# Lab Plan 6: Hypothesis Testing in Python

Let’s see some of widely used hypothesis testing type :-

1. T Test ( Student T test)
2. Z Test
3. ANOVA Test
4. Chi-Square Test

**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. It is mostly used when the data sets, like the set of data recorded as outcome from flipping a coin a 100 times, would follow a normal distribution and may have unknown [variances](https://www.investopedia.com/terms/v/variance.asp). T test is used as a [hypothesis testing](https://www.investopedia.com/terms/h/hypothesistesting.asp) tool, which allows testing of an assumption applicable to a population.

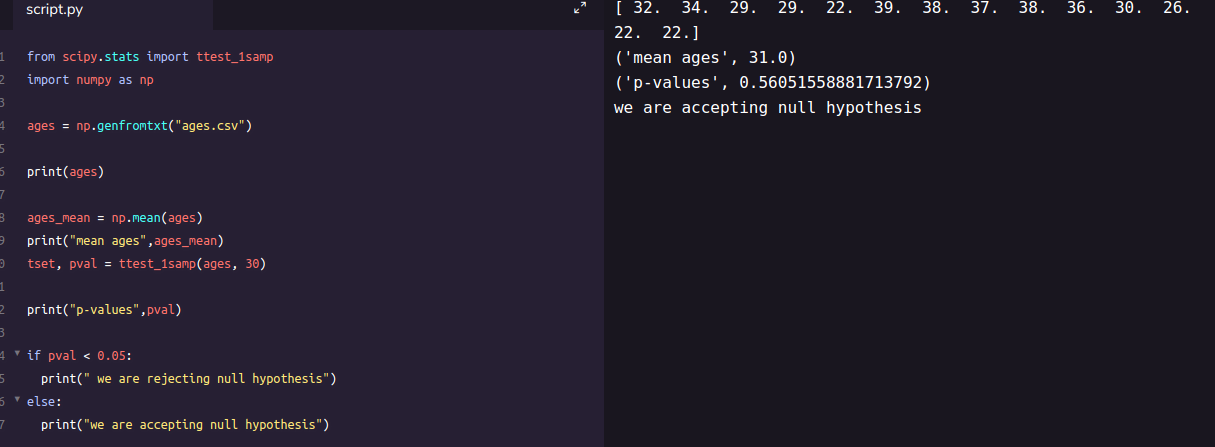
T-test has 2 types : 1. one sampled t-test 2. two-sampled t-test.

**One sample t-test** : The One Sample *t* Test determines whether the sample mean is statistically different from a known or hypothesised population mean. The One Sample *t* Test is a parametric test.

Example :- you have 10 ages and you are checking whether avg age is 30 or not. (check code below for that using python)

from scipy.stats import ttest\_1samp  
import numpy as npages = np.genfromtxt(“ages.csv”)print(ages)ages\_mean = np.mean(ages)  
print(ages\_mean)  
tset, pval = ttest\_1samp(ages, 30)print(“p-values”,pval)if pval < 0.05: # alpha value is 0.05 or 5%  
 print(" we are rejecting null hypothesis")  
else:  
 print("we are accepting null hypothesis")

Output for above code is :

