

Lucas E. Zonfrilli

*Graduando Gestão da
Tecnologia da Informação
Fatec de Assis*
lucas.zonfrilli@fatec.sp.gov.br

Matheus S. Soares

*Graduando Gestão da
Tecnologia da Informação
Fatec de Assis*
matheus.soares29@fatec.sp.gov.br

Me. Antonio R. Pepece Jr

*Professor Orientador Fatec
Assis*
antonio.pepece@fatec.sp.gov.br

RESUMO

Este artigo destaca a viabilidade de usar dados públicos para calcular modelos estatísticos, com foco no Portal da Transparência do governo brasileiro. O objetivo é facilitar o acesso aos dados públicos por meio das APIs oficiais, realizar cálculos estatísticos com Python, abordando a correlação linear e regressão linear simples. O processo inclui a coleta de dados das APIs governamentais, extração de despesas do Ministério da Saúde e consideração dos leitos disponíveis na saúde pública de 2014 a 2022. Métodos estatísticos e o uso da linguagem Python são explicados, tornando informações passadas transparentes e disponíveis. A pesquisa oferece a oportunidade de usar Python para obter, calcular indicadores estatísticos e analisar dados governamentais abertos, os quais são acessíveis através de uma API devidamente documentada. O sucesso do estudo é evidenciado pelo código disponível no GitHub, mostrando a possibilidade de extrair dados públicos e avaliar modelos estatísticos.

Palavras-chave: API; Linguagem Python; Regressão linear.

ABSTRACT

This article highlights the feasibility of using public data to calculate statistical models, with a focus on the Brazilian government's Transparency Portal. The goal is to facilitate access to public data through official APIs, perform statistical calculations with Python, addressing simple linear correlation and linear regression. The process includes data collection from government APIs, extraction of expenses from the Ministry of Health, and consideration of available beds in the public health system from 2014 to 2022. Statistical methods and the use of the Python language are explained, making past information transparent and accessible. The research provides an opportunity to use Python to obtain and calculate statistical indicators and analyze open government data, which is accessible through a well-documented API. The success of the study is evidenced by the code available on GitHub, demonstrating the ability to extract public data and assess statistical models..

Keywords: API; Python Language; Linear Regression.

1 INTRODUÇÃO

Atualmente vive-se um período em que a todo momento se recebem milhares e milhares de informações que chegam pelas redes sociais, mídias digitais, TV, rádio entre outros meios. Saber analisar e separar as informações verdadeiras das falsas, também conhecidas como “fakenews¹”, é algo complexo, principalmente quando se refere a dados públicos relacionados aos gastos dos governos para benefício da sociedade.

O Portal da Transparência, do Governo Federal, é uma ferramenta útil para que a população tenha acesso e acompanhe as despesas e receitas do Governo Brasileiro. Por meio deste portal, é possível consultar os gastos de diversas áreas, economia, previdência social, educação, segurança, saúde, dentre outras. Além disso, há a possibilidade de acessar informações detalhadas sobre cada gasto realizado. Assim, por meio deste portal, a população pode verificar onde os recursos financeiros disponibilizados pelo governo federal foram aplicados.

Nesse sentido, faz necessário que qualquer cidadão tenha acesso aos dados disponíveis no referido Portal, para que possa verificar e analisar os gastos de acordo com as necessidades da população. Dessa forma, pode-se apontar onde há necessidade de melhorias nos serviços públicos, além de identificar possíveis desvios de recursos e aplicações indevidas.

Nesse sentido, o artigo objetiva disponibilizar meios de consultas aos dados públicos das API's (Interface de Programação de Aplicação) oficiais do Governo Federal Brasileiro. Posteriormente, realizar cálculos que utilizam modelos estatísticos como o uso da "Correlação Linear Simples" e "Regressão Linear Simples", utilizando a linguagem de programação Python.

Dessa forma será preciso contextualizar sobre os portais de dados públicos disponíveis, também conhecido como API (Interface de Programação de Aplicação). Apresentar os modelos estatístico Correlação Linear Simples e Regressão Linear Simples. Conceituar a linguagem de programação Python e apresentar as bibliotecas da linguagem que vamos utilizar.

Em seguida é descrito todo o processo de coleta de dados via API oficial do Governo Federal, no Portal da Transparência e Portal DataSUS; extração de dados referentes às despesas do Ministério da Saúde e, como parâmetro temporal, é considerado o número de leitos de internação disponíveis pelo SUS do período de 2014 a 2022. Desta forma, após a coleta de todos os dados, é aplicado o referido modelo matemático supracitado, juntamente com as bibliotecas Pandas, Matplotlib, Scikit-learn e Numpy.

Uma das lacunas de pesquisa detectadas é como prover acesso às informações de forma mais precisa e com maior facilidade de interpretação. O compartilhamento de informações sem análise objetiva e de difícil verificação, assim como a avaliação do desempenho com base no gasto e não na eficiência torna o presente trabalho como uma alternativa viável.

Na era da informação, tornar as informações do passado acessíveis e transparentes, íntegras e claras se faz necessário para que se construa um futuro abundante.

Além disso, disponibilizar às pessoas e empresas a possibilidade de uso da linguagem Python para obter informações de dados abertos governamentais com a finalidade de calcular indicadores estatísticos para obter informações para tomada de decisões.

2 DESENVOLVIMENTO

Para organizar e desenvolver os objetivos do presente artigo, será apresentado como a correlação e regressão linear simples se dá na utilização de dados estatísticos, a busca de dados pelo Portal DataSUS e API do Portal da Transparência e como a linguagem Python irá associar esses itens e calcular os modelos estatísticos e reproduzir o resultado.

2.1 MÉTODOS ESTATÍSTICOS

2.1.1 CORRELAÇÃO LINEAR SIMPLES

De acordo com Bussab e Morettin (2017) a correlação linear é uma função que determina a relação entre duas variáveis, elas podem ser representadas por pares ordenados (X,Y), sendo o X a variável exploratória e o Y a variável dependente. Essa correlação tem como objetivo indicar a força da associação entre as variáveis X e Y..

Quando é representado em um gráfico de dispersão, os dados estão distribuídos ao redor de uma linha imaginária, então pode-se dizer que visualmente os dados sugerem uma correlação linear.

Ao calcular o coeficiente de correlação linear se obtém o grau de relacionamento entre as variáveis, esses graus circulam entre -1 e 1 sendo quanto mais próximo de -1 a correlação é mais fraca entre as variáveis, se próximo ou igual a 0 evidencia uma correlação nula e próximo de 1 a relação é considerada forte ou perfeita em caso de ser igual a 1, conforme ilustra a figura 1.

Figura 1 - Fórmula utilizada para se obter o coeficiente (r).

$$r = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2} \sqrt{n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i\right)^2}}$$

Se confirmado a correlação entre as variáveis, aguça o experimento do cientista, surge a necessidade de uma função matemática que descreva esse relacionamento em uma linha reta de tendência, ou seja, uma equação de regressão.

2.1.2 REGRESSÃO LINEAR SIMPLES

De acordo com os estudos de HOFFMANN (2006), a regressão linear simples é uma técnica muito usada em estatística e econometria para fazer previsões de uma variável x através de uma variável y , para se fazer a reta de regressão é necessário conhecer duas características importantes sendo o coeficiente angular e o coeficiente linear da reta. A figura 2 ilustra a fórmula para obter a reta de regressão linear.

Figura 2 – Fórmula para obter a regressão linear simples.

$$\hat{y} = ax + b$$
$$a = \frac{n \cdot \sum x \cdot y - \sum x \cdot \sum y}{n \cdot \sum x^2 - (\sum x)^2}$$
$$b = \frac{\sum y}{n} - a \cdot \frac{\sum x}{n}$$

O coeficiente linear da reta é representado por α , b é o coeficiente angular e N é o tamanho da amostra. A variável x funciona como o dado preditor então deve ser de fácil acesso e y a variável menos acessível. Segundo HOFFMANN (2006) é importante ter em mente que nem todas as situações são bem aproximadas por uma equação linear, por isso é necessário se fazer o cálculo de correlação linear e apresentar os dados em um diagrama de dispersão para verificar que os valores de x e y apresentam uma predisposição linear.

2.2 API (INTERFACE DE PROGRAMAÇÃO DE APLICAÇÃO)

API em sua origem significa Application Programming Interface, traduzindo para o português significa Interface de Programação de Aplicativos, e segundo Jacobson, Brail e Woods (2012), corresponde a um conjunto de softwares com a função de intermediar a comunicação de dados entre aplicações. Através de uma API é possível que uma ou mais aplicações se comuniquem de forma transparente e ágil para o usuário, como se fosse uma só aplicação.

Atividades do dia-a-dia como: postar uma foto no Instagram e também enviá-la para o Facebook, consultar o saldo bancário, passar o cartão de crédito, são exemplos da utilização de uma API. Sua integração agrega a interação em nível de dados onde estes são enviados ou recebidos entre as aplicações envolvidas.

Para uma organização, uma API é um ponto de conexão entre ela e seus parceiros e clientes, compartilhando processos de negócio, serviços e conteúdo; conectando equipes de trabalho internas a desenvolvedores independentes, de uma forma fácil e segura (RICHARDSON E RUBY, 2007).

O propósito do desenvolvimento de uma API pode ser para uso por um grupo restrito de usuários ou pode ser desenvolvida tendo como alvo o público em geral. Assim, as APIs podem ser

classificadas em privadas ou públicas (DOERRFELD, 2016), cabendo à organização avaliar e decidir o tipo de API que pretende utilizar, com base na sua finalidade e objetivos.

Os dois principais estilos de arquitetura que têm dominado o mercado nos últimos anos são: REST e SOAP. No entanto é importante salientar que o próprio significado de SOAP deixa claro a intenção de ser um protocolo enquanto que “REST é definido apenas como um estilo arquitetural” (FIELDING, 2014), o que faz com que não seja adequado estabelecer uma comparação direta entre os dois estilos.

Para realizar a comunicação entre as diversas aplicações e uma API, verifica-se a necessidade de se ter padrões de comunicação e a definição do formato no qual os dados serão transmitidos. Atualmente existem dois formatos largamente utilizados para transportar e apresentar dados pela Internet: 1) Formato JSON (Java Script Object Notation); e 2) Formato XML (eXtensible Markup Language).

O formato JSON possui características de ser disponibilizado com tamanho menor que o XML devido ao maior número de tags (caracteres especiais de formatação) utilizadas ao escrever um arquivo XML, sendo que o formato JSON tem sido o formato de apresentação de dados escolhido na maioria das APIs desenvolvidas.

2.2.1 API PORTAL DA TRANSPARÊNCIA

Segundo o Portal da Transparência do Governo Federal, para utilizar a API de dados disponibilizada pelo Portal da Transparência, é necessário primeiro cadastrar um endereço de e-mail em <http://portaldatransparencia.gov.br/api-de-dados/cadastrar-email>. Após o cadastro, será enviado um token para o e-mail informado, que deverá ser utilizado nas consultas realizadas através da API.

Os dados dos serviços oferecidos pela API podem ser acessados através de requisições HTTP aos serviços REST. A documentação completa da API está disponível em <http://api.portaldatransparencia.gov.br/>. Na página "API de dados - Exemplos de uso" é possível encontrar exemplos de código-fonte para consumo da API em diversas linguagens de programação, como Javascript, Java, PHP e .NET.

É importante destacar que a utilização da API de dados do Portal da Transparência do Governo Federal é gratuita e aberta de livre utilização, em conformidade com as disposições do Decreto nº 8.777, de 11 de maio de 2016.

2.2.2 DATASUS

Segundo o site do DataSUS o Tabnet é um portal que disponibiliza informações relevantes para análises objetivas da situação sanitária, tomadas de decisão baseadas em evidências e elaboração de programas de ações de saúde.

O site traz dados de mortalidade, morbidade, acesso a serviços, qualidade da atenção, condições de vida e fatores ambientais, que são utilizados na construção de indicadores de saúde.

Além disso, também são encontradas informações sobre assistência à saúde da população, cadastros de redes hospitalares e ambulatoriais, cadastro de estabelecimentos de saúde, recursos financeiros e informações demográficas e socioeconômicas.

Para consultar os dados é possível acessar o link: <https://datasus.saude.gov.br/informacoes-de-saude-tabnet/>, e selecionar a categoria que desejar. No presente estudo será selecionado: Rede Assistencial, após CNES - Recursos Físicos e por fim seleciona Hospitalar - Leitos Internação *.

O sistema direcionará para o link: <http://tabnet.datasus.gov.br/cgi/deftohtm.exe?cnes/cnv/leiintbr.def>. Onde é feito a filtragem da amostra de dados que será analisada. Filtra-se então em “Linha”: Ano/mês compet. em “Coluna” não ativa, “Conteúdo” Quantidade SUS e no “Período disponível” seleciona-se janeiro de 2014 até dezembro de 2022.

Por fim é só clicar no botão Mostra e os Dados são gerados e salvos em .xlsx.

Para finalizar os ajustes para executar a Correlação Linear Simples e Regressão Linear Simples, foi calculado o valor médio anual, visto que o dado se dava a critério mensal.

2.3 LINGUAGEM PYTHON

A linguagem escolhida para aplicar o experimento é o Python, essa foi escolhida por ser de altíssimo nível, ter uma tipagem dinâmica e forte, além de ser orientada a objetos. Outra vantagem é que o Python é de código aberto, ou seja, pode ser incorporado em outros produtos sem custos de utilização. Segundo a TIOBE Index, Python é hoje a linguagem mais utilizada no mundo, representando 14,83% de market share.

Segundo McKinney (2018), “Nos últimos dez anos, Python passou de uma linguagem de computação científica inovadora, ou para ser usada “por sua própria conta e risco”, para uma das linguagens mais importantes em ciência de dados, aprendizado de máquina (Machine Learning) e desenvolvimento de softwares em geral, no ambiente acadêmico e no mercado. ”, neste trabalho são utilizadas bibliotecas de ciência de dados disponíveis para alcançar o objetivo.

Para fazer operações matemáticas complexas entre arrays e funções lineares será utilizado a biblioteca Numpy. Também será usado como um contêiner de dados para a manipulação durante o desenvolvimento do trabalho.

Grande parte da popularidade do Python em ciência de dados se dá a biblioteca Pandas, que traz uma gama de ferramentas para manipulação, seleção e tratamento de informações em alta performance, ele trabalha com uma estrutura tabular (tabelas) chamada Dataframe indexado por colunas e linhas. Depois de se tratar e transformar os dados vem a necessidade de apresentar esses dados de forma visual, para isso iremos usar o matplotlib. Ela é uma biblioteca para plotar e visualizar dados de forma bidimensional, é amplamente utilizada e essa sua popularidade a tornou a ferramenta padrão para visualização de dados.

3 RESULTADOS E DISCUSSÕES

Para melhor organização do código foi separado em duas etapas em arquivos diferentes denominados de: “ETL” e “PLOTAGEM”.

Na etapa de ETL o código é dividido em três partes: a importação das bibliotecas necessárias, a definição da função `get_variables()`, a chamada da função. Neste contexto, é comentado linha a linha do código dentro da função `get_variables()`, conforme ilustra as figuras 3, 4 e 5.

Figura 3 – Importação das bibliotecas

```
ETL.py > get_variables
1 import requests
2 import pandas as pd
3 import json
4 import time
```

Fonte: Autores

Figura 4 - Definição da função `get_variables()`.

```
6 def get_variables():
7     api_key = 'insira sua chave aqui'
8
9     dataframes = []
```

Fonte: Autores

A função `get_variables()` é definida e uma chave de API é definida em `api_key`. A variável `dataframes` é inicializada como uma lista vazia.

Figura 5 – Código de loop para obter dados.

```
11 for ano in range(2014, 2023):
12     url = f'https://api.portaldatransparencia.gov.br/api-de-dados/despesas/por-orgao?ano={ano}&orgao=36000&pagina=1'
13     print(url)
14     headers = {"chave-api-dados": api_key}
15
16     response = requests.get(url, headers=headers)
17
18     if response.status_code == 200:
19         # A requisição foi bem-sucedida
20         print(response.text)
21         data = json.loads(response.text)
22         df = pd.DataFrame(data)
23         dataframes.append(df)
24         time.sleep(120)
25         # Faça o que quiser com os dados da API
26     else:
27         # A requisição falhou
28         print(f"A requisição falhou com código de status {response.status_code}.")
29
```

Fonte: Autores

Um loop “for” é criado para iterar de 2014 a 2022 (anos inclusos). Para cada iteração, uma URL é criada usando o ano atual e uma variável de página igual a 1. A URL é impressa na tela e os cabeçalhos para a requisição da API são definidos. A função requests.get() é usada para fazer a chamada da API e a resposta é armazenada em response.

Se a resposta tiver um status_code igual a 200, o bloco de código dentro do if é executado. A resposta é impressa na tela, a resposta em formato JSON é carregada em data e, em seguida, convertida em um DataFrame do pandas em df. df é adicionado à lista dataframes e a execução do código é pausada por 2 minutos usando time.sleep(120), atendendo a requisitos da API do Portal da Transparência do Governo Federal.

Se a resposta tiver um status_code diferente de 200, o bloco de código dentro do else é executado e uma mensagem de erro é impressa na tela.

Figura 6 – Código para concatenação dos Dataframes.

```
31 df = pd.concat(dataframes, ignore_index=True)
32
33 x = df['pago'].str.replace('.', '').str.replace(',', '.').astype(float).values
34
35 leitos = pd.read_excel("Leitos.xlsx")
36 y = leitos['Quantidade_SUS'].values
37
38 return x, y
```

Fonte: Autores

Após a etapa de ETL (extração, armazenagem e modelagem) de dados, se dá início ao cálculo estatístico da Correlação Linear Simples e Regressão Linear Simples que se denomina de PLOTAGEM.

Nesta etapa o código também é dividido em três blocos principais: Importação de bibliotecas, processamento dos dados, plotagem do gráfico de dispersão com a linha de regressão linear, conforme ilustra as figuras 7,8 e 9.

Figura 7 – Importação das bibliotecas sklearn, matplotlib e ETL

```
PLOTAGEM.py > ...
1 from sklearn.linear_model import LinearRegression
2 import matplotlib.pyplot as plt
3 import ETL
```

Fonte: Autores

As bibliotecas sklearn, matplotlib e ETL são importadas. A primeira importa a classe LinearRegression da biblioteca scikit-learn, que é usada para calcular a linha de regressão linear,

matplotlib é usada para plotar gráficos e ETL importa o módulo ETL, que contém a função `get_variables()` que será usada para obter as variáveis `x` e `y`.

Figura 8 – Chamada da função “`get_variables()`” do módulo ETL.

```
5 # Obtendo as variáveis x e y da função ETL.get_variables()
6 x, y = ETL.get_variables()
7
8 # Calculando a linha de regressão linear
9 reg = LinearRegression().fit(x.reshape(-1, 1), y)
10 r = reg.coef_[0]
```

Fonte: Autores

Chama-se a função `get_variables()` do módulo ETL, que retorna as variáveis `x` e `y` obtidas a partir de um processo de Extração, Transformação e Limpeza (ETL) de dados;

Após esse procedimento, é utilizado o objeto `LinearRegression` da Scikit-learn para calcular a linha de regressão linear. Para isso, ele utiliza o método `fit()`, que recebe os valores de `x` e `y` e ajusta uma linha de regressão linear.

Em seguida, é calculado o coeficiente de regressão linear (`r`) a partir do objeto `LinearRegression`, que representa a inclinação da linha de regressão.

Figura 9 – Cálculo da regressão linear

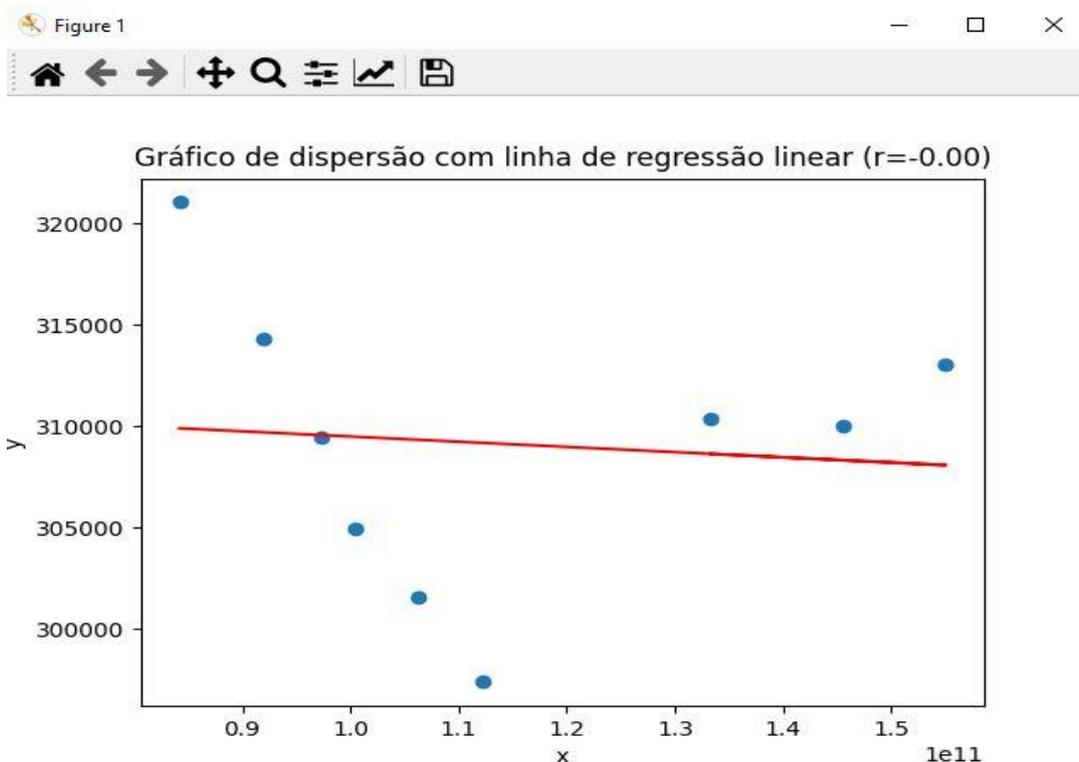
```
12 # Plotando o gráfico de dispersão com a linha de regressão linear
13 plt.scatter(x, y)
14 plt.plot(x, reg.predict(x.reshape(-1, 1)), color='red')
15 plt.xlabel('x')
16 plt.ylabel('y')
17 plt.title(f"Gráfico de dispersão com linha de regressão linear (r={r:.2f})")
18 plt.show()
```

Fonte: Autores

Por fim, plota-se o gráfico de dispersão utilizando o método `scatter()` da biblioteca `matplotlib`, passando as variáveis `x` e `y` como parâmetros, e plota-se a linha de regressão linear utilizando o método `plot()`, passando os valores de `x` e `y` como parâmetros, junto com a linha de regressão calculada a partir do método `predict()` do objeto `LinearRegression`. Consecutivamente, são adicionados os títulos aos eixos e um título geral para o gráfico.

Ao chamar o método `show()` do objeto `pyplot`, o gráfico é exibido na tela, conforme ilustra a figura 10.

Figura 10 – Gráfico de dispersão



Fonte: Autores

Visualiza-se no gráfico acima, o eixo “y”, representando o número de leitos de internações. No eixo “x” o total de gastos do Ministério da Saúde Brasileiro. O presente gráfico mostra a dispersão destes dados distribuídos ao longo de 2014 a 2022 e calcula mostra uma linha de tendência através do cálculo da Regressão Linear Simples. É possível avaliar o índice de correlação através do indicador “r”, exibido após o título do gráfico.

4 CONSIDERAÇÕES FINAIS

A facilidade de acesso aos dados públicos foi fundamental para a realização do estudo. O uso de dados públicos permitiu que o trabalho fosse desenvolvido sem a necessidade de recursos financeiros ou autorização de terceiros, o que tornou o projeto viável e acessível.

A documentação da API utilizada para acessar os dados públicos neste trabalho foi excelente. As informações estavam bem organizadas e as funcionalidades da API foram bem descritas, o que permitiu uma fácil compreensão de como utilizar a ferramenta.

Embora tenhamos enfrentado algumas dificuldades, como o limite de uma requisição por minuto e a restrição de informações de apenas um ano por vez, conseguimos contorná-las com a inserção de um tempo de espera adequado entre as requisições. Essas limitações não impediram o sucesso do trabalho, que foi possível graças aos métodos de ciência de dados empregados na análise dos dados coletados.

O objetivo do trabalho foi alcançado com sucesso. Utilizando o modelo estatístico de regressão linear, foi possível calcular e reproduzir os dados coletados. O resultado obtido confirmou a hipótese inicial do estudo e demonstrou que os dados públicos podem ser utilizados de forma eficiente na análise de dados.

O código desenvolvido para este trabalho está disponível em um repositório no GitHub. Este repositório contém todo o código utilizado na análise dos dados, bem como as bibliotecas utilizadas e os dados coletados.

Portanto, conclui-se que a utilização de dados públicos em projetos de ciência de dados é uma estratégia viável e eficiente para a obtenção de informações valiosas, desde que sejam empregados métodos e técnicas adequados para a sua extração, modelagem e reprodução.

5 REFERÊNCIAS

DATASUS. Ministério da Saúde. Disponível em: <<https://datasus.saude.gov.br/informacoes-de-saude-tabnet/>>. Acesso em: 14 abr. 2023.

DOERRFELD, B., WOOD, C., ANTHONY, A, SALDOVAL, K. LAURET, A. **The API Economy: Disruption and the Business of APIs** 1ed. Nordic APIs, 2016.

FIELDING, R. and Reschke, J. **Hypertext Transfer Protocol (http/1.1): Message Syntax and Routing.** RFC 7230, Internet Requests for Comments, RFC Editor, 2014.

HOFFMANN, R. **Análise de Regressão: Uma Introdução a Econometria.** São Paulo: Editora Hucitec, 2006.

HUTTNER, Luiz Ricardo Goulart. **É fake news?: como elementos do jornalismo são utilizados para a elaboração de “notícias falsas”.** Dissertação de mestrado. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2020.

JACOBSON, D., BRAIL, F., WOODS, D. **APIs: A Strategy Guide.** 1 ed. Gravenstein Highway North, Sebastopol, USA: O'Reilly, 2012.

MCKINNEY, Wes. **Python Para Análise de Dados: Tratamento de Dados com Pandas, NumPy e IPython.** São Paulo: Novatec, 2018.

MORETTIN, Pedro Alberto; BUSSAB, Wilton O. **Estatística básica.** 9. ed. São Paulo: Saraiva, 2017. 468 p. ISBN 978-85-472-2023-5.

PORTAL DA TRANSPARÊNCIA. Governo Federal. Disponível em: <<https://portaldatransparencia.gov.br/>>. Acesso em: 14 abr. 2023

RICHARDSON, L., RUBY, S. **RESTful Web Services.** 1 ed. Gravenstein Highway North, Sebastopol, USA O'Reilly, 2007.

TIOBE INDEX. Tlobe Software. Disponível em: <<https://www.tiobe.com/tiobe-index/>>. Acesso em: 05 abr. 2023.