Sampling and Probability distributions

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

# Set seed for reproducibility
np.random.seed(42)

# Number of samples
n_samples = 1000

# Generate samples from different probability distributions
uniform_samples = np.random.uniform(0, 1, n_samples)
normal_samples = np.random.normal(0, 1, n_samples)
exponential_samples = np.random.exponential(1, n_samples)
binomial_samples = np.random.binomial(n=10, p=0.5, size=n_samples)
poisson_samples = np.random.poisson(lam=3, size=n_samples)

# Plot histograms
fig, axes = plt.subplots(3, 2, figsize=(12, 10))

# Uniform Distribution
axes[0, 0].hist(uniform_samples, bins=30, density=True, alpha=0.6, color='b')
axes[0, 0].set_title("Uniform Distribution (0,1)")

# Normal Distribution
axes[0, 1].hist(normal_samples, bins=30, density=True, alpha=0.6, color='g')
x = np.linspace(-4, 4, 100)
```

```
axes[0, 1].plot(x, stats.norm.pdf(x, 0, 1), 'r-', lw=2)  # Theoretical PDF
axes[0, 1].set_title("Normal Distribution (μ=0, σ=1)")

# Exponential Distribution
axes[1, 0].hist(exponential_samples, bins=30, density=True, alpha=0.6, color='c')
x = np.linspace(0, 5, 100)
axes[1, 0].plot(x, stats.expon.pdf(x, scale=1), 'r-', lw=2)  # Theoretical PDF
axes[1, 0].set_title("Exponential Distribution (λ=1)")

# Binomial Distribution
axes[1, 1].hist(binomial_samples, bins=10, density=True, alpha=0.6, color='m',
align='left')
x = np.arange(0, 11)
axes[1, 1].plot(x, stats.binom.pmf(x, n=10, p=0.5), 'ro-', lw=2)  # Theoretical PMF
axes[1, 1].set_title("Binomial Distribution (n=10, p=0.5)")

# Poisson Distribution
axes[2, 0].hist(poisson_samples, bins=10, density=True, alpha=0.6, color='y',
align='left')
x = np.arange(0, 10)
axes[2, 0].plot(x, stats.poisson.pmf(x, mu=3), 'ro-', lw=2)  # Theoretical PMF
axes[2, 0].set_title("Poisson Distribution (λ=3)")

# Hide empty subplot
axes[2, 1].axis('off')

plt.tight_layout()
plt.show()
```

```
# Print basic statistics for sampled data

print("Basic Statistics:")

print(f"Uniform Distribution: Mean={np.mean(uniform_samples):.2f},
Std={np.std(uniform_samples):.2f}")

print(f"Normal Distribution: Mean={np.mean(normal_samples):.2f},
Std={np.std(normal_samples):.2f}")

print(f"Exponential Distribution: Mean={np.mean(exponential_samples):.2f},
Std={np.std(exponential_samples):.2f}")

print(f"Binomial Distribution: Mean={np.mean(binomial_samples):.2f},
Std={np.std(binomial_samples):.2f}")

print(f"Poisson Distribution: Mean={np.mean(poisson_samples):.2f},
Std={np.std(poisson_samples):.2f}")
```

1. **Generates random samples** from:
   - **Uniform Distribution** (0,1)
   - **Normal Distribution** (Mean=0, Std=1)
   - **Exponential Distribution** (Lambda=1)
   - **Binomial Distribution** (n=10, p=0.5)
   - **Poisson Distribution** (Lambda=3)
2. **Plots histograms** of the generated samples.
   - Overlay **theoretical probability density/mass functions** on histograms for validation.
3. **Computes and prints basic statistics** (mean and standard deviation) for each sample.