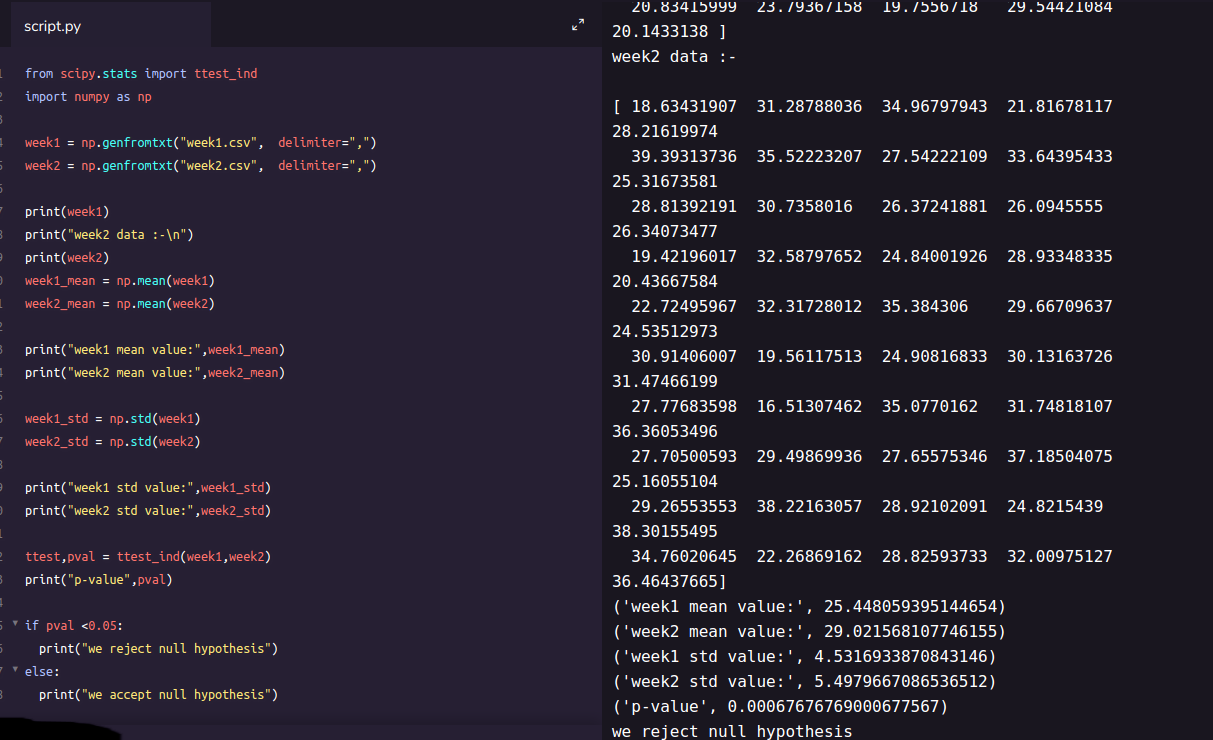


one-sample t-test output

**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**.

Example : is there any association between week1 and week2 ( code is given below in python)

from scipy.stats import ttest\_ind  
import numpy as npweek1 = np.genfromtxt("week1.csv", delimiter=",")  
week2 = np.genfromtxt("week2.csv", delimiter=",")print(week1)  
print("week2 data :-\n")  
print(week2)  
week1\_mean = np.mean(week1)  
week2\_mean = np.mean(week2)print("week1 mean value:",week1\_mean)  
print("week2 mean value:",week2\_mean)week1\_std = np.std(week1)  
week2\_std = np.std(week2)print("week1 std value:",week1\_std)  
print("week2 std value:",week2\_std)ttest,pval = ttest\_ind(week1,week2)  
print("p-value",pval)if pval <0.05:  
 print("we reject null hypothesis")  
else:  
 print("we accept null hypothesis")



2-sampled t-test output

**Paired sampled t-test :-**The paired sample t-test is also called dependent sample t-test. It’s an uni variate test that tests for a significant difference between 2 related variables. An example of this is if you where to collect the blood pressure for an individual before and after some treatment, condition, or time point.

**H0 :- means difference between two sample is 0**

**H1:- mean difference between two sample is not 0**

check the code below for same

import pandas as pd  
from scipy import stats  
df = pd.read\_csv("blood\_pressure.csv")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")

When you can run a Z Test.

Several different types of tests are used in statistics (i.e. [f test](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/hypothesis-testing/f-test/), [chi square test](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/chi-square/), [t test](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/t-test/)). You would use a Z test if:

* Your [sample size](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/find-sample-size/) is greater than 30. Otherwise, use a [t test](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/t-test/).
* Data points should be [independent](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/dependent-events-independent/#or)from each other. In other words, one data point isn’t related or doesn’t 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](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/find-sample-size/) should be equal if at all possible.

Example again we are using z-test for blood pressure with some mean like 156 (python code is below for same) **one-sample Z test.**

import pandas as pd  
from scipy import stats  
from statsmodels.stats import weightstats as stestsztest ,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")

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

**H0 : mean of two group is 0**

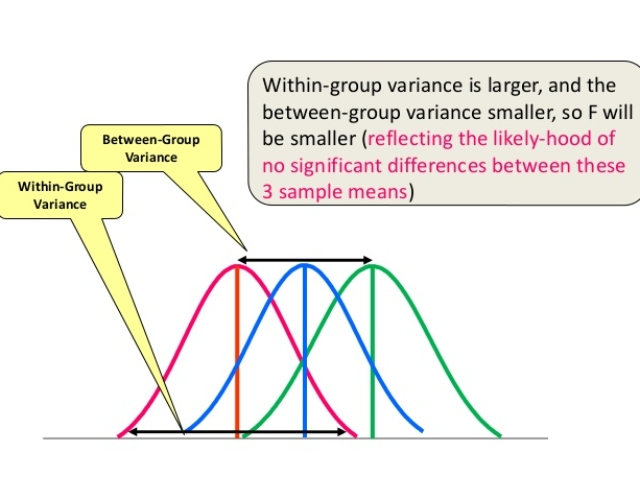
**H1 : mean of two group is not 0**

Example : we are checking in blood data after blood and before blood data.(code in python below)

ztest ,pval1 = stests.ztest(df['bp\_before'], x2=df['bp\_after'], value=0,alternative='two-sided')  
print(float(pval1))if pval<0.05:  
 print("reject null hypothesis")  
else:  
 print("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](https://en.wikipedia.org/wiki/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**



F-Test or Anova concept image

Unlike the z and t-distributions, the F-distribution does not have any negative values because between and within-group variability are always positive due to squaring each deviation.

**One Way F-test(Anova) :-**It tell whether two or more groups are similar or not based on their mean similarity and f-score.

Example : there are 3 different category of plant and their weight and need to check whether all 3 group are similar or not (code in python below)

df\_anova = pd.read\_csv('PlantGrowth.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")

**Two Way F-test :-** Two way F-test is extension of 1-way f-test, it is used when we have 2 independent variable and 2+ groups. 2-way F-test does not tell which variable is dominant. if we need to check individual significance then **Post-hoc** testing need to be performed.

Now let’s take a look at the Grand mean crop yield (the mean crop yield not by any sub-group), as well the mean crop yield by each factor, as well as by the factors grouped together

import statsmodels.api as sm  
from statsmodels.formula.api import olsdf\_anova2 = pd.read\_csv("<https://raw.githubusercontent.com/Opensourcefordatascience/Data-sets/master/crop_yield.csv>")model = ols('Yield ~ C(Fert)\*C(Water)', df\_anova2).fit()  
print(f"Overall model F({model.df\_model: .0f},{model.df\_resid: .0f}) = {model.fvalue: .3f}, p = {model.f\_pvalue: .4f}")res = sm.stats.anova\_lm(model, typ= 2)  
res

**Chi-Square Test-**The test is applied when you have two [categorical variables](https://stattrek.com/Help/Glossary.aspx?Target=Categorical%20variable) 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

check example in python below

df\_chi = pd.read\_csv('chi-test.csv')  
contingency\_table=pd.crosstab(df\_chi["Gender"],df\_chi["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 = 0.05from 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")

You can get all code in my [git](https://github.com/yug95/MachineLearning) repository.

ah, finally we came to end of this article. I hope this article would have helped. any feedback is always appreciated.

For more update check my [git](https://github.com/yug95/MachineLearning) and follow we on medium.