* Installing Python

1. Download and install
2. Setting environmental variable
3. Open terminal (e.i. cmd)
4. Type “Python”
5. “Exit” to quit

Or

1. Open terminal (e.i. cmd)
2. Python –version

* Install PIP

1. Download the *get-pip.py* script.
2. Compile it python *get-pip.py*.
3. Checking *pip --version*

* Install pandas: Pip install pandas
* Install Scipy: Pip install scipy
* Install notebook: Pip install notebook
* Install numpy: Pip install numpy
* Defining an Array

We can use built-in list in python. But all other modules were built using numpy array. So we should use numpy array.

To calculate mean of an array:

*import numpy as np*

*X = np.array([10, 12, 15, 11, 13])*

*Y = np.array([18, 20, 16, 19, 22])*

*mea=np.mean(X)*

*mea*

np.float64(12.2)

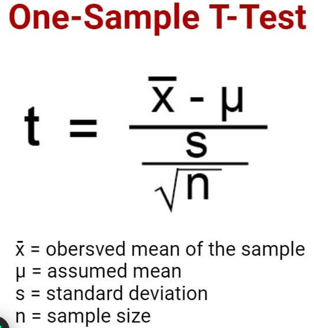
For standard Deviation: **std**, for variance: **var**

**T-test for a Single Mean**

What is Hypothesis? Why do we need to draw hypothesis?

If we guess that population average is 13 the this a hypothesis. In formal for we should write this hypothesis:

First gauss is call null hypothesis and second one is called alternative hypothesis. Now our duty is to test which one is true. If hull hypothesis rejected then alternative hypothesis is accepted. If null hypothesis is accepted then no need to think about alternative hypothesis. Formula is



*From scipy import stats  
# Hypothesized population mean  
population\_mean = 13  
  
# Perform one-sample t-test  
t\_statistic, p\_value = stats.ttest\_1samp(X, population\_mean)  
  
print("T-statistic:", t\_statistic)  
print("P-value:", p\_value)*

T-statistic: -0.929981109950555

P-value: 0.40502331430906013

P-value should be less than 0.05 to reject the null hypothesis. Since our p-value is 0.405 so we can not reject reject null hypothesis. That is population mean is 13 can considered.

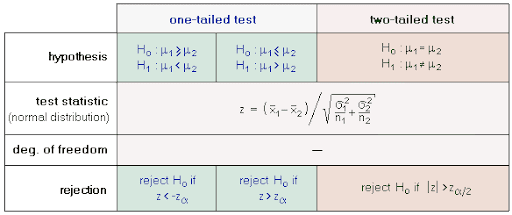
1. What is Hypothesis? Why do we need to draw hypothesis?

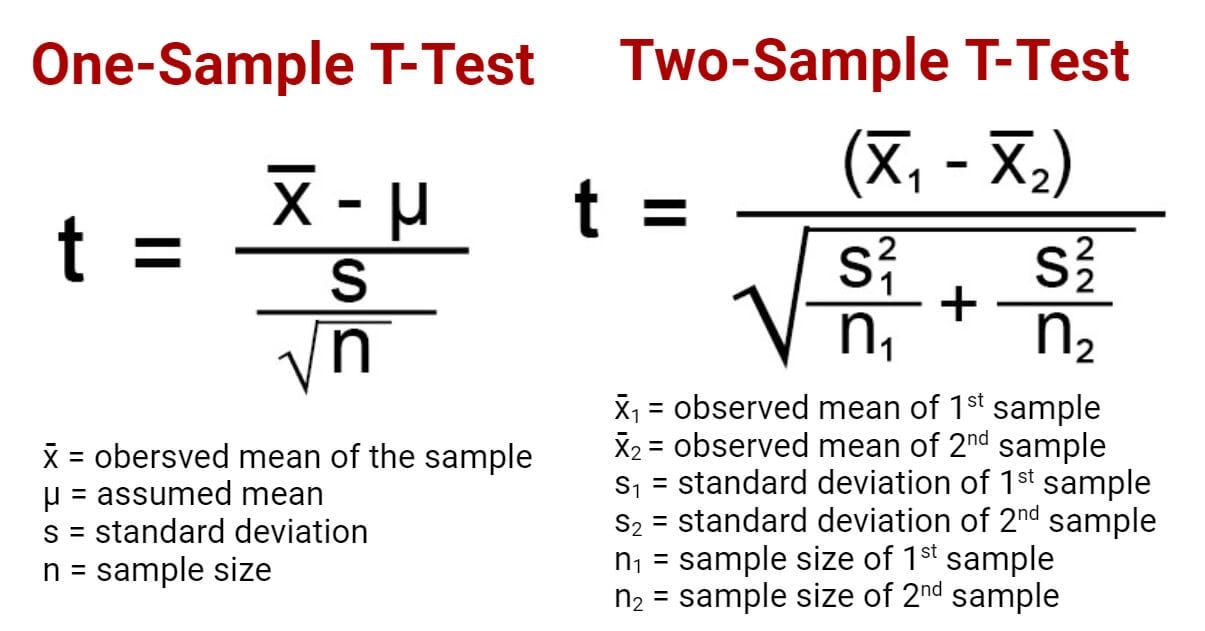
2. What do you mean by null hypothesis? When do we reject our null guess (null hypothesis).

3. what is your understanding on “T-test for single mean”?

**T-test for a comparison of the Mean of two groups**

If we need to compare i.e. test equivalency of mean of two groups then this test is called comparison of mean test. There test called t-test is used to in this regards. Lets have the concept of hypothesis in this this context.





*import numpy as np*

*from scipy import stats*

*X = np.array([10, 12, 15, 11, 13])*

*Y = np.array([18, 20, 16, 19, 22])*

T-statistic: -5.155066696336544

P-value: 0.0008689022188524722

Since p-value is less than 0.05 so null hypothesis is rejected. Their population average are not equal.

**T-test to compare of the Mean before and after of a group**

When we need to evaluate an effect of treatment that applied on the population then we use paired-sample t-test. There is a strict condition: measurement before applying treatment and after applying treatment have to be collected using same tools and same method.

*# Sample data for two related groups*

*before = np.array([15, 18, 20, 16, 19])*

*after = np.array([17, 21, 22, 18, 20])*

*# Perform paired samples t-test*

*t\_statistic, p\_value = stats.ttest\_rel(before, after)*

*print("T-statistic:", t\_statistic)*

*print("P-value:", p\_value)*

*T-statistic: -6.324555320336758*

*P-value: 0.0031982021523353082*

Since p-value is less than 0.05 so null hypothesis is rejected. There is a significant difference before and after. That is treatment was effective.

1. What is Hypothesis? Why do we need to draw hypothesis?

2. What do you mean by null hypothesis? When do we reject our null guess (null hypothesis).

3. what is your understanding on “T-test for single mean”?

4. When do we use paired Sample t-test?

Cross Tabulation

**import** **pandas** **as** **pd**

data = pd.DataFrame({

'Gender': ['Male', 'Female', 'Female', 'Male', 'Male'],

'Education': ['Graduate', 'Undergraduate', 'Undergraduate', 'Graduate', 'Graduate']

})

crosstab\_result = pd.crosstab(data['Gender'], data['Education'])

print(crosstab\_result)

crosstab\_with\_margins = pd.crosstab(data['Gender'], data['Education'], margins=**True**)

print(crosstab\_with\_margins)

import numpy as np

import numpy as np

cr\_tab = pd.crosstab(data['gender'], data['Refund'],values=data['income'], aggfunc=np.mean, margins=True)

print(cr\_tab)

**Chi-square Test**

from scipy.stats import chi2\_contingency

# defining the table

data = [[207, 282, 241], [234, 242, 232]]

stat, p, dof, expected = chi2\_contingency(data)

# interpret p-value

alpha = 0.05

print("p value is " + str(p))

if p <= alpha:

print('Dependent (reject H0)')

else:

print('Independent (H0 holds true)')

Form dataFrame

contigency\_pct = pd.crosstab(df['Gender'], df['isSmoker'], normalize='index')  
contigency\_pct

contigency\_pct = pd.crosstab(df['Gender'], df['isSmoker'])

stat, p, dof, expected = chi2\_contingency(contigency\_pct)