------------------------
```

Read test file 2

```
MyDataFrame df2 = pd.readCSV("/Users/qiankejin/Desktop/Project2/test2.txt");
df2.print();
```

Output:

```
state, gender, year, name, count
---------------------------------
IN,M,2009,Garrison,6
IN,M,2009,Gustavo,6
IN,M,2009,Jaedyn,6
IN,M,2007,Willem,5
IN,M,2008,Jacob,538
IN,M,2008,Ethan,522
IN,M,2008,Noah,427
IN,M,2008,William,410
IN,F,1959,Leslie,101
IN,F,1959,Regina,101
IN,F,1959,Alice,96
IN,F,1959,Darla,96
---------------------------------
```

Read an empty txt file

```
MyDataFrame edf = pd.readCSV("/Users/qiankejin/Desktop/Project2/empty.txt");
edf.print();
```

Output:

```
state, gender, year, name, count
---------------------------------
---------------------------------
```

2. **Test MyDataFrame.head(int n)**
   - **Test case 1**

```
df.head(5).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
AK,F,1910,Margaret,8
AK,F,1910,Helen,7
AK,F,1911,Mary,12
AK,F,1911,Annie,6
AK,F,1911,Elizabeth,6
---------------------------------
```

   - **Test case 2**

```
System.out.println(df2.getShape()[0]);
```

Output:


```
df2.head(20).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
IN,M,2009,Garrison,6
IN,M,2009,Gustavo,6
IN,M,2009,Jaedyn,6
IN,M,2007,Willem,5
IN,M,2008,Jacob,538
IN,M,2008,Ethan,522
IN,M,2008,Noah,427
IN,M,2008,William,410
IN,F,1959,Leslie,101
IN,F,1959,Regina,101
IN,F,1959,Alice,96
IN,F,1959,Darla,96
---------------------------------
```

Because df2 only has 12 rows which is smaller than 20, so df2.head(20) can only return all of the 12 rows df2 has.

- **Test case 3 (test on empty txt file)**

```
edf.head(5).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
---------------------------------
```

3. **Test MyDataFrame.tail(int n)**
   - **Test case 1**

```
df.tail(5).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
NC,M,1920,Billy,44
DC,F,1910,Thelma,15
DC,F,1910,Viola,15
DC,F,1910,Edna,14
DC,F,1910,Clara,13
---------------------------------
```

- **Test case 2 (test on empty txt file)**

```
edf.tail(5).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
---------------------------------
```

4. **test MyDataFrame.dtype(int index)**
   - **Test case 1 (test on empty file)**

```
System.out.println(edf.dType(0));
```

Output:
Because edf is an empty txt file, based on assumption, edf.dType will return None

```
None | Dataframe is empty
```

   - **Test case 2**

```
System.out.println(df.dType(0));
```

Output:

```
String
```

   - **Test case 3**

```
System.out.println(df.dType(2));
```

Output:

```
Integer
```

5. **test MyDataFrame.dtype(String name)**
   - **Test case 1**

```
System.out.println(df.dType("state"));
```

Output:

```
String
```

   - **Test case 2**

```
System.out.println(df.dType("year"));
```

Output:

```
Integer
```

6. **test MyDataFrame.slice(int index)**
   - **Test case 1**

```
df.slice(3).head(10).print();
```

Output:

```
name
--------------------------------
Margaret
Helen
Mary
Annie
Elizabeth
Helen
Mary
Jack
Patrick
Dale
--------------------------------
```

7. **test MyDataFrame.slice(String name)**
   - **Test case 1**

```
df.slice("name").head(10).print();
```

Output:

```
name
--------------------------------
Margaret
Helen
Mary
Annie
Elizabeth
Helen
Mary
Jack
Patrick
Dale
--------------------------------
```

8. **test MyDataFrame.slice(int[] indexArr)**
   - **Test case 1**

```
df.slice(new int[]{3, 4}).head(5).print();
```

Output:

```
name, count
--------------------------------
Margaret,8
Helen,7
Mary,12
Annie,6
Elizabeth,6
--------------------------------
```

   - **Test case 2**

```
df.slice(new int[]{1,2,3}).tail(5).print();
```

Output:

