

```

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")

data = [1,2,3,4,5]
data

[1, 2, 3, 4, 5]

np.mean(data)

3.0

df = sns.load_dataset('tips')
df

{"summary":{"\n  \"name\": \"df\",\n  \"rows\": 244,\n  \"fields\": [\n    {\n      \"column\": \"total_bill\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 8.902411954856856,\n        \"min\": 3.07,\n        \"max\": 50.81,\n        \"num_unique_values\": 229,\n        \"samples\": [\n          22.12,\n          20.23,\n          14.78\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"tip\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 1.3836381890011826,\n        \"min\": 1.0,\n        \"max\": 10.0,\n        \"num_unique_values\": 123,\n        \"samples\": [\n          3.35,\n          1.5,\n          6.73\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n        \"samples\": [\n          \"Male\",\n          \"Female\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"smoker\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n        \"samples\": [\n          \"Yes\",\n          \"No\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"day\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 4,\n        \"samples\": [\n          \"Sat\",\n          \"Fri\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"time\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n        \"samples\": [\n          \"Lunch\",\n          \"Dinner\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"size\",\n      \"properties\": {\n        \"dtype\": \"number\",

```

```
\ "std\ ": 0,\n          \ "min\ ": 1,\n          \ "max\ ": 6,\n
\ "num_unique_values\ ": 6,\n          \ "samples\ ": [\n          2,\n
3\n          ],\n          \ "semantic_type\ ": \ "\",\n
\ "description\ ": \ "\",\n          }\n          }\n          ]\n
n}","type":"dataframe","variable_name":"df"}
```

```
np.mean(df['total_bill'])
```

```
19.78594262295082
```

```
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
```

```
np.median(data)
```

```
3.0
```

```
np.median(df["total_bill"])
```

```
17.795
```

#mean is affected by outliers

```
np.mean(data)
```

```
3.0
```

```
data = [1 , 2 , 3 , 4, 500]
```

```
np.mean(data)
```

```
102.0
```

#median is not affected by outliers

```
data = data = [1 , 2 , 3 , 4 , 5]
```

```
np.median(data)
```

```
3.0
```

```
data = [1, 2, 3, 4, 500]
np.median(data)      #median is same for both data it is not affected
by the outliers
```

3.0

```
data = [1, 2, 100, 100, 100, 3]
import statistics
statistics.mode(data)
```

100

```
df.describe()
```

```
{ "summary": "{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n    {\n      \"column\": \"total_bill\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 80.57800601480773,\n        \"min\": 3.07,\n        \"max\": 244.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          19.78594262295082,\n          17.795,\n          244.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"tip\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 85.10953159599222,\n        \"min\": 1.0,\n        \"max\": 244.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          2.99827868852459,\n          2.9,\n          244.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"size\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 85.39686221774684,\n        \"min\": 0.9510998047322332,\n        \"max\": 244.0,\n        \"num_unique_values\": 7,\n        \"samples\": [\n          244.0,\n          2.569672131147541,\n          3.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}", "type": "dataframe" }
```

