**INTERFERÊNCIA DOS RUÍDOS, SUJEIRAS, VALORES AUSENTES NA**

**BASE DE DADOS DO COVID E VACINAÇÃO.**

*INTERFERENCE FROM NOISE, DIRT, MISSING VALUES IN THE COVID AND VACCINATION DATABASE.*

**Helvio Rezende**

Graduação em sistemas de informação pelo Centro Universitário Una de Uberlândia-MG, servidor público municipal lotado na Secretaria Municipal de Saúde, com experiência nos sistemas, SIA, SIH, CNES.

email: [helviorezende.ti@gmail.com](mailto:helviorezende.ti@gmail.com),

**RESUMO**

Na área da saúde, detém grandes volumes de informações em seus banco de dados, que contém além de informações pessoais dos pacientes, dados sobre os respectivos atendimentos. E, minerar esses dados para descobrir conhecimento não é tarefa trivial. É preciso conhecer os dados, o processo de análise e descoberta, as tarefas e técnicas de mineração e as ferramentas matemáticas e computacionais que se aplicam nesse contexto, como as bases de dados geradas pelas notificações do Covid19 e Vacinação contra covid19. O estudo de caso apresentado ilustra a importância do tratamento dos ruídos, sujeiras e valores ausentes contidos nessas bases de dados que podem atrapalhar a respectiva análise, que tanto auxilia a tomada de decisão.

PALAVRAS-CHAVE: Python, covid19, mineração, filtro, opendatasus, IBGE, Dataframe, numpy, pandas, matplotlib, linux, Os X, Windows.

**ABSTRACT**

In the health area, it holds large volumes of information in its database, which contains, in addition to patients' personal information, data on the respective attendances. And, mining this data to discover knowledge is no trivial task. It is necessary to know the data, the analysis and discovery process, the mining tasks and techniques and the mathematical and computational tools that apply in this context, such as the Databases generated by Covid19 and Covid19 Vaccination notifications. The case study presented illustrates the importance of treating noise, dirt and missing values contained in these databases that can interfere with the respective analysis, which helps so much in decision making.

**KEYWORDS:** Python, covid19, mining, filter, opendatasus, IBGE, Dataframe, numpy, pandas, matplotlib, linux, Os X, Windows.

**1 INTRODUÇÃO**

Na área da saúde, são encontrados diversos programas que alimentam extensas bases de dados, contendo as mais variadas informações, por exemplo, Sistema de Informação sobre Mortalidade (SIM), Sistemas de Informações sobre Nascidos Vivos (SINASC), Sistema de Informação Sobre Agravos Notificados (SINAN), dentre outras.

O Ministério da Saúde, dentre a legislação e normas de utilização dos usuários da saúde pública e aplicações desenvolvidas para registro e contabilidade dos casos notificados, padronizou esse registro através das fichas de atendimento, contendo informações pessoais dos pacientes, informações de sua localização, dos sintomas e exames realizados. Sistemas dispersos nos quais essa informação não tenha a tratativa correta para gerar conhecimento.

Destacam-se nessas fichas disponibilizadas pelo Ministério da Saúde, acidentes com animais peçonhentos, acidentes graves, doenças como aids adulto, aids criança, coqueluche, cólera, dengue, etc, cada formulário com questões que vão nortear o profissional a classificar se os sintomas relatados pelo paciente é alguma complicação ou se poderá ser descartado.

Observa-se que em casos específicos, por exemplo, em 2020, o Mundo foi acometido, pelo que tudo indica, a maior pandemia da história, causada pelo Vírus Siscorv 2, coronavírus, causando covid19. Através de um empenho de todo mundo, a medida que pacientes eram identificados, procuravam unidades de saúde, tinham seus quadros de saúde avaliados, alimentavam bases de dados, por exemplo, no site: covid.saude.gov.br, base de dados nacional, ou Coronavírus Resource Center na John Hopkins University & Medicine, no Open Datasus que disponibiliza a base de dados sobre a vacinação.

Portanto, o objetivo principal deste artigo científico é utilizar Python, para analisar base de dados do covid19 e da vacinação, verificando valores ausentes, ruídos, dados digitados incorretamente que possam dificultar a análise de dados.

**2 REVISÃO LITERÁRIA**

Diante de uma pesquisa bibliográfica em artigos e livros sobre o tema de Análise de dados utilizando python, Mineração de dados, tem-se nesta seção, a definição das variáveis que serão estudadas, esclarecendo o significado de cada uma delas e como acontecem as relações dentro do tema.

2.1Mineração de Dados

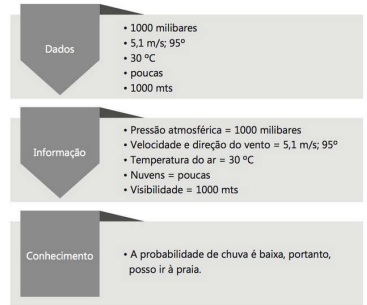
Mineração de Dados (em inglês, Data Mining) são processos de exploração e análise de grandes volumes de dados em busca de padrões, previsões, erros, associações, dentre outros (AMARAL, 2016). E surgiu como área de pesquisa e aplicação independente em meados da década de 1990, mas suas origens na matemática, estatística e computação são muito anteriores a esse período. A área também ganhou evidência nos últimos anos depois de ser cunhado o termo Big Data e com a publicação do relatório intitulado “Big Data: The Next Frontier for Innovation, Competition, and Productivity”, pelo McKinsey Global Instituto em meados de 2011.

Para Amaral (2016), Big Data é a produção de informação com velocidade, volume e variedade e está diretamente relacionado com a Mineração de Dados. E, como exemplo de uso do Big Data o portal Canal Tech (https.://canaltech.com.br/big-data/o-que-e-bg-data) cita aquelas empresas que almejam realizar os desejos de seus clientes antes que eles o peçam; como a Amazon, com as sugestões de sites de compras; o Netflix e Spotify, com recomendações de serviços.

Silva, Peres e Boscarioli(2016), definem Mineração de Dados como uma área que existe devido a uma grande quantidade de dados disponível para ser explorada e a evolução tecnológica evidenciada pelo aumento da capacidade de armazenamento propiciado pela popularização dos gadgets, massificação do uso da internet e influência das redes sociais, contribuíram para aumento da capacidade de gerar dados, não só pelos dados estruturados como os não estruturados.

No cenário de crescimento exponencial na quantidade de dados coletados e armazenados não tem exclusividade apenas na internet. Por causa do desenvolvimento tecnológico, tanto na qualidade quanto na quantidade de sensores geradores e monitores de dados, as empresas melhoraram sua capacidade de armazenamento e disponibilidade de dados. Além do tempo e esforço, despendidos, nas organizações para construção e manutenção de bases de dados, surgindo assim, especialidades como administradores de banco de dados (DBA) e de indexação de bancos de dados, deixa a base de dados com tanta informação que não se tem ideia das informações que se pode extrair. Cabe ressaltar que frequentemente os dados não podem ser analisados manualmente em virtude de fatores como grande quantidade de registros, elevado número de atributos, valores ausentes, presença de dados qualitativos e não quantitativos, entre outros, que surge a mineração de dados, como um processo sistemático, interativo e iterativo, de preparação e extração de conhecimentos a partir de grandes bases de dados. A Figura 1, apresentada por De Castro e Ferrari (2016), apresenta um esquema de extração de conhecimento a partir da informação obtida através de dados.

A figura 1, apresentada por De Castro e Ferrari(2016, apresenta um esquema de extração de conhecimento a partir da informação obtida através dos dados.



Diferença entre dados, informação e conhecimento.

Fonte: DE CASTRO e FERRARI, 2016.

A mineração de dados faz parte de um processo mais amplo, conhecido como descoberta de conhecimento em bases de dados (Knowledge Discovery in Databases, ou KDD)(AMARAL, 2016).

Segundo De Castro e Ferrari (2016) o processo de KDD consiste em quatro etapas, sendo:

* a primeira, a base de dados; uma coleção organizada de dados; que são valores quantitativos ou qualitativos referentes a um conjunto de itens propiciando eficientemente a recuperação de dados que podem ser entendidos como o nível mais básico de abstração a partir do qual a informação em seguida a extração de conhecimento;
* a segunda etapa é a preparação ou pré-processamento de dados; etapa que antecede a mineração, pois prepara os dados para uma análise eficiente e eficaz, a qual constitui na remoção de ruídos e dados inconsistentes, a combinação de dados obtidos de várias fontes, ou seja, integração, seleção ou redução, na escolha dos dados relevantes à análise e a transformação ou consolidação dos dados em formatos apropriados a mineração;
* a terceira etapa, mineração de dados, corresponde à aplicação de algoritmos capazes de extrair conhecimentos a partir dos dados pré-processados, através de técnicas de análise descritiva, agrupamento, predição, associação e detecção de anomalias; e,
* finalmente, avaliação ou validação do conhecimento, responsável por identificar conhecimentos verdadeiramente úteis e não triviais através da avaliação dos resultados da mineração.

Com tudo, tais etapas são correlacionadas e interdependentes, e consiste em considerar suas inter-relações, de forma ideal a abordagem para extrair informações relevantes em bancos de dados e sua influência no resultado, como ilustrado na Figura 2.

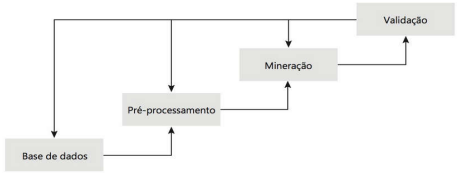


Figura2 - Processo de Descoberta do conhecimento em bases de dados

Fonte: DE CASTRO e FERRARI, 2016

2.2 Análise Descritiva dos Dados

A mineração de dados é uma disciplina interdisciplinar e multidisciplinar que envolve conhecimento de áreas como banco de dados, estatística, aprendizagem de máquina, computação de alto desempenho, reconhecimento de padrões, computação natural, visualização de dados, recuperação de informação, processamento de imagens e de sinais, análise espacial de dados, inteligência artificial, entre outras. Duas categorias de tarefas são classificadas para especificar os tipos de informações a serem obtidas na mineração de dados, conforme De Castro e Ferrari (2016):

1. Descritivas: caracterizam as propriedades gerais dos dados;

2. Preditivas: fazem inferência a partir dos dados objetivando precisões.

Uma etapa inicial do processo de mineração que não requer elevado nível de sofisticação é a análise descritiva dos dados, ou seja, o uso de ferramentas capazes de medir, explorar e descrever características intrínsecas aos dados. Essas análises permitem investigar a distribuição de frequência, as medidas de centro e variação, e as medidas de posição relativa e associação de dados. Assim como utilizar técnicas elementares de visualização para um melhor entendimento da natureza e distribuição dos dados. (DE CASTRO e FERRARI, 2016).

As funcionalidades da mineração de dados especificam os tipos de informações a serem obtidas nas tarefas de mineração. E, nas tarefas de aprendizado de máquina, esse aprendizado pode ser dividido em classificação, regressão, agrupamentos e regras de associação. (DE CASTRO e FERRARI, 2016; AMARAL, 2016).

2.3 Entendimento e Preparação dos Dados

Para melhor entendimento dos dados e de sua análise é preciso compreender a forma de armazenamento destes.

2.4 Entendimento dos Dados

Amaral(2016) explica que uma tarefa de banco de dados é composta por linhas e colunas, e que ao aprendizado de máquina as colunas são atributos e as linhas uma instância. Enquanto, Silva e Peres e Boscarioli (2016) dizem que a estatística descritiva é uma ferramenta capaz de descrever e resumir dados, mostrando aspectos importantes do conjunto de dados, por exemplo, tipo de distribuição associada e os valores mais representativos do conjunto.

