PYTHON FOR DATA SCIENCE

TABLE OF CONTENTS

- Introduction
  - Assumptions & Hypotheses
- Independent Sample t test with Python
  - ... using Researchpy
  - ... using Scipy.stats
- Assumption Check
- References

INDEPENDENT T-TEST

The indepentent T-test is a parametric test used to test for a statistically significant difference in the means between 2 groups. As with all parametric tests, there are certain conditions that need to be met in order for the test results to be considered reliable.

Parametric test assumptions

- Population distributions are normal
- Samples have equal variances
- The two samples are independent

Hypothesis

1. $H_0 : \mu_1 - \mu_2 \leq D_o$

2. $H_0 : \mu_1 - \mu_2 \geq D_o$
3. $H_0 : \mu_1 - \mu_2 = D_o$

1. $H_A : \mu_1 - \mu_2 > D_o$
2. $H_A : \mu_1 - \mu_2 < D_o$
3. $H_A : \mu_1 - \mu_2 \neq D_o$

Typically D_o is set to 0 and the 3rd hypothesis is being tested, i.e. there is no difference between the groups. The test statistic is the *t value* and can be calculated using the following formula:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - D_0}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Where s_p is the pooled standard deviation and is calculated as

$$s_p = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}}$$

One rejects the the null hypothesis, H_0, if the computed t value is greater than or less than the critical t value. The critical t value is determined by the degrees of freedom and alpha, α, value. Degrees of freedom is calculated as $df = n_1 + n_2 - 2$ and α is commonly set to 0.05. Reject H_0 if:

1. $t \geq t_\alpha$
2. $t \leq -t_\alpha$
3. $|t| \geq t_{\frac{\alpha}{2}}$

Before the decision to accept or reject the null hypothesis the assumptions need to be checked. See this page on how to check the parametric assumptions.

Fear not if math is not your strong suit. All this is being calucated when using the methods of a statistical software or programming language. It's good to know what is going on behind the scences. References for this section are provided at the end of the page.

T-TEST WITH PYTHON

Don't forget to check the assumptions before interpreting the results! First to load the libraries needed. This demonstration will include 2 ways to conduct an independent sample t-test in Python. One with Researchpy and the other with Scipy.stats.

```
import pandas as pd
import researchpy as rp
import scipy.stats as stats
```

Now to load the data set and take a high level look at the variables.

```
df = pd.read_csv("https://raw.githubusercontent.com/researchpy/Data-
sets/master/blood_pressure.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120 entries, 0 to 119
```

```
Data columns (total 5 columns):
patient      120 non-null int64
sex          120 non-null object
agegrp       120 non-null object
bp_before    120 non-null int64
bp_after     120 non-null int64
dtypes: int64(3), object(2)
memory usage: 4.8+ KB
```

INDEPDENT T-TEST USING RESEARCHPY

The method returns 2 data frames, one that contains the summary statistical information and the other that contains the statistical test information. If the returned data frames are not stored as a Python object then the output will be less clean than it can be since it will be displayed as a tuple - see below.

```
rp.ttest(group1= df['bp_after'][df['sex'] == 'Male'], group1_name= "Male",
         group2= df['bp_after'][df['sex'] == 'Female'], group2_name= "Female")
```

```
(   Variable      N        Mean         SD        SE   95% Conf.    Interval
 0      Male   60.0  155.516667  15.243217  1.967891  151.578926  159.454407
 1    Female   60.0  147.200000  11.742722  1.515979  144.166533  150.233467
 2  combined  120.0  151.358333  14.177622  1.294234  148.795621  153.921046,
              Independent t-test   results
 0  Difference (Male - Female) =     8.3167
 1          Degrees of freedom =   118.0000
 2                           t =     3.3480
 3       Two side test p value =     0.0011
 4      Difference > 0 p value =     0.9995
 5      Difference < 0 p value =     0.0005
 6                  Cohen's d =     0.6112
 7                  Hedge's g =     0.6074
 8              Glass's delta =     0.5456
 9                           r =     0.2945)
```

If stored as Python objects, they will be stored as Pandas data frames. This means that anything that can be done to a traditional Pandas data frame can be done to these results.

```
summary, results = rp.ttest(group1= df['bp_after'][df['sex'] == 'Male'], group1_name=
"Male",
                            group2= df['bp_after'][df['sex'] == 'Female'],
group2_name= "Female")
print(summary)
```

	Variable	N	Mean	SD	SE	95% Conf.	Interv
0	Male	60.0	155.516667	15.243217	1.967891	151.578926	159.454
1	Female	60.0	147.200000	11.742722	1.515979	144.166533	150.233
2	combined	120.0	151.358333	14.177622	1.294234	148.795621	153.921

```
print(results)
```

	Independent t-test	results
0	Difference (Male - Female) =	8.3167
1	Degrees of freedom =	118.0000
2	t =	3.3480
3	Two side test p value =	0.0011
4	Difference > 0 p value =	0.9995
5	Difference < 0 p value =	0.0005
6	Cohen's d =	0.6112
7	Hedge's g =	0.6074
8	Glass's delta =	0.5456
9	r =	0.2945

Before the results should be interpreted, the assumptions of the test should be checked. For example purposes, the results will be interpreted before checking the assumptions.

Interpretation

The average blood pressure after the treatment for males, M= 155.2 (151.6, 159.5), was statistically signigicantly higher than females, M= 147.2 (144.2, 150.2); t(118)= 3.3480, p= 0.001.

INDEPDENT T-TEST USING SCIPY.STATS

This method conducts the independent sample t-test and returns only the t test statistic and it's associated p-value. For more information about this method, please refer to the official documentation page.

```
stats.ttest_ind(df['bp_after'][df['sex'] == 'Male'],
                df['bp_after'][df['sex'] == 'Female'])
```

```
Ttest_indResult(statistic=3.3479506182111387, pvalue=0.0010930222986154283)
```

Interpretation

There is a statistically significant difference in the average post blood pressure between males and females, t= 3.3480, p= 0.001.

ASSUMPTION CHECK

The assumptions in this section need to be met in order for the test results to be considered valid. A more in-depth look at parametric assumptions is provided here, which includes some potential remedies.

THE TWO SAMPLES ARE INDEPENDENT

This assumption is tested when the study is designed. What this means is that no individual has data in group A and B.

POPULATION DISTRIBUTIONS ARE NORMAL

One of the assumptions is that the sampling distribution is normally distributed. This test of normality applies to the difference in values between the groups. One method for testing this assumption is the Shapiro-Wilk test. This can be completed using the shapiro() method from Scipy.stats.

```
sampling_difference = df['bp_after'][df['sex'] == 'Male'].values - \
                      df['bp_after'][df['sex'] == 'Female'].values

stats.shapiro(sampling_difference)
```