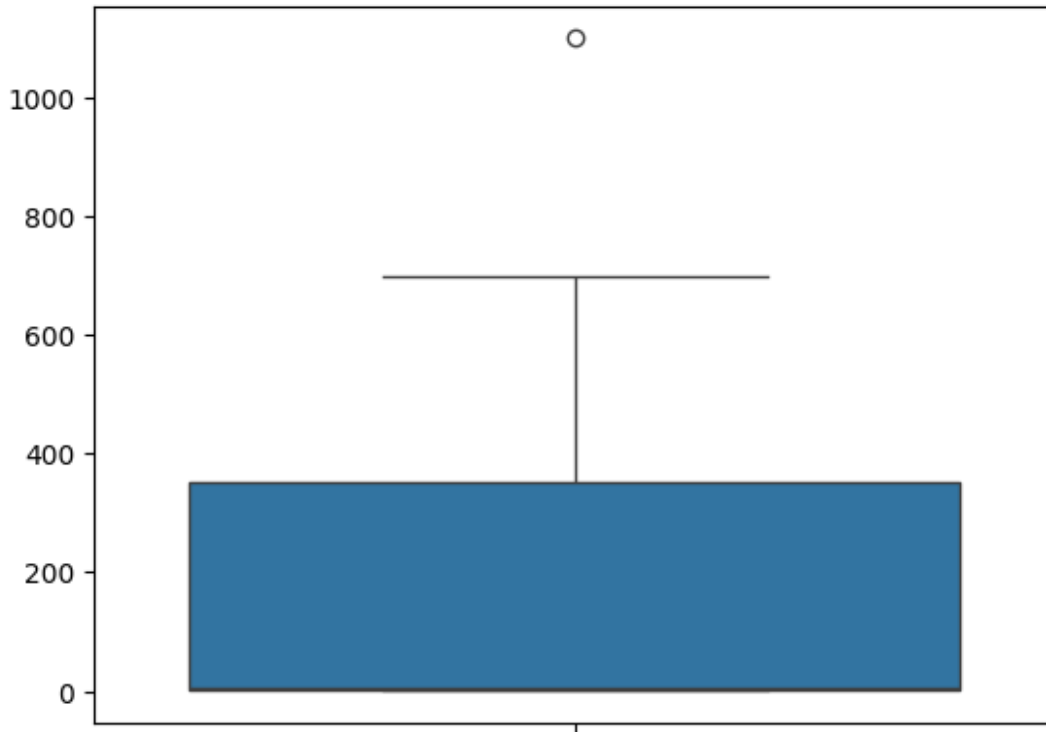
```
data = [1,2,3,4,5]
data.append(1100)
data.append(700)
```

data

[1, 2, 3, 4, 5, 1100, 700]

```
sns.boxplot(data)
```

<Axes: >



```

np.var(data)
175635.91836734695

np.mean(data)
259.2857142857143

np.std(data)
419.0893918573303

statistics.variance(data) #sample variance
204908.57142857142

statistics.pvariance(data)
175635.91836734695

df.corr(numeric_only = True)

{"summary":{"\n  \"name\": \"df\",\n  \"rows\": 3,\n  \"fields\": [\n    {\n      \"column\": \"total_bill\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.21310904884233575,\n        \"min\": 0.5983151309049014,\n        \"max\": 1.0,\n        \"num_unique_values\": 3,\n        \"samples\": [\n          1.0,\n          0.6757341092113648,\n          0.5983151309049014\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}}

```

```

n      },\n      {\n          \"column\": \"tip\", \n          \"properties\": {\n\n\"dtype\": \"number\", \n          \"std\": 0.25843188743740914, \n          \"min\": 0.48929877523035714, \n          \"max\": 1.0, \n          \"num_unique_values\": 3, \n          \"samples\": [\n0.6757341092113648, \n          1.0, \n          0.48929877523035714\n], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n}\n      }, \n      {\n          \"column\": \"size\", \n          \"properties\": {\n\n\"dtype\": \"number\", \n          \"std\": 0.2689643845213583, \n          \"min\": 0.48929877523035714, \n          \"max\": 1.0, \n          \"num_unique_values\": 3, \n          \"samples\": [\n0.5983151309049014, \n          0.48929877523035714, \n          1.0\n], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n}\n      }\n    ], \n    \"type\": \"dataframe\"}

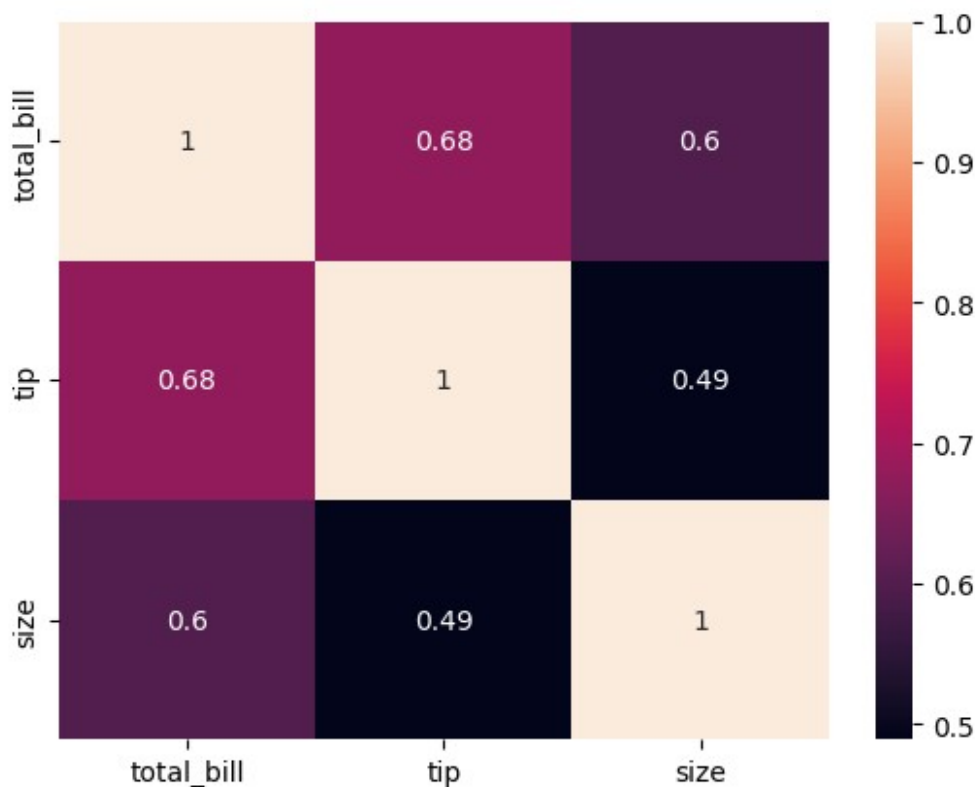
```

```

corr = df.corr(numeric_only = True)
sns.heatmap(corr, annot=True)

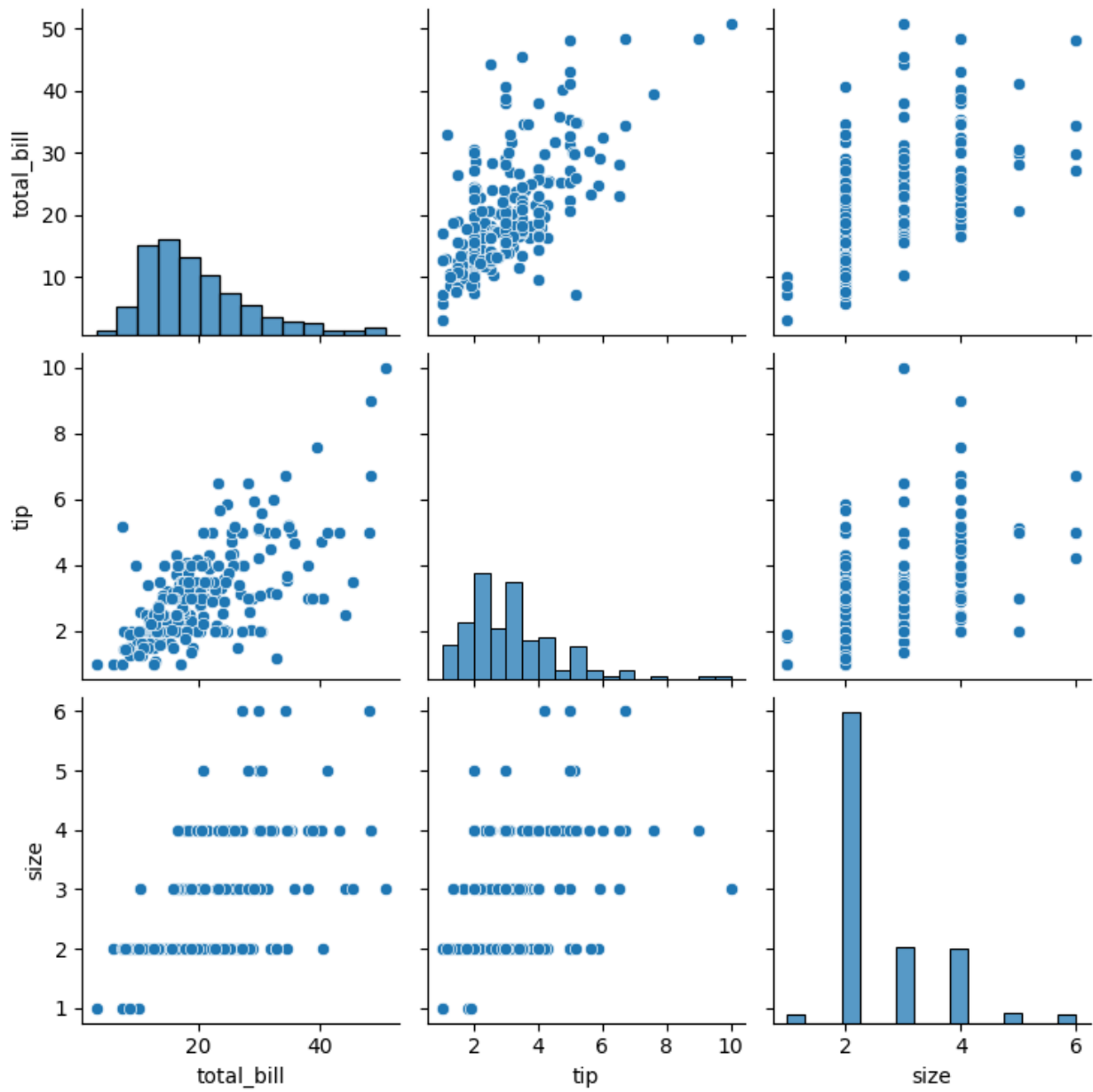
```