```
gender, year, name
---------------------------------
M,1920,Billy
F,1910,Thelma
F,1910,Viola
F,1910,Edna
F,1910,Clara
---------------------------------
```

9. **test MyDataFrame.slice(String[] nameArr)**
   - **Test case 1**

```
df.slice(new String[]{"name", "year"}).head(5).print();
```

Output:

```
name, year
---------------------------------
Margaret,1910
Helen,1910
Mary,1911
Annie,1911
Elizabeth,1911
---------------------------------
```

   - **Test case 2**

```
df.slice(new String[]{"gender", "name", "count"}).tail(5).print();
```

Output:

```
gender, name, count
---------------------------------
M,Billy,44
F,Thelma,15
F,Viola,15
F,Edna,14
F,Clara,13
---------------------------------
```

10. **test MyDataFrame.filter(String col, String op, Object o)**
   - **Test case 1**

```
df.filter("name", "=", "Mary").print();
```

Output:

```
state, gender, year, name, count
---------------------------------
AK,F,1911,Mary,12
AK,F,1912,Mary,9
LA,F,1910,Mary,586
---------------------------------
```

- **Test case 2**

```
df.filter("count", ">=", 10).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
AK,F,1911,Mary,12
AK,M,1947,Jack,12
AK,M,1947,Patrick,12
AK,M,1947,Dale,11
FL,F,1910,Annie,101
FL,F,1910,Ethel,82
FL,F,1910,Willie,71
FL,F,1910,Louise,70
FL,F,1910,Lillie,69
FL,F,1910,Ruby,65
FL,F,1910,Ruth,56
LA,F,1910,Mary,586
LA,F,1910,Annie,169
LA,F,1910,Louise,157
LA,F,1910,Marie,152
LA,F,1910,Lillian,116
LA,F,1910,Helen,113
LA,F,1910,Elizabeth,100
LA,F,1910,Rose,90
WI,F,1910,Marie,145
WI,F,1910,Florence,134
WI,F,1910,Esther,121
WI,F,1910,Irene,113
WI,F,1910,Rose,105
NC,M,1920,Ira,46
NC,M,1920,Isaac,46
NC,M,1920,Jimmie,46
NC,M,1920,Tom,46
NC,M,1920,Garland,45
NC,M,1920,Billy,44
DC,F,1910,Thelma,15
DC,F,1910,Viola,15
DC,F,1910,Edna,14
DC,F,1910,Clara,13
---------------------------------
```

- **Test case 3**

```
df.filter("year", "<", 1944).print();
```

```
state, gender, year, name, count
---------------------------------
AK,F,1910,Margaret,8
AK,F,1910,Helen,7
AK,F,1911,Mary,12
AK,F,1911,Annie,6
AK,F,1911,Elizabeth,6
AK,F,1911,Helen,6
AK,F,1912,Mary,9
FL,F,1910,Annie,101
FL,F,1910,Ethel,82
FL,F,1910,Willie,71
FL,F,1910,Louise,70
FL,F,1910,Lillie,69
FL,F,1910,Ruby,65
FL,F,1910,Ruth,56
LA,F,1910,Mary,586
LA,F,1910,Annie,169
LA,F,1910,Louise,157
LA,F,1910,Marie,152
LA,F,1910,Lillian,116
LA,F,1910,Helen,113
LA,F,1910,Elizabeth,100
LA,F,1910,Rose,90
WI,F,1910,Marie,145
WI,F,1910,Florence,134
WI,F,1910,Esther,121
WI,F,1910,Irene,113
WI,F,1910,Rose,105
NC,M,1920,Ira,46
NC,M,1920,Isaac,46
NC,M,1920,Jimmie,46
NC,M,1920,Tom,46
NC,M,1920,Garland,45
NC,M,1920,Billy,44
DC,F,1910,Thelma,15
DC,F,1910,Viola,15
DC,F,1910,Edna,14
DC,F,1910,Clara,13
---------------------------------
```

Output:

11. **test MyDataFrame.loc(int index)**
    - **Test case 1**

```
df.loc(10).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
FL,F,1910,Annie,101
FL,F,1910,Ethel,82
FL,F,1910,Willie,71
FL,F,1910,Louise,70
FL,F,1910,Lillie,69
FL,F,1910,Ruby,65
FL,F,1910,Ruth,56
LA,F,1910,Mary,586
LA,F,1910,Annie,169
LA,F,1910,Louise,157
LA,F,1910,Marie,152
LA,F,1910,Lillian,116
LA,F,1910,Helen,113
LA,F,1910,Elizabeth,100
LA,F,1910,Rose,90
WI,F,1910,Marie,145
WI,F,1910,Florence,134
WI,F,1910,Esther,121
WI,F,1910,Irene,113
WI,F,1910,Rose,105
NC,M,1920,Ira,46
NC,M,1920,Isaac,46
NC,M,1920,Jimmie,46
NC,M,1920,Tom,46
NC,M,1920,Garland,45
NC,M,1920,Billy,44
DC,F,1910,Thelma,15
DC,F,1910,Viola,15
DC,F,1910,Edna,14
DC,F,1910,Clara,13
---------------------------------
```

12. **test MyDataFrame.loc(int from, int to)**
    - **Test case 1**

```
df.loc(2, 4).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
AK,F,1911,Mary,12
AK,F,1911,Annie,6
AK,F,1911,Elizabeth,6
---------------------------------
```

13. **test MyDataFrame.sort(int index)**
    - **Test case 1**

```
df.sort(4).print();
```

Output:

```
state, gender, year, name, count
--------------------------------
AK,F,1911,Annie,6
AK,F,1911,Elizabeth,6
AK,F,1911,Helen,6
AK,F,1910,Helen,7
AK,F,1910,Margaret,8
AK,F,1912,Mary,9
AK,M,1947,Dale,11
AK,F,1911,Mary,12
AK,M,1947,Jack,12
AK,M,1947,Patrick,12
DC,F,1910,Clara,13
DC,F,1910,Edna,14
DC,F,1910,Thelma,15
DC,F,1910,Viola,15
NC,M,1920,Billy,44
NC,M,1920,Garland,45
NC,M,1920,Ira,46
NC,M,1920,Isaac,46
NC,M,1920,Jimmie,46
NC,M,1920,Tom,46
FL,F,1910,Ruth,56
FL,F,1910,Ruby,65
FL,F,1910,Lillie,69
FL,F,1910,Louise,70
FL,F,1910,Willie,71
FL,F,1910,Ethel,82
LA,F,1910,Rose,90
LA,F,1910,Elizabeth,100
FL,F,1910,Annie,101
WI,F,1910,Rose,105
LA,F,1910,Helen,113
WI,F,1910,Irene,113
LA,F,1910,Lillian,116
WI,F,1910,Esther,121
WI,F,1910,Florence,134
WI,F,1910,Marie,145
LA,F,1910,Marie,152
LA,F,1910,Louise,157
LA,F,1910,Annie,169
LA,F,1910,Mary,586
--------------------------------
```

- **Test case 2**

```
df.sort(2).print();
```

Output:

```
state, gender, year, name, count
---------------------------------
AK,F,1910,Margaret,8
AK,F,1910,Helen,7
FL,F,1910,Annie,101
FL,F,1910,Ethel,82
FL,F,1910,Willie,71
FL,F,1910,Louise,70
FL,F,1910,Lillie,69
FL,F,1910,Ruby,65
FL,F,1910,Ruth,56
LA,F,1910,Mary,586
LA,F,1910,Annie,169
LA,F,1910,Louise,157
LA,F,1910,Marie,152
LA,F,1910,Lillian,116
LA,F,1910,Helen,113
LA,F,1910,Elizabeth,100
LA,F,1910,Rose,90
WI,F,1910,Marie,145
WI,F,1910,Florence,134
WI,F,1910,Esther,121
WI,F,1910,Irene,113
WI,F,1910,Rose,105
DC,F,1910,Thelma,15
DC,F,1910,Viola,15
DC,F,1910,Edna,14
DC,F,1910,Clara,13
AK,F,1911,Mary,12
AK,F,1911,Annie,6
AK,F,1911,Elizabeth,6
AK,F,1911,Helen,6
AK,F,1912,Mary,9
NC,M,1920,Ira,46
NC,M,1920,Isaac,46
NC,M,1920,Jimmie,46
NC,M,1920,Tom,46
NC,M,1920,Garland,45
NC,M,1920,Billy,44
AK,M,1947,Jack,12
AK,M,1947,Patrick,12
AK,M,1947,Dale,11
---------------------------------
```

14. **test MyDataFrame.sort(String name)**

- **Test case 1**

```
df.sort("name").print();
```

Output:

```
state, gender, year, name, count
--------------------------------
AK,F,1911,Annie,6
FL,F,1910,Annie,101
LA,F,1910,Annie,169
NC,M,1920,Billy,44
DC,F,1910,Clara,13
AK,M,1947,Dale,11
DC,F,1910,Edna,14
AK,F,1911,Elizabeth,6
LA,F,1910,Elizabeth,100
WI,F,1910,Esther,121
FL,F,1910,Ethel,82
WI,F,1910,Florence,134
NC,M,1920,Garland,45
AK,F,1910,Helen,7
AK,F,1911,Helen,6
LA,F,1910,Helen,113
NC,M,1920,Ira,46
WI,F,1910,Irene,113
NC,M,1920,Isaac,46
AK,M,1947,Jack,12
NC,M,1920,Jimmie,46
LA,F,1910,Lillian,116
FL,F,1910,Lillie,69
FL,F,1910,Louise,70
LA,F,1910,Louise,157
AK,F,1910,Margaret,8
LA,F,1910,Marie,152
WI,F,1910,Marie,145
AK,F,1911,Mary,12
AK,F,1912,Mary,9
LA,F,1910,Mary,586
AK,M,1947,Patrick,12
LA,F,1910,Rose,90
WI,F,1910,Rose,105
FL,F,1910,Ruby,65
FL,F,1910,Ruth,56
DC,F,1910,Thelma,15
NC,M,1920,Tom,46
DC,F,1910,Viola,15
FL,F,1910,Willie,71
--------------------------------
```

15. **test MyDataFrame.getMin(int index)**

- **Test case 1**

```
System.out.println(df.getMin(2));
```

Output:

```
1910
```

16. **test MyDataFrame.getMin(String name)**

- **Test case 1**

```
System.out.println(df.getMin("name"));
```

Output:

```
Annie
```

- **Test case 2**

```
System.out.println(df.getMin("count"));
```

Output:

```
6
```

17. **test MyDataFrame.getMax(int index)**
    - **Test case 1**

```
System.out.println(df.getMax(2));
```

Output:

```
1947
```

18. **test MyDataFrame.getMax(String name)**
    - **Test case 1**

```
System.out.println(df.getMax("name"));
```

Output:

