**DOCUMENTAÇÃO DO PROJETO DE ANÁLISE DE ÓBITOS E INDICADORES SOCIOECONÔMICOS**

**Introdução**

Este projeto tem como objetivo analisar a relação entre condições socioeconômicas e óbitos, considerando indicadores como o Índice de Vulnerabilidade Social (IVS) e o Índice de Equidade e Dimensionamento (IED).

**Estrutura do Projeto**

**Semana 1: Planejamento e Estruturação do Projeto**

* Definição dos objetivos do projeto e fontes de dados.
* Criação de repositório no GitHub com documentação inicial (README).
* Levantamento das bibliotecas necessárias (ex: requests, BeautifulSoup, pandas, numpy, matplotlib, seaborn, scikit-learn).
* Definição do pipeline geral do projeto.

**Semana 2: Coleta de Dados via Web Scraping**

* Implementação da coleta de dados de saúde usando bibliotecas como BeautifulSoup ou Scrapy.
* Extração de dados:
  + Óbitos e causas (CID-10)
  + Relação com o trabalho
  + Correção de causa
  + Informações demográficas (município, escolaridade, raça/cor)
* Testes e validação da integridade dos dados coletados.

**Semana 3: Obtenção de Dados Socioeconômicos**

* Coleta dos valores de IED e IVS por município.
* Obtenção do número de habitantes e faixa de porte populacional segundo o IBGE.
* Limpeza e padronização dos dados.

**Semana 4: Construção do Pipeline de Processamento**

* Definição do fluxo de tarefas:
  + Coleta via web scraping
  + Tratamento de dados ausentes
  + Conversão de tipos de variáveis
  + Padronização e junção das bases
  + Geração da base final para análise
* Elaboração de um fluxograma ilustrando as etapas do pipeline.

**Semana 5: Manipulação e Transformação dos Dados**

* Criação de scripts para limpeza e padronização dos dados utilizando pandas.
* Tratamento de valores nulos, conversão de datas e normalização de categorias.

**Semana 6: Agregação e Junção das Bases de Dados**

* Uso de merge e joins para consolidar as informações.
* Criação de chaves de ligação entre bases (ex: código do município).
* Validação da integridade da base final.

**Semana 7: Análise Descritiva das Variáveis**

* Análise de pelo menos 10 variáveis principais:
  + Variáveis numéricas: cálculo de média, mediana, geração de boxplots.
  + Variáveis categóricas: cálculo de percentuais de cada categoria.
  + Variáveis de data: avaliação de mínimo e máximo.

**Semana 8: Documentação e Entrega Final**

* Criação de um dicionário de variáveis contendo:
  + Nome da variável
  + Descrição breve
  + Tipo de dado
  + Completude na base de dados
* Melhoria do README do repositório do GitHub.
* Organização e documentação dos scripts.
* Submissão final do projeto para revisão.

Tecnologias Utilizadas

* Linguagem: Python
* Bibliotecas: pandas, numpy, matplotlib, seaborn, requests, BeautifulSoup, Scrapy
* Ferramentas: Jupyter Notebook, GitHub

**Estrutura do Repositório**

* data/ - Armazena os dados brutos e processados
* notebooks/ - Contém os notebooks de análise
* scripts/ - Scripts para processamento de dados
* docs/ - Documentação do projeto
* README.md - Resumo do projeto e instruções

Contato

Para dúvidas ou sugestões, entre em contato através do repositório no GitHub.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|x|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|x|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Semana 1: Planejamento e Estruturação do Projeto**

1. **Definir objetivos**: Estabeleça a relação entre IVS, IED e óbitos como foco do estudo.
2. **Criar repositório no GitHub**: No GitHub, crie um novo repositório e adicione um README.md inicial com a descrição do projeto.
3. **Instalar bibliotecas necessárias**: No terminal ou no Jupyter Notebook, instale as bibliotecas executando:

ip install requests beautifulsoup4 pandas numpy matplotlib seaborn scikit-learn scrapy

1. **Planejar o pipeline**: Desenhe um fluxograma no papel ou em uma ferramenta online (Ex: draw.io) detalhando o fluxo de dados desde a coleta até a análise.

**Semana 2: Coleta de Dados via Web Scraping**

1. **Identificar fontes de dados**: Descubra sites que fornecem os dados desejados (ex: portais governamentais de saúde).
2. **Fazer requisições web**: Use requests para acessar páginas:

import requests

response = requests.get("URL\_DO\_SITE")

print(response.text)

1. **Extrair informações com BeautifulSoup**:

from bs4 import BeautifulSoup

soup = BeautifulSoup(response.text, 'html.parser')

print(soup.prettify())

1. **Salvar os dados coletados** em arquivos .csv ou .json para análise posterior.

**Semana 3: Obtenção de Dados Socioeconômicos**

1. **Baixar dados do IBGE e outras fontes**: Acesse sites como o IBGE para obter IVS, IED e população.
2. **Carregar e visualizar os dados com pandas**:

import pandas as pd

df = pd.read\_csv("dados\_socieconomicos.csv")

print(df.head())

1. **Limpar dados ausentes e padronizar formatos** para facilitar a fusão posterior.

**Semana 4: Construção do Pipeline de Processamento**

1. **Definir um fluxo de tarefas** com as etapas principais.
2. **Criar funções modulares** para cada etapa do pipeline, como esta para carregar dados:

def carregar\_dados(arquivo):

return pd.read\_csv(arquivo)

1. **Criar um fluxograma** usando draw.io ou mermaid.js.

**Semana 5: Manipulação e Transformação dos Dados**

1. **Remover valores nulos**:

df.dropna(inplace=True)

1. **Converter colunas de data**:

df['data\_obito'] = pd.to\_datetime(df['data\_obito'])

1. **Padronizar textos (ex: nomes de municípios)**:

df['municipio'] = df['municipio'].str.upper()

**Semana 6: Agregação e Junção das Bases de Dados**

1. **Usar merge() para unir bases**:

df\_final = df\_obitos.merge(df\_socioeconomico, on="codigo\_municipio", how="left")

1. **Criar chaves de ligação** (ex: código do município).
2. **Verificar a integridade da base final**.

**Semana 7: Análise Descritiva das Variáveis**

1. **Gerar estatísticas descritivas**:

print(df\_final.describe())

1. **Criar boxplots para variáveis numéricas**:

import seaborn as sns

sns.boxplot(x=df\_final['idade'])

1. **Analisar distribuição de variáveis categóricas**:

print(df\_final['raça'].value\_counts(normalize=True))

**Semana 8: Documentação e Entrega Final**

1. **Criar um dicionário de variáveis**:

variaveis = {

"codigo\_municipio": "Código do município de residência",

"idade": "Idade no momento do óbito",

"raca": "Raça/cor do indivíduo",

}

1. **Melhorar README do GitHub** com exemplos de uso.
2. **Organizar os scripts e documentar** cada etapa para facilitar a replicação.