

Exploring Pandas DataFrame

```
In [8]: import pandas as pd
```

Data Loading

```
In [21]: file_path = '../..data/iris.data'  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
```

```
In [22]: # df = pd.read_csv(file_path)
# df = pd.read_csv(file_path, delimiter=',')
# df = pd.read_csv(file_path, header=None, delimiter=',')

column_names = ['sepal length',
                 'sepal width',
                 'petal length',
                 'petal width',
                 'class label']

df = pd.read_csv(url, names=column_names)
df
```

```
Out[22]:
```

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

150 rows × 5 columns

```
In [23]: df.head()
```

```
Out[23]:
```

	sepal length	sepal width	petal length	petal width	class label
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
In [24]: df.head(10)
```

```
Out[24]:
```

	sepal length	sepal width	petal length	petal width	class label
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa

```
In [25]: df.tail()
```

```
Out[25]:
```

	sepal length	sepal width	petal length	petal width	class label
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

```
In [26]: df.tail(10)
```

```
Out[26]:
```

	sepal length	sepal width	petal length	petal width	class label
140	6.7	3.1	5.6	2.4	Iris-virginica
141	6.9	3.1	5.1	2.3	Iris-virginica
142	5.8	2.7	5.1	1.9	Iris-virginica
143	6.8	3.2	5.9	2.3	Iris-virginica
144	6.7	3.3	5.7	2.5	Iris-virginica
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

```
In [28]: df.sample()
```

```
Out[28]:
```

	sepal length	sepal width	petal length	petal width	class label
47	4.6	3.2	1.4	0.2	Iris-setosa

```
In [29]: df.sample(10)
```

```
Out[29]:
```

	sepal length	sepal width	petal length	petal width	class label
146	6.3	2.5	5.0	1.9	Iris-virginica
0	5.1	3.5	1.4	0.2	Iris-setosa
89	5.5	2.5	4.0	1.3	Iris-versicolor
84	5.4	3.0	4.5	1.5	Iris-versicolor
91	6.1	3.0	4.6	1.4	Iris-versicolor
33	5.5	4.2	1.4	0.2	Iris-setosa
136	6.3	3.4	5.6	2.4	Iris-virginica
19	5.1	3.8	1.5	0.3	Iris-setosa
140	6.7	3.1	5.6	2.4	Iris-virginica
30	4.8	3.1	1.6	0.2	Iris-setosa

```
In [34]: df.shape
```

```
Out[34]: (150, 5)
```

```
In [32]: df.size
```

```
Out[32]: 750
```

```
In [33]: df.describe()
```

```
Out[33]:
```

	sepal length	sepal width	petal length	petal width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Titanic dataset

```
In [35]: file_path = '../..data/titanic.csv'
```

```
In [37]: df = pd.read_csv(file_path)
df
```

Out[37]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.00	female	1	1
1	Allison, Miss Helen Loraine	1st	2.00	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	female	0	1
4	Allison, Master Hudson Trevor	1st	0.92	male	1	0
...
1308	Zakarian, Mr Artun	3rd	27.00	male	0	0
1309	Zakarian, Mr Maprieder	3rd	26.00	male	0	0
1310	Zenni, Mr Philip	3rd	22.00	male	0	0
1311	Lievens, Mr Rene	3rd	24.00	male	0	0
1312	Zimmerman, Leo	3rd	29.00	male	0	0

1313 rows × 6 columns

```
In [38]: df.isnull()
```

```
Out[38]:
```

	Name	PClass	Age	Sex	Survived	SexCode
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...
1308	False	False	False	False	False	False
1309	False	False	False	False	False	False
1310	False	False	False	False	False	False
1311	False	False	False	False	False	False
1312	False	False	False	False	False	False

1313 rows × 6 columns

```
In [39]: df.isnull().sum()
```

```
Out[39]: Name      0
PClass    0
Age       557
Sex       0
Survived  0
SexCode   0
dtype: int64
```

