

```
In [2]: import pandas as pd
```

```
In [3]: df = pd.DataFrame()
```

```
In [4]: Age = [23,3,4,5,7,8,6,5]
Income = [456,657,665,889,665,432,567,9087]
Height = [3.5,5.75,5.5,6.0,4.76,6.5,5.8,6.0]
```

```
#Adding data to our dataframe
```

```
df['Age'] = Age
df['Income'] = Income
df['Height'] = Height
```

```
In [5]: df
```

```
Out[5]:
```

	Age	Income	Height
0	23	456	3.50
1	3	657	5.75
2	4	665	5.50
3	5	889	6.00
4	7	665	4.76
5	8	432	6.50
6	6	567	5.80
7	5	9087	6.00

```
In [6]: df.index = ['Ma', 'Ay', 'Rt', 'As', 'Yk', 'Kb', 'Qy', 'Fm']
```

```
In [7]: df
```

```
Out[7]:
```

	Age	Income	Height
Ma	23	456	3.50
Ay	3	657	5.75
Rt	4	665	5.50
As	5	889	6.00
Yk	7	665	4.76
Kb	8	432	6.50
Qy	6	567	5.80
Fm	5	9087	6.00

```
In [18]: import pandas as pd
```

```
In [19]: fileName = 'Book1.geo.csv'
filePath = 'C:/Users/user/.jupyter' # change all backwards slashes to a forward slash in the filepath
fullFilePath = 'C:/Users/user/.jupyter/Book1.geo.csv'
```

```
In [17]: dataFile =pd.read_csv(filePath + '/' +fileName)
```

```
In [20]: filePath + '/' +fileName
```

```
Out[20]: 'C:/Users/user/.jupyter/Book1.geo.csv'
```

```
In [21]: "C:/Users/user/.jupyter/Book1.geo.csv"
```

```
Out[21]: 'C:/Users/user/.jupyter/Book1.geo.csv'
```

```
In [22]: dataFile =pd.read_csv(filePath + '/' +fileName)
```

```
In [23]: dataFile1 =pd.read_csv(fullFilePath)
```

```
In [24]: dataFile
```

```
Out[24]:
```

	latitude	logitude	address	alk	calcium	choles	sodium	tbilirub	albumin	resu
0	7.443262	3.947426	ojoo/ajobobeilaniba	20	9.65	247	1.00	0.20	0.20	
1	7.384823	3.871384	nigeria railway corporation quarters	22	9.00	245	2.00	0.20	0.20	
2	7.363055	3.865576	new gra	25	9.50	243	1.00	0.40	0.30	
3	7.388649	3.948576	olubadan estate	27	9.40	267	0.90	0.90	0.14	
4	7.385559	3.971009	egeda	35	9.30	258	0.80	1.00	0.40	
5	7.414209	3.955686	lagelu market/kajola	32	9.76	240	2.00	1.00	0.23	
6	7.407599	3.853924	idi ishin jericho reserve	48	9.65	234	2.00	0.60	0.50	
7	7.330916	3.872143	ibadan south west	22	5.45	230	1.50	0.67	0.23	
8	7.366615	3.837482	oluyole	28	4.55	256	1.00	0.50	0.20	
9	7.409418	3.947868	iwo road	36	5.55	200	2.00	0.65	0.28	
10	7.354201	3.944512	ogbere tiayo	45	6.45	255	1.00	0.50	0.30	
11	7.367516	3.899890	Old quarter	46	7.40	247	0.90	0.40	0.35	
12	7.410047	3.865699	eleyele	32	8.20	263	0.80	0.90	0.50	
13	7.522760	3.911803	oyo-emo/moniya	35	6.65	240	0.67	1.00	0.20	
14	7.336162	3.920764	ibadan south east	37	7.45	239	1.50	1.00	0.28	

```
In [25]: columns in our data are referred to as data features
rows in our data are referred to as samples
```

In [26]: dataFile.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   latitude    15 non-null     float64
1   logitude    15 non-null     float64
2   address     15 non-null     object
3   alk         15 non-null     int64
4   calcium     15 non-null     float64
5   choles      15 non-null     int64
6   sodium      15 non-null     float64
7   tbilirub    15 non-null     float64
8   albumin     15 non-null     float64
9   result      15 non-null     int64
dtypes: float64(6), int64(3), object(1)
memory usage: 1.3+ KB
```

In [27]: *# Observations:abs*

1. There are a number of features **in** the data where some cells do **not** have **any** values **in** them.
 This **is** referred **as** Missing data ()
 3. Our data **is** of shape; 15 samples by 10 features

In [28]: dataFile.describe() *#This function provides details analysis of the dispersion and distribution behavior in our data*

Out[28]:

	latitude	logitude	alk	calcium	choles	sodium	tbilirub	albumin
count	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000
mean	7.392319	3.907583	32.666667	7.864000	244.266667	1.271333	0.661333	0.287333
std	0.047475	0.043688	8.893550	1.782858	16.166397	0.507415	0.291667	0.108987
min	7.330916	3.837482	20.000000	4.550000	200.000000	0.670000	0.200000	0.140000
25%	7.364835	3.868541	26.000000	6.550000	239.500000	0.900000	0.450000	0.200000
50%	7.385559	3.911803	32.000000	8.200000	245.000000	1.000000	0.650000	0.280000
75%	7.409733	3.947647	36.500000	9.450000	255.500000	1.750000	0.950000	0.325000
max	7.522760	3.971009	48.000000	9.760000	267.000000	2.000000	1.000000	0.500000



In []: