T- Test :- A t-test is a type of inferential statistic which is used to determine if there is a significant difference between the means of two groups which may be related in certain features.

```
from scipy.stats import ttest_1samp
import numpy as np
import pandas as pd
ages = pd.read_csv('/content/ages.csv')
print(ages)
ages_mean = np.mean(ages)
print(ages_mean)
tset, pval = ttest_1samp(ages, 30)
print("p-values",pval)
if pval.any() < 0.05:    # alpha value is 0.05 or 5%
   print("we are rejecting null hypothesis")
else:
  print("we are accepting null hypothesis")
```

```
     n  age
0    1   20
1    2   99
2    3   42
3    4   79
4    5   66
5    6   55
6    7   74
7    8   39
8    9   33
9   10   81
10  11   36
11  12   27
12  13   95
13  14   50
14  15   15
15  16   25
16  17   84
17  18   56
18  19   50
19  20   98
20  21   20
21  22   59
22  23   96
23  24   72
24  25   67
25  26   39
26  27   34
27  28   11
28  29   43
29  30   28
n       15.5
age     53.1
dtype: float64
```

```
p-values [6.46799481e-10 5.02652426e-05]
we are accepting null hypothesis
```

Two sampled T-test :-The Independent Samples t Test or 2-sample t-test compares the means of two independent groups in order to determine whether there is statistical evidence that the associated population means are significantly different. The Independent Samples t Test is a parametric test. This test is also known as: Independent t Test.

```
from scipy.stats import ttest_ind
import numpy as np
import pandas as pd
from pandas import DataFrame

sales = pd.read_csv('/content/weeks.csv')
df=DataFrame(data=sales, columns=['week1','week2'])
df
```

	week1	week2
0	144	2663
1	102	2461
2	1781	1046
3	2749	759
4	3222	4362
5	1907	2422
6	4968	2480
7	1996	1576
8	1963	602
9	1971	2049
10	2207	4775
11	3377	776

```
week1_mean = np.mean(sales['week1'])
week2_mean = np.mean(sales['week2'])
print("week1 mean value:",week1_mean)
print("week2 mean value:",week2_mean)
week1_std = np.std(sales['week1'])
week2_std = np.std(sales['week2'])
print("week1 std value:",week1_std)
print("week2 std value:",week2_std)
ttest,pval = ttest_ind(sales['week1'],sales['week2'])
print("p-value",pval)
if pval <0.05:
  print("we reject null hypothesis")
else:
  print("we accept null hypothesis")
```

```
week1 mean value: 2221.633333333333
week2 mean value: 2633.9
week1 std value: 1486.1584366711677
week2 std value: 1412.0187758430598
p-value 0.2832938455413314
we accept null hypothesis
```

Paired sampled t-test: The paired sample t-test is also called dependent sample t-test. It’s an univariate test designed to identify a significant difference between 2 related variables.

27 2067 4059

```
import pandas as pd
from scipy import stats
df = pd.read_csv("/content/blood_pressure.csv", encoding='utf-8')
```

```
df[['bp_before','bp_after']].describe()
ttest,pval = stats.ttest_rel(df['bp_before'], df['bp_after'])
print(pval)
if pval<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

```
0.0011297914644840823
reject null hypothesis
```

Z Test

Several different types of tests are used in statistics (i.e. f test, chi square test, t test). You would use a Z test if:

- Your sample size is greater than 30. Otherwise, use a t test.
- Data points should be independent from each other. In other words, one data point is not related or does not affect another data point.
- Your data should be normally distributed. However, for large sample sizes (over 30) this doesn’t always matter.
- Your data should be randomly selected from a population, where each item has an equal chance of being selected.
- Sample sizes should be equal if at all possible.

```
import pandas as pd
from scipy import stats
from statsmodels.stats import weightstats as stests
ztest ,pval = stests.ztest(df['bp_before'], x2=None, value=156)
print(float(pval))
if pval<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

```
0.6651614730255063
accept null hypothesis
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureW
  import pandas.util.testing as tm
```

Two-sample Z test

In two sample z-test, similar to t-test, we are checking two independent data groups and deciding whether the sample mean of two group is equal or not.

- H0 : mean of two group is 0
- H1 : mean of two group is not 0

```
ztest ,pval1 = stests.ztest(df['bp_before'], x2=df['bp_after'], value=0,alternati
print(float(pval1))
if pval<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

```
0.002162306611369422
accept null hypothesis
```

ANOVA (F-TEST) :

- The t-test works well when dealing with two groups, but sometimes we want to compare more than two groups at the same time. For example, if we wanted to test whether voter age differs based on some categorical variable like race, we have to compare the means of each level or group the variable. We could carry out a separate t-test for each pair of groups, but when you conduct many tests you increase the chances of false positives. The analysis of variance or ANOVA is a statistical inference test that lets you compare multiple groups at the same time.
- F = Between group variability / Within group variability

One Way F-test (ANOVA) :

- It determines whether two or more groups are similar or not based on their mean similarity and f-score.

```
df_anova = pd.read_csv('/content/plant_growth.csv')
df_anova = df_anova[['weight','group']]
grps = pd.unique(df_anova.group.values)
d_data = {grp:df_anova['weight'][df_anova.group == grp] for grp in grps}

F, p = stats.f_oneway(d_data['ctrl'], d_data['trt1'], d_data['trt2'])
print("p-value for significance is: ", p)
if p<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

```
p-value for significance is:  0.0159099583256229
reject null hypothesis
```

Two Way F-test:

- Two way F-test is an extension of one-way F-test.
- It is used when we have two independent variables and 2+ groups.

- A two-way F-test does not tell which variable is dominant.
- If we need to check individual significance, then Post-hoc testing needs to be performed.

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
df_anova2 = pd.read_csv("https://raw.githubusercontent.com/Opensourcefordatascier
model = ols('Yield ~ C(Fert)*C(Water)', df_anova2).fit()
print(f"Overall model F({model.df_model: .0f},{model.df_resid: .0f}) = {model.fva
res = sm.stats.anova_lm(model, typ= 2)
res
```

```
Overall model F( 3, 16) =  4.112, p =  0.0243
```

	sum_sq	df	F	PR(>F)
C(Fert)	69.192	1.0	5.766000	0.028847
C(Water)	63.368	1.0	5.280667	0.035386
C(Fert):C(Water)	15.488	1.0	1.290667	0.272656
Residual	192.000	16.0	NaN	NaN

Chi-Square Test:

- The test is applied when you have two categorical variables from a single population.
- It is used to determine whether there is a significant association between the two variables.
- For example, in an election survey, voters might be classified by gender (male or female) and voting preference (Democrat, Republican, or Independent).
- We could use a chi-square test for independence to determine whether gender is related to voting preference.

```
df_chi = pd.read_csv('/content/chi_square_test.csv')
df_chi
```

	Gender	Like Shopping?
0	Male	No
1	Female	Yes

```
contingency_table=pd.crosstab(df_chi["Gender"],df_chi["Like Shopping?"])
print('contingency_table :-\n',contingency_table)
#Observed Values
Observed_Values = contingency_table.values
print("Observed Values :-\n",Observed_Values)
b=stats.chi2_contingency(contingency_table)
Expected_Values = b[3]
print("Expected Values :-\n",Expected_Values)
no_of_rows=len(contingency_table.iloc[0:2,0])
no_of_columns=len(contingency_table.iloc[0,0:2])
ddof=(no_of_rows-1)*(no_of_columns-1)
print("Degree of Freedom:-",ddof)
alpha = 0.05
from scipy.stats import chi2
chi_square=sum([(o-e)**2./e for o,e in zip(Observed_Values,Expected_Values)])
chi_square_statistic=chi_square[0]+chi_square[1]
print("chi-square statistic:-",chi_square_statistic)
critical_value=chi2.ppf(q=1-alpha,df=ddof)
print('critical_value:',critical_value)
#p-value
p_value=1-chi2.cdf(x=chi_square_statistic,df=ddof)
print('p-value:',p_value)
print('Significance level: ',alpha)
print('Degree of Freedom: ',ddof)
print('chi-square statistic:',chi_square_statistic)
print('critical_value:',critical_value)
print('p-value:',p_value)
if chi_square_statistic>=critical_value:
    print("Reject H0,There is a relationship between 2 categorical variables")
else:
    print("Retain H0,There is no relationship between 2 categorical variables")

if p_value<=alpha:
    print("Reject H0,There is a relationship between 2 categorical variables")
else:
    print("Retain H0,There is no relationship between 2 categorical variables")
```

```
contingency_table :-
 Like Shopping?  No  Yes
Gender
Female           2    3
Male             2    2
Observed Values :-
 [[2 3]
 [2 2]]
Expected Values :-
 [[2.22222222 2.77777778]
```

```
 [1.77777778 2.22222222]]
Degree of Freedom:- 1
chi-square statistic:- 0.09000000000000008
critical_value: 3.841458820694124
p-value: 0.7641771556220945
Significance level:  0.05
Degree of Freedom:  1
chi-square statistic: 0.09000000000000008
critical_value: 3.841458820694124
p-value: 0.7641771556220945
Retain H0,There is no relationship between 2 categorical variables
Retain H0,There is no relationship between 2 categorical variables
```

✓ 0s completed at 4:17 PM