MVP – Engenharia de Dados

Disciplina: Sprint: Engenharia de Dados (40530010057_20250_01)

Avaliação da Pesquisa Brasileira em Medicina: Uma Análise Bibliométrica (2005-2021)

Priscila Costa Albuquerque

Matrícula: 4052024002118

Sumário

1. Objetivo do MVP - Sprint de Engenharia de dados

Este trabalho teve como objetivo principal **analisar a produção científica dos pesquisadores brasileiros na área de Medicina**, utilizando como proxy a classificação das revistas indexadas na Scopus (All Science Journal Classification - ASJC). O planejamento contemplou responder aos objetivos específicos:

1. **Quantificar a produção científica brasileira** no período de 2005 – 2021, na área de Medicina, identificando tendências anuais e dos principais tipos de publicação.
2. **Mapear as parcerias de colaboração científica**, nacionais e internacionais, por meio da análise das afiliações institucionais e países coautores envolvidos nas publicações.
3. **Avaliar o impacto das publicações** com base em indicadores bibliométricos, como número de citações.
4. **Avaliar o financiamento da pesquisa** brasileira com base nas menções de agradecimento dos artigos científicos.

Essas questões foram selecionadas com o intuito de direcionar a coleta e análise dos dados bibliométricos, identificando áreas com maior potencial de expansão e possíveis lacunas estratégicas que exigem maior investimento. Este trabalho pode ajudar a compreensão do cenário atual da pesquisa brasileira em Medicina, ofertando subsídios para políticas científicas e estratégias institucionais voltadas ao fortalecimento da produção científica nacional, à ampliação das redes colaborativas e ao incentivo à ciência aberta, alinhando-se às tendências globais de transparência e impacto na disseminação do conhecimento.

2. Coleta

Os dados foram coletados a partir do banco de dados Scopus (Elsevier B.V.) (1), utilizando uma API de pesquisa. A biblioteca utilizada para extração, armazenamento em cache e manipulação dos dados foi a Pybliometrics (2), uma API-Wrapper baseada

em Python especificamente desenvolvida para interagir com o banco de dados Scopus, permitindo a recuperação eficiente de artigos científicos originais e publicações de revisão, como descrito a seguir:

2.1. Recuperação de dados

Somente publicações na área temática dá medicina, publicadas entre 2005 e 2021, com pelo menos um autor afiliado a uma instituição brasileira, foram incluídas na análise. O período de 2005 a 2021 foi escolhido para capturar tendências de longo prazo e, ao mesmo tempo, garantir a inclusão de desenvolvimentos recentes até a pandemia pela COVID-19. Foi utilizada a seguinte estrutura de consulta:

```
for a in areas:
  for y in years:
    query = "SUBJAREA({}) AND (AFFILCOUNTRY(ZIMBABWE) AND PUBYEAR = {})\
    AND (LIMIT-TO(DOCTYPE, \"ar\") OR LIMIT-TO(DOCTYPE, \"re\"))".format(a, y)
    s = ScopusSearch(query)
    pd.DataFrame(pd.DataFrame(s.results)).to_csv("{}-{}.csv".format(a,y))
    print(a + ", " + str(y) + ", " + str(s.get_results_size()))
    #time.sleep(120)
```

Devido ao grande volume de dados e às limitações de solicitação da API, o processo de recuperação foi executado separadamente para cada ano, com intervalos entre as solicitações para atender aos limites de tempo da API. Para permitir a análise comparativa, os dados de publicação global do mesmo período também foram recuperados usando a pesquisa avançada de documentos do Scopus.

Após a coleta, foram carregados e transformados na plataforma de nuvem Databricks Community Edition (3) garantindo persistência e acessibilidade, conforme documentado nas evidências de recuperação e ETL.

2.2. Evidência da recuperação de dados

O código utilizado para a recuperação dos dados está disponível no repositório GitHub do projeto (Coleta_Medi_Scopus.ipynb). Para fins de documentação e validação técnica, os registros (logs) da execução dos notebooks também estão incluídos neste documento como evidências adicionais.

```
In [1]: pip install pandas

Requirement already satisfied: pandas in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from pandas) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

In [2]: pip install pybliometrics

Requirement already satisfied: pybliometrics in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (3.5.2)
Requirement already satisfied: requests in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from pybliometrics) (2.29.0)
Requirement already satisfied: tqdm in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from pybliometrics) (4.65.0)
Requirement already satisfied: urllib3 in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from pybliometrics) (1.26.16)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from requests->pybliometrics) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from requests->pybliometrics) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in /Users/priscilaalbuquerque/anaconda3/lib/python3.11/site-packages (from requests->pybliometrics) (2023.5.7)
Note: you may need to restart the kernel to use updated packages.

In [3]: from pybliometrics.scopus import ScopusSearch, AuthorSearch, config, utils

        import pandas as pd
        import time

        [redacted]

In [4]: print(config['Authentication']['APIKey'])  # Show keys

        d999d39cd0ef9dabb2cea1a683f5fea3

In [5]: utils.config.keys

Out[5]: <bound method Mapping.keys of <configparser.ConfigParser object at 0x1068789d0>>

In [6]: areas = ["medi"]
        years = range(2010,2020)
```

