What is a Probability Distribution?

Referance: https://freedium.cfd/https://news.lunartech.ai/fundamentals-of-statistics-for-data-scientists-and-data-analysts-69d93a05aae7 (https://freedium.cfd/https://news.lunartech.ai/fundamentals-of-statistics-for-data-scientists-and-data-analysts-69d93a05aae7)

A probability distribution describes how probabilities are assigned to different values of a random variable. It tells us the likelihood of different outcomes in an experiment.

There are two main types of probability distributions:

Discrete Probability Distributions (for countable outcomes).

Continuous Probability Distributions (for uncountable outcomes like real numbers).

Three key related concepts associated with probability distributions include:

1. Probability Mass Function (PMF): associated with discrete variables, the PMF gives the probabilities for individual outcomes, for example, the probability of a family having one child, having two children, and so on. Assuming X is a random variable like the number of children in a family, and x is a specific value for X, then the PMF is denoted by F(X)=P[X=x].

2. Probability Density Function (PDF): associated with continuous variables, the PDF describes the likelihood of a value falling within a range, for instance probability of a person’s height falling between 5.5 feet and 6 feet. Following a similar notation, the PDF is denoted by F(X)=P[x<=X<=x’], with x<x’.

3. Cumulative Distribution Function (CDF): it indicates the cumulated probability up to a certain value, for example, the probability of a newborn baby weighing up to 2.5kg. It is denoted by F(x)=P[X≤x].

1.Discrete Probability Distributions

A discrete probability distribution applies to a discrete random variable, which takes finite or countable values (e.g., number of defective products, number of heads in coin flips)

1. Binomial Distribution

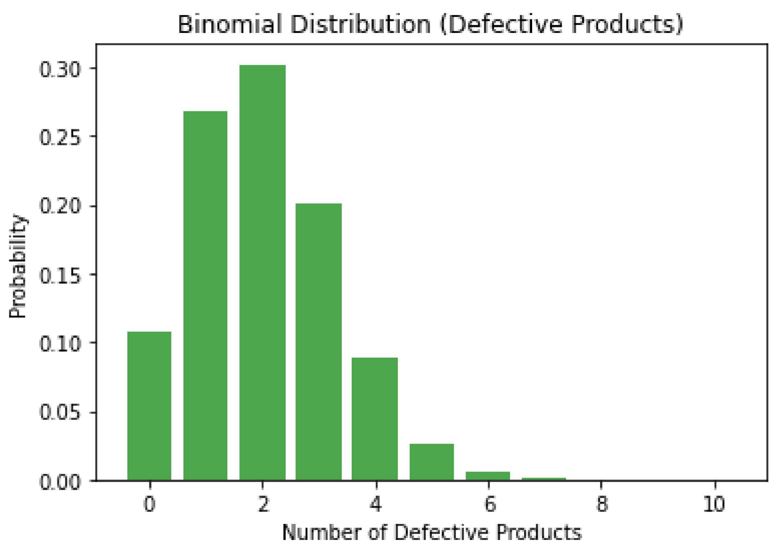
Use Case: It models the number of successes in a fixed number of independent trials, like predicting the probability of defective items in a production batch. Example: Estimating the likelihood of defective products in a factory.

Reference:

In [5]:

```python
from scipy.stats import binom
import matplotlib.pyplot as plt

# Parameters
n = 10  # number of trials (products inspected)
p = 0.2  # probability of a defective product

# Generate binomial distribution
x = np.arange(0, n + 1)
probabilities = binom.pmf(x, n, p)
print(probabilities)
# Plot the probabilities
plt.bar(x, probabilities, color='green', alpha=0.7)
plt.title("Binomial Distribution (Defective Products)")
plt.xlabel("Number of Defective Products")
plt.ylabel("Probability")
plt.show()
```

```
[1.07374182e-01 2.68435456e-01 3.01989888e-01 2.01326592e-01
 8.80803840e-02 2.64241152e-02 5.50502400e-03 7.86432000e-04
 7.37280000e-05 4.09600000e-06 1.02400000e-07]
```

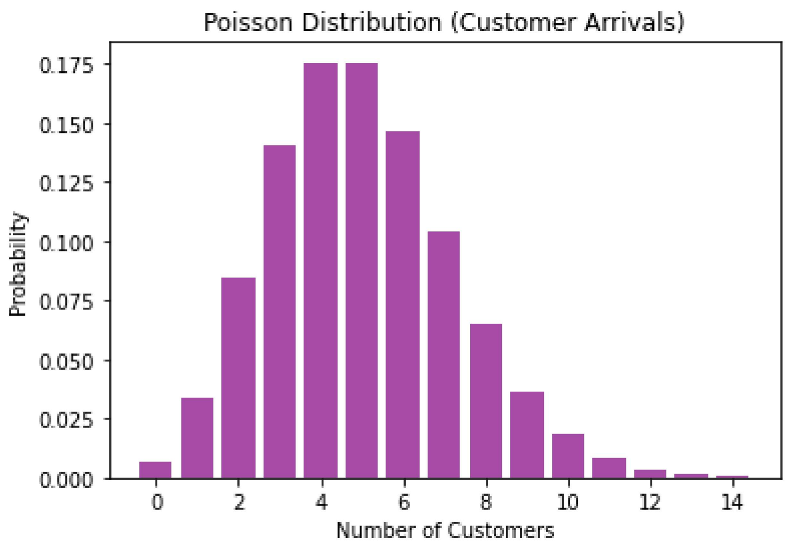


2. Poisson Distribution

Use Case: Used to model the number of events occurring within a fixed time, such as customer arrivals or call-center requests. Example: Modeling the number of customer arrivals at a coffee shop in an hour.

Reference:[https://www.statology.org/poisson-distribution/ (https://www.statology.org/poisson-distribution/)](https://www.statology.org/poisson-distribution/)

In [30]:
```
from scipy.stats import poisson

# Parameters
lambda_val = 5  # average arrivals per hour

# Generate Poisson distribution
x = np.arange(0,15)
probabilities = poisson.pmf(x, lambda_val)

# Plot the distribution
plt.bar(x, probabilities, color='purple',alpha=0.7)
plt.title("Poisson Distribution (Customer Arrivals)")
plt.xlabel("Number of Customers")
plt.ylabel("Probability")
plt.show()
```



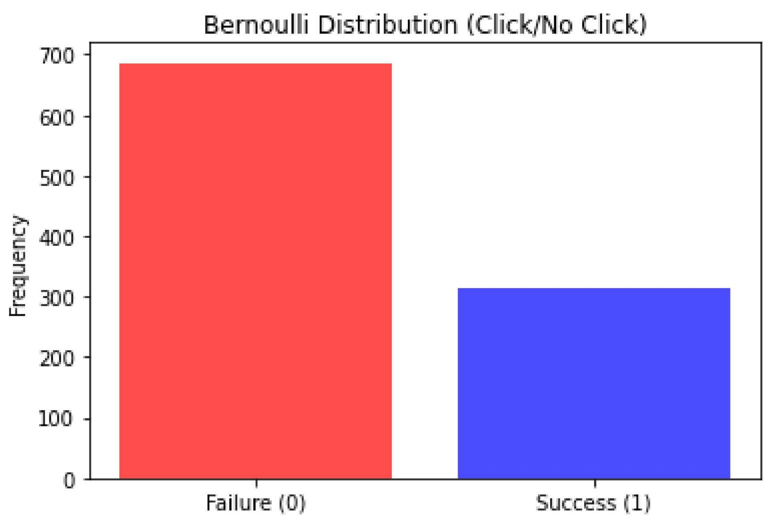
3.Bernoulli Distribution

Use Case: The Bernoulli distribution models a single experiment (trial) that has exactly two possible outcomes: success (1) or failure (0). It's commonly used in scenarios where we are dealing with binary outcomes, such as:

Example: Imagine you're running an A/B test to determine whether users click on a new feature button. The outcome for each user is binary: clicked (1) or not clicked (0).

In [5]:
```
import numpy as np
import matplotlib.pyplot as plt

# Probability of success (e.g., probability a user clicks on the button)
p = 0.3

# Generate 1000 Bernoulli trials
bernoulli_trials = np.random.binomial(n=1, p=p, size=1000)

# Calculate success and failure counts
success_count = sum(bernoulli_trials)
failure_count = len(bernoulli_trials) - success_count

print(f"Successes (Clicks): {success_count}")
print(f"Failures (No Clicks): {failure_count}")

# Plot Bernoulli Distribution
plt.bar([0, 1], [failure_count, success_count], color=['red', 'blue'], alpha
plt.xticks([0, 1], ['Failure (0)', 'Success (1)'])
plt.title("Bernoulli Distribution (Click/No Click)")
plt.ylabel("Frequency")
plt.show()
```

```
Successes (Clicks): 314
Failures (No Clicks): 686
```

