Relatório – Case Técnico de Dados Nexodata

**Demanda**: O time de epidemiologia da Nexodata fez o envio de datasets e quer saber quais são as regiões pelo Brasil, na granularidade de cidade, que apresentam o maior crescimento na taxa de mortalidade por COVID-19 no primeiro trimestre de 2021. E quer saber quais as características em comum dessas cidades que justifiquem tal variação da taxa de mortalidade.

Compreendido a demanda, as etapas para realizar a análise requisitada serão divididas da seguinte forma:

- 1. Entendimento dos datasets disponibilizados e suas características

- 2. Tratamento dos dados

- 3. Análise exploratória

- 4. Interpretação dos resultados

- 5. Criação de um repositório no GITHUB

1. Entendimento dos datasets disponibilizados e suas características

Será utilizado o editor de código Jupyter Notebook (linguagem Python) para a análise inicial dos datasets e as bibliotecas os, pandas e gzip. O script ‘Tratamento.jpynb’ deve estar na mesma pasta onde os datasets se encontram para seu devido funcionamento.

Os datasets podem ser acessados através do link <https://brasil.io/dataset/covid19/files/>, onde nele contém 4 arquivos ‘.csv’, que se encontram compactados no formato ‘.gz’:

* ‘boletim’: dataset com o histórico de origem das informações coletadas para a elaboração dos outros datasets, cujas colunas são:
  + date: data de publicação do boletim
  + notes: comentários sobre a publicação
  + state: estado do Brasil ao qual a publicação se refere
  + url: link de acesso do boletim

Para fins de análise, este dataset não possuí nenhuma aplicação prática e objetiva para responder a demanda requisitada, portanto ele será desconsiderado da análise.

* ‘caso’: dataset com informações resumidas de casos de COVID-19 no Brasil, cujas colunas são:
  + date: data de coleta dos dados
  + state: estado do Brasil dos dados
  + city: cidade do Brasil dos dados
  + place\_type: o tipo do local, se é uma cidade ou estado
  + confirmed: quantidade acumulada de casos confirmados de COVID-19
  + deaths: quantidade acumulada de mortes por COVID-19
  + order\_for\_place: numeração da linha
  + is\_last: informação booleana que distingue a última informação coletada da data dos dados
  + estimated\_population\_2019: população estimada do ano de 2019
  + estimated\_population: população estimada do ano corrente
  + city\_ibge\_code: código IBGE da cidade
  + confirmed\_per\_100k\_inhabitants: número de casos confirmados por 100 mil habitantes
  + death\_rate: taxa de mortalidade
* ‘caso\_full’: dataset com informações completas de casos de COVID-19 no Brasil, cujas colunas são:
  + city: cidade do Brasil dos dados
  + city\_ibge\_code: código IBGE da cidade
  + date: data de coleta dos dados
  + epidemiological\_week: semana epidemiológica da coleta dos dados
  + estimated\_population: população estimada do ano corrente
  + estimated\_population\_2019: população estimada do ano de 2019
  + is\_last: informação booleana que distingue a última informação coletada da data dos dados
  + is\_repeated: informação booleana que define se o dado está sendo repetido
  + last\_available\_confirmed: quantidade acumulada de últimos casos confirmados de COVID-19
  + last\_available\_confirmed\_per\_100k\_inhabitants: quantidade acumulada de últimos casos confirmados por 100 mil habitantes
  + last\_available\_date: última data da coleta dos dados
  + last\_available\_death\_rate: última taxa de mortalidade da coleta dos dados
  + last\_available\_deaths: última quantidade acumulada de mortes da coleta dos dados
  + order\_for\_place: numeração da linha
  + place\_type: o tipo do local, se é uma cidade ou estado
  + state: estado do Brasil dos dados
  + new\_confirmed: número de novos casos confirmados de COVID-19
  + new\_deaths: número de novas mortes por COVID-19

Devido ao fato do dataset ‘caso\_full’ ter dados mais completos, contendo o período de 2021 requisitado na demanda, ele será utilizado para a análise definitiva. Sendo assim, o dataset ‘caso’ será desconsiderado.

