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

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

One-sample T-test with Python

The test will tell us whether means of the sample and the population are different

In [1]:

```
ages=[10,20,35,50,28,40,55,18,16,55,30,25,43,18,30,28,14,24,16,17,32,35,26,27,65,18,43,
23,21,20,19,70]
```

In [2]:

```
len(ages)
```

Out[2]:

```
32
```

In [3]:

```
import numpy as np
ages_mean=np.mean(ages)
print(ages_mean)
```

```
30.34375
```

In [4]:

```
sample_size=10
age_sample=np.random.choice(ages,sample_size)
```

In [5]:

```
age_sample
```

Out[5]:

```
array([30, 24, 30, 28, 20, 43, 55, 70, 70, 23])
```

In [6]:

```
from scipy.stats import ttest_1samp
```

In [7]:

```
ttest,p_value=ttest_1samp(age_sample,30)
```

In [8]:

```
print(p_value)
```

```
0.15997019001297158
```

In [9]:

```
if p_value < 0.05:    # alpha value is 0.05 or 5%
    print(" we are rejecting null hypothesis")
else:
    print("we are accepting null hypothesis")
```

```
we are accepting null hypothesis
```

Some More Examples

Consider the age of students in a college and in Class A

In [10]:

```
import numpy as np
import pandas as pd
import scipy.stats as stats
import math
np.random.seed(6)
school_ages=stats.poisson.rvs(loc=18,mu=35,size=1500)
classA_ages=stats.poisson.rvs(loc=18,mu=30,size=60)
```

In [11]:

```
classA_ages.mean()
```

Out[11]:

```
46.9
```

In [12]:

```
_,p_value=stats.ttest_1samp(a=classA_ages,popmean=school_ages.mean())
```

In [13]:

```
p_value
```

Out[13]:

```
1.139027071016194e-13
```

In [14]:

```
school_ages.mean()
```

Out[14]:

```
53.303333333333335
```

In [15]:

```
if p_value < 0.05:    # alpha value is 0.05 or 5%
    print(" we are rejecting null hypothesis")
else:
    print("we are accepting null hypothesis")
```

```
 we are rejecting null hypothesis
```

Two-sample T-test With Python

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

In [16]:

```
np.random.seed(12)
ClassB_ages=stats.poisson.rvs(loc=18,mu=33,size=60)
ClassB_ages.mean()
```

Out[16]:

```
50.63333333333333
```

In [18]:

```
_,p_value=stats.ttest_ind(a=classA_ages,b=ClassB_ages,equal_var=False)
```

In [19]:

```
if p_value < 0.05:    # alpha value is 0.05 or 5%
    print(" we are rejecting null hypothesis")
else:
    print("we are accepting null hypothesis")
```

```
 we are rejecting null hypothesis
```

Paired T-test With Python

When you want to check how different samples from the same group are, you can go for a paired T-test

In [20]:

```
weight1=[25,30,28,35,28,34,26,29,30,26,28,32,31,30,45]
weight2=weight1+stats.norm.rvs(scale=5,loc=-1.25,size=15)
```

In [21]:

```
print(weight1)
print(weight2)
```

```
[25, 30, 28, 35, 28, 34, 26, 29, 30, 26, 28, 32, 31, 30, 45]
[30.57926457 34.91022437 29.00444617 30.54295091 19.86201983 37.57873174
 18.3299827  21.3771395  36.36420881 32.05941216 26.93827982 29.519014
 26.42851213 30.50667769 41.32984284]
```

In [22]:

```
weight_df=pd.DataFrame({"weight_10":np.array(weight1),
                        "weight_20":np.array(weight2),
                       "weight_change":np.array(weight2)-np.array(weight1)})
```

In [23]:

```
weight_df
```

Out[23]:

	weight_10	weight_20	weight_change
0	25	30.579265	5.579265
1	30	34.910224	4.910224
2	28	29.004446	1.004446
3	35	30.542951	-4.457049
4	28	19.862020	-8.137980
5	34	37.578732	3.578732
6	26	18.329983	-7.670017
7	29	21.377139	-7.622861
8	30	36.364209	6.364209
9	26	32.059412	6.059412
10	28	26.938280	-1.061720
11	32	29.519014	-2.480986
12	31	26.428512	-4.571488
13	30	30.506678	0.506678
14	45	41.329843	-3.670157

In [24]:

```
_,p_value=stats.ttest_rel(a=weight1,b=weight2)
```

In [25]:

```
print(p_value)
```

```
0.5732936534411279
```

In [26]:

```
if p_value < 0.05:     # alpha value is 0.05 or 5%
    print(" we are rejecting null hypothesis")
else:
    print("we are accepting null hypothesis")
```

```
we are accepting null hypothesis
```

Correlation

In [27]:

```
import seaborn as sns
df=sns.load_dataset('iris')
```

In [28]:

```
df.shape
```

Out[28]:

```
(150, 5)
```

In [29]:

```
df.corr()
```

Out[29]:

