Central Limit Theorem

Bernoulli Distribution

The Bernoulli distribution is one of the simplest probability distributions. It describes a situation where there are only two possible outcomes, often labeled as 1 (success) and 0 (failure). Think of it like flipping a coin:

Example:

If you flip a coin, there are two outcomes: heads (success) or tails (failure).

The probability of success (getting heads) is p, and the probability of failure (getting tails) is 1−p1.

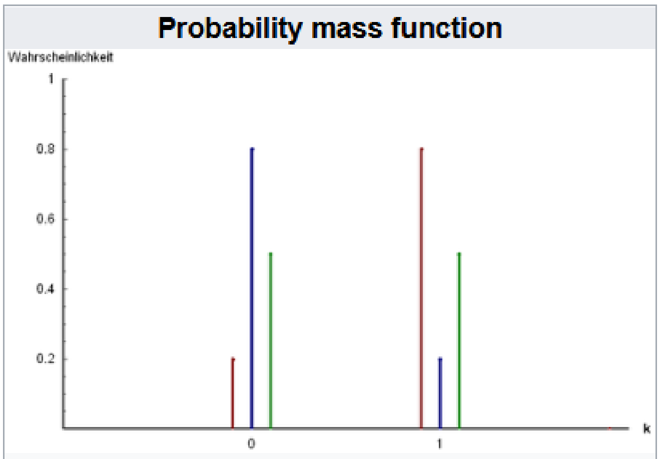
A Bernoulli distribution is defined by one parameter, ppp, which is the probability of success. The formula for its probability mass function (PMF) is:

$$P(X = x) = p^x(1 - p)^{1-x} \quad \text{where } x \in \{0, 1\}$$

Key Points:

- **It's used when there is a single trial or experiment.**
- **The outcome is binary (yes/no, success/failure).**
- **Example in real life: Did a light bulb turn on (yes or no)?**

Bernoulli distribution



Three examples of Bernoulli distribution:

- $P(x = 0) = 0.2$ and $P(x = 1) = 0.8$
- $P(x = 0) = 0.8$ and $P(x = 1) = 0.2$
- $P(x = 0) = 0.5$ and $P(x = 1) = 0.5$

Central Limit Theorem

The Bernoulli distribution is commonly used in machine learning for modelling binary outcomes, such as whether a customer will make a purchase or not, whether an email is spam or not, or whether a patient will have a certain disease or not.

Binomial Distribution

The Binomial distribution is like an extension of the Bernoulli distribution. It deals with multiple independent trials of a Bernoulli process. Instead of asking about the outcome of a single trial, we ask about the number of successes in a fixed number of trials.

- **Example:**
  - **Imagine flipping a coin 10 times. You want to know the probability of getting exactly 7 heads.**
  - **Here, each coin flip is a Bernoulli trial, and the total number of heads (successes) follows a Binomial distribution.**

A Binomial distribution is defined by two parameters:

- **n: The total number of trials.**
- **p: The probability of success in a single trial.**

The formula for its PMF is:

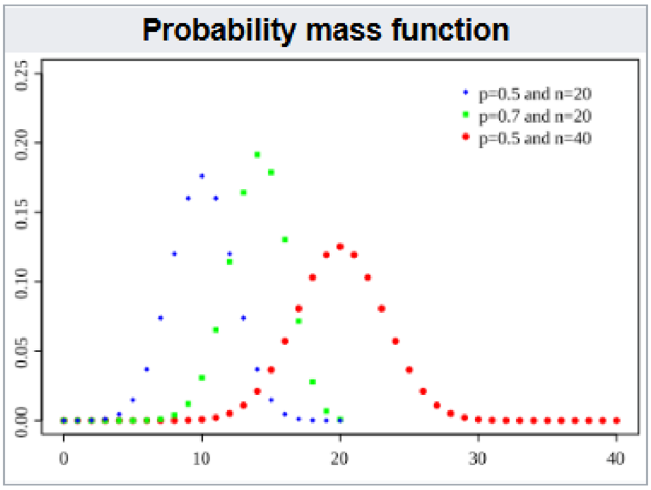
$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Key Points:

- **It's used for multiple independent trials of a Bernoulli process.**
- **Each trial must have the same probability of success, ppp.**
- **Example in real life: How many customers out of 20 will buy a product if the probability of buying is 0.3?**

Central Limit Theorem

Binomial distribution



Uses of Binomial Distribution in Machine Learning :

1. **Modeling Binary Outcomes: Used in logistic regression and other tasks with success/failure outcomes (e.g., spam detection).**
2. **Performance Metrics: Helps evaluate classification metrics like accuracy and precision.**
3. **Probabilistic Models: Forms the basis of models like Naive Bayes and logistic regression.**
4. **Hypothesis Testing: Used in A/B testing to compare success rates of different strategies.**
5. **Anomaly Detection: Identifies unusual patterns in binary event data (e.g., fraud detection).**
6. **NLP: Models word occurrences in text analysis (e.g., Bag-of-Words).**
7. **Ensemble Learning: Analyzes behavior of models trained on random subsets of data.**
8. **Confidence Intervals: Calculates ranges for success probabilities in predictions.**
9. **Reinforcement Learning: Models binary rewards (success/failure) in actions.**
10. **Healthcare Applications: Predicts binary outcomes like disease presence/absence.**

Central Limit Theorem

Sampling Distribution

A sampling distribution is the distribution of a specific statistic (like mean, median, proportion) calculated from multiple samples of the same size taken from a population.

Imagine This:

- **You have a big jar of candies with different weights (this is your population).**
- **You randomly take out small groups (samples) of candies and calculate the average weight for each group.**
- **If you keep doing this multiple times and plot all the averages, the resulting distribution is called the sampling distribution of the mean.**

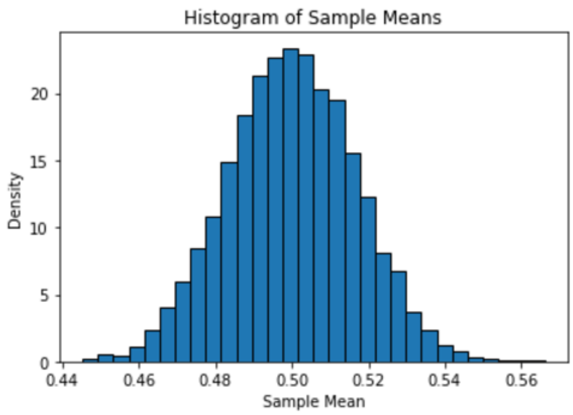
Code Example:

```
# Set the parameters
num_samples = 10000
sample_size = 300
distribution_range = (0, 1)

# Generate samples from a uniform distribution
samples = np.random.uniform(distribution_range[0], distribution_range[1], (num_samples, sample_size))

# Calculate the sample means
sample_means = np.mean(samples, axis=1)

# Plot the histogram of the sample means
plt.hist(sample_means, bins=30, density=True, edgecolor='black')
plt.title('Histogram of Sample Means')
plt.xlabel('Sample Mean')
plt.ylabel('Density')
plt.show()
```



Key Points:

1. **Based on Samples: It shows how a statistic (e.g., mean or proportion) varies across different samples from the same population.**

Central Limit Theorem

2. **Center Around True Value: The center (mean) of the sampling distribution is often close to the true population statistic.**
3. **Depends on Sample Size:**
    - **Larger samples make the sampling distribution narrower (less spread).**
    - **Smaller samples make it wider.**

Why is it Important?

1. **Estimate Population Parameters: Helps infer the true mean, proportion, etc., of the whole population.**
2. **Central Limit Theorem (CLT): If sample size is large enough, the sampling distribution of the mean becomes approximately normal, even if the population is not.**
3. **Hypothesis Testing: Used to calculate p-values and confidence intervals.**

Central Limit Theorem :

The **Central Limit Theorem (CLT)** is a fundamental idea in statistics. It says:

When you take many random samples of the same size from any population and calculate the sample mean, the distribution of those means will form a bell-shaped curve (normal distribution) as the sample size gets large, no matter the shape of the original population.

Imagine This:

- **You have a population that could have any shape (e.g., skewed, uniform, or even weird shapes).**
- **You randomly pick small groups (samples) from the population and calculate their average (mean).**
- **If you repeat this process many times and plot the averages, you'll see a normal curve!**

The conditions required for the CLT to hold are:

1. The sample size is large enough, typically greater than or equal to 30.

Central Limit Theorem

2.The sample is drawn from a finite population or an infinite population with a finite variance.

3. The random variables in the sample are independent and identically distributed.

Key Points:

1. **Works for Any Population Shape: The population could be normal, skewed, or irregular, and CLT still applies.**
2. **Larger Samples = Better Normality:**
   - **For small sample sizes, the sampling distribution might not be normal.**
   - **For larger sample sizes (e.g., n≥30), the sampling distribution will look more like a normal bell curve.**
3. **Why It's Useful:**
   - **Makes it easier to use statistical methods, as many rely on normality.**
   - **Helps in hypothesis testing and confidence intervals.**

Real-Life Example:

Imagine the weights of apples in a store:

- **The apple weights might not be evenly distributed (some are small, some big).**
- **If you take random samples of 30 apples and calculate their average weight, the distribution of those averages will look like a bell curve, even if the individual weights don't.**

Why CLT Is important?

The CLT is important in statistics and machine learning because it allows us to make probabilistic inferences about a population based on a sample of data. For example, we can use the CLT to construct confidence intervals, perform hypothesis tests, and make predictions about the population mean based on the sample data. The CLT also provides a theoretical justification for many commonly used statistical techniques, such as t-tests, ANOVA, and linear regression.

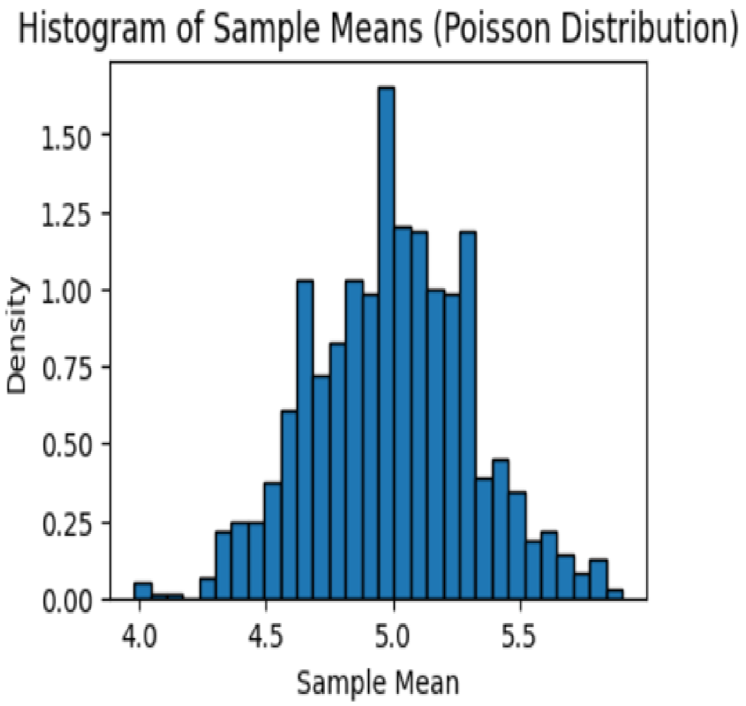
Central Limit Theorem

Code Examples:

1 CTL on Poisson Distribution :

```
# Set the parameters
num_samples = 1000
sample_size = 50
# Poisson distribution parameters
poisson_lambda = 5
# Generate samples from the distributions
poisson_samples = np.random.poisson(lam=poisson_lambda, size=(num_samples, sample_size))
# Calculate the sample means
poisson_means = np.mean(poisson_samples, axis=1)
# Poisson distribution
plt.figure(figsize=(4,3))
plt.hist(poisson_means, bins=30, density=True, edgecolor='black')
plt.title('Histogram of Sample Means (Poisson Distribution)')
plt.xlabel('Sample Mean')
plt.ylabel('Density')
# You can see its looks like well shape curve(Normal distribution)
```

```
Text(0, 0.5, 'Density')
```

