03 Confidence Interval

November 20, 2022

1 Confidence Intervals

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

1.1 Compute confidence interval with known standard deviation

Let $X_1, \ldots, X_n$ be independent measurements with $X_i \sim N(\mu, \sigma^2)$, where σ^2 is known and $\theta = \mu$ should be estimated from the data. Then

$$\hat{\theta}_L = \bar{X} - z_{1-\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}}$$
$$\hat{\theta}_U = \bar{X} + z_{1-\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}}$$

The length of the interval can be computed as follows

$$l(\alpha, n) = \hat{\theta}_U - \hat{\theta}_L = 2 * \sigma * \frac{z_{1-\frac{\alpha}{2}}}{\sqrt{n}}$$

[2]:
```
n = 100  # Number of measurements
sigma = 0.1  # Known standard deviation
alpha = 0.05  # Significance level
mu0 = 1  # Average of the population

x = stats.norm(mu0, sigma).rvs(n)
x_mean = np.mean(x)

z = stats.norm().ppf(1 - alpha / 2)

lo = x_mean - z * sigma / np.sqrt(n)
up = x_mean + z * sigma / np.sqrt(n)

print(f"Confidence interval: [{lo}; {up}]")
```

```
Confidence interval: [0.9834028507512727; 1.0226021304420738]
```

[3]:
```
# length of the confidence interval
length_1 = up - lo
length_2 = 2 * sigma * z / np.sqrt(n)

print(length_1, length_2)
```

```
0.03919927969080117 0.03919927969080108
```

1.2 Compute confidence interval with unknown standard deviation

Let $X_1, \ldots, X_n$ be independent measurements with $X_i \sim N(\mu, \sigma^2)$, where σ^2 is unknown and $\theta = \mu$ has to be estimated from the data. Then

$$
\begin{aligned}
\hat{\theta}_L &= \bar{X} - t_{1-\frac{\alpha}{2};n-1} * \frac{s_x}{\sqrt{n}} \\
\hat{\theta}_U &= \bar{X} + t_{1-\frac{\alpha}{2};n-1} * \frac{s_x}{\sqrt{n}}
\end{aligned}
$$

where

$$
s_x = \sqrt{\frac{1}{n-1} * \sum_{i=1}^{n} (X_i - \bar{X})^2}
$$

The length of the interval can be computed as follows

$$
l(\alpha, n) = \hat{\theta}_U - \hat{\theta}_L = 2 * s_x * \frac{t_{1-\frac{\alpha}{2};n-1}}{\sqrt{n}} = 2 * \sqrt{\frac{1}{n-1} * \sum_{i=1}^{n} (X_i - \bar{X})^2} * \frac{t_{1-\frac{\alpha}{2};n-1}}{\sqrt{n}}
$$

[4]:
```
n = 100   # Numbers of measurements
alpha = 0.05   # Significance level
mu0 = 1   # Average of the population

x = stats.norm(mu0, 0.2).rvs(n)
x_mean = np.mean(x)

t = stats.t(n - 1).ppf(1 - alpha / 2)
sx = np.sqrt(1 / (n - 1) * np.sum((x - x_mean) ** 2)) or np.std(x, ddof=1)

lo = x_mean - t * sx / np.sqrt(n)
up = x_mean + t * sx / np.sqrt(n)

print(f"Confidence interval: [{lo}; {up}]")
```

```
Confidence interval: [0.9270880963847782; 1.0068578615614334]
```

[5]:
```python
# length of the confidence interval
length_1 = up - lo
length_2 = 2 * np.std(x, ddof=1) * t / np.sqrt(n)

print(length_1, length_2)
```

```
0.07976976517665524 0.07976976517665525
```

Why can the normal distribution be used for calculating the confidence interval?

When computing a confidence interval, we only use the distribution of the mean of the data to construct the bounds. The central limit theorem tells us, that the distribution of the mean of a sample will be more and more normally distributed the larger the sample size gets.

[ ]:
```python

```