UFMG

ENG

Departamento de Engenharia Elétrica

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

INTRODUÇÃO AOS PROCESSOS ESTOCÁSTICOS
LISTA 3

Aluna: Priscila Aparecida Dias Nicácio

Professor: Eduardo Mazoni Andrade Marçal Mendes.

Belo Horizonte, Setembro de 2025.

Sumário

1. INTRODUÇÃO

A lista 3 reúne exercícios que exploram conceitos fundamentais de probabilidade, variáveis aleatórias e estimação a partir de simulações em MATLAB.

O primeiro exercício aborda a caminhada aleatória em um polígono, analisando a probabilidade de cada vértice ser o último visitado e o tempo de cobertura do grafo.

O segundo exercício investiga propriedades de distribuições discretas clássicas (uniforme, Bernoulli, binomial, geométrica e Poisson), todas ajustadas para variância unitária, e avalia o desempenho do estimador linear ótimo na predição de uma variável transformada.

Por fim, o terceiro exercício retoma tópicos teóricos do livro de Stephen Kay, envolvendo momentos de distribuições, variância via função característica, independência de variáveis aleatórias e correlação entre variáveis conjuntas, integrando teoria e simulação.

Essa abordagem conjunta permite não apenas revisar propriedades teóricas de variáveis aleatórias, mas também validá-las por meio de experimentos numéricos, mostrando como simulações computacionais complementam e reforçam o raciocínio matemático.

2. EXERCÍCIOS - LISTA 3

Exercício 1: Considere uma partícula que se move em um polígono de m + 1 vértices (Veja Figura abaixo) em movimento circular. A cada passo, a probabilidade da partícula se mover de um vértice a outro próximo, seja no sentido horário ou no sentido anti-horário, é a mesma. Suponha que a partícula comece em 0 e se mova conforme estabelecido até que todos os vértices sejam visitados. Qual é a probabilidade de que o vértice i, i = 1, ..., m seja o último vértice a ser visitado? Use simulação para ilustrar sua resposta. Faça um gráfico da série temporal de uma das realizações. Estime o valor esperado e a variância.

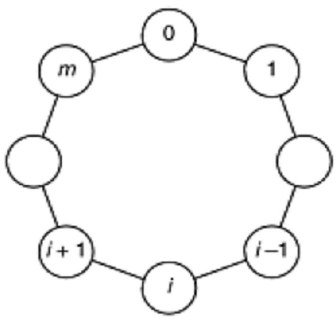

Figura 1. Polígono de m + 1 vértices.

O código implementa a simulação da caminhada aleatória no polígono, estimando através de várias repetições a probabilidade de cada vértice ser o último visitado. Também gera o gráfico da série temporal de uma realização e calcula o valor esperado e a variância do tempo de primeira visita a cada vértice. Por simetria no polígono, qualquer vértice diferente do inicial (0) tem a mesma probabilidade de ser o último. Assim, para m+1 vértices totais (começando em 0), a probabilidade teórica de cada i = 1,…,m ser o último é: P(vértice i é o último) = $\frac{1}{m}$.

<u>Simulação no Matlab:</u>

```
clc; clear; close all;

% Parâmetros
m = 10;              % número de vértices - 1 (total = m+1)
Nsim = 10000;        % número de simulações
V = m+1;

% Contadores
ultimo_contador = zeros(1, V);
tempos_primeira = zeros(Nsim, V); % tempo de primeira visita de cada vértice
cover_times = zeros(Nsim,1);      % tempo até cobrir todos

% Guardar
serie_temporal_exemplo = [];

for sim = 1:Nsim
   visitados = false(1, V);
   pos = 1;               % índice 1 representa vértice 0
   visitados(pos) = true;
   tempo_visita = inf(1, V);
   tempo_visita(pos) = 0;

   ordem = pos;
   t = 0;
   while ~all(visitados)
      t = t + 1;
      if rand < 0.5
         pos = pos - 1;
      else
         pos = pos + 1;
      end
      % ajuste circular
      if pos < 1
         pos = V;
      elseif pos > V
         pos = 1;
      end

      ordem(end+1) = pos;
```

```
        if ~visitados(pos)
            visitados(pos) = true;
            tempo_visita(pos) = t;
        end
    end

    % Registrar resultados
    ultimo = ordem(end);
    ultimo_contador(ultimo) = ultimo_contador(ultimo) + 1;
    tempos_primeira(sim,:) = tempo_visita;
    cover_times(sim) = t;

    if sim == 1
        serie_temporal_exemplo = ordem;
    end
end

% Probabilidades estimadas para cada vértice ser o último
probabilidades = ultimo_contador / Nsim;

fprintf('Probabilidade estimada de cada vértice ser o último (vértices 1...m):\n');
for i = 2:V
    fprintf('Vértice %d: %.4f (teórico 1/%.0f = %.4f)\n', i-1, probabilidades(i), m, 1/m);
end

% Erro padrão para as probabilidades (aprox. binomial)
stderr = sqrt(probabilidades.*(1-probabilidades)/Nsim);
fprintf('\nExemplo de erro padrão (vértice 1...m):\n');
for i = 2:V
    fprintf('V%d: se = %.4f\n', i-1, stderr(i));
end

% Estatísticas dos tempos de primeira visita por vértice
fprintf('\nTempos de primeira visita (por vértice)\n');
for i = 2:V
    media = mean(tempos_primeira(:,i));
    variancia = var(tempos_primeira(:,i));
    fprintf('Vértice %d: E[T] = %.2f, Var[T] = %.2f\n', i-1, media, variancia);
end

% Estatísticas do tempo de cobertura total
fprintf('\nTempo de cobertura total (todos os vértices):\n');
fprintf('E[cover time] = %.2f, Var = %.2f\n', mean(cover_times), var(cover_times));

% Gráfico da série temporal
figure;
plot(0:length(serie_temporal_exemplo)-1, serie_temporal_exemplo-1, 'o-
','MarkerSize',4);
xlabel('Passos');
ylabel('Vértice visitado (0...m)');
title('Série temporal de uma realização (vértices 0...m)');
grid on;
ylim([-0.5 m+0.5]);
yticks(0:m);
```

