**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.

* 1. Webscrapping de dados de saúde (Processo utilizando alguma das bibliotecas ministradas no curso).
  2. Projeto no GitHub (Com documentação e README).
  3. Desenvolvimento de um pipeline, detalhando as etapas envolvidas desde o webscrapping até a entrega de uma base pronta para análise (ferramentas, sistema, linguagem utilizados). Pode ser em formato de texto usando fluxo de tarefas, com imagens usando um fluxograma e texto para complementar com descrição de cada tarefa.
  4. Manipulação de dados e transformação (avaliados através de scripts de manipulação usando técnicas similares as ministradas em aula).
  5. Agregação de dados e junção (merge) de bases de dados (também avaliado através dos scripts).
  6. Descritiva simples de pelo menos 10 variáveis da base final (se for numérica, avaliar média,
  7. Mediana e fazer um boxplot, se for categórica mostrar percentuais de preenchimento em cada categoria, se for data, avaliar mínimo e máximo).
  8. Entregar um dicionário das principais variáveis na base de dados (entre 10-15 variáveis)
  9. Contendo o nome da variável, uma descrição breve, o tipo, e a completude na base de dados.
  10. Levantamento das bibliotecas necessárias (ex: requests, BeautifulSoup, pandas, numpy, matplotlib, seaborn, scikit-learn, stats). Em sua maioria as bibliotecas, serão utilizadas para ajustes, leitura e interpretação, assim como criação de gráficos.
  11. Definição do pipeline geral do projeto, coletando informações sobre:
  12. Sobre óbito;
  13. Se o evento está relacionado ao processo de trabalho;
  14. Se houve correção ou alteração da causa do óbito após investigação;
  15. O (s) CID (s) informados no atestado de óbito, a causa básica da declaração do óbito (CID 10);
  16. O código do município de residência, a data de nascimento e data de óbito;
  17. Também tem interesse em saber o nível de escolaridade do indivíduo e escolaridade da mãe (se disponível);
  18. E informações sobre raça/cor do indivíduo.
  19. E para informações socioeconômicas queremos os valores de IED e IVS, além do número de habitantes e a faixa de porte populacional segundo o IBGE.
  20. Criação de repositório no GitHub com documentação inicial (README).

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

* 1. Implementação da coleta de dados de saúde usando bibliotecas como BeautifulSoup, Selenium fazendo webscraping da página através do link https://www.in.gov.br/en/web/dou/-/portaria-gm/ms-n-3.493-de-10-de-abril-de-2024-553573811.
  2. Extração de dados:
  3. Óbitos e causas (CID-10)
  4. Relação com o trabalho
  5. Correção de causa
  6. Informações demográficas (município, escolaridade, raça/cor)
  7. Testes e validação da integridade dos dados coletados.

**Semana 3: Obtenção de Dados Socioeconômicos**

3.1 Coleta dos valores de IED e IVS por município.

3.2 Obtenção do número de habitantes e faixa de porte populacional segundo o IBGE.

3.3 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.