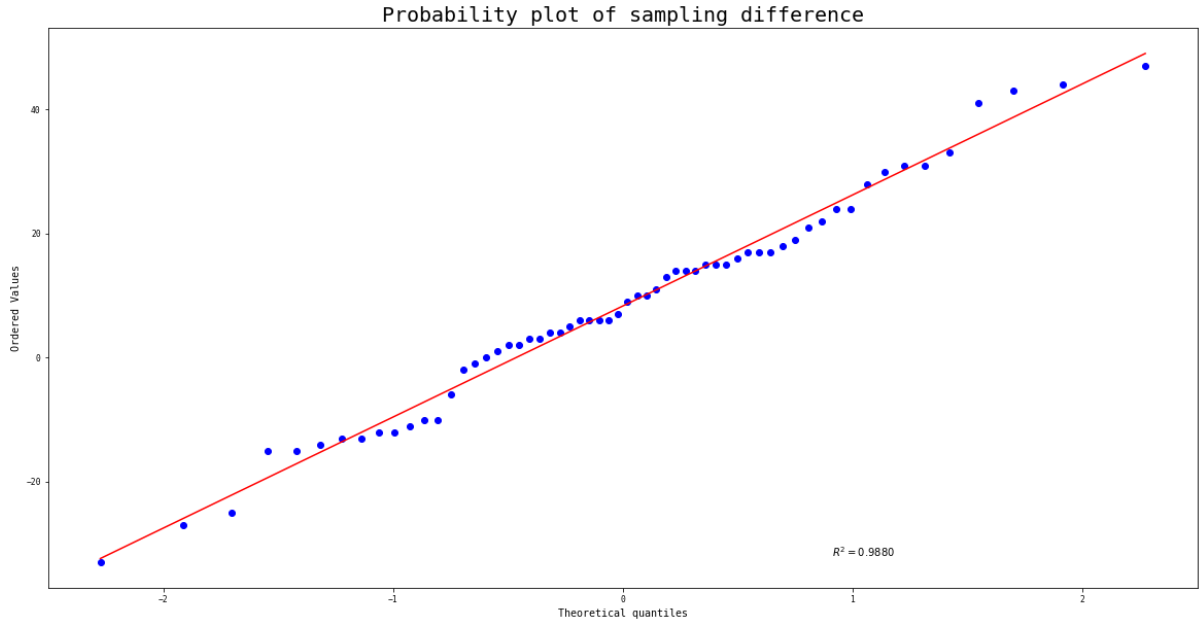
```
(0.98586106300354, 0.7147841453552246)
```

Unfortunately, the output is not labelled but is in the format of (W test statistic, p-value). The test is not significant which indicates the sampling distribution is normally distributed.

Another way to test the assumption is through a visual check- this is helpful when the sample is large. The reason this is true is that as the sample size increases, the statistical test's ability to reject the null hypothesis increases, i.e. it gains power to detect smaller differences as the sample size n increases.

One method of visually checking the distribution is to use a probability plot with or without the correlation value, R^2, to assess the observed values correlation with the theoretical distribution in question - in the current case it would be the Gaussian (a.k.a the normal) distribution. This can be completed by using the probplot() method from Scipy.stats.

```
fig = plt.figure(figsize= (20, 10))
ax = fig.add_subplot(111)

normality_plot, stat = stats.probplot(sampling_difference, plot= plt, rvalue= True)
ax.set_title("Probability plot of sampling difference", fontsize= 20)
ax.set

plt.show()
```



Both methods support the same finding which is that the sampling distribution is normally distributed.

HOMOGENEITY OF VARIANCE

One of the assumptions is that both groups have equal variances. One method for testing this assumption is the Levene's test of homogeneity of variances. This can be completed using the levene() method from Scipy.stats.

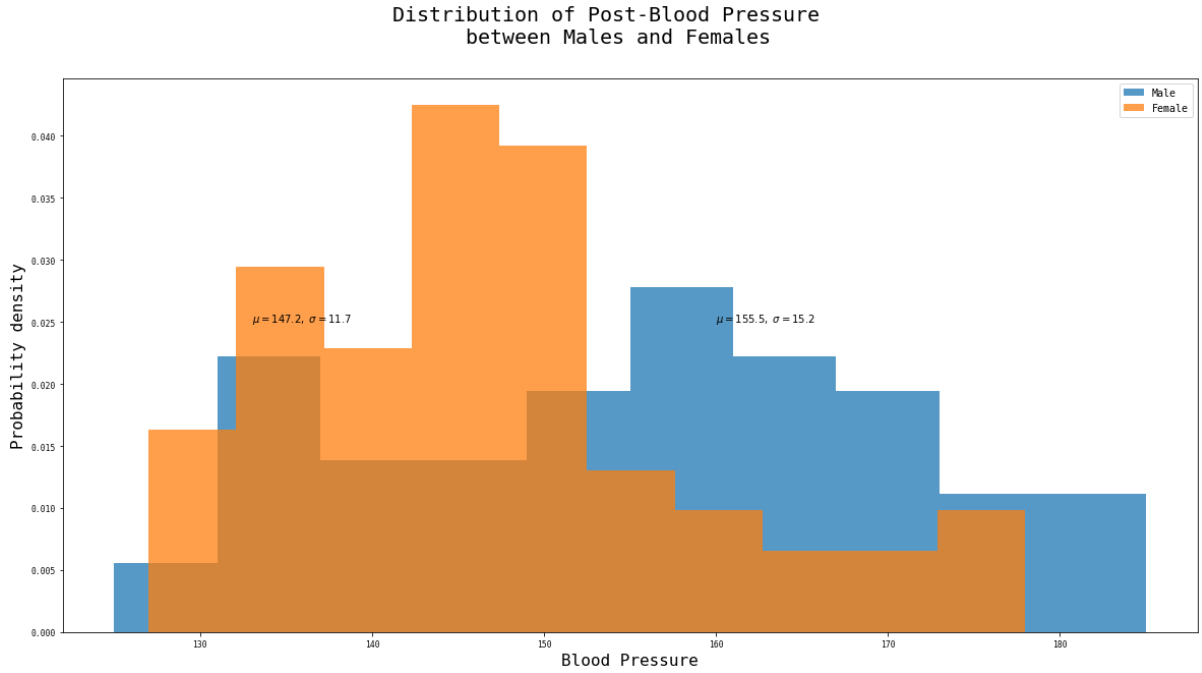
```
stats.levene(df['bp_after'][df['sex'] == 'Male'],
             df['bp_after'][df['sex'] == 'Female'],
             center= 'mean')
```

```
LeveneResult(statistic=5.865854141268659, pvalue=0.01695904277978066)
```

The test is significant which indicates the groups have a different amount of variation and that the t-test may not be the best statistical method to be used. Again, it may be worthwhile to check this assumption visually as well.

```
fig = plt.figure(figsize= (20, 10))
ax = fig.add_subplot(111)


p_bp_male = plt.hist(df['bp_after'][df['sex'] == 'Male'], label= "Male",
                     density= True,
                     alpha=0.75)
p_bp_female = plt.hist(df['bp_after'][df['sex'] == 'Female'], label= "Female",
                       density= True,
                       alpha=0.75)


plt.suptitle("Distribution of Post-Blood Pressure \n between Males and Females",
fontsize= 20)
plt.xlabel("Blood Pressure", fontsize= 16)
plt.ylabel("Probability density", fontsize= 16)

plt.text(133, .025,
         f"$\mu= {df['bp_after'][df['sex'] == 'Female'].mean(): .1f}, \ \sigma=
{df['bp_after'][df['sex'] == 'Female'].std(): .1f}$")
plt.text(160, .025,
         f"$\mu= {df['bp_after'][df['sex'] == 'Male'].mean(): .1f}, \ \sigma=
```

```
{df['bp_after'][df['sex'] == 'Male'].std(): .1f}$")

plt.show()
```



There are different ways to handle heteroskedasticity (unequal variance) and a decision needs to be made. Some options include, but is not limited to, transformming the dependent variable (outcome), could use trimmed means, robust standard errors, or use a parametric test suchs as the Welch's t-test.

REFERENCES

Kim T. K. (2015). T test as a parametric statistic. *Korean journal of anesthesiology, 68(6),* 540–546. doi:10.4097/kjae.2015.68.6.540

Rosner, B. (2015). *Fundamentals of Biostatistics* (8th ed.). Boston, MA: Cengage Learning.

Ott, R. L., and Longnecker, M. (2010). *An introduction to statistical methods and data analysis.* Belmon, CA: Brooks/Cole.

MATH SYMBOLS

Symbol	Meaning
n	Sample size
N	Population size
s^2	Sample variance
σ^2	Population variance
s	Sample standard deviation
σ	Population standard deviation
μ	Mean