[CC511] Homework 8 20204817 Federico Berto

November 5, 2020

1 Homework 8 - Federico Berto

[95]:
```
# Useful libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
import statsmodels.stats.weightstats as sms
from scipy.stats import t
from scipy.stats import norm
import math
```

1.1 9.2.8

[26]:
```
data=pd.read_excel("DS9.2.8.xls")
dat_A= data['Standard Antibiotic']
dat_B= data['New Antibiotic']
# We create the difference distribution
print("Difference distribution:\n", diff.describe())
```

```
Difference distribution:
 count    8.000000
mean     1.375000
std      1.784657
min     -1.100000
25%      0.025000
50%      1.250000
75%      2.550000
max      4.100000
dtype: float64
```

We have the difference $z_i = x_i - y_i$ has a mean $\bar{z} = \mu_S - \mu_N$ where S is the standard antibiotic and N is the new one. So, $\bar{z} = 1.375$ and the sample standard deviation is $s = 1.7847$ The hypoteses are: - $H_0 : \mu \leq 0$ - $H_A : \mu > 0$

Test statistics:

$$t = \frac{\sqrt{n}\bar{z}}{s} = \frac{\sqrt{8} \times 1.375}{1.7847} = 2.179 \tag{1}$$

[27]:
```python
print(("P-value using sf (1-cdf) = {:.4f}").format(t.sf(2.179, 7)))
```

```
P-value using sf (1-cdf) = 0.0329
```

So, there is some evidence, although it is not strong at all, that the new antibiotic is quicker than the standard one

1.2 9.3.2

We first declare some variables we will use based on the distribution:

[81]:
```python
# Population A
n = 14
x_bar = 32.45
s_x = 4.30

# Population B
m = 14
y_bar = 41.45
s_y = 5.23

alpha = 0.01 # = 1 - confidence level
```

The pooled variance is given by the following expression:

$$s_p^2 = \frac{(n-1)s_x^2 + (m-1)s_y^2}{n+m-2} \tag{2}$$

[82]:
```python
# Pooled variance:
def pooled_variance(n, m, s_x, s_y):
    return ( ((n-1)*s_x**2 + (m-1)*s_y**2) / (n+m-2))

def wing_span(s_p, n, m, alpha):
    return t.ppf(1-alpha/2, n+m-2) * s_p*math.sqrt(1/n + 1/m)

def print_confidence_interval(alpha, mu, wing_span):
    print(("Confidence interval with {:.2f}% confidence level: ({:.4f})({:.
 ↪4f})").format(((1-alpha)*100), mu - wing_span, mu + wing_span))

diff = x_bar - y_bar
s_p = math.sqrt(pooled_variance(n, m, s_x, s_y))
wing_span = wing_span(s_p, n, m, alpha)
print("a)")
print_confidence_interval(alpha, diff, wing_span)
```

```
a)
Confidence interval with 99.00% confidence level: (-14.0282)(-3.9718)
```

Now, we need to calculate the degrees of freedom by the following formula:

$$v^* = \frac{(s_x^2/n + s_y^2/m)^2}{s_x^4/n^2(n-1) + s_y^4/m^2(m-1)} \tag{3}$$

[83]:

```
def degrees_of_freedom(n, m, s_x, s_y):
    nu =  ((s_x**2/n + s_y**2/m)**2) / ( s_x**4 / (n**2*(n-1)) + s_y**4 /␣
↪(m**2*(m-1)))
    return math.floor(v)
nu = degrees_of_freedom( n, m, s_x, s_y)

def wing_span(alpha, nu, n, m, s_x, s_y):
    return t.ppf(1-alpha/2, nu) * math.sqrt( s_x**2/n + s_y**2/m)

def print_confidence_interval(alpha, mu, wing_span):
    print(("Confidence interval with {:.2f}% confidence level: ({:.4f})({:.
↪4f})").format(((1-alpha)*100), mu - wing_span, mu + wing_span))

wing_span = wing_span(alpha, nu, n, m, s_x, s_y)
print("b)")
print_confidence_interval(alpha, diff, wing_span)
```

```
b)
Confidence interval with 99.00% confidence level: (-14.0440)(-3.9560)
```

We consider the statistics is

$$T = \frac{\bar{x} - \bar{y} - (\mu_A - \mu_B)}{\sqrt{\frac{s_x^2}{n} + \frac{s_y^2}{m}}} \tag{4}$$