* ‘obito\_cartorio’: dataset com informações, de acordo com os cartórios, do tipo e volume das causas de óbito por COVID-19, cujas colunas são:
  + date: data de coleta dos dados
  + state: estado do Brasil dos dados
  + epidemiological\_week\_2019: semana epidemiológica da coleta dos dados de 2019
  + epidemiological\_week\_2020: semana epidemiológica da coleta dos dados de 2020
  + deaths\_indeterminate\_2019: número acumulado de mortes indeterminadas em 2019
  + deaths\_respiratory\_failure\_2019: número acumulado de mortes por falha respiratória em 2019
  + deaths\_others\_2019: número acumulado de mortes por outros motivos em 2019
  + deaths\_pneumonia\_2019: número acumulado de mortes por pneumonia em 2019
  + deaths\_septicemia\_2019: número acumulado de mortes por sepse em 2019
  + deaths\_sars\_2019: número acumulado de mortes por SARS-CoV-2 em 2019
  + deaths\_covid19: número acumulado de mortes por COVID-19
  + deaths\_indeterminate\_2020: número acumulado de mortes indeterminadas em 2020
  + deaths\_respiratory\_failure\_2020: número acumulado de mortes por falha respiratória em 2020
  + deaths\_others\_2020: número acumulado de mortes por outros motivos em 2020
  + deaths\_pneumonia\_2020: número acumulado de mortes por pneumonia em 2020
  + deaths\_septicemia\_2020: número acumulado de mortes por sepse em 2020
  + deaths\_sars\_2020: número acumulado de mortes por SARS-CoV-2 em 2020
  + deaths\_total\_2019: número acumulado de mortes total em 2019
  + deaths\_total\_2020: número acumulado de mortes total em 2020
  + new\_deaths\_indeterminate\_2019: número de novas mortes indeterminadas em 2019
  + new\_deaths\_respiratory\_failure\_2019: número de novas mortes por falha respiratória em 2019
  + new\_deaths\_others\_2019: número de novas mortes por outros motivos em 2019
  + new\_deaths\_pneumonia\_2019: número de novas mortes por pneumonia em 2019
  + new\_deaths\_septicemia\_2019: número de novas mortes por sepse em 2019
  + new\_deaths\_sars\_2019: número de novas mortes por SARS-CoV-2 em 2019
  + new\_deaths\_covid19: número de novas mortes por COVID-19
  + new\_deaths\_indeterminate\_2020: número de novas mortes indeterminadas em 2020
  + new\_deaths\_respiratory\_failure\_2020: número de novas mortes por falha respiratória em 2020
  + new\_deaths\_others\_2020: número de novas mortes por outros motivos em 2020
  + new\_deaths\_pneumonia\_2020: número de novas mortes por pneumonia em 2020
  + new\_deaths\_septicemia\_2020: número de novas mortes por sepse em 2020
  + new\_deaths\_sars\_2020: número de novas mortes por SARS-CoV-2 em 2020
  + new\_deaths\_total\_2019: número de novas mortes totais em 2019
  + new\_deaths\_total\_2020: número de novas mortes totais em 2020

Como o dataset ‘obito\_cartorio’ possui somente dados de datas entre 2019 e 2020, ele não poderá ser utilizado para tirar conclusões definitivas para o período de 2021, porém ainda pode ser usado para determinar algumas tendências em regiões de cada estado.

1. Tratamento dos dados

Continuando com o uso do editor de código Jupyter Notebook, o tratamento de dados foi somente realizado no dataset ‘caso\_full’.

