Data Input and Output

Pandas can read a variety of file types using its pd.read_ methods.

```
import numpy as np
import pandas as pd
```

CSV

CSV Input

```
# df = pd.read_csv('Sample1.csv')
df = pd.read_csv('C:/Users/Shyamala/Desktop/SDP - ML & IP/python basics/Sample1.csv')
print(df)
print(df.size)
print(df.shape)
```

```
  Unnamed: 0   a   b   c   d
0          P   0   1   2   3
1          Q   4   5   6   7
2          R   8   9  10  11
3          S  12  13  14  15
20
(4, 5)
```

CSV Output

```
df.to_csv('test1.csv',index=False)
df
```

	Unnamed: 0	a	b	c	d
0	P	0	1	2	3
1	Q	4	5	6	7
2	R	8	9	10	11
3	S	12	13	14	15

Excel

Pandas can read and write excel files, keep in mind, this only imports data. Not formulas or images, having images or macros may cause this read_excel method to crash.

Excel Input

```
df1=pd.read_excel('Sample2.xlsx')
# df1.size
# df1.shape
df1
```

Excel Output

```
df1.to_excel('test2.xlsx', index=False)
```

```
One interesting thing is the use of Pandas for conversion.
So, maybe we are inputting data from a CSV, but we really
want to display that data to HTML on your website.
Since HTML is one of the datatypes, we can just export to HTML.
```

```
df1.to_html('test3.html')
```

HTML

Pandas can read table tabs off of html. For example:

HTML Input

Pandas read_html function will read tables off of a webpage and return a list of DataFrame objects:

```
df = pd.read_html('https://www.fdic.gov/resources/resolutions/bank-failures/failed-bank-list/
df
```

```
[                         Bank NameBank           CityCity StateSt  CertCert  \
 0                    Almena State Bank             Almena      KS     15426   
 1           First City Bank of Florida  Fort Walton Beach      FL     16748   
 2                 The First State Bank      Barboursville      WV     14361   
 3                   Ericson State Bank            Ericson      NE     18265   
 4     City National Bank of New Jersey             Newark      NJ     21111   
 ..                                 ...                ...     ...       ...   
 558                 Superior Bank, FSB           Hinsdale      IL     32646   
 559                Malta National Bank              Malta      OH      6629   
560    First Alliance Bank & Trust Co.         Manchester      NH     34264
561  National State Bank of Metropolis         Metropolis      IL      3815
562                   Bank of Honolulu           Honolulu      HI     21029

                 Acquiring InstitutionAI Closing DateClosing  FundFund
0                            Equity Bank    October 23, 2020     10538
1              United Fidelity Bank, fsb    October 16, 2020     10537
2                         MVB Bank, Inc.       April 3, 2020     10536
3             Farmers and Merchants Bank   February 14, 2020     10535
4                        Industrial Bank    November 1, 2019     10534
..                                   ...                 ...       ...
558                Superior Federal, FSB       July 27, 2001      6004
559                    North Valley Bank         May 3, 2001      4648
560  Southern New Hampshire Bank & Trust    February 2, 2001      4647
561              Banterra Bank of Marion   December 14, 2000      4646
562                   Bank of the Orient    October 13, 2000      4645

[563 rows x 7 columns]]
```

Let us now read a most popular dataset named "iris"

```
df = pd.read_csv('iris.csv')
#df = pd.read_csv('D:/shyam/Users/Welcome/Desktop/SDP - IP & ML using python/Day 2 - Data Ana
df
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

```
# df.head()
df.tail()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal.length  150 non-null    float64
 1   sepal.width   150 non-null    float64
 2   petal.length  150 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df.shape
```

```
(150, 5)
```

```
df['sepal.length']
```

```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ... 
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: sepal.length, Length: 150, dtype: float64
```

```
df['sepal.length'][149]
```

```
5.9
```

```
df.isnull()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

```
df['sepal.width'].isnull()
```

```
df.iloc[0:3]
```

```
df.loc[[0,1],['sepal.length','petal.length']]
# df.loc[[0],['sepal.length']]
```

Let us now read "iris1" that has some null values in it

```
df1 = pd.read_csv('iris1.csv')
df1
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal.length  147 non-null    float64
 1   sepal.width   148 non-null    float64
 2   petal.length  148 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df=df.fillna(value=df.mean())
```

```
df.info()
```