z test

Suppose you are a data scientist at a company, and you want to determine if the average time spent by users on your website is less than from the industry average.

The industry average time spent on a website is 8 minutes.

You collect a sample of 100 users from your website, and you find that they spend an average of 7.5 minutes with a standard deviation of 2 minutes

Z-TEST

- Formula to find the value of Z (z-test) Is:

$$Z = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$$

- $\bar{x}$ = mean of sample
- μ_0 = mean of population
- σ = standard deviation of population
- n = no. of observations

1. **Define the Hypotheses:**

    - **Null Hypothesis (H0)**: The average time spent by users on the website is equal to the industry average.
        - $H_0 : \mu = 8$
    - **Alternative Hypothesis (H1)**: The average time spent by users on the website is different from the industry average.
        - $H_1 : \mu \neq 8$

2. **Collect Sample Data:**

    - Sample mean ($\bar{X}$) = 7.5 minutes
    - Sample standard deviation (σ) = 2 minutes
    - Sample size (n) = 100

3. **Calculate the Z-score:**

    - Use the formula: $Z = \frac{\bar{X}-\mu}{\sigma/\sqrt{n}}$

4. **Calculate the p-value:**

    - Use the cumulative distribution function (CDF) of the standard normal distribution.

5. **Compare the p-value with the significance level (α):**

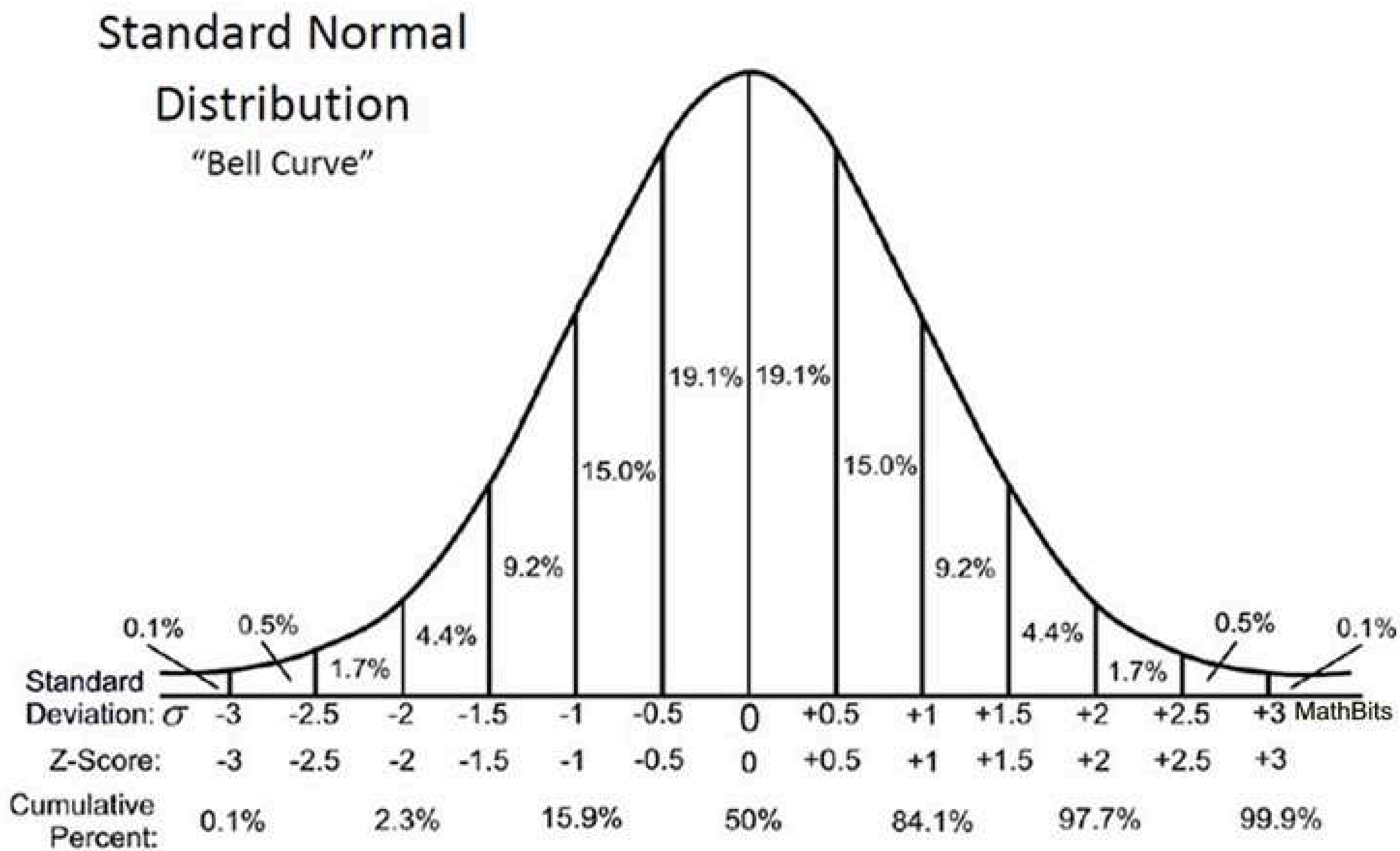
    - Common significance level is 0.05.

