Distribuições de Probabilidade (Discretas e Contínuas)

Analisar e Aplicar as Principais Distribuições de Probabilidade em Problemas Reais

Márcio Nicolau

2025-10-22

Table of contents

Introdução e Objetivos **2**
Objetivos de Aprendizagem . . . 2

O Que São Distribuições de Probabilidade? **3**
Variáveis Aleatórias: Discretas vs. Contínuas . . . 3
Função Massa de Probabilidade (FMP) e Função Densidade de Probabilidade (FDP) . . . 3

Distribuições de Probabilidade Discretas **3**
Distribuição de Bernoulli . . . 3
Exemplo: Distribuição de Bernoulli . . . 4
Python . . . 4
R . . . 4
Distribuição Binomial . . . 5
Exemplo: Distribuição Binomial . . . 6
Python . . . 6
R . . . 6
Distribuição de Poisson . . . 7
Exemplo: Distribuição de Poisson . . . 8
Python . . . 8
R . . . 8

Distribuições de Probabilidade Contínuas **9**
Distribuição Uniforme . . . 9
Exemplo: Distribuição Uniforme . . . 10
Python . . . 10
R . . . 10
Distribuição Normal (Gaussiana) . . . 11
Exemplo: Distribuição Normal . . . 12
Python . . . 12
R . . . 13

Distribuição Exponencial 13
Exemplo: Distribuição Exponencial 14
Python 14
R 15

Relação com Outros Conceitos **15**

Verificação de Aprendizagem **17**
Soluções da Atividade 17

Referências Bibliográficas **17**

List of Figures

1 Relação das Distribuições de Probabilidade com Outros Conceitos Estatísticos 16

Introdução e Objetivos

Nas aulas anteriores, estabelecemos a base da probabilidade, da análise combinatória, da estatística descritiva, da amostragem e da estimação. Agora, vamos explorar o coração da probabilidade: as **Distribuições de Probabilidade**. Uma distribuição de probabilidade descreve todos os valores possíveis que uma variável aleatória pode assumir e a probabilidade associada a cada um desses valores.

Compreender as distribuições de probabilidade é fundamental porque muitos fenômenos do mundo real podem ser modelados por elas. Isso nos permite não apenas descrever o comportamento de variáveis aleatórias, mas também fazer previsões, realizar inferências e tomar decisões em contextos de incerteza, desde a previsão de demanda até o controle de qualidade e a avaliação de riscos.

Nesta aula, diferenciaremos as distribuições discretas das contínuas e exploraremos as características, fórmulas e aplicações das mais importantes em cada categoria, utilizando Python e R para visualização e cálculo.

Objetivos de Aprendizagem

Ao final desta aula, você será capaz de:

- Distinguir entre variáveis aleatórias discretas e contínuas.
- Definir e interpretar Funções Massa de Probabilidade (FMP) e Funções Densidade de Probabilidade (FDP).
- Caracterizar e aplicar as distribuições de Bernoulli, Binomial e Poisson para variáveis discretas.
- Caracterizar e aplicar as distribuições Uniforme, Normal e Exponencial para variáveis contínuas.
- Calcular probabilidades e gerar amostras a partir dessas distribuições usando Python e R.
- Identificar a distribuição de probabilidade mais apropriada para modelar diferentes cenários do mundo real.

O Que São Distribuições de Probabilidade?

Uma **distribuição de probabilidade** é uma função matemática que descreve a probabilidade de diferentes resultados ou valores para uma variável aleatória. Em outras palavras, ela nos diz quais valores uma variável aleatória pode assumir e com que frequência (ou probabilidade) cada um deles ocorre.

Variáveis Aleatórias: Discretas vs. Contínuas

A natureza da variável aleatória determina o tipo de distribuição de probabilidade.

- **Variável Aleatória Discreta:** Assume um número finito ou contável de valores. Geralmente são resultados de contagens.
  - **Exemplos:** Número de caras em 3 lançamentos de moeda ($\{0, 1, 2, 3\}$), número de carros que passam por um pedágio em uma hora ($\{0, 1, 2, ...\}$).
  - Associada à **Função Massa de Probabilidade (FMP)**.
- **Variável Aleatória Contínua:** Assume um número infinito de valores dentro de um intervalo. Geralmente são resultados de medições.
  - **Exemplos:** Altura de uma pessoa, tempo de espera em uma fila, temperatura de um ambiente.
  - Associada à **Função Densidade de Probabilidade (FDP)**.

Função Massa de Probabilidade (FMP) e Função Densidade de Probabilidade (FDP)

- **Função Massa de Probabilidade (FMP - $P(X = x)$):** Para variáveis discretas, a FMP atribui uma probabilidade a cada valor específico que a variável pode assumir. A soma de todas as probabilidades é igual a 1.
- **Função Densidade de Probabilidade (FDP - $f(x)$):** Para variáveis contínuas, a FDP não dá a probabilidade de um valor *exato* (pois a probabilidade de qualquer ponto específico é zero), mas sim a probabilidade de a variável cair dentro de um *intervalo*. A área sob a curva da FDP em um dado intervalo representa essa probabilidade, e a área total sob a curva é igual a 1.

Distribuições de Probabilidade Discretas

Distribuição de Bernoulli

A **Distribuição de Bernoulli** modela um experimento aleatório com apenas dois resultados possíveis: "sucesso" (com probabilidade p) e "fracasso" (com probabilidade $1 - p$). É a base para a distribuição binomial. (Bussab; Morettin, 2017, p. 116)

- **Parâmetro:** p (probabilidade de sucesso).
- **Variável Aleatória:** $X = 1$ para sucesso, $X = 0$ para fracasso.
- **Função Massa de Probabilidade (FMP):**

  $P(X = x) = p^x(1 - p)^{1-x}$, para $x \in \{0, 1\}$

- **Média:** $E[X] = p$
- **Variância:** $Var[X] = p(1-p)$

Exemplo: Lançamento de uma moeda justa (sucesso = cara, $p = 0.5$).

Exemplo: Distribuição de Bernoulli

Python

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import bernoulli

p = 0.7  # Probabilidade de sucesso

# Calcule a FMP para x=0 e x=1
pmf_0 = bernoulli.pmf(0, p)
pmf_1 = bernoulli.pmf(1, p)

print(f"P(X=0) (fracasso) = {pmf_0:.2f}")
print(f"P(X=1) (sucesso) = {pmf_1:.2f}")

# Simulação de 10 ensaios de Bernoulli
simulacao_bernoulli = bernoulli.rvs(p, size=10)
print(f"\nSimulação de 10 ensaios de Bernoulli (p={p}): {simulacao_bernoulli}")

# Visualização da FMP
x_values =
pmf_values = [pmf_0, pmf_1]

plt.figure(figsize=(6, 4))
plt.bar(x_values, pmf_values, color=['red', 'green'], width=0.4)
plt.title(f'FMP da Distribuição de Bernoulli (p={p})')
plt.xlabel('Resultado (0=Fracasso, 1=Sucesso)')
plt.ylabel('Probabilidade')
plt.xticks(x_values)
plt.ylim(0, 1)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

p <- 0.7 # Probabilidade de sucesso
```

```r
# Calcule a FMP para x=0 e x=1
pmf_0 <- dbinom(0, size = 1, prob = p) # dbinom para Bernoulli (n=1)
pmf_1 <- dbinom(1, size = 1, prob = p)

cat(sprintf("P(X=0) (fracasso) = %.2f\n", pmf_0))
cat(sprintf("P(X=1) (sucesso) = %.2f\n", pmf_1))

# Simulação de 10 ensaios de Bernoulli
set.seed(42) # para reprodutibilidade
simulacao_bernoulli <- rbinom(10, size = 1, prob = p)
cat(sprintf("\nSimulação de 10 ensaios de Bernoulli (p=%.1f): %s\n", p, paste(simulacao_bernoulli, colla

# Visualização da FMP
df_bernoulli <- data.frame(
  Resultado = factor(c(0, 1), labels = c("Fracasso", "Sucesso")),
  Probabilidade = c(pmf_0, pmf_1)
)

ggplot(df_bernoulli, aes(x = Resultado, y = Probabilidade, fill = Resultado)) +
  geom_bar(stat = "identity", width = 0.4) +
  labs(title = paste0('FMP da Distribuição de Bernoulli (p=', p, ')'),
       x = 'Resultado',
       y = 'Probabilidade') +
  scale_fill_manual(values = c("Fracasso" = "red", "Sucesso" = "green")) +
  ylim(0, 1) +
  theme_minimal() +
  theme(legend.position = "none")
```

Distribuição Binomial

A **Distribuição Binomial** modela o número de sucessos em uma sequência fixa de n ensaios de Bernoulli independentes e idênticos. (Bussab; Morettin, 2017, p. 116–118)

- **Parâmetros:** n (número de ensaios), p (probabilidade de sucesso em um único ensaio).
- **Variável Aleatória:** X = número de sucessos, onde $X \in \{0, 1, 2, \ldots, n\}$.
- **Função Massa de Probabilidade (FMP):**

  $P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$, para $x = 0, 1, \ldots, n$. (Onde $\binom{n}{x} = \frac{n!}{x!(n-x)!}$ é o coeficiente binomial, número de combinações de n itens tomados x a x).
- **Média:** $E[X] = np$
- **Variância:** $Var[X] = np(1-p)$

Exemplo: Número de peças defeituosas em uma amostra de 10 peças, onde cada peça tem 5% de chance de ser defeituosa.

Exemplo: Distribuição Binomial

Python

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import binom

n = 10  # Número de ensaios
p = 0.3 # Probabilidade de sucesso

# Calcule a probabilidade de obter exatamente 3 sucessos
pmf_3_sucessos = binom.pmf(3, n, p)
print(f"P(X=3) (exatamente 3 sucessos em {n} ensaios) = {pmf_3_sucessos:.3f}")

# Calcule a probabilidade acumulada (CDF) de obter até 3 sucessos
cdf_ate_3_sucessos = binom.cdf(3, n, p)
print(f"P(X<=3) (até 3 sucessos em {n} ensaios) = {cdf_ate_3_sucessos:.3f}")

# Simulação de 500 resultados de um experimento Binomial
simulacao_binomial = binom.rvs(n, p, size=500)
print(f"\nSimulação de 10 resultados de experimento Binomial (n={n}, p={p}): {simulacao_binomial[:10]}.

# Visualização da FMP
x_values = np.arange(0, n + 1)
pmf_values = binom.pmf(x_values, n, p)

plt.figure(figsize=(10, 6))
plt.bar(x_values, pmf_values, color='skyblue', edgecolor='black')
plt.title(f'FMP da Distribuição Binomial (n={n}, p={p})')
plt.xlabel('Número de Sucessos')
plt.ylabel('Probabilidade')
plt.xticks(x_values)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

n <- 10 # Número de ensaios
p <- 0.3 # Probabilidade de sucesso

# Calcule a probabilidade de obter exatamente 3 sucessos
pmf_3_sucessos <- dbinom(3, size = n, prob = p)
```

```r
cat(sprintf("P(X=3) (exatamente 3 sucessos em %d ensaios) = %.3f\n", n, pmf_3_sucessos))

# Calcule a probabilidade acumulada (CDF) de obter até 3 sucessos
cdf_ate_3_sucessos <- pbinom(3, size = n, prob = p)
cat(sprintf("P(X<=3) (até 3 sucessos em %d ensaios) = %.3f\n", n, cdf_ate_3_sucessos))

# Simulação de 500 resultados de um experimento Binomial
set.seed(42) # para reprodutibilidade
simulacao_binomial <- rbinom(500, size = n, prob = p)
cat(sprintf("\nSimulação de 10 resultados de experimento Binomial (n=%d, p=%.1f): %s...\n", n, p, paste

# Visualização da FMP
x_values <- 0:n
pmf_values <- dbinom(x_values, size = n, prob = p)
df_binomial <- data.frame(Numero_Sucessos = x_values, Probabilidade = pmf_values)

ggplot(df_binomial, aes(x = Numero_Sucessos, y = Probabilidade)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(title = paste0('FMP da Distribuição Binomial (n=', n, ', p=', p, ')'),
       x = 'Número de Sucessos',
       y = 'Probabilidade') +
  scale_x_continuous(breaks = x_values) +
  theme_minimal()
```

Distribuição de Poisson

A **Distribuição de Poisson** modela o número de ocorrências de um evento em um intervalo fixo de tempo ou espaço, quando esses eventos ocorrem com uma taxa média constante e são independentes entre si. (Bussab; Morettin, 2017, p. 120)

- **Parâmetro:** λ (lambda - taxa média de ocorrências no intervalo).
- **Variável Aleatória:** X = número de ocorrências, onde $X \in \{0, 1, 2, ...\}$.
- **Função Massa de Probabilidade (FMP):**

  $P(X = x) = \frac{e^{-\lambda}\lambda^x}{x!}$, para $x = 0, 1, 2, ...$ ($e \approx 2.71828$ é a base do logaritmo natural).
- **Média:** $E[X] = \lambda$
- **Variância:** $Var[X] = \lambda$

Exemplo: Número de chamadas recebidas por um call center em 5 minutos, se a taxa média é de 3 chamadas por 5 minutos.

Exemplo: Distribuição de Poisson

Python

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import poisson

lambda_param = 3.5 # Taxa média de ocorrências por intervalo

# Calcule a probabilidade de ocorrer exatamente 2 eventos
pmf_2_eventos = poisson.pmf(2, lambda_param)
print(f"P(X=2) (exatamente 2 eventos com  ={lambda_param}) = {pmf_2_eventos:.3f}")

# Calcule a probabilidade acumulada (CDF) de ocorrer até 2 eventos
cdf_ate_2_eventos = poisson.cdf(2, lambda_param)
print(f"P(X<=2) (até 2 eventos com  ={lambda_param}) = {cdf_ate_2_eventos:.3f}")

# Simulação de 1000 resultados de um processo de Poisson
simulacao_poisson = poisson.rvs(lambda_param, size=1000)
print(f"\nSimulação de 10 resultados de Poisson ( ={lambda_param}): {simulacao_poisson[:10]}...")

# Visualização da FMP
x_values = np.arange(0, int(lambda_param * 3)) # Abrange a maior parte da distribuição
pmf_values = poisson.pmf(x_values, lambda_param)

plt.figure(figsize=(10, 6))
plt.bar(x_values, pmf_values, color='lightcoral', edgecolor='black')
plt.title(f'FMP da Distribuição de Poisson ( ={lambda_param})')
plt.xlabel('Número de Ocorrências')
plt.ylabel('Probabilidade')
plt.xticks(x_values)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

lambda_param <- 3.5 # Taxa média de ocorrências por intervalo

# Calcule a probabilidade de ocorrer exatamente 2 eventos
pmf_2_eventos <- dpois(2, lambda = lambda_param)
cat(sprintf("P(X=2) (exatamente 2 eventos com  =%.1f) = %.3f\n", lambda_param, pmf_2_eventos))
```

```r
# Calcule a probabilidade acumulada (CDF) de ocorrer até 2 eventos
cdf_ate_2_eventos <- ppois(2, lambda = lambda_param)
cat(sprintf("P(X<=2) (até 2 eventos com  =%.1f) = %.3f\n", lambda_param, cdf_ate_2_eventos))

# Simulação de 1000 resultados de um processo de Poisson
set.seed(42) # para reprodutibilidade
simulacao_poisson <- rpois(1000, lambda = lambda_param)
cat(sprintf("\nSimulação de 10 resultados de Poisson ( =%.1f): %s...\n", lambda_param, paste(head(simula

# Visualização da FMP
x_values <- 0:(ceiling(lambda_param * 3))
pmf_values <- dpois(x_values, lambda = lambda_param)
df_poisson <- data.frame(Numero_Ocorrencias = x_values, Probabilidade = pmf_values)

ggplot(df_poisson, aes(x = Numero_Ocorrencias, y = Probabilidade)) +
  geom_bar(stat = "identity", fill = "lightcoral", color = "black") +
  labs(title = paste0('FMP da Distribuição de Poisson ( =', lambda_param, ')'),
       x = 'Número de Ocorrências',
       y = 'Probabilidade') +
  scale_x_continuous(breaks = x_values) +
  theme_minimal()
```

Distribuições de Probabilidade Contínuas

Distribuição Uniforme

A **Distribuição Uniforme** descreve uma variável aleatória contínua onde todos os valores dentro de um determinado intervalo têm a mesma probabilidade de ocorrência. É como um dado de infinitas faces em um intervalo. (Bussab; Morettin, 2017, p. 125)

- **Parâmetros:** a (limite inferior), b (limite superior).
- **Variável Aleatória:** $X \in [a, b]$.
- **Função Densidade de Probabilidade (FDP):**

  $f(x) = \frac{1}{b-a}$, para $a \leq x \leq b$ $f(x) = 0$, caso contrário.
- **Média:** $E[X] = \frac{a+b}{2}$
- **Variância:** $Var[X] = \frac{(b-a)^2}{12}$

Exemplo: O tempo de espera por um ônibus que chega a cada 10 minutos (entre 0 e 10 minutos).

Exemplo: Distribuição Uniforme

Python

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import uniform

a = 0   # Limite inferior
b = 10  # Limite superior

# Probabilidade de X estar entre 3 e 7
prob_intervalo = uniform.cdf(7, loc=a, scale=b-a) - uniform.cdf(3, loc=a, scale=b-a)
print(f"P(3 <= X <= 7) na Distribuição Uniforme [{a}, {b}] = {prob_intervalo:.2f}")

# Simulação de 1000 valores
simulacao_uniforme = uniform.rvs(loc=a, scale=b-a, size=1000)
print(f"\nSimulação de 10 valores Uniformes [{a}, {b}]: {simulacao_uniforme[:10].round(2)}...")

# Visualização da FDP
x_values = np.linspace(a - 2, b + 2, 500) # Extende um pouco para mostrar f(x)=0
pdf_values = uniform.pdf(x_values, loc=a, scale=b-a)

plt.figure(figsize=(10, 6))
plt.plot(x_values, pdf_values, color='darkblue', linewidth=2)
plt.fill_between(x_values, pdf_values, color='skyblue', alpha=0.5)
plt.title(f'FDP da Distribuição Uniforme (a={a}, b={b})')
plt.xlabel('Valor de X')
plt.ylabel('Densidade de Probabilidade')
plt.ylim(bottom=0)
plt.grid(linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

a <- 0   # Limite inferior
b <- 10  # Limite superior

# Probabilidade de X estar entre 3 e 7
prob_intervalo <- punif(7, min = a, max = b) - punif(3, min = a, max = b)
cat(sprintf("P(3 <= X <= 7) na Distribuição Uniforme [%d, %d] = %.2f\n", a, b, prob_intervalo))

# Simulação de 1000 valores
```

```r
set.seed(42) # para reprodutibilidade
simulacao_uniforme <- runif(1000, min = a, max = b)
cat(sprintf("\nSimulação de 10 valores Uniformes [%d, %d]: %s...\n", a, b, paste(round(head(simulacao_u

# Visualização da FDP
df_uniforme <- data.frame(x = c(a - 2, a, b, b + 2), y = c(0, 1 / (b - a), 1 / (b - a), 0))

ggplot(df_uniforme, aes(x = x, y = y)) +
  geom_line(color = "darkblue", size = 1) +
  geom_ribbon(aes(ymin = 0, ymax = y), fill = "skyblue", alpha = 0.5) +
  labs(title = paste0('FDP da Distribuição Uniforme (a=', a, ', b=', b, ')'),
       x = 'Valor de X',
       y = 'Densidade de Probabilidade') +
  ylim(0, 1 / (b - a) * 1.5) + # Ajusta o limite Y para visualização
  theme_minimal()
```

Distribuição Normal (Gaussiana)

A **Distribuição Normal**, ou Gaussiana, é a mais importante e amplamente utilizada distribuição contínua. É caracterizada por sua forma de sino simétrica. (Bussab; Morettin, 2017, p. 128–132)

- **Parâmetros:** μ (média), σ (desvio padrão).
- **Variável Aleatória:** $X \in (-\infty, +\infty)$.
- **Função Densidade de Probabilidade (FDP):**

  $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$
- **Média:** $E[X] = \mu$
- **Variância:** $Var[X] = \sigma^2$

Propriedades Chave:

- **Simetria:** A média, mediana e moda são todas iguais.
- **Regra Empírica (ou 68-95-99.7):**
  - Aproximadamente 68% dos dados estão dentro de 1 desvio padrão da média.
  - Aproximadamente 95% dos dados estão dentro de 2 desvios padrão da média.
  - Aproximadamente 99.7% dos dados estão dentro de 3 desvios padrão da média.
- **Teorema do Limite Central:** A soma ou média de um grande número de variáveis aleatórias independentes e identicamente distribuídas (mesmo que não sejam normais) tenderá a ter uma distribuição aproximadamente normal. Esta é a razão de sua ubiquidade na estatística inferencial.

Exemplo: Distribuição Normal

Python

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

mu = 70   # Média
sigma = 5 # Desvio Padrão

# Probabilidade de X ser menor que 65
prob_menor_65 = norm.cdf(65, loc=mu, scale=sigma)
print(f"P(X < 65) na Normal ( ={mu},  ={sigma}) = {prob_menor_65:.3f}")

# Probabilidade de X estar entre 65 e 75
prob_entre_65_75 = norm.cdf(75, loc=mu, scale=sigma) - norm.cdf(65, loc=mu, scale=sigma)
print(f"P(65 <= X <= 75) na Normal ( ={mu},  ={sigma}) = {prob_entre_65_75:.3f}")

# Simulação de 1000 valores
simulacao_normal = norm.rvs(loc=mu, scale=sigma, size=1000)
print(f"\nSimulação de 10 valores Normais ( ={mu},  ={sigma}): {simulacao_normal[:10].round(2)}...")

# Visualização da FDP
x_values = np.linspace(mu - 4*sigma, mu + 4*sigma, 500)
pdf_values = norm.pdf(x_values, loc=mu, scale=sigma)

plt.figure(figsize=(10, 6))
plt.plot(x_values, pdf_values, color='darkgreen', linewidth=2)
plt.fill_between(x_values, pdf_values, color='lightgreen', alpha=0.5)

# Marcar média e desvios padrão para a regra empírica
plt.axvline(mu, color='gray', linestyle='--', label=f'Média ( ={mu})')
for i in range(1, 4):
    plt.axvline(mu - i*sigma, color='purple', linestyle=':', alpha=0.7)
    plt.axvline(mu + i*sigma, color='purple', linestyle=':', alpha=0.7)

plt.title(f'FDP da Distribuição Normal ( ={mu},  ={sigma})')
plt.xlabel('Valor de X')
plt.ylabel('Densidade de Probabilidade')
plt.ylim(bottom=0)
plt.grid(linestyle='--', alpha=0.7)
plt.legend()
plt.show()
```

R

```r
library(ggplot2)

mu <- 70   # Média
sigma <- 5 # Desvio Padrão

# Probabilidade de X ser menor que 65
prob_menor_65 <- pnorm(65, mean = mu, sd = sigma)
cat(sprintf("P(X < 65) na Normal ( =%.0f,  =%.0f) = %.3f\n", mu, sigma, prob_menor_65))

# Probabilidade de X estar entre 65 e 75
prob_entre_65_75 <- pnorm(75, mean = mu, sd = sigma) - pnorm(65, mean = mu, sd = sigma)
cat(sprintf("P(65 <= X <= 75) na Normal ( =%.0f,  =%.0f) = %.3f\n", mu, sigma, prob_entre_65_75))

# Simulação de 1000 valores
set.seed(42) # para reprodutibilidade
simulacao_normal <- rnorm(1000, mean = mu, sd = sigma)
cat(sprintf("\nSimulação de 10 valores Normais ( =%.0f,  =%.0f): %s...\n", mu, sigma, paste(round(head(s

# Visualização da FDP
x_values <- seq(mu - 4 * sigma, mu + 4 * sigma, length.out = 500)
pdf_values <- dnorm(x_values, mean = mu, sd = sigma)
df_normal <- data.frame(x = x_values, y = pdf_values)

ggplot(df_normal, aes(x = x, y = y)) +
  geom_line(color = "darkgreen", size = 1) +
  geom_ribbon(aes(ymin = 0, ymax = y), fill = "lightgreen", alpha = 0.5) +
  geom_vline(xintercept = mu, linetype = "dashed", color = "gray", size = 0.8, aes(linetype = "Média")) 
  geom_vline(xintercept = c(mu - sigma, mu + sigma), linetype = "dotted", color = "purple", size = 0.6,
  geom_vline(xintercept = c(mu - 2*sigma, mu + 2*sigma), linetype = "dotted", color = "purple", size = 
  geom_vline(xintercept = c(mu - 3*sigma, mu + 3*sigma), linetype = "dotted", color = "purple", size = 
  labs(title = paste0('FDP da Distribuição Normal ( =', mu, ',  =', sigma, ')'),
       x = 'Valor de X',
       y = 'Densidade de Probabilidade',
       linetype = "Marcadores") +
  theme_minimal() +
  scale_linetype_manual(values = c("Média" = "dashed", "1 SD" = "dotted", "2 SD" = "dotted", "3 SD" = "
                        labels = c("Média", "±1 SD", "±2 SD", "±3 SD")) +
  theme(legend.position = "bottom")
```

Distribuição Exponencial

A **Distribuição Exponencial** modela o tempo entre eventos em um processo de Poisson, ou o tempo até a primeira ocorrência de um evento, onde os eventos ocorrem com uma taxa média constante. Possui a

propriedade de “ausência de memória”. (Bussab; Morettin, 2017, p. 126)

- **Parâmetro:** λ (lambda - taxa de ocorrência, inversa da média do tempo).
- **Variável Aleatória:** $X \geq 0$.
- **Função Densidade de Probabilidade (FDP):**

  $f(x) = \lambda e^{-\lambda x}$, para $x \geq 0$ $f(x) = 0$, para $x < 0$.
- **Média:** $E[X] = 1/\lambda$
- **Variância:** $Var[X] = 1/\lambda^2$

Exemplo: O tempo de vida de um componente eletrônico, ou o tempo de espera entre chegadas de clientes em uma fila.

Exemplo: Distribuição Exponencial

Python

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import expon

lambda_param = 0.5 # Taxa (eventos por unidade de tempo)
mean_time = 1 / lambda_param # Tempo médio entre eventos

# Probabilidade de um evento ocorrer em menos de 2 unidades de tempo
prob_menor_2 = expon.cdf(2, scale=mean_time) # scale é 1/lambda_param
print(f"P(X < 2) na Exponencial ( ={lambda_param}) = {prob_menor_2:.3f}")

# Simulação de 1000 valores
simulacao_exponencial = expon.rvs(scale=mean_time, size=1000)
print(f"\nSimulação de 10 valores Exponenciais ( ={lambda_param}): {simulacao_exponencial[:10].round(2)}

# Visualização da FDP
x_values = np.linspace(0, 4 * mean_time, 500) # Vai até cerca de 4 vezes o tempo médio
pdf_values = expon.pdf(x_values, scale=mean_time)

plt.figure(figsize=(10, 6))
plt.plot(x_values, pdf_values, color='darkred', linewidth=2)
plt.fill_between(x_values, pdf_values, color='lightsalmon', alpha=0.5)
plt.title(f'FDP da Distribuição Exponencial ( ={lambda_param}, Média={mean_time:.1f})')
plt.xlabel('Tempo (X)')
plt.ylabel('Densidade de Probabilidade')
plt.ylim(bottom=0)
plt.grid(linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

lambda_param <- 0.5 # Taxa (eventos por unidade de tempo)
mean_time <- 1 / lambda_param # Tempo médio entre eventos

# Probabilidade de um evento ocorrer em menos de 2 unidades de tempo
prob_menor_2 <- pexp(2, rate = lambda_param) # rate é lambda
cat(sprintf("P(X < 2) na Exponencial ( =%.1f) = %.3f\n", lambda_param, prob_menor_2))

# Simulação de 1000 valores
set.seed(42) # para reprodutibilidade
simulacao_exponencial <- rexp(1000, rate = lambda_param)
cat(sprintf("\nSimulação de 10 valores Exponenciais ( =%.1f): %s...\n", lambda_param, paste(round(head(s

# Visualização da FDP
x_values <- seq(0, 4 * mean_time, length.out = 500)
pdf_values <- dexp(x_values, rate = lambda_param)
df_exponential <- data.frame(x = x_values, y = pdf_values)

ggplot(df_exponential, aes(x = x, y = y)) +
  geom_line(color = "darkred", size = 1) +
  geom_ribbon(aes(ymin = 0, ymax = y), fill = "lightsalmon", alpha = 0.5) +
  labs(title = paste0('FDP da Distribuição Exponencial ( =', lambda_param, ', Média=', mean_time, ')'),
       x = 'Tempo (X)',
       y = 'Densidade de Probabilidade') +
  theme_minimal()
```

Relação com Outros Conceitos

As distribuições de probabilidade são a espinha dorsal de muitos conceitos em estatística e ciência de dados.

- **Probabilidade e Análise Combinatória:** As FMPs das distribuições discretas (como a binomial) são construídas diretamente a partir de princípios de contagem.
- **Estatística Descritiva:** A média e a variância de uma distribuição são seus parâmetros descritivos teóricos, que são estimados por suas contrapartes amostrais na estatística descritiva.
- **Amostragem:** A distribuição amostral de uma estatística (como a média amostral) é ela própria uma distribuição de probabilidade, frequentemente aproximada pela Normal (graças ao TLC).
- **Estimação:** Os intervalos de confiança dependem da distribuição teórica da estatística de teste (Normal, t de Student, etc.), que por sua vez são distribuições de probabilidade.
- **Testes de Hipóteses:** A tomada de decisão em testes de hipóteses baseia-se na comparação de uma estatística de teste com uma distribuição de probabilidade (por exemplo, Z-score com a Normal padrão, t-score com a t de Student) para calcular p-valores.

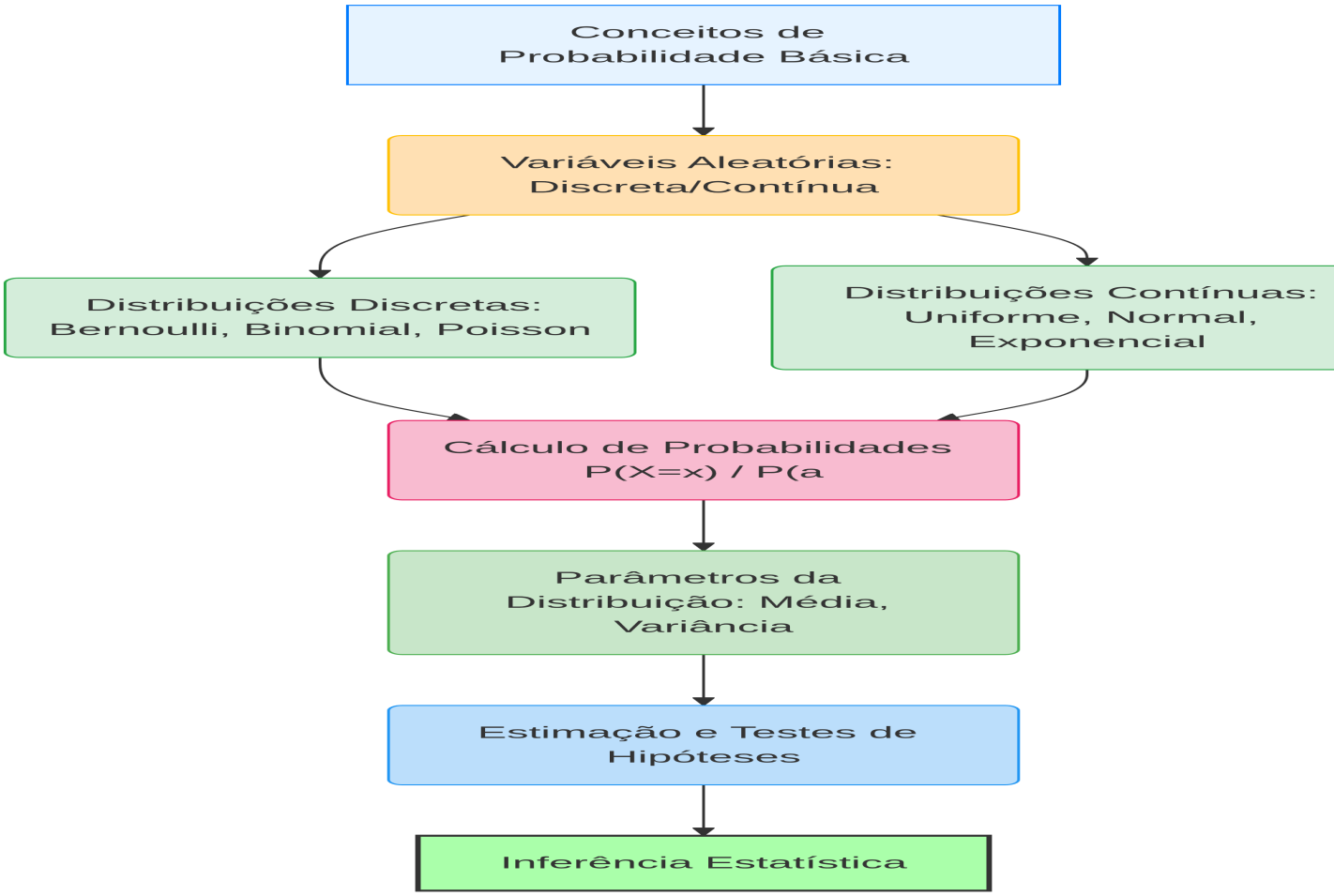

Figure 1: Relação das Distribuições de Probabilidade com Outros Conceitos Estatísticos

Verificação de Aprendizagem

Resolva os problemas abaixo, identificando a distribuição de probabilidade apropriada e utilizando Python ou R para calcular as probabilidades ou simular os resultados.

1. **Problema 1 (Controle de Qualidade - Binomial):**

   Uma linha de produção fabrica resistores, e 5% deles são defeituosos. Uma amostra de 20 resistores é selecionada aleatoriamente.

   a) Qual a probabilidade de que exatamente 2 resistores na amostra sejam defeituosos?
   b) Qual a probabilidade de que no máximo 1 resistor na amostra seja defeituoso?
   c) Qual a probabilidade de que pelo menos 3 resistores na amostra sejam defeituosos?

2. **Problema 2 (Chamadas de Suporte - Poisson):**

   O número de chamadas de suporte técnico recebidas por minuto segue uma distribuição de Poisson com uma taxa média de 2.5 chamadas por minuto.

   a) Qual a probabilidade de que exatamente 3 chamadas sejam recebidas em um minuto?
   b) Qual a probabilidade de que 5 ou mais chamadas sejam recebidas em um minuto?

3. **Problema 3 (Tempo de Espera - Exponencial):**

   O tempo (em minutos) que um cliente espera na fila de um caixa eletrônico segue uma distribuição exponencial com uma taxa média de 0.2 clientes por minuto (ou seja, o tempo médio de espera é de 5 minutos).

   a) Qual a probabilidade de um cliente esperar mais de 10 minutos?
   b) Qual a probabilidade de um cliente esperar entre 3 e 7 minutos?

4. **Problema 4 (Notas de Exame - Normal):**

   As notas de um exame final seguem uma distribuição normal com média de 70 e desvio padrão de 8.

   a) Qual a probabilidade de um aluno tirar uma nota maior que 85?
   b) Qual a probabilidade de um aluno tirar uma nota entre 60 e 80?
   c) Que nota separa os 10% melhores alunos?

Soluções da Atividade

Referências Bibliográficas

BUSSAB, Luiz O. de M.; MORETTIN, Pedro A. **Estatística Básica**. 9. ed. São Paulo: Saraiva, 2017.