```
In [43]: df[df['Age'].isnull()]
```

```
Out[43]:
```

	Name	PClass	Age	Sex	Survived	SexCode
12	Aubert, Mrs Leontine Pauline	1st	NaN	female	1	1
13	Barkworth, Mr Algernon H	1st	NaN	male	1	0
14	Baumann, Mr John D	1st	NaN	male	0	0
29	Borebank, Mr John James	1st	NaN	male	0	0
32	Bradley, Mr George	1st	NaN	male	1	0
...
1300	Wiseman, Mr Phillippe	3rd	NaN	male	0	0
1302	Yalsevac, Mr Ivan	3rd	NaN	male	1	0
1305	Youssef, Mr Gerios	3rd	NaN	male	0	0
1306	Zabour, Miss Hileni	3rd	NaN	female	0	1
1307	Zabour, Miss Tamini	3rd	NaN	female	0	1

557 rows × 6 columns


```
In [44]: df[df['Age'].isnull()].head(30)
```

```
Out[44]:
```

	Name	PClass	Age	Sex	Survived	SexCode
12	Aubert, Mrs Leontine Pauline	1st	NaN	female	1	1
13	Barkworth, Mr Algernon H	1st	NaN	male	1	0
14	Baumann, Mr John D	1st	NaN	male	0	0
29	Borebank, Mr John James	1st	NaN	male	0	0
32	Bradley, Mr George	1st	NaN	male	1	0
35	Brewe, Dr Arthur Jackson	1st	NaN	male	0	0
40	Calderhead, Mr Edward P	1st	NaN	male	1	0
45	Carrau, Mr Francisco M	1st	NaN	male	0	0
46	Carrau, Mr Jose Pedro	1st	NaN	male	0	0
52	Cassebeer, Mrs Henry Arthur jr (Genevieve Fosd...	1st	NaN	female	1	1
54	Cavendish, Mrs Tyrell William Julia Florence S...	1st	NaN	female	1	1
59	Cherry, Miss Gladys	1st	NaN	female	1	1
60	Chevre, Mr Paul	1st	NaN	male	1	0
61	Chibnall (Bowerman), Mrs Edith Martha	1st	NaN	female	1	1
62	Chisholm, Mr Roderick Robert	1st	NaN	male	0	0
65	Clifford, Mr George Quincy	1st	NaN	male	0	0
66	Colley, Mr Edward Pomeroy	1st	NaN	male	0	0
71	Crafton, Mr John Bertram	1st	NaN	male	0	0
77	Daly, Mr Peter Denis	1st	NaN	male	1	0
81	de Villiers, Madame Berthe	1st	NaN	female	1	1
84	Dodge, Dr Washington	1st	NaN	male	1	0
85	Dodge, Mrs Washington (Ruth Vidaver)	1st	NaN	female	1	1
96	Flegenheim, Mrs Alfred (Antoinette)	1st	NaN	female	1	1
97	Flynn, Mr John Irving	1st	NaN	male	1	0
105	Franklin, Mr Thomas Parham	1st	NaN	male	0	0
107	Frauenthal, Mrs Henry William (Clara Heinsheimer)	1st	NaN	female	1	1
118	Goldenberg, Mrs Samuel L (Edwiga Grabowsko)	1st	NaN	female	1	1
133	Hawksford, Mr Walter James	1st	NaN	male	1	0
137	Head, Mr Christopher	1st	NaN	male	0	0

	Name	PClass	Age	Sex	Survived	SexCode
138	Hilliard, Mr Herbert Henry	1st	NaN	male	0	0

```
In [45]: df_clean = df.dropna()
df_clean
```

```
Out[45]:
```

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.00	female	1	1
1	Allison, Miss Helen Loraine	1st	2.00	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	female	0	1
4	Allison, Master Hudson Trevor	1st	0.92	male	1	0
...
1308	Zakarian, Mr Artun	3rd	27.00	male	0	0
1309	Zakarian, Mr Maprieder	3rd	26.00	male	0	0
1310	Zenni, Mr Philip	3rd	22.00	male	0	0
1311	Lievens, Mr Rene	3rd	24.00	male	0	0
1312	Zimmerman, Leo	3rd	29.00	male	0	0

756 rows × 6 columns

```
In [48]: df_clean.replace("female", "Woman")
```

Out[48]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.00	Woman	1	1
1	Allison, Miss Helen Loraine	1st	2.00	Woman	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	Woman	0	1
4	Allison, Master Hudson Trevor	1st	0.92	male	1	0
...
1308	Zakarian, Mr Artun	3rd	27.00	male	0	0
1309	Zakarian, Mr Maprieder	3rd	26.00	male	0	0
1310	Zenni, Mr Philip	3rd	22.00	male	0	0
1311	Lievens, Mr Rene	3rd	24.00	male	0	0
1312	Zimmerman, Leo	3rd	29.00	male	0	0

756 rows × 6 columns

```
In [49]: df_clean.replace(1, "One")
```

Out[49]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.0	female	One	One
1	Allison, Miss Helen Loraine	1st	2.0	female	0	One
2	Allison, Mr Hudson Joshua Creighton	1st	30.0	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.0	female	0	One
4	Allison, Master Hudson Trevor	1st	0.92	male	One	0
...
1308	Zakarian, Mr Artun	3rd	27.0	male	0	0
1309	Zakarian, Mr Maprieder	3rd	26.0	male	0	0
1310	Zenni, Mr Philip	3rd	22.0	male	0	0
1311	Lievens, Mr Rene	3rd	24.0	male	0	0
1312	Zimmerman, Leo	3rd	29.0	male	0	0

756 rows × 6 columns

```
In [51]: df1 = df_clean.replace(["female", "male"], ["Woman", "Man"])
df1
```

```
Out[51]:
```

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.00	Woman	1	1
1	Allison, Miss Helen Loraine	1st	2.00	Woman	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	Man	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	Woman	0	1
4	Allison, Master Hudson Trevor	1st	0.92	Man	1	0
...
1308	Zakarian, Mr Artun	3rd	27.00	Man	0	0
1309	Zakarian, Mr Maprieder	3rd	26.00	Man	0	0
1310	Zenni, Mr Philip	3rd	22.00	Man	0	0
1311	Lievens, Mr Rene	3rd	24.00	Man	0	0
1312	Zimmerman, Leo	3rd	29.00	Man	0	0

756 rows × 6 columns

```
In [52]: df1['SexCode'].replace([0, 1], ["Zero", "One"])
```

```
Out[52]: 0      One
1      One
2      Zero
3      One
4      Zero
...
1308    Zero
1309    Zero
1310    Zero
1311    Zero
1312    Zero
Name: SexCode, Length: 756, dtype: object
```

```
In [53]: df1['SexCode'] = df1['SexCode'].replace([0, 1], ["Zero", "One"])
df1
```

```
Out[53]:
```

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.00	Woman	1	One
1	Allison, Miss Helen Loraine	1st	2.00	Woman	0	One
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	Man	0	Zero
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	Woman	0	One
4	Allison, Master Hudson Trevor	1st	0.92	Man	1	Zero
...
1308	Zakarian, Mr Artun	3rd	27.00	Man	0	Zero
1309	Zakarian, Mr Maprieder	3rd	26.00	Man	0	Zero
1310	Zenni, Mr Philip	3rd	22.00	Man	0	Zero
1311	Lievens, Mr Rene	3rd	24.00	Man	0	Zero
1312	Zimmerman, Leo	3rd	29.00	Man	0	Zero

756 rows × 6 columns

```
In [54]: df1['Age'].mean()
```

```
Out[54]: 30.397989417989418
```

Basic Statistics

```
In [55]: file_path = '../..data/world_population.csv'
```

```
In [57]: df = pd.read_csv(file_path, skiprows=4)
df
```

```
Out[57]:
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2012	2013
0	Aruba	ABW	Population, total	SP.POP.TOTL	54208.0	55434.0	56234.0	56699.0	57029.0	57357.0	...	102565.0	103160.0
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130836765.0	134159786.0	137614644.0	141202036.0	144920186.0	148769974.0	...	547482863.0	562601570.0
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8996967.0	9169406.0	9351442.0	9543200.0	9744772.0	9956318.0	...	31161378.0	32269590.0
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	96396419.0	98407221.0	100506960.0	102691339.0	104953470.0	107289875.0	...	370243017.0	380437890.0
4	Angola	AGO	Population, total	SP.POP.TOTL	5454938.0	5531451.0	5608499.0	5679409.0	5734995.0	5770573.0	...	25107925.0	26015780.0
...
261	Kosovo	XKX	Population, total	SP.POP.TOTL	947000.0	966000.0	994000.0	1022000.0	1050000.0	1078000.0	...	1807106.0	1818110.0
262	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	5315351.0	5393034.0	5473671.0	5556767.0	5641598.0	5727745.0	...	24473176.0	25147110.0
263	South Africa	ZAF	Population, total	SP.POP.TOTL	17099836.0	17524533.0	17965733.0	18423157.0	18896303.0	19384838.0	...	52832659.0	53687120.0
264	Zambia	ZMB	Population, total	SP.POP.TOTL	3070780.0	3164330.0	3260645.0	3360099.0	3463211.0	3570466.0	...	14465148.0	14926550.0
265	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	3776679.0	3905038.0	4039209.0	4178726.0	4322854.0	4471178.0	...	13115149.0	13350370.0

266 rows × 66 columns



```
In [58]: df.shape
```

```
Out[58]: (266, 66)
```

```
In [59]: df.size
```

```
Out[59]: 17556
```

```
In [60]: df.describe()
```

```
Out[60]:
```

	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...
count	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	...
mean	1.173116e+08	1.188586e+08	1.209679e+08	1.235882e+08	1.262297e+08	1.289366e+08	1.317868e+08	1.346309e+08	1.375436e+08	1.405603e+08	...
std	3.705555e+08	3.749081e+08	3.814322e+08	3.899277e+08	3.984666e+08	4.072394e+08	4.165735e+08	4.258523e+08	4.353820e+08	4.452976e+08	...
min	2.833000e+03	3.077000e+03	3.367000e+03	3.703000e+03	4.063000e+03	4.460000e+03	4.675000e+03	4.922000e+03	5.194000e+03	5.461000e+03	...
25%	5.022802e+05	5.109642e+05	5.206540e+05	5.311622e+05	5.421252e+05	5.533362e+05	5.647475e+05	5.823645e+05	5.981078e+05	6.100030e+05	...
50%	3.718330e+06	3.826398e+06	3.929109e+06	4.015834e+06	4.124521e+06	4.242788e+06	4.326013e+06	4.387887e+06	4.474171e+06	4.550402e+06	...
75%	2.636053e+07	2.721235e+07	2.808607e+07	2.890669e+07	2.972333e+07	3.055227e+07	3.134845e+07	3.200449e+07	3.244145e+07	3.277149e+07	...
max	3.032156e+09	3.071596e+09	3.124561e+09	3.189656e+09	3.255146e+09	3.322047e+09	3.392098e+09	3.461620e+09	3.532783e+09	3.606554e+09	...

