AN7914 Week 10 Python

April 2, 2024

1 Week 10

[1]:
```
import pandas as pd
```

1.1 Intro to Hypothesis Testing in Python

Now we turn our attention to hypothesis testing in Python. For this we will be using `Scipy` library. `Scipy` is library for Scientific Computing and Stastics. Make sure this is installed in your lapop. If it is not installed go to **terminal** or **command prompt** and type `pip install scipy`

[2]:
```
import numpy as np
from scipy.stats import ttest_ind
```

We imported `numpy`. Notice we only imported `ttest_ind` from scipy for the time being. We will import other methods as we go along. For more info on all the stats that scipy does see the following link: https://docs.scipy.org/doc/scipy/reference/stats.html

Let's now create two pandas series object.

[3]:
```
class1=pd.Series([80,90,78,60,20,21,60,71])
class2=pd.Series([82,87,81,68,26,21,60,76])
```

Use `describe()` method to see the descriptive stats.

[4]:
```
class1.describe()
```

[4]:
```
count     8.000000
mean     60.000000
std      26.365562
min      20.000000
25%      50.250000
50%      65.500000
75%      78.500000
max      90.000000
dtype: float64
```

[5]:
```
class2.describe()
```

[5]:
```
count     8.000000
mean     62.625000
std      25.623301
min      21.000000
25%      51.500000
50%      72.000000
75%      81.250000
max      87.000000
dtype: float64
```

[6]:
```
ttest_ind(class1,class2)
```

[6]:
```
Ttest_indResult(statistic=-0.20194575014849533, pvalue=0.8428641579637326)
```

The mean score from the first class is 60 and the second class is 62. But is this difference statistically signifant? In order do check we need to do a *t-test*. - H0: the mean between two samples are equal . - H1: the mean between two samples are not equal.

[7]:
```
t_stat, p_value= ttest_ind(class1,class2)
print('t-stat', t_stat)
print ('p-value', p_value)
```

```
t-stat -0.20194575014849533
p-value 0.8428641579637326
```

Now we just used two pandas series `class1` and `class2` to do this test. We could have just as easily used two columns from the same dataframe. To illustrate let's create some fake dataframe with 50 rows. We will be using numpy's random method here. Notice that below we used `random.seed(10)`. This is just to help us create same random sequence of numbers every time we run this code.

[8]:
```
import random
random.seed(10)
df_50=pd.DataFrame({'A':np.random.randint(0,101,50),
                    'B': np.random.randint(0,101,50)})
df_50
```

[8]:

	A	B
0	57	32
1	31	10
2	92	12
3	93	71
4	79	38
5	70	78
6	32	4
7	34	1
8	53	98
9	38	0
10	22	18

```
11  12  89
12  85  49
13  15  68
14  62  48
15  17  22
16  23  15
17  47  27
18  98  79
19  77  77
20  74  72
21  48  46
22  91  27
23   3  29
24  27  63
25  33  58
26  66  89
27  61  91
28  15  54
29  93  72
30  42  90
31  15  78
32  25  40
33   6  33
34  91  86
35  66  24
36  27  61
37  82  59
38  24  76
39   3  37
40  97  87
41  92  74
42  85  64
43  32  36
44  14  75
45  57   6
46  85  77
47  86   1
48  70  30
49  28  84
```

[9]:
```
df_50['A'].mean()
```

[9]:
```
51.5
```

[10]:
```
df_50['B'].mean()
```

[10]:
```
51.1
```

[11]:
```
t_stat, p_value= ttest_ind(df_50['A'],df_50['B'])
print('t-stat', t_stat)
print ('p-value', p_value)
```

```
t-stat 0.0672940836792605
p-value 0.9464846896933561
```

How do you interpret this result? What happens when p-value is larger than a specificied significance level?

Now let's do a different kind of *t-test*. Let's test that average score is equal some number. For this we will
`import ttest_1samp`. This is a test for the null hypothesis that the expected value (mean) of a sample of independent observations a is equal to the given population mean, *popmean*.

[12]:
```
from scipy.stats import ttest_1samp
t_stat, p_value= ttest_1samp(df_50['A'],60)
print('t-stat', t_stat)
print ('p-value', p_value)
```

```
t-stat -1.987700027117136
p-value 0.05244805857031437
```

How do you intepret this result?