In [4]:
```
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
```

Define Inverse CDF Functions

Define the inverse CDF functions for the Logistic, Rayleigh, and Exponential distributions.

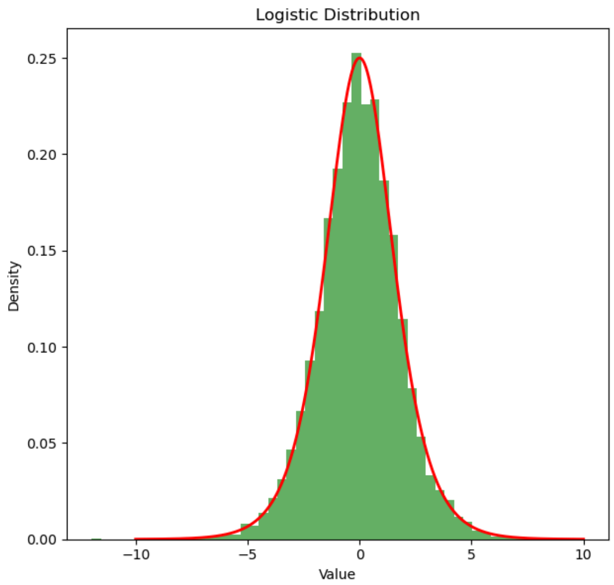
In [5]:
```
def inverse_cdf_logistic(u):
    """Inverse CDF for the Logistic distribution."""
    return np.log(u / (1 - u))

def inverse_cdf_rayleigh(u):
    """Inverse CDF for the Rayleigh distribution."""
    return np.sqrt(-2 * np.log(1 - u))

def inverse_cdf_exponential(u):
    """Inverse CDF for the Exponential distribution."""
    return -np.log(1 - u)
```

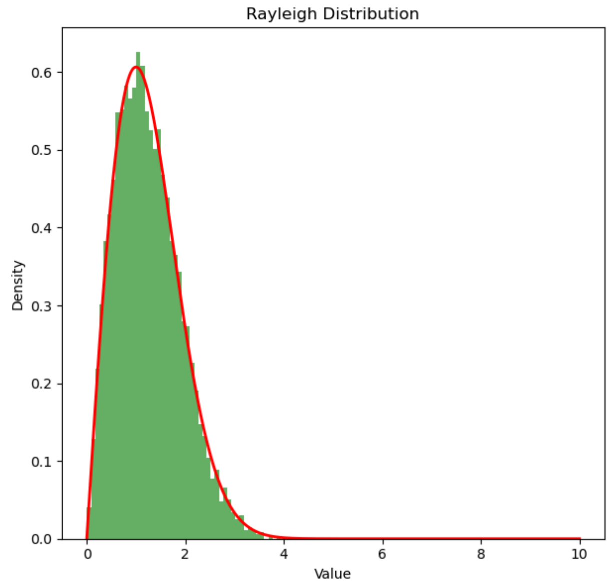
Generate Samples

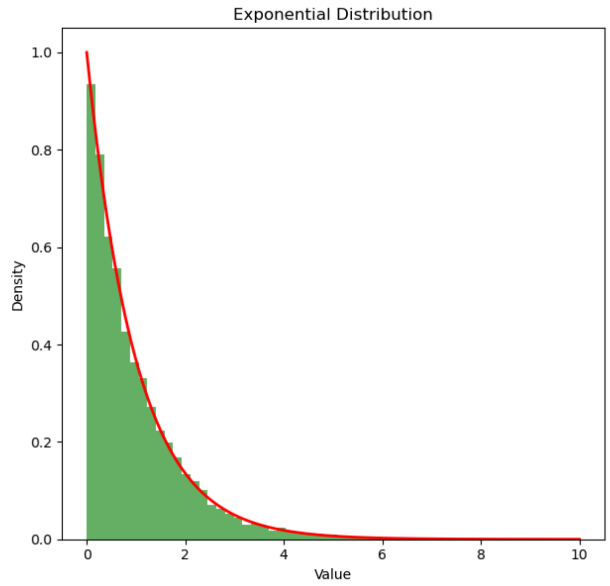
Use the inverse CDF functions to generate samples from each distribution.

In [6]:
```
num_samples = 10000

uniform_samples = np.random.uniform(0, 1, num_samples)

logistic_samples = inverse_cdf_logistic(uniform_samples)
rayleigh_samples = inverse_cdf_rayleigh(uniform_samples)
exponential_samples = inverse_cdf_exponential(uniform_samples)

fig, axs = plt.subplots(3, 2, figsize=(12, 18))

axs[0, 0].hist(logistic_samples, bins=50, density=True, alpha=0.6, color=
x = np.linspace(-10, 10, 1000)
pdf_logistic = np.exp(x) / (1 + np.exp(x))**2
axs[0, 0].plot(x, pdf_logistic, 'r', lw=2)
axs[0, 0].set_title('Logistic Distribution')

axs[1, 0].hist(rayleigh_samples, bins=50, density=True, alpha=0.6, color=
x = np.linspace(0, 10, 1000)
pdf_rayleigh = x * np.exp(-x**2 / 2)
axs[1, 0].plot(x, pdf_rayleigh, 'r', lw=2)
axs[1, 0].set_title('Rayleigh Distribution')

axs[2, 0].hist(exponential_samples, bins=50, density=True, alpha=0.6, col
x = np.linspace(0, 10, 1000)
pdf_exponential = np.exp(-x)
axs[2, 0].plot(x, pdf_exponential, 'r', lw=2)
```

```
axs[2, 0].set_title('Exponential Distribution')

for ax in axs.flat:
    ax.set(xlabel='Value', ylabel='Density')

for i in range(3):
    axs[i, 1].axis('off')

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

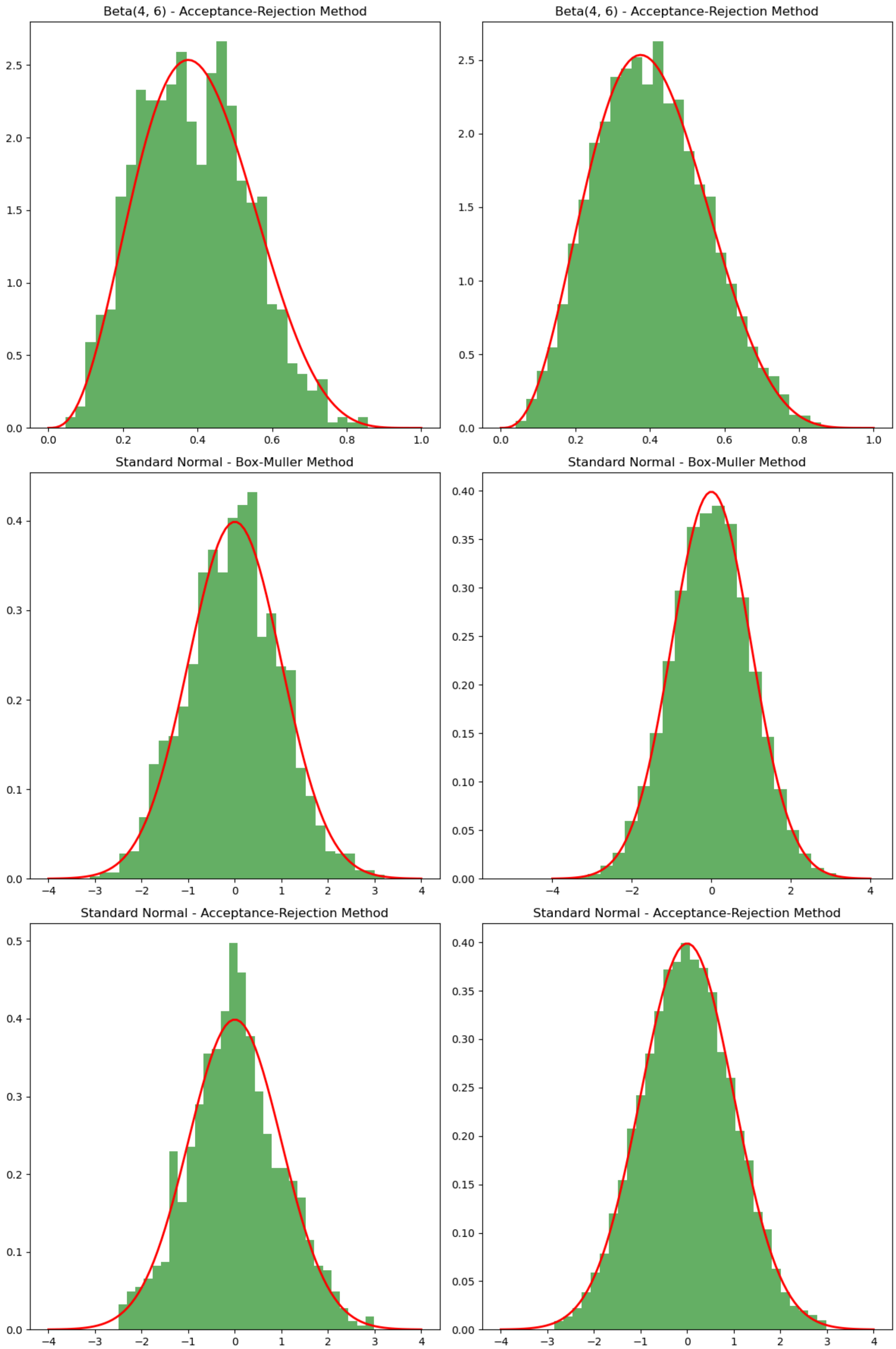

Logistic Distribution
Density
Value
0.00
0.05
0.10
0.15
0.20
0.25
−10
−5
0
5
10


Rayleigh Distribution
Density
Value
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
2
4
6
8
10


Exponential Distribution
Density
Value
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import beta, norm

def beta_acceptance_rejection(alpha, beta, size):
    samples = []
    while len(samples) < size:
        u = np.random.uniform(0, 1)
        v = np.random.uniform(0, 1)
        x = u ** (1 / alpha)
        if v <= (1 - x) ** (beta - 1):
            samples.append(x)
    return np.array(samples)

def box_muller(size):
    u1 = np.random.uniform(0, 1, size)
    u2 = np.random.uniform(0, 1, size)
    z0 = np.sqrt(-2 * np.log(u1)) * np.cos(2 * np.pi * u2)
    z1 = np.sqrt(-2 * np.log(u1)) * np.sin(2 * np.pi * u2)
    return np.concatenate((z0, z1))

def normal_acceptance_rejection(size):
    samples = []
    while len(samples) < size:
        u = np.random.uniform(0, 1)
        v = np.random.uniform(-3, 3)
        if u <= np.exp(-0.5 * v ** 2):
            samples.append(v)
    return np.array(samples)

beta_samples1 = beta_acceptance_rejection(4, 6, 1000)
beta_samples2 = beta_acceptance_rejection(4, 6, 10000)
box_muller_samples1 = box_muller(1000)
box_muller_samples2 = box_muller(10000)
normal_ar_samples1 = normal_acceptance_rejection(1000)
normal_ar_samples2 = normal_acceptance_rejection(10000)

fig, axs = plt.subplots(3, 2, figsize=(12, 18))

axs[0, 0].hist(beta_samples1, bins=30, density=True, alpha=0.6, color='g'
x = np.linspace(0, 1, 100)
axs[0, 0].plot(x, beta.pdf(x, 4, 6), 'r-', lw=2)
axs[0, 0].set_title('Beta(4, 6) - Acceptance-Rejection Method')

axs[0, 1].hist(beta_samples2, bins=30, density=True, alpha=0.6, color='g'
x = np.linspace(0, 1, 100)
axs[0, 1].plot(x, beta.pdf(x, 4, 6), 'r-', lw=2)
axs[0, 1].set_title('Beta(4, 6) - Acceptance-Rejection Method')

axs[1, 0].hist(box_muller_samples1, bins=30, density=True, alpha=0.6, col
x = np.linspace(-4, 4, 100)
axs[1, 0].plot(x, norm.pdf(x), 'r-', lw=2)
axs[1, 0].set_title('Standard Normal - Box-Muller Method')

axs[1, 1].hist(box_muller_samples2, bins=30, density=True, alpha=0.6, col
x = np.linspace(-4, 4, 100)
axs[1, 1].plot(x, norm.pdf(x), 'r-', lw=2)
axs[1, 1].set_title('Standard Normal - Box-Muller Method')
```

```
axs[2, 0].hist(normal_ar_samples1, bins=30, density=True, alpha=0.6, colo
axs[2, 0].plot(x, norm.pdf(x), 'r-', lw=2)
axs[2, 0].set_title('Standard Normal - Acceptance-Rejection Method')

axs[2, 1].hist(normal_ar_samples2, bins=30, density=True, alpha=0.6, colo
axs[2, 1].plot(x, norm.pdf(x), 'r-', lw=2)
axs[2, 1].set_title('Standard Normal - Acceptance-Rejection Method')

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



In terms of sampling Normal distribution, their variance are similar, while the sample efficiency of BoxMuller is higher with also higher running speed.

Box-Muller:

Pros:

It is easy to implement, and the method only uses Unif(0, 1) as the basis data sample, which is simple to sample.

Cons:

Only the standard normal distribution can be sampled by this method.

Acceptance-Rejection:

Pros:

It can sample many kinds of probability distribution including many distributions that is difficult to sample directly.

Cons:

The domain of function $g(x)$ must cover the domain of function $f(x)$. If c is closed to 1, the basis distribution g is still difficult to sample; while if c is closed to 0, the probability of acceptance success will be small, which will cause low efficiency.

p3

December 10, 2024

[7]:
```python
import numpy as np
import math
from decimal import getcontext

from scipy.stats import norm

N = 10_000_00
x = np.random.rand(N)
fx = 4 / (1 + x**2)
estimate_a = np.mean(fx)

print("Monte Carlo estimate result (a):", estimate_a)
print("Theoretical value:", np.pi)

def integrand(x):
  return np.sqrt(x + np.sqrt(x + np.sqrt(x + np.sqrt(x))))

x = np.random.uniform(0, 4, N)
fx = integrand(x)
estimate_b = (4 - 0) * np.mean(fx)

print("Monte Carlo estimate result (b):", estimate_b)

theoretical_c = 1 - norm.cdf(8)

N = 100_0000
mu = 8
y_shifted = np.random.randn(N) + mu

w = norm.pdf(y_shifted, 0, 1) / norm.pdf(y_shifted, 8, 1)

I = (y_shifted > 8).astype(float)

estimate_c = np.mean(I * w)
print("Importance sampling Monte Carlo estimate (c):", estimate_c)
print("Theoretical value:", theoretical_c)
```

```
Monte Carlo estimate result (a): 3.141655418534438
```

```
Theoretical value: 3.141592653589793
Monte Carlo estimate result (b): 7.67652499380115
Importance sampling Monte Carlo estimate (c): 6.211953979014197e-16
Theoretical value: 6.661338147750939e-16
```

SI 140A-02 Probability & Statistics for EECS, Fall 2024
Homework 9

Name:
Student ID:

Due on Dec. 10, 2024, 11:59 UTC+8

Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- You are required to write down all the major steps towards making your conclusions; otherwise you may obtain limited points of the problem.
- Write your homework in English; otherwise you will get no points of this homework.
- Any form of plagiarism will lead to 0 point of this homework.

Note: for the programming assignment, you are required to submit the whole solution in the format of Jupyter Notebook (Formerly known as the IPython Notebook), including source codes, theory, algorithms, simulation result and analysis. Do NOT print the file or the source codes.

Problem 1

Use the methods of inverse transform sampling (or called the method of inverse CDF) to obtain samples from each of the following continuous distributions:

(a) Logistic distribution with CDF $F(x) = e^x/(1+e^x), x \in R$.

(b) Rayleigh distribution with CDF $F(x) = 1 - e^{-x^2/2}, x > 0$.

(c) Exponential distribution with CDF $F(x) = 1 - e^{-x}, x > 0$.

After obtaining enough samples, please plot the corresponding histogram and corresponding theoretical PDF.

Problem 2

Acceptance-Rejection Method

(a) Use the Acceptance-Rejection Method to obtain samples from Beta distribution $\text{Beta}(4, 6)$.

(b) Use both the Box-Muller method and the Acceptance-Rejection Method to obtain samples from the standard Normal distribution $\mathcal{N}(0, 1)$, then discuss the pros and cons of each method.

After obtaining enough samples, please plot the corresponding histogram and corresponding theoretical PDF.

Problem 3

Monte Carlo Integration

(a) Evaluate the integration

$$\int_0^1 \frac{4}{1+x^2}dx.$$

(b) Evaluate the integration

$$\int_0^4 \sqrt{x+\sqrt{x+\sqrt{x+\sqrt{x}}}}dx.$$

(c) Evaluate the probability of rare event $c = \mathbb{P}(Y > 8)$, where $Y \sim \mathcal{N}(0,1)$.

Problem 4

Use your own words to describe the geometric perspective of Jacobian Matrix and Jacobian Determinant.

Solution:

The Jacobian matrix of a vector-valued function represents the best linear approximation to the function at a given point. Geometrically, it describes how the function stretches, rotates, and shears the space around that point. Each entry in the Jacobian matrix corresponds to the partial derivative of one component of the function with respect to one of the input variables, indicating how sensitive that component is to changes in that variable.

The Jacobian determinant provides a measure of how the function transforms volumes near a given point. Geometrically, it represents the factor by which the function scales volumes in the input space when mapping them to the output space. If the determinant is positive, the orientation of the space is preserved; if negative, the orientation is reversed. A determinant of zero indicates that the function compresses the space into a lower dimension at that point.

In summary, the Jacobian matrix describes the local linear transformation properties of a function, while the Jacobian determinant quantifies the local volume scaling effect of that transformation.

Problem 5

The PDF of Gamma distribution Gamma(a, λ) is:

$$f(x) = \frac{1}{\Gamma(a)}(\lambda x)^a e^{-\lambda x}\frac{1}{x}, x > 0,$$

where $a > 0, \lambda > 0$, and $\Gamma(a)$ is the Gamma Function:

$$\Gamma(a) = \int_0^\infty z^{a-1}e^{-z}dz.$$

(a) If $X \sim$ Gamma(a, λ), find $E(X)$ and Var(X).

(b) If $Y \sim \mathcal{N}(0, 1)$, show that $Y^2 \sim$ Gamma$\left(\frac{1}{2}, \frac{1}{2}\right)$.

(c) If $V = Y_1^2 + \ldots + Y_n^2$, where $Y_i, i = 1, \ldots, n$ are i.i.d. random variables and $Y_i \sim \mathcal{N}(0, 1)$, then V satisfies chi-square distribution, i.e. $V \sim \chi_n^2$. Show that $V \sim$ Gamma$\left(\frac{n}{2}, \frac{1}{2}\right)$ and find the PDF of V.

(d) If Y and V are independent, define random variable Z as follows

$$Z = \frac{Y}{\sqrt{\frac{V}{n}}}.$$

then Z satisfies Student's t-distribution, i.e. $Z \sim t_n$. Please adopt the change of variable method to find the PDF of Z.

(e) Given two independent random variables V_1 and V_2, where $V_1 \sim \chi_m^2$ and $V_2 \sim \chi_n^2$. Define random variable W as follows

$$W = \frac{\frac{V_1}{m}}{\frac{V_2}{n}}.$$

then W satisfies F -distribution, i.e. $W \sim F(m, n)$. Please adopt the change of variable method to find the PDF of W.

Solution:

(a):

To find the expected value $E(X)$ and variance Var(X) of a Gamma distribution $X \sim$ Gamma(a, λ), we use the properties of the Gamma function.
The expected value is given by:

$$E(X) = \int_0^\infty x\frac{1}{\Gamma(a)}(\lambda x)^a e^{-\lambda x}\frac{1}{x}dx = \frac{\Gamma(a+1)}{\lambda\Gamma(a)} = \frac{a}{\lambda}$$

The second moment is:

$$E(X^2) = \int_0^\infty x^2\frac{1}{\Gamma(a)}(\lambda x)^a e^{-\lambda x}\frac{1}{x}dx = \frac{\Gamma(a+2)}{\lambda^2\Gamma(a)} = \frac{a(a+1)}{\lambda^2}$$

Thus, the variance is:

$$\text{Var}(X) = E(X^2) - (E(X))^2 = \frac{a(a+1)}{\lambda^2} - \left(\frac{a}{\lambda}\right)^2 = \frac{a}{\lambda^2}$$

Problem 5 continued on next page...

(b):