8 rows × 62 columns



```
In [68]: temp_df = df['1960']  
type(df), type(temp_df), temp_df
```

```
Out[68]: (pandas.core.frame.DataFrame,  
pandas.core.series.Series,  
0      54208.0  
1    130836765.0  
2     8996967.0  
3    96396419.0  
4     5454938.0  
...  
261    947000.0  
262    5315351.0  
263   17099836.0  
264    3070780.0  
265    3776679.0  
Name: 1960, Length: 266, dtype: float64)
```

```
In [69]: temp_df = df[['1960']]
type(df), type(temp_df), temp_df
```

```
Out[69]: (pandas.core.frame.DataFrame,
pandas.core.frame.DataFrame,
1960
0      54208.0
1    130836765.0
2     8996967.0
3    96396419.0
4    5454938.0
..      ...
261    947000.0
262   5315351.0
263  17099836.0
264   3070780.0
265   3776679.0

[266 rows x 1 columns])
```

```
In [70]: temp_df = df.iloc[0]
type(df), type(temp_df), temp_df
```

```
Out[70]: (pandas.core.frame.DataFrame,
pandas.core.series.Series,
Country Name      Aruba
Country Code      ABW
Indicator Name    Population, total
Indicator Code    SP.POP.TOTL
1960              54208.0
...
2017              105361.0
2018              105846.0
2019              106310.0
2020              106766.0
Unnamed: 65      NaN
Name: 0, Length: 66, dtype: object)
```



```
In [71]: df.sample()
```

Out[71]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2012	2013	2014	2015
58	Denmark	DNK	Population, total	SP.POP.TOTL	4579603.0	4611687.0	4647727.0	4684483.0	4722072.0	4759012.0	...	5591572.0	5614932.0	5643475.0	5683480.0

1 rows × 66 columns



Indexing

```
In [72]: df = pd.read_csv(file_path, skiprows=4, index_col=1)
df
```

Out[72]:

	Country Name	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966	...	2012
Country Code												
ABW	Aruba	Population, total	SP.POP.TOTL	54208.0	55434.0	56234.0	56699.0	57029.0	57357.0	57702.0	...	102565.0
AFE	Africa Eastern and Southern	Population, total	SP.POP.TOTL	130836765.0	134159786.0	137614644.0	141202036.0	144920186.0	148769974.0	152752671.0	...	547482863.0
AFG	Afghanistan	Population, total	SP.POP.TOTL	8996967.0	9169406.0	9351442.0	9543200.0	9744772.0	9956318.0	10174840.0	...	31161378.0
AFW	Africa Western and Central	Population, total	SP.POP.TOTL	96396419.0	98407221.0	100506960.0	102691339.0	104953470.0	107289875.0	109701811.0	...	370243017.0
AGO	Angola	Population, total	SP.POP.TOTL	5454938.0	5531451.0	5608499.0	5679409.0	5734995.0	5770573.0	5781305.0	...	25107925.0
...
XKX	Kosovo	Population, total	SP.POP.TOTL	947000.0	966000.0	994000.0	1022000.0	1050000.0	1078000.0	1106000.0	...	1807106.0
YEM	Yemen, Rep.	Population, total	SP.POP.TOTL	5315351.0	5393034.0	5473671.0	5556767.0	5641598.0	5727745.0	5816241.0	...	24473176.0
ZAF	South Africa	Population, total	SP.POP.TOTL	17099836.0	17524533.0	17965733.0	18423157.0	18896303.0	19384838.0	19888259.0	...	52832659.0
ZMB	Zambia	Population, total	SP.POP.TOTL	3070780.0	3164330.0	3260645.0	3360099.0	3463211.0	3570466.0	3681953.0	...	14465148.0
ZWE	Zimbabwe	Population, total	SP.POP.TOTL	3776679.0	3905038.0	4039209.0	4178726.0	4322854.0	4471178.0	4623340.0	...	13115149.0

266 rows × 65 columns



In [73]: `df.loc['ABW']`

Out[73]:

Country Name	Aruba
Indicator Name	Population, total
Indicator Code	SP.POP.TOTL
1960	54208.0
1961	55434.0
...	
2017	105361.0
2018	105846.0
2019	106310.0
2020	106766.0
Unnamed: 65	NaN

Name: ABW, Length: 65, dtype: object

In [74]: `df.loc[['ABW']]`

Out[74]:

	Country Name	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966	...	2012	2013	2014	2015	201
Country Code																
ABW	Aruba	Population, total	SP.POP.TOTL	54208.0	55434.0	56234.0	56699.0	57029.0	57357.0	57702.0	...	102565.0	103165.0	103776.0	104339.0	104865.0

1 rows × 65 columns

In [75]: `df.iloc[[0]]`

Out[75]:

	Country Name	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966	...	2012	2013	2014	2015	201
Country Code																
ABW	Aruba	Population, total	SP.POP.TOTL	54208.0	55434.0	56234.0	56699.0	57029.0	57357.0	57702.0	...	102565.0	103165.0	103776.0	104339.0	104865.0

1 rows × 65 columns

```
In [76]: df.iloc[0]
```

```
Out[76]: Country Name      Aruba
Indicator Name      Population, total
Indicator Code      SP.POP.TOTL
1960                54208.0
1961                55434.0
...
2017                105361.0
2018                105846.0
2019                106310.0
2020                106766.0
Unnamed: 65         NaN
Name: ABW, Length: 65, dtype: object
```

Cetral Tendency and Dispersion

```
In [77]: df.mean()
```

```
c:\users\saif7\appdata\local\programs\python\python37\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
"""Entry point for launching an IPython kernel.
```

```
Out[77]: 1960          1.173116e+08
1961          1.188586e+08
1962          1.209679e+08
1963          1.235882e+08
1964          1.262297e+08
...
2017          3.068068e+08
2018          3.105688e+08
2019          3.142511e+08
2020          3.178433e+08
Unnamed: 65    NaN
Length: 62, dtype: float64
```

```
In [78]: df.mean(axis=0)
```

```
c:\users\saif7\appdata\local\programs\python\python37\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
    """Entry point for launching an IPython kernel.
```

```
Out[78]: 1960      1.173116e+08
         1961      1.188586e+08
         1962      1.209679e+08
         1963      1.235882e+08
         1964      1.262297e+08
         ...
         2017      3.068068e+08
         2018      3.105688e+08
         2019      3.142511e+08
         2020      3.178433e+08
         Unnamed: 65      NaN
         Length: 62, dtype: float64
```

```
In [79]: df.mean(axis=1)
```

```
c:\users\saif7\appdata\local\programs\python\python37\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
    """Entry point for launching an IPython kernel.
```

```
Out[79]: Country Code
         ABW      7.676846e+04
         AFE      3.363310e+08
         AFG      1.869939e+07
         AFW      2.287737e+08
         AGO      1.413545e+07
         ...
         XKX      1.595687e+06
         YEM      1.400506e+07
         ZAF      3.699079e+07
         ZMB      8.775118e+06
         ZWE      9.491677e+06
         Length: 266, dtype: float64
```

```
In [81]: df['2020'].mean()
```

```
Out[81]: 317843304.5037879
```

```
In [82]: df.loc['ZWE']
```

```
Out[82]: Country Name      Zimbabwe
Indicator Name      Population, total
Indicator Code      SP.POP.TOTL
1960      3776679.0
1961      3905038.0
...
2017      14236599.0
2018      14438812.0
2019      14645473.0
2020      14862927.0
Unnamed: 65      NaN
Name: ZWE, Length: 65, dtype: object
```

```
In [83]: df['2020'].loc['ZWE']
```

```
Out[83]: 14862927.0
```

```
In [84]: df.loc['ZWE']['2020']
```

```
Out[84]: 14862927.0
```

```
In [86]: df.loc[['CAN']][['2020']]
```

```
Out[86]:      2020
Country Code
CAN      38005238.0
```

```
In [87]: df.columns
```

```
Out[87]: Index(['Country Name', 'Indicator Name', 'Indicator Code', '1960', '1961',
               '1962', '1963', '1964', '1965', '1966', '1967', '1968', '1969', '1970',
               '1971', '1972', '1973', '1974', '1975', '1976', '1977', '1978', '1979',
               '1980', '1981', '1982', '1983', '1984', '1985', '1986', '1987', '1988',
               '1989', '1990', '1991', '1992', '1993', '1994', '1995', '1996', '1997',
               '1998', '1999', '2000', '2001', '2002', '2003', '2004', '2005', '2006',
               '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015',
               '2016', '2017', '2018', '2019', '2020', 'Unnamed: 65'],
              dtype='object')
```