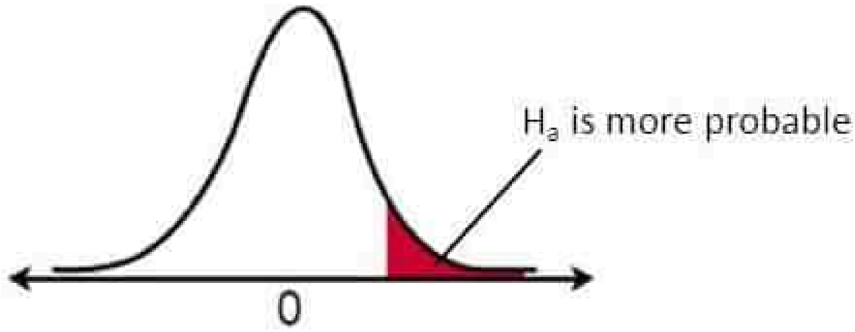
In [1]:
```
import numpy as np
from scipy import stats
# Step 1: Define the hypotheses
# H1: mu < 8 (alternative hypothesis)

# Step 2: Collect sample data
sample_mean = 7.5 # Sample mean, xbar
sample_std = 2     # Sample standard deviation signa
n = 100            # Sample size n
population_mean = 8  # Population mean

# Step 3: Calculate the Z-score
z_score = (sample_mean - population_mean) / (sample_std / np.sqrt(n))

z_score
```

Out[1]: -2.5

In [2]:
```
x=[2,2.4,3,3,3,4,3.5]
np.mean(x)
```

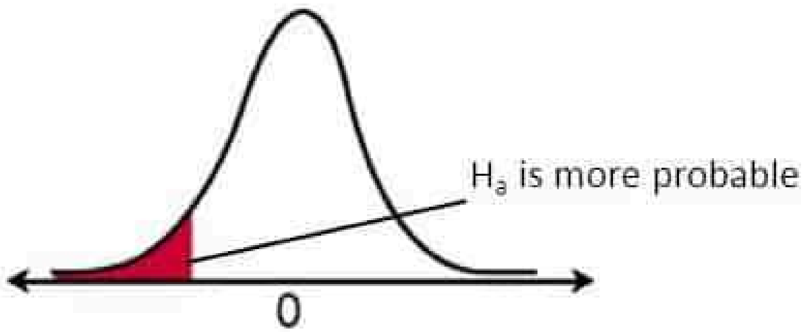
Out[2]: 2.9857142857142853


Standard Normal
Distribution
"Bell Curve"
19.1%
19.1%
15.0%
15.0%
9.2%
9.2%
4.4%
4.4%
1.7%
1.7%
0.5%
0.5%
0.1%
0.1%
Standard
Deviation: σ -3 -2.5 -2 -1.5 -1 -0.5 0 +0.5 +1 +1.5 +2 +2.5 +3
MathBits
Z-Score: -3 -2.5 -2 -1.5 -1 -0.5 0 +0.5 +1 +1.5 +2 +2.5 +3
Cumulative
Percent: 0.1% 2.3% 15.9% 50% 84.1% 97.7% 99.9%



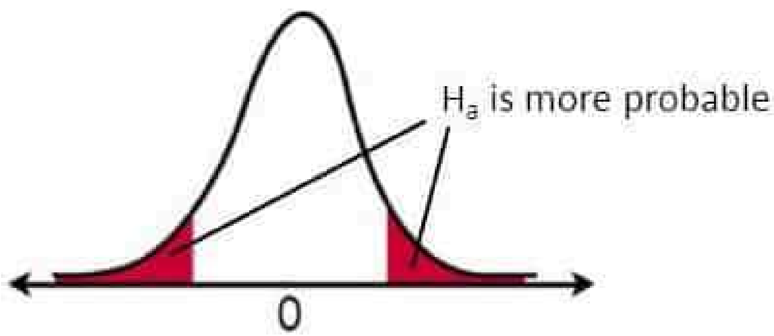
Right-tail test

$$H_a\!: \mu > \text{value}$$



Left-tail test

$$H_a\!: \mu < \text{value}$$



Two-tail test

$$H_a\!: \mu \neq \text{value}$$

In [3]:
```
# Step 4: Calculate the p-value (one-tailed test, left tail)
p_value = stats.norm.cdf(z_score)

# Print the results
print("Z-score:", z_score)
print("P-value:", p_value)

# Step 5: Compare the p-value with the significance level
alpha = 0.05  # Significance level
if p_value < alpha:
    print("Reject the null hypothesis")
else:
    print("Fail to reject/ accept the null hypothesis")
```

```
Z-score: -2.5
P-value: 0.006209665325776132
Reject the null hypothesis
```

1. **One-Tailed Test (Right Tail)**: Tests if a value is significantly **greater** than the threshold.

$$\text{p-value (right tail)} = 1 - \text{stats.norm.cdf}(z\_score)$$

2. **One-Tailed Test (Left Tail)**: Tests if a value is significantly **less** than the threshold.

$$\text{p-value (left tail)} = \text{stats.norm.cdf}(z\_score)$$

3. **Two-Tailed Test**: Tests if a value is significantly **different** (either direction) from the threshold.

$$\text{p-value (two-tailed)} = 2 \times (1 - \text{stats.norm.cdf}(|z\_score|))$$

In [ ]:

In [4]:
```
"""# One-tailed p-value (right tail)
p_value_one_tailed_right = 1 - stats.norm.cdf(z_score)
print("One-tailed p-value (right tail):", p_value_one_tailed_right)

# One-tailed p-value (left tail)
p_value_one_tailed_left = stats.norm.cdf(z_score)
print("One-tailed p-value (left tail):", p_value_one_tailed_left)

# Two-tailed p-value
p_value_two_tailed = 2 * (1 - stats.norm.cdf(abs(z_score)))
print("Two-tailed p-value:", p_value_two_tailed)"""
```

Out[4]:
```
'# One-tailed p-value (right tail)\np_value_one_tailed_right = 1 - stats.norm.cdf(z_score)\nprint("One-taile
d p-value (right tail):", p_value_one_tailed_right)\n\n# One-tailed p-value (left tail)\np_value_one_tailed_
left = stats.norm.cdf(z_score)\nprint("One-tailed p-value (left tail):", p_value_one_tailed_left)\n\n# Two-t
ailed p-value\np_value_two_tailed = 2 * (1 - stats.norm.cdf(abs(z_score)))\nprint("Two-tailed p-value:", p_v
alue_two_tailed)'
```

Type *Markdown* and LaTeX: α^2

In [ ]:
```
```

anova- mean euqallity

A retail company launched three different marketing campaigns over the past three months to boost sales for a particular product. The campaigns used different channels (e.g., social media, email, and TV ads). Now, they want to analyze whether there is a significant difference in the average sales increase across the three campaigns.

Goal:

The goal is to determine if the type of marketing campaign had a significant impact on the sales increase.

Data:

The sales increase (in percentage) for each campaign is recorded:

Campaign A (Social Media): [10, 12, 14, 11, 15, 13, 16]

Campaign B (Email): [5, 6, 7, 6, 8, 7, 9]

Campaign C (TV): [20, 25, 23, 22, 24, 21, 26]

Hypothesis:

Null Hypothesis (H0):

There is no significant difference in sales increase between the three campaigns.

Alternative Hypothesis (H1):

At least one campaign led to a significantly different sales increase.

Type *Markdown* and LaTeX: α^2

In [5]:

```
from scipy.stats import f_oneway # (one way)

# Sales increase data for each campaign
campaign_A = [10, 12, 14, 11, 15, 13, 19]
campaign_B = [5, 6, 7, 6, 8, 7, 9]
campaign_C = [20, 25, 23, 22, 24, 21, 26]

# Perform the one-way ANOVA test
f_stat, p_value = f_oneway(campaign_A, campaign_B, campaign_C)

print(f"F-statistic: {f_stat}")
print(f"P-value: {p_value}")

# Decision based on the significance level
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant difference in sales increase between the campai
else:
    print("Fail to reject the null hypothesis: No significant difference in sales increase between the campai
```

```
F-statistic: 89.69444444444423
P-value: 4.3606198203255143e-10
Reject the null hypothesis: There is a significant difference in sales increase between the campaigns.
```

Interpretation and Decision

1. If the p-value is less than 0.05, it means there is enough evidence to reject the null hypothesis, indicating that at least one campaign led to a significantly different increase in sales. The company could then focus on optimizing that specific campaign or try to understand what factors contributed to its success.
2. If the p-value is greater than 0.05, we fail to reject the null hypothesis, meaning the difference in sales increases between campaigns is not statistically significant, and the company might consider trying other strategies.

example2,

we’ll use two factors (independent variables):

Diet Type: (A, B, C)

Exercise Type: (Cardio, Strength)

In [6]:
```python
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols

# Sample Data: Weight loss by Diet and Exercise Type
data = {
    'Diet': ['A', 'A', 'A', 'A', 'B', 'B', 'B', 'B', 'C', 'C', 'C', 'C', 'A', 'A', 'A', 'A', 'B', 'B', 'B', '
    'Exercise': ['Cardio', 'Cardio', 'Strength', 'Strength', 'Cardio', 'Cardio', 'Strength', 'Strength', 'Car
                 'Cardio', 'Cardio', 'Strength', 'Strength', 'Cardio', 'Cardio', 'Strength', 'Strength', 'Car
    'WeightLoss': [10, 12, 14, 11, 8, 10, 9, 7, 15, 13, 16, 14, 11, 10, 12, 13, 7, 8, 10, 9, 13, 15, 14, 16]
}

# Convert the data to a DataFrame
df = pd.DataFrame(data)

df.head()
```

Out[6]:

	Diet	Exercise	WeightLoss
0	A	Cardio	10
1	A	Cardio	12
2	A	Strength	14
3	A	Strength	11
4	B	Cardio	8

In [7]:

```
# Perform two-way ANOVA, ordinary least square
model = ols('WeightLoss ~ C(Diet) + C(Exercise) + C(Diet):C(Exercise)', data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)

# Display the ANOVA table
print(anova_table)
```

```
                         sum_sq    df          F        PR(>F)
C(Diet)              144.083333   2.0  51.356436  3.644772e-08
C(Exercise)            7.041667   1.0   5.019802  3.791188e-02
C(Diet):C(Exercise)    1.583333   2.0   0.564356  5.784700e-01
Residual              25.250000  18.0        NaN           NaN
```

In [ ]:

In [9]:

```
Requirement already satisfied: seaborn in c:\users\hp\anaconda3\lib\site-packages (0.11.2)
Requirement already satisfied: pandas>=0.23 in c:\users\hp\anaconda3\lib\site-packages (from seaborn) (1.4.
4)
Requirement already satisfied: numpy>=1.15 in c:\users\hp\anaconda3\lib\site-packages (from seaborn) (1.21.
5)
Requirement already satisfied: scipy>=1.0 in c:\users\hp\anaconda3\lib\site-packages (from seaborn) (1.9.1)
Requirement already satisfied: matplotlib>=2.2 in c:\users\hp\anaconda3\lib\site-packages (from seaborn) (3.
5.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib
>=2.2->seaborn) (4.25.0)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib>=2.2
->seaborn) (0.11.0)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib>=
2.2->seaborn) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib
>=2.2->seaborn) (1.4.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\anaconda3\lib\site-packages (from matplot
lib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib>=2.
2->seaborn) (9.2.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib>
=2.2->seaborn) (3.0.9)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=0.23->s
eaborn) (2022.1)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-packages (from python-dateutil>=2.
7->matplotlib>=2.2->seaborn) (1.16.0)
```

In [ ]: