Introduction to Hypothesis Testing in Python

October 13, 2020

1 Hypothesis Testing

From lecture, we know that hypothesis testing is a critical tool in determing what the value of a parameter could be.

We know that the basis of our testing has two attributes:
Null Hypothesis: H_0
Alternative Hypothesis: H_a
The tests we have discussed in lecture are:

- One Population Proportion
- Difference in Population Proportions
- One Population Mean
- Difference in Population Means

In this tutorial, I will introduce some functions that are extremely useful when calculating a t-statistic and p-value for a hypothesis test.

Let's quickly review the following ways to calculate a test statistic for the tests listed above.
The equation is:

$$\frac{Best\ Estimate - Hypothesized\ Estimate}{Standard\ Error\ of\ Estimate}$$

We will use the examples from our lectures and use python functions to streamline our tests.

```
In [1]: import statsmodels.api as sm
        import numpy as np
        import pandas as pd
        import scipy.stats.distributions as dist
```

1.0.1 One Population Proportion

Research Question In previous years 52% of parents believed that electronics and social media was the cause of their teenager's lack of sleep. Do more parents today believe that their teenager's lack of sleep is caused due to electronics and social media?

Population: Parents with a teenager (age 13-18)
Parameter of Interest: p
Null Hypothesis: p = 0.52
Alternative Hypthosis: p > 0.52 (note that this is a one-sided test)
1018 Parents
56% believe that their teenager's lack of sleep is caused due to electronics and social media.

```
In [2]: n = 1018
        pnull = .52
        phat = .56 #population proportion
        sm.stats.proportions_ztest(phat * n, n, pnull, alternative='larger', prop_var=0.52)
```

```
Out[2]: (2.5545334262132955, 0.005316510991822442)
```

1.0.2 Difference in Population Proportions

Research Question Is there a significant difference between the population proportions of parents of black children and parents of Hispanic children who report that their child has had some swimming lessons?

Populations: All parents of black children age 6-18 and all parents of Hispanic children age 6-18
Parameter of Interest: p1 - p2, where p1 = black and p2 = hispanic
Null Hypothesis: p1 - p2 = 0
Alternative Hypthosis: p1 - p2 $\neq$ = 0

91 out of 247 (36.8%) sampled parents of black children report that their child has had some swimming lessons.

120 out of 308 (38.9%) sampled parents of Hispanic children report that their child has had some swimming lessons.

```
In [9]: n1 = 247
        p1 = .37

        n2 = 308
        p2 = .39

        population1 = np.random.binomial(1,p1,n1)
        population2 = np.random.binomial(1,p2,n2)

        #t test using statsmodels
        sm.stats.ttest_ind(population1, population2)
```

```
Out[9]: (-1.0463666097350164, 0.29584893903397824, 553.0)
```

```
In [11]: #null hypothesis
         p1-p2
```

```
Out[11]: -0.020000000000000018
```

In conclusion: the value of population 2 is greater than alpha value therefore we do not reject the null hypothesis in this case and we therefore assume that p1-p2 = 0 and there is no difference between black children and hispanic children in % having swimming lessons.

```
In [ ]: # This example implements the analysis from the "Difference in Two Proportions" lectur

        # Sample sizes
        n1 = 247
```

```
n2 = 308

# Number of parents reporting that their child had some swimming lessons
y1 = 91
y2 = 120

# Estimates of the population proportions
p1 = round(y1 / n1, 2)
p2 = round(y2 / n2, 2)

# Estimate of the combined population proportion
phat = (y1 + y2) / (n1 + n2)

# Estimate of the variance of the combined population proportion
va = phat * (1 - phat)

# Estimate of the standard error of the combined population proportion
se = np.sqrt(va * (1 / n1 + 1 / n2))

# Test statistic and its p-value
test_stat = (p1 - p2) / se
pvalue = 2*dist.norm.cdf(-np.abs(test_stat))

# Print the test statistic its p-value
print("Test Statistic")
print(round(test_stat, 2))

print("\nP-Value")
print(round(pvalue, 2))
```

1.0.3 One Population Mean

Research Question Is the average cartwheel distance (in inches) for adults more than 80 inches?
Population: All adults
Parameter of Interest: μ, population mean cartwheel distance. **Null Hypothesis:** $\mu = 80$ **Alternative Hypthosis:** $\mu > 80$
25 Adults
$\mu = 82.46$
$\sigma = 15.06$

```
In [12]: df = pd.read_csv("Cartwheeldata.csv")
         df.head()
```

```
Out[12]:    ID  Age Gender  GenderGroup Glasses  GlassesGroup  Height  Wingspan  \
         0   1   56      F            1       Y             1    62.0      61.0
         1   2   26      F            1       Y             1    62.0      60.0
         2   3   33      F            1       Y             1    66.0      64.0
         3   4   39      F            1       N             0    64.0      63.0
```

```
4   5   27      M            2       N            0    73.0      75.0

   CWDistance Complete  CompleteGroup  Score
0          79        Y              1      7
1          70        Y              1      8
2          85        Y              1      7
3          87        Y              1     10
4          72        N              0      4
```

```
In [13]: n = len(df)
         mean = df["CWDistance"].mean()
         sd = df["CWDistance"].std()
         (n, mean, sd)
```

```
Out[13]: (25, 82.48, 15.058552387264852)
```

```
In [15]: #calculate hypothesis and p statistic
         #value is null hypothesis
         sm.stats.ztest(df["CWDistance"], value = 80, alternative = "larger") #alternative to
```

```
Out[15]: (0.8234523266982029, 0.20512540845395266)
```

The p value is 0.205 so we can't reject the null hypothesis in this case.

1.0.4 Difference in Population Means

Research Question Considering adults in the NHANES data, do males have a significantly higher mean Body Mass Index than females?

Population: Adults in the NHANES data.
Parameter of Interest: $\mu_1 - \mu_2$, Body Mass Index.
Null Hypothesis: $\mu_1 = \mu_2$
Alternative Hypthosis: $\mu_1 \neq \mu_2$

2976 Females $\mu_1 = 29.94$
$\sigma_1 = 7.75$

2759 Male Adults
$\mu_2 = 28.78$
$\sigma_2 = 6.25$

$\mu_1 - \mu_2 = 1.16$

```
In [16]: url = "nhanes_2015_2016.csv"
         da = pd.read_csv(url)
         da.head()
```

```
Out[16]:     SEQN  ALQ101  ALQ110  ALQ130  SMQ020  RIAGENDR  RIDAGEYR  RIDRETH1  \
0  83732     1.0     NaN     1.0       1         1        62         3
1  83733     1.0     NaN     6.0       1         1        53         3
2  83734     1.0     NaN     NaN       1         1        78         3
3  83735     2.0     1.0     1.0       2         2        56         3
4  83736     2.0     1.0     1.0       2         2        42         4
```

```
   DMDCITZN  DMDEDUC2  ...  BPXSY2  BPXDI2  BMXWT  BMXHT  BMXBMI  BMXLEG  \
0       1.0       5.0  ...   124.0    64.0   94.8  184.5    27.8    43.3
1       2.0       3.0  ...   140.0    88.0   90.4  171.4    30.8    38.0
2       1.0       3.0  ...   132.0    44.0   83.4  170.1    28.8    35.6
3       1.0       5.0  ...   134.0    68.0  109.8  160.9    42.4    38.5
4       1.0       4.0  ...   114.0    54.0   55.2  164.9    20.3    37.4

   BMXARML  BMXARMC  BMXWAIST  HIQ210
0     43.6     35.9     101.1     2.0
1     40.0     33.2     107.9     NaN
2     37.0     31.0     116.5     2.0
3     37.7     38.3     110.1     2.0
4     36.0     27.2      80.4     2.0

[5 rows x 28 columns]
```

In [18]:
```
#seperate male vs. female populations
females = da[da["RIAGENDR"] == 2]
male = da[da["RIAGENDR"] == 1]
```

In [19]:
```
females.head()
```

Out[19]:
```
     SEQN  ALQ101  ALQ110  ALQ130  SMQ020  RIAGENDR  RIDAGEYR  RIDRETH1  \
3   83735     2.0     1.0     1.0       2         2        56         3
4   83736     2.0     1.0     1.0       2         2        42         4
5   83737     2.0     2.0     NaN       2         2        72         1
7   83742     1.0     NaN     1.0       2         2        32         1
12  83752     1.0     NaN     2.0       1         2        30         2

    DMDCITZN  DMDEDUC2  ...  BPXSY2  BPXDI2  BMXWT  BMXHT  BMXBMI  BMXLEG  \
3        1.0       5.0  ...   134.0    68.0  109.8  160.9    42.4    38.5
4        1.0       4.0  ...   114.0    54.0   55.2  164.9    20.3    37.4
5        2.0       2.0  ...   122.0    58.0   64.4  150.0    28.6    34.4
7        2.0       4.0  ...   114.0    70.0   64.5  151.3    28.2    34.1
12       1.0       4.0  ...   104.0    50.0   71.2  163.6    26.6    37.3

    BMXARML  BMXARMC  BMXWAIST  HIQ210
3      37.7     38.3     110.1     2.0
4      36.0     27.2      80.4     2.0
5      33.5     31.4      92.9     NaN
7      33.1     31.5      93.3     2.0
12     35.7     31.0      90.7     2.0

[5 rows x 28 columns]
```

In [21]:
```
male.head()
```

Out[21]:
```
     SEQN  ALQ101  ALQ110  ALQ130  SMQ020  RIAGENDR  RIDAGEYR  RIDRETH1  \
0   83732     1.0     NaN     1.0       1         1        62         3
```

```
1  83733     1.0     NaN     6.0       1         1        53         3
2  83734     1.0     NaN     NaN       1         1        78         3
6  83741     1.0     NaN     8.0       1         1        22         4
8  83743     NaN     NaN     NaN       2         1        18         5

   DMDCITZN  DMDEDUC2  ...  BPXSY2  BPXDI2  BMXWT  BMXHT  BMXBMI  BMXLEG  \
0       1.0       5.0  ...   124.0    64.0   94.8  184.5    27.8    43.3
1       2.0       3.0  ...   140.0    88.0   90.4  171.4    30.8    38.0
2       1.0       3.0  ...   132.0    44.0   83.4  170.1    28.8    35.6
6       1.0       4.0  ...   112.0    74.0   76.6  165.4    28.0    38.8
8       1.0       NaN  ...     NaN     NaN   72.4  166.1    26.2     NaN

   BMXARML  BMXARMC  BMXWAIST  HIQ210
0     43.6     35.9     101.1     2.0
1     40.0     33.2     107.9     NaN
2     37.0     31.0     116.5     2.0
6     38.0     34.0      86.6     NaN
8      NaN      NaN       NaN     2.0

[5 rows x 28 columns]
```

Calculate n, mean, std for both populations.

```
In [22]: n1 = len(females)
         mu1 = females["BMXBMI"].mean()
         sd1 = females["BMXBMI"].std()

         (n1, mu1, sd1)

Out[22]: (2976, 29.93994565217392, 7.753318809545674)

In [23]: n2 = len(male)
         mu2 = male["BMXBMI"].mean()
         sd2 = male["BMXBMI"].std()

         (n2, mu2, sd2)

Out[23]: (2759, 28.778072111846942, 6.2525676168014614)
```

Now we will perform the hypothesis test

```
In [24]: sm.stats.ztest(females["BMXBMI"].dropna(), male["BMXBMI"].dropna())

Out[24]: (6.1755933531383205, 6.591544431126401e-10)
```

Conclusion: Large test statistic and small p value. We can reject the null hypothesis that the BMI for both populations is the same and accept the alternative hypothesis that the BMIs are very different.