

Projeto Final: Análise de Desempenho Acadêmico com Python

Albert E. F. Muritiba

12/02/2025

Projeto Final: Análise de Desempenho Acadêmico com Python

Objetivo Geral

Desenvolver um programa em Python para realizar a análise estatística de um conjunto de dados fictício de desempenho acadêmico. O objetivo é aplicar conceitos de programação aprendidos ao longo do semestre, incluindo manipulação de coleções, uso de funções, cálculos estatísticos e visualizações gráficas.

Restrições

Para fins didáticos, não será permitido o uso da biblioteca `pandas`, porque ela já fornece todas as funcionalidades necessárias para a análise de dados. O objetivo é que vocês apliquem os conceitos aprendidos em aula para realizar a leitura, estruturação e análise dos dados de forma manual.

Conjunto de Dados

Um arquivo CSV será fornecido contendo informações sobre o desempenho acadêmico de uma faculdade fictícia. O arquivo terá as seguintes colunas:

| MAT | Nome | Disciplina | Nota1 | Nota2 | Frequência (%) |
|-----|-------------------|-------------|-------|-------|----------------|
| 1 | Ana Beatriz Silva | Estatística | 8.5 | 7.0 | 90 |
| 2 | João Alex Pereira | Matemática | 6.0 | 5.5 | 75 |
| ... | ... | ... | ... | ... | ... |

Requisitos do Programa

1. Leitura e Estruturação dos Dados

- Importar os dados do arquivo CSV fornecido.
- Mostrar as 5 primeiras e últimas linhas do conjunto de dados.
- Imprimir a lista dos alunos sem repetições.
- Exibir a lista de disciplinas sem repetições.

2. Cálculos Estatísticos

- Calcular as seguintes estatísticas:
 - Média, mediana e desvio padrão das notas (separadas por disciplina).
 - Qual disciplina tem a maior e menor média de notas.
 - Frequência média dos alunos.
 - Quantidade de alunos aprovados, em recuperação e reprovados:
 - * Aprovado: Média das notas ≥ 7.0 e Frequência $\geq 75\%$.
 - * Recuperação: Média das notas ≥ 5.0 e < 7.0 e Frequência $\geq 75\%$.

* Reprovado: Média das notas < 5.0 ou Frequência $< 75\%$.

3. Classificação do Desempenho

- Calcular as seguintes estatísticas:
 - Média, mediana e desvio padrão das notas (separadas por aluno).
 - Top 3 alunos com a maior e menor média geral.
 - Top 3 alunos com a maior e menor frequência.
- Para cada disciplina:
 - Quantidade de alunos aprovados, em recuperação e reprovados.
 - Identificar os alunos acima da média da disciplina.
 - Identificar os alunos com frequência abaixo de 75%.

4. Modularização

- Estruturar o programa em funções bem definidas, como:
 - `ler_dados()`: Leitura e carregamento dos dados.
 - `calcular_estatisticas()`: Realização dos cálculos estatísticos.
 - `classificar_alunos()`: Determinação da situação dos alunos.
- Utilizar funções auxiliares para facilitar a organização do código.

Saída do Programa

Além de exibir as informações na tela, o programa deve gerar um arquivo de texto, relatório, contendo as estatísticas calculadas. O arquivo deve ser estruturado de forma clara e organizada, facilitando a leitura e interpretação dos resultados independentemente do código fonte.

Avaliação

- **Funcionalidade do Código:** 60%
- **Organização do Código:** 20%
- **Qualidade do Relatório:** 20%

Entrega

- **Formato:** Arquivo Python (.py) contendo o código e um relatório com as respostas para as análises realizadas.
- **Data:** Até o último dia do semestre.
- **Método de Submissão:** Enviar os arquivos na plataforma de ensino utilizada pela disciplina.

Observação

O conjunto de dados fictício será gerado e fornecido pelo professor, utilizando um script Python. O arquivo conterá informações simuladas para garantir que todos os alunos trabalhem com o mesmo material.