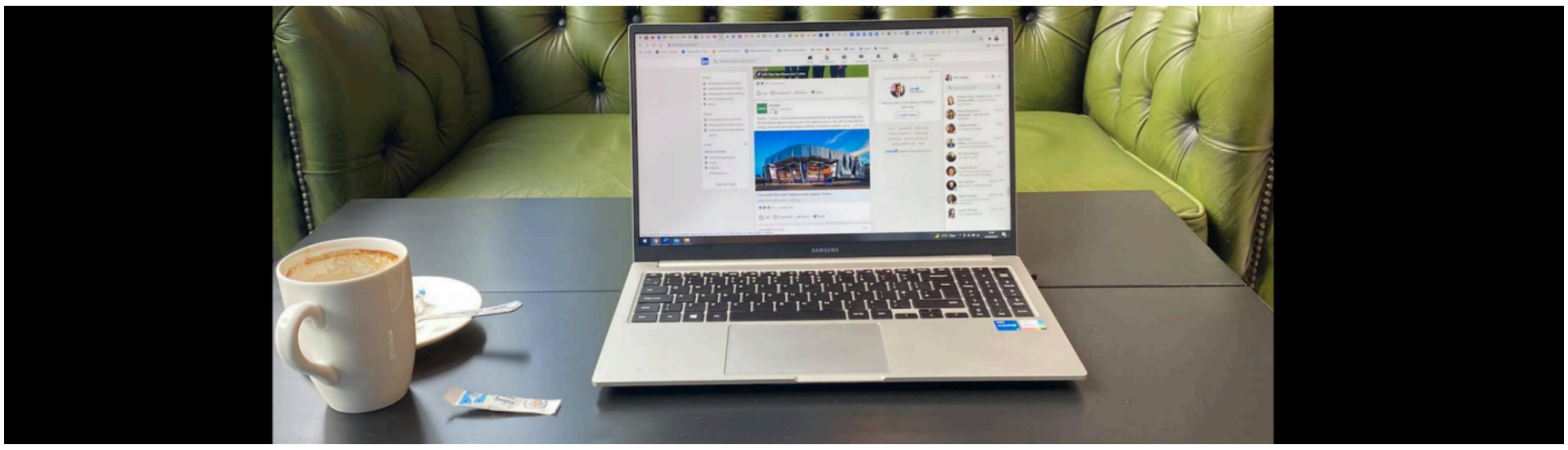
CHAPTER 3 - STANDARD SCORES



Topic: Z Score, Normal Distribution, Standard Normal Distribution & Probability

In [1]:
```
import math
import statistics as st
import random
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import yfinance as yf
%matplotlib inline
from scipy.stats import norm
```

Hemant Thapa

1. Comparing scores from different distributions

A young women name Alice is considering whether to concentrate seriously on athletics. She knows that she is a good athlete and trains regularly at a large athletics club. She is able to observe the elite performance in a number of disciplines at the club and believes that she might be good enough to join their ranks in the 400 meter or the high jump. her best performance was 61.20 seconds in the 400 metres and 1 metre 35 centimetres in the high jump.

The question she would like to answer is whether these scores can be used to provide her with information on how good she is at these events compared to the club's best athletes. To answer this question, she has to compare to the club best athletes.

She learn that average performance (the mean) of the elite women at the club is 60 second in 400 metre runner and 1.50 metre for high jumps.

In both case, she is not good as average elite performer, 1.2 seconds slower in the 400 metres and 15 centimetres shorter in the high jump. The problem is how to decide if 1.2 second in the 400 metres ios the better than 15 centimeters in the high jump.

To compare two scores that come from different distribution we need to standardise them. we do this by calculating as statistic called, not suprisingly, a standard score (or z score)

STANDARD SCORE

The position of a score within a distribution of scores. It provides a measure of how many standard deviation units a specific score fails above or below the mean. It is also referred to as a z score.

The Z-score measures how many standard deviations a data point is from the mean of its distribution. By converting the scores to Z-scores, you can compare them directly, regardless of the original distribution.

$z = (x-\mu)/\sigma$

x : Data Points

μ : Mean

σ : Standard Deviation

In [2]:
```
def calculate_z_score(data_point, mean, std_dev):
    z_score = (data_point - mean) / std_dev
    return z_score
```

In [3]:
```
data_point = 61.2
mean = 60
```

```
std_dev = 3

z_score_result = calculate_z_score(data_point, mean, std_dev)
print(f"The Z-score for the data point {data_point} is {z_score_result:.2f}")
```

```
The Z-score for the data point 61.2 is 0.40
```

In [4]:

```
data_point = 1.35
mean = 1.50
std_dev = 0.15

z_score_result = calculate_z_score(data_point, mean, std_dev)
print(f"The Z-score for the data point {data_point} is {z_score_result:.2f}")
```

```
The Z-score for the data point 1.35 is -1.00
```

Explanation

1. Standardization: The Z-score standardizes the data, transforming it into a common scale where the mean is 0 and the standard deviation is 1. This process allows us to compare and interpret data points more easily.
2. Positive and Negative Z-scores: A positive Z-score indicates that the data point is above the mean, while a negative Z-score indicates that the data point is below the mean.
3. Magnitude of Z-score: The magnitude of the Z-score tells us how far away a data point is from the mean in terms of standard deviations. For example, a Z-score of 2 means the data point is 2 standard deviations above the mean.
4. Interpretation: A Z-score of 0 means the data point is exactly at the mean. Positive Z-scores indicate above-average values, while negative Z-scores indicate below-average values.

Usage of Z-score

1. Outlier Detection: Z-scores can help identify outliers in a dataset. Data points with Z-scores far from 0 (typically greater than 3 or less than -3) are considered outliers.
2. Comparisons: Z-scores allow us to compare data from different distributions. It helps us determine whether a data point is relatively high or low compared to other data points in the dataset.
3. Standard Normal Distribution: Z-scores play a crucial role in the standard normal distribution (normal distribution with mean 0 and standard deviation 1). They help find the probability of a value occurring within a specific range using the standard normal distribution table or statistical software.
4. Data Transformation: Z-score transformation is used in data preprocessing and normalization, especially in machine learning algorithms.

In [5]:

```
exam_scores = [85, 90, 78, 92, 88, 95, 80, 85, 87, 90]
mean_score = np.mean(exam_scores)
std_deviation = np.std(exam_scores)
z_scores = [(score - mean_score) / std_deviation for score in exam_scores]
print("Z-scores:", z_scores)
```

```
Z-scores: [-0.4032389192727559, 0.6048583789091339, -1.8145751367274017, 1.0080972981818899, 0.20161945963637795, 1.61
29556770910236, -1.4113362174546458, -0.4032389192727559, 0.0, 0.6048583789091339]
```

EXAMPLE

Let's say we have a dataset of exam scores from a particular class, and we want to calculate the Z-score for a specific student's score. The Z-score measures how many standard deviations a data point is away from the mean of the dataset.

X is the individual data point (student's exam score),

μ is the mean of the dataset, and

σ is the standard deviation of the dataset.

Scores: 85, 90, 78, 92, 88, 95, 80, 85, 87, 90

In [6]:

```
scores = [85, 90, 78, 92, 88, 95, 80, 87, 90]
scores
```

Out[6]:

```
[85, 90, 78, 92, 88, 95, 80, 87, 90]
```

In [7]:

```
score_mean = sum(scores)/len(scores)
score_mean = math.ceil(score_mean)
score_mean
```

Out[7]:

```
88
```

In [8]:

```
deviation = []
for i in range(len(scores)):
    deviation.append(scores[i] - score_mean)
```

In [9]:

```
deviation
```

Out[9]:

```
[-3, 2, -10, 4, 0, 7, -8, -1, 2]
```

In [10]:

```
deviation_square = []
for i in range(len(deviation)):
    deviation_square.append((deviation[i])**2)
```

In [11]:

```
deviation_square
```

Out[11]:

```
[9, 4, 100, 16, 0, 49, 64, 1, 4]
```

In [12]:

```
Standard_deviation = math.sqrt(sum(deviation_square)/len(deviation_square))
```

In [13]:

```
Standard_deviation
```

Out[13]:

```
5.23874454850057
```

Now, let's say we want to calculate the Z-score for a student who scored 90 on the exam.

In [14]:

```
data_point = 90
mean = 88
std_dev = Standard_deviation

z_score_result = calculate_z_score(data_point, mean, std_dev)
print(f"The Z-score for the data point {data_point} is {z_score_result:.2f}")
```

```
The Z-score for the data point 90 is 0.38
```

The Z-score for a score of 90 is approximately 0.05. A positive Z-score indicates that the student's score is above the mean of the dataset, while a negative Z-score would indicate a score below the mean. A Z-score close to 0 indicates that the student's score is close to the mean.

2. Normal Distribution

Normal distribution, also known as the Gaussian distribution, is one of the most important and widely used probability distributions in statistics and probability theory. It describes a continuous, symmetric probability distribution that is characterized by a bell-shaped curve. The curve is symmetric around its mean, with the majority of the data clustered around the mean, and tails that extend infinitely in both directions.

Key characteristics of a normal distribution:

1. Bell-shaped curve: The curve is symmetrical, meaning the left and right halves are mirror images of each other. The highest point of the curve represents the mean (average) of the data.

2. Mean, median, and mode coincide: In a normal distribution, the mean, median, and mode are all equal and located at the center of the curve.

3. 68-95-99.7 rule: This rule, also known as the empirical rule, states that approximately 68% of the data falls within one standard deviation of the mean, about 95% within two standard deviations, and roughly 99.7% within three standard deviations.

4. Continuous and unbounded: The normal distribution extends indefinitely in both directions without ever touching the x-axis. It covers the entire real number line.

5. Standard deviation determines spread: The spread or dispersion of data is determined by the standard deviation. A smaller standard deviation results in a taller and narrower curve, while a larger standard deviation leads to a flatter and wider curve.

In [15]:

```
np.random.seed(42)
mean_height = 170
std_deviation = 10
num_people = 1000
```

In [16]:

```
# Generate random height data
heights = np.random.normal(loc=mean_height, scale=std_deviation, size=num_people)
```

In [17]:

```
heights[:10]
```

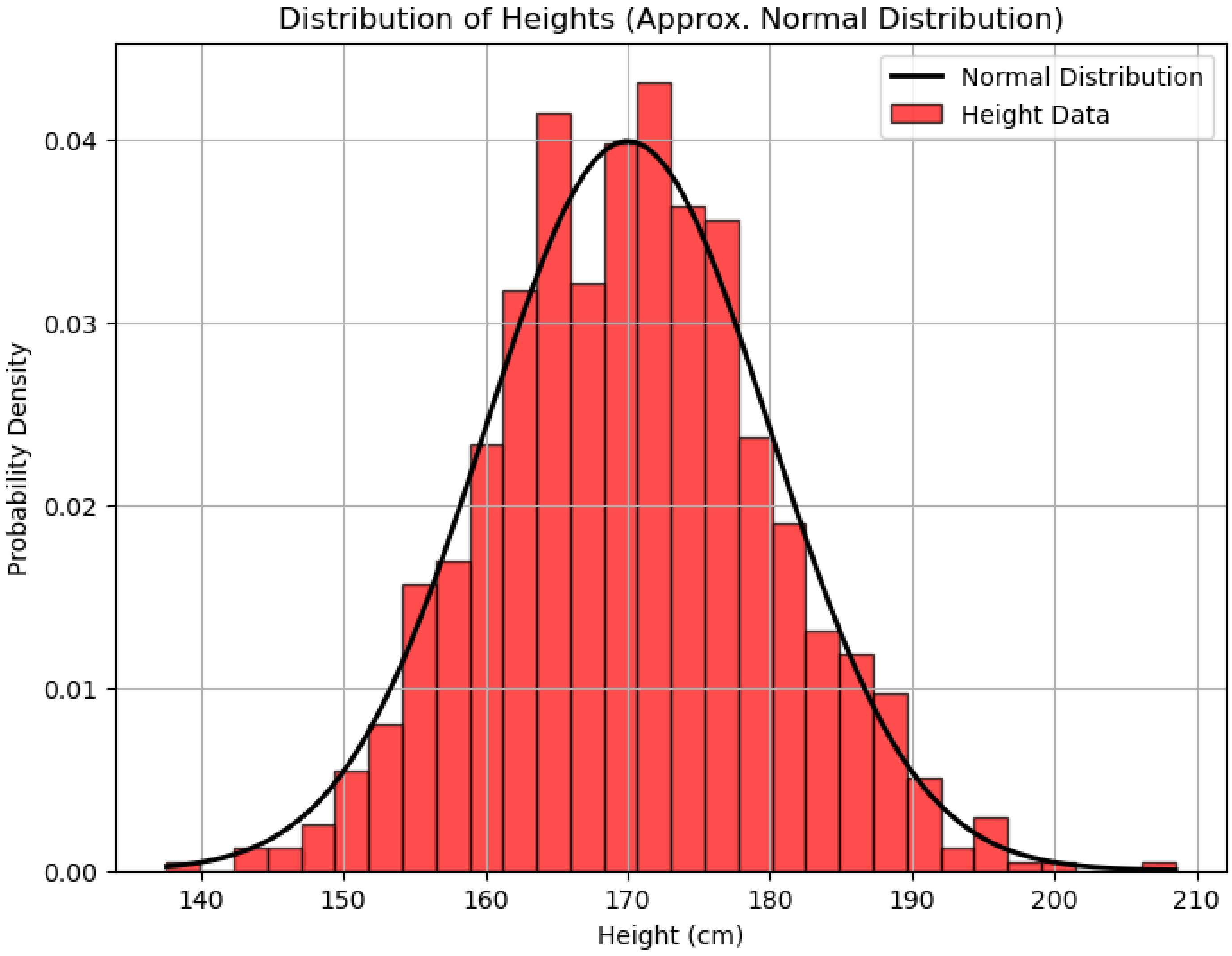
Out[17]:

```
array([174.96714153, 168.61735699, 176.47688538, 185.23029856,
       167.65846625, 167.65863043, 185.79212816, 177.67434729,
       165.30525614, 175.42560044])
```

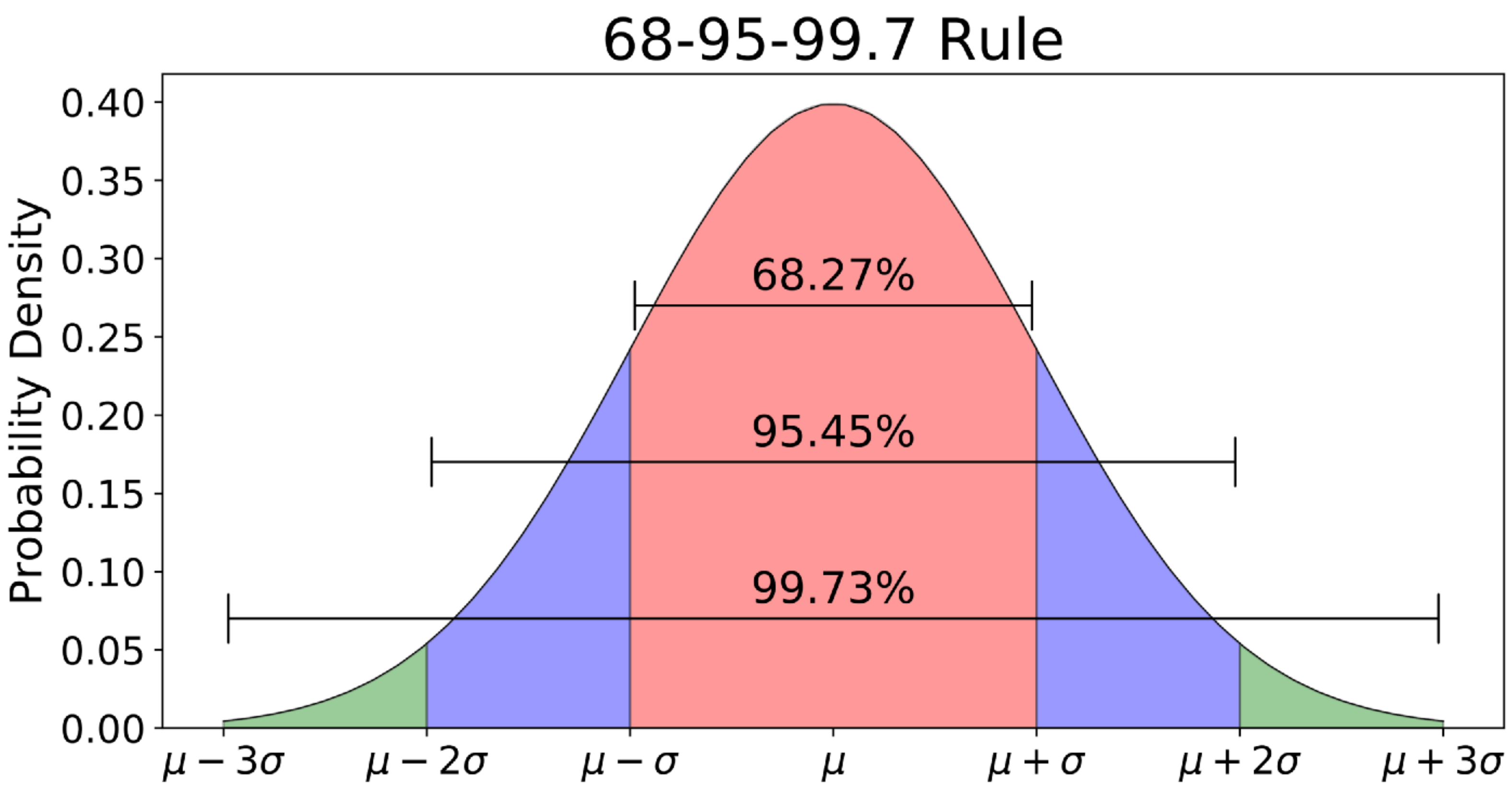
In [18]:

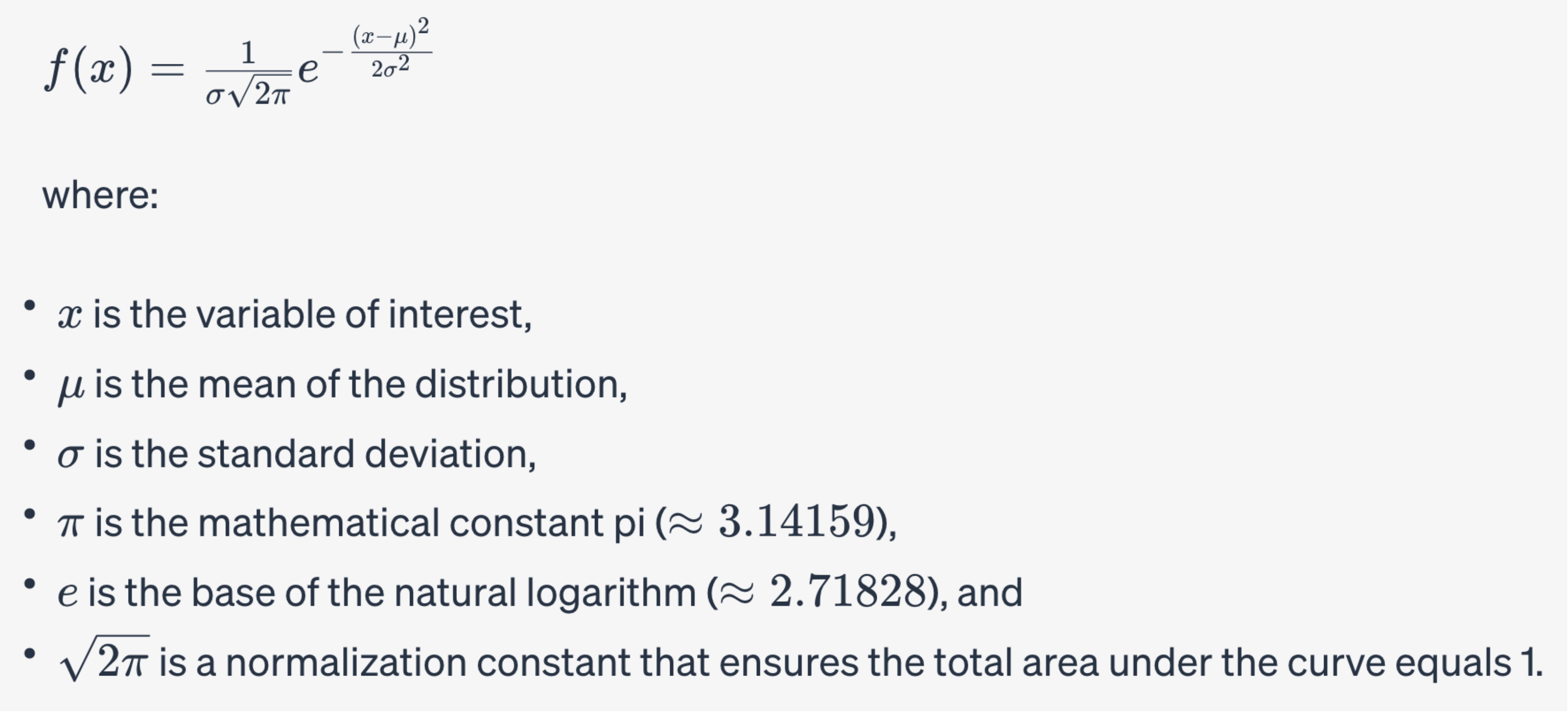
```
plt.figure(figsize=(8, 6))
plt.hist(heights, bins=30, density=True, color='red', edgecolor='black', alpha=0.7)
x = np.linspace(min(heights), max(heights), 100)
pdf = norm.pdf(x, loc=mean_height, scale=std_deviation)
plt.plot(x, pdf, color='black', linewidth=2)
```

```python
plt.xlabel('Height (cm)')
plt.ylabel('Probability Density')
plt.title('Distribution of Heights (Approx. Normal Distribution)')
plt.grid(True)
plt.legend(['Normal Distribution', 'Height Data'])
plt.show()
```



The probability density function (PDF) of a normal distribution is given by the formula





$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

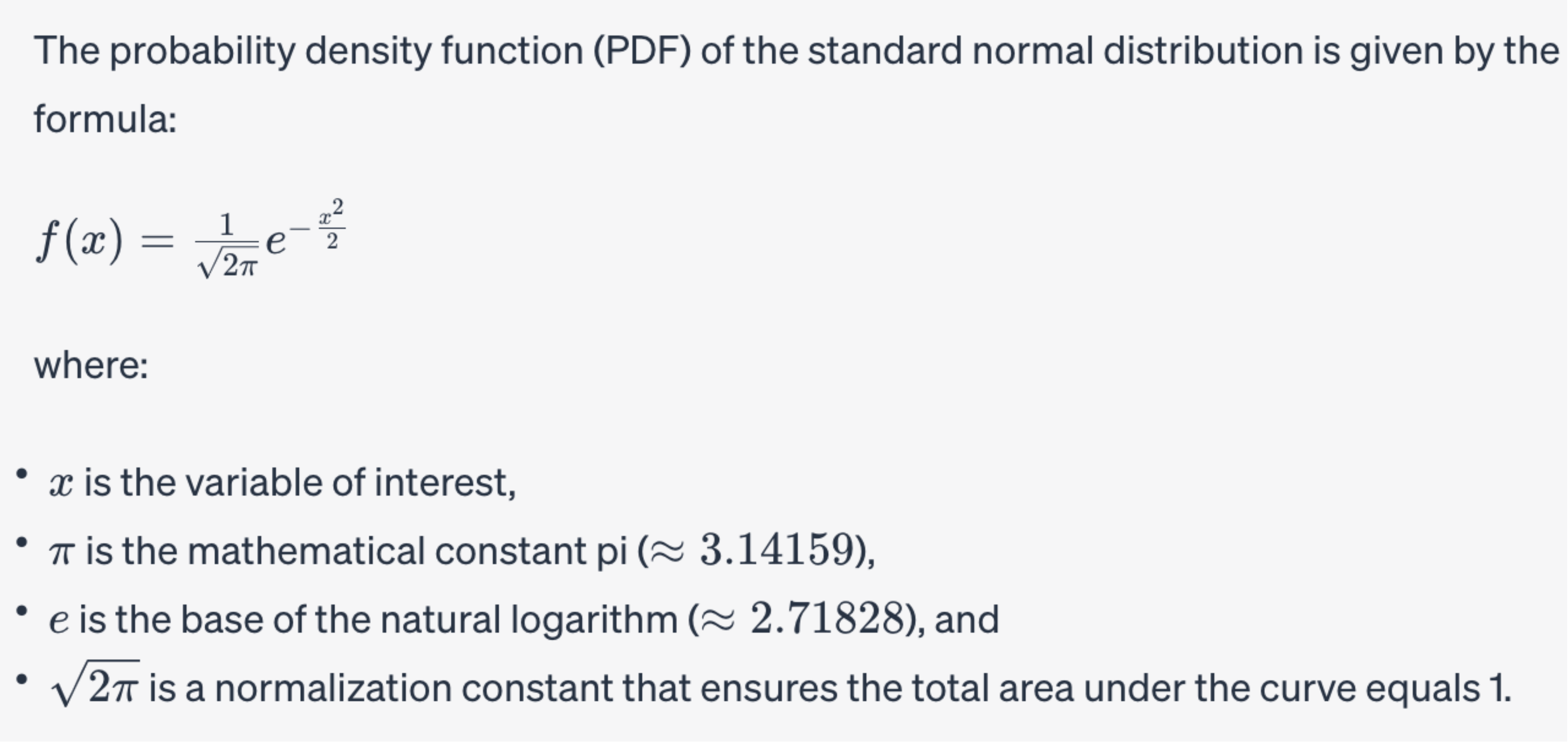
where:

- x is the variable of interest,
- μ is the mean of the distribution,
- σ is the standard deviation,
- π is the mathematical constant pi (≈ 3.14159),
- e is the base of the natural logarithm (≈ 2.71828), and
- $\sqrt{2\pi}$ is a normalization constant that ensures the total area under the curve equals 1.

This formula describes the shape of the bell-shaped curve for a normal distribution, where the mean μ represents the center of the curve, and the standard deviation σ controls the spread or width of the curve. The PDF provides the relative likelihood of observing a particular value x in a normal distribution. The higher the PDF value at a specific point, the more likely that value is to occur in the distribution.

3. Standard Normal Distribution

The standard normal distribution, also known as the Z-distribution or Gaussian distribution, is a specific type of normal distribution with a mean (μ) of 0 and a standard deviation (σ) of 1. It is a fundamental and essential concept in statistics and probability theory.



The probability density function (PDF) of the standard normal distribution is given by the formula:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

where:

- x is the variable of interest,
- π is the mathematical constant pi (≈ 3.14159),
- e is the base of the natural logarithm (≈ 2.71828), and
- $\sqrt{2\pi}$ is a normalization constant that ensures the total area under the curve equals 1.

Key properties of the standard normal distribution:

1. Symmetry: The standard normal distribution is symmetric around its mean of 0. This means that the curve is perfectly centered at x=0 and is mirror-symmetric on both sides of the mean.

1. Bell-shaped curve: The PDF of the standard normal distribution follows a bell-shaped curve, similar to other normal distributions. The curve reaches its maximum at x=0 and tails off to both sides.

1. Total area under the curve: The total area under the standard normal curve is equal to 1, representing 100% probability. The area under the curve between any two points a and b represents the probability of a randomly selected value falling within that interval.

1. Z-scores: Z-scores, also known as standard scores, are a measure of how many standard deviations a data point is away from the mean. For any value x in the standard normal distribution, the Z-score (Z) can be calculated as Z = (x−μ)/σ =x.

1. 68-95-99.7 rule: The standard normal distribution follows the same empirical rule as other normal distributions. Approximately 68% of the data falls within one standard deviation of the mean (Z∣≤1), about 95% within two standard deviations (Z∣≤2), and roughly 99.7% within three standard deviations (∣Z∣≤3).

In [19]:

```
np.random.seed(42)
num_data_points = 1000
standard_normal_data = np.random.standard_normal(num_data_points)
```

In [20]:

```
standard_normal_data[:10]
```

Out[20]:

```
array([ 0.49671415, -0.1382643 ,  0.64768854,  1.52302986, -0.23415337,
       -0.23413696,  1.57921282,  0.76743473, -0.46947439,  0.54256004])
```

In [21]:

```
mean_data = np.mean(standard_normal_data)
std_dev_data = np.std(standard_normal_data)
z_scores = (standard_normal_data - mean_data) / std_dev_data
```

In [22]:

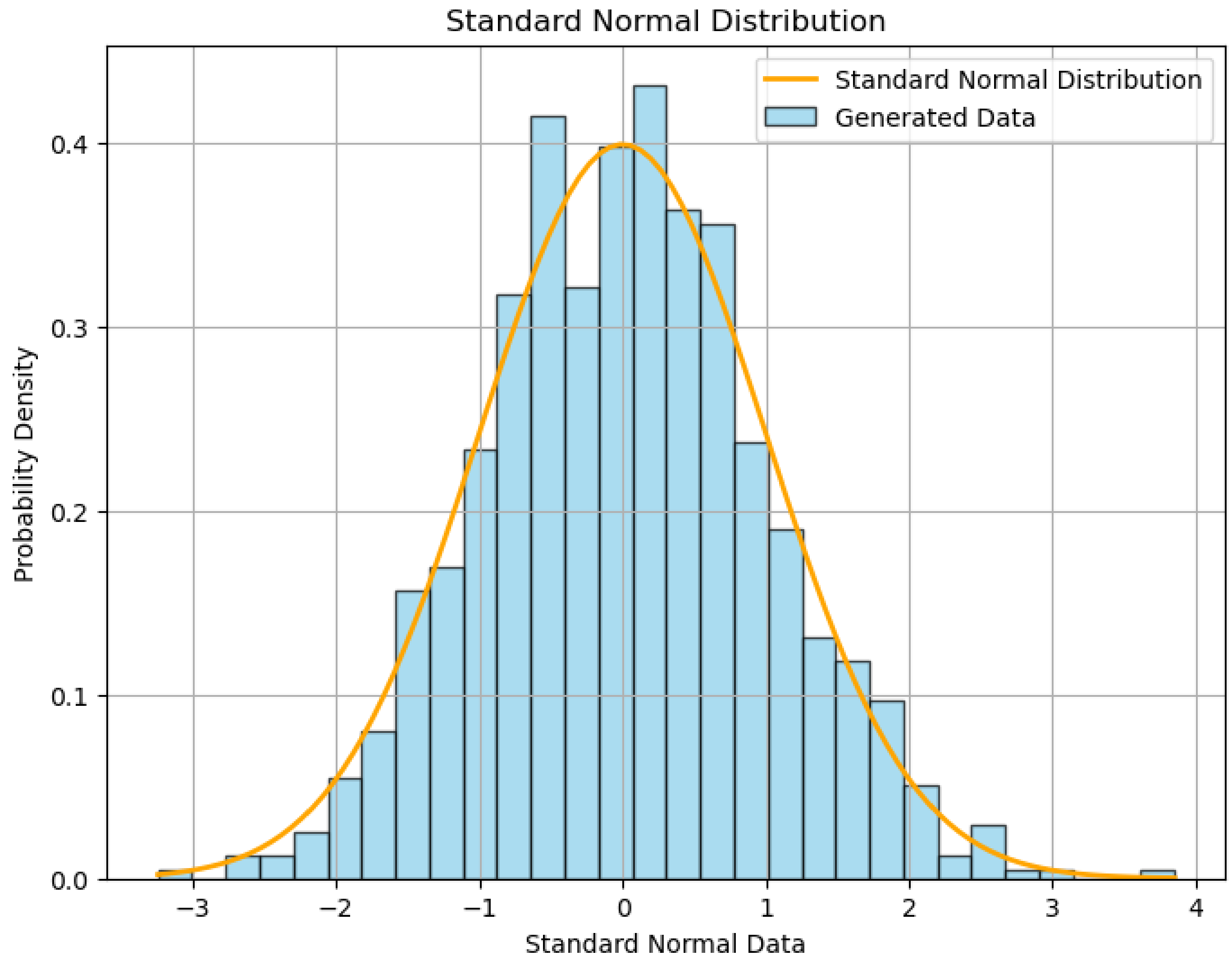
```
z_scores[:10]
```

Out[22]:

```
array([ 0.48775857, -0.1610219 ,  0.64201457,  1.53638248, -0.25899524,
       -0.25897847,  1.59378665,  0.76436359, -0.49943124,  0.53460098])
```

In [23]:

```
plt.figure(figsize=(8, 6))
plt.hist(standard_normal_data, bins=30, density=True, color='skyblue', edgecolor='black', alpha=0.7)
x = np.linspace(min(standard_normal_data), max(standard_normal_data), 100)
pdf = norm.pdf(x, loc=0, scale=1)
plt.plot(x, pdf, color='orange', linewidth=2)
plt.xlabel('Standard Normal Data')
plt.ylabel('Probability Density')
plt.title('Standard Normal Distribution')
plt.grid(True)
plt.legend(['Standard Normal Distribution', 'Generated Data'])
plt.show()
```



4. Probability

Probability is a fundamental concept in mathematics and statistics that measures the likelihood or chance of an event occurring. It quantifies the uncertainty associated with random phenomena and helps us understand and predict the outcomes of uncertain events. In simple terms, probability answers the question, "What is the chance of something happening?"

In the context of probability theory, an event is any specific outcome or collection of outcomes of an experiment or random process. The probability of an event is a real number between 0 and 1, where 0 indicates that the event is impossible, and 1 indicates that the event is certain to occur.

The probability of an event A is denoted as P(A) and can be calculated using various methods depending on the nature of the experiment or random process. There are two main types of probability:

1. Classical Probability: In situations where all outcomes are equally likely, classical probability can be used. For example, when flipping a fair coin, the probability of getting heads or tails is 0.5 each, as there are only two equally likely outcomes.
2. Empirical (Experimental) Probability: In real-world scenarios or experiments, empirical probability is used. It is based on observations and data from actual experiments. For example, the probability of a basketball player making a free throw can be estimated by observing the player's historical success rate in making free throws.

Here are some key applications of probability in the standard normal distribution:

1. Calculating Probabilities: The standard normal distribution is used to calculate probabilities associated with specific Z-scores or intervals of Z-scores. For example, given a Z-score, you can find the probability of a data point falling to the left or right of that Z-score using the cumulative distribution function (CDF).
2. Finding Critical Values: Critical values, such as the Z-score that corresponds to a certain tail probability, are important in hypothesis testing and confidence intervals. By knowing the critical values, you can determine the boundaries for acceptance or rejection regions in hypothesis testing.
3. Standardization of Data: The standard normal distribution provides a way to transform data from different normal distributions into a common standard scale. By converting raw data to Z-scores, we can compare and interpret data points from different datasets with different means and standard deviations.
4. Z-Tests and T-Tests: The Z-test and t-test are hypothesis tests that use the standard normal distribution (or the t-distribution for small sample sizes) to compare sample means with population means or to compare two sample means. These tests help determine whether the differences between groups are statistically significant.
5. Confidence Intervals: Confidence intervals for population parameters, such as the population mean, are calculated using the standard normal distribution. These intervals provide a range within which the true population parameter is likely to lie with a certain level of confidence.
6. Estimation of Percentiles and Percentile Ranks: The standard normal distribution allows us to estimate percentiles (e.g., 90th percentile) and percentile ranks for specific data values.
7. Monte Carlo Simulations: The standard normal distribution is widely used in Monte Carlo simulations for generating random numbers and performing probabilistic simulations.

In [24]:

```
your_height = 180 #cm
```

In [25]:

```
# Mean and standard deviation of the population (hypothetical values)
mean_height = 170
std_deviation = 10
```

In [26]:

```
# Calculate the Z-score for your height
z_score = (your_height - mean_height) / std_deviation
z_score
```

Out[26]:

```
1.0
```

In [27]:

```
# Calculate the probability of being taller than you
fraction_taller_than_you = 1 / 7
probability_taller_than_you = 1 - fraction_taller_than_you
```

In [28]:

```
fraction_taller_than_you
```

Out[28]:

```
0.14285714285714285
```

In [29]:

```
probability_taller_than_you
```

Out[29]:

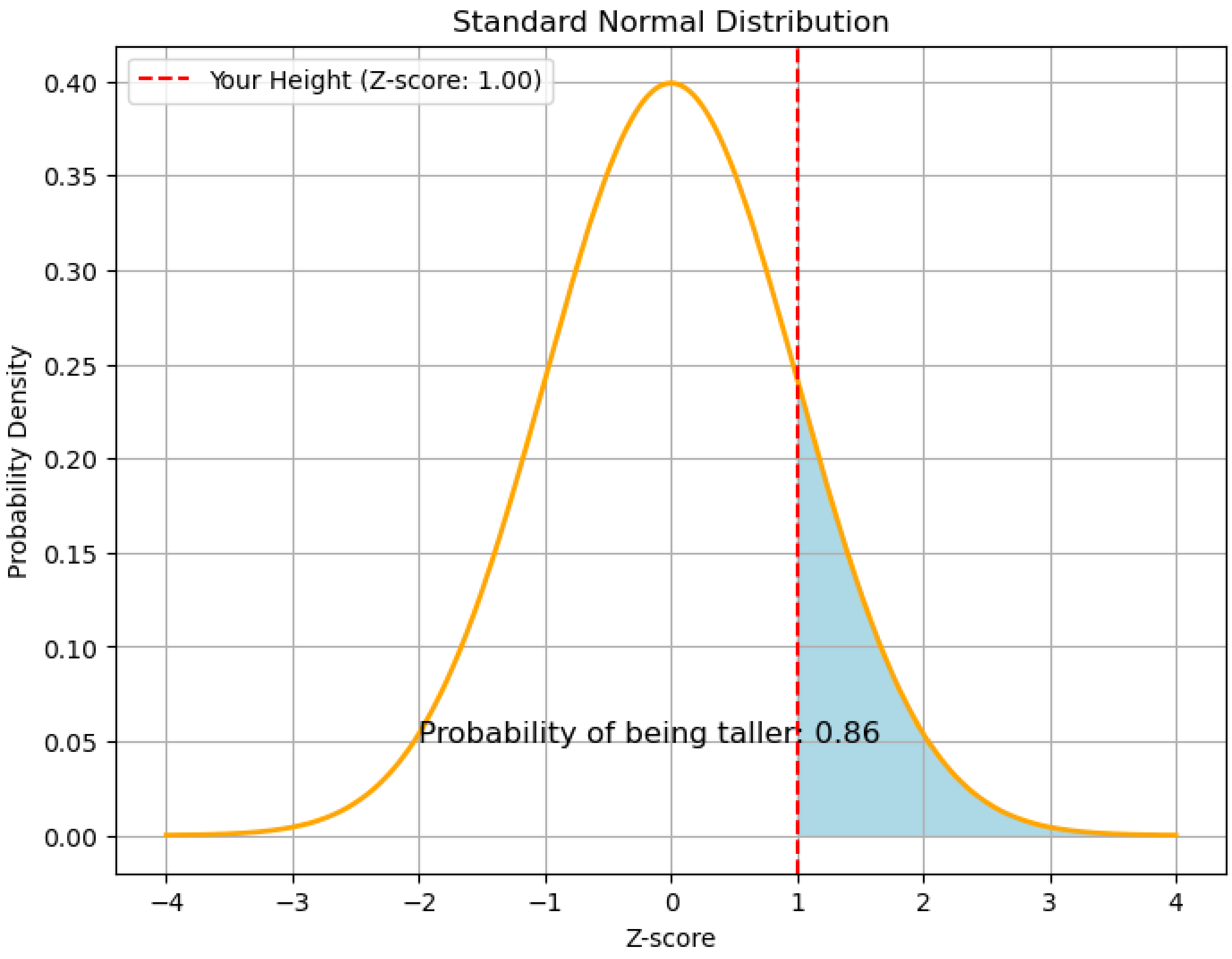
```
0.8571428571428572
```

In [30]:

```
plt.figure(figsize=(8, 6))
x = np.linspace(-4, 4, 1000)  # Range of Z-scores from -4 to 4
pdf = norm.pdf(x, loc=0, scale=1)
plt.plot(x, pdf, color='orange', linewidth=2)

shade_x = np.linspace(z_score, 4, 1000)  # From your Z-score to the end of the curve
shade_y = norm.pdf(shade_x, loc=0, scale=1)
plt.fill_between(shade_x, shade_y, color='lightblue')

plt.xlabel('Z-score')
plt.ylabel('Probability Density')
plt.title('Standard Normal Distribution')
plt.axvline(x=z_score, color='red', linestyle='--', label=f'Your Height (Z-score: {z_score:.2f})')
plt.text(-2, 0.05, f'Probability of being taller: {probability_taller_than_you:.2f}', fontsize=12)
plt.grid(True)
plt.legend()
plt.show()
```



5. Z Score Usage in Finance

Z-scores are commonly used in finance for various purposes, including risk management, portfolio optimization, credit analysis, and detecting outliers. Here are some specific applications of Z-scores in finance:

1. Credit Risk Assessment: In credit analysis, Z-scores (also known as bankruptcy prediction models) are used to assess the financial health and creditworthiness of a company. Z-scores are calculated based on financial ratios and provide a measure of the probability of a firm going bankrupt. Lower Z-scores indicate higher bankruptcy risk.

2. Portfolio Optimization: Z-scores are used in portfolio management to standardize and compare the performance of individual assets or securities. By calculating Z-scores for each asset's returns, investors can identify the relative performance of each asset and make more informed decisions about asset allocation and diversification.

3. Value-at-Risk (VaR) Calculation: Value-at-Risk is a measure of potential losses in an investment portfolio due to adverse market movements. Z-scores are employed to estimate VaR by converting historical return data into standardized Z-scores, representing the probability of different levels of losses.

4. Volatility Analysis: In finance, Z-scores are utilized to assess the volatility of a financial asset or market index. By standardizing returns using Z-scores, analysts can compare the volatility of different assets or market segments and identify periods of unusually high or low volatility.

5. Detecting Outliers: Z-scores are valuable for identifying outliers or extreme values in financial data. By calculating Z-scores for individual data points, analysts can flag data that deviates significantly from the mean, which might indicate erroneous or unusual market events.

6. Trading Strategies: Traders use Z-scores in pairs trading strategies, where they identify pairs of assets that historically move together. Z-scores help identify periods when the price relationship between the two assets diverges significantly from historical norms, potentially providing trading opportunities.

7. Hedging Strategies: Z-scores are employed in hedging strategies to determine when an asset's price deviates significantly from its historical relationship with another asset. This can help traders decide when to initiate or unwind hedging positions.

```
In [31]: import yfinance as yf
```

```
In [32]: ticker_symbols = ['AAPL', 'MSFT']
```

```
In [33]: data = yf.download(ticker_symbols, start='2020-01-01', end='2023-07-01')['Adj Close']
```

```
[*********************100%***********************]  2 of 2 completed
```

