Confidence Interval case study

How to Calculate Confidence Intervals in Python?

- Read
- Discuss
- Courses
- Practice

• In this article, we will be looking at the different ways to calculate confidence intervals using various distributions in the Python programming language Confidence interval for a mean is a range of values that is likely to contain a population mean with a certain level of confidence.

Formula:

Confidence Interval = $x(+/-)t^*(s/\sqrt{n})$

- **x:** sample mean
- **t:** t-value that corresponds to the confidence level
- **s:** sample standard deviation
- **n:** sample size

Method 1: Calculate confidence Intervals using the t Distribution

This approach is used to calculate confidence Intervals for the small dataset where the n<=30 and for this, the user needs to call the t.interval() function from the scipy.stats library to get the confidence interval for a population means of the given dataset in python.

Example 1:

In this example, we will be using the data set of size(n=20) and will be calculating the 90% confidence Intervals using the t Distribution using the t.interval() function and passing the alpha parameter to 0.90 in the python.

```
import numpy as np

import scipy.stats as st


# define sample data

gfg_data = [1, 1, 1, 2, 2, 2, 3, 3, 3, 3,3, 4, 4, 5, 5, 5, 6, 7, 8, 10]

# create 90% confidence interval

st.t.interval(alpha=0.90, df=len(gfg_data)-1,
                          loc=np.mean(gfg_data),
                          scale=st.sem(gfg_data))
```

In this example, we will be using the data set of size(n=20) and will be calculating the 90% confidence Intervals using the t Distribution using the t.interval() function and passing the alpha parameter to 0.99 in the python.

•Python

```
import numpy as np
import scipy.stats as st

# define sample data
gfg_data = [1, 1, 1, 2, 2, 2, 3, 3, 3,
            3, 3, 4, 4, 5, 5, 5, 6,
            7, 8, 10]

# create 99% confidence interval
st.t.interval(alpha=0.99,
              df=len(gfg_data)-1,
              loc=np.mean(gfg_data),
              scale=st.sem(gfg_data))
```

Example 3:

In this example, we will be using the random data set of size(n=100) and will be calculating the 90% confidence Intervals using the norm Distribution using the norm.interval() function and passing the alpha parameter to 0.90 in the python.

```
import numpy as np

import scipy.stats as st

# define sample data

gfg_data = np.random.randint(5, 10, 100)

# create 90% confidence interval

# for population mean weight

st.norm.interval(alpha=0.90,
                 loc=np.mean(gfg_data),
                 scale=st.sem(gfg_data))
```