

AAG01 (Parte 2)

Repetir a parte 1 com os mesmos parâmetros de entrada, mas utilizando funções prontas de bibliotecas Python para curvas empíricas.

Atenção:

- Diferente da parte 1, nesta etapa é necessário **plotar também a PDF** (além da CDF e PMF) com as funções prontas das bibliotecas.

Lendo os dados do dataset

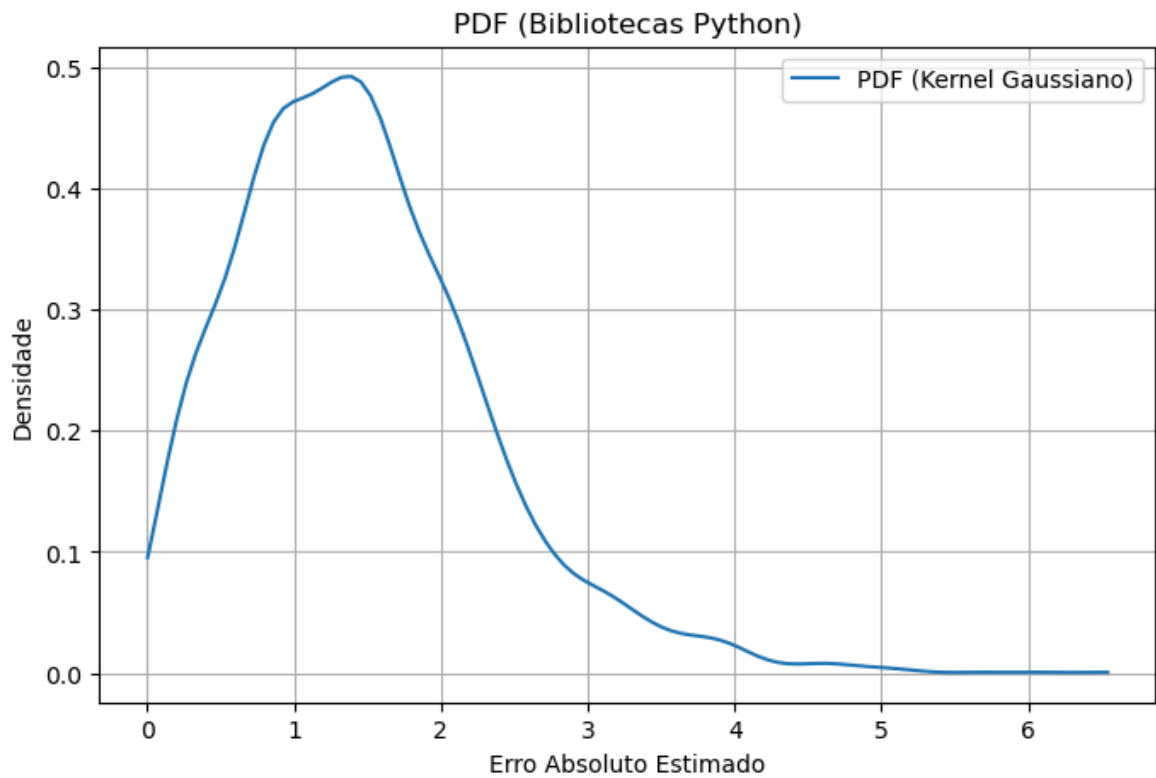
```
In [1]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.stats import gaussian_kde

# Preparando dataset
df = pd.read_csv("../DataSet/outdoor40.csv")
erros = df['absErrRaw']
```

Suavizando os dados utilizando o Kernel Gaussiano e plotando a PDF

```
In [2]: kde = gaussian_kde(erros)
x_vals = np.linspace(erros.min(), erros.max(), 100)
pdf_vals = kde(x_vals)

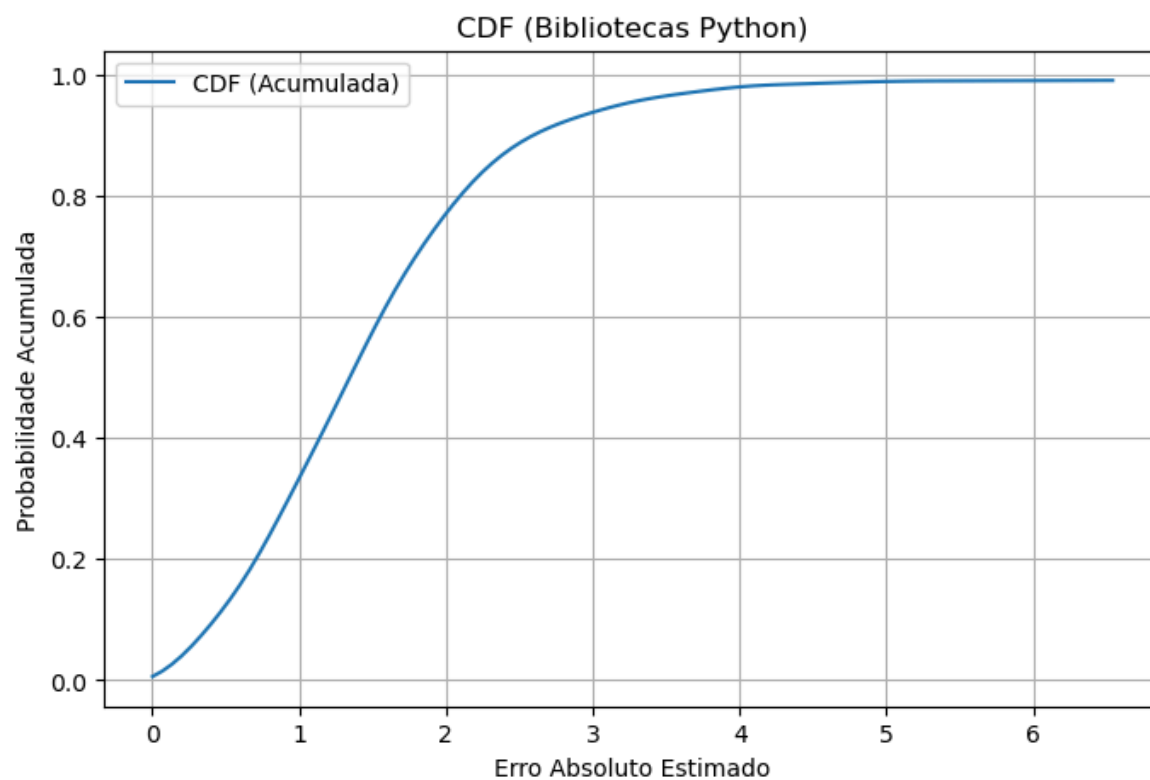
# Plotar PDF
plt.figure(figsize=(8, 5))
plt.plot(x_vals, pdf_vals, label="PDF (Kernel Gaussiano)")
plt.title("PDF (Bibliotecas Python)")
plt.xlabel("Erro Absoluto Estimado")
plt.ylabel("Densidade")
plt.grid(True)
plt.legend()
plt.show()
```



Calculando a CDF a partir da PDF e plotando

```
In [3]: # Calcular CDF com integração acumulada
cdf_vals = np.cumsum(pdf_vals) * (x_vals[1] - x_vals[0])

# Plotar CDF
plt.figure(figsize=(8, 5))
plt.plot(x_vals, cdf_vals, label="CDF (Acumulada)")
plt.title("CDF (Bibliotecas Python)")
plt.xlabel("Erro Absoluto Estimado")
plt.ylabel("Probabilidade Acumulada")
plt.grid(True)
plt.legend()
plt.show()
```



In []: