GEIA 25.1- INTRODUÇÃO A CIÊNCIA DE DADOS

Aluno:

Dárcio Henrique Barbosa de Paiva (**2514877**)

Monitor:

Arthur Veras

Fortaleza
2025

Sumário

Importação dos dados

```python
import csv
import numpy as np

class ImportarDados():
    def __init__(self, caminho_arquivo):
        self.caminho = caminho_arquivo
        self.dados_numericos = []
        self.ids = []

    def importar(self):
        with open(self.caminho, newline='') as csvfile:
            reader = csv.DictReader(csvfile)
            for row in reader:
                self.ids.append(row['ID'])
                self.dados_numericos.append([
                    float(row['Idade']),
                    float(row['Nota_Matematica']),
                    float(row['Nota_Estatistica']),
                    float(row['Nota_Programacao']),
                    float(row['Faltas'])
                ])
        return np.array(self.dados_numericos), self.ids
```

Classe: ImportarDados()

A classe `ImportarDados` tem função carregar dados numéricos de um arquivo no formato `.csv`. É útil para transformar informações armazenadas em arquivos de texto estruturados em uma estrutura apropriada para análise, como arrays.

Atributos:

`self.caminho`: representa o caminho para o arquivo CSV.
`self.dados_numericos`: armazena a matriz de dados numéricos extraída do arquivo. Cada linha representa um registro, e cada coluna representa uma característica.
`self.ids`: lista que armazena os identificadores de cada linha do arquivo, geralmente usados para vincular os dados a uma entidade específica.

Análise Estatística

```python
class AnaliseEstatistica():
    def mean(array):
        mean = np.mean(array)
        return mean
```

```python
def generate_report(dados, ids, colunas):
    from outliers import DeteccaoOutliers

    print("\n-- Relatório --")
    for i, nome_coluna in enumerate(colunas):
        col = dados[:, i]
        print(f"\n--- {nome_coluna} ---")
        print(f"Média: {AnaliseEstatistica.mean(col):.2f}")
        print(f"Moda: {AnaliseEstatistica.mode(col)}")
        print(f"Mediana: {AnaliseEstatistica.median(col):.2f}")
        print(f"Desvio Padrão: {AnaliseEstatistica.std(col):.2f}")
        print(f"Variância: {AnaliseEstatistica.variance(col):.2f}")
        print(f"Intervalo: {AnaliseEstatistica.interval(col):.2f}")

        z_vals, z_outs = DeteccaoOutliers.zscore(col)
        (_, _), tukey_outs = DeteccaoOutliers.cerca_tukey(col)

        print(f"Outliers (Z-Score): {len(z_outs)} → {[ids[i] for i in z_outs]}")
        print(f"Outliers (Tukey): {len(tukey_outs)} → {[ids[i] for i in tukey_outs]}")
```

Classe: AnaliseEstatistica()

A classe `AnaliseEstatistica` é conjunto de métodos estáticos destinados ao cálculo de estatísticas de vetores numéricos. Ideal para obter rapidamente medidas de tendência central, dispersão e identificar outliers nos dados.

Métodos:

`mean(array)`: calcula e retorna a média.
`mode(array)`: identifica e retorna a moda.
`median(array)`: retorna a mediana..
`std(array)`: calcula e retorna o desvio padrão.
`variance(array)`: retorna a variância.
`interval(array)`: calcula o intervalo dos dados.
`generate_report(dados, ids, colunas)`: gera um relatório estatístico completo para cada coluna fornecida nos dados. Este relatório é exibido no terminal e contém todas as medidas mencionadas acima.

Detecção de Outliers

```python
import numpy as np

class DeteccaoOutliers():
    @staticmethod
    def zscore(array):
        score = (array - np.mean(array)) / np.std(array)
        return score, np.where(abs(score) > 3)[0]
    @staticmethod
    def cerca_tukey(array):
        q1 = np.percentile(array, 25)
        q3 = np.percentile(array, 75)
        iqr = q3 - q1
        limite_inferior = q1 - 1.5 * iqr
        limite_superior = q3 + 1.5 * iqr
        indice = np.where((array < limite_inferior) | (array > limite_superior))[0]
        return (limite_inferior, limite_superior), indice
```

Classe: DeteccaoOutliers()

A classe `DeteccaoOutliers` é responsável por detectar valores extremos (outliers) dentro de um conjunto de dados. Ela contém dois métodos estáticos que implementam abordagens estatísticas distintas para essa finalidade.

Métodos:

`zscore(array)`: calcula o z para cada elemento do array, retornando tanto os valores dos escores quanto os índices dos elementos que possuem um absoluto maior que 3 (considerado-os outliers).

`cerca_tukey(array)`: aplica o método de Tukey, que calcula os quartis (`q1` e `q3`) e o intervalo interquartil (`iqr`). A partir disso, define limites superior e inferior para detecção de outliers e retorna os valores desses limites junto com os índices dos valores que estão fora da faixa esperada.

Main e o Terminal

```python
from importar import ImportarDados
from analise import AnaliseEstatistica

class Main():
    caminho = 'base.csv'
    importador = ImportarDados(caminho)
    dados, ids = importador.importar()

    colunas = ['Idade', 'Nota_Matematica', 'Nota_Estatistica', 'Nota_Programacao', 'Faltas']
    AnaliseEstatistica.generate_report(dados, ids, colunas)

if __name__ == "__Main__":
    Main()
```

Classe: Main()

A classe `Main` é responsável por executar o fluxo principal do programa. Ela atua como o ponto de entrada da aplicação e organiza a sequência das etapas da análise estatística.

Operações realizadas:

Define o caminho do arquivo CSV a ser analisado, neste caso `'base.csv'`.
Cria uma instância da classe `ImportarDados` e chama o método `importar()` para obter os dados e os IDs correspondentes.
Define uma lista com os nomes das colunas que representam as variáveis a serem analisadas.
Chama o método `generate_report` da classe `AnaliseEstatistica` para processar os dados e gerar um relatório, incluindo detecção de outliers.

Como operar o código

```
-- Relatório --

--- Idade ---
Média: 24.00
Moda: [19.0, 20.0, 21.0, 22.0, 23.0]
Mediana: 23.00
Desvio Padrão: 4.18
Variância: 17.50
Intervalo: 14.00
Outliers (Z-Score): 0 → []
Outliers (Tukey): 0 → []
```

Terminal: Relatório Impresso.

Antes de executar o código, certifique-se de que há um arquivo chamado `base.csv` na mesma pasta do programa. Esse arquivo deve conter uma tabela com os seguintes campos por linha (incluindo cabeçalho):
`ID`
`Idade`
`Nota_Matematica`
`Nota_Estatistica`
`Nota_Programacao`
`Faltas`
Esses campos devem conter valores numéricos (exceto o `ID`, que pode ser texto), pois serão convertidos e analisados estatisticamente.

Execução do programa

Após garantir que o arquivo está no lugar correto e no formato esperado, basta executar o arquivo principal do projeto, que contém a classe `Main` e então o código fará o seguinte:

Lê o arquivo `base.csv` por meio da classe `ImportarDados`.
Extrai os dados numéricos e os identificadores.
Processa as colunas usando a classe `AnaliseEstatistica`.
Calcula e imprime no terminal medidas estatísticas como média, moda, mediana, desvio padrão, variância e intervalo.
Detecta e exibe os outliers com os métodos: Z-Score e Cerca de Tukey (IQR).