De início, foi criado uma variável df\_caso\_full que recebe o dataframe descompactado com o uso da biblioteca pandas. Em seguida, foram retiradas as linhas onde na coluna ‘is\_repeated’ continham a informação booleana ‘True’, que mostram que os dados são repetidos, portanto não há proposito de mantê-los na tabela. A linha de código utilizada foi:

df\_caso\_full = df\_caso\_full.drop(df\_caso\_full[df\_caso\_full['is\_repeated'] == True].index)

E logo após a execução da linha de código acima, a coluna foi excluída por não ter informações analíticas pertinentes. A linha de código usada foi:

df\_caso\_full = df\_caso\_full.drop('is\_repeated', axis=1)

Como a granularidade deve ser de cidade, a coluna ‘city’ não pode conter nenhum campo vazio. Para realizar a exclusão de todas as linhas sem o campo preenchido, a linha de código usada foi:

df\_caso\_full = df\_caso\_full.dropna(subset=['city'])

Feito o entendimento anterior dos datasets, algumas colunas acabam não sendo pertinentes à análise, como as colunas:

* ‘city\_ibge\_code’ (informação sem importância analítica)
* ‘is\_last’ (não há necessidade de saber qual a última data de coleta dos dados)
* ‘last\_available\_date’ (é exatamente igual a coluna ‘date’)
* ‘place\_type’ (como a análise será somente de cidades, esta coluna não tem mais propósito)
* ‘estimated\_population\_2019’ (estimativas de anos diferentes de 2021 não são pertinentes à análise)

Portanto estas colunas foram excluídas, e a linha de código utilizada foi:

colunas\_drop = ['city\_ibge\_code', 'is\_last', 'last\_available\_date', 'place\_type', 'estimated\_population\_2019']

for coluna in colunas\_drop:

df\_caso\_full = df\_caso\_full.drop(coluna, axis=1)

Com o tratamento dos dados contidos na variável ‘df\_caso\_full’, será feito a extração compactada do dataframe com a extensão ‘.csv.gz’, usando a linha de código:

df\_caso\_full.to\_csv('novo\_caso\_full.csv.gz', sep=',', index=False, compression='gzip')

\*Nesta etapa é importante se atentar ao separador ‘sep=’,’’, pois, dependendo do idioma das plataformas de visualização (português ou inglês), o separador pode ser confundido com o separador de decimais (‘,’ em português e ‘.’ em inglês). Recomenda-se trocar o separador para ‘sep=’;’’, caso não saiba qual o idioma da plataforma de visualização.

Por fim, agora o dataset tratado e salvo como 'novo\_caso\_full.csv.gz' está pronto para a análise exploratória.

1. Análise exploratória

A ferramenta escolhida para visualização foi o Power BI, por ser a ferramenta que possuo maior conhecimento e por ter dashboards visuais limpos.

Para acrescentar o arquivo ‘novo\_caso\_full.csv.gz’, sem precisar descompactar e manter o arquivo ‘leve’, deve-se acrescentar uma nova *Blank Query*, depois abrir o Advanced Editor e colar o seguinte código:

let

#"novo\_caso\_full csv" = let

Source = Binary.Decompress(

File.Contents(

"C:\Users\[**local completo de onde encontra-se o arquivo ‘novo\_caso\_full.csv.gz’**]"),

Compression.GZip),

#"Imported CSV" = Csv.Document(Source,

[Delimiter=",",Encoding=1252])

in

Source

Depois é só dar ‘Ok’/’Done’ e a tabela já poderá ser visualizada.

Em seguida foi criado uma tabela calendário auxiliar, com o propósito de cobrir todo o intervalo de datas existente no dataset. Para tal foi criado um outro *Blank Query* e criado as seguintes variáveis:

= List.Min(novo\_caso\_full[date]) #com nome DataMin

= List.Max(novo\_caso\_full[date]) #com nome DataMax

= Duration.Days(DataMax-DataMin)+1 #com nome QtdDias

= List.Dates(DataMin, QtdDias, #duration(1,0,0,0))

Depois basta clicar em ‘Para a tabela’, no canto esquerdo superior e a tabela estará pronta para ser correlacionada no Model View.