Figura 1: Captura da execução do notebook

```
In [4]: print(config['Authentication']['APIKey'])  # Show keys

        [redacted]

In [5]: utils.config.keys

Out[5]: <bound method Mapping.keys of <configparser.ConfigParser object at 0x1068789d0>>

In [6]: areas = ["medi"]
        years = range(2010,2020)

In [7]: # areas = ["nurs", "vete", "dent", "heal", "mult", "agri", "bioc", "immu", "neur", "phar"]
        # areas = ["vete"]
        # years = range(2005,2022)
        # years = range(2010,2020)
        # years = range(2019,2020)

In [27]: for a in areas:
             for y in years:
                 query = "SUBJAREA({}) AND (AFFILCOUNTRY(Brazil) AND PUBYEAR = {})\
                 AND (LIMIT-TO(DOCTYPE, \"ar\") OR LIMIT-TO(DOCTYPE, \"re\"))".format(a, y)

                 s = ScopusSearch(query)
                 pd.DataFrame(pd.DataFrame(s.results)).to_csv("{}-{}.csv".format(a,y))
                 print(a + ", " + str(y) + ", " + str(s.get_results_size()))
                 #time.sleep(120)

        medi, 2010, 142
        medi, 2011, 127
        medi, 2012, 150
        medi, 2013, 156
        medi, 2014, 188
        medi, 2015, 210
        medi, 2016, 274
        medi, 2017, 312
        medi, 2018, 313
        medi, 2019, 364
```

Figura 2: Registro final indicando sucesso na extração dos dados.

3. Modelagem

O processo de modelagem dos dados foi estruturado em camadas incrementais com os dados persistidos separadamente em tabelas individuais desde a camada inicial, seguindo as boas práticas recomendadas para Data Lakehouse, organizadas em camadas Bronze, Silver e Gold.

Estrutura de diretórios utilizada:

/FileStore/tables/
└── MVP_project/
  ├── raw/ - **Camada Bronze**
  (dados brutos, originais em .csv)
  | └── MediBR_Scopus.csv
  ├── bronze/ -**Dados brutos persistidos separadamente em formato Delta**
  (fato_publications, dim_fund, dim_author_affil)
  ├── silver/ - **Dados limpos e transformados**
  (mesma estrutura modular da Bronze)
  └── gold/ - **Modelo dimensional final para análises**
  (esquema estrela)

Na **Camada Bronze**, os dados brutos originais (MediBR_Scopus.csv) foram inicialmente armazenados sem alterações para garantir auditabilidade e preservação integral. Em seguida, esses dados foram separados e persistidos individualmente em formato Delta, gerando desde já três tabelas distintas: uma tabela fato (fato_publications), contendo métricas-chave relacionadas às publicações, e duas tabelas dimensionais (dim_fund, sobre fontes de financiamento das pesquisas, e dim_author_affil, contendo dados dos autores e suas afiliações institucionais).

Na **Camada Silver**, cada uma dessas três tabelas passou por processos específicos e detalhados de limpeza, validação e padronização, resultando em dados tratados e consistentes, prontos para análises robustas. Essa abordagem modular desde a origem

permitiu maior controle e rastreabilidade no tratamento das diferentes entidades (fatos e dimensões), facilitando manutenção e escalabilidade da solução.

Por fim, na **Camada Gold**, adotou-se um modelo dimensional em esquema estrela, estruturado em torno da tabela fato central (fato_publications), diretamente relacionada às dimensões dim_fund e dim_author_affil. Essa estrutura permite uma consulta eficiente e clara dos dados, viabilizando análises ágeis que respondem diretamente às questões analíticas e objetivos de negócio previamente definidos pelo projeto.

A validação e verificação dos dados em cada camada (Raw, Bronze, Silver e Gold) foi realizada por meio de comandos Spark no ambiente Databricks Community, garantindo integridade, consistência e correta estruturação dos dados. Exemplos práticos de comandos executados foram incluídos na documentação técnica, comprovando a correta persistência e transformação dos dados.