To show that $Y^2 \sim \text{Gamma}\left(\frac{1}{2}, \frac{1}{2}\right)$ for $Y \sim \mathcal{N}(0,1)$, we start with the PDF of the standard normal distribution:

$$\int_{-\infty}^{\infty} f(x)dx = 1 = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$$

Using the change of variable $u = \frac{x^2}{2}$, we get:

$$\frac{1}{\sqrt{2\pi}} \int_{0}^{\infty} \frac{1}{\sqrt{u}} e^{-u} du = \frac{1}{\sqrt{\pi}} \Gamma\left(\frac{1}{2}\right)$$

Since $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$, we have:

$$f(x) = \frac{1}{\sqrt{2\pi x}} e^{-\frac{x}{2}}$$

This matches the PDF of the Gamma distribution $\text{Gamma}\left(\frac{1}{2}, \frac{1}{2}\right)$:

$$f_G(x) = \frac{1}{\Gamma\left(\frac{1}{2}\right)} \left(\frac{1}{2}x\right)^{\frac{1}{2}-1} e^{-\frac{x}{2}} = \frac{1}{\sqrt{2\pi x}} e^{-\frac{x}{2}}$$

Thus, $Y^2 \sim \text{Gamma}\left(\frac{1}{2}, \frac{1}{2}\right)$.
(c):

To show that $V = Y_1^2 + \ldots + Y_n^2 \sim \text{Gamma}\left(\frac{n}{2}, \frac{1}{2}\right)$, we use the property that the sum of independent Gamma random variables is also Gamma distributed.
Assume $X \sim \text{Gamma}(a, \lambda)$ and $Y \sim \text{Gamma}(b, \lambda)$. Then $Z = X + Y$ has the PDF:

$$f_Z(z) = \int_{0}^{z} f_X(x) f_Y(z-x) dx = \frac{e^{-z\lambda}\lambda^{a+b}}{\Gamma(a)\Gamma(b)} \int_{0}^{z} x^{a-1}(z-x)^{b-1} dx$$

Using the Beta function identity, we get:

$$f_Z(z) = \frac{\lambda^{a+b}}{\Gamma(a+b)} z^{a+b-1} e^{-z\lambda}$$

Thus, $V = Y_1^2 + \ldots + Y_n^2 \sim \text{Gamma}\left(\frac{n}{2}, \frac{1}{2}\right)$.
(d):

To find the PDF of $Z = \frac{Y}{\sqrt{\frac{V}{n}}}$ where $Y \sim \mathcal{N}(0,1)$ and $V \sim \chi_n^2$, we use the joint PDF of Y and V:

$$f_{Y,V}(y,v) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \frac{1}{\Gamma\left(\frac{n}{2}\right)} \left(\frac{v}{2}\right)^{\frac{n}{2}-1} e^{-\frac{v}{2}}$$

Using the change of variables and the Jacobian determinant $|J| = \frac{v}{n}$, we get:

$$f_Z(z) = \int_{0}^{\infty} f_{Z,V}(z,v) dv = 2\frac{1}{\sqrt{2\pi n}} \frac{1}{\Gamma\left(\frac{n}{2}\right)} \int_{0}^{\infty} e^{-\frac{z^2 v/n^2}{2}} \left(\frac{v}{2}\right)^{\frac{n-1}{2}} e^{-\frac{v}{2}} d\left(\frac{v}{2}\right)$$

This simplifies to:

$$f_Z(z) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)} \frac{\left(1 + \frac{z^2}{n}\right)^{-\frac{n+1}{2}}}{\sqrt{\pi n}}$$

(e):

Problem 5 continued on next page...

To find the PDF of $W = \frac{\frac{V_1}{m}}{\frac{V_2}{n}}$ where $V_1 \sim \chi_m^2$ and $V_2 \sim \chi_n^2$, we use the joint PDF and the Jacobian determinant $|J| = \frac{m}{n}z$:

$$f_W(w) = \frac{\Gamma\left(\frac{m+n}{2}\right)}{\Gamma\left(\frac{m}{2}\right)\Gamma\left(\frac{n}{2}\right)} \left(\frac{m}{n}\right)^{\frac{m}{2}} w^{\frac{m}{2}-1} \left(1 + \frac{nw}{m}\right)^{-\frac{m+n}{2}}$$

Problem 6

(**Optional Challenging Problem**). Given n i.i.d. continuous random variables $X_1, \ldots, X_n$, where $X_i \sim F_X$. Define

$$D_n = \sup_{x \in \mathbb{R}} \left| \hat{F}_n(x) - F_X(x) \right|.$$

where

$$\hat{F}_n(x) = \frac{1}{n} \sum_{j=1}^{n} I\left(X_j \leq x\right)$$

(a) Find the CDF of D_n.

(b) Find the limiting CDF of $\sqrt{n} D_n$ when $n \to \infty$.