```
import pandas as pd
```

## Making a Data frame

## Working on DataSet from Seaborn Library

```
import seaborn as sns
df=sns.load_dataset("tips")
print(df)
```

```
     total_bill   tip     sex smoker   day    time  size
0         16.99  1.01  Female     No   Sun  Dinner     2
1         10.34  1.66    Male     No   Sun  Dinner     3
2         21.01  3.50    Male     No   Sun  Dinner     3
3         23.68  3.31    Male     No   Sun  Dinner     2
4         24.59  3.61  Female     No   Sun  Dinner     4
..          ...   ...     ...    ...   ...     ...   ...
239       29.03  5.92    Male     No   Sat  Dinner     3
240       27.18  2.00  Female    Yes   Sat  Dinner     2
241       22.67  2.00    Male    Yes   Sat  Dinner     2
242       17.82  1.75    Male     No   Sat  Dinner     2
243       18.78  3.00  Female     No  Thur  Dinner     2

[244 rows x 7 columns]
```

## Checking information about data

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   total_bill  244 non-null    float64
 1   tip         244 non-null    float64
 2   sex         244 non-null    category
 3   smoker      244 non-null    category
 4   day         244 non-null    category
 5   time        244 non-null    category
 6   size        244 non-null    int64
dtypes: category(4), float64(2), int64(1)
memory usage: 7.4 KB
```

## Checking first five entries

```
df.head()
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |

## Checking last five entries

```
df.tail()
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **239** | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 |
| **240** | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 |
| **241** | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 |
| **242** | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 |
| **243** | 18.78 | 3.00 | Female | No | Thur | Dinner | 2 |

## Summary Statistics

```
df.describe()
```

| | total_bill | tip | size |
|---|---|---|---|
| **count** | 244.000000 | 244.000000 | 244.000000 |
| **mean** | 19.785943 | 2.998279 | 2.569672 |
| **std** | 8.902412 | 1.383638 | 0.951100 |
| **min** | 3.070000 | 1.000000 | 1.000000 |
| **25%** | 13.347500 | 2.000000 | 2.000000 |
| **50%** | 17.795000 | 2.900000 | 2.000000 |
| **75%** | 24.127500 | 3.562500 | 3.000000 |
| **max** | 50.810000 | 10.000000 | 6.000000 |

Double-click (or enter) to edit

## Checking number of rows and columns

```
df.shape
```

```
(244, 7)
```

```
df.shape[0]
```

```
244
```

```
df.shape[1]
```

```
7
```

```
rows="The number of rows are",df.shape[0]
columns="The number of columnnbs are",df.shape[1]
print(rows)
print(columns)
```

```
('The number of rows are', 244)
('The number of columnnbs are', 7)
```

## checking columns names

```
df.columns
```

```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtype='object')
```

## checking row headings

```
df.index
```

```
RangeIndex(start=0, stop=244, step=1)
```

## removing specific columns

```
df1=df.drop(["day","smoker"],axis=1)
(df1)
```

| | total_bill | tip | sex | time | size |
|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | Dinner | 3 |
| **3** | 23.68 | 3.31 | Male | Dinner | 2 |
| **4** | 24.59 | 3.61 | Female | Dinner | 4 |
| **...** | ... | ... | ... | ... | ... |
| **239** | 29.03 | 5.92 | Male | Dinner | 3 |
| **240** | 27.18 | 2.00 | Female | Dinner | 2 |
| **241** | 22.67 | 2.00 | Male | Dinner | 2 |
| **242** | 17.82 | 1.75 | Male | Dinner | 2 |
| **243** | 18.78 | 3.00 | Female | Dinner | 2 |

244 rows × 5 columns

## checking missing value

```
df.isnull().sum()
```

```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

## Checking unique value

```
df.time.unique()
```

```
['Dinner', 'Lunch']
Categories (2, object): ['Lunch', 'Dinner']
```

```
df.day.unique()
```

```
['Sun', 'Sat', 'Thur', 'Fri']
Categories (4, object): ['Thur', 'Fri', 'Sat', 'Sun']
```

## Groupby

```
df.groupby(["size"]).mean()
```

```
<ipython-input-53-fb39ccfefd0e>:1: FutureWarning: The default value of numeric_only in [
  df.groupby(["size"]).mean()
```

| size | total_bill | tip |
|---|---|---|
| **1** | 7.242500 | 1.437500 |
| **2** | 16.448013 | 2.582308 |
| **3** | 23.277632 | 3.393158 |
| **4** | 28.613514 | 4.135405 |
| **5** | 30.068000 | 4.028000 |
| **6** | 34.830000 | 5.225000 |

Colab paid products - Cancel contracts here

✓ 0s completed at 12:16 AM