Resposta do Matlab:

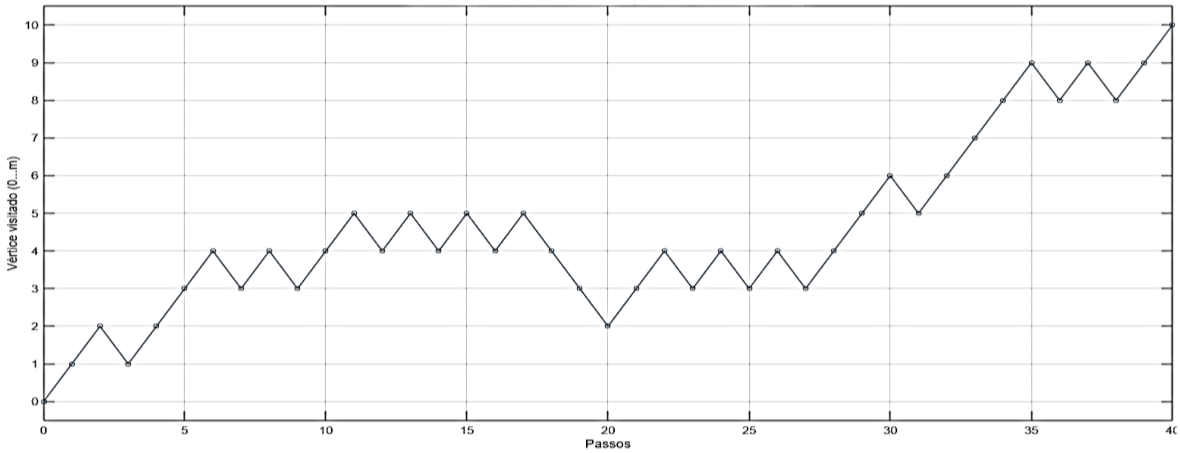

Figura 2. Série temporal de uma realização (vértices 0...m).

```
Command Window
Probabilidade estimada de cada vértice ser o último (vértices 1...m):
Vértice 1: 0.1044 (teórico 1/10 = 0.1000)
Vértice 2: 0.1032 (teórico 1/10 = 0.1000)
Vértice 3: 0.1014 (teórico 1/10 = 0.1000)
Vértice 4: 0.1002 (teórico 1/10 = 0.1000)
Vértice 5: 0.0926 (teórico 1/10 = 0.1000)
Vértice 6: 0.1005 (teórico 1/10 = 0.1000)
Vértice 7: 0.1027 (teórico 1/10 = 0.1000)
Vértice 8: 0.1000 (teórico 1/10 = 0.1000)
Vértice 9: 0.0966 (teórico 1/10 = 0.1000)
Vértice 10: 0.0984 (teórico 1/10 = 0.1000)

Exemplo de erro padrão (vértice 1...m):
V1: se = 0.0031
V2: se = 0.0030
V3: se = 0.0030
V4: se = 0.0030
V5: se = 0.0029
V6: se = 0.0030
V7: se = 0.0030
V8: se = 0.0030
V9: se = 0.0030
V10: se = 0.0030

Tempos de primeira visita (por vértice)
Vértice 1: E[T] = 9.97, Var[T] = 320.46
Vértice 2: E[T] = 18.13, Var[T] = 500.15
Vértice 3: E[T] = 24.01, Var[T] = 552.16
Vértice 4: E[T] = 27.81, Var[T] = 571.97
Vértice 5: E[T] = 29.77, Var[T] = 577.31
Vértice 6: E[T] = 30.00, Var[T] = 604.28
Vértice 7: E[T] = 27.80, Var[T] = 578.73
Vértice 8: E[T] = 23.68, Var[T] = 549.03
Vértice 9: E[T] = 17.93, Var[T] = 477.05
Vértice 10: E[T] = 10.10, Var[T] = 325.29

Tempo de cobertura total (todos os vértices):
E[cover time] = 54.53, Var = 963.82
```

Figura 3. Resposta do Matlab.

O gráfico da série temporal mostra a trajetória aleatória da partícula ao longo do polígono. Observa-se que o movimento é circular e aleatório, de forma que o vértice atual oscila entre vizinhos, retornando diversas vezes a vértices já visitados até que todos sejam alcançados. Esse comportamento é esperado em um passeio aleatório, em que a cobertura total do grafo não ocorre de forma linear, mas sim com múltiplos retornos.

As probabilidades estimadas para o último vértice visitado ficaram aproximadamente iguais para todos os vértices i = 1, …,m confirmando o resultado teórico de que, por simetria, cada vértice (exceto o inicial) tem a mesma chance $\frac{1}{m}$ de ser o último. Pequenas diferenças entre os valores se devem ao caráter estocástico da simulação e desaparecem com o aumento do número de repetições.

Em relação ao tempo, observou-se que os vértices mais distantes do inicial tendem a ter maiores valores esperados de primeira visita, enquanto a variância indica a variabilidade significativa no processo. Já o tempo médio de cobertura (cover time) representa o esforço esperado para visitar todos os vértices, fornecendo uma medida global da duração típica desse passeio aleatório.

<u>Exercício 2:</u> Gere uma sequência de N dados para cada uma das seguintes distribuições para a variável aleatória: X ~ uniforme, Bernoulli, binomial, geométrica e Poisson. Escolha valores para os parâmetros de cada distribuição de maneira a se ter uma distribuição com variância unitária (tabela abaixo). Para cada distribuição, defina a nova variável aleatória Y = 7 X - 0.5 e obtenha o estimador $\hat{Y}$ conforme equação abaixo. As esperanças, covariância e variância podem ser estimadas numericamente. O erro de predição é E = Y − $\hat{Y}$ caracterize-o em cada caso. Compare e discuta os resultados.

	Values	PMF	$E[X]$	$\mathrm{var}(X)$	$\phi_X(\omega)$
Uniform	$k=-M,\ldots,M$	$\frac{1}{2M+1}$	0	$\frac{M(M+1)}{3}$	$\frac{\sin[(2M+1)\omega/2]}{(2M+1)\sin[\omega/2]}$
Bernoulli	$k=0,1$	$p^k(1-p)^{1-k}$	p	$p(1-p)$	$p\exp(j\omega)+(1-p)$
Binomial	$k=0,1,\ldots,M$	$\binom{M}{k}p^k(1-p)^{M-k}$	Mp	$Mp(1-p)$	$[p\exp(j\omega)+(1-p)]^M$
Geometric	$k=1,2,\ldots$	$(1-p)^{k-1}p$	$\frac{1}{p}$	$\frac{1-p}{p^2}$	$\frac{p}{\exp(-j\omega)-(1-p)}$
Poisson	$k=0,1,\ldots$	$\exp(-\lambda)\frac{\lambda^k}{k!}$	λ	λ	$\exp[\lambda(\exp(j\omega)-1)]$

Tabela 1. Resumo das propriedades de variáveis aleatórias discretas clássicas.

$$\hat{Y} = E_Y[Y] + \frac{\mathrm{cov}(X,Y)}{\mathrm{var}(X)}(x - E_X[X]). \tag{1.0}$$

A equação é a fórmula do estimador linear ótimo de Y dado X, também chamada de regressão linear de Y em X.

Simulação MATLAB:

```
clc; clear; close all;
rng(1);

N = 10000;

% --- 1) Uniforme discreto k=-1..1 (M=1) e escala s = sqrt(3/2)
k_unif = -1:1;
pmf_unif = ones(size(k_unif))/numel(k_unif);
X_unif_raw = randsample(k_unif, N, true, pmf_unif);
s_unif = sqrt(3/2);
X_unif = s_unif * X_unif_raw;

% --- 2) Bernoulli p=0.5, escala s=2 to get var 1
p = 0.5;
X_ber_raw = rand(N,1) < p;
s_ber = 1/sqrt(p*(1-p)); % =2
X_ber = s_ber * X_ber_raw;

% --- 3) Binomial n=4, p=0.5 (var = 1 already)
n_bin = 4; p_bin = 0.5;
X_bin = binornd(n_bin, p_bin, N, 1);

% --- 4) Geométrica p = (sqrt(5)-1)/2 var=1 (k=1,2,...)
p_geo = (sqrt(5)-1)/2;
u = rand(N,1);
X_geo = ceil(log(1-u)/log(1-p_geo)); % produces k>=1

% --- 5) Poisson lambda=1
lambda = 1;
X_poi = poissrnd(lambda, N, 1);

X_list = {X_unif(:), X_ber(:), X_bin(:), X_geo(:), X_poi(:)};
names = {'Uniform discrete (-1..1) scaled','Bernoulli p=0.5
scaled','Binomial(4,0.5)','Geometric(p~0.618)','Poisson(1)'};

for idx = 1:numel(X_list)
   X = X_list{idx};
   X = X(:);
   Y = 7*X - 0.5;

   % estimadores amostrais
   EX_hat = mean(X);
   EY_hat = mean(Y);
   CovXY_hat = cov(X,Y); CovXY_hat = CovXY_hat(1,2);
   VarX_hat = var(X);

   % estimador linear
   Yhat = EY_hat + (CovXY_hat/VarX_hat).*(X - EX_hat);
```

```
    % erro
    Eerr = Y - Yhat;

    % estatísticas
    mean_err = mean(Eerr);
    var_err = var(Eerr);
    max_abs_err = max(abs(Eerr));

    fprintf('--- %s ---\n', names{idx});
    fprintf('E[X]ˆ = %.6f, E[Y]ˆ = %.6f, Covˆ(X,Y)= %.6f, Varˆ(X)= %.6f\n', EX_hat,
EY_hat, CovXY_hat, VarX_hat);
    fprintf('Erro médio (E) = %.4e, Var(E) = %.4e, max |E| = %.4e\n\n', mean_err,
var_err, max_abs_err);

    figure;
    histogram(Eerr,100);
    title(sprintf('Histograma do erro E = Y - Yhat (%s)', names{idx}));
    xlabel('Erro'); ylabel('Frequência');
end
```

