Lista 1 → Gabriel Tambara Rabelo

1 - Quesitos Analíticos:

a) Definimos $D = B \cup C$ arbitrariamente, de modo que $P(A \cup B \cup C) = P(A \cup D)$. Para o cenário mais geral, há intersecção entre os conjuntos.

$P(A \cup D) = P(A) + P(D) - P(A \cap D)$, logo, segue que

$$P(A \cup D) = P(A) + P(B \cup C) - P(A \cap [B \cup C])$$

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(B \cap C) - [P(A \cap B) + P(A \cap C) - P(A \cap B \cap C)]$$

portanto:

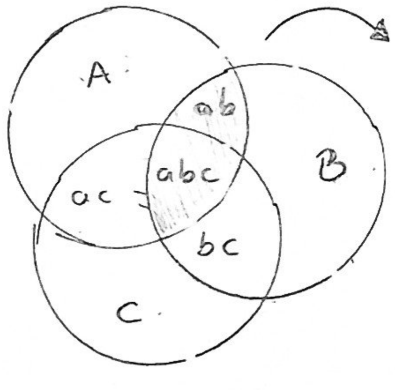
$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$$

b) Para conjuntos disjuntos, não há interseções, e $P(A \cup B \cup C) = P(A) + P(B) + P(C)$ mas quando há intersecções $P(A \cup D) = P(A) + P(D) - P(A \cap D)$, ou seja, para $P(A \cap D) \neq \emptyset$, $P(A) + P(D) > P(A \cup D)$. Analogamente:

$$P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C) = P(A \cup B \cup C),$$

o mesmo vale já que:

$$\frac{A + ab + ac + abc}{T} + \frac{B + ab + bc + abc}{T} + \frac{C + ac + bc + abc}{T} - \left(\frac{ab + abc}{T}\right) - \left(\frac{ac + abc}{T}\right) - \left(\frac{bc + abc}{T}\right) + \frac{abc}{T} = \frac{A + B + C + ab + bc + ac + abc}{T}$$



logo:

$$A + B + C + ab + bc + ac + abc \leq A + B + C + 2ab + 2bc + 2ac + 3abc$$

$$\emptyset \leq ab + bc + ac + 2abc$$

2 - $P\{A\} = \frac{1}{2}$ $\quad P\{B\} = \frac{2}{3}$ $\quad P\{C\} = \frac{3}{4}$

a) $P\{\Omega | B\} - P\{\Omega | A\} = \emptyset$

$P\{\Omega | B\} = \frac{P\{\Omega \cap B\}}{P\{B\}} = \frac{P\{B\}}{P\{B\}} = 1$

b) $P\{A | B\} = \frac{P\{A \cap B\}}{P\{B\}} = \frac{5/12}{2/3} = \frac{5}{8}$

c) $P\{B | A\} = \frac{5/12}{1/2} = \frac{10}{12} = \frac{5}{6}$

d) $P\{\bar{A} \cap B\}$, $\bar{A} = \Omega - A$

$P\{\bar{A}\} = p\{\Omega\} - p\{A\}$

$p\{\Omega \cap B\} - p\{A \cap B\} = p\{B\} - p\{A \cap B\}$

$= 2/3 - 5/12 = 3/12$

3 - $p_x(x) = \begin{cases} 3/4\,(1 - x^2), & -1 \leq x \leq 1 \\ \emptyset, & \text{resto} \end{cases}$

$p\{x > \emptyset\} = \int_0^{\infty} p_x(x)\,dx = \int_0^1 0{,}75\,(1 - x^2)\,dx = 0{,}75\left[x - x^3/3\right]\Big|_0^1 = 0{,}75(1 - 1/3) = 0{,}5$

$p\{x < 0{,}5\} = 0{,}75\left(x - \frac{x^3}{3}\right)\Big|_{-1}^{0,5} = 0{,}75\left(1{,}5 - \left[\frac{(1/2)^3}{3} - \frac{(-1)^3}{3}\right]\right) = \frac{3}{4}\left(\frac{3}{2} - \left(\frac{1}{3\cdot 3} + \frac{3}{8\cdot 3}\right)\right) = \frac{27}{32}$

$p(|x| > 0{,}75) = 1 - p(|x| < 0{,}75) = 1 - 0{,}75(x - x^3/3)\Big|_{-0,75}^{0,75} = 1 - \frac{3}{4}\left(\frac{3}{2} - 2\cdot 0{,}140625\right) = \emptyset{,}086$

4 - $\lim_{x \to -\infty} F(x) = \emptyset$ ✓

$\lim_{x \to \infty} F(x) = 1$ ✓

$\lim_{t \to x \,|\, t > x} F_x(t) = F_x(x)$ ✓

$\emptyset \leq \frac{1}{4} + \frac{c}{2}(x + 2)^2 < 1$

$\downarrow$

$-\frac{1}{8} \leq c < \frac{3}{32}$

para ser monotônica não decrescente: $F(x_1) \leq F(x_2)\ \forall\, x_1 < x_2$

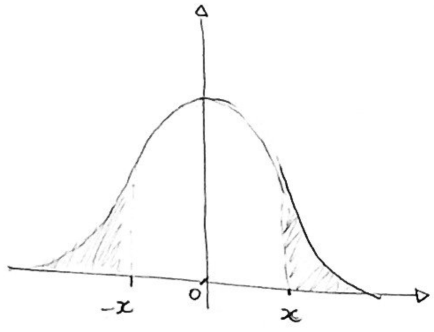
logo, $\frac{dF_x(x)}{dx} = \frac{c}{2}\cdot 2(x+2)\ \forall\ \emptyset \leq x < 2$

$c(x+2) \geq \emptyset \rightarrow$ $c \geq \emptyset$, $x \geq -2 \rightarrow$ razoável para $x \geq \emptyset$

$c < \emptyset$, $x < -2 \rightarrow$ fora da área, logo $c \geq \emptyset$

logo, $\boxed{\emptyset \leq c < 3/32}$

5 - $P\{|X| \leq x\} = 2\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp(-\frac{t^2}{2})dt - 1$



isto é, $P\{|X| \leq x = 2P\{X \leq x\} - 1$

vale avaliar a simetria da função:

$P\{X \leq -x\} = 1 - P\{X \leq x\}$

$$\begin{aligned} P\{|X| \leq x\} &= P\{X \leq x\} - P\{X \leq -x\} \\ &= P\{X \leq x\} - [1 - P\{X \leq x\}] \\ &= P\{X \leq x\} \cdot 2 - 1 \end{aligned}$$

Quesitos Numéricos

1 - a) A primeira transforação possui como média os valores nulos e matriz de covariância, igual à identidade, como esperado conforme transformação de U para distribuição χ^2_n. Percebe-se o caráter de normalização e centralização dos dados.

b) Para a segunda transformação, sabendo que $\mathcal{E}\{(X - \mathcal{E}\{X\})^2\} = \sigma^2$, percebe-se que sua média retorna os valores de variância, corroborando a teoria.

2 - $p_y(y) = p_x(f^{-1}(y)) \left| \frac{dy}{dx} \right|^{-1}_{x=f^{-1}(y)}$ $\quad x \sim U(0,1) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{resto} \end{cases}$

$y = -\ln(x) \rightarrow x = e^{-y} \; \forall y \geq 0$ $\quad \mathcal{E}\{y^2\} = \int_0^\infty y^2 e^{-y} dy = -y^2 e^{-y} \Big|_0^\infty + 2 \cancel{\int_0^\infty e^{-y} dy}^{1} = 2$

$p_y = 1 \cdot |(-e^{-y})| = e^{-y}$

$\mathcal{E}\{y\} = \int_0^\infty y\, p_y\, dy = \int_0^\infty y e^{-y} dy = -ye^{-y} + \int_0^\infty e^{-y} dy = \lim_{t \to \infty} \frac{-1}{e^{t}} + \frac{1}{e^0} = 1$

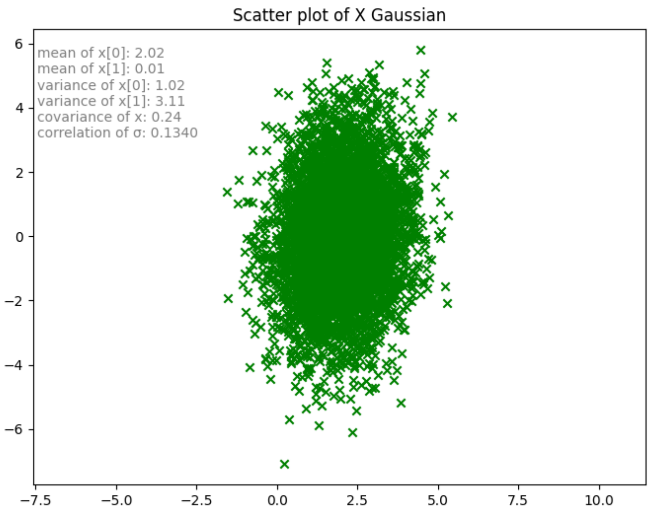
$var\{y\} = \mathcal{E}\{y^2\} - \mathcal{E}\{y\}^2 = 2 - 1 = 1$

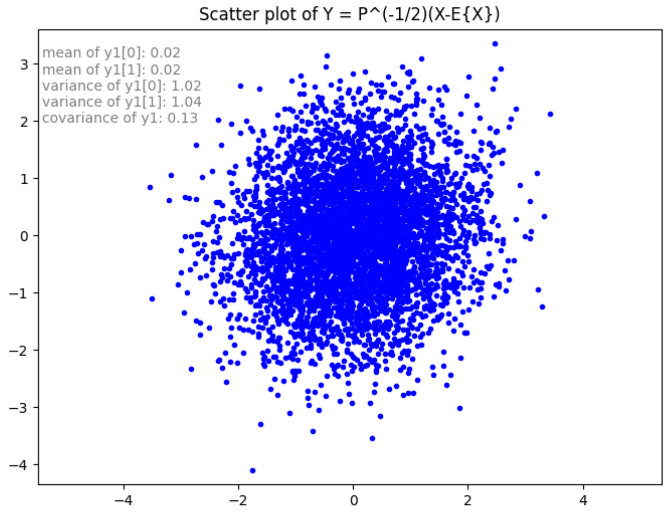
3 - Como a área do domínio é um quadrado de -1 a 1 em ambos eixos, igual a 4, para estimar π, numericamente igual à área de um círculo de raio unitário, pode-se dividir as amostras dentro do círculo (o preenchendo) pelas amostras totais, basta multiplicar por 4.

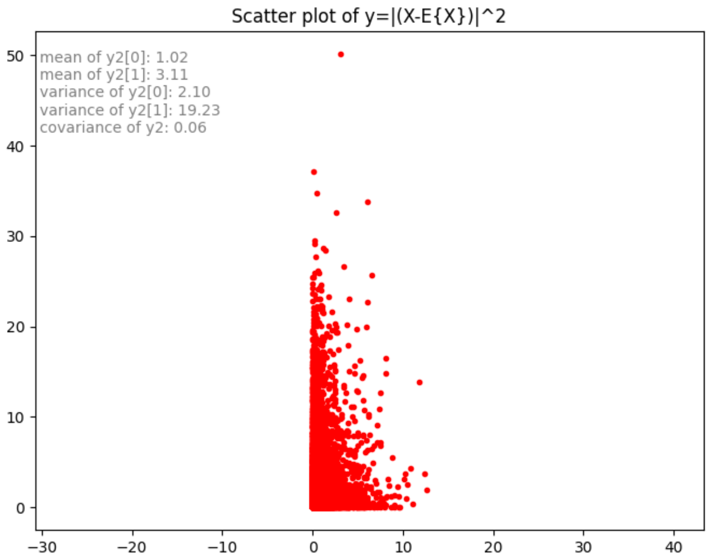
Percebe-se que ambas as curvas seguem o mesmo perfil já que a média (um bom estimador de convergência para este caso) do erro se aproxima de $\emptyset$ em torno de 10.000 dados simulados, 5.000 pares de pontos de amostra uniforme.

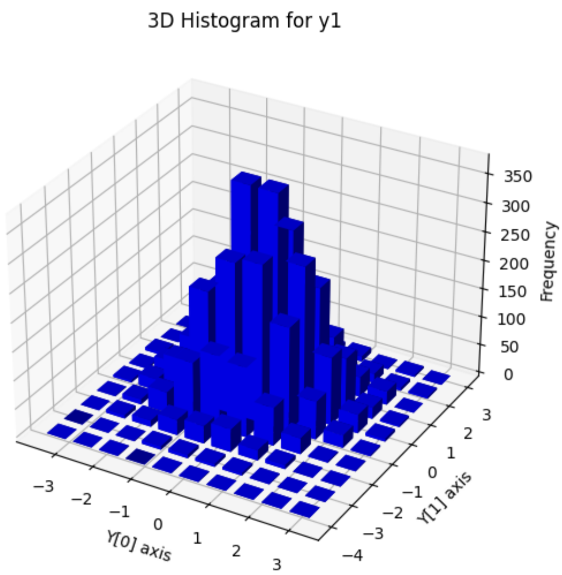
```python
import numpy as np
from matplotlib import pyplot as plt
from scipy.linalg import sqrtm, polar
import random

mean = np.array([2, 0])
covariance = np.array([[1, 0.2], [0.2, 3]])

x = np.random.default_rng().multivariate_normal(mean, covariance, 5000, method='cholesky')

# 1-a)

inferior_tri = np.linalg.cholesky(covariance)
superior_tri = np.transpose(inferior_tri)
inv_sup_tri = np.linalg.inv(superior_tri)
u, p = polar(inv_sup_tri, 'left')
sqrtm_covariance = inv_sup_tri * np.transpose(u)
x_centered = x - mean
y1 = np.matmul(x_centered, sqrtm_covariance)

# 1-b)

def covar(x1, x2):
    return np.mean(x1 * x2) - np.mean(x1) * np.mean(x2)

def correlation(x1, x2):
    return covar(x1, x2)/(np.sqrt(covar(x1, x1)) * np.sqrt(covar(x2, x2)))

y2 = x_centered**2

#plotting the distributions
plt.figure(figsize=(8, 6))
plt.scatter(x[:, 0], x[:, 1], color='green', marker='x')
plt.title('Scatter plot of X Gaussian')
plt.suptitle(
          "mean of x[0]: " + "{:.2f}".format(np.mean(x[:,0])) + "\n" +
          "mean of x[1]: " + "{:.2f}".format(np.mean(x[:,1])) + "\n" +
          "variance of x[0]: " + "{:.2f}".format(np.var(x[:,0])) + "\n" +
          "variance of x[1]: " + "{:.2f}".format(np.var(x[:,1])) + '\n' +
          "covariance of x: " + "{:.2f}".format((covar(x[:,0], x[:,1]))) + '\n' +
          "correlation of  : " + "{:.4f}".format((correlation(x[:,0], x[:,1])))
          , x=0.13 , y=0.85, fontsize=10, color='gray', ha = 'left')
plt.axis('equal')

plt.figure(figsize=(8, 6))
plt.scatter(y1[:, 0], y1[:, 1], color='blue', marker='.')
plt.title('Scatter plot of Y = P^(-1/2)(X-E{X})')
plt.suptitle(
          "mean of y1[0]: " + "{:.2f}".format(np.mean(y1[:,0])) + "\n" +
          "mean of y1[1]: " + "{:.2f}".format(np.mean(y1[:,0])) + "\n" +
          "variance of y1[0]: " + "{:.2f}".format(np.var(y1[:,0])) + "\n" +
          "variance of y1[1]: " + "{:.2f}".format(np.var(y1[:,1])) + '\n' +
          "covariance of y1: " + "{:.2f}".format(covar(y1[:,0], y1[:,1]))
          , x=0.13 , y=0.85, fontsize=10, color='gray', ha = 'left')
plt.axis('equal')

plt.figure(figsize=(8, 6))
plt.scatter(y2[:, 0], y2[:, 1], color='red', marker='.')
plt.title('Scatter plot of y=|(X-E{X})|^2')
plt.suptitle(
          "mean of y2[0]: " + "{:.2f}".format(np.mean(y2[:,0])) + "\n" +
```

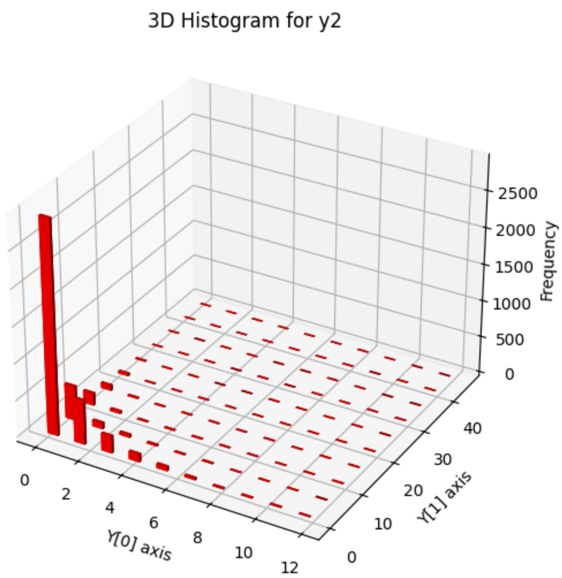
```
         "mean of y2[1]: " + "{:.2f}".format((np.mean(y2[:,1]))) + "\n" +
         "variance of y2[0]: " + "{:.2f}".format(np.var(y2[:,0])) + "\n" +
         "variance of y2[1]: " + "{:.2f}".format(np.var(y2[:,1])) + '\n' +
         "covariance of y2: " + "{:.2f}".format(covar(y2[:,0], y2[:,1]))
         , x=0.13  , y=0.85, fontsize=10, color='gray', ha = 'left')
plt.axis('equal')

#plotting the PDFs of y1 and y2
hist_y1, xedges_y1, yedges_y1 = np.histogram2d(y1[:,0], y1[:,1])
hist_y2, xedges_y2, yedges_y2 = np.histogram2d(y2[:,0], y2[:,1])

xpos_y1, ypos_y1 = np.meshgrid(xedges_y1[:-1], yedges_y1[:-1])
xpos_y1 = xpos_y1.flatten('F')
ypos_y1 = ypos_y1.flatten('F')
zpos_y1 = np.zeros_like(xpos_y1)

xpos_y2, ypos_y2 = np.meshgrid(xedges_y2[:-1], yedges_y2[:-1])
xpos_y2 = xpos_y2.flatten('F')
ypos_y2 = ypos_y2.flatten('F')
zpos_y2 = np.zeros_like(xpos_y2)

dx_y1 = dy_y1 = 0.5 * np.ones_like(zpos_y1)
dz_y1 = hist_y1.flatten()
dx_y2 = dy_y2 = 0.5 * np.ones_like(zpos_y2)
dz_y2 = hist_y2.flatten()

fig = plt.figure(figsize=(14, 7))

ax1 = fig.add_subplot(121, projection='3d')
ax1.bar3d(xpos_y1, ypos_y1, zpos_y1, dx_y1, dy_y1, dz_y1, color='b')
ax1.set_title('3D Histogram for y1')
ax1.set_xlabel('Y[0] axis')
ax1.set_ylabel('Y[1] axis')
ax1.set_zlabel('Frequency')

ax2 = fig.add_subplot(122, projection='3d')
ax2.bar3d(xpos_y2, ypos_y2, zpos_y2, dx_y2, dy_y2, dz_y2, color='r')
ax2.set_title('3D Histogram for y2')
ax2.set_xlabel('Y[0] axis')
ax2.set_ylabel('Y[1] axis')
ax2.set_zlabel('Frequency')

plt.show()
```


