# TESTE SHAPIRO-WILK (NORMALIDADE) POR ChatGPT

O teste de normalidade de Shapiro-Wilk é um teste estatístico que é utilizado para verificar se uma amostra de dados veio de uma distribuição normal. O teste é bastante preciso, mas exige um tamanho de amostra relativamente pequeno (geralmente n ≤ 2000). Aqui está um exemplo de como usar o teste de Shapiro-Wilk em Python:

## Código Shapiro-Wilk

from scipy.stats import shapiro

**# Gerar uma amostra de dados aleatórios de uma distribuição normal**

data = np.random.normal(0, 1, 1000)

**# Executar o teste de Shapiro-Wilk**

stat, p = shapiro(data)

**# Interpretar os resultados**

alpha = 0.05

if p > alpha:

print('Amostra parece ser normal (falha para rejeitar H0)')

else:

print('Amostra não parece ser normal (rejeitar H0)')

## Gráfico da distribuição normal

Neste exemplo, geramos uma amostra de dados aleatórios de uma distribuição normal e, em seguida, executamos o teste de Shapiro-Wilk usando a função shapiro da biblioteca SciPy. O valor-p retornado pelo teste é comparado com um nível de significância alpha pré-determinado (neste caso, alpha = 0,05). Se o valor-p for maior que alpha, não podemos rejeitar a hipótese nula de que a amostra veio de uma distribuição normal; caso contrário, rejeitamos a hipótese nula e concluímos que a amostra não veio de uma distribuição normal.

Um dos melhores gráficos para verificar se uma distribuição é normal é o histograma, juntamente com uma sobreposição da curva normal teórica. Em Python, você pode usar a biblioteca "matplotlib" para criar um histograma e a biblioteca "scipy" para gerar a curva normal.

Para criar um histograma, você pode usar a função "plt.hist()" do matplotlib e especificar os dados, o número de barras (bins) e outras opções, como rótulos de eixo e títulos.

Para gerar a curva normal, você pode usar a função "scipy.stats.norm.pdf()", que gera os valores da função de densidade de probabilidade normal para uma determinada média e desvio padrão.

Exemplo de como criar um histograma com curva normal em Python:

### Código

import matplotlib.pyplot as plt

import numpy as np

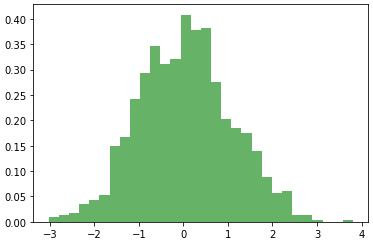
from scipy.stats import norm

**# Dados gerados aleatoriamente**

data = np.random.normal(0, 1, 1000)

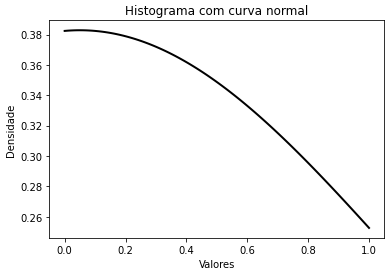
**# Criando o histograma**

plt.hist(data, bins=30, density=True, alpha=0.6, color='g')



**# Gerando a curva normal**

mu, std = norm.fit(data)



xmin, xmax = plt.xlim()

x = np.linspace(xmin, xmax, 100)

p = norm.pdf(x, mu, std)

plt.plot(x, p, 'k', linewidth=2)

plt.title("Histograma com curva normal")

plt.xlabel("Valores")

plt.ylabel("Densidade")

plt.show()

Este exemplo gera um histograma dos dados gerados aleatoriamente e sobrepõe a curva normal teórica. Ajustando os parâmetros de **mu(População ) e std(Desvio padrão)** para os dados, você pode comparar a distribuição dos dados com a curva normal e verificar se ela segue uma distribuição normal