Question 1

1. Let X_1, X_2, X_3 be independent Gamma random variables with scale parameter $\lambda = 2$ and shape parameters $k_1 = 1$, $k_2 = 2$ and $k_3 = 3$, respectively.

a. Find the moment generating function of X_i ($i = 1, 2, 3$).

Solution.

For $X \sim \Gamma(\lambda, k)$ the MGF is

$$M_X(s) = E(e^{sX}) = \left(\frac{\lambda}{\lambda - s}\right)^k, \qquad s < \lambda.$$

Thus, with $\lambda = 2$,

$$M_{X_1}(s) = \frac{2}{2-s}, \qquad M_{X_2}(s) = \left(\frac{2}{2-s}\right)^2, \qquad M_{X_3}(s) = \left(\frac{2}{2-s}\right)^3, \quad s < 2.$$

□

b. Find the probability distribution of aX_i for $a > 0$ ($i = 1, 2, 3$).

Solution.

Using $M_{aX}(s) = M_X(as)$,

$$M_{aX_i}(s) = \left(\frac{2}{2 - as}\right)^{k_i} = \left(\frac{2/a}{2/a - s}\right)^{k_i},$$

which is the MGF of $\Gamma(2/a, k_i)$. Hence $aX_i \sim \Gamma(2/a, k_i)$.

□

c. Find the probability distribution of $X_1 + X_2 + X_3$.

Solution.

Independence and common λ give

$$M_{X_1+X_2+X_3}(s) = \prod_{i=1}^{3} M_{X_i}(s) = \left(\frac{2}{2-s}\right)^{1+2+3} = \left(\frac{2}{2-s}\right)^6.$$

Therefore $X_1 + X_2 + X_3 \sim \Gamma(2, 6)$.

□

d. Find the probability distribution of $(X_1 + X_2 + X_3)/3$.

Solution.

Let $S = X_1 + X_2 + X_3 \sim \Gamma(2, 6)$. For $a = 1/3$,

$$\frac{S}{3} \sim \Gamma\left(\frac{2}{1/3}, 6\right) = \Gamma(6, 6).$$

Equivalently, $M_{S/3}(s) = M_S(s/3) = (\frac{6}{6-s})^6$.

□

e. Find $E(X_1 + X_2 + X_3)^2$.

Solution.

From $S \sim \Gamma(2, 6)$,

$$E(S) = \frac{6}{2} = 3, \qquad V(S) = \frac{6}{2^2} = \frac{3}{2}.$$

Thus

$$E\big((X_1 + X_2 + X_3)^2\big) = E(S^2) = V(S) + [E(S)]^2 = \frac{3}{2} + 9 = \frac{21}{2} = 10.5.$$

□

Question 2

a. Generate in Matlab or in Python a sample $(x_{1,i}, x_{2,i}, x_{3,i})$ with $(i = 1, \dots, 1000)$, of size 1000 from the distribution of three independent Gamma random variables with scale parameter $\lambda = 2$ and shape parameter $k = 1$.

Solution.

```python
import numpy as np

# Reproducibility
rng = np.random.default_rng(42)

# Sample of size 1000: three independent Gamma(k=1, scale=2)
n = 1000
scale = 2.0
X = rng.gamma(shape=1.0, scale=scale, size=(n, 3))
x1, x2, x3 = X.T
```

□

b. Compute $y_i = x_{1,i} + x_{2,i} + x_{3,i}$.

