# Descriptive Statistics

```
import warnings
warnings.filterwarnings('ignore')
```

# Load 'tips.csv' Data

```
import seaborn as sns

DF = sns.load_dataset('tips')
```

# I. pandas

## 1) DataFrame Information

```
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: 7.3 KB
```

```
DF.head()
```

## 2) .describe( )

```
DF.tip.describe()
```

```
count    244.000000
mean       2.998279
std        1.383638
min        1.000000
25%        2.000000
50%        2.900000
75%        3.562500
max       10.000000
Name: tip, dtype: float64
```

## 3) .sum( )

```
%precision 5

DF.tip.sum()
```

```
731.58000
```

## 4) .mean( )

```
DF.tip.mean()
```

```
2.99828
```

## 5) .min( )

```
DF.tip.min()
```

```
1.00000
```

## 6) .quantile(q = 0.25)

```
DF.tip.quantile(q = 0.25)
```

```
2.00000
```

### 7) .median( )

```
DF.tip.median()
```

```
2.90000
```

### 8) .quantile(q = 0.75)

```
DF.tip.quantile(q = 0.75)
```

```
3.56250
```

### 9) .max( )

```
DF.tip.max()
```

```
10.00000
```

### 10) .var(ddof = 0)

- Dynamic Degrees Of Freedom(default = 1)

```
DF.tip.var(ddof = 0)
```

```
1.90661
```

### 11) .std(ddof = 0)

```
DF.tip.std(ddof = 0)
```

```
1.38080
```

## II. numpy

```
import numpy as np
```

### 1) Casting to Array

```
AR = np.array(DF.tip)
AR = np.array(DF.tip)
```

```
AR[:5]
```

```
array([1.01, 1.66, 3.5 , 3.31, 3.61])
```

## 2) .sum( )

```
AR.sum()
```

```
731.58000
```

## 3) .mean( )

```
AR.mean()
```

```
2.99828
```

## 4) .min( )

```
AR.min()
```

```
1.00000
```

## 5) .max( )

```
AR.max()
```

```
10.00000
```

## 6) .var(ddof = 0)

- default = 0

```
AR.var(ddof = 0)
```

```
1.90661
```

## 7) .std(ddof = 0)

```
AR.std(ddof = 0)
```

```
1.38080
```

## III. scipy

```
import scipy as sp
```

### 1) .sum( )

```
sp.sum(DF['tip'])
```

```
731.58000
```

### 2) .mean( )

```
sp.mean(DF['tip'])
```

```
2.99828
```

### 3) .amin( )

```
sp.amin(DF['tip'])
```

```
1.00000
```

### 4) .median( )

```
sp.median(DF['tip'])
```

```
2.90000
```

### 5) .amax( )

```
sp.amax(DF['tip'])
```

```
10.00000
```

## 6) .var(ddof = 0)

- default = 0

```
sp.var(DF['tip'], ddof = 0)
```

```
1.90661
```

## 7) .std(ddof = 0)

```
sp.std(DF['tip'], ddof = 0)
```

```
1.38080
```

## 8) .stats.mode( )

- DF.tip.value_counts( )

```
sp.stats.mode(DF['tip'])
```

```
ModeResult(mode=array([2.]), count=array([33]))
```

#

#

#

# The End

#

#

#