trab_final

July 7, 2023

Um Framework vetorial para a interpretação e computação estatística e probabilística

Sumário

0.1 Introdução a Probabilidade

0.1.1 Probabilidade

Probabilidade é uma medida numérica que quantifica a chance ou a possibilidade de um evento ocorrer. Denotamos a probabilidade de um evento A como P(A), onde P é a função de probabilidade.

Espaço Amostral e Eventos Em probabilidade, trabalhamos com um espaço amostral, denotado por Ω, que é o conjunto de todos os resultados possíveis de um experimento aleatório. Um evento é um subconjunto do espaço amostral, que consiste em um ou mais resultados possíveis.

Axiomas de Probabilidade A teoria das probabilidades é fundamentada em três axiomas:

1. **Axioma da não-negatividade**: Para qualquer evento A, a probabilidade de A é um número não negativo: P(A) 0.
2. **Axioma da aditividade**: Para qualquer sequência de eventos mutuamente exclusivos A , A , ..., a probabilidade da união dos eventos é igual à soma das probabilidades individuais: P(A A ...) = P(A) + P(A) + ...
3. **Axioma da normalização**: A probabilidade do espaço amostral completo Ω é igual a 1: $P(\Omega) = 1$.

A partir desses axiomas, podemos deduzir várias propriedades e teoremas da teoria das probabilidades.

Probabilidade Condicional A probabilidade condicional é a probabilidade de um evento ocorrer, dado que outro evento já ocorreu. Denotamos a probabilidade condicional de A dado B como P(A|B). A fórmula para calcular a probabilidade condicional é: P(A|B) = P(A B) / P(B)

Regra do Produto e Probabilidade Conjunta A regra do produto é usada para calcular a probabilidade conjunta de dois eventos A e B, ou seja, a probabilidade de ambos os eventos ocorrerem simultaneamente. A fórmula para a probabilidade conjunta é:

P(A B) = P(A|B) * P(B)

Regra da Soma e Probabilidade Marginal A regra da soma é usada para calcular a probabilidade de um evento ocorrer, considerando diferentes cenários ou possibilidades. A fórmula para a probabilidade de um evento A é dada pela regra da soma:

P(A) = P(A B)

onde B são eventos mutuamente exclusivos que cobrem todo o espaço amostral.

Teorema de Bayes O Teorema de Bayes é uma ferramenta importante na teoria das probabilidades para atualizar a probabilidade de um evento dado o conhecimento de outro evento relacionado. A fórmula do Teorema de Bayes é:

P(A|B) = (P(B|A) * P(A)) / P(B)

onde P(A) e P(B) são as probabilidades marginais e P(B|A) é a probabilidade condicional.

0.1.2 Variáveis Aleatórias

Uma variável aleatória é uma função que associa um número real a cada resultado de um experimento aleatório. Ela mapeia os resultados do espaço amostral para valores numéricos, permitindo a quantificação probabilística dos eventos.

Variáveis Aleatórias Discretas Uma variável aleatória discreta assume um conjunto finito ou infinito contável de valores possíveis. A função de probabilidade de uma variável aleatória discreta, denotada por P(X), atribui probabilidades a cada valor possível da variável aleatória.

A função de probabilidade de uma variável aleatória discreta deve satisfazer as seguintes propriedades:

1. **Não-negatividade**: A probabilidade de um valor da variável aleatória é não negativa: P(X = x) 0 para todos os valores de x.
2. **Normalização**: A soma das probabilidades de todos os valores possíveis é igual a 1: P(X = x) = 1, onde a soma é realizada sobre todos os valores de x.

A partir da função de probabilidade, podemos calcular a função de distribuição acumulada (FDA) de uma variável aleatória discreta, que fornece a probabilidade acumulada de obter um valor menor ou igual a um determinado valor.

Variáveis Aleatórias Contínuas Uma variável aleatória contínua pode assumir qualquer valor em um intervalo contínuo. A função densidade de probabilidade (FDP) de uma variável aleatória contínua, denotada por f(x), descreve a distribuição de probabilidade ao longo do intervalo.

A função densidade de probabilidade deve satisfazer as seguintes propriedades:

1. **Não-negatividade**: A densidade de probabilidade é não negativa para todos os valores de x: f(x) 0.
2. **Normalização**: A área sob a curva da densidade de probabilidade é igual a 1: f(x) dx = 1, onde a integral é realizada sobre todos os valores de x.

A partir da função densidade de probabilidade, podemos calcular a função de distribuição acumulada (FDA) de uma variável aleatória contínua, que fornece a probabilidade acumulada de obter um valor menor ou igual a um determinado valor.

Esperança e Momentos A esperança (valor esperado) de uma variável aleatória é uma medida numérica que representa o valor médio esperado da variável. Para uma variável aleatória discreta, a esperança é calculada como a soma ponderada dos valores possíveis, multiplicados pelas probabilidades correspondentes.

Para uma variável aleatória contínua, a esperança é calculada como a integral ponderada dos valores possíveis, multiplicados pelas densidades de probabilidade correspondentes.

Os momentos de uma variável aleatória são medidas estatísticas que descrevem sua distribuição. O momento de ordem r é dado por E[X^r] para uma variável aleatória X.

Funções de Distribuição As funções de distribuição são usadas para caracterizar completamente uma variável aleatória. A função de distribuição acumulada (FDA) é uma função que fornece a probabilidade de obter um valor menor ou igual a um determinado valor.

Para uma variável aleatória discreta, a FDA é dada pela soma acumulada das probabilidades. Para uma variável aleatória contínua, a FDA é dada pela integral acumulada da densidade de probabilidade.

Esses conceitos fundamentais das variáveis aleatórias fornecem a base para a análise probabilística de eventos e a modelagem de incertezas em problemas estatísticos.

As funções de distribuição são usadas para caracterizar completamente uma variável aleatória. A função de distribuição acumulada (FDA) é uma função que fornece a probabilidade de obter um valor menor ou igual a um determinado valor.

Para uma variável aleatória discreta, a FDA é dada pela soma acumulada das probabilidades. Para uma variável aleatória contínua, a FDA é dada pela integral acumulada da densidade de probabilidade.

Esses conceitos fundamentais das variáveis aleatórias fornecem a base para a análise probabilística de eventos e a modelagem de incertezas em problemas estatísticos.

0.1.3 Distribuições de Probabilidade

Uma distribuição de probabilidade descreve a forma como os valores de uma variável aleatória estão distribuídos. Ela especifica as probabilidades associadas a cada possível resultado ou intervalo de valores.

Distribuições Discretas

Distribuição de Bernoulli A distribuição de Bernoulli modela um experimento aleatório que tem dois resultados possíveis, geralmente rotulados como sucesso (1) ou fracasso (0). A função de probabilidade de uma variável aleatória com distribuição de Bernoulli é dada por:

P(X = x) = p^x * (1 - p)^(1 - x)

onde x assume os valores 0 ou 1, e p é a probabilidade de sucesso.

Distribuição Binomial A distribuição binomial descreve o número de sucessos em uma sequência de experimentos independentes e identicamente distribuídos (i.i.d.) com probabilidade de sucesso p. A função de probabilidade de uma variável aleatória com distribuição binomial é dada por:

P(X = k) = C(n, k) * p^k * (1 - p)^(n - k)

onde n é o número de experimentos, k é o número de sucessos, p é a probabilidade de sucesso e C(n, k) é o coeficiente binomial.

Distribuição de Poisson A distribuição de Poisson descreve o número de eventos que ocorrem em um intervalo de tempo ou espaço fixo, quando os eventos ocorrem independentemente com uma taxa média conhecida. A função de probabilidade de uma variável aleatória com distribuição de Poisson é dada por:

P(X = k) = (e^(-) * ^k) / k!

onde k é o número de eventos, é a taxa média de ocorrência e e é a base do logaritmo natural.

Distribuições Contínuas

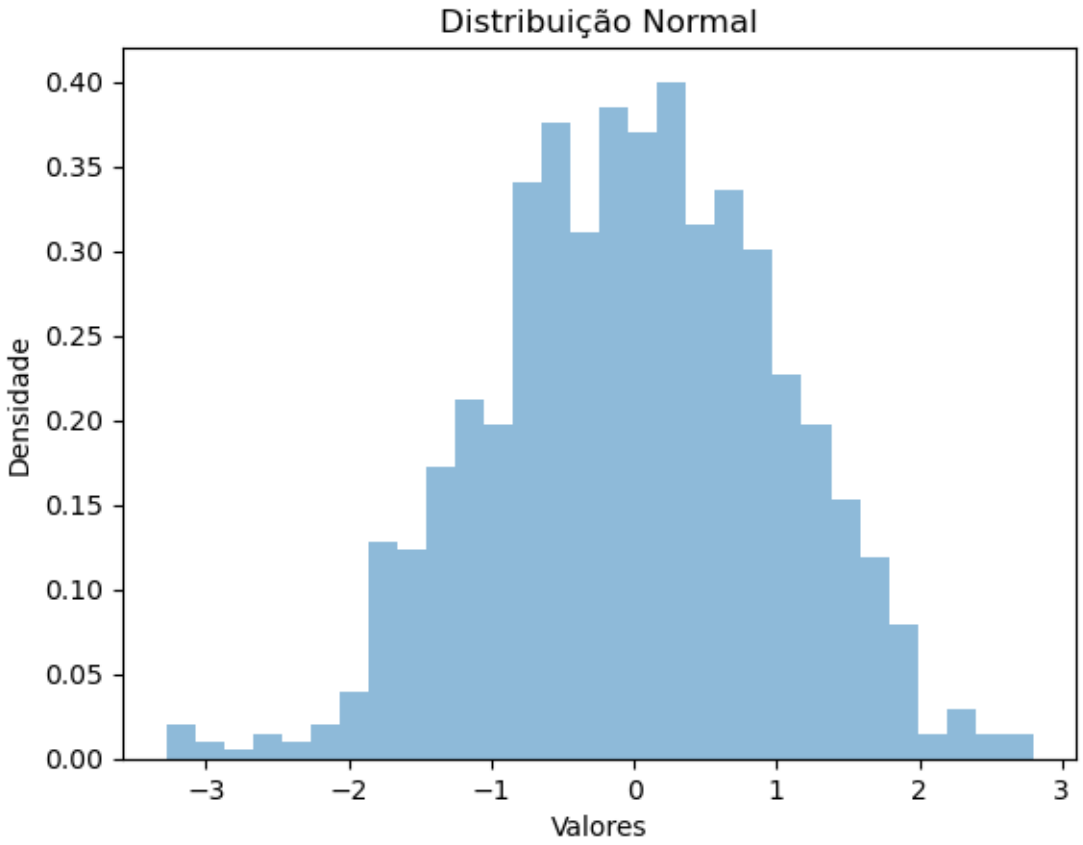
Distribuição Normal (Gaussiana) A distribuição normal, também conhecida como distribuição gaussiana, é uma das distribuições mais importantes e amplamente utilizadas na teoria das probabilidades e estatística. Ela descreve muitos fenômenos naturais e possui uma forma de sino simétrica. A função densidade de probabilidade (FDP) de uma variável aleatória com distribuição normal é dada por:

f(x) = (1 / (* sqrt(2))) * exp(-(x -)^2 / (2 ^2))

onde é a média da distribuição e é o desvio padrão.

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt

# Gerar dados a partir de uma distribuição normal
mu = 0
sigma = 1
n_samples = 1000
data = np.random.normal(mu, sigma, n_samples)

# Plotar histograma dos dados
plt.hist(data, bins=30, density=True, alpha=0.5)
plt.xlabel('Valores')
```

```
plt.ylabel('Densidade')
plt.title('Distribuição Normal')
plt.show()
```



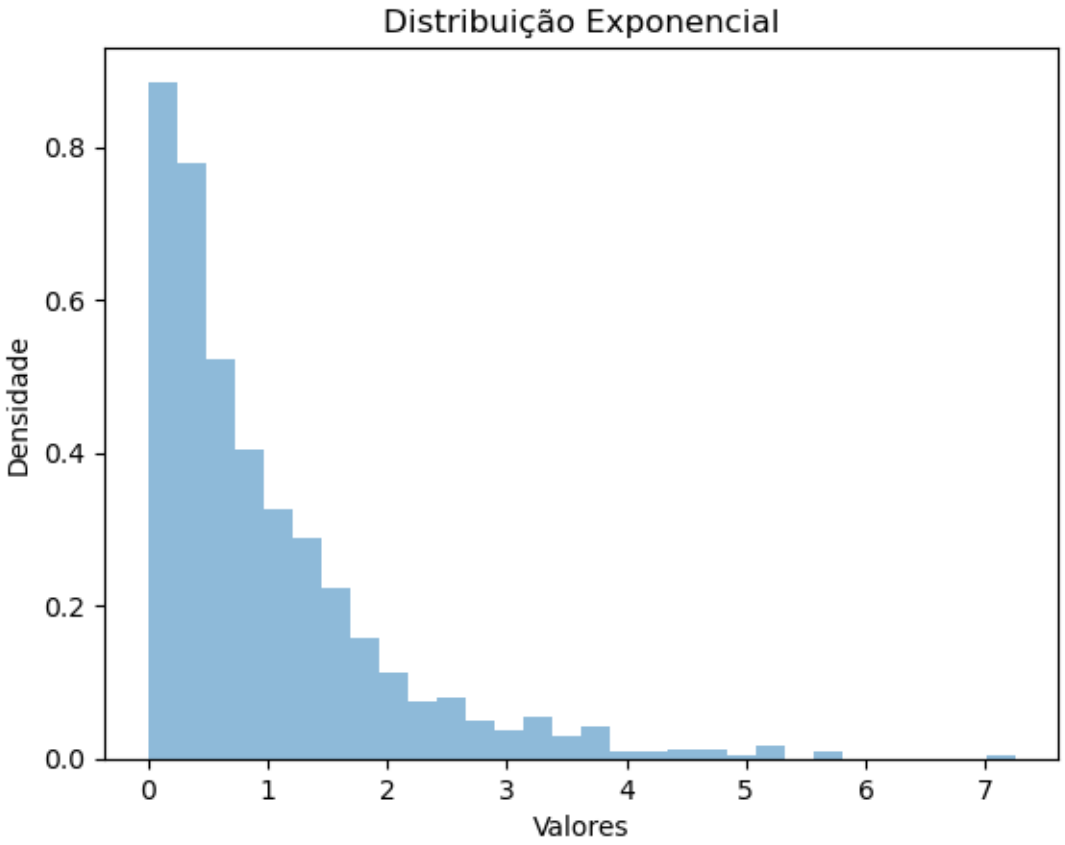
Distribuição Exponencial A distribuição exponencial descreve o tempo entre eventos em um processo de Poisson, onde os eventos ocorrem independentemente com uma taxa média . A função densidade de probabilidade (FDP) de uma variável aleatória com distribuição exponencial é dada por:

f(x) = * exp(- x)

onde x é o tempo entre eventos e é a taxa média de ocorrência.

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt

# Gerar dados a partir de uma distribuição normal
mu = 1
n_samples = 1000
data = np.random.exponential(mu, n_samples)
```

```
# Plotar histograma dos dados
plt.hist(data, bins=30, density=True, alpha=0.5)
plt.xlabel('Valores')
plt.ylabel('Densidade')
plt.title('Distribuição Exponencial')
plt.show()
```



0.2 Introdução à Estatística

A estatística é uma disciplina que envolve a coleta, análise, interpretação e apresentação de dados. Ela fornece métodos e técnicas para descrever e inferir informações sobre uma população com base em uma amostra observada.

0.2.1 População e Amostra

Na estatística, trabalhamos com duas principais unidades de estudo: população e amostra. A população é o conjunto completo de elementos ou indivíduos que queremos estudar, enquanto a amostra é uma parte representativa da população que é selecionada para análise.

0.2.2 Estatística Descritiva

A estatística descritiva envolve a organização, resumo e interpretação dos dados observados. Alguns conceitos importantes na estatística descritiva incluem:

Média A média é uma medida de tendência central que representa o valor médio de um conjunto de dados. Para uma amostra, a média é denotada por x-barra ($\bar{x}$), enquanto que para uma população, é denotada por (mu). A fórmula para calcular a média amostral é:

$\bar{x} = (x + x + \ldots + x) / n$

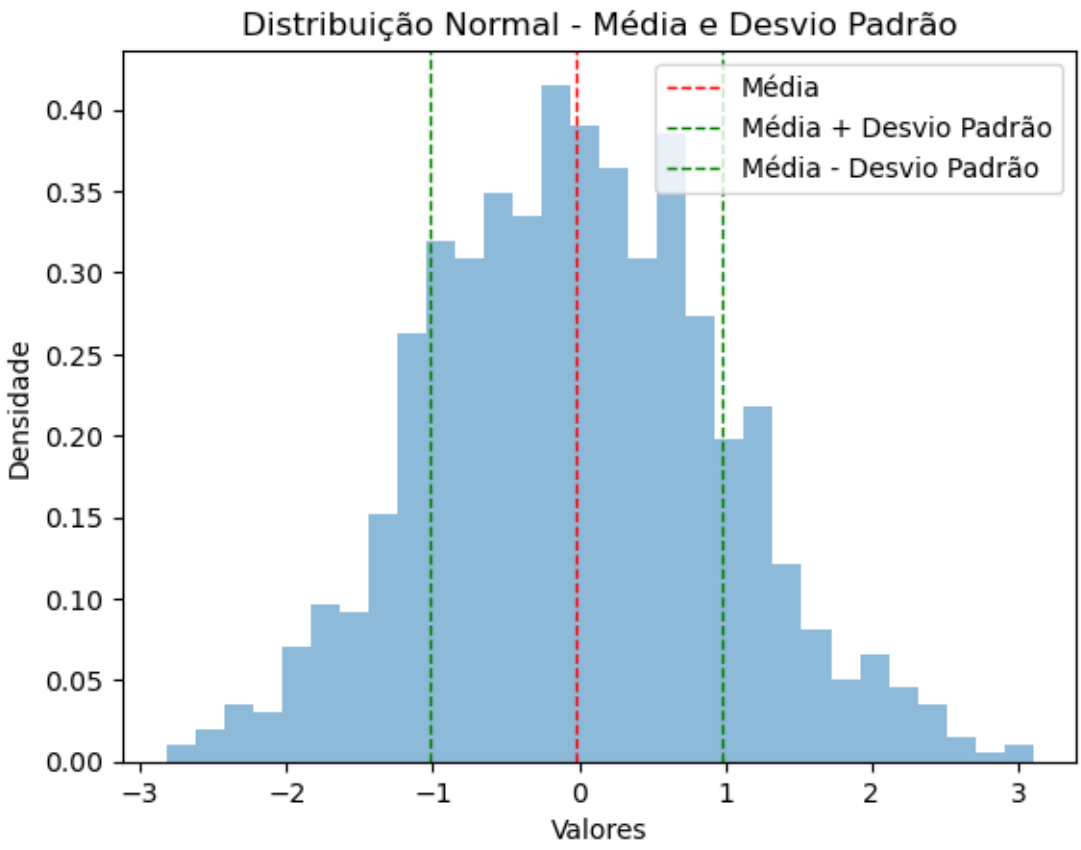
onde x , x , ..., x são os valores observados e n é o tamanho da amostra.

Variância e Desvio Padrão A variância mede a dispersão dos dados em relação à média. O desvio padrão é a raiz quadrada da variância e também é uma medida de dispersão. Para uma amostra, a variância é denotada por s^2, enquanto que para uma população, é denotada por ^{2}. A fórmula para calcular a variância amostral é:

$s^2 = \Sigma(x - \bar{x})^2 / (n - 1)$

onde x são os valores observados, $\bar{x}$ é a média amostral e n é o tamanho da amostra.

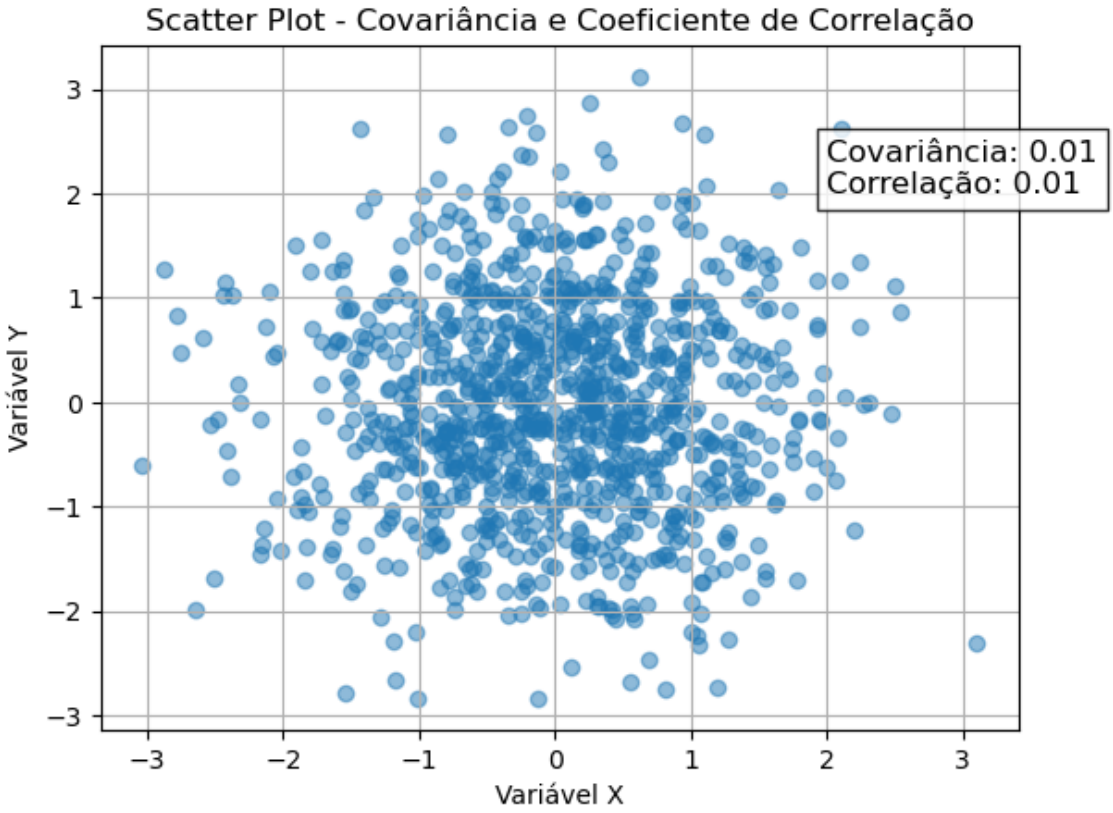
[ ]:
```
import numpy as np
import matplotlib.pyplot as plt

# Gerar dados aleatórios com média 0 e desvio padrão 1
data = np.random.normal(0, 1, 1000)

# Calcular a média e o desvio padrão
mean = np.mean(data)
std = np.std(data)

# Plotar histograma dos dados
plt.hist(data, bins=30, density=True, alpha=0.5)
plt.xlabel('Valores')
plt.ylabel('Densidade')
plt.title('Distribuição Normal - Média e Desvio Padrão')
plt.axvline(mean, color='r', linestyle='dashed', linewidth=1, label='Média')
plt.axvline(mean + std, color='g', linestyle='dashed', linewidth=1,␣
 ↪label='Média + Desvio Padrão')
plt.axvline(mean - std, color='g', linestyle='dashed', linewidth=1,␣
 ↪label='Média - Desvio Padrão')
plt.legend()
plt.show()
```



Covariância e Coeficiente de Correlação A covariância mede o grau de interdependência linear entre duas variáveis. A covariância amostral é denotada por sxy para uma amostra e a covariância populacional é denotada por xy para uma população. O coeficiente de correlação é uma medida padronizada da covariância e varia de -1 a 1. Uma correlação próxima de 1 indica uma forte relação positiva, enquanto uma correlação próxima de -1 indica uma forte relação negativa. O coeficiente de correlação amostral é denotado por r para uma amostra e o coeficiente de correlação populacional é denotado por para uma população.

[ ]:
```python
import numpy as np
import matplotlib.pyplot as plt

# Gerar dados aleatórios para duas variáveis
x = np.random.normal(0, 1, 1000)
y = np.random.normal(0, 1, 1000)

# Calcular a covariância e o coeficiente de correlação
covariance = np.cov(x, y)[0, 1]
correlation = np.corrcoef(x, y)[0, 1]
```

```
# Plotar scatter plot dos dados
plt.scatter(x, y, alpha=0.5)
plt.xlabel('Variável X')
plt.ylabel('Variável Y')
plt.title('Scatter Plot - Covariância e Coeficiente de Correlação')
plt.text(2, 2, f'Covariância: {covariance:.2f}\nCorrelação: {correlation:.2f}',␣
↪fontsize=12, bbox={'facecolor': 'white', 'alpha': 0.8})
plt.grid(True)
plt.show()
```



0.2.3 Estatística Inferencial

Estimadores Um estimador é uma função ou estatística calculada a partir dos dados da amostra e usada para estimar um parâmetro desconhecido da população. Um estimador pontual fornece uma única estimativa do parâmetro.

Alguns estimadores comuns incluem:

- **Média Amostral**: O estimador da média populacional é a média amostral, $\bar{x}$.
- **Variância Amostral**: O estimador da variância populacional é a variância amostral, s^2.
- **Covariância Amostral**: O estimador da covariância populacional é a covariância amostral, sxy.

Um estimador é avaliado quanto às suas propriedades desejáveis, como viés (quão próximo ele está do valor verdadeiro do parâmetro), consistência (se converge para o valor verdadeiro à medida que o tamanho da amostra aumenta) e eficiência (quão precisamente ele estima o parâmetro).

Estimadores de Máxima Verossimilhança Os estimadores de máxima verossimilhança são obtidos maximizando a função de verossimilhança, que mede a probabilidade de obter os dados observados para diferentes valores do parâmetro desconhecido. Esses estimadores são amplamente utilizados devido às suas propriedades estatísticas favoráveis.

Dado um conjunto de dados observados x , x , ..., x , assumindo que as observações são independentes e identicamente distribuídas (i.i.d.) de acordo com uma distribuição de probabilidade parametrizada por , a função de verossimilhança L( ) é definida como o produto das funções de densidade de probabilidade (f(x ;)) correspondentes a cada observação:

L( ) = f(x ;) * f(x ;) * ... * f(x ;)

A ideia é encontrar o valor do parâmetro que maximiza a função de verossimilhança, ou seja, o valor que torna os dados observados mais prováveis de acordo com a distribuição especificada por .

Em muitos casos, é mais conveniente trabalhar com o logaritmo natural da função de verossimilhança (log-verossimilhança), que simplifica os cálculos e não altera a posição do máximo. Portanto, a log-verossimilhança é dada por:

log L( ) = log f(x ;) + log f(x ;) + ... + log f(x ;)

A estimação de máxima verossimilhança (EMV) envolve encontrar o valor de que maximiza a log-verossimilhança. Isso pode ser feito através de técnicas de otimização, como o método do gradiente ou métodos iterativos como o algoritmo de Newton-Raphson.

0.3 Espaço Vetorial de Variáveis Aleatórias

No contexto da estatística, podemos considerar as variáveis aleatórias como vetores em um espaço vetorial. Isso nos permite explorar propriedades e operações matemáticas com as variáveis aleatórias de forma mais estruturada. Neste espaço, as variáveis aleatórias são tratadas como objetos matemáticos e podem ser manipuladas usando as regras do álgebra linear.

0.3.1 Variáveis Aleatórias como Vetores

Uma variável aleatória pode ser vista como um vetor em um espaço vetorial. O espaço vetorial das variáveis aleatórias possui duas principais operações: adição e multiplicação por um escalar.

- **Adição**: A adição de variáveis aleatórias é realizada componente por componente. Dadas duas variáveis aleatórias X e Y, a soma das variáveis aleatórias é dada por:

  ```
  (X + Y)(x) = X(x) + Y(x)
  ```

  onde x é um valor específico da variável aleatória.

- **Multiplicação por Escalar**: A multiplicação de uma variável aleatória por um escalar é feita multiplicando cada componente pela constante. Dada uma variável aleatória X e um escalar c, a multiplicação da variável aleatória por um escalar é dada por:

  ```
  (cX)(x) = c * X(x)
  ```

onde x é um valor específico da variável aleatória.

0.3.2 Vetor Aleatório

Em estatística, um vetor aleatório é uma generalização de uma variável aleatória para múltiplas dimensões. Formalmente, um vetor aleatório é uma função que mapeia um espaço amostral em um espaço vetorial. Um vetor aleatório pode ser representado por uma matriz, onde cada linha ou coluna representa uma variável aleatória.

Definição Matemática Seja X um vetor aleatório com dimensões d x 1, onde d é o número de variáveis aleatórias no vetor. Podemos definir um vetor aleatório como uma função que associa um vetor numérico a cada ponto amostral do espaço amostral Ω:

$$X(\omega) = [X_1(\omega), X_2(\omega), ..., X_d(\omega)]^T$$

onde X ( ) é a i-ésima variável aleatória no vetor e é um ponto no espaço amostral.

Distribuição Normal Vetorial A distribuição normal multivariada (ou distribuição normal vetorial) é uma das distribuições mais importantes para vetores aleatórios. Uma distribuição normal vetorial é caracterizada por sua média vetorial e sua matriz de covariância.

Seja X um vetor aleatório d-dimensional com média e matriz de covariância Σ. A distribuição normal vetorial é denotada por:

$$X \sim N(\mu, \Sigma)$$

A função de densidade de probabilidade (PDF) da distribuição normal vetorial é dada por:

$$f(x; \mu, \Sigma) = (2\pi)^(- d/2) * |\Sigma|^(- 1/2) * exp(-0.5 * (x - \mu)^T * \Sigma^(- 1) * (x - \mu))$$

onde x é o valor do vetor aleatório, é a média vetorial, Σ é a matriz de covariância, e |Σ| representa o determinante de Σ.

Demonstração da Distribuição Normal Vetorial A distribuição normal vetorial pode ser provada usando a função característica e a técnica de transformação linear. Aqui, forneceremos um esboço da prova.

Dado um vetor onde cada componente é uma variável aleatória X ~ N(0, 1), a pdf da v.a é dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{1}$$

Logo, se todas as componentes são v.a I.I.D da mesma distribuição a distribuição conjunta delas é

$$f_{\mathbf{Z}}(\mathbf{z}) = f_{Z_1,Z_2,...,Z_n}(z_1, z_2, ..., z_n) \tag{2}$$

$$= \prod_{i=1}^{n} f_{Z_i}(z_i) \tag{3}$$

$$= \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n} z_i^2\right\} \tag{4}$$

$$= \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left\{-\frac{1}{2}\mathbf{z}^T\mathbf{z}\right\}. \tag{5}$$

Agora precisamos genarilizar essa formula. Considere um vetor aleatório X com média m e matriz de covariância C. Denotamos $X \sim N(m, C)$. Supomos ainda que C seja uma matriz definida positiva. Essa suposição não limita a generalidade, pois sabemos que C é positiva semidefinida (Teorema 6.2), portanto, $\det(C) \geq 0$. Também sabemos que C é definida positiva se e somente se $\det(C) > 0$ (Teorema 6.3). Aqui, estamos excluindo o caso $\det(C) = 0$, pois podemos mostrar que podemos escrever algumas variáveis X_i como combinação linear de outras, portanto, podemos removê-las do vetor sem perder informações.

Da álgebra linear, sabemos que existe uma matriz Q de dimensão $n \times n$ tal que: $Q \cdot Q^T = I$ (I é a matriz identidade) $C = Q \cdot D \cdot Q^T$,

onde D é uma matriz diagonal: $D = \begin{bmatrix} d_{11} & 0 & ... & 0 \\ 0 & d_{22} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & d_{nn} \end{bmatrix}$,

e os d_{ii} são todos positivos. Definimos ainda: $D_{12} = \begin{bmatrix} \sqrt{d_{11}} & 0 & ... & 0 \\ 0 & \sqrt{d_{22}} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \sqrt{d_{nn}} \end{bmatrix}$,

de modo que $D_{12} \cdot D_{12} = D$ e $D_{12} = D_{12}^T$. Definimos também: $A = Q \cdot D_{12} \cdot Q^T$.

Dessa forma, $A \cdot A^T = A^T \cdot A = C$.

Agora estamos prontos para definir a transformação que converte um vetor gaussiano padrão em $X \sim N(m, C)$. Seja Z um vetor gaussiano padrão, ou seja, $Z \sim N(0, I)$. Definimos: $X = A \cdot Z + m$.

Afirmamos que $X \sim N(m, C)$. Para ver isso, observe primeiramente que X é um vetor aleatório normal. A razão é que qualquer combinação linear dos componentes de X é, na verdade, uma combinação linear dos componentes de Z mais uma constante. Assim, toda combinação linear dos componentes de X é uma variável aleatória normal. Resta mostrar que $\mathbb{E}[X] = m$ e $\mathbb{C}\mathbb{o}\mathbb{v}[X] = C$. Primeiro, observe que pela linearidade da expectativa temos: $\mathbb{E}[X] = \mathbb{E}[A \cdot Z + m] = A \cdot \mathbb{E}[Z] + m = m$.

Além disso, pela Exemplo 6.12, temos: $\mathbb{C}\mathbb{o}\mathbb{v}[X] = A \cdot \mathbb{C}\mathbb{o}\mathbb{v}[Z] \cdot A^T = A \cdot I \cdot A^T = A \cdot A^T = C$ (já que $\mathbb{C}\mathbb{o}\mathbb{v}[Z] = I$).

Portanto, mostramos que X é um vetor aleatório com média m e matriz de covariância C. Agora podemos usar sua inversa para encontrar a função densidade de probabilidade (PDF) de X. Temos: $f_X(x) = \frac{1}{|\det(A)|} \cdot f_Z(A^{-1}(x - m)) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(C)}} \cdot \exp\{-\frac{1}{2}(x - m)^T \cdot C^{-1} \cdot (x - m)\}$.

Para um vetor aleatório normal X com média m e matriz de covariância C, a PDF é dada por: $f_X(x) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(C)}} \cdot \exp\{-\frac{1}{2}(x-m)^T\}$

0.3.3 Demonstração Visual

Vamos ilustrar as operações de adição e multiplicação por escalar utilizando distribuições de probabilidade comuns. Usaremos as bibliotecas NumPy e Matplotlib para a geração e visualização dos gráficos.

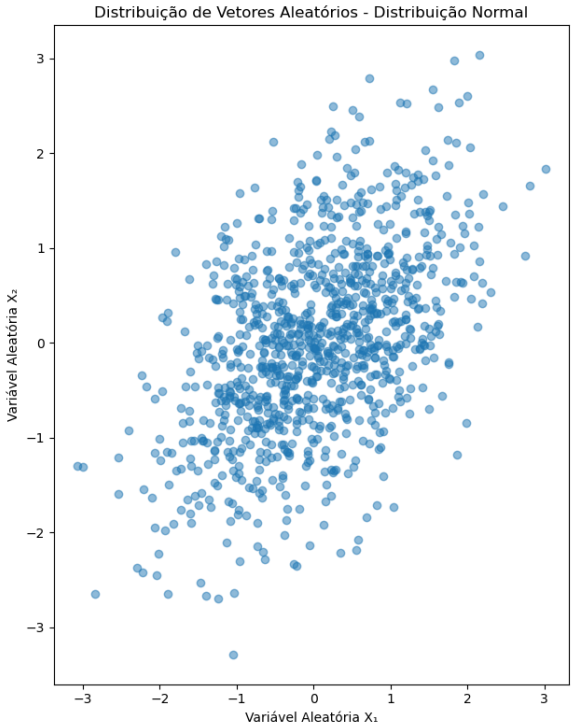
Primeiro, vamos gerar uma variável aleatória X a partir de uma distribuição normal com média 0 e desvio padrão 1:

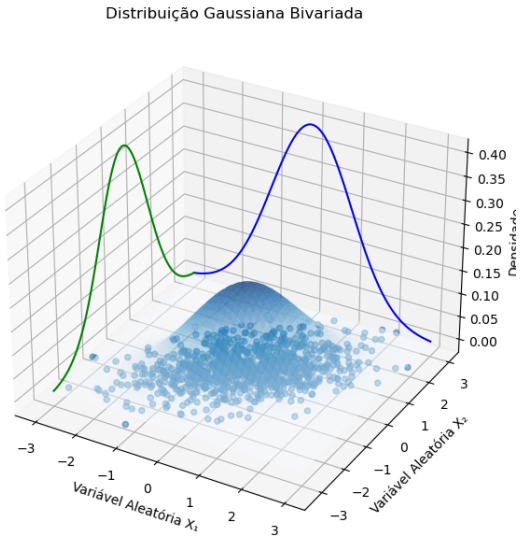
[ ]:
```
import numpy as np
import matplotlib.pyplot as plt

# Número de vetores aleatórios a serem gerados
n_vectors = 1000

# Gerar matriz de variáveis aleatórias com distribuição normal
mu = 0
sigma = 1
X = np.random.normal(mu, sigma, size=(2, n_vectors))
```

Agora, vamos plotar os vetores aleatórios no espaço R^2:

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal
from scipy.stats import norm

# Set random seed for reproducibility
np.random.seed(0)

# Generate synthetic data
mean = [0, 0]
cov = [[1, 0.5], [0.5, 1]]
X = np.random.multivariate_normal(mean, cov, size=1000)

# Plotting
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(1,2,2, projection='3d')

# Scatter plot on the floor
ax.scatter(X[:, 0], X[:, 1], np.zeros_like(X[:, 0]), alpha=0.5)
ax.set_xlabel('Variável Aleatória X ')
ax.set_ylabel('Variável Aleatória X ')
ax.set_zlabel('Piso')
ax.set_title('Distribuição de Vetores Aleatórios - Distribuição Normal (Piso)')
```

```
# 3D Gaussian contours on the walls
x = np.linspace(-3, 3, 100)
y = np.linspace(-3, 3, 100)
X_axis, Y_axis = np.meshgrid(x, y)
Z = multivariate_normal.pdf(np.column_stack((X_axis.flatten(), Y_axis.
 ↪flatten())), mean, cov)
Z = Z.reshape(X_axis.shape)
ax.plot_surface(X_axis, Y_axis, Z, alpha=0.5, cmap='Blues', edgecolor='none')

# X1 PDF along the X-axis
x1 = np.linspace(-3, 3, 100)
y = np.ones_like(x1)*3
z1 = norm.pdf(x1, mean[0], np.sqrt(cov[0][0]))
ax.plot(x1, y, z1, color='b')

# X2 PDF along the Y-axis
x = np.ones_like(x1)*-3
y2 = np.linspace(-3, 3, 100)
z2 = norm.pdf(y2, mean[1], np.sqrt(cov[1][1]))
ax.plot(x, y2, z2, color='g')

# Set labels and title
ax.set_xlabel('Variável Aleatória X ')
ax.set_ylabel('Variável Aleatória X ')
ax.set_zlabel('Densidade')
ax.set_title('Distribuição Gaussiana Bivariada')

#Only the scatter
ax2 = fig.add_subplot(1,2,1)
# Scatter plot on the floor
ax2.scatter(X[:, 0], X[:, 1], alpha=0.5)
ax2.set_xlabel('Variável Aleatória X ')
ax2.set_ylabel('Variável Aleatória X ')
ax2.set_title('Distribuição de Vetores Aleatórios - Distribuição Normal')

plt.tight_layout()
plt.show()
```

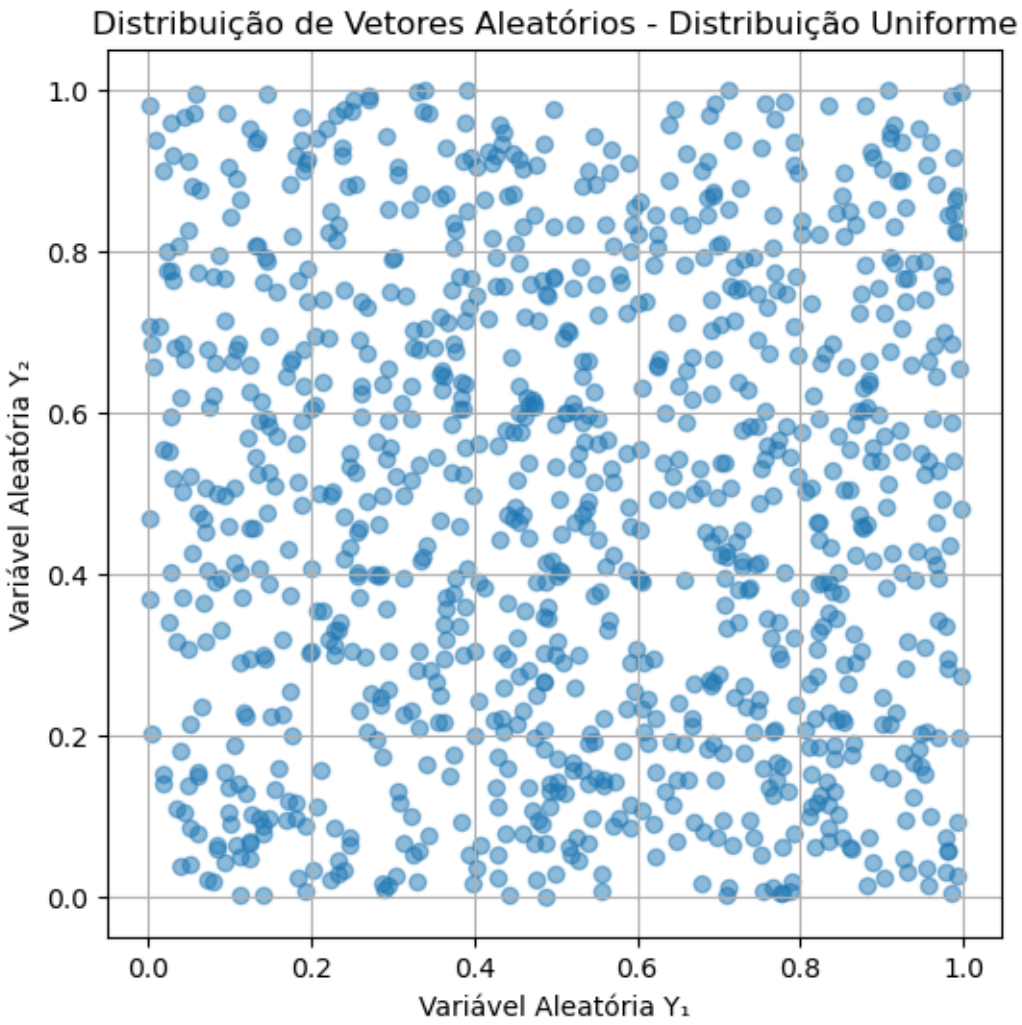




Esse código irá gerar uma visualização com vários vetores aleatórios no espaço R^2, onde cada coordenada representa o valor de uma variável aleatória. A dispersão dos vetores demonstra como a distribuição normal se parece no espaço vetorial de variáveis aleatórias.

Agora, vamos repetir o processo utilizando a distribuição uniforme:

[ ]:
```
# Gerar matriz de variáveis aleatórias com distribuição uniforme
low = 0
high = 1
Y = np.random.uniform(low, high, size=(2, n_vectors))

# Plotar vetores aleatórios
plt.figure(figsize=(6, 6))
plt.scatter(Y[0, :], Y[1, :], alpha=0.5)
plt.xlabel('Variável Aleatória Y ')
plt.ylabel('Variável Aleatória Y ')
plt.title('Distribuição de Vetores Aleatórios - Distribuição Uniforme')
plt.grid(True)
plt.show()
```



0.4 PCA e SVD: Análise de Componentes Principais e Decomposição em Valores Singulares

0.4.1 PCA (Análise de Componentes Principais)

PCA é uma técnica que busca encontrar uma representação de baixa dimensionalidade dos dados originais, capturando a maior quantidade possível de variabilidade dos dados. Isso é alcançado através da identificação dos componentes principais, que são combinações lineares das variáveis originais.

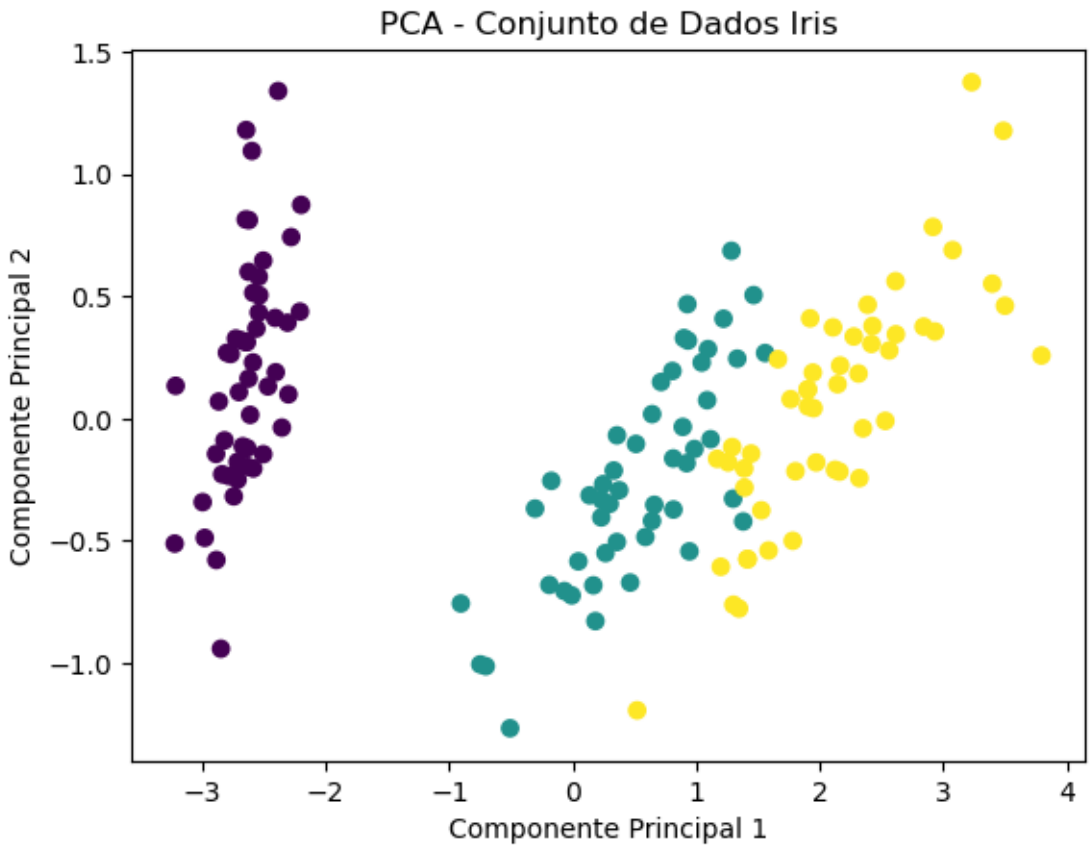
Formulação Matemática Dado um conjunto de dados X com n amostras e p variáveis, podemos realizar a PCA da seguinte forma:

1. Centralize os dados: Calcule a média de cada variável e subtraia a média de cada amostra.
2. Calcule a matriz de covariância: A matriz de covariância captura as relações entre as variáveis originais.

3. Calcule os autovetores e autovalores: Aplique a decomposição espectral na matriz de covariância para obter os autovetores e autovalores.
4. Escolha os componentes principais: Selecione os autovetores com os maiores autovalores como componentes principais.
5. Projete os dados nos componentes principais: Multiplique a matriz centralizada dos dados pelos autovetores selecionados para obter a projeção dos dados em um novo espaço.

Demonstração Visual Vamos realizar uma demonstração visual da PCA usando o conjunto de dados Iris. Usaremos a biblioteca scikit-learn para carregar o conjunto de dados e realizar a PCA.

[ ]:
```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA

# Carregar o conjunto de dados Iris
iris = load_iris()
X = iris.data
y = iris.target

# Aplicar a PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

# Plotar os dados projetados nos dois primeiros componentes principais
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y)
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2')
plt.title('PCA - Conjunto de Dados Iris')
plt.show()
```



0.4.2 SVD (Decomposição em Valores Singulares)

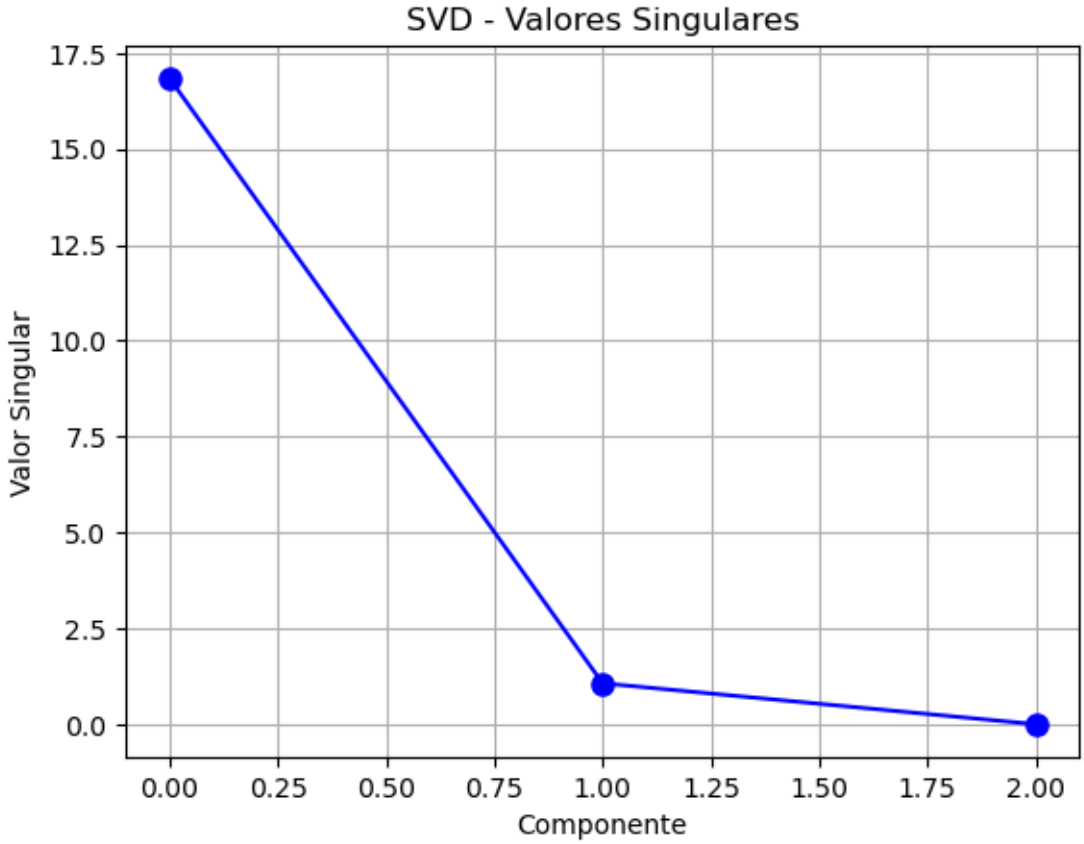
SVD é uma técnica que permite a decomposição de uma matriz em três componentes: uma matriz de vetores singulares esquerda, uma matriz de valores singulares e uma matriz de vetores singulares direita. Essa decomposição é útil para entender a estrutura dos dados e realizar operações como reconstrução, redução de dimensionalidade e filtragem de ruído.

Dada uma matriz X de dimensões m × n, a SVD é dada por:

X = U * S * V^T

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt

# Gerar uma matriz de exemplo
X = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

# Calcular a SVD
U, S, Vt = np.linalg.svd(X)

# Plotar os valores singulares
```

```
plt.plot(S, 'bo-', markersize=8)
plt.xlabel('Componente')
plt.ylabel('Valor Singular')
plt.title('SVD - Valores Singulares')
plt.grid(True)
plt.show()
```



0.4.3 Relação entre PCA e SVD

PCA (Principal Component Analysis) e SVD (Singular Value Decomposition) estão intimamente relacionados e compartilham conceitos fundamentais. De fato, a PCA pode ser vista como uma aplicação especial da SVD em uma matriz de covariância.

PCA como uma Aplicação da SVD Dada uma matriz de dados X de dimensões n x p, onde n é o número de amostras e p é o número de variáveis, a PCA busca encontrar uma base ortogonal de p componentes principais que capture a maior variabilidade dos dados. Essa base de componentes principais é composta pelos autovetores da matriz de covariância dos dados.

A matriz de covariância dos dados é dada por:

$ C = (1/n) * X^T * X $

onde X^T é a matriz transposta de X.

Podemos então realizar a decomposição espectral na matriz de covariância C para obter os autovetores (V) e autovalores ( ). Os autovetores representam a direção dos componentes principais, enquanto os autovalores indicam a importância dessas direções.

A matriz de projeção dos dados nos componentes principais, Z, é obtida multiplicando a matriz de dados X pelos autovetores selecionados:

$ Z = X * V $

Essa matriz Z representa os dados projetados nos componentes principais. Portanto, podemos ver que a PCA pode ser interpretada como uma aplicação da SVD na matriz de covariância.

Relação Matemática entre PCA e SVD Podemos estabelecer uma relação matemática entre a PCA e a SVD ao comparar as equações da PCA e da SVD.

Na PCA, a matriz de dados X pode ser decomposta como:

$ X = Z * V^T $

onde Z é a matriz dos dados projetados nos componentes principais e V^T é a matriz transposta dos autovetores selecionados.

Por outro lado, a SVD da matriz de dados X é dada por:

$ X = U * S * V^T $

onde U é a matriz dos autovetores esquerda, S é a matriz diagonal dos valores singulares e V^T é a matriz transposta dos autovetores direita.

Ao comparar essas duas equações, podemos concluir que a matriz dos autovetores esquerda da SVD (U) é equivalente à matriz dos dados projetados nos componentes principais (Z) da PCA. Além disso, os valores singulares (S) da SVD correspondem aos autovalores ( ) da PCA.

Essa relação estabelece a equivalência conceitual entre PCA e SVD, mostrando que os autovetores e autovalores da PCA são obtidos através da SVD da matriz de covariância dos dados.

0.5 Regressão Linear e o estimador de mínimos quadrados

0.5.1 Introdução à Regressão Linear em Termos Estatísticos

A regressão linear é uma técnica estatística que visa modelar a relação entre uma variável dependente e uma ou mais variáveis independentes. Essa relação pode ser expressa através de uma equação linear. Em termos de álgebra linear, a regressão linear pode ser formulada como um problema de encontrar uma combinação linear ótima dos vetores de características para prever o valor da variável dependente.

O objetivo da regressão linear é encontrar os coeficientes que minimizam a soma dos erros quadrados entre os valores observados e os valores previstos pelo modelo linear. Esses coeficientes são estimados usando o método dos mínimos quadrados. No entanto, a regressão linear padrão não considera a multicolinearidade, que ocorre quando as variáveis independentes estão altamente correlacionadas entre si. Além disso, a regressão linear pode sofrer de overfitting quando há muitas variáveis independentes.

0.5.2 Função de Custo da Regressão Linear e Máxima Verossimilhança

A função de custo da regressão linear é derivada a partir da abordagem da máxima verossimilhança. A ideia básica é encontrar os coeficientes que maximizam a probabilidade de observar os valores observados dado o modelo linear. Supomos que os erros entre os valores observados e os valores previstos sigam uma distribuição normal com média zero e variância constante. Logo é fácil ver que $ E(Y) = ^T x$ pelas propriedades da esperança, logo a regressão pode ser encarado como a média da distribuição de y

Pelas propriedades vistas no capitulo sobre derivação de um vetor aleatório sabemos que se $Z \sim N(0,1)$ então $X = A*Z + m N(m,A) $, logo como $\epsilon \sim N(0,\sigma)$ então podemos pensar em y como uma v.a normal com média $ E(Y) = ^T x$ e variancia σ

Dado um conjunto de dados de treinamento composto por pares de valores (x, y), onde x é o vetor de características e y é o valor observado, a probabilidade de observar os valores y dado o modelo linear é dada pela função de densidade de probabilidade (PDF) da distribuição normal:

$$P(y|x,\theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\theta^T x)^2}{2\sigma^2}}$$

onde θ é o vetor de coeficientes e σ é o desvio padrão dos erros. A função de verossimilhança é o produto das probabilidades individuais de cada valor observado. Como é mais conveniente maximizar a função de verossimilhança, tomamos o logaritmo da função de verossimilhança para obter a função de log-verossimilhança:

$$L(\theta) = \prod_{i=1}^{n} P(y|x,\theta)$$

$$\log L(\theta) = -\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \theta^T x_i)^2$$

onde n é o número de observações. O objetivo é encontrar o vetor de coeficientes θ que maximiza a função de log-verossimilhança.

A função de custo da regressão linear é definida como o negativo da função de log-verossimilhança, multiplicado por -1 para transformar o problema de maximização em minimização:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta^T x_i)^2$$

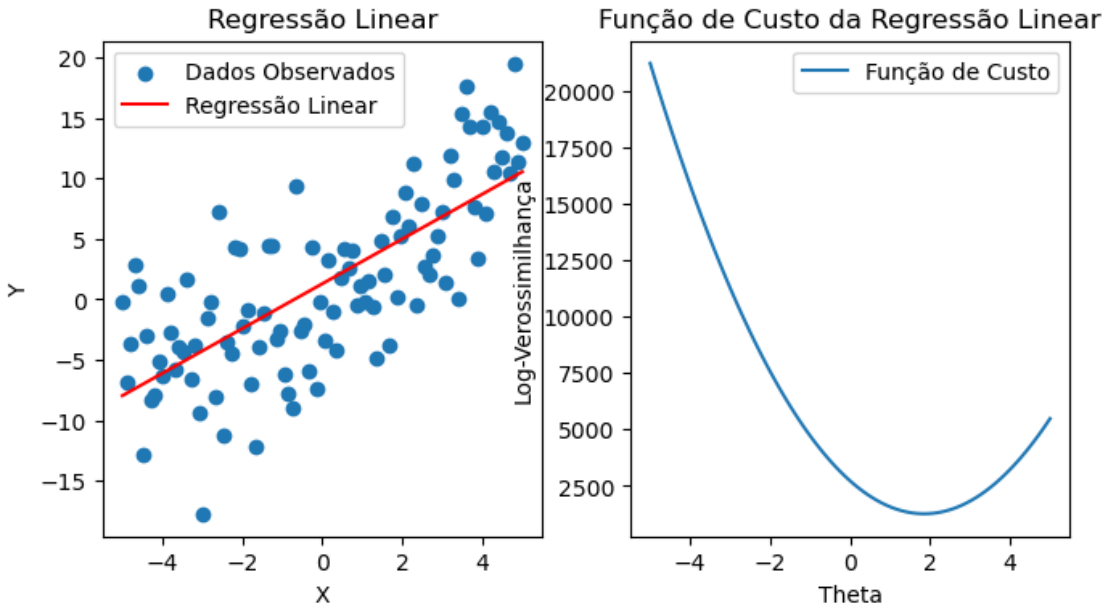
Essa é a função de custo que queremos minimizar para encontrar os coeficientes ótimos da regressão linear.

Note que em estatística fala-se de estimadores, ou seja, um valor dependente das observações que espera estimar o valor do parametro real. Nesse caso, o valor dos coeficientes reais seria θ, porém como desconhecemos esse valor buscamos aproximar θ por um estimador θ que é encontrado a partir da minimização da função de custo, esse estimador é chamado de estimador dos mínimos quadrados pois é derivado da forma dos mínimos quadrados da álgebra

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(0)

# Gerando dados sintéticos
X = np.linspace(-5, 5, 100)
noise = np.random.normal(0, 5, size=100)
Y = 2*X + 1 + noise

# Gerando valores previstos
Y_pred = 2*X + 1

# Calculando coeficientes da regressão linear
X_matrix = np.column_stack((np.ones_like(X), X))
coefficients_normal = np.linalg.inv(X_matrix.T @ X_matrix) @ X_matrix.T @ Y


# Gerando valores thetas para função custo
theta_values = np.linspace(-5, 5, 100)
cost_values = [(1/2)*np.sum((Y - theta*X - 1)**2) for theta in theta_values]

# Gerando valores lambda para regressão ridge
lambda_values = np.linspace(0, 100, 100)
coefficient_values = []
for lam in lambda_values:
    X_matrix = np.column_stack((np.ones_like(X), X))
    coefficient = np.linalg.inv(X_matrix.T @ X_matrix + lam*np.eye(2)) @␣
 ↪X_matrix.T @ Y
    coefficient_values.append(coefficient)

# Plotting
plt.figure(figsize=(12, 4))
Y_normal = coefficients_normal[0] + coefficients_normal[1]*X
plt.subplot(1, 3, 1)
plt.title("Regressão Linear")
plt.xlabel("X")
plt.ylabel("Y")
plt.scatter(X, Y, label="Dados Observados")
plt.plot(X, Y_normal, color='red', label="Regressão Linear")
plt.legend()

plt.subplot(1, 3, 2)
plt.title("Função de Custo da Regressão Linear")
plt.xlabel("Theta")
plt.ylabel("Log-Verossimilhança")
plt.plot(theta_values, cost_values, label="Função de Custo")
```

```
plt.legend()
```

```
[ ]: <matplotlib.legend.Legend at 0x24376fb3f70>
```



0.6 Regressão Ridge (Tikhonov Regularization)

A regressão ridge, também conhecida como regularização de Tikhonov, é uma extensão da regressão linear que introduz um termo de penalidade na função objetivo da regressão linear. O objetivo é minimizar a soma dos erros quadrados, ao mesmo tempo em que reduz o impacto da multicolinearidade e evita o overfitting.

Na regressão ridge, adicionamos um termo de penalidade que encolhe os coeficientes em direção a zero:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta^T x_i)^2 + \lambda \sum_{j=1}^{p} \theta_j^2$$

onde λ é o parâmetro de regularização e p é o número de características. O termo de penalidade $\lambda \sum_{j=1}^{p} \theta_j^2$ desencoraja coeficientes grandes, reduzindo efetivamente o impacto da multicolinearidade.

A solução da regressão ridge pode ser obtida através da derivação da função de custo em relação aos coeficientes θ e igualando a zero:

$$\nabla J(\theta) = X^T(X\theta - y) + \lambda\theta = 0$$

Simplificando a equação:

$$X^T X\theta + \lambda\theta = X^T y$$

Rearranjando os termos:

$$(X^T X + \lambda I)\theta = X^T y$$

Para obter a solução ótima para a regressão ridge, resolvemos para θ isolando-o no lado esquerdo da equação:

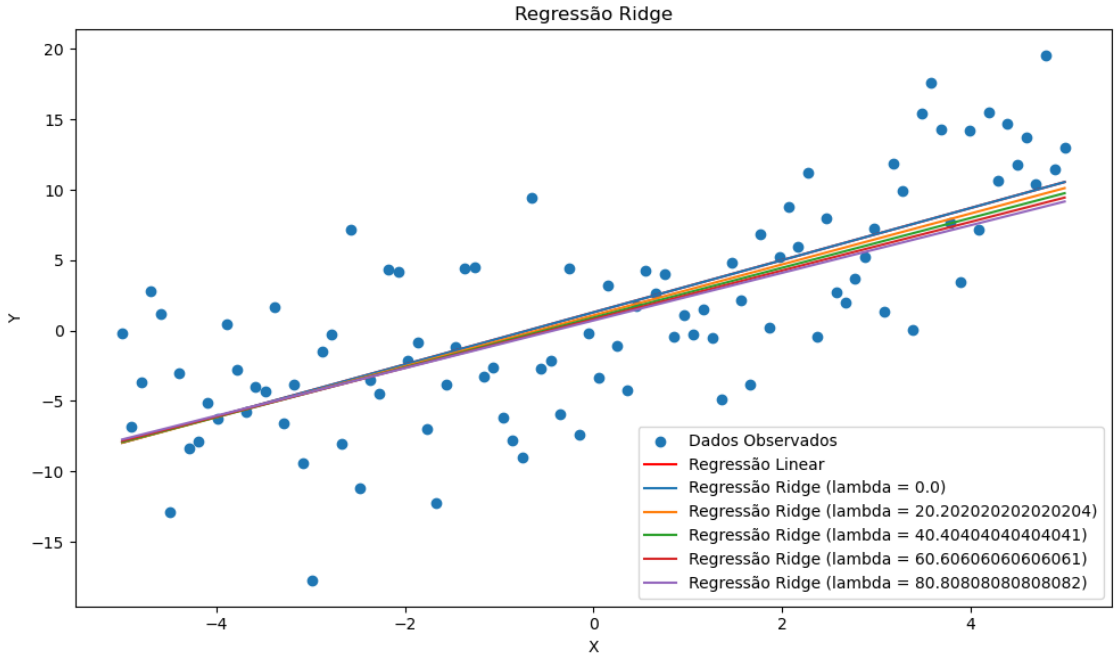
$$\theta = (X^T X + \lambda I)^{-1} X^T y$$

Esses são os coeficientes θ que minimizam a função de custo para a regressão ridge.

Na prática, em vez de calcular a inversa diretamente, podemos usar técnicas como a decomposição em valores singulares (SVD) ou a decomposição de Cholesky para calcular eficientemente a solução.

A regressão ridge ajuda a estabilizar o modelo e reduzir o impacto da multicolinearidade, adicionando uma penalidade aos coeficientes. O parâmetro de regularização λ controla a quantidade de encolhimento aplicada aos coeficientes. Um λ maior resulta em um encolhimento maior, reduzindo efetivamente a complexidade do modelo.

[ ]:
```
# Ridge regression
plt.figure(figsize=(10, 6))
plt.scatter(X, Y, label="Dados Observados")
plt.plot(X, Y_normal, color='red', label="Regressão Linear")
for i in range(0,100,20):
    Y_ridge = coefficient_values[i][0] + coefficient_values[i][1]*X
    plt.plot(X, Y_ridge, label=f"Regressão Ridge (lambda = {lambda_values[i]})")
plt.title("Regressão Ridge")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()


plt.tight_layout()
plt.show()
```



0.7 Referências

- Ross, S. M. (2019). A First Course in Probability (10th ed.). Pearson.
- Wackerly, D. D., Mendenhall III, W., & Scheaffer, R. L. (2014). Mathematical Statistics with Applications (7th ed.). Cengage Learning.
- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
- Golub, G. H., & Van Loan, C. F. (2012). Matrix Computations. JHU Press.
- Ross, S. M. (2019). A First Course in Probability (10th ed.). Pearson.
- https://www.probabilitycourse.com/chapter6/6_1_5_random_vectors.php
- Pfeiffer, P. (2020) Probability, Mathematical Statistics, and Stochastic Processes (Siegrist) Rice University.