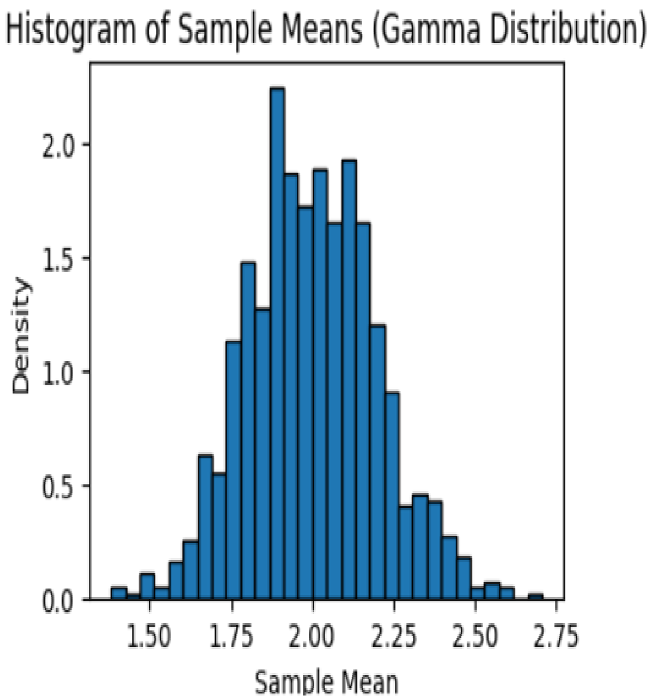


Central Limit Theorem

2.CTL on Gamma Distribution:

```
# Set the parameters
num_samples = 1000
sample_size = 50
# gamma distribution parameters
gamma_shape = 2
gamma_scale = 1
# Generate samples from the distributions
gamma_samples = np.random.gamma(shape=gamma_shape, scale=gamma_scale, size=(num_samples, sample_size))
# Calculate the sample means
gamma_means = np.mean(gamma_samples, axis=1)
# Gamma distribution
plt.figure(figsize=(4,3))
plt.hist(gamma_means, bins=30, density=True, edgecolor='black')
plt.title('Histogram of Sample Means (Gamma Distribution)')
plt.xlabel('Sample Mean')
plt.ylabel('Density')
# You can see its looks like well shape curve(Normal distribution)
```

```
Text(0, 0.5, 'Density')
```



Central Limit Theorem

3.Case Study : On Average Salary of Population,

```
# CLT Case Study
# Set the parameters
population_size = 100000
sample_size = 50
ndarray: population_salaries
ndarray with shape (100000,)
ve sample of salaries (in thousands)
seed for reproducibility
population_salaries = np.random.lognormal(mean=4.5, sigma=0.8, size=population_size)

# Generate multiple samples and calculate the sample means and standard deviations
sample_means = []
sample_std_devs = []

for _ in range(num_samples):
  sample_salaries = np.random.choice(population_salaries, size=sample_size)
  sample_means.append(np.mean(sample_salaries))
  sample_std_devs.append(np.std(sample_salaries))

# Calculate the average of the sample means and the standard error
average_sample_means = np.mean(sample_means)
standard_error = np.std(sample_means) / np.sqrt(num_samples)

# Calculate the 95% confidence interval
margin_of_error = 1.96 * standard_error
lower_limit = average_sample_means - margin_of_error
upper_limit = average_sample_means + margin_of_error

# Report the results
print(f"Estimated average salary (in thousands): {average_sample_means:.2f}")
print(f"95% confidence interval (in thousands): ({lower_limit:.2f}, {upper_limit:.2f})")
print('Population Average Salary (in thousands) :',population_salaries.mean())
```

```
Estimated average salary (in thousands): 124.74
95% confidence interval (in thousands): (121.23, 128.26)
Population Average Salary (in thousands) : 124.08776901808871
```

Central Limit Theorem

4 Case Study on TiTanic Fare :

```
# Case Study On TiTanic Fare Column
train = pd.read_csv('train (7).csv')
test = pd.read_csv('test (4).csv')
# Concat Both Dataset
data = pd.concat([train.drop(columns='Survived'),test]).sample(1309) # Suffle da

# samples
sample = []
for i in range(20):
  sample.append(data['Fare'].dropna().sample(30).values.tolist())

# covert sample list into array
samples = np.array(sample)
print('Samples Shape :',samples.shape)

# calculate Sample Means of every sample
sample_means = samples.mean(axis=1)
print('Sample Means :',sample_means)

# Calculate mean of sample means
mean_of_sample_means = sample_means.mean()
print('Mean of Sample means :',mean_of_sample_means)

# Calculate sampling standard deviation
std_of_sample_means = sample_means.std()
print('Standard deviation of sample means :',std_of_sample_means)
```

```
# Since normal distribution cover 95% area in 2 std .
# so we calculate upper limit and lower limit with 95% confidence on 2 std.
lower_limit = mean_of_sample_means - (2 * std_of_sample_means)
upper_limit = mean_of_sample_means + (2* std_of_sample_means)

# The Range of population fare
print("The Range Of Population average Fare  :",lower_limit ,' , ', upper_limit)
print('Actual Population Average Fare : ',data['Fare'].mean())
```

```
Samples Shape : (20, 30)
Sample Means : [31.98569333 30.94972667 40.86097333 44.82361333 40.62027    46.63458667
 33.73319    19.21152333 36.14542    49.21305333 29.28888    24.15875667
 25.62222    28.39625333 26.14555333 26.85847333 32.22166667 44.78444333
 41.33500333 32.28347333]
Mean of Sample means : 34.263638666666665
Standard deviation of sample means : 8.197587795063553
The Range Of Population average Fare  : 17.86846307653956  ,  50.658814256793775
Actual Population Average Fare :  33.29547928134557
```