In [91]: df.iloc[:2]

Out[91]:

	Country Name	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966	...	2012	
Country Code													
ABW	Aruba	Population, total	SP.POP.TOTL	54208.0	55434.0	56234.0	56699.0	57029.0	57357.0	57702.0	...	102565.0	
AFE	Africa Eastern and Southern	Population, total	SP.POP.TOTL	130836765.0	134159786.0	137614644.0	141202036.0	144920186.0	148769974.0	152752671.0	...	547482863.0	562

2 rows × 65 columns



Slicing and Iterating

In [94]: df.iloc[10:12, :]

Out[94]:

	Country Name	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966	...	2012	2013	2014
Country Code														
ARM	Armenia	Population, total	SP.POP.TOTL	1874119.0	1941498.0	2009524.0	2077584.0	2145004.0	2211316.0	2276038.0	...	2884239.0	2897593.0	2912403.0
ASM	American Samoa	Population, total	SP.POP.TOTL	20127.0	20605.0	21246.0	22029.0	22850.0	23675.0	24473.0	...	55669.0	55717.0	55791.0

2 rows × 65 columns



In [95]: df.iloc[:, 2]

Out[95]:

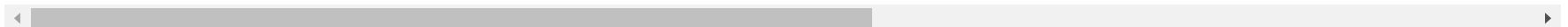
	Country Name	Indicator Name
Country Code		
ABW	Aruba	Population, total
AFE	Africa Eastern and Southern	Population, total

```
In [96]: dff = df.loc[['CAN', 'BGD', 'IND', 'CHN', 'USA', 'PAK']]
dff
```

Out[96]:

	Country Name	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966	...	2012
Country Code												
CAN	Canada	Population, total	SP.POP.TOTL	17909009.0	18271000.0	18614000.0	18964000.0	19325000.0	19678000.0	20048000.0	...	3.471422e+07
BGD	Bangladesh	Population, total	SP.POP.TOTL	48013505.0	49362834.0	50752150.0	52202008.0	53741721.0	55385114.0	57157651.0	...	1.510057e+08
IND	India	Population, total	SP.POP.TOTL	450547675.0	459642166.0	469077191.0	478825602.0	488848139.0	499123328.0	509631509.0	...	1.265780e+09
CHN	China	Population, total	SP.POP.TOTL	667070000.0	660330000.0	665770000.0	682335000.0	698355000.0	715185000.0	735400000.0	...	1.354190e+09
USA	United States	Population, total	SP.POP.TOTL	180671000.0	183691000.0	186538000.0	189242000.0	191889000.0	194303000.0	196560000.0	...	3.138777e+08
PAK	Pakistan	Population, total	SP.POP.TOTL	44988690.0	46065229.0	47198886.0	48387293.0	49627623.0	50917975.0	52260183.0	...	1.872801e+08

6 rows × 65 columns



```
In [98]: dff[['1960', '2020']]
```

Out[98]:

	1960	2020
Country Code		
CAN	17909009.0	3.800524e+07
BGD	48013505.0	1.646894e+08
IND	450547675.0	1.380004e+09
CHN	667070000.0	1.410929e+09
USA	180671000.0	3.294841e+08
PAK	44988690.0	2.208923e+08


```
In [103]: dff = df.loc[['CAN', 'BGD', 'IND', 'CHN', 'USA', 'PAK']][['1960', '2020']]
dff
```

Out[103]:

	1960	2020
Country Code		
CAN	17909009.0	3.800524e+07
BGD	48013505.0	1.646894e+08
IND	450547675.0	1.380004e+09
CHN	667070000.0	1.410929e+09
USA	180671000.0	3.294841e+08
PAK	44988690.0	2.208923e+08

```
In [102]: for index, row in dff.iterrows():
          print(index, "\n=====")
          print(row)
```

CAN

=====

1960 17909009.0

2020 38005238.0

Name: CAN, dtype: float64

BGD

=====

1960 48013505.0

2020 164689383.0

Name: BGD, dtype: float64

IND

=====

1960 4.505477e+08

2020 1.380004e+09

Name: IND, dtype: float64

CHN

=====

1960 6.670700e+08

2020 1.410929e+09

Name: CHN, dtype: float64

USA

=====

1960 180671000.0

2020 329484123.0

Name: USA, dtype: float64

PAK

=====

1960 44988690.0

2020 220892331.0

Name: PAK, dtype: float64

Fitering and Reshaping

```
In [104]: dff
```

Out[104]:

	1960	2020
Country Code		
CAN	17909009.0	3.800524e+07
BGD	48013505.0	1.646894e+08
IND	450547675.0	1.380004e+09
CHN	667070000.0	1.410929e+09
USA	180671000.0	3.294841e+08
PAK	44988690.0	2.208923e+08

```
In [105]: dff[(dff['1960'] < 50000000)]
```

Out[105]:

	1960	2020
Country Code		
CAN	17909009.0	38005238.0
BGD	48013505.0	164689383.0
PAK	44988690.0	220892331.0

```
In [106]: dff[(dff['1960'] < 50000000) & (dff['1960'] > 30000000)]
```

Out[106]:

	1960	2020
Country Code		
BGD	48013505.0	164689383.0
PAK	44988690.0	220892331.0

```
In [108]: dff[(dff['1960'] < 30000000) | (dff['1960'] > 50000000)]
```

```
Out[108]:
```

	1960	2020
Country Code		
CAN	17909009.0	3.800524e+07
IND	450547675.0	1.380004e+09
CHN	667070000.0	1.410929e+09
USA	180671000.0	3.294841e+08

```
In [109]: dff.sort_values(by='Country Code')
```

```
Out[109]:
```

	1960	2020
Country Code		
BGD	48013505.0	1.646894e+08
CAN	17909009.0	3.800524e+07
CHN	667070000.0	1.410929e+09
IND	450547675.0	1.380004e+09
PAK	44988690.0	2.208923e+08
USA	180671000.0	3.294841e+08

```
In [110]: dff.sort_index()
```

```
Out[110]:
```

	1960	2020
Country Code		
BGD	48013505.0	1.646894e+08
CAN	17909009.0	3.800524e+07
CHN	667070000.0	1.410929e+09
IND	450547675.0	1.380004e+09
PAK	44988690.0	2.208923e+08
USA	180671000.0	3.294841e+08

```
In [111]: dff.sort_values(by='1960')
```

Out[111]:

	1960	2020
Country Code		
CAN	17909009.0	3.800524e+07
PAK	44988690.0	2.208923e+08
BGD	48013505.0	1.646894e+08
USA	180671000.0	3.294841e+08
IND	450547675.0	1.380004e+09
CHN	667070000.0	1.410929e+09

```
In [112]: dff.sort_values(by='2020')
```

Out[112]:

	1960	2020
Country Code		
CAN	17909009.0	3.800524e+07
BGD	48013505.0	1.646894e+08
PAK	44988690.0	2.208923e+08
USA	180671000.0	3.294841e+08
IND	450547675.0	1.380004e+09
CHN	667070000.0	1.410929e+09

```
In [113]: dff.sort_values(by='2020', ascending=False)
```

Out[113]:

	1960	2020
Country Code		
CHN	667070000.0	1.410929e+09
IND	450547675.0	1.380004e+09
USA	180671000.0	3.294841e+08
PAK	44988690.0	2.208923e+08
BGD	48013505.0	1.646894e+08
CAN	17909009.0	3.800524e+07

```
In [ ]:
```