Scatter plot of X Gaussian
mean of x[0]: 2.02
mean of x[1]: 0.01
variance of x[0]: 1.02
variance of x[1]: 3.11
covariance of x: 0.24
correlation of σ: 0.1340

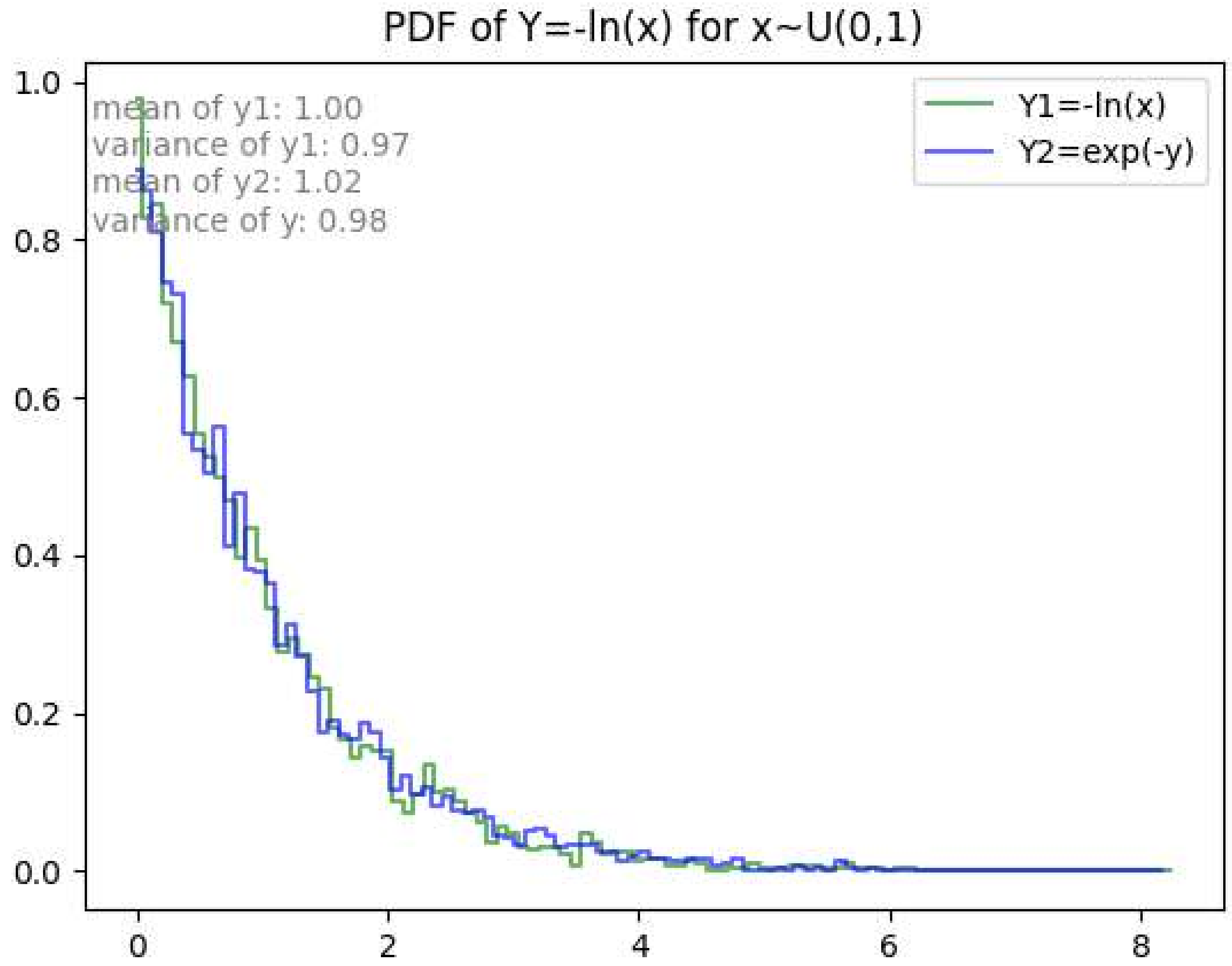

Scatter plot of Y = P^(-1/2)(X-E{X})
mean of y1[0]: 0.02
mean of y1[1]: 0.02
variance of y1[0]: 1.02
variance of y1[1]: 1.04
covariance of y1: 0.13


Scatter plot of y=|(X-E{X})|^2
mean of y2[0]: 1.02
mean of y2[1]: 3.11
variance of y2[0]: 2.10
variance of y2[1]: 19.23
covariance of y2: 0.06


3D Histogram for y1
Frequency
Y[0] axis
Y[1] axis

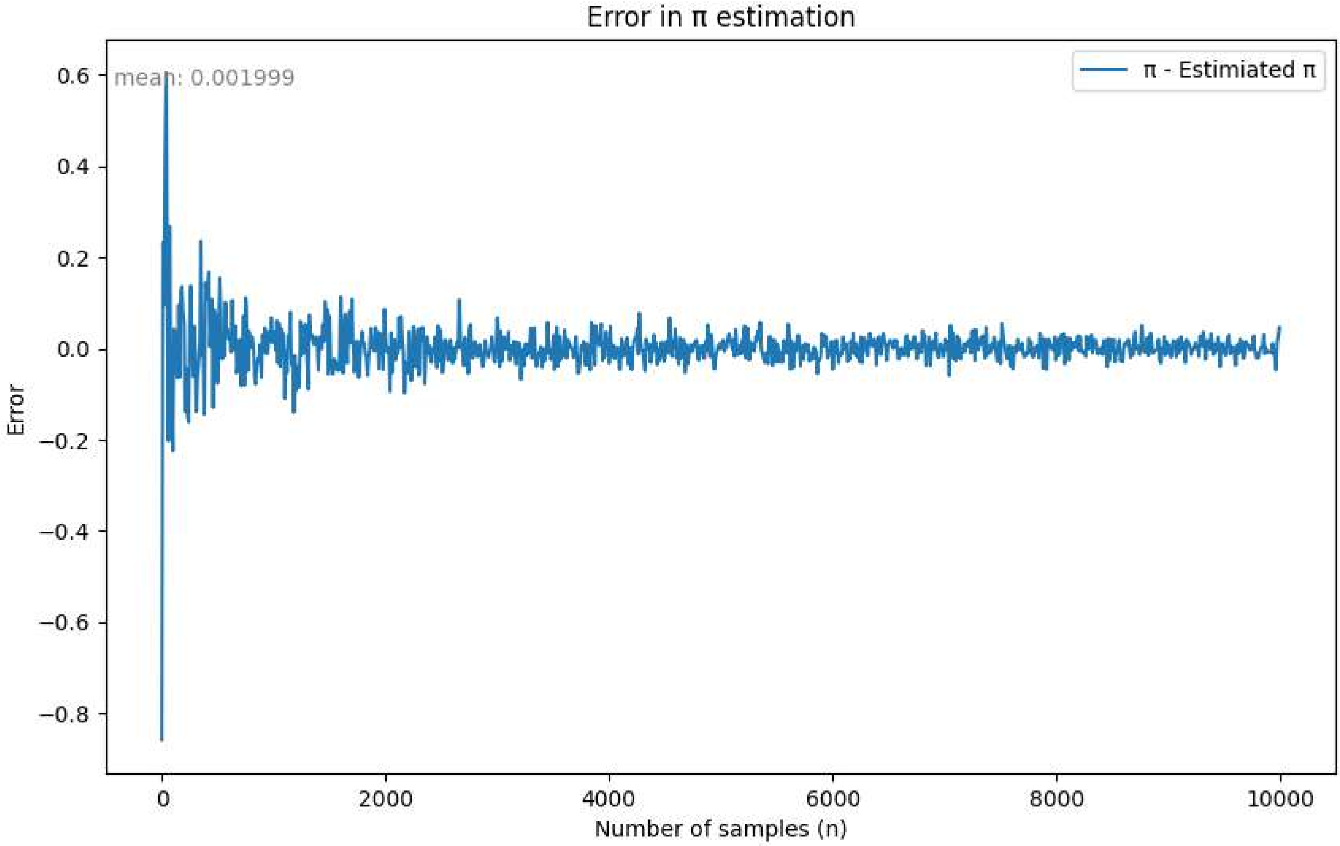

3D Histogram for y2
Frequency
Y[0] axis
Y[1] axis

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

x = np.random.uniform(0, 1, 5000)
y = - np.log(x)

counts, bin_edges = np.histogram(y, bins=100, density=True)
plt.step(bin_edges[:-1], counts, where='mid',alpha=0.6, color='green', label='Y1=-ln(x)')

Y = np.random.exponential(1, size=5000)

def pdf_y(y):
    return np.exp(-y) if y >= 0 else 0

pdf_values = np.vectorize(pdf_y)(Y)

plt.title("PDF of Y=-ln(x) for x~U(0,1)")
counts2, bin_edges2 = np.histogram(Y, bins=100, density=True)
plt.step(bin_edges2[:-1], counts2, where='mid',alpha=0.6, color='blue', label='Y2=exp(-y)')
plt.suptitle(
          "mean of y1: " + "{:.2f}".format(np.mean(y)) + "\n" +
          "variance of y1: " + "{:.2f}".format(np.var(y)) + "\n" +
          "mean of y2: " + "{:.2f}".format(np.mean(Y)) + "\n" +
          "variance of y: " + "{:.2f}".format(np.var(Y))
          , x=0.13 , y=0.85, fontsize=10, color='gray', ha = 'left')

plt.legend()
plt.show()
```



PDF of Y=-ln(x) for x~U(0,1)
mean of y1: 1.00
variance of y1: 0.97
mean of y2: 1.02
variance of y: 0.98
Y1=-ln(x)
Y2=exp(-y)
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8

```python
import numpy as np
from matplotlib import pyplot as plt
from scipy.linalg import norm

error = []

for n in range(1, 10000, 10):
    u = np.random.uniform(-1, 1, 2 * n)
    u_pairs = u.reshape(-1, 2)

    inside_counts = 0

    for pair in u_pairs:
        if((pair[0]**2 + pair[1]**2) <= 1):
            inside_counts = inside_counts + 1

    probability_inside_circle = inside_counts/n
    estimated_pi = probability_inside_circle * 4
    error.append((np.pi - estimated_pi))

plt.figure(figsize=(10, 6))
plt.plot(range(1, 10000, 10), error, label='pi- Estimiated pi')
plt.xlabel('Number of samples (n)')
plt.ylabel('Error')
plt.title('Error in pi estimation')
plt.suptitle(
        "mean: " + "{:.6f}".format(np.mean(error))
        , x=0.13 , y=0.85, fontsize=10, color='gray', ha = 'left')
plt.legend()
plt.show()
```


Error in π estimation
mean: 0.001999
π - Estimiated π
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
Error
0
2000
4000
6000
8000
10000
Number of samples (n)