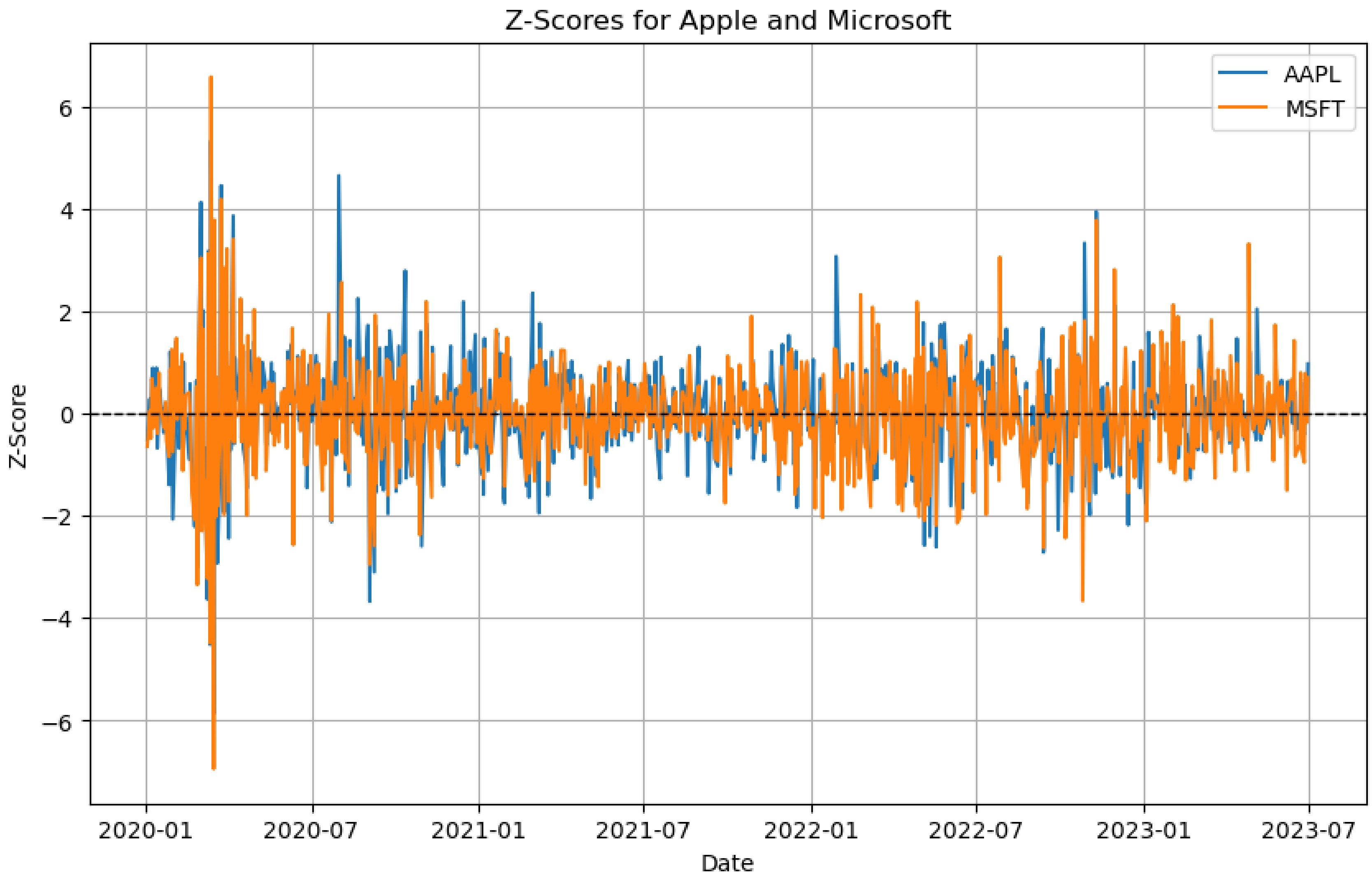
```
In [34]: returns = data.pct_change().dropna()
```

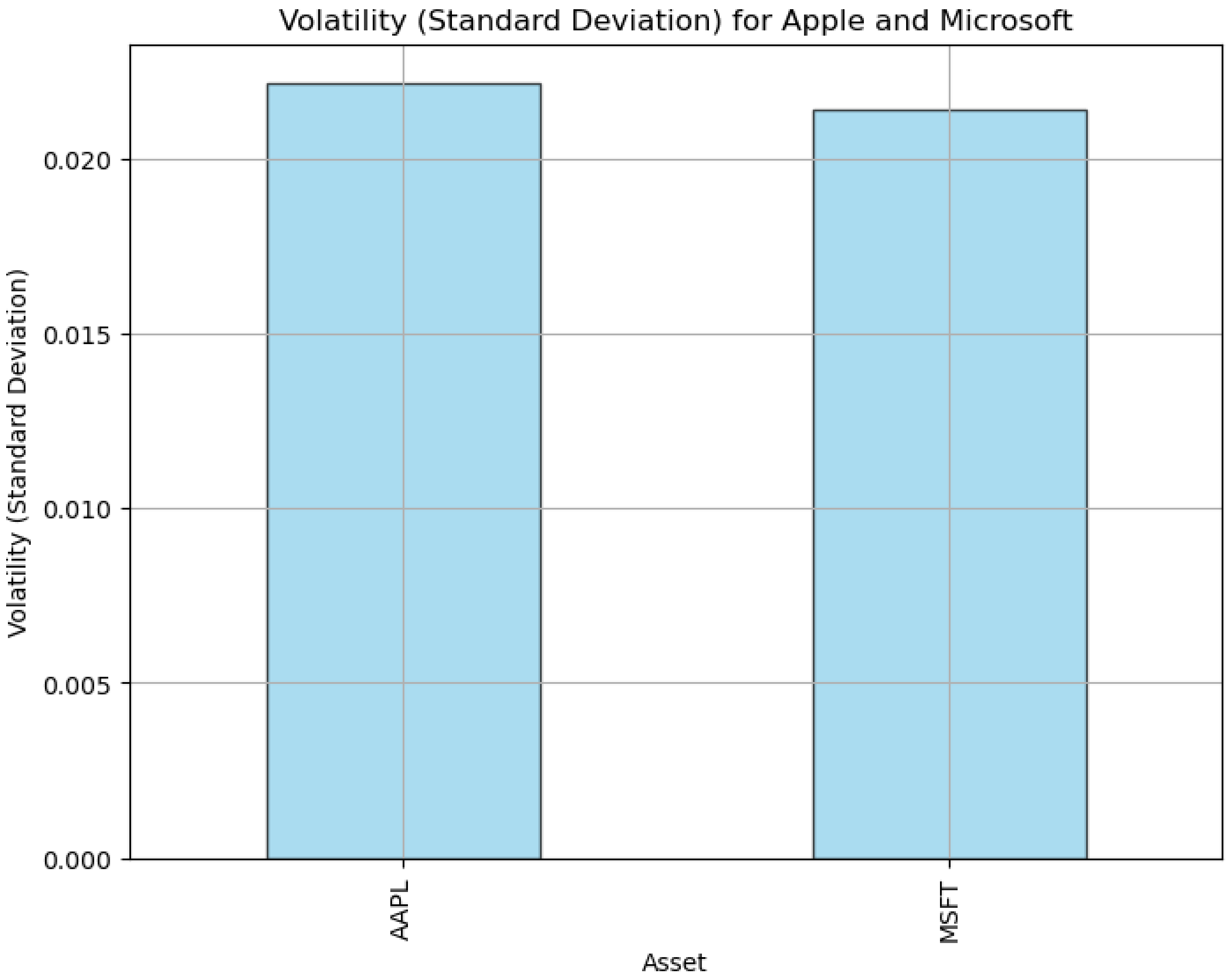
```
In [35]: # Calculate Z-scores for each asset's daily returns
z_scores = (returns - returns.mean()) / returns.std()
# Calculate volatility (standard deviation) for each asset
volatility = returns.std()
```

In [36]:

```
plt.figure(figsize=(10, 6))
for symbol in ticker_symbols:
    plt.plot(z_scores.index, z_scores[symbol], label=symbol)
plt.axhline(0, color='black', linestyle='--', linewidth=1)
plt.title('Z-Scores for Apple and Microsoft')
plt.xlabel('Date')
plt.ylabel('Z-Score')
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(8, 6))
volatility.plot(kind='bar', color='skyblue', edgecolor='black', alpha=0.7)
plt.title('Volatility (Standard Deviation) for Apple and Microsoft')
plt.xlabel('Asset')
plt.ylabel('Volatility (Standard Deviation)')
plt.grid(True)
plt.show()
```





In [37]:

```
data = yf.download(ticker_symbols, start='2020-01-01', end='2023-07-01')['Adj Close']
returns = data.pct_change().dropna()
```

```
# Combine returns for the two assets into a single portfolio
portfolio_returns = returns.dot(np.array([0.5, 0.5]))
```

```
[*********************100%***********************]  2 of 2 completed
```

In [46]:
```
returns.tail(10)
```

Out[46]:

Date	AAPL	MSFT
2023-06-16	-0.005860	-0.016576
2023-06-20	0.000487	-0.012503
2023-06-21	-0.005675	-0.013282
2023-06-22	0.016525	0.018437
2023-06-23	-0.001711	-0.013806
2023-06-26	-0.007553	-0.019163
2023-06-27	0.015059	0.018168
2023-06-28	0.006328	0.003826
2023-06-29	0.001797	-0.002382
2023-06-30	0.023103	0.016386

In [49]:
```
portfolio_returns[:10]
```

Out[49]:
```
Date
2020-01-03   -0.011087
2020-01-06    0.005277
2020-01-07   -0.006910
2020-01-08    0.016007
2020-01-09    0.016867
2020-01-10   -0.001183
2020-01-13    0.016694
2020-01-14   -0.010273
2020-01-15    0.001095
2020-01-16    0.015425
dtype: float64
```

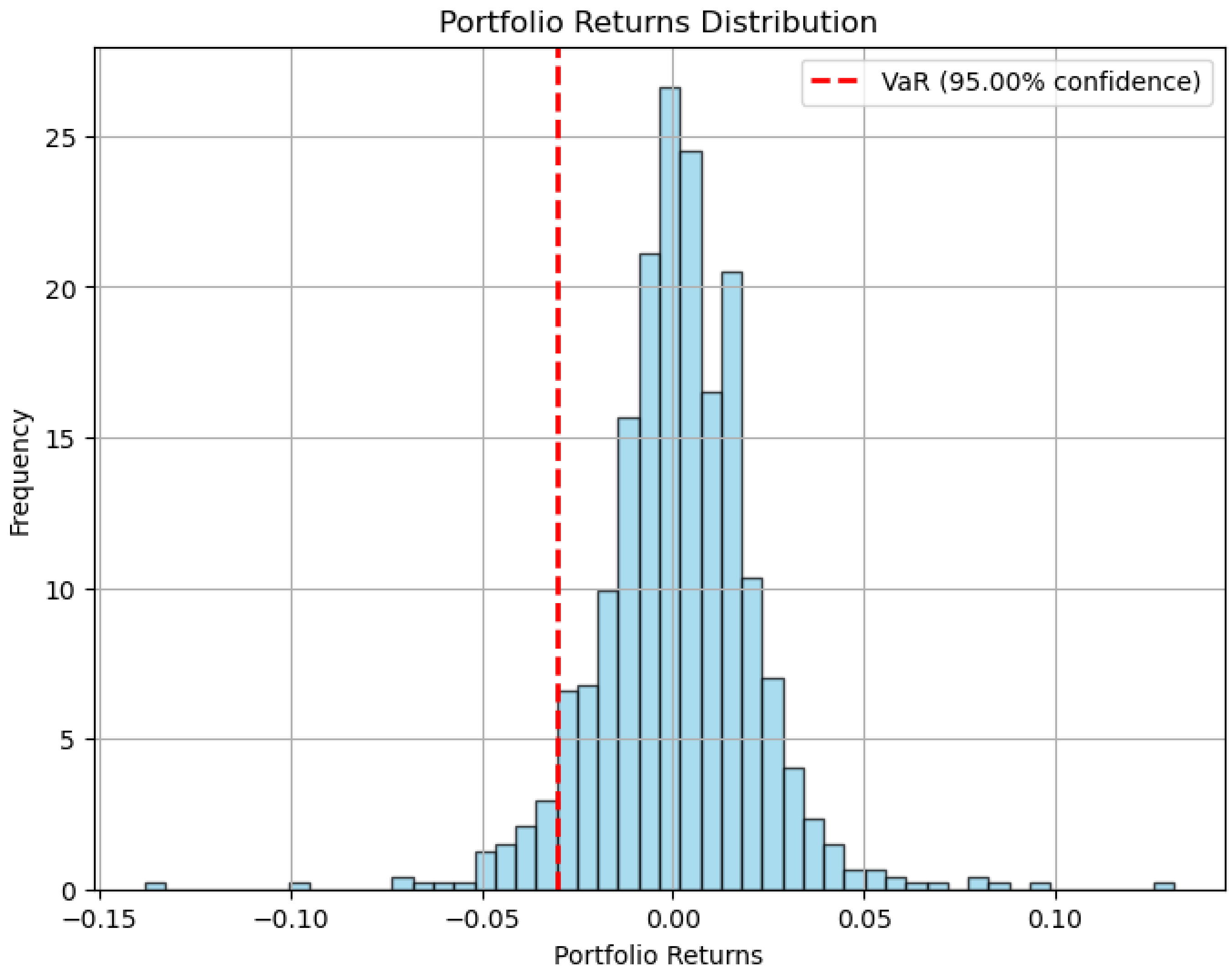
In [38]:
```
# Calculate portfolio VaR at 95% confidence level for a 1-day time horizon
confidence_level = 0.95
time_horizon_days = 1
```

In [39]:
```
# Calculate the percentile corresponding to the confidence level
portfolio_var = np.percentile(portfolio_returns, 100 * (1 - confidence_level))
```

In [40]:
```
print(f"Portfolio VaR at {confidence_level*100:.2f}% confidence level for {time_horizon_days}-day time horizon:")
print(f"{portfolio_var:.4f}")
```

```
Portfolio VaR at 95.00% confidence level for 1-day time horizon:
-0.0304
```

In [41]:
```
plt.figure(figsize=(8, 6))
plt.hist(portfolio_returns, bins=50, density=True, alpha=0.7, color='skyblue', edgecolor='black')
plt.axvline(portfolio_var, color='red', linestyle='--', linewidth=2, label=f'VaR ({confidence_level*100:.2f}% confiden
plt.xlabel('Portfolio Returns')
plt.ylabel('Frequency')
plt.title('Portfolio Returns Distribution')
plt.legend()
plt.grid(True)
plt.show()
```



The output "Portfolio VaR at 95.00% confidence level for a 1-day time horizon: -0.0304" means the Value at Risk (VaR) for the portfolio of Apple and Microsoft stocks at a 95% confidence level for a 1-day time horizon is approximately -0.0304.

1. VaR: Value at Risk (VaR) is a measure of the maximum potential loss that a portfolio may experience over a specified time horizon and at a certain level of confidence. In this case, the VaR represents the expected maximum loss for the portfolio.

2. 95.00% Confidence Level: The VaR is calculated at the 95% confidence level. This means that there is a 95% probability that the actual portfolio returns will not fall below the VaR value (-0.0304) over the 1-day time horizon. In other words, there is a 5% chance that the portfolio will experience losses greater than the VaR value.

3. 1-day Time Horizon: The VaR is calculated for a 1-day time horizon. This means that the VaR represents the potential loss that the portfolio may experience in a single trading day.

4. VaR Value: -0.0304: The VaR value itself is approximately -0.0304. Since the value is negative, it indicates a potential loss. The magnitude of the VaR (-0.0304) represents the expected loss in terms of portfolio returns. For example, if the portfolio's daily return is normally distributed, the VaR value indicates that there is a 95% probability of not losing more than 3.04% of the portfolio's value in a single trading day.

References:

Simply Psychology - Z-score: The link (https://www.simplypsychology.org/z-score.html) provides an explanation of Z-scores in the context of psychology and research. While the content is psychology-focused, the concepts of Z-scores and their applications are relevant to various fields, including statistics and finance.

Statistics How To - Z-table: The page (https://www.statisticshowto.com/tables/z-table/) offers a detailed explanation of the Z-table, which is a tabulated version of the standard normal distribution. The Z-table provides critical values and probabilities associated with Z-scores, making it a useful reference for probability calculations.

Simply Psychology - Z-table: The link (https://www.simplypsychology.org/z-table.html) is another resource from Simply Psychology, offering a Z-table for reference. It provides critical values corresponding to specific Z-scores for probability calculations.

University of Arizona - Standard Normal Table: The link (https://www.math.arizona.edu/~rsims/ma464/standardnormaltable.pdf) leads to a standard normal table (Z-table) that provides probabilities associated with Z-scores. This table is helpful for determining the probability of specific events in a standard normal distribution.