# Exercise 3

1. Suppose $X \sim Pois(\lambda)$ $\lambda > 0$, hence $P(X=x) = \frac{e^{-\lambda}\lambda^x}{x!}$.

(a) $M_X(t) = E[e^{tX}] = \sum_{x=0}^{\infty} e^{tx}\frac{e^{-\lambda}\lambda^x}{x!} = e^{-\lambda}\sum_{x=0}^{\infty}\frac{(\lambda e^t)^x}{x!} = e^{-\lambda}e^{\lambda e^t} = e^{\lambda(e^t-1)}$

(b) Define the comulant $n$ as $K_n = \frac{d^n}{dt^n}\ln M_X(t)\Big|_{t=0}$

(c) $\forall n\in\mathbb{N}$ $\frac{d^n}{dt^n}\ln e^{\lambda(e^t-1)}\Big|_{t=0} = \lambda\frac{d^n}{dt^n}(e^t-1)\Big|_{t=0} = \lambda e^t\Big|_{t=0} = \lambda$

(d) $m_1 = \frac{d}{dt}e^{\lambda(e^t-1)}\Big|_{t=0} = \lambda e^t e^{\lambda(e^t-1)}\Big|_{t=0} = \lambda = K_1$

$m_2 = \frac{d^2}{dt^2}e^{\lambda(e^t-1)}\Big|_{t=0} = \lambda e^t e^{\lambda(e^t-1)} + \lambda^2 e^{2t}e^{\lambda(e^t-1)}\Big|_{t=0} = \lambda+\lambda^2 \rightarrow m_2 - m_1^2 = \lambda = K_2$

$m_3 = \frac{d^3}{dt^3}e^{\lambda(e^t-1)}\Big|_{t=0} = \lambda e^t e^{\lambda(e^t-1)} + \lambda^2 e^{2t}e^{\lambda(e^t-1)} + 2\lambda^2 e^{2t}e^{\lambda(e^t-1)} + \lambda^3 e^{3t}e^{\lambda(e^t-1)}\Big|_{t=0} = \lambda+3\lambda^2+\lambda^3 \rightarrow$

$(\lambda+3\lambda^2+\lambda^3) - 3\lambda(\lambda+\lambda^2) + 2\lambda^3 = \lambda = K_3$

(e) $K_1 = m_1 = \lambda$

$K_2 = m_2 - m_1^2 \rightarrow \lambda = m_2 - \lambda^2 \rightarrow m_2 = \lambda^2+\lambda$

$K_3 = m_3 - 3m_1m_2 + 2m_1^3 \rightarrow \lambda = m_3 - 3\lambda(\lambda^2+\lambda) + 2\lambda^3 \rightarrow m_3 = \lambda^3+3\lambda^2+\lambda$

2. Suppose $X$ is a random variable

(a) Let $M_X(t) = e^{ct}$ where $c\in\mathbb{R}$, then $E[e^{tX}] = e^{ct}$ implies that $X=c$, meaning that $F_X(x) = \begin{cases} 0 & x<c \\ 1 & x\geq c\end{cases}$

(b) Let $M_X(t) = \frac{1}{2}+\frac{1}{4}e^t+\frac{1}{4}e^{-t}$. This points to a discrete PMF like $\sum_{x=1}^{3} e^{tx}P(X=x) = e^{t0}\frac{1}{2}+e^{t\cdot 1}\frac{1}{4}+e^{t(-1)}\frac{1}{4}$,
meaning that the PMF is $P(X=x) = \begin{cases} 1/4 & x=-1 \\ 1/2 & x=0 \\ 1/4 & x=1\end{cases}$ and the CMF is $F_X(x) = \begin{cases} 0 & x<-1 \\ 1/4 & -1\leq x<0 \\ 3/4 & 0\leq x<1 \\ 1 & 1\leq x\end{cases}$.

