

```
from scipy.stats import ttest_1samp
import numpy as np

# Creating a sample of ages
ages = [45, 89, 23, 46, 12, 69, 45, 24, 34, 67]
print(ages)

# Calculating the mean of the sample
mean = np.mean(ages)
print(mean)

# Performing the T-Test
t_test, p_val = ttest_1samp(ages, 30)
print("P-value is: ", p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

[45, 89, 23, 46, 12, 69, 45, 24, 34, 67]
45.4
P-value is:  0.07179988272763561
We can accept the null hypothesis

from scipy.stats import ttest_ind
import numpy as np

# Creating the data groups
data_group1 = np.array([12, 18, 12, 13, 15, 1, 7,
                        20, 21, 25, 19, 31, 21, 17,
                        17, 15, 19, 15, 12, 15])
data_group2 = np.array([23, 22, 24, 25, 21, 26, 21,
                        21, 25, 30, 24, 21, 23, 19,
                        14, 18, 14, 12, 19, 15])

# Calculating the mean of the two data groups
mean1 = np.mean(data_group1)
mean2 = np.mean(data_group2)

# Print mean values
print("Data group 1 mean value:", mean1)
print("Data group 2 mean value:", mean2)

# Calculating standard deviation
std1 = np.std(data_group1)
std2 = np.std(data_group2)

# Printing standard deviation values
print("Data group 1 std value:", std1)
print("Data group 2 std value:", std2)

# Implementing the t-test
t_test, p_val = ttest_ind(data_group1, data_group2)
print("The P-value is: ", p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print("We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

Data group 1 mean value: 16.25
Data group 2 mean value: 20.85
Data group 1 std value: 6.171507109288622
Data group 2 std value: 4.452808102759426
The P-value is:  0.012117171124028792
We can reject the null hypothesis
```

```
# Python program to implement Paired Sample T-Test on the two dependent samples

# Importing the required libraries
import pandas as pd
from scipy import stats

# Creating two samples
sample1 = [29, 30, 33, 41, 38, 36,
           35, 31, 29, 30]
sample2 = [31, 32, 33, 39, 30, 33,
           30, 28, 29, 31]

# Performing paired sample t-test
t_test, p_val = stats.ttest_rel(sample1, sample2)
print("The P-value of the test is: ", p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print("We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

    The P-value of the test is: 0.15266056244408904
    We can accept the null hypothesis

# Python program to implement One Sample Z-Test

# Importing the required libraries
import pandas as pd
from scipy import stats
from statsmodels.stats import weightstats as stests

# Creating a dataset
data = [89, 93, 95, 93, 97, 98, 96, 99, 93, 97,
        110, 104, 119, 105, 104, 110, 110, 112, 115, 114]

# Performing the z-test
z_test ,p_val = stests.ztest(data, x2 = None, value = 160)
print(p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print("We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

    2.417334226169332e-186
    We can reject the null hypothesis

# Python program to implement Two Sample Z-Test

# Importing the required libraries
import pandas as pd
from scipy import stats
from statsmodels.stats import weightstats as stests

# Creating a dataset
data1 = [83, 85, 86, 90, 90, 93, 93, 95, 97, 97,
        106, 108, 106, 108, 111, 113, 113, 112, 116, 111]

data2 = [92, 92, 90, 93, 93, 97, 94, 98, 109, 108,
        110, 117, 110, 115, 114, 114, 130, 130, 149, 131]

# Implementing the two-sample z-test
z_test ,p_val = stests.ztest(data1, x2 = data2, value = 0, alternative = 'two-sided')
print(p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print("We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

    0.04813782199434202
    We can reject the null hypothesis
```

```
# Python program to implement One-Way f-test

# Importing the required libraries
import scipy.stats

# Creating sample data
data1 = [0.0842, 0.0368, 0.0847, 0.0935, 0.0376, 0.0963, 0.0684,
         0.0758, 0.0854, 0.0855]
data2 = [0.0785, 0.0845, 0.0758, 0.0853, 0.0946, 0.0785, 0.0853,
         0.0685]
data3 = [0.0864, 0.2522, 0.0894, 0.2724, 0.0853, 0.1367, 0.853]

# Performing the F-Test
f_test, p_val = scipy.stats.f_oneway(data1, data2, data3)
print("p-value is: ", p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

    p-value is:  0.04043792126789144
    We can reject the null hypothesis

from scipy.stats import chi2_contingency

# defining our data
data = [[231, 256, 321], [245, 312, 213]]

# Performing chi-square test
test, p_val, dof, expected_val = chi2_contingency(data)

# interpreting the p-value
alpha = 0.05
print("The p-value of our test is " + str(p_val))

# Checking the hypothesis
if p_val <= alpha:
    print('We can reject the null hypothesis')
else:
    print('We can accept the null hypothesis')

    The p-value of our test is 1.4585823594475804e-06
    We can reject the null hypothesis

from scipy.stats import mannwhitneyu

# Creating the dataset
data1 = [0.978, 2.792, 0.248, -0.820, -0.102, -1.203, 0.102, -1.392, -1.395, -1.928]
data2 = [1.283, -0.284, -0.821, -0.792, -0.793, -0.294, 0.600, 1.294, -1.183, -0.284]

# Implementing the test
k_test, p_val = mannwhitneyu(data1, data2)
print("P-value is: ", p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

    P-value is:  0.520366020896531
    We can accept the null hypothesis

from scipy.stats import wilcoxon

# Creating the dataset
data1 = [0.978, 2.792, 0.248, -0.820, -0.102, -1.203, 0.102, -1.392, -1.395, -1.928]
data2 = [1.283, -0.284, -0.821, -0.792, -0.793, -0.294, 0.600, 1.294, -1.183, -0.284]

# Implementing the test
w_test, p_val = wilcoxon(data1, data2)
print("P-value is: ", p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")
```

```

P-value is: 0.625
We can accept the null hypothesis

from scipy.stats import kruskal

# Creating the dataset
data1 = [0.978, 2.792, 0.248, -0.820, -0.102, -1.203, 0.102, -1.392, -1.395, -1.928]
data2 = [1.283, -0.284, -0.821, -0.792, -0.793, -0.294, 0.600, 1.294, -1.183, -0.284]

# Implementing the test
k_test, p_val = kruskal(data1, data2)
print("P-value is: ", p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

P-value is: 0.49612971494281877
We can accept the null hypothesis

from scipy.stats import friedmanchisquare

# Creating the dataset
data1 = [0.978, 2.792, 0.248, -0.820, -0.102, -1.203, 0.102, -1.392, -1.395, -1.928]
data2 = [1.283, -0.284, -0.821, -0.792, -0.793, -0.294, 1.100, 0.294, -0.183, -1.284]
data3 = [-0.324, 1.346, 1.148, -1.258, -0.233, 0.749, 0.157, 0.529, -0.240, -1.254]

# Implementing the test
f_test, p_val = friedmanchisquare(data1, data2, data3)
print("P-value is: ", p_val)

# taking the threshold value as 0.05 or 5%
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

P-value is: 0.496585303791408
We can accept the null hypothesis

#
import pandas as pd
import seaborn as sb
from scipy import stats
#
d=sb.load_dataset('penguins')
d.info()
d.columns
#
'''
['species', 'island',
'bill_length_mm', 'bill_depth_mm', 'flipper_length_mm',
'body_mass_g', 'sex']
species = ['Adelie', 'Chinstrap', 'Gentoo']
island = ['Biscoe', 'Dream', 'Torgersen']
'''
#
d.dropna(inplace=True) #dropping rows having any cells with null value
#
# Independent Samples t-test - two tailed - variances are equal
# Null hypothesis H0: Mean of the body_mass of samples s1, s2 are equal
# Alternate hyp. H1: Mean of the body_mass of samples s1, s2 are NOT equal

is_t_BT, p = stats.ttest_ind(d.body_mass_g, d.body_mass_g,
equal_var = True)
p
# p > 0.05; failed to reject the null hypothesis
# Mean of the body_mass of samples are equal
# The body mass of 'Torgersen' island penguins
# donot differ from that in 'Dream'
#

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   species          344 non-null   object

