CSC380: Principles of Data Science

Brief Introduction to Pandas

Prof. Jason Pacheco

TA: Enfa Rose George

TA: Saiful Islam Salim

Administrative Items

- Homework 4 : Due this **Sunday** (10/3) @ 11:59pm
  - If you can get it in earlier plead do so TAs can start grading
  - Small amount of extra credit to HWs submitted tonight
- Homework 5 : Out tomorrow (Due: **Tuesday** 10/12)
- Midterm : Out **Thursday** 10/14, Due **Tuesday** 10/19

Pandas

Open source library for data handling and manipulation in high-performance environments.

Installation If you are using Anaconda package manager,

```
conda install pandas
```

Or if you are using PyPi (pip) package manager,

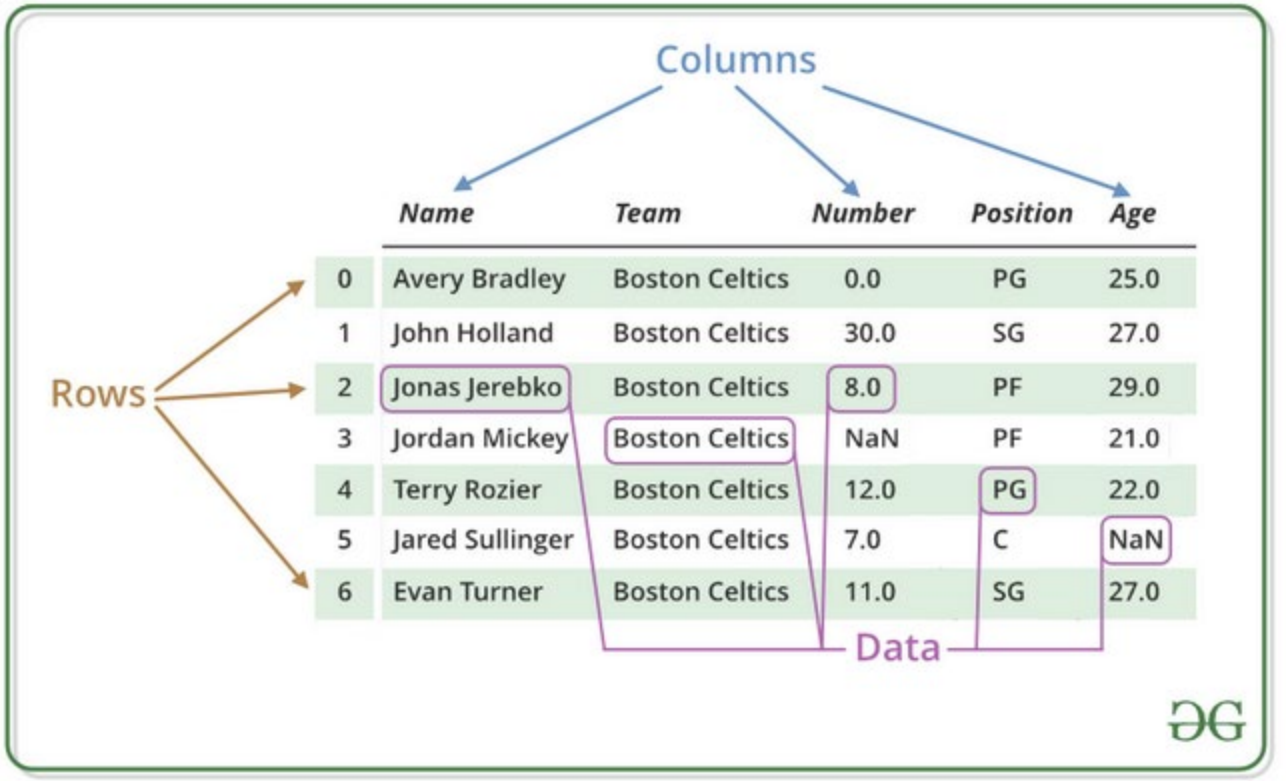
```
pip install pandas
```

See Pandas documentation for more detailed instructions

https://pandas.pydata.org/docs/getting_started/install.html

DataFrame

Primary data structure : Essentially a table



[Source: https://www.geeksforgeeks.org/python-pandas-dataframe/]

DataFrame Example

Create and print an entire DataFrame

```
# import pandas as pd
import pandas as pd

# list of strings
lst = ['Geeks', 'For', 'Geeks', 'is',
            'portal', 'for', 'Geeks']

# Calling DataFrame constructor on list
df = pd.DataFrame(lst)
print(df)
```

	0
0	Geeks
1	For
2	Geeks
3	is
4	portal
5	for
6	Geeks

[Source: https://www.geeksforgeeks.org/python-pandas-dataframe/]

DataFrame Example

Can create named columns using dictionary

```
import pandas as pd

# intialise data of lists.
data = {'Name':['Tom', 'nick', 'krish', 'jack'],
        'Age':[20, 21, 19, 18]}

# Create DataFrame
df = pd.DataFrame(data)

# Print the output.
print(df)
```

