02_ttests

April 1, 2022

1 Student's t-test:

1. One-sample student's t-test Test a sample with a known standard value.

Assumptions - Observations in each sample are independent and identically distributed. - Observations in each sample are normally distributed. **Interpretation**
H0: the means of the samples are equal to the known value.
H1: the means of the samples are unequal to the known value.

[ ]:
```
# one sample t-test
# import libraries

import seaborn as sns
import pandas as pd
from scipy.stats import ttest_1samp

# load dataset

df = sns.load_dataset('titanic')
```

[ ]:
```
df.head()
```

[ ]:
```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

[ ]:

```
df1 = df[['sex', 'age','fare']]
df1.head()
```

[ ]:

```
      sex   age     fare
0    male  22.0   7.2500
1  female  38.0  71.2833
2  female  26.0   7.9250
3  female  35.0  53.1000
4    male  35.0   8.0500
```

[ ]:

```
#cdescription
df1.describe()
```

[ ]:

```
              age        fare
count  714.000000  891.000000
mean    29.699118   32.204208
std     14.526497   49.693429
min      0.420000    0.000000
25%     20.125000    7.910400
50%     28.000000   14.454200
75%     38.000000   31.000000
max     80.000000  512.329200
```

[ ]:

```
# check the age and compare witht a known value of 45 years

ttest_1samp(df1['fare'], 50)

stat, p = ttest_1samp(df1['fare'], 50)

print('stat=%.3f, p=%.3f' % (stat, p))

# make a conditional arguement for ease
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different Distribution')
```

```
stat=-10.689, p=0.000
Probably different Distribution
```

1.1 Two sample t-test

Independent student's t-test

Assumptions - Observations in each sample are independent and identically distributed. - Observations in each sample are normally distributed. - Observations in each sample have the same variance.

Interpretation

H0: the means of the samples are equal.
H1: the means of the samples are unequal.

[ ]:

```
# we will compare

#splitting dataset
df_male = df1.loc[df1['sex']== 'male']
df_female = df1.loc[df1['sex']== 'female']

# library
from scipy.stats import ttest_ind
stat, p = ttest_ind(df_male['fare'], df_female['fare'])

print('stat=%.3f, p=%.3f' % (stat, p))

# make a conditional arguement for ease
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different Distribution')
```

```
stat=-5.529, p=0.000
Probably different Distribution
```

[ ]:

```
df_female.describe()
```

[ ]:

```
              age        fare
count  261.000000  314.000000
mean    27.915709   44.479818
std     14.110146   57.997698
min      0.750000    6.750000
25%     18.000000   12.071875
50%     27.000000   23.000000
75%     37.000000   55.000000
max     63.000000  512.329200
```

Paired student's t-test Tests whether the means of two paired samples are significantly different.

Assumptions - Observations in each sample are independent and identically distributed. - Observations in each sample are normally distributed. - Observations in each sample have the same variance. - Observations across each sample are paired.
Interpretation
H0: the means of the samples are equal.
H1: the means of the samples are unequal.

[ ]:

```
df.head()
```

```
[ ]:    survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
     0         0       3    male  22.0      1      0   7.2500        S  Third
     1         1       1  female  38.0      1      0  71.2833        C  First
     2         1       3  female  26.0      0      0   7.9250        S  Third
     3         1       1  female  35.0      1      0  53.1000        S  First
     4         0       3    male  35.0      0      0   8.0500        S  Third

          who  adult_male deck  embark_town alive  alone
     0    man        True  NaN  Southampton    no  False
     1  woman       False    C    Cherbourg   yes  False
     2  woman       False  NaN  Southampton   yes   True
     3  woman       False    C  Southampton   yes  False
     4    man        True  NaN  Southampton    no   True
```

```
[ ]: #select only male's date
     df_m = df.loc[df['sex']== 'male']
     df_m.head()
```

```
[ ]:    survived  pclass   sex   age  sibsp  parch     fare embarked  class    who  \
     0         0       3  male  22.0      1      0   7.2500        S  Third    man
     4         0       3  male  35.0      0      0   8.0500        S  Third    man
     5         0       3  male   NaN      0      0   8.4583        Q  Third    man
     6         0       1  male  54.0      0      0  51.8625        S  First    man
     7         0       3  male   2.0      3      1  21.0750        S  Third  child

        adult_male deck  embark_town alive  alone
     0        True  NaN  Southampton    no  False
     4        True  NaN  Southampton    no   True
     5        True  NaN   Queenstown    no   True
     6        True    E  Southampton    no   True
     7       False  NaN  Southampton    no  False
```

```
[ ]: df.head()
```

```
[ ]:    survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
     0         0       3    male  22.0      1      0   7.2500        S  Third
     1         1       1  female  38.0      1      0  71.2833        C  First
     2         1       3  female  26.0      0      0   7.9250        S  Third
     3         1       1  female  35.0      1      0  53.1000        S  First
     4         0       3    male  35.0      0      0   8.0500        S  Third

          who  adult_male deck  embark_town alive  alone
     0    man        True  NaN  Southampton    no  False
     1  woman       False    C    Cherbourg   yes  False
     2  woman       False  NaN  Southampton   yes   True
     3  woman       False    C  Southampton   yes  False
     4    man        True  NaN  Southampton    no   True
```

[ ]:

```
#select_ only teo classes
df_male_first = df_m.loc[df_m['class']== 'First']
df_male_second = df_m.loc[df_m['class']== 'Second']
df_male_third = df_m.loc[df_m['class']== 'Third']
```

[ ]:

```
# check our data
df_male_first.head()
df_male_first.describe()
```

[ ]:

```
         survived  pclass         age       sibsp       parch        fare
count  122.000000   122.0  101.000000  122.000000  122.000000  122.000000
mean     0.368852     1.0   41.281386    0.311475    0.278689   67.226127
std      0.484484     0.0   15.139570    0.546695    0.658853   77.548021
min      0.000000     1.0    0.920000    0.000000    0.000000    0.000000
25%      0.000000     1.0   30.000000    0.000000    0.000000   27.728100
50%      0.000000     1.0   40.000000    0.000000    0.000000   41.262500
75%      1.000000     1.0   51.000000    1.000000    0.000000   78.459375
max      1.000000     1.0   80.000000    3.000000    4.000000  512.329200
```

[ ]:

```
df_1st = df_male_first.sample(n=100)
df_2nd = df_male_first.sample(n=100)
print("The numerb of instances in 2st classs are ", df_1st.shape)
print("The numerb of instances in 2st classs are ", df_2nd.shape)
```

```
The numerb of instances in 2st classs are  (100, 15)
The numerb of instances in 2st classs are  (100, 15)
```

[ ]:

```
# import librarry
from scipy.stats import ttest_rel

#apply test to comapre classs one -1 and class-3
stat, p = ttest_rel(df_1st['age'], df_2nd['age'])
print('stat%.3f. p=%.3f' % (stat, p))
# make a conditional arguement for ease
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different Distribution')
```

```
statnan. p=nan
Probably different Distribution
```