(c) Let $M_X(t) = \frac{e^t}{2-e^t}$ where $t<\ln 2$. Then $e^t<2$ and $p=\frac{1}{2}e^t<1$. Hence, $\frac{e^t}{2-e^t} = \frac{2p}{2-2p} = \frac{p}{1-p} =$
$p\sum_{k=0}^{\infty}p^k = \sum_{k=1}^{\infty}p^k = \sum_{k=1}^{\infty}\frac{e^{kt}}{2^k}$. This again point to a discrete PMF. $e^{t\cdot 1}\cdot\frac{1}{2}+e^{t\cdot 2}\frac{1}{2^2}+e^{t\cdot 3}\frac{1}{2^3}+\ldots$ which matches the PMF
$P(X=x) = \frac{1}{2^x}$ with the support being $X\geq 1$. This is a valid PMF since $\sum_{k=1}^{\infty}\frac{1}{2^k} = \frac{1}{2}\frac{1}{1-1/2} = 1$.
Hence $\forall x\geq 1$ $F_X(x) = \sum_{k=1}^{\lfloor x\rfloor}\frac{1}{2^k} = \frac{1}{2}\frac{1-\frac{1}{2^{\lfloor x\rfloor}}}{1-\frac{1}{2}} = 1-\frac{1}{2^{\lfloor x\rfloor}}$ and $F_X(x)=0$ otherwise.

3. Let $y>0$ and $g(y)=\frac{1}{y}$ then $\frac{d^2}{dy^2}\frac{1}{y} = \frac{2}{y^3}>0$, meaning that $g$ is a convex function on $y>0$. Let $X$ a positive random variable, then according to Jensen's inequality $E[g(X)]\geq g(E[X]) \rightarrow E\left[\frac{1}{X}\right]\geq\frac{1}{E[X]}$.

4. Suppose $X$ is a random variable, $n\in\mathbb{N}$, and $P_X(X=x_i)=\frac{1}{n}$ for $i=1,\ldots,n$. Since $\frac{d^2}{dx^2}\ln x = -\frac{1}{x^2}<0$, then $\ln x$ is concave and according to Jensen's inequality $\ln E[X]\geq E[\ln X]$. Hence $\ln\frac{1}{n}\sum_{k=1}^{n}x_k\geq\frac{1}{n}\sum_{k=1}^{n}\ln x_k \rightarrow$
$\ln\frac{1}{n}\sum_{k=1}^{n}x_k\geq\ln\left(\prod_{k=1}^{n}x_k\right)^{1/n}$. Since $\ln$ is an increasing function we can conclude that $\frac{1}{n}\sum_{k=1}^{n}x_k\geq\left(\prod_{k=1}^{n}x_k\right)^{1/n}$.

5. Suppose $X$ is a random variable whose MGF is $M_X(t)$ defined $\forall t\in\mathbb{R}$.

(a) Let $t\geq 0$, $Y=e^{tX}$, then according to Markov inequality $P(Y\geq y)\leq\frac{E[Y]}{y} \rightarrow P(e^{tX}\geq e^{tx})\leq\frac{E[e^{tX}]}{e^{tx}}$.

Since $e^{tX}$ is increasing, this is also equivalent to $P(X \geq x) \leq \frac{E[e^{tX}]}{e^{tx}}$. In general, $P(X \geq x) \leq \min_{t \geq 0} \frac{E[e^{tX}]}{e^{tx}}$.

But $\min_{t \geq 0} \frac{E[e^{tX}]}{e^{tx}} = \min_{t \geq 0} e^{\ln M_X(t) - tX} = e^{\min_{t \geq 0}(\ln M_x(t) - tx)} = e^{-\max_{t \geq 0}(tx - \ln M_x(t))}$, hence $P(X \geq x) \leq e^{-\Lambda(x)}$

where $\Lambda(x) = \max_{t \geq 0}(tx - \ln M_x(t))$.

(b) Let $X_1, X_2, \dots, X_n$ iid's, $X_i \sim B(p)$ and $\bar{X}$ their arithmetic mean. $M_{X_1} = E[e^{tX_1}] = e^{t \cdot 1} \cdot p + e^{t \cdot 0}(1-p) = p(e^t - 1) + 1$

$M_{\bar{X}} = E[e^{t\bar{X}}] = E[e^{\frac{t}{n}\sum X_i}] = E[\prod_{i=1}^{n} e^{\frac{t}{n}X_i}] = E[e^{\frac{t}{n}X_1}]^n = \left(M_{X_1}\left(\frac{t}{n}\right)\right)^n$. Hence $\Lambda(x) = \max_{t \geq 0}\left(tx - n \cdot \ln\left(M_{X_1}\left(\frac{t}{n}\right)\right)\right) =$