Contudo, para Silva, Peres e Boscarioli(2016), faz parte das fases de descoberta do conhecimento a coleta de dados, o pré e o pós-processamento dos resultados provenientes da mineração, o que resulta antes na preparação dos mesmos para que sejam numerados.

2.5 Preparação dos dados

* A preparação dos dados consiste no processamento e ajuste das diversas informações levantadas no entendimento sobre o assunto a ser minerado e os dados. Castro e Ferrari (2016), explicam que ocorrem basicamente três tipos de problemas com os dados a serem observados na base:
* Incompletude: podem ser falta de valores de um dado atributo, entretanto essa falta pode passar despercebida;
* inconsistência: são dados diferente e, conflitantes versões do mesmo dado aparecem em locais variados;
* Ruído: dependendo do contexto, em vídeo pode ser chuvisco na imagem, em rádio, interferência do sinal de áudio, mas em mineração de dados, se aproxima do conceito de ruído em estatística e do processamento de sinais.

Diante disso é preciso fazer o pré-processamento dos dados, para isso Catro e Ferrari(2016, definem as principais tarefas de pré-processamento como:

* Limpeza: para imputação de valores ausentes, correção de incosistèncias e remoção de ruídos
* Integração: unir várias fontes em único local, como data warehouse;
* Redução: reduzir a dimensão da base de dados, agrupando ou eliminando atributos redundantes, ou reduzir quantidade de objetivos da base;
* Transformação: para padronizar e deixar os dados em formato possível, mas as técnicas de mineração.

Continuando com os conceitos de Castro e Ferrari(2016), limpeza dos dados é um problema que afeta a maior parte das bases de dados reais e as respectivas ferramentas atuam no sentido de imputar valores ausentes, suavizar ruidos, identificar valores discrepantes (outliers) e corrigir inconsistências.

Silvia, Peres e Boscarioli (2016), dizem que a limpeza dos dados ameniza a existência de valores ausentes (missing values) e a existência de valores ruidosos (noise values).

Castro e Ferrari (2016), apresentam métodos tradicionais de imputação de valores ausentes:

* Ignorar objeto: remover todos os objetos que possuem um ou mais valores ausentes;
* Imputar manualmente os valores ausentes, ou seja, substituir os valores ausentes por outro valor;
* Utilizar uma constante global para imputar os valores ausentes: imputar um valor constante ao valor ausente;
* Imputação do tipo hot-ceck: imputar um valor de objeto similar a um objeto ausente;
* Imputar de acordo com a última observação: ordenar a base e imputar valor ausente com valor da célula anterior;
* Usar a média pela média ou moda dos valores do atributo;
* Usar modelos preditivos para imputar valor ausente.

Ainda na fase de pré-processamento dos dados, Silva, Peres e Boscarioli (2016), retratam que a integração dos dados consiste em integrar dados de diversas fontes de bases.

Castro e Ferrari (2016), explicam que a concatenação de todos os dados se torna um dos passos essenciais antes de se realizar a mineração de dados, entretanto a integração desses dados pode resultar vários problemas, por exemplo, as formas de armazenagem, convenções dos dados, datas, chaves de acesso, padronizações e outras características.

Segundo Castro e Ferrari (2016), processo de integração abrange três aspectos:

* Redundância: quando mesmo dado aparecem em dois locais diferentes da base, são dados repetidos, duplicidade;
* Duplicidade;
* Conflitos: para uma mesma entidade diferentes valores aparecem em diferentes fontes de dados.

Quando a base de dados para análise é imensa, a mineração consome um esforço computacional representados pelo espaço e tempo de processamento, necessitando a aplicação das técnicas de redução de dados reduzindo a quantidade de objetos da base e da quantidade de atributos, no caso de dimensionalidade. Dentre eles, destacam-se os métodos(CASTRO e FERRARI, 2016):

Seleção de atributos (ou características): atributos irrelevantes, pouco relevantes ou redundantes são detectados e removidos na redução de dimensionalidade;

Compressão de atributos: através de algoritmos de codificação ou transformação de dados na redução de dimensionalidade, em vez de seleção;

Redução no número de dados: dados são removidos substituídos ou estimados por representações menores, modelos paramétricos e métodos não paramétricos, como agrupamento, amostragem e histogramas;

Discretização: os valores de atributos são substituídos por intervalos ou níveis conceituais mais elevados, reduzindo a quantidade final de atributos.

2.6 Tratando valores ausentes

Segundo McKinney(2018), dados ausentes são representados em objetos do pandas, para dados numéricos, o pandas utiliza o valor de ponto flutuante NaN(Not a Number), chamado também de valor de sentinela, convenção usada na linguagem de programação R, referenciando os dados ausentes como NA, que significa Not Available(indisponível). Em aplicações estatísticas, dados NA podem ser dados inexistentes ou dados que existem, porém não foram observados.

McKinney(2018), mostra na tabela 1 lista de algumas funções relacionadas ao tratamento de dados ausentes.

|  |  |
| --- | --- |
| argumento | descrição |
| dropna | Filtra rótulos de eixos, baseado no fato de os valores para cada rótulo terem dados ausentes, com limites variados para a quantidade de dados ausentes a ser tolerada. |
| fillna | Preenche os dados ausentes com algum valor ou utilizando um método de interpolação como ‘ffill' ou ‘bfill’. |
| isnull | Desenvolve valores booleanos informando quais valores estão ausentes/são NA. |
| notnull | Negação de isnull |

Tabela 1 - Tratamento para valores ausentes

Assim, seguindo a tabela 1, de McKinney(2018), para filtrar dados ausentes manualmente, pandas.isnull() que traz a base de dados, representando os valores ausentes por True, onde não tem valores ausentes por False, e utilizando o comando pandas.isnull().sum() traz a quantidade de valores ausentes contidos nas colunas. Para descartar os valores ausentes, basta utilizar data.dropna(), o qual excluirá qualquer linha que contenha somente valores ausente, utilizando o comando data.dropna(axis=1, how = ‘all’) excluirá toda coluna que contenha valores ausentes.

Nos ensina também que filtrando dados ausentes ou excluindo, pode também preenchê-los com comando df.fillna(‘valor a sustituir’), além disso com criatividade, substituir pela média, através do df.fillna(df.means()).

McKinney também traz tabela 2 contendo referências para fillna.

|  |  |
| --- | --- |
| argumento | descrição |
| value | Valor escalar ou um objeto do tipo dicionário a ser usado para preencher valores ausentes |
| method | Interpolação; por padrão, será ‘ffill’ se a função for chamada sem outros argumentos |
| axis | Eiro a ser preenchido; o default é axis=0 |
| inplace | Modifica o objeto que faz a chamada, sem gerar uma cópia |
| limit | Para preenchimento para a frente (foward) e para trás (backward), é o número máximo de valores consecutivos a serem preenchidos. |

Tabela 2 - Referências para tratamento de valores ausentes

Sob uma perspectiva de armazenagem de dados (data warehouse), o processo de mineração pode ser visto como um estágio avançado do processamento analítico on-line (On-line Analytical Processing – OLAP), conforme ilustração retirada da referência De Castro e Ferrari (2016).

Portanto, nessa etapa do trabalho, a revisão bibliográfica identificou as etapas de se trabalhar com análise de dados, começando com obter a base de dados, sendo assim as bases de dados, tanto relacionadas ao Covid 19, tanto a vacinação, foram disponibilizadas pelo Ministério da Saúde, através do site covid.saude.gov.br que forneceu a base do covid e opendatasus.saude.gov.br, que forneceu a base de vacinação contra covid19.no Próprio site do covid apresenta também, na opção de menu sobre, a descrição das informações fornecidas na base de dados e conceitos básicos para entendimento da mesma. também foi realizada uma análise das bases de dados, na aplicação Python, para identificar ruidos, sujeitas e valores ausentes, cumprindo assim as etapas, descritas nesta revisão bibliográfica, que se dispõe a análise de dados.

**3 METODOLOGIA**

É importante destacar o principal objetivo deste trabalho, realizar um estudo de caso sobre as bases de dados do Covid19 e da vacinação contra covid19, que no noticiário retrata como a maior pandemia que o Mundo passou, para isso, foi utilizado a aplicação Python, jupyter Notebook para análise.

3.1 O que é e motivo de utilizar Python

Nesta seção serão apresentadas as ferramentas, técnicas, equipamentos, amostras e os procedimentos usados para se obter os resultados desejados com o uso da aplicação Python, mais especificamente, Jupyter Notebook.

Primeiramente, McKinney(2018) explica que Python surgiu em 1991, e com Perl, Ruby e outras, se tornou uma das linguagens de programação interpretadas mais populares, sendo Python e Ruby popularizaram-se mesmo a partir de 2005 na construção de sites, através de seus frameworks Rails do Ruby e Django do Python. Foram chamadas de linguagens de scripting, pois permitiam escrever pequenos programas ou scripts, rapidamente, para automatizar tarefas. Após 10 anos, Python deixou de ser uma linguagem inovadora para ser usada em ciência de dados, aprendizado de máquina(machine learning) e desenvolvimento de softwares em geral, tanto no ambiente acadêmico, quanto no mercado.

McKinney, explica também, que decorrer da análise e modelagem dos dados , 80% do tempo é gasto em sua preparação como carga, limpeza, transformação e reorganização, algumas bases de dados não constituem o formato correto para determinada tarefa, porém a biblioteca pandas, do Python, traz ferramentas de alto nível, rápido e flexível, para permitir manipulação dos dados e deixá-los no formato correto, explica McKinney(2018).

Para Al Sweigart Python se refere à linguagem de programação (com regras de sintaxe para escrever o que é considerado um código Python válido) e ao software do interpretador Python, que lê o código-fonte(escrito na linguagem Python) é executa suas instruções. O interpretador Python é gratuito e pode ser baixado de http://python.org/ , há várias versões para Linux, OS X e WINDOWS.

O Python é proveniente do grupo surreal de comédia britânico monty python, e não do nome da cobra e seus programadores chamados carinhosamente de Pythonistas.

Cabe ressaltar também, que devido a gravidade da pandemia, um exército de profissionais no mundo, se determinaram a enfrentar a pandemia e socorrer os enfermos, com isso também o exército de profissionais que tinham por objetivo alimentar sistemas com as bases de dados obtidas, consequentemente, os Municípios, Estados e Governo Federal, trabalhando nessas informações, diversos sistemas alimentando essas bases de dados, diversos profissionais empenhados em alimentar essas informações em tão pouco tempo, com isso, a pressa, falta de padronização nos sistemas, a falta de padronização na digitação, podem influenciar na base de dados que apesar de ter uma aplicação que facilite a análise de dados, a base de dados ficou comprometida pois sistema não identifica palavras com acentuação, campos de valores ausentes, ou seja, informações que não foram preenchidas, bases de dados extensas que exigem poder computacional grande para análise.

**4 RESULTADOS E ANÁLISE DOS DADOS**

Em uma breve apresentação da aplicação Python, na figura abaixo, mostra importação das bibliotecas necessárias para análise e manipulação dos dados, como a biblioteca numpy, pandas e matplotlib.

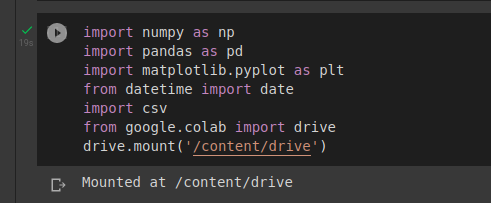


Figura 3 Importando bibliotecas no python para análise da base de dados.

Já na figura 4, mostra importando a base de dados para aplicação, são 5 arquivos divididos por semestres desde 2020, no qual tem-se 2020 parte 1 e 2, 2021 parte 1 e 2 e 2022 parte 1.

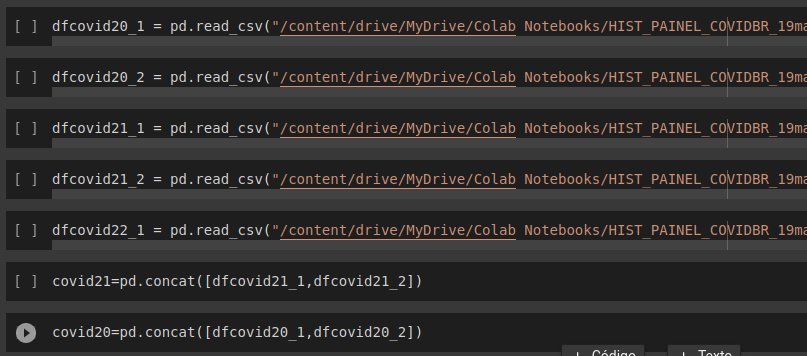


Figura 4 Mostrando importação da base de dados para Python.

Explicando as formas de tratamento da base de dados, na figura 5, a linha 12 cria um DataFrame com nome de geral e concatena as partes da base importada, formando uma base completa e na linha 13, procedimento renomeia, na coluna mês, o número do mês pelo nome.

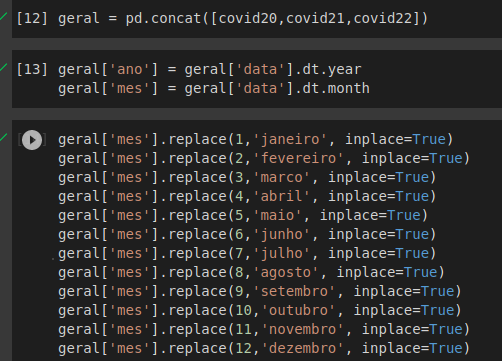
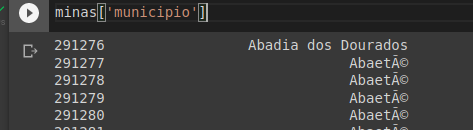


Figura 5 - Concatena as bases e transforma de número para nome do mês de registro das informações.

A figura 6, mostra a forma como alguns municípios foram digitados, por exemplo o Município de Abaeté, palavra acentuada que o sistema não reconhece tipo de caractere e importou utilizando caracteres não reconhecidos.



A Figura 6 Filtro mostra nome dos Municípios com caracteres não reconhecidos.

Outro filtro realizado, selecionando Município de Uberlândia, sem acento, e a coluna óbitos Acumulados, porém a aplicação não reconheceu e não trouxe resultado satisfatório.

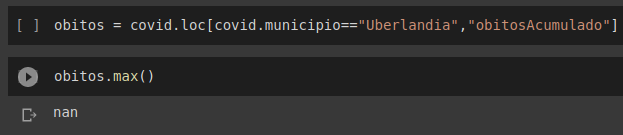


Figura 7 - Filtro realizado com nome do município sem acento

Na Figura 8 foi realizado o mesmo filtro, porém com nome do Município com acento, mas o sistema também não reconhece essa forma, assim também não mostra resultado satisfatório.

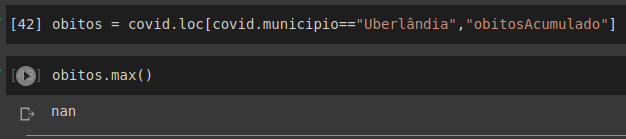


Figura 8 - Filtro realizado com nome do Município acentuado.

Já na Figura 9, houve comando para selecionar, na base de dados, para selecionar a coluna que contém código do IBGE do Município de Uberlândia, 317020.0, e substituir na coluna municipal, por Uberlandia, quando se permitiu realizar análise das informações.

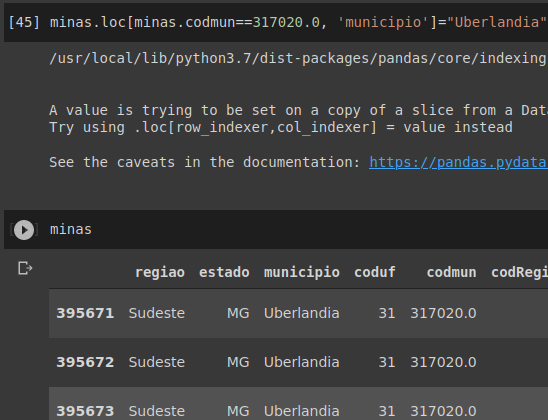
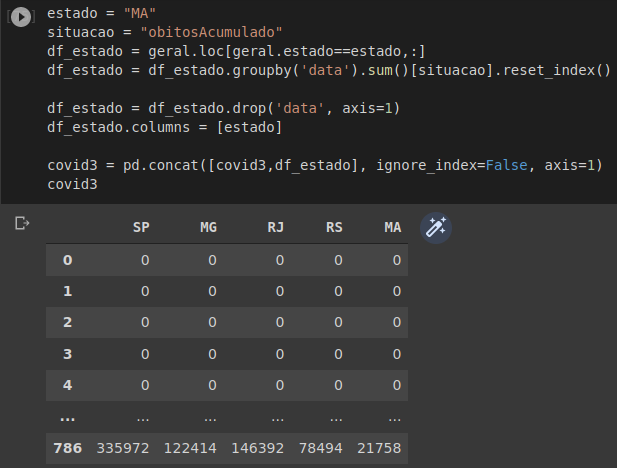


Figura 9 - substituir o nome do município através da coluna de código do IBGE.

Conforme explicação do Professor Sanderson de Macedo, apresentador do Canal do Sandeco, no Youtube, ensina como selecionar dados de óbitos acumulados por município e por estado dos dados, servido de base para realizar função da figura 10, 11, 12 e 13 entretanto a base utilizada nessa experiência foi retirada do site covid.saude.gov.br, assim como gráfico da figura 11.

Portanto uma mostra da potencialidade da aplicação que permite selecionar municípios, somar as informações, por data e mostrar resultados que auxiliarão na tomada de decisão, assim na figura 10, tem uma função que pode selecionar os estados, por data e óbitos acumulados, e após gerar a base de dados, aplicação também gera o gráfico ilustrando a evolução da pandemia, e nessa função conforme seleciona os estados, fica o resultados de todos assim como no gráfico também mostra evolução de todos.

Figura 10 - função que seleciona óbitos acumulados por data e estado.

Fonte: própria.

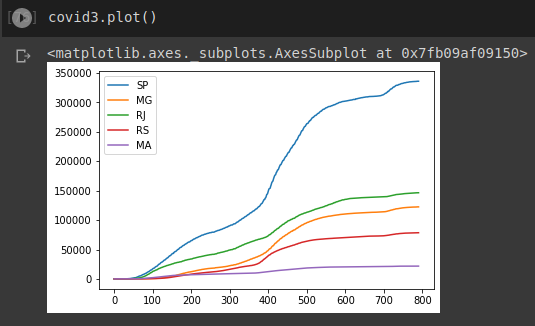


Figura 11 - Gráfico da Evolução do covid por estado, de 25/02/2020 a 25/04/2022.

Fonte: própria

Assim como na figura 10, a figura 11 mostra o gráfico referente a função gerada na figura 10, mostrando a evolução dos óbitos acumulados por data e estado, no período de 2020, 2021 e de janeiro a abril de 2022, estado de São Paulo, teve o maior número de óbitos, conforme a tabela abaixo:

Da mesma forma que Python pode gerar informações por estado, também é possível por município, através de filtro realizado selecionando por Município e óbitos acumulados, na Figura 12, observa-se a evolução dos óbitos nos respectivos municípios, porém com o problema de ter município digitado com aceto e a aplicação não reconhece esse tipo de caracter, impossibilitando seleção das informações, não localizando o Município de Patrocĩnio, que pode dar a entender não ter informação.

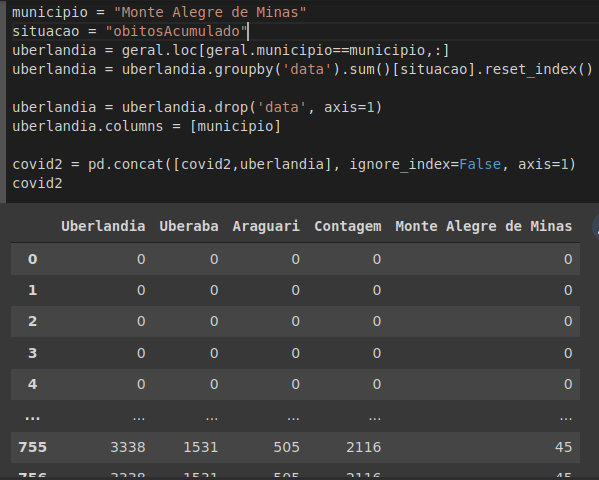
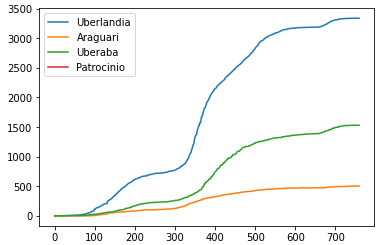


Figura 12 - Seleção dos óbitos acumulados por município.

Fonte Própria.

Fonte: Própria.Figura 13 - Gráfico dos óbitos acumulados por município

Fonte: própria

|  |  |
| --- | --- |
| **ESTADO** | **NÚMERO DE TOTAL DE ÓBITOS** |
| MARANHÃO | 10.879 |
| MINAS GERAIS | 61.227 |
| RIO DE JANEIRO | 73.233 |
| SÃO PAULO | 168.009 |

Tabela 3 - Total de óbitos por estado, período de 25/02/2020 a 25/04/2022.

Analisando a base de vacinação contra covid, base disponibilizada no site opendatasus.saude.gov.br, arquivo extenso, a base referente ao estado de Minas Gerais, são 3 arquivos de 6 gb de tamanho, principalmente por ser lançamento diário, a base tem como padrão letras maiúsculas e sem acento, que facilita construção dos filtros e análise dos dados, na Figura 14, a base foi importada para python, e os três arquivos foram concatenados, se tornando uma base e a Figura 15, a base já importada.



Figura 14 - Importando base de dados da vacinação.

Fonte Própria.

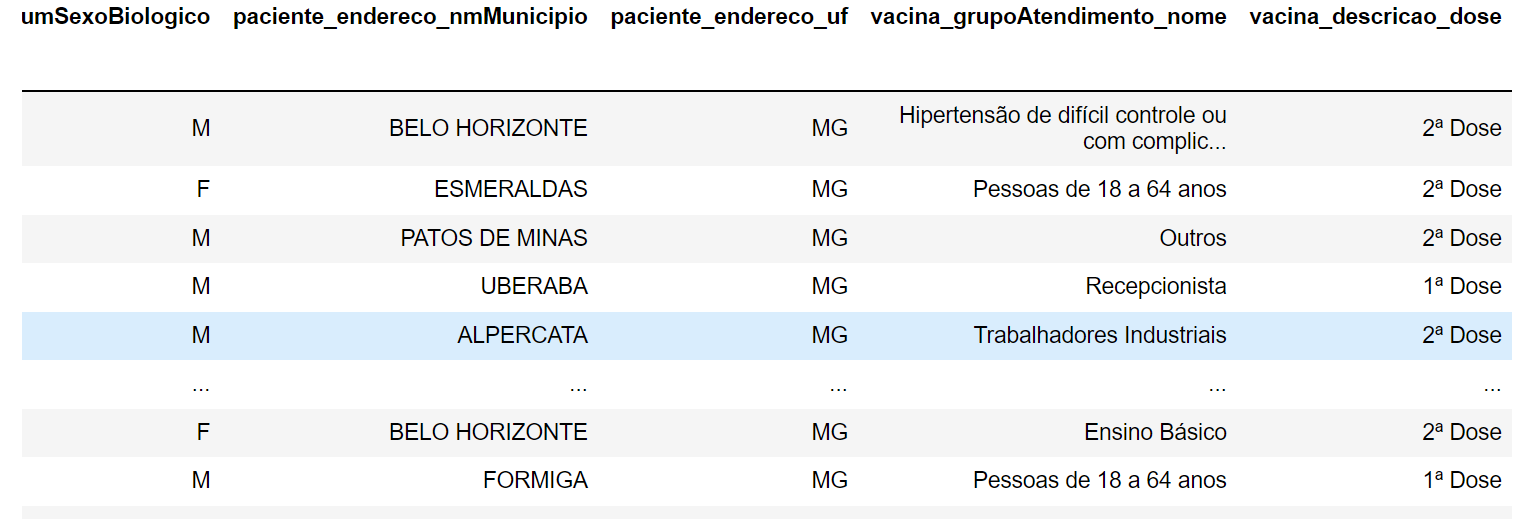
Já na Figura 13, mostra a base 

Figura 15 - Base de dados importada para python

Fonte Própria.

Figura 16 mostra filtro realizado de quantidade de vacinas aplicadas por dose em Minas Gerais.

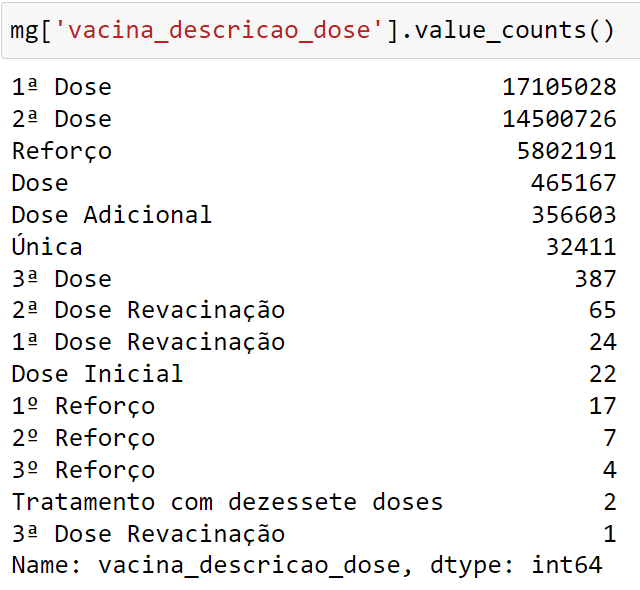


Figura 16 - Quantidade de vacinas aplicadas em Minas Gerais

Fonte Própria.

Analisando a Base de vacinas do covid19, observou-se, ao realizar filtro por data de aplicação da vacina, nome dos municípios e dose aplicada, aparece datas que não correspondem ao período de pandemia, sugerindo erro de digitação, exemplificados na figura 17.

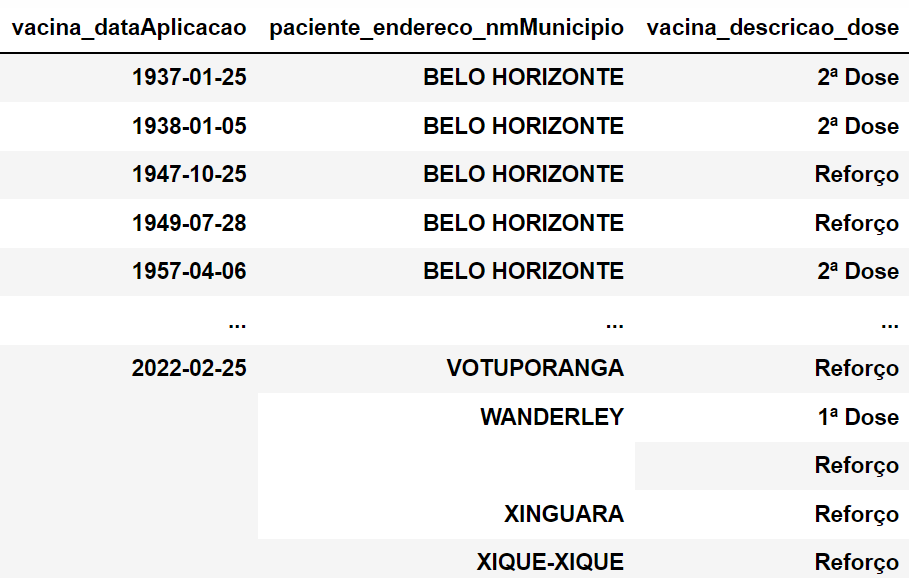


Figura 17 - Filtro realizado selecionando data de aplicação da vacina como índice.

Fonte Própria.