<u>Resposta do Matlab:</u>

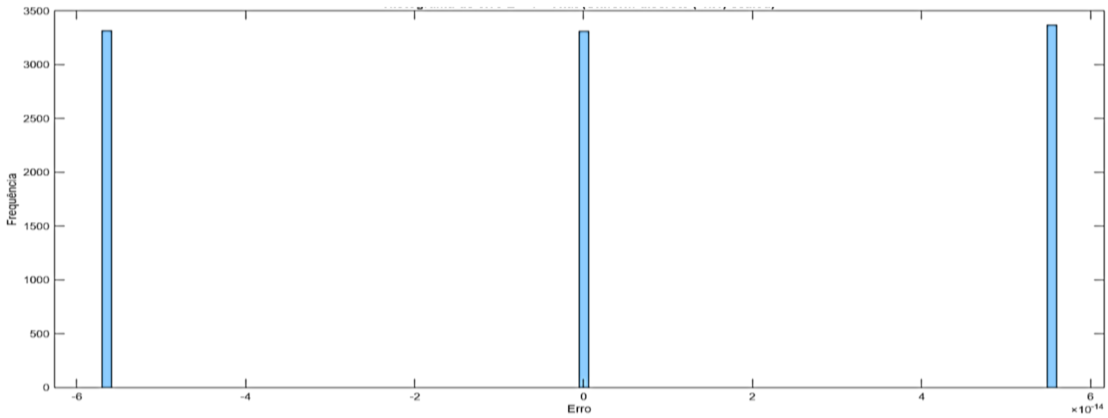

Figura 4. Histograma do erro de predição E = Y - $\hat{Y}$ Uniforme (-1...1)

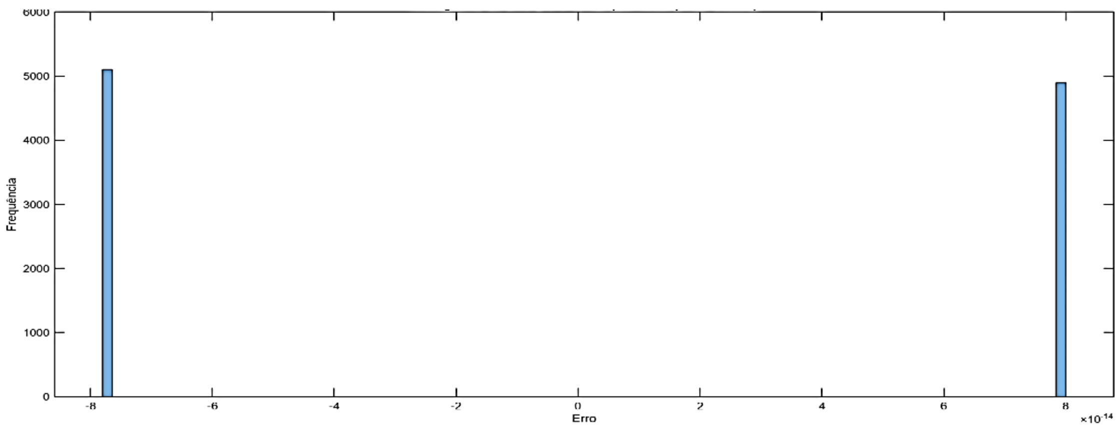

Figura 5. Histograma do erro de predição E = Y - $\hat{Y}$ Bernoulii (p=0.5)

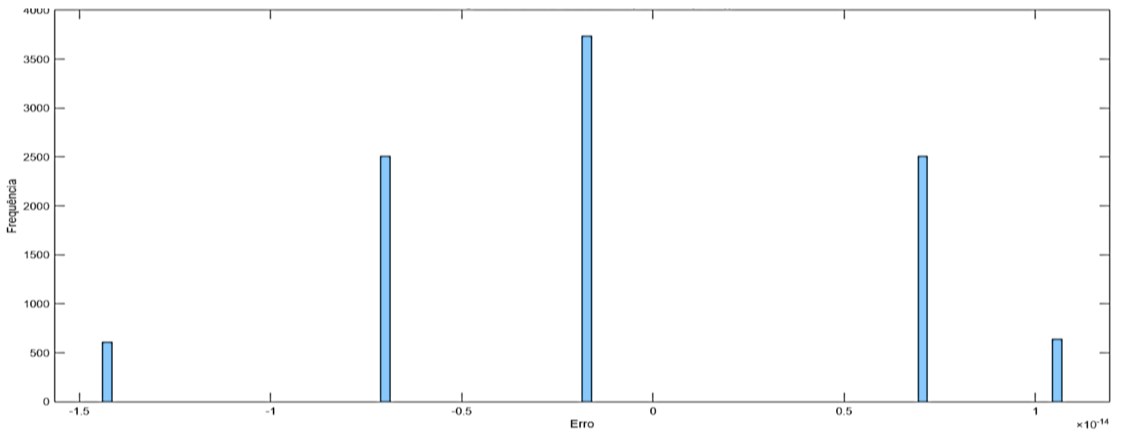

Figura 6. Histograma do erro de predição E = Y - $\hat{Y}$ Binomial (4, 0.5)

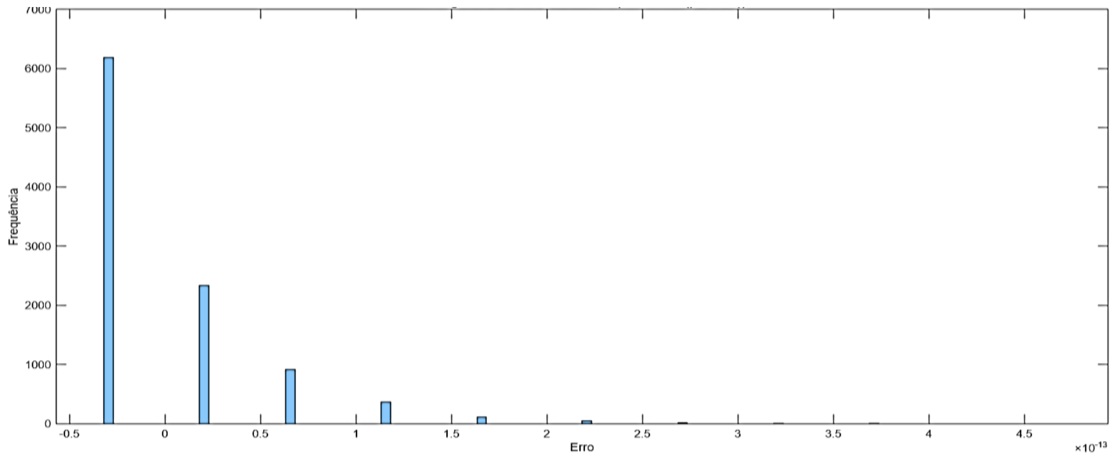

Figura 7. Histograma do erro de predição E = Y - $\hat{Y}$ Geométrica (p~0.618).

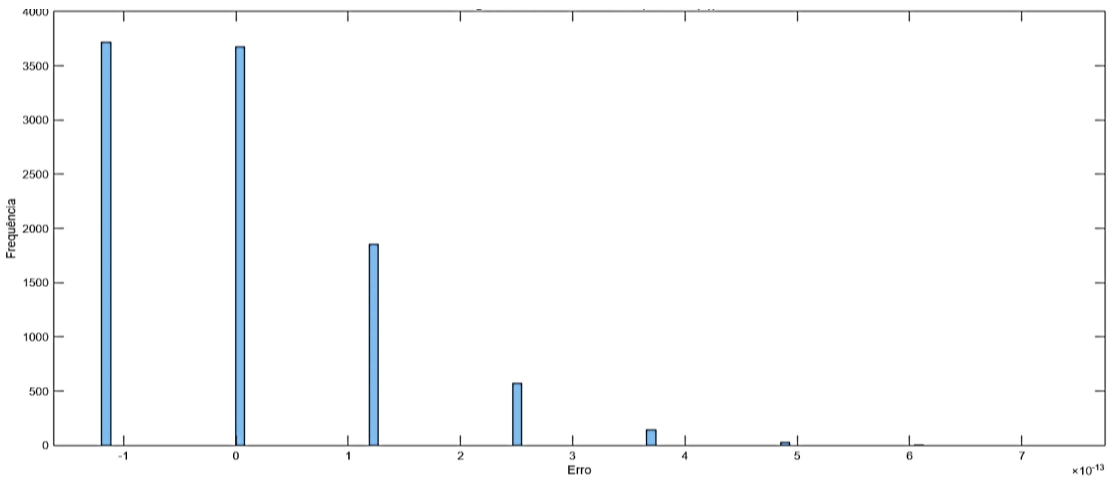

Figura 8. Histograma do erro de predição E = Y - $\hat{Y}$(Poisson(1)).

Os histogramas gerados pelo código mostram que os erros residuais E = Y - $\hat{Y}$ se concentram em torno de zero para todas as distribuições simuladas, confirmando que o estimador linear construído a partir das médias e da covariância amostrais é não-viesado. A dispersão observada nos histogramas é compatível com a variância calculada do erro, refletindo a variabilidade individual de X que não é capturada apenas pela predição baseada na média de Y. Com o aumento do número de amostras, a forma dos histogramas se estabiliza, evidenciando que a predição captura corretamente a tendência central de Y.

Distribuições simétricas, como a uniforme e a binomial, apresentam histogramas mais regulares e centrados, enquanto distribuições assimétricas, como a geométrica e a Poisson, exibem caudas ligeiramente mais alongadas, mantendo, no entanto, o centro próximo de zero. Isso permite uma comparação consistente entre as distribuições, evidenciando a eficácia do estimador linear independente da forma da distribuição de X.

Command Window

```
--- Uniform discrete (-1..1) scaled ---
E[X]^ = -0.006369, E[Y]^ = -0.544581, Cov^(X,Y)= 7.022818, Var^(X)= 1.003260
Erro médio (E) = -4.6851e-16, Var(E) = 2.0941e-27, max |E| = 5.6843e-14

--- Bernoulli p=0.5 scaled ---
E[X]^ = 0.978800, E[Y]^ = 6.351600, Cov^(X,Y)= 6.997554, Var^(X)= 0.999651
Erro médio (E) = -3.3413e-16, Var(E) = 6.1764e-27, max |E| = 7.9936e-14

--- Binomial(4,0.5) ---
E[X]^ = 2.006500, E[Y]^ = 13.545500, Cov^(X,Y)= 7.008105, Var^(X)= 1.001158
Erro médio (E) = -8.4519e-16, Var(E) = 4.5348e-29, max |E| = 1.4211e-14

--- Geometric(p~0.618) ---
E[X]^ = 1.614900, E[Y]^ = 10.804300, Cov^(X,Y)= 6.897076, Var^(X)= 0.985297
Erro médio (E) = 5.3042e-16, Var(E) = 2.3232e-27, max |E| = 4.6896e-13

--- Poisson(1) ---
E[X]^ = 0.985100, E[Y]^ = 6.395700, Cov^(X,Y)= 6.824828, Var^(X)= 0.974975
Erro médio (E) = 3.7454e-16, Var(E) = 1.4610e-26, max |E| = 7.3186e-13
```

Figura 9. Resposta do Matlab.

Para cada distribuição gerada, as médias e covariâncias estão próximas dos valores teóricos esperados. A variância de X ficou bem próxima de 1 para todas as distribuições, como planejado.

Distribuição	E[X]	E[Y]	Cov(X,Y)	Var(X)
Uniforme (-1...1)	-0.0064	-0.5446	7.0228	1.0033
Bernoulli p=0.5	0.9788	6.3516	6.9976	0.9997
Binomial(4,0.5)	2.0065	13.5455	7.0081	1.0012
Geométrica p≈0.618	1.6149	10.8043	6.8971	0.9853
Poisson(1)	0.9851	6.3957	6.8248	0.9750

Tabela 2. Estatísticas estimadas para diferentes distribuições de X e Y.

Sobre a caracterização do erro E = Y - $\hat{Y}$:

- a média do erro é da ordem de 10^{-16}→ estimador não-viesado.
- a variância do erro é extremamente baixa (10^{-27} a 10^{-26}) → erro quase nulo;

- Resultado esperado, haja vista que, foi usado o estimador linear ótimo baseado em amostras grandes.

O histograma reforça essa conclusão, mostrando erros concentrados em torno de zero com dispersão mínima, evidenciando a excelente precisão da predição.

Comparando as distribuições, todas apresentam comportamento semelhante em termos de erro médio e variância do erro, independentemente da simetria ou do suporte discreto de X. As diferenças observadas nos histogramas refletem apenas características da distribuição original, como a discreção do Bernoulli ou a assimetria positiva da geométrica e da Poisson.

O exercício demonstra que o estimador linear ótimo $\hat{Y} = E_Y[Y] + \frac{\text{cov}(X,Y)}{\text{var}(X)}(x - E_X[X])$. consegue praticamente eliminar o erro de predição, e a normalização da variância de X garante que as comparações entre distribuições sejam consistentes e significativas.

<u>Exercício 3:</u> Fazer exercícios 6.25, 6.29, 7.21, 7.49 do livro: Intuitive Probability and Random Processes using MATLAB de Stephen M. Kay, Springer, 2006.

3. EXERCÍCIOS DO LIVRO (KAY – CAPÍTULO 6 e 7)

<u>Exercicio 6.25:</u> Uma PMF simétrica satisfaz pX [−k] = pX [k] para k = −1,0,1,.... Prove que todos os momentos de ordem ímpar E[Xn] para n ímpar são zero.

<u>Resolução no Matlab:</u>

```
clc; clear; close all;
syms k n integer

% Definindo PMF simétrica genérica
syms p [1 3] real % com 3 pontos: -1,0,1
assumeAlso(p >= 0);
px = [p(1), p(2), p(3)];
k_vals = [-1 0 1];

% Momento de ordem n
Xn = sum(k_vals.^n .* px);

% Mostrar resultado para n ímpar
```

```
disp('Momento de ordem ímpar:');
simplify(subs(Xn, n, 1))   % n=1
simplify(subs(Xn, n, 3))   % n=3
```

Resultado no Matlab:

```
Command Window
Momento de ordem ímpar:

ans =

p3 - p1


ans =

p3 - p1
```

Figura 10. Resposta do Matlab.

Exercicio 6.29: Determine a variância de uma binomial usando a função característica.

Resolução no Matlab:

```
clc; clear; close all;
syms t n p real

q = 1 - p;

% Função característica da Binomial
phiX = (q + p*exp(1i*t))^n;

% Primeiro e segundo momentos
EX = diff(phiX, t);         % E[X] = i*EX(0)
EX2 = diff(phiX, t, 2);     % E[X^2] = -EX2(0)

% Avaliando em t=0
EX_val = simplify(subs(1i*EX, t, 0))
EX2_val = simplify(subs(-EX2, t, 0))

% Variância
VarX = simplify(EX2_val - EX_val^2)
```

Resultado no Matlab:

```
Command Window

EX_val =

-n*p


EX2_val =

n*(n - 1)*p^2 + n*p


VarX =

-n*p*(p - 1)
```

Figura 11. Resposta do Matlab.

<u>Exercicio 7.21:</u> O problema envolve verificar se duas variáveis aleatórias X e Y são independentes com base na função de massa de probabilidade conjunta (PMF) fornecida na tabela 3, abaixo.

$$p_{X,Y}[i,j] = \begin{cases} a & (i,j) = (0,0) \\ b & (i,j) = (0,1) \\ c & (i,j) = (1,0) \\ d & (i,j) = (1,1) \end{cases} \quad (2.0)$$

A condição mencionada é:

a*d = b*c onde a = $P_{XY}[0,0]$, b = $P_{XY}[0,1]$, c = $P_{XY}[1,0]$, d = $P_{XY}[1,1]$.

	$j=0$	$j=1$	$p_X[i]$
$i=0$	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{2}$
$i=1$	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{1}{2}$
$p_Y[j]$	$\frac{1}{2}$	$\frac{1}{2}$	

Tabela 3. PMF fornecida no livro para a resolução da atividade.

$p_X(0) = a+b \qquad p_X(1) = 1-(a+b)$

$p_X(0) = a+c \qquad p_X(1) = 1-(a+c)$

Para independência:

$a = (a+b)(a+c) \rightarrow (0,0)$

$b = (a+b)(1-(a+c)) \rightarrow (0,1)$

$c = (1-(a+b))(a+c) \rightarrow (1,0)$

$d = (1-(a+b))(1-(a+c)) \rightarrow (1,1)$

$\rightarrow \frac{a}{b} = \frac{c}{d}$

$\Rightarrow \boxed{ad = bc}$ resposta.

A condição necessária para independência é ad = bc. No exemplo da Tabela 3, não vale ad = bc. Portanto, X e Y são dependentes.

<u>Resolução no Matlab:</u>

```
% PMF conjunta
a = 3/8; b = 1/8; c = 1/8; d = 3/8;

% Verificando a condição ad = bc
lhs = a*d;  % ad
rhs = b*c;  % bc

disp(['a*d = ', num2str(lhs)])
```

```
disp(['b*c = ', num2str(rhs)])

if abs(lhs - rhs) < 1e-10
   disp('Variáveis possivelmente independentes')
else
   disp('Variáveis NÃO independentes')
end

% Checar a independência (usando marginais)
px = [a+b, c+d];  % P_X[0], P_X[1]
py = [a+c, b+d];  % P_Y[0], P_Y[1]

% Teste completo: verificar se PXY = PX*PY
PXY = [a b; c d];
PX_PY = px' * py;  % produto externo

disp('PXY calculado:')
disp(PXY)
disp('Produto das marginais PX*PY:')
disp(PX_PY)
```

<u>Resultado no Matlab:</u>

Figura 12. Resposta do Matlab.

Logo, a*d = 0.140625 e b*c = 0.015625 não são iguais, a condição necessária para independência falha.

<u>Exercicio 7.49:</u> Esse exercício está pedindo três coisas principais, todas relacionadas à estatística e probabilidade com variáveis aleatórias discretas.

1. Calcular o coeficiente de correlação teórico usando a função de massa de probabilidade conjunta (PMF) apresentada na tabela 4 para calcular o coeficiente de correlação entre as variáveis aleatórias X e Y.

2. Simular realizações do vetor aleatório (X, Y) usando um computador deve-se gerar M realizações (amostras) do vetor (X, Y), com base na PMF dada.

3. Estimar o coeficiente de correlação com a fórmula amostral com a fórmula fornecida para calcular uma estimativa empírica do coeficiente de correlação com base nas amostras geradas.

A estimativa deve se aproximar do valor teórico conforme o número de amostras M aumenta.

$$\hat{\rho}_{X,Y} = \frac{\frac{1}{M}\sum_{m=1}^{M} x_m y_m - \bar{x}\bar{y}}{\sqrt{\left(\frac{1}{M}\sum_{m=1}^{M} x_m^2 - \bar{x}^2\right)\left(\frac{1}{M}\sum_{m=1}^{M} y_m^2 - \bar{y}^2\right)}} \tag{3.0}$$

onde: $\bar{x} = \frac{1}{M}\sum_{m=1}^{M} x_m$, $\bar{y} = \frac{1}{M}\sum_{m=1}^{M} y_m$ e (x_m, y_m) é a m-ésima realização.

	$j=0$	$j=1$
$i=0$	$\frac{1}{8}$	$\frac{1}{8}$
$i=1$	$\frac{1}{4}$	$\frac{1}{2}$

Tabela 4. PMF fornecida no livro para a resolução da atividade.

	$j=0$	$j=1$	$P_X(i)$
$i=0$	$1/8$	$1/8$	$1/4$
$i=1$	$1/4$	$1/2$	$3/4$
$P_Y(j)$	$3/8$	$5/8$	

$E_X(x) = 3/4 \qquad E_Y(y) = 5/8$

$E_{XY}(xy) = 1/2$

$Cov(x,y) = 1/2 - 3/4\,(5/8) \Rightarrow 1/32$

$Var(x) = 3/4 - (3/4)^2 = 3/16$

$Var(y) = 5/8 - (5/8)^2 = 15/64$

$$\rho_{x,y} = \frac{1/32}{\sqrt{\frac{3}{16}\cdot\frac{15}{64}}} \Rightarrow \frac{1/32 \cdot (32)}{\sqrt{45}}$$

$$\rho_{x,y} = \frac{1}{3\sqrt{5}} = \frac{\sqrt{5}}{15} \Rightarrow 0{,}1490$$

$\hat{\rho}_{x,y} = 0{,}1497$ (correlação entre x e y)
↳ (ρ)

↳ correlação estimada dependência linear fraca entre x e y.

↳ não é uma probabilidade.

Resolução no Matlab:

```
clear all

rng(0,'twister');

M = 100000;
x = zeros(M,1);
y = zeros(M,1);

for m = 1:M
    u = rand;
    if u <= 1/8
        x(m) = 0; y(m) = 0;
    elseif u > 1/8 && u <= 1/4
        x(m) = 0; y(m) = 1;
    elseif u > 1/4 && u < 1/2
        x(m) = 1; y(m) = 0;
    else
        x(m) = 1; y(m) = 1;
    end
end

% Médias
EX = mean(x);
EY = mean(y);

% Variâncias (normalização por N)
varX = var(x,1);
varY = var(y,1);

% Covariância
covXY = mean(x.*y) - EX*EY;

% Correlação
rho = covXY / sqrt(varX*varY);

% Probabilidade conjunta (Phat)
Phat = zeros(2,2); % linhas X=0,1; colunas Y=0,1
Phat(1,1) = sum(x==0 & y==0)/M;
Phat(1,2) = sum(x==0 & y==1)/M;
Phat(2,1) = sum(x==1 & y==0)/M;
Phat(2,2) = sum(x==1 & y==1)/M;

% Resultados
disp('Médias:')
disp(['EX = ', num2str(EX), ', EY = ', num2str(EY)])
disp('Variâncias:')
disp(['VarX = ', num2str(varX), ', VarY = ', num2str(varY)])
disp(['CovXY = ', num2str(covXY)])
disp(['Correlação rho = ', num2str(rho)])
disp('Matriz de probabilidades conjuntas Phat (linhas X=0,1; colunas Y=0,1):')
disp(Phat)
```

Resultado no Matlab:

```
Command Window
>> Q749_lista3
Médias:
EX = 0.74939, EY = 0.62636
Variâncias:
VarX = 0.1878, VarY = 0.23403
CovXY = 0.030972
Correlação rho = 0.14773
Matriz de probabilidades conjuntas Phat (linhas X=0,1; colunas Y=0,1):
    0.1246    0.1260
    0.2490    0.5004
```

Figura 13. Resposta do Matlab.

4. CONCLUSÃO

Os resultados obtidos confirmam as propriedades esperadas de cada modelo probabilístico. No problema da caminhada aleatória, a simetria do polígono garante que todos os vértices, exceto o inicial, tenham igual probabilidade de serem o último visitado, fato confirmado pelas simulações.

Nos experimentos com distribuições discretas, a normalização da variância e o uso do estimador linear ótimo resultaram em erros médios praticamente nulos e histogramas concentrados em torno de zero, demonstrando a robustez do método mesmo em distribuições assimétricas.

Já nos exercícios teóricos, verificou-se que momentos ímpares de distribuições simétricas são nulos, que a variância da binomial pode ser obtida diretamente pela função característica, e que as condições de independência e correlação podem ser testadas de forma analítica e empírica.

De forma geral, a lista evidencia como teoria e simulação caminham lado a lado: enquanto a teoria fornece as relações fundamentais, as simulações ilustram e validam os resultados em cenários práticos.

A integração é essencial na engenharia e em demais áreas aplicadas, pois garante compreensão conceitual sólida aliada à capacidade de explorar e validar fenômenos estocásticos em situações reais.

5. REFERÊNCIAS

1. KAY, Steven M. Fundamentals of Statistical Signal Processing: Estimation Theory. Upper Saddle River: Prentice Hall, 1993.
2. MATHWORKS. MATLAB: The Language of Technical Computing. Natick: The MathWorks, 2023.