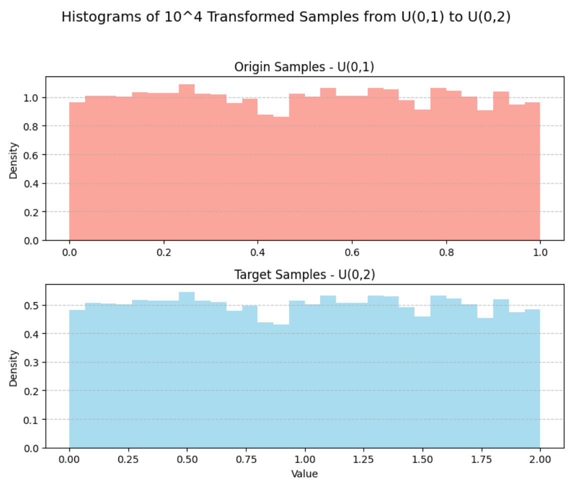
$\max_{t \geq 0}\left(tx - n \cdot \ln\left(p(e^{\frac{t}{n}} - 1) + 1\right)\right)$. $\frac{d}{dt}\left(tx - n \cdot \ln\left(p(e^{\frac{t}{n}} - 1) + 1\right)\right) = x - \frac{pe^{t/n}}{p(e^{\frac{t}{n}} - 1) + 1} = 0 \rightarrow xpe^{\frac{t}{n}} - xp + x = pe^{\frac{t}{n}}$

$\rightarrow e^{\frac{t}{n}} = \frac{xp - x}{xp - p} \rightarrow t = n \cdot \ln \frac{x(p-1)}{p(x-1)}$. Hence, $\Lambda(x) = xn \cdot \ln \frac{x(p-1)}{p(x-1)} - n \cdot \ln \frac{p-1}{x-1}$ and $P(\bar{X} \geq x) \leq e^{-\Lambda(x)}$.

6. Suppose $X$ is a continuous random variable with strictly increasing CDF $F_X(x)$ and $V \sim U(0,1)$.

(a) Let $X \sim U(0,a)$ for $a > 0$, then $f_X(x) = \frac{1}{a}$, $F_X(x) = \int_0^x \frac{1}{a}\,dt = \frac{x}{a}$, and $X = F_X^{-1}(v) = av$.

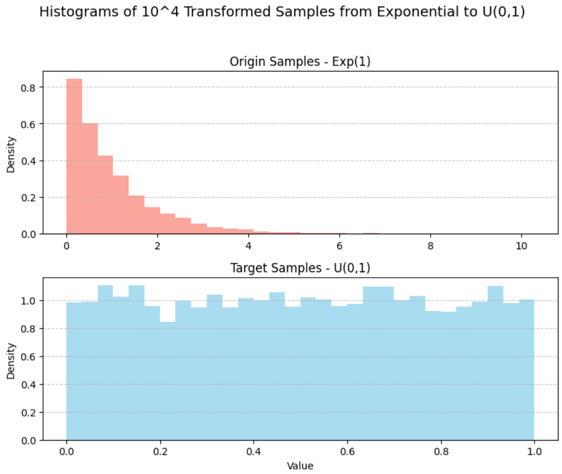
```python
import numpy as np
import matplotlib.pyplot as plt

def plot_histograms(
    origin_samples,
    target_samples,
    origin_label="Origin Samples",
    target_label="Target Samples",
    main_title="Sample Comparison",
    bins=30,
):
    """
    Plots normalized histograms for two sets of samples on separate subplots,
    without legends within each subplot.
    """
    fig, axs = plt.subplots(2, 1, figsize=(8, 7), sharex=False)
    fig.suptitle(main_title, fontsize=14)

    # Plot origin histogram
    axs[0].hist(origin_samples, bins=bins, density=True, alpha=0.7, color="salmon")
    axs[0].set_title(origin_label)
    axs[0].set_ylabel("Density")
    axs[0].grid(axis="y", linestyle="--", alpha=0.7)

    # Plot target histogram
    axs[1].hist(target_samples, bins=bins, density=True, alpha=0.7, color="skyblue")
    axs[1].set_title(target_label)
    axs[1].set_xlabel("Value")
    axs[1].set_ylabel("Density")
    axs[1].grid(axis="y", linestyle="--", alpha=0.7)

    plt.tight_layout(rect=[0, 0.03, 1, 0.95])
    plt.show()


# ---- Part A ----

# Parameters for the target distribution
a = 2
N = 10000
INVERSE_UNI_CDF = lambda v: a * v

# Generate uniform samples from U(0,1)
uni_samples= np.random.uniform(0, 1, N)

# Generate samples from the target distribution using inverse transform sampling
target_samples = INVERSE_UNI_CDF(uni_samples)

# Plot the histogram of the generated samples
plot_histograms(
    uni_samples,
    target_samples,
    origin_label="Origin Samples - U(0,1)",
    target_label="Target Samples - U(0,2)",
    main_title="Histograms of 10^4 Transformed Samples from U(0,1) to U(0,2)",
    bins=30,
)
```




(b) Let $X \sim Exp(1)$, then $f_X(x) = e^{-x}$, $Y = F_X(x) = \int_0^x e^{-t}\,dt = -[e^{-t}]_0^x = 1 - e^{-x}$, and $Y \sim U(0,1)$.

For sampling $X$ just like in part (a), we also need $F$'s inverse: $F^{-1}(v) = -\ln(1-v)$.

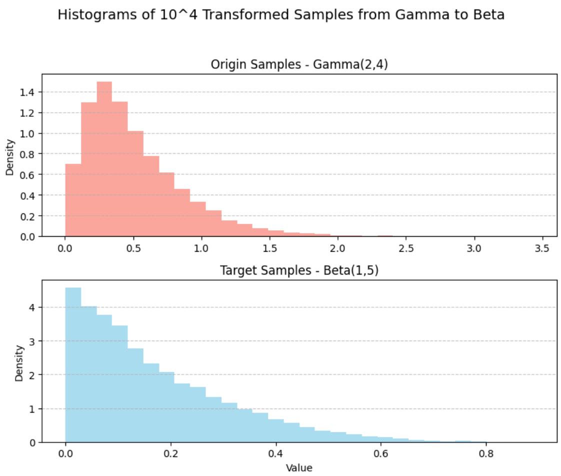
```python
# Parameters for the target distribution
N = 10000
EXP_CDF = lambda x: np.where(x >= 0, 1 - np.exp(-x), 0)
EXP_CDF_INV = lambda v: -np.log(1 - v)  # Inverse CDF for the exponential distribution

# Generate uniform samples from U(0,1)
uni_samples = np.random.uniform(0, 1, N)

# Generate samples from the exponential distribution
exp_samples = EXP_CDF_INV(uni_samples)
# Transform the samples to U(0,1) using the CDF of the exponential distribution
uni_samples = EXP_CDF(exp_samples)

# Plot the histograms of the original and transformed samples
plot_histograms(
    exp_samples,
    uni_samples,
    origin_label="Origin Samples - Exp(1)",
    target_label="Target Samples - U(0,1)",
    main_title="Histograms of 10^4 Transformed Samples from Exponential to U(0,1)",
    bins=30,
)
```



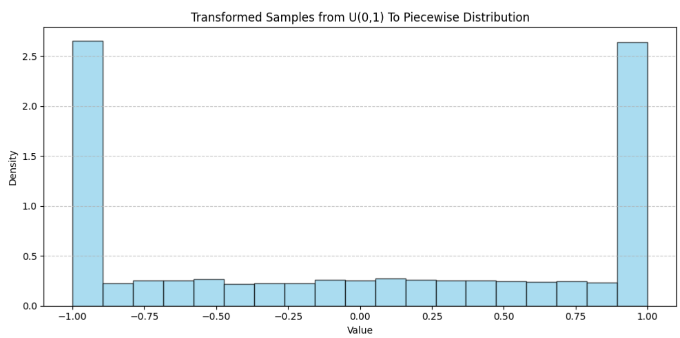

(c) Let $X \sim Gamma(2,4)$, then $f_X(x) = \frac{4^2}{\Gamma(2)} x^{2-1} e^{-4x} = 16xe^{-4x}$, $F_X(x) = 16\int_0^x te^{-4t}dt = -4e^{-4t}(t+\frac{1}{4})\Big]_0^x = 1-4e^{-4x}(x+\frac{1}{4})$.