```

```

1  island          344 non-null  object
2  bill_length_mm  342 non-null  float64
3  bill_depth_mm   342 non-null  float64
4  flipper_length_mm 342 non-null  float64
5  body_mass_g     342 non-null  float64
6  sex             333 non-null  object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
1.0

#
previous_buy=[5,6,5,5,2,4,4,4,5,6,5,4,2,5,6,3,6,3,5,5,2,5,5,5,5,5,5,5,5]
product_aware=[5,6,5,5,5,3,5,5,5,2,6,4,4,5,6,3,6,6,5,5,4,5,5,5,5,4,5,5,6
]
brand_aware=[5,6,5,5,5,4,5,5,5,6,6,6,5,6,6,6,6,5,5,4,5,5,4,5,5,5,3,6]
brand_loyal=[5,6,5,5,5,4,6,5,6,6,6,5,5,6,6,6,5,5,5,5,5,5,4,5,5,6,5,5]
#
from scipy import stats
import numpy as np
#
# H0: means of brand loyalty and previous buy are equal
# Ha: means of brand loyalty and previous buy are NOT equal
tps, p = stats.ttest_rel(brand_loyal,previous_buy)
p # 0.004
# p-value is less than 0.05; H0 is rejected
# means of brand loyalty and previous buy are NOT equal
#
# H0: means of brand loyalty and brand awareness are equal
tps, p = stats.ttest_rel(brand_loyal, brand_aware)
p # 0.374

```

```
0.37463839343035743
```

```

#
import pandas as pd
import seaborn as sb
from scipy import stats
#
d=sb.load_dataset('penguins')
d.info()
d.columns
#
'''

['species', 'island',
'bill_length_mm', 'bill_depth_mm', 'flipper_length_mm',
'body_mass_g', 'sex']
species = ['Adelie', 'Chinstrap', 'Gentoo']
island = ['Biscoe', 'Dream', 'Torgersen']
'''
#
d.dropna(inplace=True) #dropping rows having any cells with null value
#
# f-test to check whether the mean of all the distributions are equal
# Ho: var1 = var2 = var3
# Ha: the mean of at least one of the groups is different from others
#

```

```

➡ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   species         344 non-null    object
 1   island          344 non-null    object
 2   bill_length_mm  342 non-null    float64
 3   bill_depth_mm   342 non-null    float64
 4   flipper_length_mm 342 non-null    float64
 5   body_mass_g     342 non-null    float64
 6   sex             333 non-null    object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB

```