	Name	Age
0	Tom	20
1	nick	21
2	krish	19
3	jack	18

[Source: https://www.geeksforgeeks.org/python-pandas-dataframe/]

DataFrame : Selecting Columns

Select columns to print by name,

```
# Import pandas package
import pandas as pd

# Define a dictionary containing employee data
data = {'Name':['Jai', 'Princi', 'Gaurav', 'Anuj'],
        'Age':[27, 24, 22, 32],
        'Address':['Delhi', 'Kanpur', 'Allahabad', 'Kannauj'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}

# Convert the dictionary into DataFrame
df = pd.DataFrame(data)

# select two columns
print(df[['Name', 'Qualification']])
```

	Name	Qualification
0	Jai	Msc
1	Princi	MA
2	Gaurav	MCA
3	Anuj	Phd

[Source: https://www.geeksforgeeks.org/python-pandas-dataframe/]

DataFrame : Selecting Rows

Select columns to print by name,

```
import pandas as pd
import numpy as np

# Define a dictionary containing employee data
data = {'Name':['Jai', 'Princi', 'Gaurav', 'Anuj'],
        'Age':[27, 24, 22, 32],
        'Address':['Delhi', 'Kanpur', 'Allahabad', 'Kannauj'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}

# Convert the dictionary into DataFrame
df = pd.DataFrame(data)

# Print rows 1 & 2
row = df.loc[1:2]
print(row)
```

Output

	Name	Age	Address	Qualification
1	Princi	24	Kanpur	MA
2	Gaurav	22	Allahabad	MCA

DataFrame : Selecting Rows

`head()` and `tail()` select rows from beginning / end

```
import pandas as pd
import numpy as np

# Define a dictionary containing employee data
data = {'Name':['Jai', 'Princi', 'Gaurav', 'Anuj'],
        'Age':[27, 24, 22, 32],
        'Address':['Delhi', 'Kanpur', 'Allahabad', 'Kannauj'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}

# Convert the dictionary into DataFrame
df = pd.DataFrame(data)

# Print first / last rows
first2 = df.head(2)
last2 = df.tail(2)
print(first2)
print('\n', last2)
```

Output

```
     Name  Age Address Qualification
0     Jai   27   Delhi           Msc
1  Princi   24  Kanpur            MA

     Name  Age    Address Qualification
2  Gaurav   22  Allahabad           MCA
3    Anuj   32    Kannauj           Phd
```

Reading Data from Files

Easy reading / writing of standard formats,

```
df = pd.read_json("data.json")
print(df)
df.to_csv("data.csv", index=False)
df_csv = pd.read_csv("data.csv")
print(df_csv.head(2))
```

Output

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.4
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]
   Duration  Pulse  Maxpulse  Calories
0        60    110       130     409.1
1        60    117       145     479.0
```

Data Structure Conversions

Working with DataFrames outside of Pandas can be tricky,

```
df['Duration']
```

```
0      60
1      60
2      60
3      45
4      45
      ..
164    60
165    60
166    60
167    75
168    75
Name: Duration, Length: 169, dtype: int64
```

We can easily convert to built-in types, for example to a list (e.g. to use in Numpy or whatever),

```
L = df['Duration'].to_list()
print(L)
```

```
[60, 60, 60, 45, 45, 60, 60, 45, 30, 60, 60, 60, 60, 60, 60, 60, 60, 45, 60, 45, 60, 45, 60, 45, 60, 60, 60, 60, 60,
60, 60, 45, 60, 60, 60, 60, 60, 60, 60, 45, 45, 60, 60, 60, 60, 60, 60, 45, 45, 60, 60, 80, 60, 60, 30, 60, 60, 45, 2
0, 45, 210, 160, 160, 45, 20, 180, 150, 150, 20, 300, 150, 60, 90, 150, 45, 90, 45, 45, 120, 270, 30, 45, 30, 120, 4
5, 30, 45, 120, 45, 20, 180, 45, 30, 15, 20, 20, 30, 25, 30, 90, 20, 90, 90, 90, 30, 30, 180, 30, 90, 210, 60, 45, 1
5, 45, 60, 60, 60, 60, 60, 60, 30, 45, 60, 60, 60, 60, 60, 60, 90, 60, 60, 60, 60, 60, 60, 20, 45, 45, 45, 20, 60, 6
0, 45, 45, 60, 45, 60, 60, 30, 60, 60, 60, 60, 30, 60, 60, 60, 60, 60, 30, 30, 45, 45, 45, 60, 60, 60, 75, 75]
```

Summary Statistics

Easily compute summary statistics on data

```
print('Min: ', df['Duration'].min())
print('Max: ', df['Duration'].max())
print('Median: ', df['Duration'].median())
```

```
Min:  15
Max:  300
Median:  60.0
```

Can also count occurrences of unique values,

```
df['Duration'].value_counts()
```

```
60     79
45     35
30     16
20      9
90      8
150     4
120     3
180     3
15      2
75      2
160     2
210     2
270     1
25      1
300     1
80      1
Name: Duration, dtype: int64
```