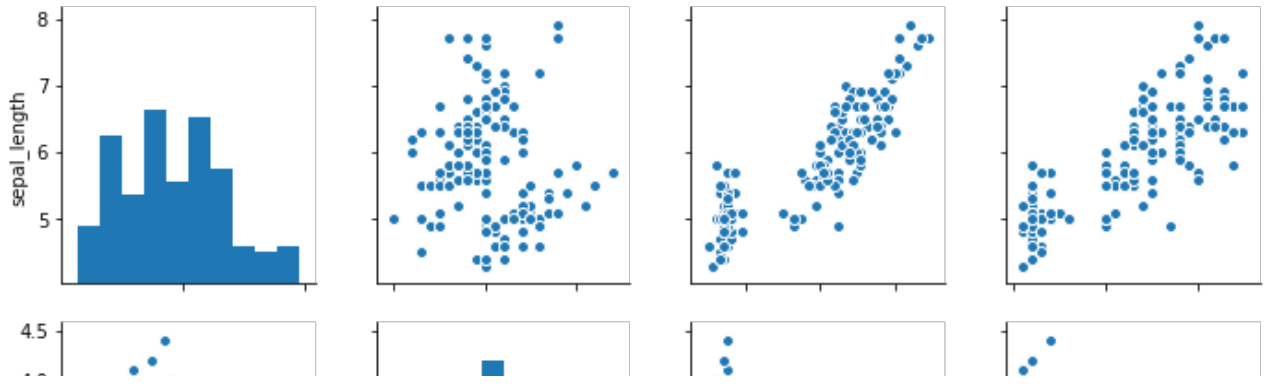
	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

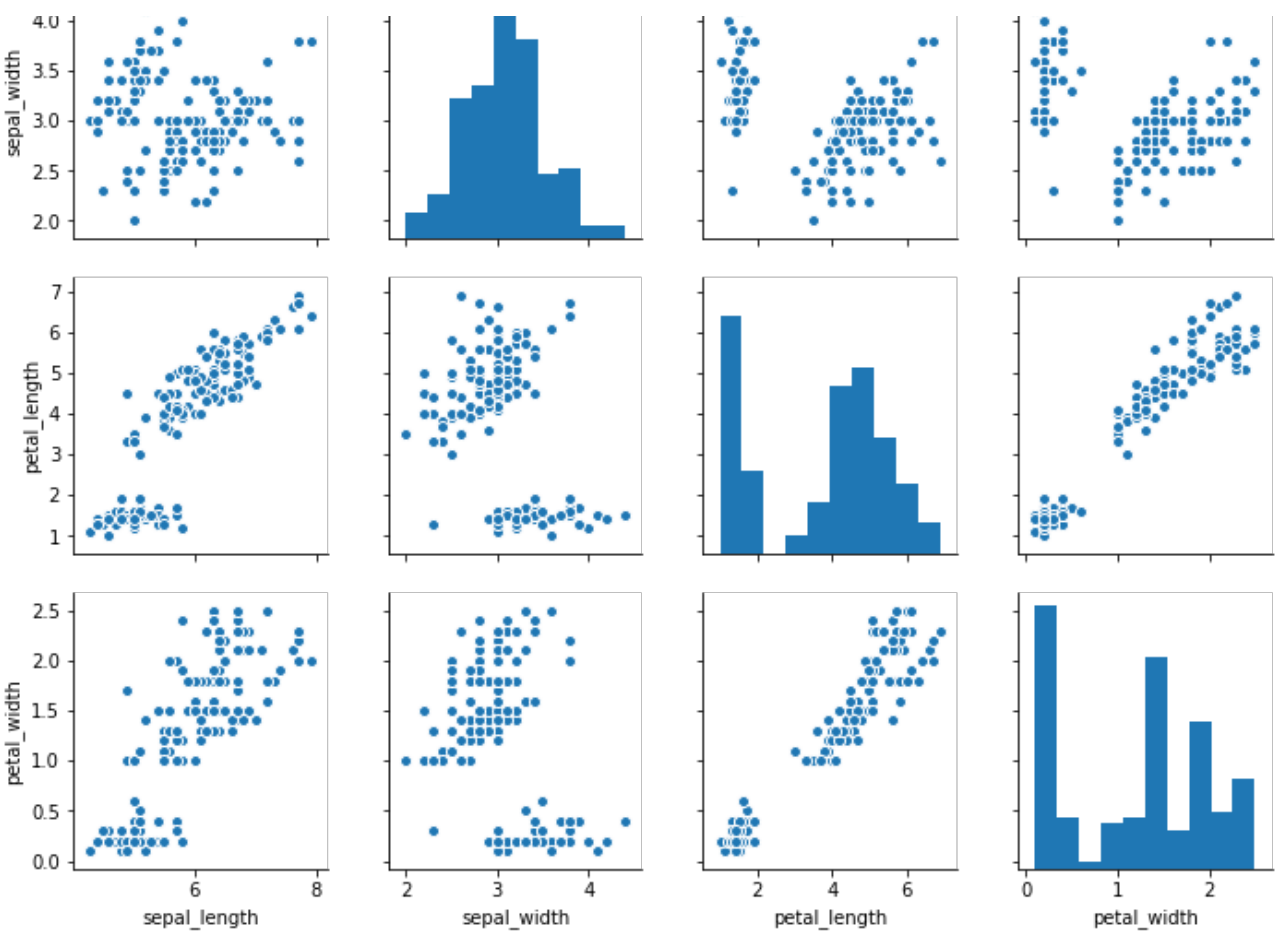
In [30]:

```
sns.pairplot(df)
```

Out[30]:

```
<seaborn.axisgrid.PairGrid at 0x2797d8f47b8>
```




sepal_width
petal_length
petal_width
sepal_length
sepal_width
petal_length
petal_width

In [ ]: