

# Passo a Passo para Reprodução dos Resultados do HW1

Aqui está o guia para reproduzir a análise, seguindo os passos solicitados.

---

## 1. Criar o .venv e instalar as dependências

Primeiro, configure o seu ambiente virtual para isolar o projeto.

Crie o ambiente virtual:

```
python -m venv venv
```

Ative o ambiente:

- No **macOS / Linux**:

```
source venv/bin/activate
```

- No **Windows (PowerShell / CMD)**:

```
.\venv\Scripts\activate
```

Instale as dependências:

```
pip install -r requirements.txt
```

Se o Jupyter Notebook não estiver no `requirements.txt`, instale-o também:

```
pip install notebook
```

---

## 2. Rodar os códigos para analisar as métricas estatísticas

Execute os scripts de análise monovariada para gerar as estatísticas descritivas (Tabelas II, III, IV) e os histogramas/boxplots iniciais (Figuras 2, 3, 4).

- **Análise Incondicional (geral):**

```
python HW1/class_unconditional.py
```

- **Análise Condicional (por classe):**

```
python HW1/class_conditional.py
```

---

### 3. Rodar o código para analisar a correlação

Execute este script para gerar a matriz de correlação (Figura 5) e os gráficos de dispersão (Figura 6).

```
python HW1/bivariate_analysis.py
```

---

### 4. Rodar o código para normalizar os dados

Execute este script para aplicar as transformações (Yeo-Johnson + Z-score) e salvar os dados processados (Figura 7). Esses dados transformados serão usados nos próximos passos.

```
python HW1/save_data_transformation.py
```

Este script também gera os resultados do **PCA** (Figuras 8, 9) salvando os scores, que serão usados no passo 6.

---

### 5. Aplicar as funções do PCA em um ambiente Jupyter Notebook

Para uma análise mais interativa dos Componentes Principais (PCA):

1. Inicie o Jupyter Notebook:

```
jupyter notebook
```

2. Crie um novo notebook (.ipynb).

3. Dentro do notebook:

- Importe as funções necessárias do diretório HW1/utils/ (ex.: `pca_calculation`, `pca_biplot`, `pca_scree_plot`).
  - Carregue os dados normalizados (criados no passo 4), por exemplo:  
`HW1/data_transformations/data_yeojohnson_zscore.csv`
  - Use as funções importadas para calcular o PCA, plotar o *scree plot* e o *biplot* interativamente.
- 

### 6. Rodar o código para realizar a detecção de outliers

Finalmente, execute o script de detecção de outliers, que usará os dados normalizados e os scores do PCA (gerados no passo 4) para identificar e plotar os outliers (Figura 10).

```
python HW1/outlier.py
```