```
Willie
```

    - **Test case 2**

```
System.out.println(df.getMax("count"));
```

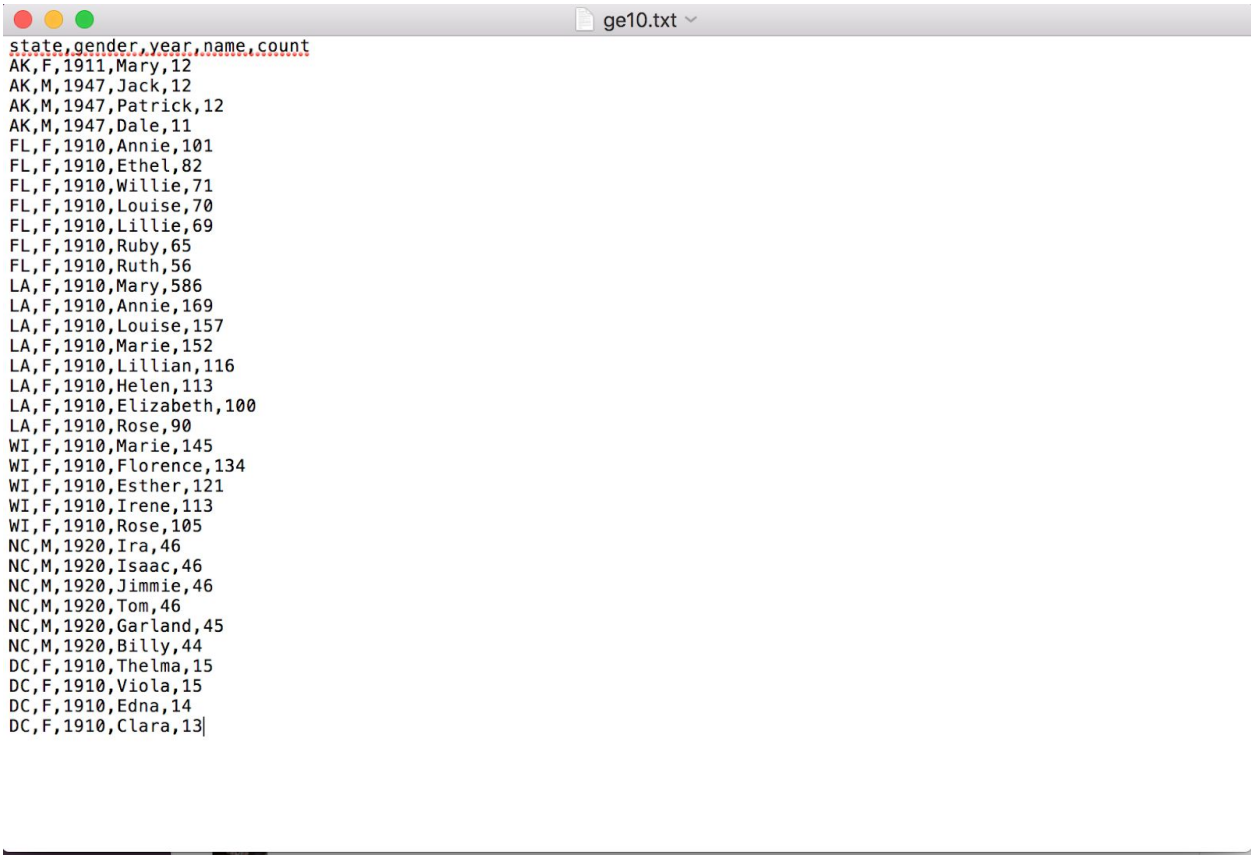
Output:

```
586
```

19. **test MyPandas.writeCSV(MyDataFrame, path)**
    - **Test case 1**

```
pd.writeCSV(df.filter("count", ">=", 10), "ge10.txt");
```

Output:
This is a screenshot of the ge10.txt file



20. **test MyPandas.concat(MyDataFrame df1, MyDataFrame df2))**

```
pd.concat(df, df2).print();
pd.writeCSV(pd.concat(df, df2), "con1.txt");
```

Output:

```
state, gender, year, name, count
---------------------------------
AK,F,1910,Margaret,8
AK,F,1910,Helen,7
AK,F,1911,Mary,12
AK,F,1911,Annie,6
AK,F,1911,Elizabeth,6
AK,F,1911,Helen,6
AK,F,1912,Mary,9
AK,M,1947,Jack,12
AK,M,1947,Patrick,12
AK,M,1947,Dale,11
FL,F,1910,Annie,101
FL,F,1910,Ethel,82
FL,F,1910,Willie,71
FL,F,1910,Louise,70
FL,F,1910,Lillie,69
FL,F,1910,Ruby,65
FL,F,1910,Ruth,56
LA,F,1910,Mary,586
LA,F,1910,Annie,169
LA,F,1910,Louise,157
LA,F,1910,Marie,152
LA,F,1910,Lillian,116
LA,F,1910,Helen,113
LA,F,1910,Elizabeth,100
LA,F,1910,Rose,90
WI,F,1910,Marie,145
WI,F,1910,Florence,134
WI,F,1910,Esther,121
WI,F,1910,Irene,113
WI,F,1910,Rose,105
NC,M,1920,Ira,46
NC,M,1920,Isaac,46
NC,M,1920,Jimmie,46
NC,M,1920,Tom,46
NC,M,1920,Garland,45
NC,M,1920,Billy,44
DC,F,1910,Thelma,15
DC,F,1910,Viola,15
DC,F,1910,Edna,14
DC,F,1910,Clara,13
IN,M,2009,Garrison,6
IN,M,2009,Gustavo,6
IN,M,2009,Jaedyn,6
IN,M,2007,Willem,5
IN,M,2008,Jacob,538
IN,M,2008,Ethan,522
IN,M,2008,Noah,427
IN,M,2008,William,410
IN,F,1959,Leslie,101
IN,F,1959,Regina,101
IN,F,1959,Alice,96
IN,F,1959,Darla,96
---------------------------------
```

Screenshot of con1.txt