3.1. Evidência da modelagem dos dados

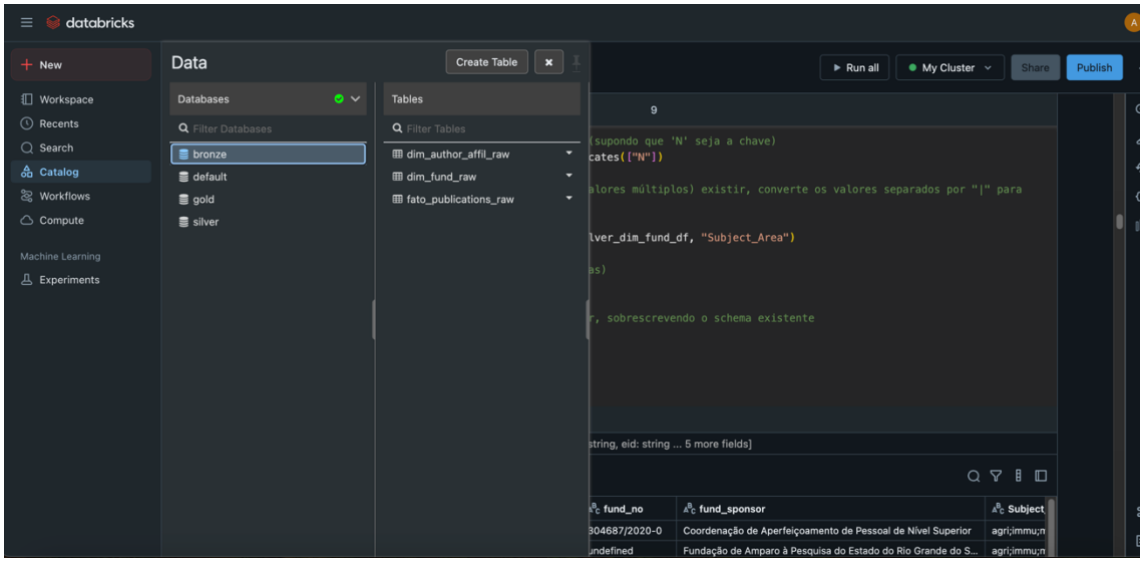

Figura 3: Arquivos raw camada Bronze.

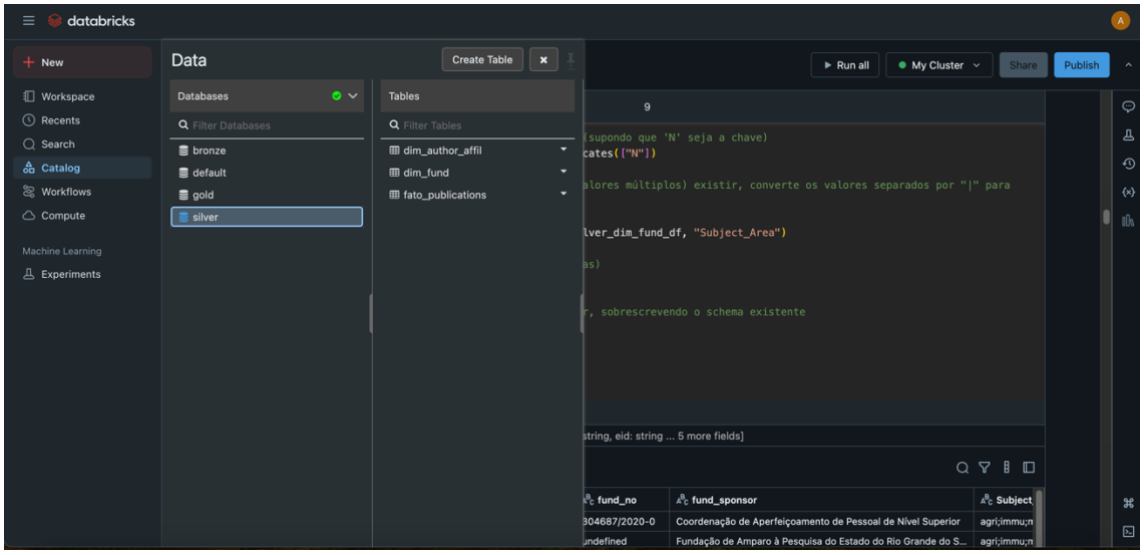

Figura 4: Tabelas transformadas na camada Prata.

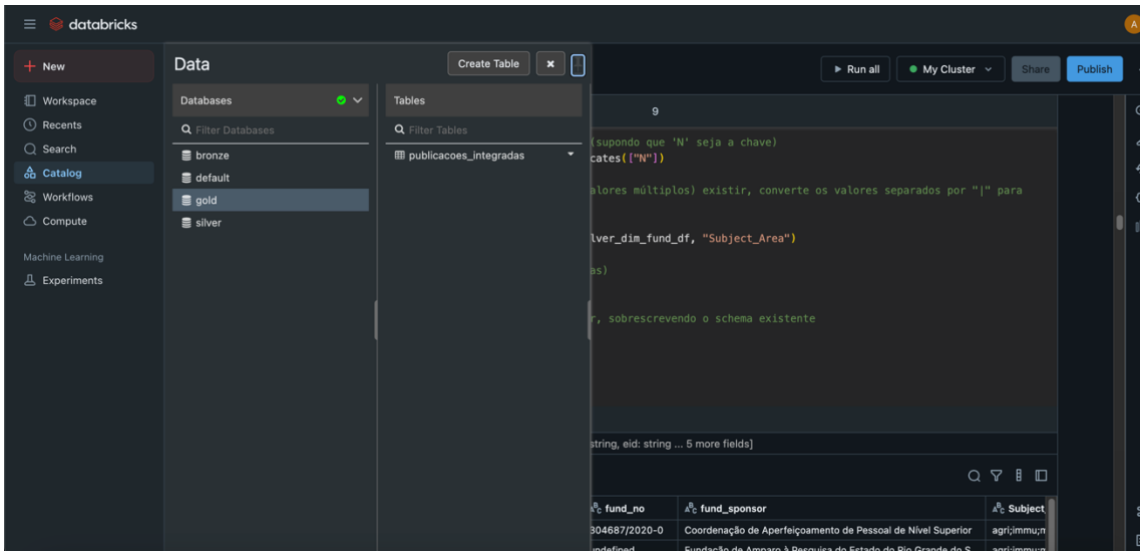

Figura 5: Database integrada na camada Gold.

3.2. Documentação do Catálogo de Dados

Este catálogo documenta o modelo dimensional implementado na camada Gold do projeto, com o objetivo de viabilizar análises rápidas e estratégicas sobre a produção científica brasileira na área de Medicina, utilizando dados extraídos da Scopus.

Tabela Fato (fato_publications)

Contém métricas quantitativas relacionadas às publicações científicas.

Atributo	Tipo (Python)	Tipo (SQL)	descrição
eid	String (Primary Key)	Varchar(255)	Identificador único da publicação na base Scopus
doi	String	Varchar(255)	Digital Object Identifier da publicação – Identificador alfanumérico único para propriedade intelectual online
pii	String	Varchar(255)	Publisher Item Identifier
pubmed_id	Float	Float	Identificador da publicação na base PubMed
title	String	Text	Título do artigo
subtype	String	Varchar(255)	Tipo resumido da publicação
subtypeDescription	String	Text	Descrição detalhada do subtipo
creator	String	Text	Autor principal
coverDate	Date	Datetime	Data da publicação
Year	Int	Int	Ano da publicação
coverDisplayDate	String	Varchar(255)	Data formatada para exibição
publicationName	String	Varchar(255)	Nome da revista
issn	String	Varchar(255)	ISSN da revista
source_id	Float	Float	Identificador da fonte na base Scopus
eIssn	String	Varchar(255)	ISSN eletrônico
aggregationType	String	Varchar(255)	Tipo de agregação (Journal, Book, etc.)
volume	Int	Float	Volume da publicação
issueIdentifier	Int	Int	Número da edição
article_number	String	Varchar(255)	Número do artigo (identificador interno)
pageRange	String	Varchar(255)	Intervalo de páginas
description	String	Text	Resumo ou descrição do artigo
authkeywords	String	Text	Palavras-chave dos autores
citedby_count	Int	Int	Número de citações

openaccess	Int	Boolean	Indicador de acesso aberto
freetoread	String	Boolean	Indicador se está livre para leitura
freetoreadLabel	String	Varchar(255)	Rótulo do tipo de acesso livre
Subject Area	String	Text	Área temática da publicação

Dimensão de Financiamento (dim_fund)

Contém informações sobre o financiamento das pesquisas.

Atributo	**Tipo (Python)**	**Tipo (SQL)**	**descrição**
fund_acr	String	Varchar(255)	Sigla da agência financiadora
fund_no	String	Varchar(255)	Número do financiamento
fund_sponsor	String	Varchar(255)	Nome da agência financiadora

Dimensão Autor e Afiliação (dim_author_affil)

Contém dados detalhados sobre autores e suas respectivas afiliações institucionais.

Atributo	**Tipo (Python)**	**Tipo (SQL)**	**descrição**
afid	string	int	IDs das afiliações
affilname	String	Text	Nomes das instituições de afiliação
affiliation_city	String	Text	Cidades das afiliações
affiliation_country	String	Text	Países das afiliações
author_count	Float	Float	Número de autores
author_names	String	Text	Nomes dos autores
author_ids	String	Int	Identificadores dos autores
author_afids	String	Int	IDs das afiliações dos autores

Relacionamentos entre tabelas:

dim_fund (fund_id) 1 ←———— N fato_publications (fund_id)

dim_author_affil (author_affil_id) 1 ←———— N fato_publications (author_affil_id)

1:N (um para muitos): Cada entrada na dimensão pode estar associada a múltiplas entradas na tabela fato, porém cada entrada na tabela fato aponta para apenas uma entrada específica nas dimensões correspondentes.

4. Carga

A carga dos dados foi executada por meio de scripts desenvolvidos em Python, utilizando as APIs do PySpark para processamento e transformação, em conjunto com consultas SQL para a validação e verificação dos resultados. Foram seguidas as seguintes etapas:

- **Ingestão dos dados RAW na camada Bronze:** Leitura dos arquivos CSV sem transformação, utilizando PySpark para carregar os dados brutos.
- **Transformações e limpeza na camada Silver:** Aplicação de funções em Python e PySpark para padronização dos nomes das colunas, remoção de espaços e conversões de tipos, além de operações de limpeza utilizando SQL para conferir padrões e integridade dos dados.
- **Integração na camada Gold:** Consolidação dos dados por meio de joins entre as tabelas, utilizando PySpark para a junção e consultas SQL para validar a conciliação dos conjuntos de dados.

A corretude dos dados foi validada por meio de visualizações limitadas (apenas 5 linhas) e testes de integridade dos joins realizados, garantindo a consistência e a qualidade dos dados, estando os mesmos persistidos com segurança na plataforma Delta Lake do Databricks.

5. Análise

5.1. Análise da qualidade dos dados

A análise de qualidade dos dados envolveu a utilização de técnicas estatísticas, visualizações e validação cruzada. Foram aplicadas funções do PySpark e consultas SQL para identificar e quantificar possíveis problemas. Por exemplo, foi verificada a completude dos dados por meio da contagem de valores nulos e vazios em colunas críticas, o que permitiu identificar eventuais lacunas de preenchimento. A consistência foi avaliada através da verificação de duplicidades, especialmente na chave primária (como a coluna "eid"), garantindo que cada registro fosse único, além de confirmar a integridade referencial entre as diferentes tabelas. Quanto à validade, foram aplicadas regras de integridade para assegurar que os valores estejam dentro dos padrões esperados, como a conferência de que a coluna "openaccess" contenha apenas os valores 0 ou 1, e a análise de estatísticas descritivas para colunas numéricas, como "Year_Cleaned". Os resultados demonstraram um alto grau de completude e consistência, com baixa incidência de valores nulos ou duplicados, e os dados se encontraram, em sua maioria, em conformidade com os padrões definidos para validade.

5.2. Solução correta do problema

Com base na análise realizada, foi possível responder às perguntas inicialmente propostas da seguinte forma:

Objetivo 1: Quantificar a produção científica brasileira

Para quantificar a produção científica brasileira no período de 2005 a 2021 na área de Medicina, os dados da camada Gold foram filtrados para identificar a quantidade de publicações por ano correspondentes à área. A visualização com gráfico de dispersão mostrou que a produção científica apresentou uma tendência crescente ao longo dos anos, com variações pontuais. Além disso, a distribuição dos tipos de publicação (por

exemplo, artigos originais versus revisões foi avaliada, permitindo identificar os principais tipos que compõem esse cenário.

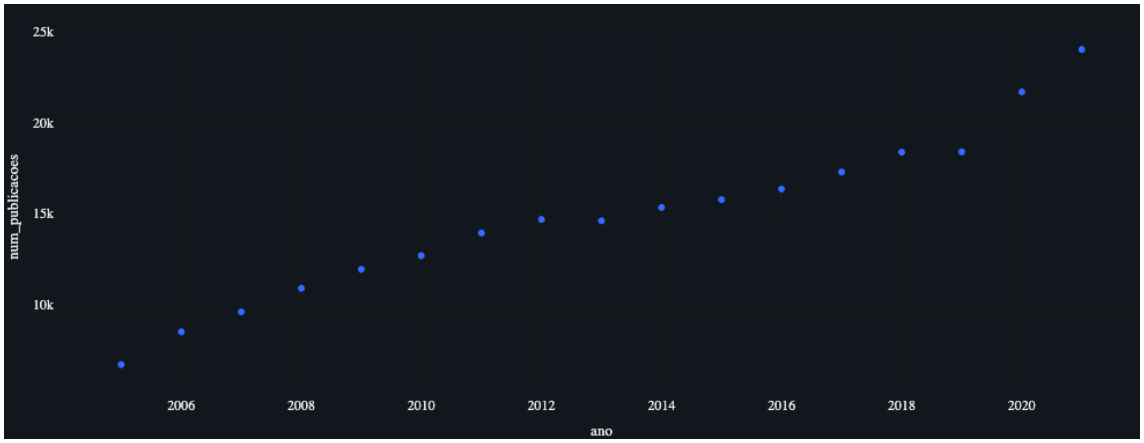

Figura 6: Número de publicações por ano.

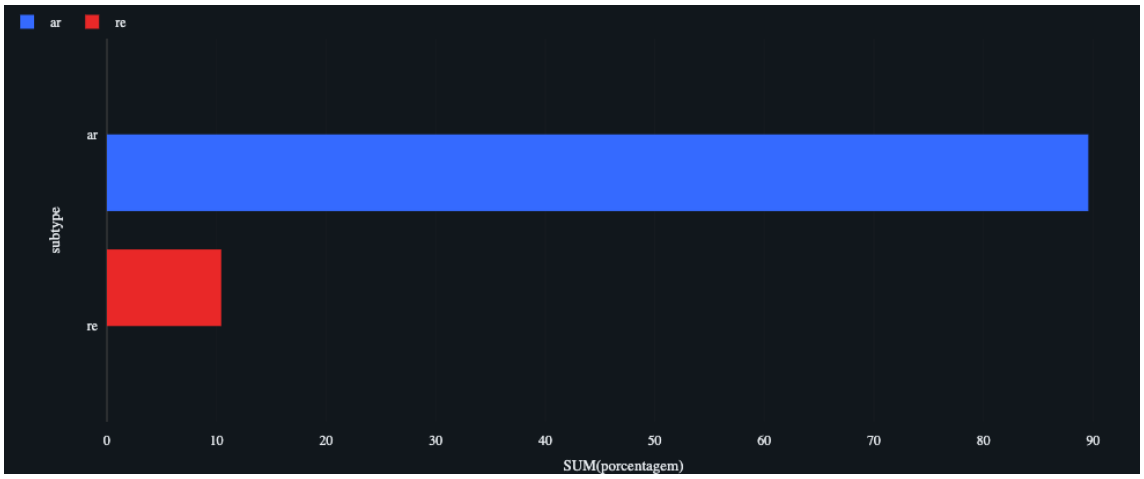

Figura 7: Proporção dos tipos de documentos.

Objetivo 2. Mapear as parcerias de colaboração científica

No mapeamento das parcerias de colaboração científica, foram analisados as afiliações institucionais e os países dos coautores. Ao agrupar as instituições e países foi possível identificar os principais parceiros nacionais quanto internacionais. Os resultados evidenciaram parcerias estratégicas, mostrando que publicações brasileiras contam com colaborações com instituições e pesquisadores de diversos países, ampliando o alcance e a relevância das pesquisas.

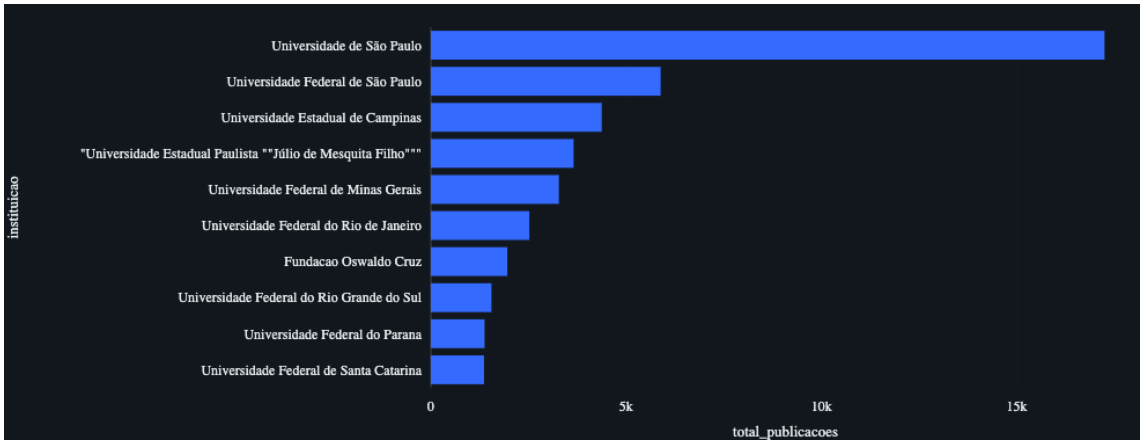

Figura 8: Distribuição de publicações por instituição.