The hypoteses are: - $H_0 : \mu_A = \mu_B$ - $H_A : \mu_A \neq \mu_B$

[88]:

```
# Calculate the t-statistics
def T_statistic(x_bar, y_bar, s_x, s_y, n, m, mu_difference = 0):
    return ( (x_bar - y_bar - mu_difference)  / math.sqrt(s_x**2/n + s_y**2/m))

t_stat = T_statistic(x_bar, y_bar, s_x, s_y, n, m, mu_difference = 0)
print("|t| value: ", abs(t_stat))
print("Critical point: ", t.ppf(1- alpha/2, n + m -2 ) )
```

```
|t| value:  4.973595778437414
Critical point:  2.7787145333289134
```

The null hypothesis is rejected because $\mid t \mid$ is greater than the critical point. The p-value can be calculated as: $2 \times P(t_{26} \geq 4.97) = 0.000$

[91]:

```
print(("P-value = {:.4f}").format(2 * t.cdf(t_stat, n + m -2)))
```

```
P-value = 0.0000
```

1.3 9.3.10

From now on, let's directly solve the exercises in Python

[92]:
```
# Population A
n = 38
x_bar = 5.782

# Population B
m = 40
y_bar = 6.443

sigma = 2.0
alpha = 0.01 # = 1 - confidence level
```

We calculate the p-value based on $H_0 : \mu_A - \mu_B > 0$

[99]:
```
# Calculate the t-statistics

def Z_statistic(x_bar, y_bar, s_x, s_y, n, m, mu_difference = 0):
    return ( (x_bar - y_bar - mu_difference)  / math.sqrt(s_x**2/n + s_y**2/m))

Z_stat = Z_statistic(x_bar, y_bar, sigma, sigma, n, m)
print("a)")
print("Z-statistics: ", Z_stat)
print("p-value: ", norm.cdf(Z_stat))
```

```
a)
Z-statistics:  -1.4589686381754354
p-value:  0.07228686987180125
```

[107]:
```
def wing_span(alpha, n, m, s_x, s_y):
    return norm.ppf(1-alpha) * math.sqrt( s_x**2/n + s_y**2/m)

def print_confidence_interval(alpha, mu, wing_span, lower_bound = False,␣
↪upper_bound = False):
    if upper_bound:
        print(("Confidence interval with {:.2f}% confidence level: (-∞)({:.
↪4f})").format(((1-alpha)*100), mu + wing_span))
        return
    if lower_bound:
        print(("Confidence interval with {:.2f}% confidence level: ({:.
↪4f})(∞))").format(((1-alpha)*100), mu - wing_span))
        return
    # Default case: double bounded
    print(("Confidence interval with {:.2f}% confidence level: ({:.4f})({:.
↪4f})").format(((1-alpha)*1, mu - wing_span, mu + wing_span)))
```

```
diff = x_bar - y_bar
wing_span = wing_span(alpha, n, m, sigma, sigma)
print("b)")
print_confidence_interval(alpha, diff, wing_span, upper_bound = True)
```

```
b)
Confidence interval with 99.00% confidence level: (-∞)(0.3930)
```

1.4 9.3.14

[111]:
```
# Population A
n = 14
x_bar = 32.45
s_x = 4.30

# Population B
m = 14
y_bar = 41.45
s_y = 5.23

alpha = 0.01 # = 1 - confidence level
```

[110]:
```
print("t value: ", t.ppf(1-alpha/2, n+m-2))
```

```
t value:  2.7787145333289134
```

The length follows the following equation:

$$L = 2 \times t_{\alpha/2,\nu} \sqrt{\frac{s_A^2}{n} + \frac{s_B^2}{m}} \leq 5 \tag{5}$$

So we get:

$$n = m \geq \frac{4t_{\alpha/2,\nu}(s_A^2 + s_B^2)}{L_0^2} = \frac{4 \times 2.779^2 \times (4.3^2 + 5.23^2)}{5^2} = 56.646 \tag{6}$$

So, 57 samples will suffice, which means $57 - 14 = 43$ more samples should be collected from each population

1.5 9.3.22

[115]:
```
# Population A
n = 16
x_bar = 1.053
s_x = 0.058

# Population B
m = 16
y_bar = 1.071
s_y = 0.062
```

```
sign_level = 0.05
```