<Axes: >

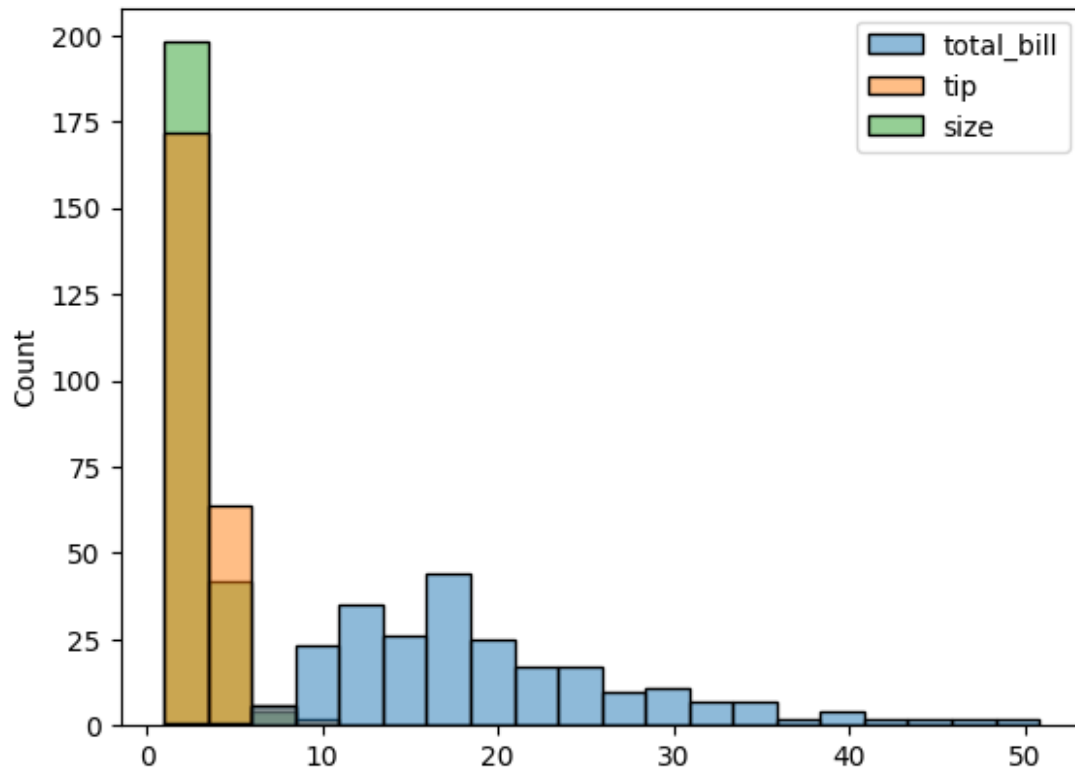


```
sns.pairplot(df)
```

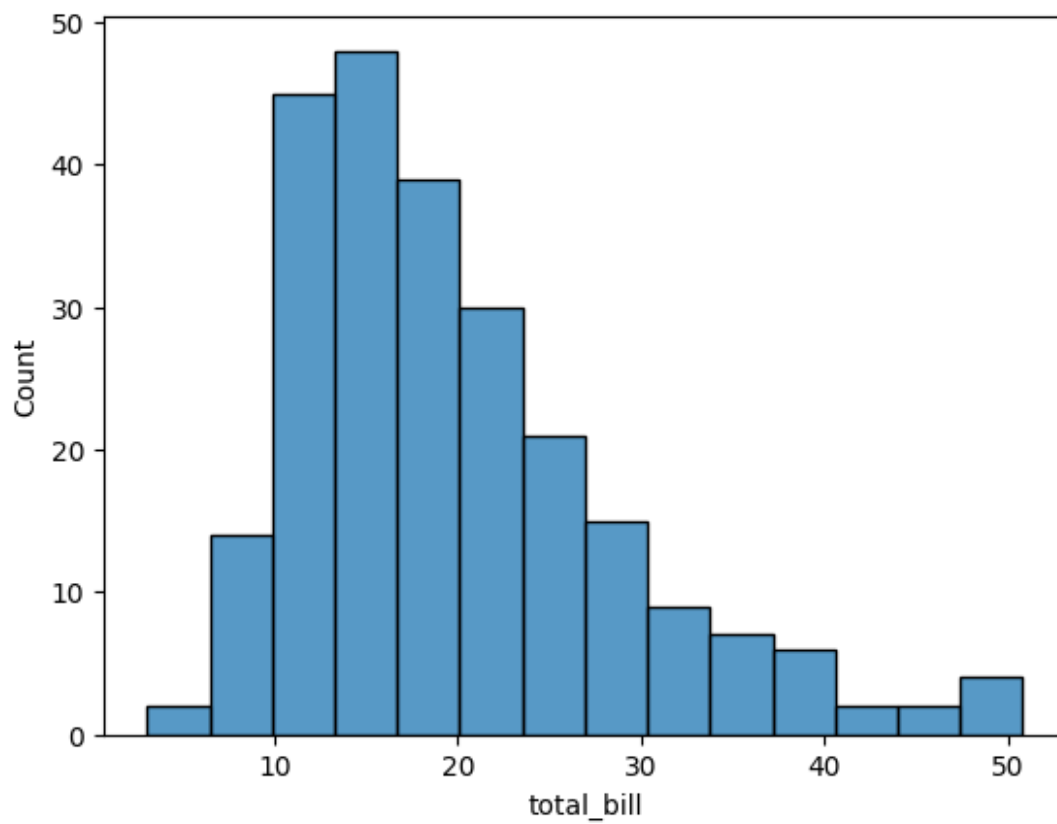
<seaborn.axisgrid.PairGrid at 0x7ceab8e196c0>



```
sns.histplot(df)
<Axes: ylabel='Count'>
```



```
sns.histplot(df['total_bill'])  
<Axes: xlabel='total_bill', ylabel='Count'>
```



```
sns.histplot(df['total_bill'], kde = True)  
<Axes: xlabel='total_bill', ylabel='Count'>
```