# [CC511] Homework 9 20204817 Federico Berto

November 17, 2020

## 1 Homework 9 - Federico Berto

[2]:
```python
# Importing 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 z
from scipy.stats import norm
import math
```

### 1.1 Exercise 10.1.4

The 95% confidence can be calculated via $z_{0.05}$

[8]:
```python
z = norm.ppf(1-0.05)
print(z)
```

```
1.6448536269514722
```

The confidence interval is:

$$\left(\frac{35}{44} - \frac{1.645}{44} \times \sqrt{\frac{35 \times (44-35)}{44}}, 1\right) = (0.695, 1) \tag{1}$$

### 1.2 Exercise 10.1.8

We know that $L = 2z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$, therefore:

$$n \geq \frac{4z^2_{\alpha/2}\hat{p}(1-\hat{p})}{L^2} \tag{2}$$

[9]:
```python
z = norm.ppf(1-0.005)
print(z)
```

```
2.5758293035489004
```

Given L $= 0.04$, then we have for $\hat{p} = 0.50$:

$$n \geq \frac{4 \times 2.5758 \times 0.50(1 - 0.50)}{0.04^2} = 1609.9 \tag{3}$$

So, n has to be at least 1610.

If $\hat{p} = 0.40$:

$$n \geq \frac{4 \times 2.5758 \times 0.40(1 - 0.40)}{0.04^2} = 1545.5 \tag{4}$$

In this case, n has to be at least 1546.

### 1.3 Exercise 10.1.18

a) The hypoteses are:

- $H_0 : p_A \leq 0.05$
- $H_A : p_B > 0.05$

We calculate the statistics for the normal approximation as:

$$z = \frac{x - np_0 - 0.5}{\sqrt{np_0(1 - p_0)}} = \frac{13 - 62 \times 0.05 - 0.5}{\sqrt{62 \times 0.05 \times (1 - 0.05)}} = 5.48 \tag{5}$$

[23]:
```
print('p-value: ', (1 - norm.cdf(5.48)) )
```

```
p-value:  2.1266291838628604e-08
```

We can conclude that with the p value close to 0, there is sufficient evidence to conclude that the probability of breakdown is above 5%.

b) The 95% confidence can be calculated via $z_{0.05}$

[ ]:
```
z = norm.ppf(1-0.05)
print(z)
```

Thus the confidence interval is:

$$\left( \frac{13}{62} - \frac{1.645}{62} \times \sqrt{\frac{13 \times (62 - 13)}{62}}, 1 \right) = (0.125, 1) \tag{6}$$

### 1.4 Exercise 10.2.2

a) The 95% confidence can be calculated via $z_{0.005}$

[26]:
```
z = norm.ppf(1-0.005)
print(z)
```

```
2.5758293035489004
```

The confidence interval can be calculated as such:

$$\hat{p}_a - \hat{p}_b \pm z_{\alpha/2}\sqrt{\frac{\hat{p}_a(1-\hat{p}_a)}{n} + \frac{\hat{p}_b(1-\hat{p}_b)}{m}} \tag{7}$$

So, the confidence interval is:

$$\frac{4}{50} - \frac{10}{50} \pm 2.576 \times \sqrt{\frac{4 \times (50-4)}{50^3} + \frac{10 \times (50-10)}{50^3}} = (-0.296, 0.056) \tag{8}$$

b) We can use the pooled probability estimate, which is $\hat{p} = \frac{x+y}{n+m} = \frac{4+10}{50+50} = 0.14$

The test statistcs is:

$$z = \frac{\hat{p}_A - \hat{p}_b}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n} + \frac{1}{m})}} \tag{9}$$

Therefore:

$$z = \frac{\frac{4}{50} - \frac{10}{50}}{\sqrt{0.14 \times (1-0.14) \times (\frac{1}{50} + \frac{1}{50})}} = -1.729 \tag{10}$$

[33]:
```
print('p value: ', 2*norm.cdf(-1.729))
```

```
p value:  0.08380909507894739
```

c) The confidence interval becomes:

$$\frac{40}{500} - \frac{100}{500} \pm 2.576 \times \sqrt{\frac{40 \times (500-40)}{500^3} + \frac{100 \times (500-100)}{500^3}} = (-0.176, 0.064) \tag{11}$$

We can use the pooled probability estimate, which is $\hat{p} = \frac{x+y}{n+m} = \frac{40+100}{500+500} = 0.14$

$$z = \frac{\frac{40}{500} - \frac{100}{500}}{\sqrt{0.14 \times (1-0.14) \times (\frac{1}{500} + \frac{1}{500})}} = -5.468 \tag{12}$$

[35]:
```
print('p value: ', 2*norm.cdf(-5.468))
```

```
p value:  4.551418826709858e-08
```

In this case, the p-value becomes almost zero.

## 1.5 Exercise 10.2.12

[36]:
```
z = norm.ppf(1-0.05)
print(z)
```

```
1.6448536269514722
```

We can construct the uppder confidence bound for $p_A - p_B$, where $p_A$ is the probability of following the link in the original design and $p_B$ is the probability after the modification, as following:

$$\left(-1, \frac{22}{542} - \frac{64}{601} + 1.645 \times \sqrt{\frac{22 \times (542-22)}{542^3} + \frac{64 \times (601-64)}{601^3}}\right) = (-1, -0.041) \tag{13}$$

a) The hypoteses are:

- $H_0 : p_A \geq p_B$
- $H_A : p_A < p_B$

We can use the pooled probability estimate, which is $\hat{p} = \frac{x+y}{n+m} = \frac{22+64}{542+601} = 0.0752$

The test statistics becomes

$$z = \frac{\frac{22}{542} - \frac{64}{601}}{\sqrt{0.0752 \times (1-0.0752) \times (\frac{1}{542} + \frac{1}{601})}} = -4.22 \tag{14}$$

[38]:
```
print('p value: ', norm.cdf(-4.22))
```

```
p value:  1.2215115925253025e-05
```

Being the p-value almost zero, there is sufficient evidence to conclude that the probability of the link being following has increased.

### 1.6 Exercise 10.3.6

Soft drink type | Formulation

Formulation I 225

Formulation II 223

Formulation III 152

If the formulations are equally likely, then the expected cell frequencies are: $e_i = 600 \times \frac{1}{3} = 200$ We can use the Pearson chi-square statistics to calculate the *p-value*:

$$X^2 = \sum_{i=1}^{k} \frac{(x_i - e_i)^2}{e_i} \tag{15}$$

yielding

$$X^2 = \frac{225-200)^2}{200} + \frac{223-200)^2}{200} + \frac{152-200)^2}{200} = 17.29 \tag{16}$$

The p-value is given by $P(\chi^2_2 \geq 17.29)$:

[41]:
```
from scipy.stats import chi2
print('p-value: ', chi2.sf(17.29, 2))
```

```
p-value:  0.00017600467513708998
```

We can state that it is not plausible that the formulations of the soft drinks are equally likely.

## 1.7 Exercise 10.3.14

We can calculate the probabilities given the Weibull distribution with $\lambda = 0.065$ and $a = 0.45$: - $p_1^* = P(X \leq 24) = 1 - e^{-(\lambda x)^\alpha} = 1 - e^{-(0.065\times 24)^{0.45}} = 0.705$ - $p_2^* = P(X \leq 48) = 1 - e^{-(\lambda x)^\alpha} = 1 - e^{-(0.065\times 48)^{0.45}} = 0.812$ - $p_3^* = P(X \leq 72) = 1 - e^{-(\lambda x)^\alpha} = 1 - e^{-(0.065\times 72)^{0.45}} = 0.865$

The observed cell frequencies are: $12, 53, 39, 21$. Therefore: - $e_1 = np_1^* = 125 \times (0.705) = 88.125$ - $e_2 = np_2^* = 125 \times (0.812 - 0.705) = 13.375$ - $e_3 = np_3^* = 125 \times (0.865 - 0.812) = 6.625$ - $e_4 = np_4^* = 125 \times (1 - 0.865) = 16.87$

We use the Pearson $X^2$ statistics which yield

$$X^2 = \sum_{i=1}^{k} \frac{(x_i - e_i)^2}{e_i} \sim 342.4 \tag{17}$$

[43]:
```
print('p-value: ', chi2.sf(342.4, 3))
```

```
p-value:  6.595603058895007e-74
```

Which is basically 0. Thus, the null hypotesis of the Weibull distribution approximation is clearly rejected.

## 1.8 Exercise 10.4.2

We do the experiment via the software package:

[56]:
```
from scipy.stats import chi2_contingency
aptocc = np.array([[48,111,186,142],[71,89,174,181],[63,95,181,190]])
aptocc = pd.DataFrame(data=aptocc,
index=['No Fertilizer','Fertilizer I', 'Fertilizer II'], columns=['Dead','Slow␣
↪Growth','Medium Growth','Strong growth'])
print("Observed cell frequencies:\n", aptocc)
```

```
Observed cell frequencies:
                Dead  Slow Growth  Medium Growth  Strong growth
No Fertilizer    48          111            186            142
Fertilizer I     71           89            174            181
Fertilizer II    63           95            181            190
```

[57]:
```
chi, pvalue, dof, expctd = chi2_contingency(aptocc)
print("Pearson's Chi-squared test \nX-squared = %.4f, p-value = %.4f, df = %d,"␣
↪%(chi,
pvalue, dof))
print("Expected cell frequencies:\n", pd.DataFrame(expctd, index=['No␣
↪Fertilizer','Fertilizer I', 'Fertilizer II'], columns=['Dead','Slow␣
↪Growth','Medium Growth','Strong growth']))
```

```
Pearson's Chi-squared test
X-squared = 13.6591, p-value = 0.0337, df = 6,
Expected cell frequencies:
```

| | Dead | Slow Growth | Medium Growth | Strong growth |
|---|---|---|---|---|
| No Fertilizer | 57.892880 | 93.837361 | 172.088178 | 163.181581 |
| Fertilizer I | 61.221424 | 99.232528 | 181.982364 | 172.563684 |
| Fertilizer II | 62.885696 | 101.930111 | 186.929458 | 177.254735 |

There is a suggestion that the growth pattern is different for the different growing conditions, but there is no overwhelming evidence.

## 1.9 Exercise 10.4.6

For the $2 \times 2$ contingency table we obtain:

| | $c1$ | $c2$ | Sum up |
|---|---|---|---|
| $r1$ | $x11$ | $x12$ | $x1\ = x11 + x12$ |
| $r2$ | $x21$ | $x22$ | $x2\ = x21 + x22$ |
| Sum up | $x\ \ 1 = x11 + x21$ | $x\ \ 2 = x12 + x22$ | $n = x1\ + x2\ = x\ \ 1 + x\ \ 2$ |

$$e_{ij} = \frac{x_{i\cdot} \cdot x_{\cdot j}}{n}$$

So, we have:

$$\frac{(x_{11} - e_{11})^2}{e_{11}} = \frac{(x_{11} - \frac{(x_{11}+x_{12})(x_{11}+x_{21})}{x_{11}+x_{21}+x_{12}+x_{22}})^2}{\frac{x_{1\cdot}x_{\cdot 1}}{x_{11}+x_{21}+x_{12}+x_{22}}} = \frac{x_{2\cdot}x_{\cdot 2}(x_{11}x_{22} - x_{12}x_{21})^2}{nx_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}}$$

We also have in a similar way:

$$\frac{(x_{21} - e_{21})^2}{e_{21}} = \frac{x_{1\cdot}x_{\cdot 2}(x_{11}x_{22} - x_{12}x_{21})^2}{nx_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}}$$

and we can easily obtain the other coefficients in the same way. Finally, we can prove the result by substituting:

$$\begin{aligned}
\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{(x_{ij} - e_{ij})^2}{e_{ij}} &= \frac{x_{2\cdot}x_{\cdot 2}(x_{11}x_{22} - x_{12}x_{21})^2}{nx_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}} + \frac{x_{2\cdot}x_{\cdot 1}(x_{11}x_{22} - x_{12}x_{21})^2}{nx_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}} \\
&+ \frac{x_{1\cdot}x_{\cdot 1}(x_{11}x_{22} - x_{12}x_{21})^2}{nx_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}} + \frac{x_{1\cdot}x_{\cdot 2}(x_{11}x_{22} - x_{12}x_{21})^2}{nx_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}} \\
&= \frac{(x_{11}x_{22} - x_{12}x_{21})^2}{nx_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}}(x_{2\cdot}x_{\cdot 2} + x_{2\cdot}x_{\cdot 1} + x_{1\cdot}x_{\cdot 2} + x_{1\cdot}x_{\cdot 1}) \\
&= \frac{(x_{11}x_{22} - x_{12}x_{21})^2}{nx_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}}(x_{\cdot 2} + x_{\cdot 1})^2 = \frac{n(x_{11}x_{22} - x_{12}x_{21})^2}{x_{2\cdot}x_{\cdot 2}x_{1\cdot}x_{\cdot 1}}
\end{aligned}$$

## 1.10 Exercise 10.4.10

We may conduct the experiment via the software package

[73]:
```python
from scipy.stats import chi2_contingency
aptocc = np.array([[31, 17, 9], [36,9,4], [56, 19, 15]])
aptocc = pd.DataFrame(data=aptocc,
index=['A', 'B', 'C'], columns=['Minor Cracking','Medium cracking', 'Severe␣
↪cracking'])
print("Observed cell frequencies:\n", aptocc)
```

```
Observed cell frequencies:
    Minor Cracking  Medium cracking  Severe cracking
A               31               17                9
B               36                9                4
C               56               19               15
```

[74]:
```python
chi, pvalue, dof, expctd = chi2_contingency(aptocc)
print("Pearson's Chi-squared test \nX-squared = %.4f, p-value = %.4f, df = %d,"␣
↪%(chi,
pvalue, dof))
print("Expected cell frequencies:\n", pd.DataFrame(expctd, index=['A', 'B',␣
↪'C'], columns=['Minor Cracking','Medium cracking', 'Severe cracking']))
```

```
Pearson's Chi-squared test
X-squared = 5.0237, p-value = 0.2849, df = 4,
Expected cell frequencies:
    Minor Cracking  Medium cracking  Severe cracking
A       35.770408        13.086735         8.142857
B       30.750000        11.250000         7.000000
C       56.479592        20.663265        12.857143
```

Therefore, the null hypotesis of independence is plausible and we have no overwhelming evidence to state the three types of asphalt are different with respect to cracking.