The hypoteses are: - $H_0 : \mu_A - \mu_B \geq 0$ - $H_A : \mu_A - \mu_B < 0$

[120]:
```
# Calculate the t-statistics
def T_statistic(x_bar, y_bar, s_x, s_y, n, m, mu_difference = 0):
    return ( (x_bar - y_bar - mu_difference)  / math.sqrt(s_x**2/n + s_y**2/m))

def degrees_of_freedom(n, m, s_x, s_y):
    nu =  ((s_x**2/n + s_y**2/m)**2) / ( s_x**4 / (n**2*(n-1)) + s_y**4 /␣
↪(m**2*(m-1)))
    return math.floor(nu)

nu = degrees_of_freedom( n, m, s_x, s_y)
print("Degrees of freedom", nu)
t_stat = T_statistic(x_bar, y_bar, s_x, s_y, n, m, mu_difference = 0)
print("Test statistics: ", t_stat)
print(("P-value using cdf = {:.4f}").format(t.cdf(t_stat, nu)))
```

```
Degrees of freedom 29
Test statistics:  -0.8480571253767827
P-value using cdf = 0.2017
```

Therefore, there is not sufficient evidence that the null hypotesis is true

1.6 9.3.26

[134]:
```
# Population A
n = 10
x_bar = 7.76
s_x = 1.07

# Population B
m = 9
y_bar = 6.88
s_y = 0.62


alpha = 0.01 # = 1 - confidence level
```

[136]:
```
def wing_span(alpha, nu, n, m, s_x, s_y):
    return t.ppf(1-alpha, nu) * math.sqrt( s_x**2/n + s_y**2/m)

def print_confidence_interval(alpha, mu, wing_span, lower_bound = False,␣
↪upper_bound = False):
    if upper_bound:
```

```
        print(("Confidence interval with {:.2f}% confidence level: (-∞)({:.
↪4f})").format(((1-alpha)*100), mu + wing_span))
        return
    if lower_bound:
        print(("Confidence interval with {:.2f}% confidence level: ({:.
↪4f})(∞))").format(((1-alpha)*100), mu - wing_span))
        return
    # Default case: double bounded
    print(("Confidence interval with {:.2f}% confidence level: ({:.4f})({:.
↪4f})").format(((1-alpha)*1, mu - wing_span, mu + wing_span)))

def degrees_of_freedom(n, m, s_x, s_y):
    nu =  ((s_x**2/n + s_y**2/m)**2) / ( s_x**4 / (n**2*(n-1)) + s_y**4 /␣
↪(m**2*(m-1)))
    return math.floor(nu)

nu = degrees_of_freedom( n, m, s_x, s_y)
print("Degrees of freedom: ", nu)

t_stat = T_statistic(x_bar, y_bar, s_x, s_y, n, m, mu_difference = 0)

diff = x_bar - y_bar
ws = wing_span(alpha, nu, n, m, s_x, s_y)
print("a)")
print_confidence_interval(alpha, diff, ws, lower_bound = True)

ws = wing_span(0.05, nu, n, m, s_x, s_y)
print("b)")
print_confidence_interval(0.05, diff, ws, lower_bound = True)
```

```
Degrees of freedom:  14
a)
Confidence interval with 99.00% confidence level: (-0.1606)(∞))
b)
Confidence interval with 95.00% confidence level: (0.1817)(∞))
```

The value of c increases, since the confidence level has decreased: we have a "tighter" constraint now

The hypoteses are: - $H_0 : \mu_A \leq \mu_B$ - $H_A : \mu_A > \mu_B$

[141]:
```
# Calculate the t-statistics
def T_statistic(x_bar, y_bar, s_x, s_y, n, m, mu_difference = 0):
    return ( (x_bar - y_bar - mu_difference)  / math.sqrt(s_x**2/n + s_y**2/m))

t_stat = T_statistic(x_bar, y_bar, s_x, s_y, n, m)
print("T statistic: ", t_stat)
print("Size alpha = 0.01: ", t.ppf(1-0.01, nu))
```

```
print(("P-value using sf (1-cdf) = {:.4f}").format(t.sf(t_stat, nu)))
```

```
T statistic:  2.2194985542975125
Size alpha = 0.01:  2.624494067560231
P-value using sf (1-cdf) = 0.0217
```

We can accept the null hypothesis since $t \leq t_{0.01,14}$. The p value is $P(t_{14} \geq 2.22) = 0.0217$

1.7 9.7.10

[150]:
```
# Population A
n = 48
x_bar = 432.7
s_x = 20.39

# Population B
m = 10
y_bar = 403.5
s_y = 15.62


alpha = 0.01 # = 1 - confidence level
```