Solution.

```python
import matplotlib.pyplot as plt
from math import gamma as gamma_fn

y = x1 + x2 + x3  # In theory: y ~ Gamma(k=3, scale=2)

def gamma_pdf(x, k, theta):
    x = np.asarray(x)
    return np.where(
        x >= 0,
        (x ** (k - 1)) * np.exp(-x / theta) / (gamma_fn(k) * (theta ** k)),
        0.0,
    )

# Histogram for z with theoretical \Gamma(3,2) PDF
fig, ax = plt.subplots()
ax.hist(
    y,
    bins="fd",
    density=True,
    alpha=0.7,
    edgecolor="black",
    linewidth=0.6,
)

```

```
x_max = max(scale * 3 * 2.5, float(np.percentile(y, 99.5)))
grid = np.linspace(0.0, x_max, 600)
ax.plot(grid, gamma_pdf(grid, 3.0, scale), linewidth=2.0, label="Gamma(k=3, scale=2) PDF")

ax.set_title("Distribution of y = x1 + x2 + x3")
ax.set_xlabel("value")
ax.set_ylabel("density")
ax.grid(True, alpha=0.3, linestyle="--", linewidth=0.5)
for spine in ("top", "right"):
    ax.spines[spine].set_visible(False)
ax.legend(frameon=False)
fig.tight_layout()
plt.show()
```

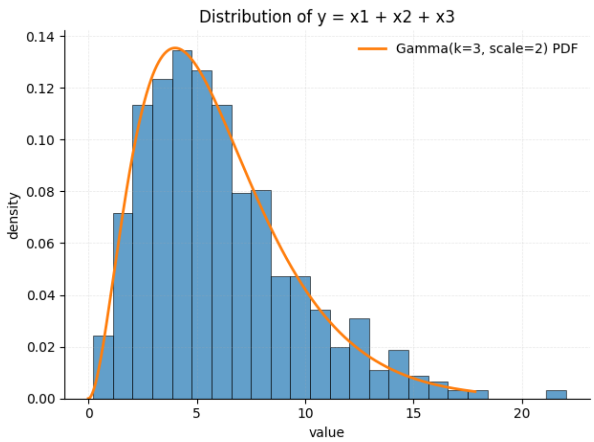

Figure 0-1: Distribution of y

$\square$

c. Generate a random sample z_i ($i = 1, \dots, 1000$) from a Gamma distribution with scale parameter $\lambda = 2$ and shape parameter $k = 3$.

Solution.

```
import matplotlib.pyplot as plt
from math import gamma as gamma_fn

z = rng.gamma(shape=3.0, scale=scale, size=n)

def gamma_pdf(x, k, theta):
    x = np.asarray(x)
    return np.where(
        x >= 0,
        (x ** (k - 1)) * np.exp(-x / theta) / (gamma_fn(k) * (theta ** k)),
        0.0,
    )

# Histogram for z with theoretical \Gamma(3,2) PDF
fig, ax = plt.subplots()
ax.hist(
    z,
    bins="fd",
    density=True,
    alpha=0.7,
    edgecolor="black",
    linewidth=0.6,
)
```

```python

x_max = max(scale * 3 * 2.5, float(np.percentile(z, 99.5)))
grid = np.linspace(0.0, x_max, 600)
ax.plot(grid, gamma_pdf(grid, 3.0, scale), linewidth=2.0, label="Gamma(k=3,
     scale=2) PDF")

ax.set_title("Distribution of z ~ Gamma(3, 2)")
ax.set_xlabel("value")
ax.set_ylabel("density")
ax.grid(True, alpha=0.3, linestyle="--", linewidth=0.5)
for spine in ("top", "right"):
    ax.spines[spine].set_visible(False)
ax.legend(frameon=False)
fig.tight_layout()
plt.show()
```

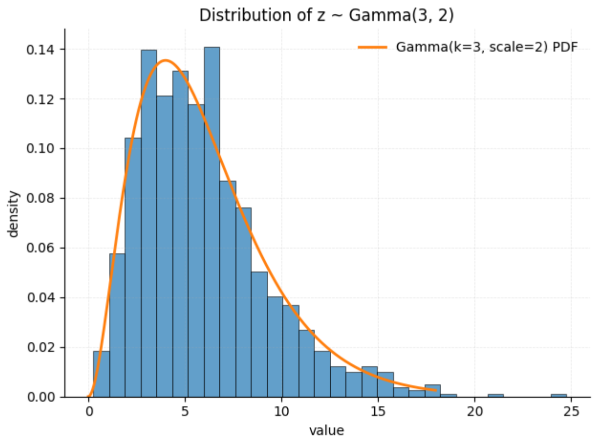

Figure 0-2: Distribution of z

□

d. Compare the histogram of $y_1, \ldots, y_{1000}$ with the histogram of $z_1, \ldots, z_{1000}$.

Solution.

```python
import matplotlib.pyplot as plt
from scipy import stats

combined = np.concatenate([y, z])
bins = np.histogram_bin_edges(combined, bins="fd")  # common bins for fair
    comparison

fig, ax = plt.subplots()
ax.hist(y, bins=bins, density=True, alpha=0.5, edgecolor="black", linewidth
    =0.6, label="y = x1 + x2 + x3")
ax.hist(z, bins=bins, density=True, histtype="step", linewidth=2.0, label="
    z ~ Gamma(3, 2)")

ax.set_title("Comparison: y vs z")
ax.set_xlabel("value")
ax.set_ylabel("density")
ax.grid(True, alpha=0.3, linestyle="--", linewidth=0.5)
for spine in ("top", "right"):
    ax.spines[spine].set_visible(False)
ax.legend(frameon=False)
fig.tight_layout()
plt.show()

```

```python
# --- Distribution comparison tests ---
y_f = y[np.isfinite(y)]
z_f = z[np.isfinite(z)]

# 1) Kolmogorov-Smirnov two-sample test (global, bin-free)
ks = stats.ks_2samp(y_f, z_f, alternative="two-sided", method="auto")

# 2) Cramer-von Mises two-sample test (sensitive across the whole CDF)
cvm = stats.cramervonmises_2samp(y_f, z_f)

# 3) Wasserstein-1 distance (Earth Mover's Distance)
w1 = stats.wasserstein_distance(y_f, z_f)

print(f"KS test:                    D = {ks.statistic:.4f}, p = {ks.pvalue:.4g}")
print(f"Cramer-von Mises test:      T = {cvm.statistic:.4f}, p = {cvm.pvalue:.4g}")
print(f"Wasserstein-1 distance:  W-1 = {w1:.4f}")
```

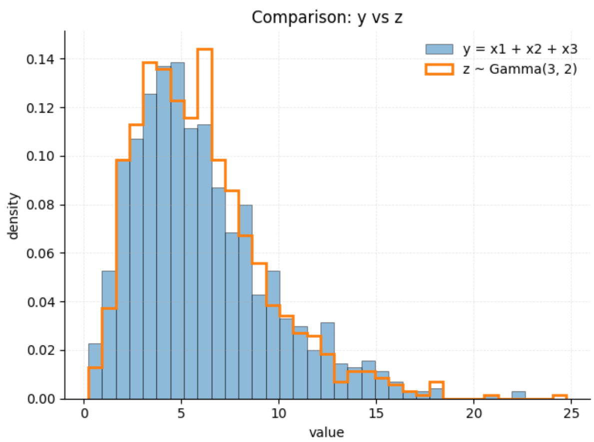

Figure 0-3: Comparison of y and z

Across 1,000 draws, the two-sample tests show no evidence that the samples differ: KS $D = 0.035$ ($p = 0.573$) and Cramér–von Mises $T = 0.106$ ($p = 0.559$) both fail to reject the null that y_i (sum of three $\Gamma(1,2)$) and z_i (direct $\Gamma(3,2)$) come from the same distribution. The Wasserstein distance $W_1 = 0.204$ is small in natural units—about $0.204/\sqrt{12} \approx 6\%$ of one standard deviation for $\Gamma(3,2)$—well within Monte Carlo variability for $n = 1000$.

Overall, results are consistent with the additivity property of the Gamma family: $x_1 + x_2 + x_3 \sim \Gamma(3,2)$.

□