Since we need a numeric approach for finding $F_X^{-1}(v)$, we would generate the initial Gamma(2,4) sample using a numpy function. Let $Y \sim Beta(1,5)$, then $f_Y(y) = \frac{\Gamma(1+5)}{\Gamma(1)\Gamma(5)} y^{1-1}(1-y)^{5-1} = 5(1-y)^4$, $F_Y(y) = 5\int_0^y (1-t)^4 dt = -(1-t)^5\Big]_0^y = 1-(1-y)^5$, and $F_Y^{-1}(v) = 1-\sqrt[5]{1-v}$.

```python
# Parameters for the target distribution
N = 10000
GAMMA_CDF = lambda x: np.where(x >= 0, 1 - 4 * np.exp(-4 * x) * (x + 0.25), 0)
BETA_CDF_INV = lambda v: 1 - (1 - v) ** (1 / 5)

# Generate samples from the Gamma(2,4) distribution
gamma_samples = np.random.gamma(shape=2, scale=0.25, size=N)

# Transform the samples to U(0,1) using the CDF of the gamma distribution
uni_samples = GAMMA_CDF(gamma_samples)

# Transform the samples to Beta(1, 5)
beta_samples = BETA_CDF_INV(uni_samples)

# Plot the histograms of the original and transformed samples
plot_histograms(
    gamma_samples,
    beta_samples,
    origin_label="Origin Samples - Gamma(2,4)",
    target_label="Target Samples - Beta(1,5)",
    main_title="Histograms of 10^4 Transformed Samples from Gamma to Beta",
    bins=30,
)
```




7. Suppose $X$ is a random variable whose CMF is $F_X(x) = \begin{cases} 0 & x<-1 \\ \frac{x}{4}+\frac{1}{2} & -1\le x<1 \\ 1 & 1\le x \end{cases}$, then $F_X^{-1}(v) = \begin{cases} -1 & 0\le v<1/4 \\ 4v-2 & 1/4\le v<3/4 \\ 1 & 3/4\le v\le 1 \end{cases}$. In order to sample from this piecewise CMF we define it using numpy's piecewise function, sample from a $U(0,1)$ distribution, and apply the inverse piecewise function on the uniform distribution sample array.



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

def plot_histogram(
    samples,
    title="Sample Distribution",
    xlabel="Value",
    ylabel="Density",
    num_bins=20,
    color="skyblue",
    edge_color="black",
    figsize=(10, 5),
):
    """
    Plots a single normalized histogram for a set of samples with distinct borders.
    """
    fig, ax = plt.subplots(1, 1, figsize=figsize)

    # Creates num_bins bins covering the interval [-1, 1]
    bin_edges = np.linspace(-1, 1, num_bins + 1)

    # Plot the histogram with density normalization and edge color
    ax.hist(samples, bins=bin_edges, density=True, alpha=0.7,
            color=color, edgecolor=edge_color) # Added edgecolor
    ax.set_title(title, fontsize=12)
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)
    ax.grid(axis="y", linestyle="--", alpha=0.7)

    # Set x-axis limits slightly wider to see the end bins clearly
    ax.set_xlim(-1.1, 1.1)

    plt.tight_layout()
    plt.show()


# --- Define the piecewise inverse CDF ---
def inverse_cdf(v):
    """
    Calculates the value of the piecewise inverse CDF:
    F_x(v) =
       -1,         if 0 <= v < 1/4
       4v - 2,     if 1/4 <= v < 3/4
       1,          if 3/4 <= v <= 1
    """
    # Ensure v is a numpy array for vectorized operations
    v = np.asarray(v)

    # Define conditions for each piece
    conditions = [
        (v >= 0) & (v < 0.25),     # 0 <= v < 1/4
        (v >= 0.25) & (v < 0.75), # 1/4 <= v < 3/4
        (v >= 0.75) & (v <= 1)    # 3/4 <= v <= 1
    ]

    # Define the corresponding functions/values for each condition
    functions = [
        lambda x: -1.0,           # Result is -1
        lambda x: 4 * x - 2,      # Result is 4v - 2
        lambda x: 1.0             # Result is 1
    ]

    # Use numpy.piecewise to apply the conditions and functions
    return np.piecewise(v, conditions, functions)

# ---- Main ----

# generate uniform samples from u(0,1)
N = 10000
uni_samples = np.random.uniform(0, 1, N)

# generate samples from the target distribution using inverse transform sampling
target_samples = inverse_cdf(uni_samples)

# plot the histogram of the generated samples
plot_histogram(
    target_samples,
    title="Transformed Samples from U(0,1) To Piecewise Distribution",
)
```