probability density functions (PDFs)

This task involves generating random variables from specific probability density functions (PDFs) over specified ranges. The goal is to simulate random variables using both **rejection sampling** and **inverse CDF sampling** methods for four different probability densities. We aim to:

1. Print the first 10 values of each simulated variable.
2. Plot histograms for each distribution to verify that the shape of the histogram matches the expected PDF.

$$\rho(x) = x \cdot e^{-x^2}, \quad x \in [1, 3]$$

$$\rho(x) = x \cdot \cos(x^2), \quad x \in \left[0, \frac{\pi}{3}\right]$$

$$\rho(x) = e^x, \quad x \in [0, 2]$$

$$\rho(x) = \frac{x}{1 + x^2}, \quad x \in [0, 4]$$

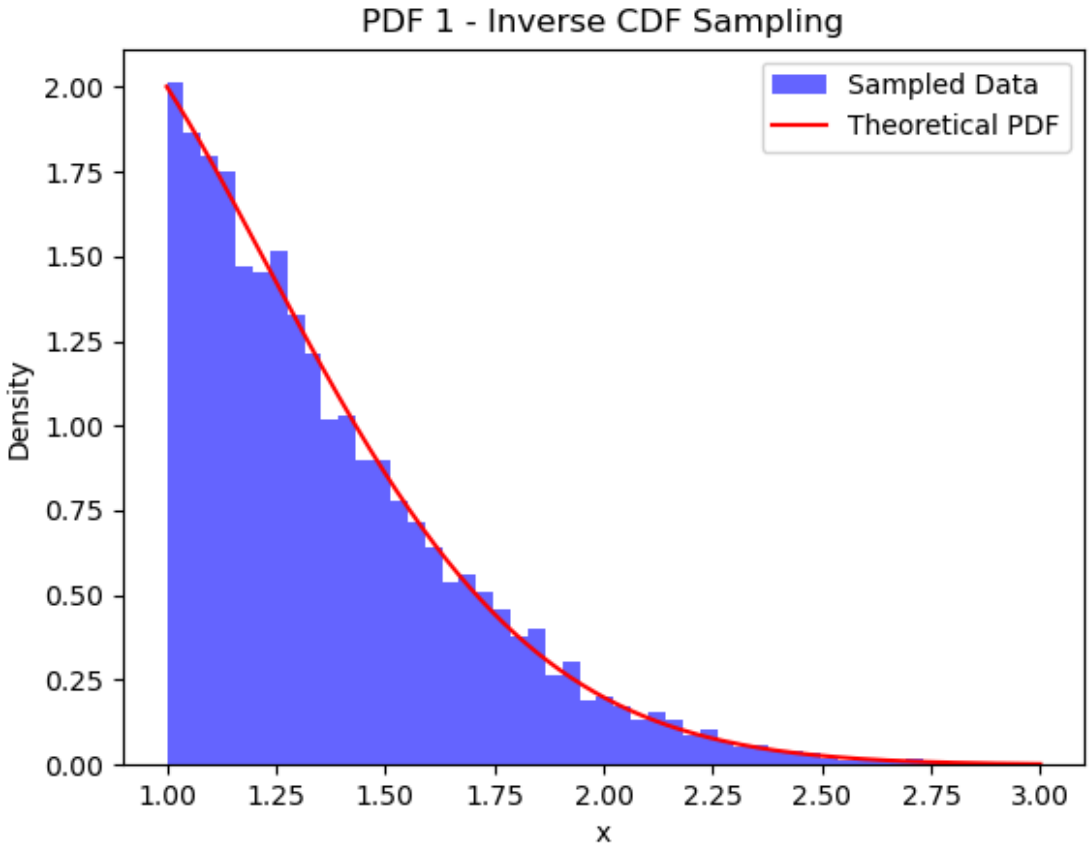
Each PDF has an unknown constant, c, which normalizes the area under the curve to 1. Once c is determined for each PDF, it is possible to generate random samples that follow the specified distributions.

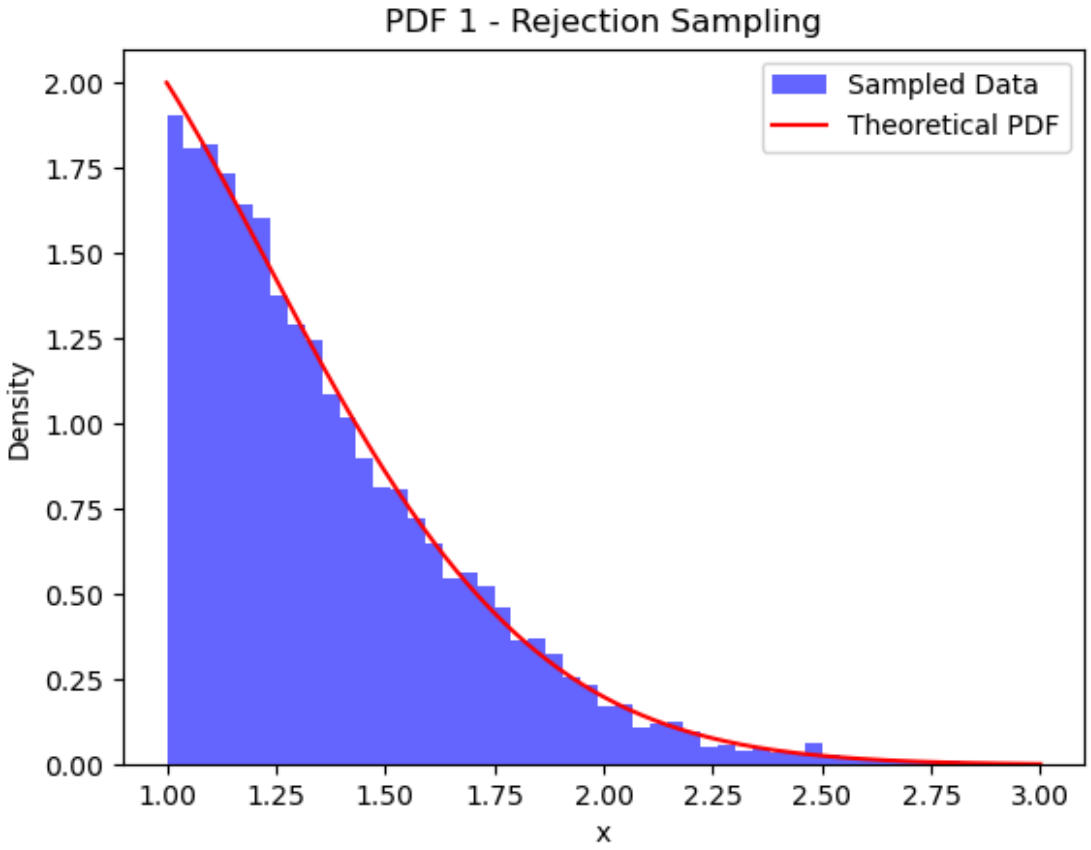
Code

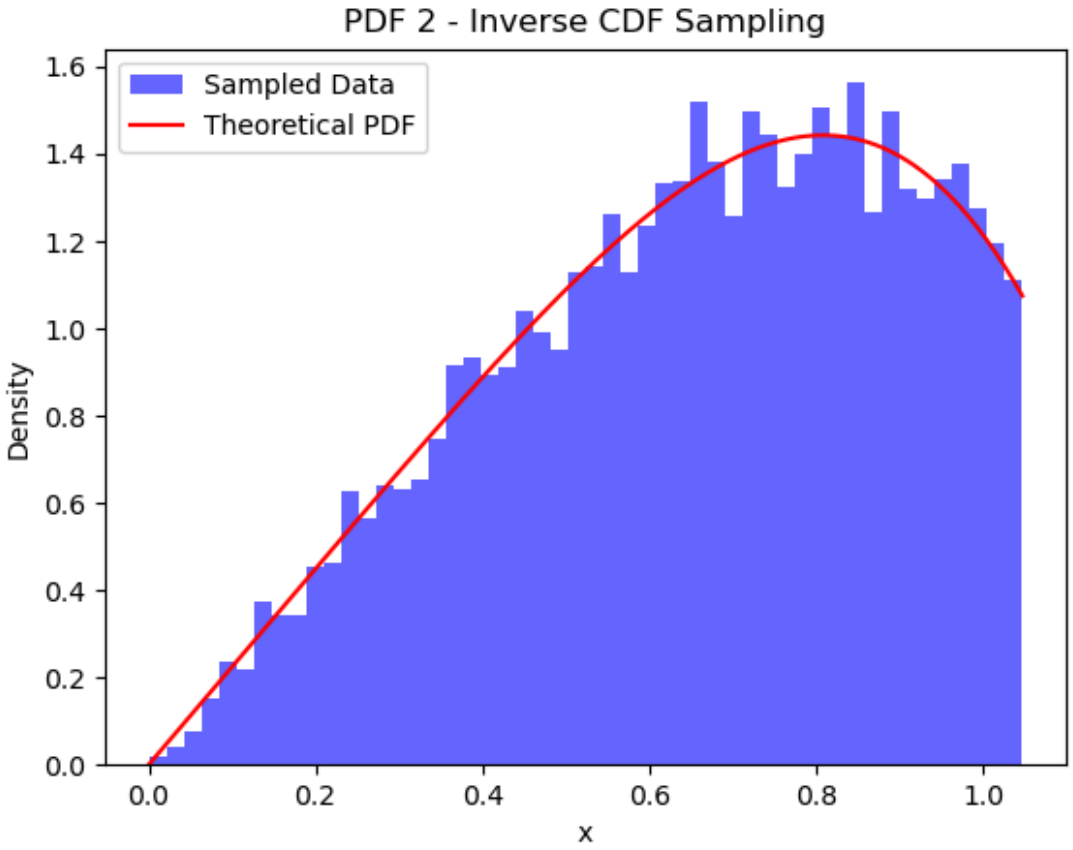
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.optimize import root_scalar
import os


def pdf_1(x, c):
    return c * x * np.exp(-x**2)

def pdf_2(x, c):
    return c * x * np.cos(x**2)

def pdf_3(x, c):
    return c * np.exp(x)

def pdf_4(x, c):
    return c * x / (1 + x**2)

def calculate_c(pdf, a, b):
    integral, _ = quad(lambda x: pdf(x, 1), a, b)
    return 1 / integral


# Rejection Sampling for both PDFs
def rejection_sampling(pdf, c, a, b, max_pdf, n_samples):
    samples = []
    while len(samples) < n_samples:
        x = np.random.uniform(a, b)
        y = np.random.uniform(0, max_pdf)
        if y < pdf(x, c):
            samples.append(x)
    return np.array(samples)


# Inverse CDF Sampling for PDFs
def inverse_cdf_sampling(cdf_inv, n_samples):
    u = np.random.uniform(0, 1, n_samples)  # Generate U(0,1) samples
    return np.array([cdf_inv(ui) for ui in u])


# PDF 1: Inverse CDF using root-finding
def cdf_inv_pdf_1(u, c):
    def equation(x):
        return quad(lambda t: pdf_1(t, c), 1, x)[0] - u
    return root_scalar(equation, bracket=[1, 3]).root


# PDF 3: Analytic Inverse CDF (since it's exponential over [0, 2])
def cdf_inv_pdf_3(u, c):
    x = np.log(u * (np.exp(2) - 1) + 1)
    return min(max(x, 0), 2)  # Ensure x is within [0, 2]


def simulate_pdf_1(n_samples, method='rejection'):
    a, b = 1, 3
    c = calculate_c(pdf_1, a, b)
    max_pdf = pdf_1(1, c)
    if method == 'rejection':
        return rejection_sampling(pdf_1, c, a, b, max_pdf, n_samples)
    else:
        return inverse_cdf_sampling(lambda u: cdf_inv_pdf_1(u, c), n_samples)


def simulate_pdf_2(n_samples, method='rejection'):
    a, b = 0, np.pi / 3
    c = calculate_c(pdf_2, a, b)
    x_vals = np.linspace(a, b, 1000)
    max_pdf = np.max(pdf_2(x_vals, c))  # Find the maximum value of the PDF
    if method == 'rejection':
        return rejection_sampling(pdf_2, c, a, b, max_pdf, n_samples)
    else:
        return rejection_sampling(pdf_2, c, a, b, max_pdf, n_samples)


def simulate_pdf_3(n_samples, method='rejection'):
    a, b = 0, 2
    c = calculate_c(pdf_3, a, b)
    max_pdf = pdf_3(b, c)
    if method == 'rejection':
        return rejection_sampling(pdf_3, c, a, b, max_pdf, n_samples)
    else:
        return inverse_cdf_sampling(lambda u: cdf_inv_pdf_3(u, c), n_samples)

```

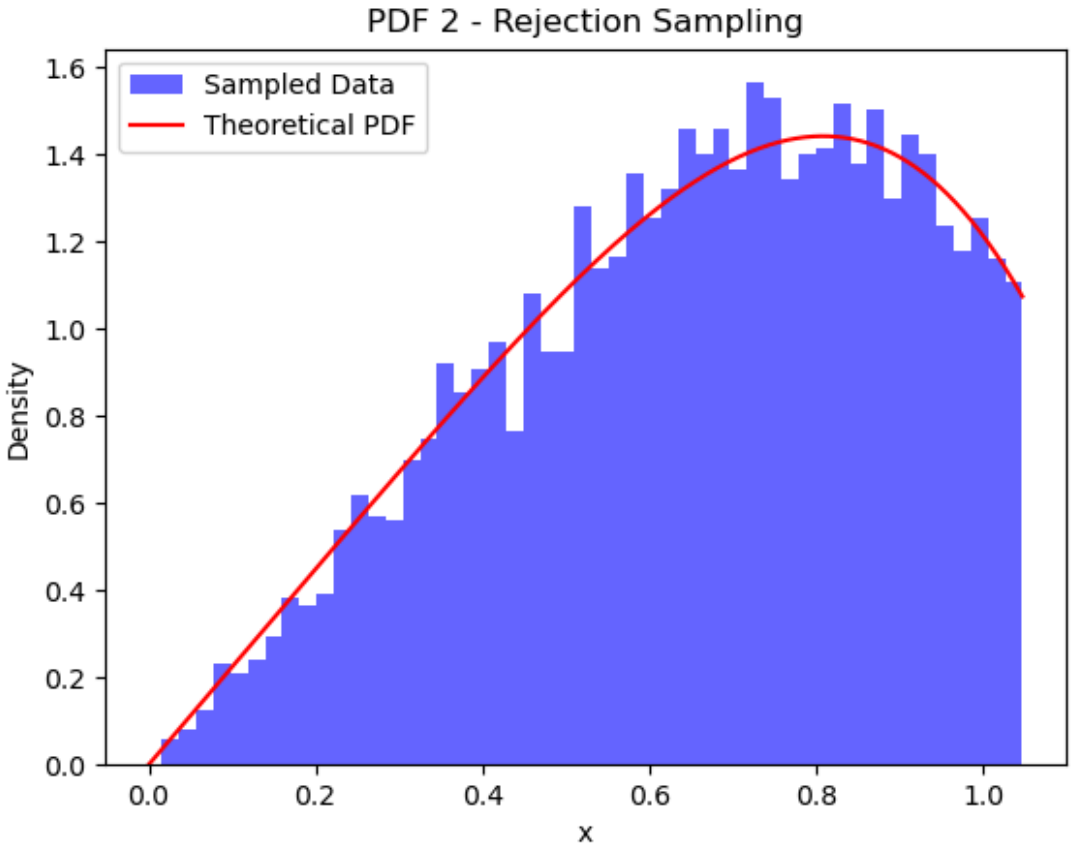
```python
def simulate_pdf_4(n_samples, method='rejection'):
    a, b = 0, 4  # Valid range for this PDF
    c = calculate_c(pdf_4, a, b)
    x_vals = np.linspace(a, b, 1000)
    max_pdf = np.max(pdf_4(x_vals, c))  # Find the maximum value of the PDF
    if method == 'rejection':
        return rejection_sampling(pdf_4, c, a, b, max_pdf, n_samples)
    else:
        return rejection_sampling(pdf_4, c, a, b, max_pdf, n_samples)


def plot_histogram(samples, pdf, c, a, b, title, filename, bins=50):
    plt.figure()
    plt.hist(samples, bins=bins, density=True, alpha=0.6, color='b', label="Sampled Data")
    x_vals = np.linspace(a, b, 1000)
    theoretical_pdf = pdf(x_vals, c)
    plt.plot(x_vals, theoretical_pdf, 'r', label="Theoretical PDF")
    plt.legend()
    plt.title(title)
    plt.xlabel("x")
    plt.ylabel("Density")
    os.makedirs("histograms", exist_ok=True)
    plt.savefig(f"histograms/{filename}.png")
    plt.close()


n_samples = 10000  # Increased number of samples
# PDF 1 - Rejection and Inverse CDF Sampling
samples_pdf_1_rejection = simulate_pdf_1(n_samples, method='rejection')
samples_pdf_1_inverse = simulate_pdf_1(n_samples, method='inverse')

print("First 10 samples (Rejection Sampling) for PDF 1:", samples_pdf_1_rejection[:10])
print("First 10 samples (Inverse CDF Sampling) for PDF 1:", samples_pdf_1_inverse[:10])

plot_histogram(samples_pdf_1_rejection, pdf_1, calculate_c(pdf_1, 1, 3), 1, 3, "PDF 1 - Rejection Sampling", "pdf_1_rejection")
plot_histogram(samples_pdf_1_inverse, pdf_1, calculate_c(pdf_1, 1, 3), 1, 3, "PDF 1 - Inverse CDF Sampling", "pdf_1_inverse")


# PDF 2 - Rejection and Inverse CDF Sampling
samples_pdf_2_rejection = simulate_pdf_2(n_samples, method='rejection')
samples_pdf_2_inverse = simulate_pdf_2(n_samples, method='inverse')

print("First 10 samples (Rejection Sampling) for PDF 2:", samples_pdf_2_rejection[:10])
print("First 10 samples (Inverse CDF Sampling) for PDF 2:", samples_pdf_2_inverse[:10])

plot_histogram(samples_pdf_2_rejection, pdf_2, calculate_c(pdf_2, 0, np.pi / 3), 0, np.pi / 3, "PDF 2 - Rejection Sampling", "pdf_2_rejection")
plot_histogram(samples_pdf_2_inverse, pdf_2, calculate_c(pdf_2, 0, np.pi / 3), 0, np.pi / 3, "PDF 2 - Inverse CDF Sampling", "pdf_2_inverse")


# PDF 3 - Rejection and Inverse CDF Sampling
samples_pdf_3_rejection = simulate_pdf_3(n_samples, method='rejection')
samples_pdf_3_inverse = simulate_pdf_3(n_samples, method='inverse')

print("First 10 samples (Rejection Sampling) for PDF 3:", samples_pdf_3_rejection[:10])
print("First 10 samples (Inverse CDF Sampling) for PDF 3:", samples_pdf_3_inverse[:10])

plot_histogram(samples_pdf_3_rejection, pdf_3, calculate_c(pdf_3, 0, 2), 0, 2, "PDF 3 - Rejection Sampling", "pdf_3_rejection")
plot_histogram(samples_pdf_3_inverse, pdf_3, calculate_c(pdf_3, 0, 2), 0, 2, "PDF 3 - Inverse CDF Sampling", "pdf_3_inverse")


# PDF 4 - Rejection and Inverse CDF Sampling
samples_pdf_4_rejection = simulate_pdf_4(n_samples, method='rejection')
samples_pdf_4_inverse = simulate_pdf_4(n_samples, method='inverse')

print("First 10 samples (Rejection Sampling) for PDF 4:", samples_pdf_4_rejection[:10])
print("First 10 samples (Inverse CDF Sampling) for PDF 4:", samples_pdf_4_inverse[:10])

plot_histogram(samples_pdf_4_rejection, pdf_4, calculate_c(pdf_4, 0, 4), 0, 4, "PDF 4 - Rejection Sampling", "pdf_4_rejection")
plot_histogram(samples_pdf_4_inverse, pdf_4, calculate_c(pdf_4, 0, 4), 0, 4, "PDF 4 - Inverse CDF Sampling", "pdf_4_inverse")

```

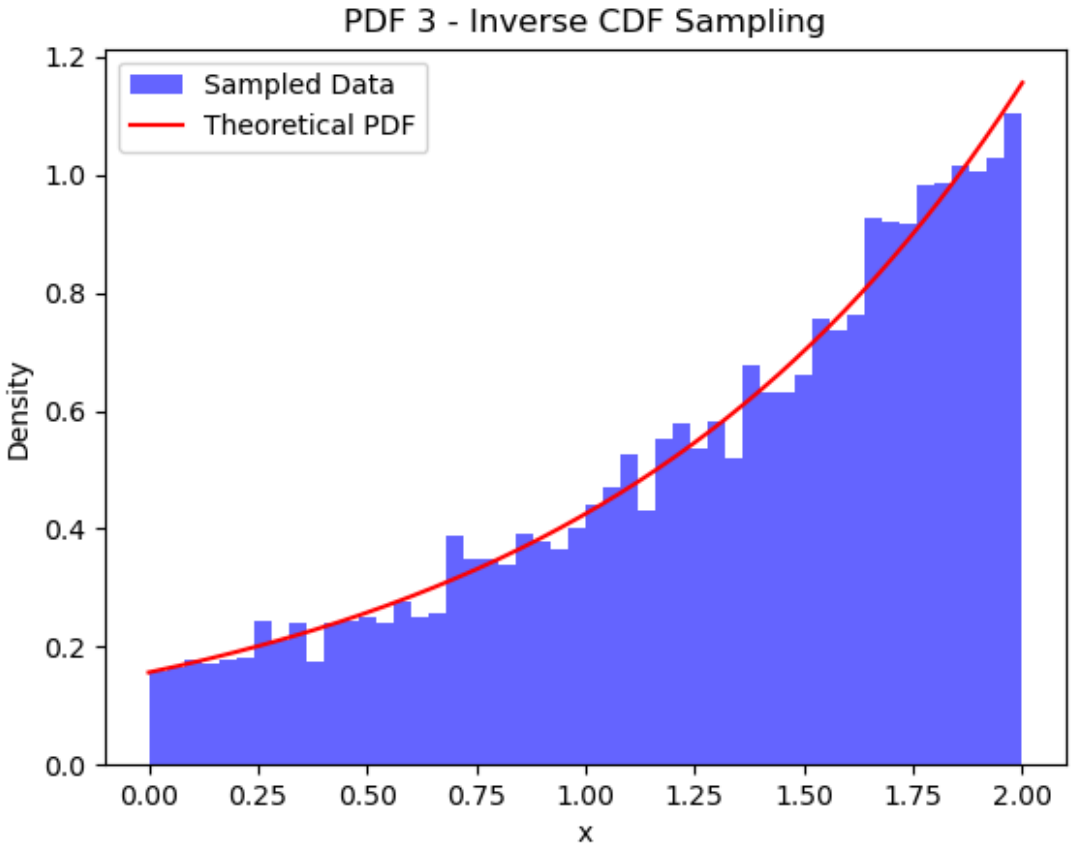
Plots

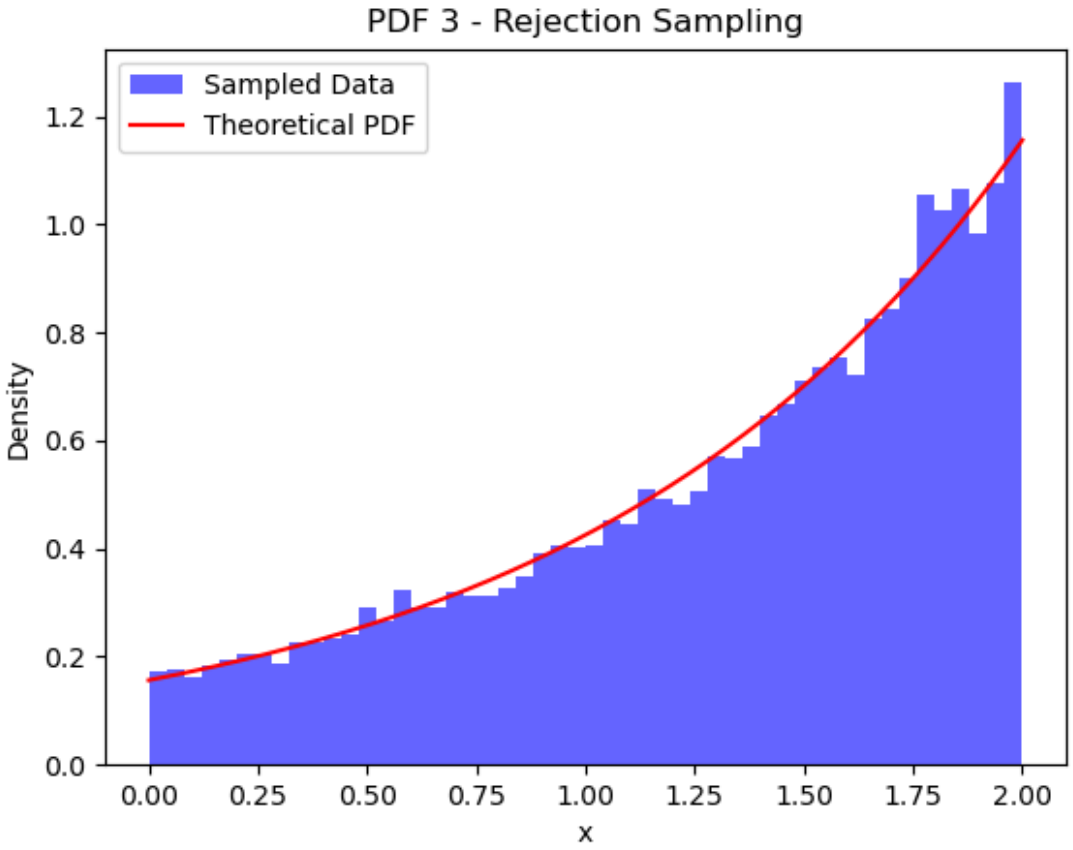

PDF 1 - Inverse CDF Sampling
Sampled Data
Theoretical PDF
Density
x
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00

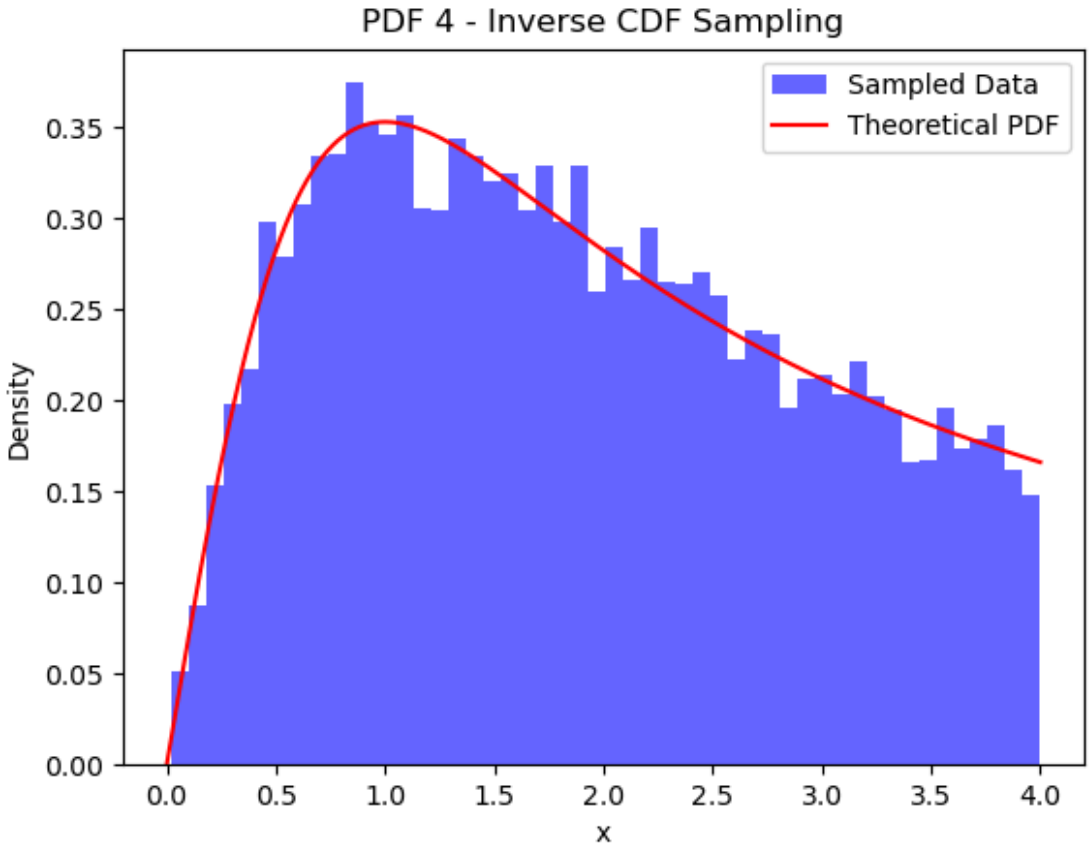

PDF 1 - Rejection Sampling
Sampled Data
Theoretical PDF
Density
x
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00

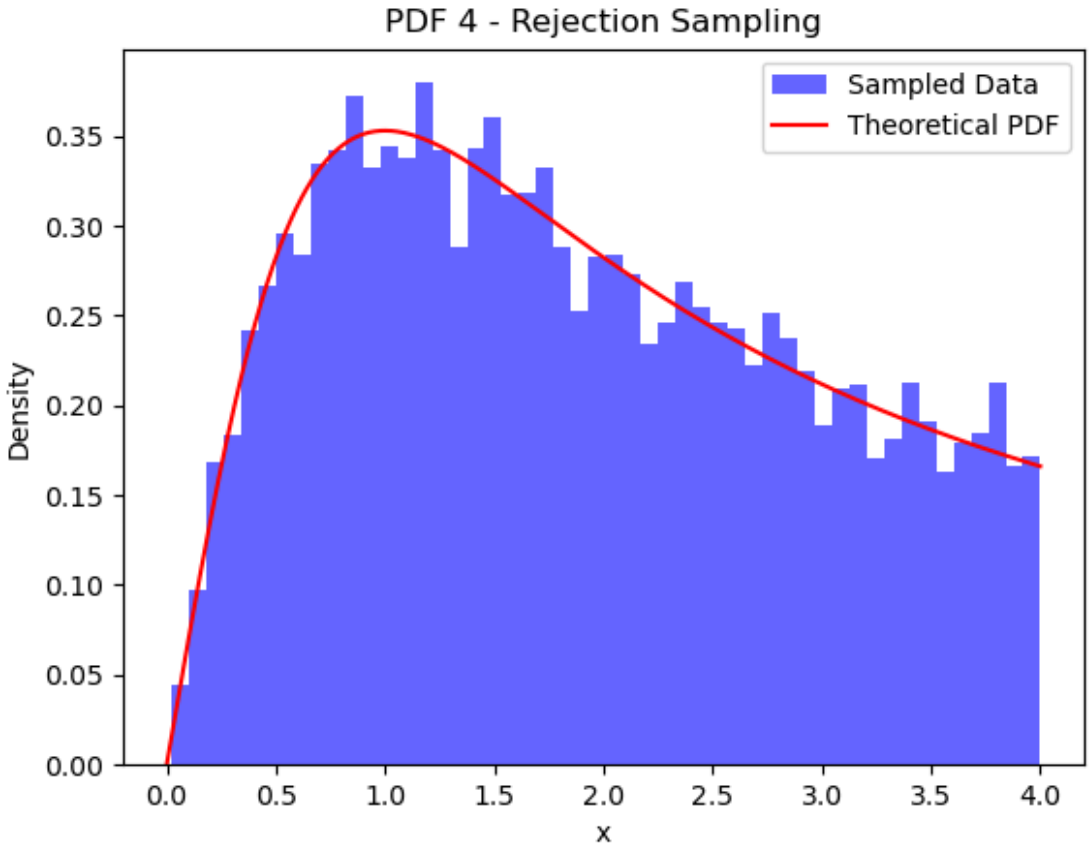

PDF 2 - Inverse CDF Sampling
Sampled Data
Theoretical PDF
Density
x
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6


PDF 2 - Rejection Sampling
Sampled Data
Theoretical PDF
Density
x
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6


PDF 3 - Inverse CDF Sampling
Sampled Data
Theoretical PDF
Density
x
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


PDF 3 - Rejection Sampling
Sampled Data
Theoretical PDF
Density
x
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


PDF 4 - Inverse CDF Sampling
Sampled Data
Theoretical PDF
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x



First 10 samples (Rejection Sampling) for **PDF 1**: [1.21603736 1.16157728 1.51091769 1.24207989 1.05884829 1.15549938 1.38731977 1.05650487 1.16961956 1.74700586]
First 10 samples (Inverse CDF Sampling) for **PDF 1**: [1.28440818 1.41887831 1.35436836 1.14120603 1.93200997 1.47813011 1.4724637 1.26052481 1.58910761 1.73445162]

First 10 samples (Rejection Sampling) for **PDF 2**: [0.65980017 0.33591692 0.44772562 0.65241113 0.74987927 0.86683287 0.36578511 0.2965901 0.70363834 0.57029666]
First 10 samples (Inverse CDF Sampling) for **PDF 2**: [0.9925059 0.51585839 0.54601783 0.88039465 0.5035869 0.45795693 0.66617668 0.99060673 0.35787233 0.9946162]

First 10 samples (Rejection Sampling) for **PDF 3**: [1.64399242 1.78343298 1.06672007 1.08821729 1.97816004 1.93789041 1.30835956 1.82414961 0.58174776 1.53543699]
First 10 samples (Inverse CDF Sampling) for **PDF 3**: [0.83591685 1.24063533 1.84835533 1.7767939 0.68084209 1.72253097 1.79747816 1.66938935 0.69765663 0.89811359]

First 10 samples (Rejection Sampling) for **PDF 4**: [0.49912334 2.89073959 1.19933991 0.98124483 0.46504211 1.49379222 0.88277124 1.58780586 0.10544903 1.62477407]
First 10 samples (Inverse CDF Sampling) for **PDF 4**: [3.28341973 3.58810416 3.39853939 1.53191396 1.99587607 2.46021472 2.76220498 2.60050247 1.06280023 1.97718883]