Como somente o primeiro trimestre de 2021 será analisado, para reduzir o tamanho do arquivo e deixar as manipulações mais rápidas, o período entre 01/01/2021 e 31/03/2021 foi filtrado e o restante foi deletado.

Devido ao fato de existirem muitas cidades no Brasil, seria praticamente inviável e improdutivo a análise particular de cada uma. E há o fato de existirem cidades muito pequenas e que podem acabar superestimando métricas baseadas na população de cada cidade, portanto foi determinado o uso das seguintes métricas/medidas:

* Total Mortes: total de mortes por COVID-19

Total Mortes = SUM(novo\_caso\_full[new\_deaths])

* Total Casos Confirmados: total de casos confirmados de COVID-19

Total Casos Confirmados = SUMX(novo\_caso\_full, novo\_caso\_full[new\_confirmed])

* Taxa de Mortalidade: taxa de mortalidade real do COVID-19

Taxa de Mortalidade = IFERROR(SUM(novo\_caso\_full[new\_deaths])/SUM(novo\_caso\_full[new\_confirmed]), 0)

* População Estimada por Cidade: população estimada não cumulativa, que pode ser filtrada por cidade

População Estimada por Cidade = SUMX(VALUES(novo\_caso\_full[city]),AVERAGE(novo\_caso\_full[estimated\_population]))

* Taxa Mortes por População: taxa de mortes real pela população, que pode ser filtrada por cidade

Taxa Mortes por População = IFERROR(sum(novo\_caso\_full[new\_deaths])/[População Estimada por Cidade],0)

* Taxa Confirmados por População: taxa de casos confirmados real de COVID-19, que pode ser filtrada por cidade

Taxa Confirmados por População = IFERROR(SUM(novo\_caso\_full[new\_confirmed])/[População Estimada por Cidade],0)

* Taxa de Mortalidade Dia Anterior: para cálculo posterior da variação da taxa de mortalidade

Taxa de Mortalidade Dia Anterior = IFERROR(CALCULATE([Taxa de Mortalidade],DATEADD('Calendário'[Data],-1,DAY)), 0)

* Delta Taxa de Mortalidade: diferença entre a taxa de mortalidade do dia seguinte com o anterior

Delta Taxa de Mortalidade = [Taxa de Mortalidade]-[Taxa de Mortalidade Dia Anterior]

* Variação Taxa de Mortalidade: calcula a variação percentual da taxa de mortalidade

Variação Taxa de Mortalidade = IFERROR([Delta Taxa de Mortalidade]/[Taxa de Mortalidade Dia Anterior],0)

* População Estimada Absoluta: para cálculo posterior da taxa mortes por população total

Populaçao Estimada Absoluta = CALCULATE([População Estimada por Cidade],ALL('novo\_caso\_full'))

* Taxa Mortes por População Total: taxa de mortes pela população total, que pode ser filtrada por cidade

Taxa Mortes por População Total = IFERROR(sum(novo\_caso\_full[new\_deaths])/[Populaçao Estimada Absoluta],0)

Com essas métricas é possível criar filtros de análises que dão relatividade tanto para casos em que as cidades forem pequenas, quanto para cidades grandes. Portanto, podendo se basear na porcentagem, foi definido a filtragem por cidades com maiores taxas de casos confirmados de COVID-19 (‘Taxa Confirmados por População’) e cidades com maiores taxas de mortes (‘Taxa Mortes por População’), assim é possível verificar em quais cidades os casos de COVID-19 estão sendo mais agravantes.

Portanto, como a demanda é verificar quais cidades apresentam o maior crescimento na taxa de mortalidade, o filtro ‘Variação da Taxa de Mortalidade’ será usado para observar quais cidades tiveram maior crescimento no período do primeiro trimestre de 2021.

1. Interpretação dos resultados