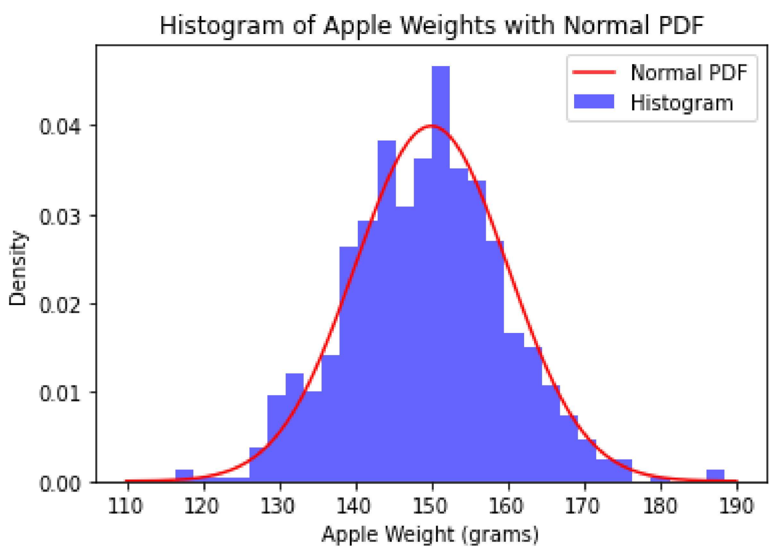


2. Continuous Probability Distributions

A continuous probability distribution applies to a continuous random variable, which can take infinite values within a range (e.g., heights, weights, temperature).

In [24]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Generate normal distribution
mean = 150  # Average weight in grams
std_dev = 10  # Standard deviation in grams
samples = np.random.normal(mean, std_dev, 1000)  # Generate 1000 samples

# Plot histogram
plt.hist(samples, bins=30, density=True, alpha=0.6, color='b', label="Histog

# Overlay Normal PDF
x = np.linspace(mean - 4*std_dev, mean + 4*std_dev, 1000)  # Generate x valu
y = norm.pdf(x, mean, std_dev)  # Compute PDF values

plt.plot(x, y, 'r-', label="Normal PDF")  # Plot PDF
plt.xlabel("Apple Weight (grams)")
plt.ylabel("Density")
plt.title("Histogram of Apple Weights with Normal PDF")
plt.legend()
plt.show()


```



2. Exponential Distribution

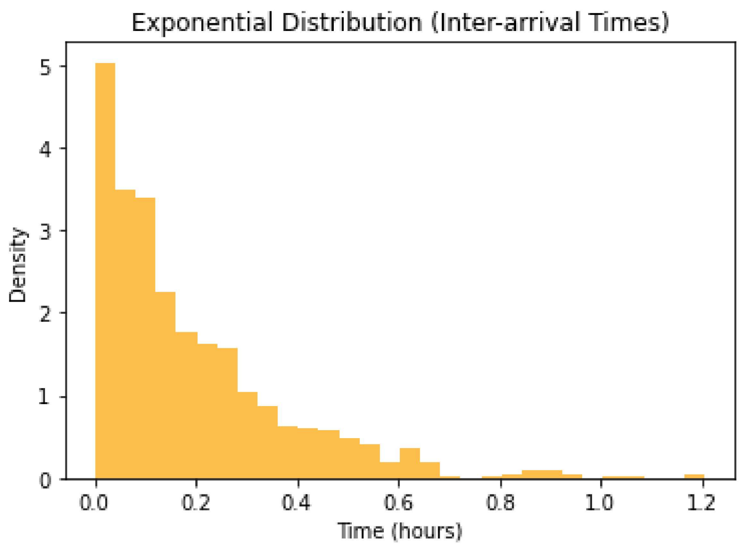
Use Case: Used to model the time between events in a Poisson process, such as wait times in queues or failure rates of machines.

Example: Predicting the time until the next customer arrives at a service counter.

Reference:[https://www.statology.org/exponential-distribution/ (https://www.statology.org/exponential-distribution/)](https://www.statology.org/exponential-distribution/)

In [7]:

```
# Generate exponential distribution
scale = 1 / 5  # mean time between arrivals (e.g., 5 customers per hour)
arrival_times = np.random.exponential(scale, 1000)

# Plot the distribution
plt.hist(arrival_times, bins=30, density=True, alpha=0.7, color='orange')
plt.title("Exponential Distribution (Inter-arrival Times)")
plt.xlabel("Time (hours)")
plt.ylabel("Density")
plt.show()
```



In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```