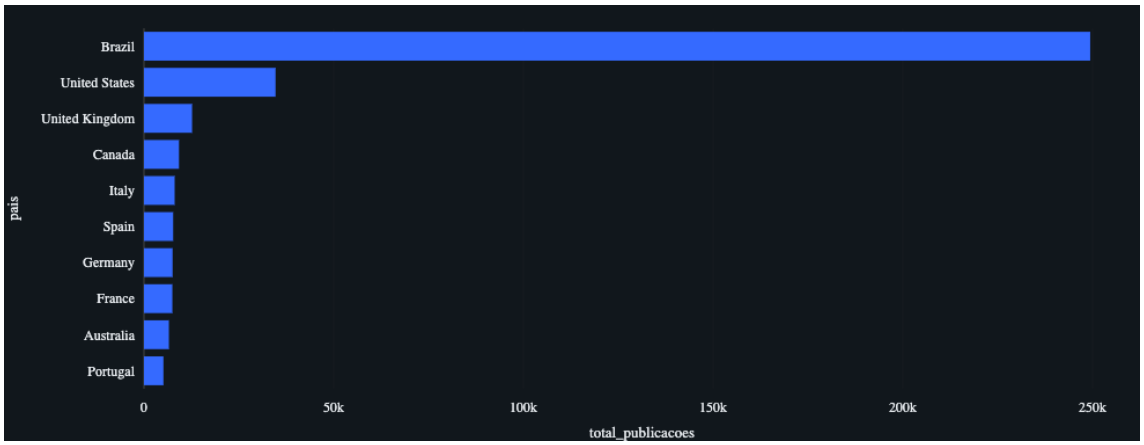

Figura 9: Distribuição de publicações por país.

Objetivo 3. Avaliar o impacto das publicações

Para avaliar o impacto das publicações com base em indicadores bibliométricos, os dados de citações foram analisados. As publicações foram ordenadas pelo número de citações, destacando os artigos com maior impacto. Essa análise permitiu não apenas identificar os trabalhos mais citados, mas também mensurar a relevância dos pesquisadores e instituições envolvidos, oferecendo uma visão quantitativa da influência dessas publicações no campo da Medicina.

Tabela 1: Top artigos por número de citações.

title	citedby_count
Sorafenib in advanced hepatocellular carcinoma	9146
Global, regional, and national prevalence of overweight and obesity in children and adults during 1980-2013: A systematic analysis for the Global Burden of Disease Study 2013	8178
Safety and efficacy of the BNT162b2 mRNA Covid-19 vaccine	7400
Apixaban versus warfarin in patients with atrial fibrillation	6743
Disability-adjusted life years (DALYs) for 291 diseases and injuries in 21 regions, 1990-2010: A systematic analysis for the Global Burden of Disease Study 2010	6407

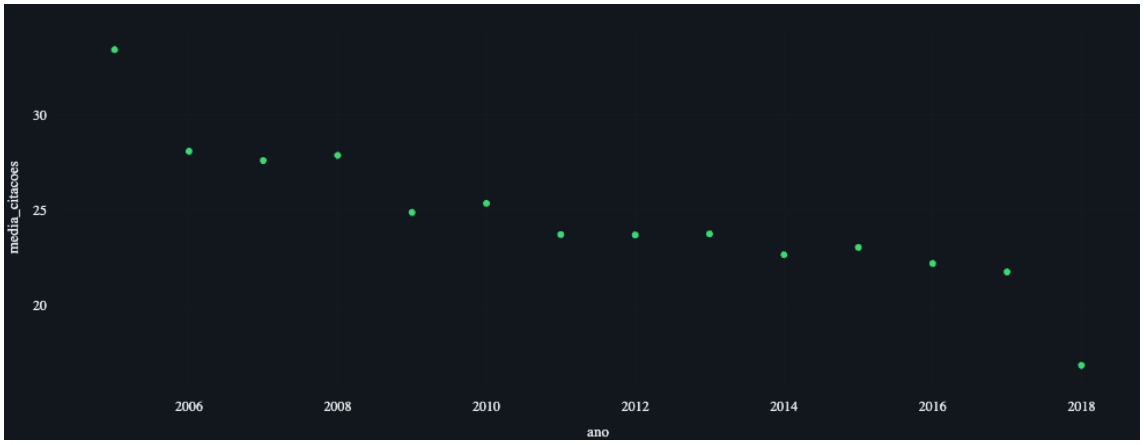

Figura 10: Evolução da Média de citações ao longo dos anos.

Objetivo 4. Avaliar o financiamento

O financiamento à pesquisa no Brasil conta com uma forte presença de agências de fomento nacionais, como a CAPES, FAPESP e CNPq, mas também recebe contribuições significativas de órgãos internacionais, em especial do National Institutes of Health (NIH) e suas subdivisões (NIAID, NHLBI, NCI, entre outras). Esse envolvimento do NIH reflete o crescente prestígio da pesquisa brasileira em escala global, pois fomenta colaborações internacionais que ampliam a visibilidade e o impacto dos estudos nacionais. Entretanto, possíveis mudanças nas políticas de investimento dos Estados Unidos ou cortes orçamentários podem levar a uma redução futura do apoio oferecido por essas agências, impactando a continuidade de projetos em áreas críticas da ciência e tecnologia.

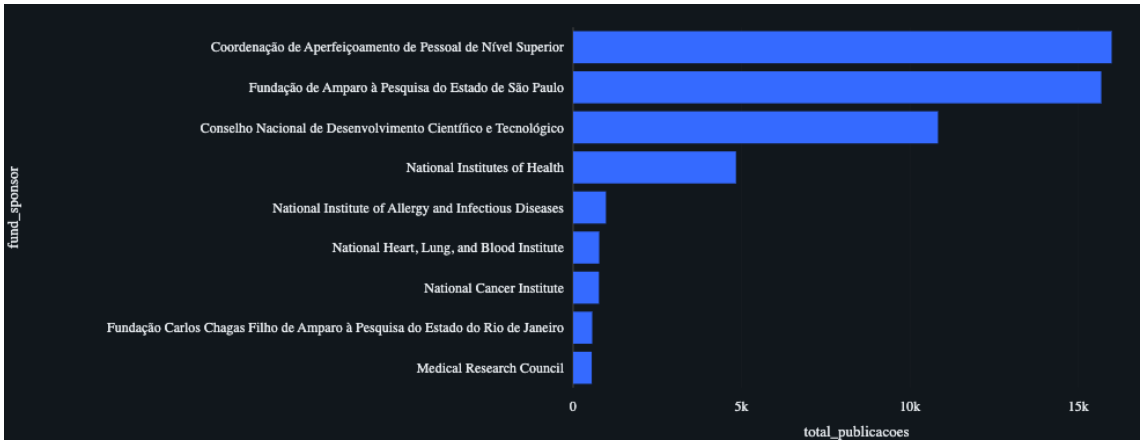

Figura 11: Análise de financiamento das pesquisas brasileiras em medicina.

5.3. Discussão crítica dos resultados

Os resultados foram analisados à luz do contexto do problema, que envolve a construção de um pipeline de dados voltado para o monitoramento da produção científica a partir da base Scopus. Destacam-se o crescimento contínuo no número de publicações ao longo dos anos, a identificação das instituições e países com maior volume de produção científica, além da visibilidade dos artigos mais citados e dos principais financiadores envolvidos.

Entretanto, alguns desafios ou limitações devem ser considerados para futuras análises. A qualidade e padronização dos dados, especialmente nas colunas de afiliação institucional e país, exigem cuidados adicionais de limpeza e normalização, visto que variações de nomenclatura podem impactar diretamente na contagem e agrupamento de entidades. Além disso, a análise de citações por ano deve ser interpretada com cautela, uma vez que publicações mais recentes naturalmente ainda não tiveram tempo suficiente para acumular citações significativas.

Outro ponto importante é que a análise bibliométrica não necessariamente reflete a qualidade científica das publicações de forma abrangente, sendo sensível a fenômenos como autocitações, variações por área do conhecimento e políticas editoriais. Por fim,

embora os dados de financiamento tragam informações valiosas, eles dependem da completude e precisão das fontes originais, o que pode gerar subnotificação.

Essas considerações indicam caminhos para o aprimoramento do pipeline, como o uso de dicionários de padronização para afiliações, enriquecimento dos dados com classificações por área do conhecimento, e a adoção de métricas complementares de impacto. Ainda assim, os resultados obtidos já oferecem uma base robusta para análises estratégicas e para o desenvolvimento de dashboards voltados à gestão da produção científica.

6. Autoavaliação

Avalio positivamente meu desempenho nesta sprint, especialmente considerando que iniciei sem nenhum conhecimento prévio em engenharia de dados e linguagens de programação. Apesar das dificuldades iniciais, consegui avançar significativamente, principalmente por buscar complementar o conteúdo das aulas com outras fontes de estudo que me ajudaram a entender melhor os conceitos e aplicá-los na prática.

Embora o conteúdo apresentado em aula tenha sido importante como ponto de partida, senti a necessidade de explorar materiais adicionais para acompanhar o ritmo das atividades. Acredito que essa iniciativa contribuiu bastante para meu desenvolvimento ao longo da sprint, e pretendo manter esse hábito de aprendizado contínuo nas próximas etapas.

7. Referências

1. Baas J, Schotten M, Plume A, Côté G, Karimi R. Scopus as a curated, high-quality bibliometric data source for academic research in quantitative science studies. Quant Sci Stud [Internet]. 2020 Feb 1 [cited 2024 Sep 30];1(1):377–86. Available from: https://doi.org/10.1162/qss_a_00019

2. Rose ME, Kitchin JR. pybliometrics: Scriptable bibliometrics using a Python interface to Scopus. SoftwareX [Internet]. 2019 Jul [cited 2024 Sep 2];10:100263. Available from: https://linkinghub.elsevier.com/retrieve/pii/S2352711019300573

3. Databricks Community Edition [Internet]. [cited 2025 Apr 2]. Available from: https://community.cloud.databricks.com/?o=4155913850843846