Question 1

Let Z be a random variable with uniform distribution on $(0, 1)$ and let X_1, X_2 be conditionally independent and identically distributed, given Z with $P(X_i = 1|Z) = 1 - P(X_i = 0|Z) = Z$ $(i = 1, 2)$.

a. Find $E(X_i)$ and $V(X_i)$ $(i = 1, 2)$.

Solution.

By the law of total expectation, $E(X_i) = E(E(X_i \mid Z)) = E(Z)$.

Since $Z \sim \text{Unif}(0, 1)$, $E(Z) = \int_0^1 z\,dz = \frac{1}{2}$, hence

$$E(X_i) = \frac{1}{2}.$$

For the variance, we use the variance decomposition (i.e. law of total variance):

$$V(X_i) = E[V(X_i \mid Z)] + V(E(X_i \mid Z)) = E[Z(1 - Z)] + V(Z).$$

Now $E[Z(1-Z)] = E(Z) - E(Z^2) = \frac{1}{2} - \frac{1}{3} = \frac{1}{6}$, and $V(Z) = E(Z^2) - E(Z)^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$. Therefore

$$V(X_i) = \frac{1}{6} + \frac{1}{12} = \frac{1}{4}.$$

□

b. Find $Cov(X_1, X_2)$.

Solution.

Conditional on Z, X_1 and X_2 are independent, so

$$E(X_1X_2) = E(E(X_1X_2 \mid Z)) = E(E(X_1 \mid Z)E(X_2 \mid Z)) = E(Z^2) = \frac{1}{3}.$$

Thus

$$\text{Cov}(X_1, X_2) = E(X_1X_2) - E(X_1)E(X_2) = \frac{1}{3} - \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}.$$

□

c. Are X_1, X_2 stochastically independent?

Solution.

No. Independence would require $P(X_1 = 1, X_2 = 1) = P(X_1 = 1)P(X_2 = 1)$.

But

$$P(X_1 = 1, X_2 = 1) = E(Z^2) = \frac{1}{3} \neq (E(Z))^2 = \left(\frac{1}{2}\right)^2 = \frac{1}{4}.$$

Hence they are not independent (they are exchangeable with positive covariance).

□

d. Find $P(Z \leq 1/2 \mid X_1 = 1)$.

Solution.

By Bayes' rule with a continuous prior (the *Bayesian scheme*), the posterior of Z after one success is

$$Z \mid X_1 = 1 \sim \text{Beta}(1 + 1, 1 + 0) = \text{Beta}(2, 1),$$

with density $f(z \mid X_1 = 1) = 2z\,\mathbf{1}_{(0,1)}(z)$.

Therefore

$$P(Z \leq \tfrac{1}{2} \mid X_1 = 1) = \int_0^{1/2} 2z\,dz = z^2\Big|_0^{1/2} = \frac{1}{4}.$$

□

Question 2

1. Generate in Python 1000 observations (x_i, y_i) from a bivariate normal distribution with mean vector $(0,0)$ and variance-covariance matrix

$$\begin{bmatrix} 1 & 0.2 \\ 0.2 & 1 \end{bmatrix}.$$

Solution.

```python
import numpy as np

# Reproducibility
rng = np.random.default_rng(40313)

# Mean vector and covariance matrix
mu = np.array([0.0, 0.0])
Sigma = np.array([[1.0, 0.2],
                  [0.2, 1.0]])

# Draw 1000 observations
data = rng.multivariate_normal(mean=mu, cov=Sigma, size=1000)
x = data[:, 0]
y = data[:, 1]
```

□

2. Draw an histogram of $x_1, \ldots, x_{1000}$.

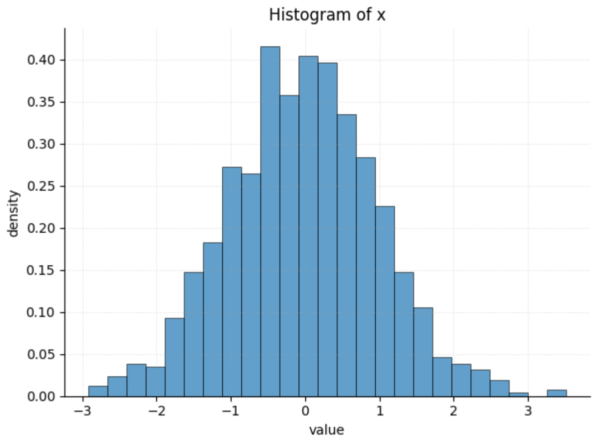
Solution.

```python
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
ax.hist(
    x,
    bins="fd",
    density=True,
    alpha=0.7,
    edgecolor="black",
    linewidth=0.6,
)

ax.set_title("Histogram of x")
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)
fig.tight_layout()
plt.show()
```

□



3. Draw an histogram of those y_i such that $x_i \leq 0$.

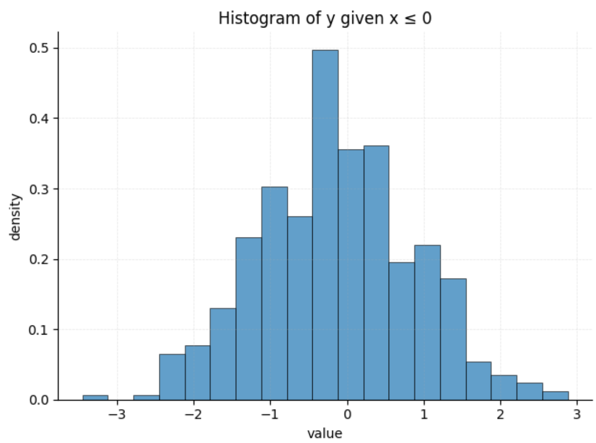
Solution.

```python
import matplotlib.pyplot as plt
import numpy as np

mask = x <= 0
y_subset = y[mask]

fig, ax = plt.subplots()
ax.hist(
    y_subset,
    bins="fd",
    density=True,
    alpha=0.7,
    edgecolor="black",
    linewidth=0.6,
)

ax.set_title("Histogram of y given x \leq 0")
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)
fig.tight_layout()
plt.show()
```



$\square$