A aplicação, numa demonstração, de sua potencialidade, permite também contar quantidade de vacinas aplicadas por ano, por mês e por município, através do respectivo filtro, conforme a tabela 4, cujas informações retiradas da base de dados de vacinação contra covid19, base disponibilizada no site opendatasus.saude.gov.br.

|  |  |  |
| --- | --- | --- |
| **ANO** | **MÊS** | **QUANTIDADE DE DOSES APLICADAS** |
| 2021 | JANEIRO | 5.051 |
| FEVEREIRO | 21.449 |
| MARÇO | 39.364 |
| ABRIL | 88.839 |
| MAIO | 76.849 |
| JUNHO | 86.362 |
| JULHO | 158.070 |
| AGOSTO | 137.335 |
| SETEMBRO | 193.523 |
| OUTUBRO | 183.998 |
| NOVEMBRO | 111.885 |
| DEZEMBRO | 87.327 |
| 2022 | JANEIRO | 106.856 |
| FEVEREIRO | 90.948 |
| TOTAL | | 1.387.356 |

Tabela 4 - total de vacinas aplicadas em Uberlândia por ano e mês.

Fonte Própria.

**CONSIDERAÇÕES FINAIS**

Contudo apesar de ter aplicações que conseguem ler grandes base de dados como Python, Power B.I, etc, é necessário ter base de dados bem estruturada, através desse estudo de caso, observou-se que foram digitadas palavras acentuadas,nas bases de dados do covid19, o nome dos municípios ficaram com Primeira letra maiúscula e letras acentuadas, o Python ao importar a base de dados não reconhece esse tipo de caractere, dessa forma existe necessidade de padronizar, programar os sistemas que formam essas bases de dados registrem palavras com todas as letras maiúsculas e sem acento, esse foi o ponto positivo da base de dados da vacinação contra covid19, pois as palavras armazenadas já estavam em maiúsculo e sem acento.

Outro problema identificado não campos de valores ausentes, campos que não foram preenchidos e ao importar essa base de dados, o Python reconhece como campo vasinho denominando como Nan, e tais campos vazios atrapalham, na contabilidade dos casos de óbitos por covid19.

Também observou-se na base de Vacinação contra covid19, que foram digitadas datas, no campo de registro de data da aplicação da vacina, mas que pelo ano, sugere que usuário, talvez digitou data de nascimento do paciente em campo errado.

Portanto, uma base de dados bem estruturada, padronizada, facilita a análise dos dados, demonstrada através dos resultados de óbitos acumulados por município, por estado, total de óbitos por estado.

**REFERENCIA**

AMARAL, Fernando. **Aprenda Mineração de Dados: teoria e prática**. Alta Books, 2016.

AMARAL, Fernando. **Aprenda Mineração de Dados: teoria e prática**. Alta Books, 2016.

CANAL DO SANDECO. Analisando Covid19 no Brasil com Python e dados oficiais, disponível em [https://www.youtube.com/watch?v=30FijUpQ1po&t=1586s.](https://www.youtube.com/watch?v=30FijUpQ1po&t=1586s,) . Acesso em 23/12/2021.

CARVALHO, Luís A. V. de. **DATAMINING: A Mineração de Dados no Marketing,Medicina, Economia, Engenharia e Administração.** 2a Edição. São Paulo-SP: Érica, 2002.

conhecimento em dados de promoção à saúde. Revista Destaques Acadêmicos,

CORONAVÍRUS BRASIL. **Base de dados e informação sobre covid**. Disponível em:<https://covid.saude.gov.br>. Acesso em 22/02/2022.

CÔRTES, Sergio da Costa; PORCARO, Rosa Maria; LIFSCHITZ, Sergio. Mineração de Dados - Funcionalidade, Técnicas e Abordagens. PUC-INFO, 2002. Disponível em ftp://obaluae.inf.puc-rio.br/pub/docs/techreports/02\_10\_cortes.pdf. Acesso em 26/05/2018.

De CASTRO, Leandro Nunes; FERRARI, Daniel Gomes. Introdução à Mineração de dados: conceitos básicos, algoritmos e aplicações. São Paulo: Saraiva, 2016.

GREGORY, Guilherme; PRETTO, Fabrício. Mineração de dados para descoberta de

GREGORY, Guilherme; PRETTO, Fabrício. Mineração de dados para descoberta de conhecimento em dados de promoção à saúde. Revista Destaques Acadêmicos, Lajeado, v. 8, n. 4, 2016. ISSN 2176-3070. Disponível em http://univates.br/revistas/index.php/destaques/article/viewFile/1234/1091. Acesso em 19/04/2018.

MCKINNEY, WES. **Python para Análise de Dados; Tratamento de dados com pandas, numpy e python**. São Paulo: Novatec Editora Ltda. 2018.

Mineração de Dados. Rio de Janeiro: Ciência Moderna, 2009.

OPENDATASUS. Campanha Nacional de Vacinação Contra Covid-19. Contexto O Ministério da Sáude, por meio do Sistema de informação ro Programa Nacional de Imunização. Disponível em: [https://opendatasus.saude.gov.br/https://covid.saude.gov.br](https://opendatasus.saude.gov.br/). Acesso em 22/02/2022.

PROGRAME PYTHON. Python PANDAS - como filtrar dados com LOC e ILOC. Disponível em:<https://www.youtube.com/watch?v=30FijUpQ1po&t=1586s>. Acessado em 12/01/2022.

PROGRAME PYTHON.Python PANDAS - **Como excluir linhas e colunas com Nan.** Disponível em:<https://www.youtube.com/watch?v=30FijUpQ1po&t=1586s>. Acesso em 15/12/2020.

SILVA, Leandro Augusta da; PERES, Sarajane Marques; BOSCARIOLLI, Clodis. **Introdução à Mineração de Dados. Com aplicações em r**. Rio de Janeiro, Elsevier Editora ltda, 2016.

SWEIGART, AL. A**utomatize tarefas maçantes com python: Programação prática para verdadeiros iniciantes**. São Paulo: Novatec Editora Ltda. 2015.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. **Introdução ao Datamining: Mineração de Dados**. Rio de Janeiro: Ciência Moderna, 2009.