Probability 1

Sure, let's discuss the formulas and explanations for the Normal Distribution, Standard Normal Distribution (Z Distribution), F-Distribution, and Chi-Square Distribution.

### 1. \*\*Normal Distribution:\*\*

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

\[ f(x | \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2} \]

Where:

- \(x\) is a random variable.

- \(\mu\) is the mean (average) of the distribution.

- \(\sigma\) is the standard deviation, representing the spread of the distribution.

The normal distribution is symmetric and bell-shaped, and about 68% of the data falls within one standard deviation of the mean, 95% within two standard deviations, and 99.7% within three standard deviations.

### 2. \*\*Standard Normal Distribution (Z Distribution):\*\*

The standard normal distribution is a special case of the normal distribution where the mean (\(\mu\)) is 0 and the standard deviation (\(\sigma\)) is 1. The probability density function for the standard normal distribution is denoted as \(Z\) and is often used in statistical tables:

\[ f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \]

### 3. \*\*F-Distribution:\*\*

The probability density function (PDF) of an F-distribution with \(d\_1\) and \(d\_2\) degrees of freedom is given by:

\[ f(x; d\_1, d\_2) = \frac{\Gamma\left(\frac{d\_1 + d\_2}{2}\right)}{\Gamma\left(\frac{d\_1}{2}\right) \Gamma\left(\frac{d\_2}{2}\right)} \left(\frac{d\_1}{d\_2}\right)^{\frac{d\_1}{2}} \frac{x^{\frac{d\_1}{2} - 1}}{\left(1 + \frac{d\_1 x}{d\_2}\right)^{\frac{d\_1 + d\_2}{2}}} \]

Where:

- \(x\) is a random variable.

- \(\Gamma\) is the gamma function.

- \(d\_1\) and \(d\_2\) are the degrees of freedom.

The F-distribution is commonly used in analysis of variance (ANOVA) and regression analysis.

### 4. \*\*Chi-Square Distribution:\*\*

The probability density function (PDF) of a chi-square distribution with \(k\) degrees of freedom is given by:

\[ f(x; k) = \frac{x^{\frac{k}{2} - 1} e^{-\frac{x}{2}}}{2^{\frac{k}{2}} \Gamma\left(\frac{k}{2}\right)} \]

Where:

- \(x\) is a random variable.

- \(\Gamma\) is the gamma function.

- \(k\) is the degrees of freedom.

The chi-square distribution is commonly used in hypothesis testing and is particularly associated with the chi-square test for independence and the chi-square goodness-of-fit test.

These distributions are fundamental in statistical theory and have widespread applications in various fields, including hypothesis testing, confidence intervals, and modeling real-world phenomena.

**1. Normal Distribution with NumPy and Pandas:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

# Generate random data from a normal distribution

mean = 0

std\_dev = 1

num\_samples = 1000

normal\_data = np.random.normal(mean, std\_dev, num\_samples)

# Create a DataFrame

df\_normal = pd.DataFrame({'Value': normal\_data})

# Plot the histogram

plt.hist(df\_normal['Value'], bins=30, density=True, alpha=0.5, color='b')

# Plot the normal distribution curve for comparison

xmin, xmax = plt.xlim()

x = np.linspace(xmin, xmax, 100)

p = np.exp(-0.5 \* ((x - mean) / std\_dev) \*\* 2) / (std\_dev \* np.sqrt(2 \* np.pi))

plt.plot(x, p, 'k', linewidth=2)

plt.title('Normal Distribution')

plt.show()

### 2. 2. ****Standard Normal Distribution (Z Distribution) with NumPy****

# Generate random data from a standard normal distribution

num\_samples = 1000

standard\_normal\_data = np.random.randn(num\_samples)

# Create a DataFrame

df\_standard\_normal = pd.DataFrame({'Value': standard\_normal\_data})

# Plot the histogram

plt.hist(df\_standard\_normal['Value'], bins=30, density=True, alpha=0.5, color='b')

# Plot the standard normal distribution curve for comparison

xmin, xmax = plt.xlim()

x = np.linspace(xmin, xmax, 100)

p = np.exp(-0.5 \* x\*\*2) / np.sqrt(2 \* np.pi)

plt.plot(x, p, 'k', linewidth=2)

plt.title('Standard Normal Distribution')

plt.show()

3. **F-Distribution and Chi-Square Distribution with NumPy:**

**from scipy.stats import f, chi2**

**# F-distribution**

**d1, d2 = 5, 10**

**f\_data = f.rvs(d1, d2, size=num\_samples)**

**# Chi-square distribution**

**degrees\_of\_freedom = 8**

**chi2\_data = chi2.rvs(degrees\_of\_freedom, size=num\_samples)**

**# Create DataFrames**

**df\_f = pd.DataFrame({'Value': f\_data})**

**df\_chi2 = pd.DataFrame({'Value': chi2\_data})**

**# Plot histograms**

**plt.subplot(1, 2, 1)**

**plt.hist(df\_f['Value'], bins=30, density=True, alpha=0.5, color='b')**

**plt.title('F-Distribution')**

**plt.subplot(1, 2, 2)**

**plt.hist(df\_chi2['Value'], bins=30, density=True, alpha=0.5, color='g')**

**plt.title('Chi-Square Distribution')**

**plt.tight\_layout()**

**plt.show()**

4. bars and charts

Certainly! Let's go through examples of creating basic graph representations, including Bar Chart, Histogram, Box Plot, and Scatterplot, along with examples of Probability Distributions.

**1. Bar Chart with NumPy and Matplotlib:**

pythonCopy code

import numpy as np import matplotlib.pyplot as plt # Sample data categories = ['Category A', 'Category B', 'Category C'] values = [20, 35, 15] # Create a bar chart plt.bar(categories, values, color='blue') plt.title('Bar Chart') plt.xlabel('Categories') plt.ylabel('Values') plt.show()

**2. Histogram with NumPy and Matplotlib:**

pythonCopy code

# Generate random data for a histogram data = np.random.randn(1000) # Create a histogram plt.hist(data, bins=30, color='green', alpha=0.7) plt.title('HistogramCertainly! Let's go through examples of creating basic graph representations, including Bar Chart, Histogram, Box Plot, and Scatterplot, along with examples of Probability Distributions.

1. Bar Chart with NumPy and Matplotlib:

python

Copy code

import numpy as np

import matplotlib.pyplot as plt

# Sample data

categories = ['Category A', 'Category B', 'Category C']

values = [20, 35, 15]

# Create a bar chart

plt.bar(categories, values, color='blue')

plt.title('Bar Chart')

plt.xlabel('Categories')

plt.ylabel('Values')

plt.show()

2. Histogram with NumPy and Matplotlib:

python

Copy code

# Generate random data for a histogram

data = np.random.randn(1000)

# Create a histogram

plt.hist(data, bins=30, color='green', alpha=0.7)

plt.title('Histogram')

plt.xlabel('Values')

plt.ylabel('Frequency')

plt.show()

3. Box Plot with Pandas and Matplotlib:

python

Copy code

import pandas as pd

# Generate random data for a box plot

data\_box = np.random.randn(100, 3)

# Create a DataFrame

df\_box = pd.DataFrame(data\_box, columns=['A', 'B', 'C'])

# Create a box plot

df\_box.boxplot()

plt.title('Box Plot')

plt.ylabel('Values')

plt.show()

4. Scatterplot with Pandas and Matplotlib:

python

Copy code

# Generate random data for a scatterplot

x\_data = np.random.rand(50)

y\_data = 2 \* x\_data + 1 + 0.1 \* np.random.randn(50)

# Create a DataFrame

df\_scatter = pd.DataFrame({'X': x\_data, 'Y': y\_data})

# Create a scatterplot

plt.scatter(df\_scatter['X'], df\_scatter['Y'], color='red')

plt.title('Scatterplot')

plt.xlabel('X')

plt.ylabel('Y')

plt.show()

5. Probability Distributions with NumPy and Matplotlib:

python

Copy code

from scipy.stats import norm, uniform

# Generate data for probability distributions

data\_normal = norm.rvs(size=1000)

data\_uniform = uniform.rvs(size=1000)

# Create histograms for normal and uniform distributions

plt.subplot(1, 2, 1)

plt.hist(data\_normal, bins=30, color='blue', alpha=0.7)

plt.title('Normal Distribution')

plt.subplot(1, 2, 2)

plt.hist(data\_uniform, bins=30, color='orange', alpha=0.7)

plt.title('Uniform Distribution')

plt.tight\_layout()

plt.show()') plt.xlabel('Values') plt.ylabel('Frequency') plt.show()

**3. Box Plot with Pandas and Matplotlib:**

pythonCopy code

import pandas as pd # Generate random data for a box plot data\_box = np.random.randn(100, 3) # Create a DataFrame df\_box = pd.DataFrame(data\_box, columns=['A', 'B', 'C']) # Create a box plot df\_box.boxplot() plt.title('Box Plot') plt.ylabel('Values') plt.show()

**4. Scatterplot with Pandas and Matplotlib:**

pythonCopy code

# Generate random data for a scatterplot x\_data = np.random.rand(50) y\_data = 2 \* x\_data + 1 + 0.1 \* np.random.randn(50) # Create a DataFrame df\_scatter = pd.DataFrame({'X': x\_data, 'Y': y\_data}) # Create a scatterplot plt.scatter(df\_scatter['X'], df\_scatter['Y'], color='red') plt.title('Scatterplot') plt.xlabel('X') plt.ylabel('Y') plt.show()

**5. Probability Distributions with NumPy and Matplotlib:**

pythonCopy code

from scipy.stats import norm, uniform # Generate data for probability distributions data\_normal = norm.rvs(size=1000) data\_uniform = uniform.rvs(size=1000) # Create histograms for normal and uniform distributions plt.subplot(1, 2, 1) plt.hist(data\_normal, bins=30, color='blue', alpha=0.7) plt.title('Normal Distribution') plt.subplot(1, 2, 2) plt.hist(data\_uniform, bins=30, color='orange', alpha=0.7) plt.title('Uniform Distribution') plt.tight\_layout() plt.show()