The hypoteses are: - $H_0 : \mu_A \leq \mu_B$ - $H_A : \mu_A > \mu_B$

[153]:
```
# Calculate the t-statistics
def T_statistic(x_bar, y_bar, s_x, s_y, n, m, mu_difference = 0):
    return ( (x_bar - y_bar - mu_difference)  / math.sqrt(s_x**2/n + s_y**2/m))

def degrees_of_freedom(n, m, s_x, s_y):
    nu =  ((s_x**2/n + s_y**2/m)**2) / ( s_x**4 / (n**2*(n-1)) + s_y**4 /
 ↪(m**2*(m-1)))
    return math.floor(nu)
nu = degrees_of_freedom(n, m, s_x, s_y)
print(nu)
t_stat = T_statistic(x_bar, y_bar, s_x, s_y, n, m)
print("T statistics:", t_stat)
print(("P-value using sf (1-cdf) = {:.4f}").format(t.sf(t_stat, nu)))
```

```
16
T statistics: 5.07845732358246
P-value using sf (1-cdf) = 0.0001
```

Given the low p-value we can accept the alternative hypothesis and conclude that the new driving route will be quicker on average than the old one

1.8 9.7.14

[195]:
```
data=pd.read_excel("DS9.6.7.xls")
dat_A= data['Procedure 1']
dat_B= data['Procedure 2']
# We create the difference distribution
z = dat_A - dat_B
```

The hypoteses are: - $H_0 : \mu_A = \mu_B$ - $H_A : \mu_A \neq \mu_B$

[197]:
```
print("Mean =", z.mean())
print("Standard deviation = ", math.sqrt(z.var()))
t_stat = 3*z.mean() /math.sqrt(z.var())
print("t statistics: ", t_stat)
print(("P-value using sf (1-cdf) = {:.4f}").format(2 * t.sf(t_stat, 8)))
```

```
Mean = -0.022222222222222143
Standard deviation =  0.5911382616989399
t statistics:  -0.1127767748869198
P-value using sf (1-cdf) = 1.0870
```

Given the high p-value, there is no evidence to conclude there is any difference between the two procedures

1.9 9.7.20

[169]:
```
data=pd.read_excel("DS9.6.11.xls")
dat_A= data['Joystick Design 1']
dat_B= data['Joystick Design 2']
# We create the difference distribution
z = dat_A - dat_B
```

The hypoteses are: - $H_0 : \mu_A = \mu_B$ - $H_A : \mu_A \neq \mu_B$

[203]:
```
print("Mean =", z.mean())
print("Standard deviation = ", math.sqrt(z.var()))
t_stat = 3*z.mean() /math.sqrt(z.var())
print("t statistics: ", t_stat)
print(("P-value using sf (1-cdf) = {:.4f}").format(2 * t.sf(abs(t_stat), 8)))
```

```
Mean = -0.022222222222222143
Standard deviation =  0.5911382616989399
t statistics:  -0.1127767748869198
P-value using sf (1-cdf) = 0.9130
```

We can say that there is some evidence, although not overwhelming, that the two joysticks result in different error rate measurements

[193]:
```
def print_confidence_interval(alpha, mu, wing_span, lower_bound = False,␣
 ↪upper_bound = False):
```

```
    if upper_bound:
        print(("Confidence interval with {:.2f}% confidence level: (-∞)({:.
↪4f})").format(((1-alpha)*100), mu + wing_span))
        return
    if lower_bound:
        print(("Confidence interval with {:.2f}% confidence level: ({:.
↪4f})(∞))").format(((1-alpha)*100), mu - wing_span))
        return
    # Default case: double bounded
    print(("Confidence interval with {:.2f}% confidence level: ({:.4f})({:.
↪4f})").format((1-alpha)*1, mu - wing_span, mu + wing_span))

ws = t.ppf(1-0.005, 8) * math.sqrt(z.var()) / 3

print_confidence_interval(0.01, z.mean(), ws)
```

```
Confidence interval with 0.99% confidence level: (-0.0151)(0.0565)
```

1.10 9.7.22

[205]:
```
data=pd.read_excel("DS9.6.13.xls")
dat_A= data['Sphygmomanometer']
dat_B= data['Finger Monitor']
# We create the difference distribution
z = dat_A - dat_B
```

The hypoteses are: - $H_0 : \mu_A = \mu_B$ - $H_A : \mu_A \neq \mu_B$

[206]:
```
print("Mean =", z.mean())
print("Standard deviation = ", math.sqrt(z.var()))
t_stat = math.sqrt(15)*z.mean() /math.sqrt(z.var())
print("t statistics: ", t_stat)
print(("P-value using sf (1-cdf) = {:.4f}").format(2 * t.sf(abs(t_stat), 14)))
```

```
Mean = 0.4
Standard deviation =  1.9566735620873066
t statistics:  0.7917484901417817
P-value using sf (1-cdf) = 0.4417
```

We can conclude from the p-value that there is enough evidence to state that the means of measurement of the two instruments are different