1. **Ciência de dados ( Data Science )**

O protagonista de toda a revolução tecnológica da ciência da computação desde o primeiro computador até os microdispositivos cada vez mais miniaturizados da modernidade é o dado. Grande parte da evolução na era digital se dá pela maior e melhor capacidade de armazenamento e processamento de dados. Se há poucas décadas o processamento de dados era privilégio de raros e gigantescos equipamentos, na atualidade existe uma abundância na produção, armazenamento e consequente processamento desses dados. A expressão “Data Science” remonta aos anos 70, sendo uma ciência nova que pode ser definida como os processos, modelos e tecnologias que estudam os dados em todo o seu ciclo de vida: desde a reprodução até o descarte ( AMARAL, 2016)

Podemos entender a Ciência de Dados como uma ciência interdisciplinar que incorpora elementos, técnicas e teorias variadas vindas de muitos campos básicos da engenharia e ciências básicas, dessa forma:

(...) associado a este espírito de aplicação interdisciplinar, a ciência de dados emerge como componente cada vez mais importante nas mais diversas áreas, tais como saúde, petróleo, energia, financeira, esporte, astronomia, bioinformatica, Internet, mobilidade urbana, defesa cibernética, comunicação móvel e biodiversidade, apenas para mencionar algumas (PORTO e ZIVIANI, 2014)

Assim, em sua aplicação a diferentes áreas a Ciência de dados lida também com diferentes tipos de dados, entretanto para fins de análise deste presente artigo, será abordado exclusivamente o dado retangular, sendo esse: uma coleção de valores associada a uma variável e uma observação, geralmente em formato de tabelas (WICKHAM, HEADLEY)

Esses dados também são chamados dados tabulares ou do tipo planilha e são comumente o tipo de dados armazenados em bancos de dados relacionais ou em arquivos de texto delimitados por tabulação ou vírgula (McKINNEY).

1. **Aprendizado de Máquina (*Machine Learning*) e Redes Neurais Artificiais**

O aprendizado de máquina usa uma variedade de algoritmos que interativamente aprendem com os dados a fim de melhorar a descrição e previsão de dados futuros. Conforme há a interação entre esses algoritmos e os dados de treinamento, se torna possível produzir modelos mais precisos. Dessa forma, um modelo de aprendizado de máquina é a saída gerada quando há o treino de uma *machine learning*. Assim, após a criação de um modelo, torna-se possível prever as saídas conforme o estabelecimento de diversos dados de entrada (HURWITZ, KIRSCH)

As Redes Neurais Artificiais (RNA) são uma forma de modelagem abstrata e simplificada dos neurônios humanos onde o processo de aprendizado é simulado. Para que o aprendizado ocorra é necessária a utilização de um conjunto de dados de exemplos de padrões com as respostas esperadas. Comumente, há uma divisão nessa base de dados em partes destinadas ao aprendizado e partes destinadas à validação, sendo esse o tipo mais comum de aprendizado de máquina o Aprendizado Supervisionado (HEINEN, OSÓRIO)

**2.1 Aprendizado Supervisionado**

A principal característica de sistemas de ML baseados em aprendizado supervisionado é a presença da resposta desejada nos dados usados para a criação dos modelos. Diz-se então, que os dados são anotados com as respostas ou classes a serem previstas. Conforme Bianchi, “ Nesta abordagem o modelo aprende a executar uma tarefa a partir de exemplos rotulados, ou seja, a partir das respostas corretas, que de alguma forma devem ser passadas para o modelo.“ Entre diversos tipos de aprendizado supervisionado temos:

**2.1.1 Regressão Linear Simples / Múltipla**

A regressão nasce da tentativa de relacionar um conjunto de observações de certas variáveis. Conforme Souza, regressão é ato de prever dados futuros à partir de dados históricos. Para a Estatística, o termo trata-se de Análise de Regressão que estuda a relação entre duas ou mais variáveis, sendo uma dependente de outra ou várias. Historicamente temos que:

“O termo ‘regressão’ foi proposto pela primeira vez por Sir Francis Galton em 1885 num estudo onde demonstrou que a altura dos filhos não tende a reflectir a altura dos pais, mas tende sim a regredir para a média da população. Actualmente, o termo “Análise de Regressão” define um conjunto vasto de técnicas estatísticas usadas para modelar relações entre variáveis e predizer o valor de uma ou mais variáveis dependentes (ou de resposta) a partir de um conjunto de variáveis independentes (ou preditoras).” (Maroco, 2003)

Matematicamente a Regressão Linear Simples se trata de uma função do primeiro grau  *ŷ = ax + b* , tal que “ a “ é igual ao coeficiente linear sendo o valor de ŷ para o x = 0, “ b” igual ao coeficiente angular responsável por expressar o acréscimo em y para um acréscimo de uma unidade em x, ŷ a variável dependente e “x” a variável independente. A variável independente trata-se da variável preditora, a qual tem o poder de influenciar na variável alvo ou variável dependente. O Modelo de regressão Linear múltiplo assume uma relação linear entre a variável dependente e várias variáveis independentes ( x1, x 2… xp)

Outro importante coeficiente relacionado à regressão linear é o R² ou coeficiente de correlação linear de Pearson trata-se do coeficiente de determinação e é capaz de dizer o grau de assertividade tem um modelo linear, ou seja, diz o percentual de variação de y que é explicado pela variação de x (RODRIGUES, 2012).

**2.1.2 Árvores de Decisão**

Para a computação, de modo geral, as árvores são estruturas de dados organizadas em um conjunto de elementos capazes de armazenar informações chamados de nós, por sua vez toda árvore possui um nó raiz o qual possui o mais alto nível hierárquico sendo esse o ponto de partida. Além disso, cada novo nó pode possuir seus próprios nós filhos, ao passo que dá-se o nome de nó folha ou terminal ao nó que não possui nenhum filho. Assim, especificamente, uma Árvore de Decisão trata-se de uma estrutura de árvore que armazena regras em seus nós, dessa forma, cada nó ou folha representa uma decisão a ser tomada ( CAMPOS, 2017)

**2.2 Aprendizado Não Supervisionado**

Quando a obtenção de dados anotados se torna difícil, custosa ou impossível uma das soluções é deixar que o modelo tente descobrir perfis e padrões a partir da correlação entre os dados apresentados. A falta de uma variável anotada faz com que seja difícil estabelecer o quão assertivo é um modelo baseado nessa técnica. Uma das técnicas mais comuns de aprendizado não supervisionado é por sinal o uso das redes neurais. Este modelo aprende a executar uma tarefa a partir de dados não-rotulados (sem um resultado conhecido), apenas com base em suas características e padrões semelhantes, ou seja, o modelo deduz estruturas a partir de uma amostra do problema; bem utilizado em situações com muitas observações e muitas features como: áudios, imagens e vídeos (BIANCHI, 2020)

**2.3 Aprendizagem por reforço**

A terceira abordagem de aprendizagem de máquinas é a chamada “aprendizagem por reforço”, em que a máquina tenta aprender qual é a melhor ação a ser tomada, dependendo das circunstâncias na qual essa ação será executada. Conforme Bianchi, nesse modelo de aprendizagem cada ação é avaliada com uma recompensa, funcionando da seguinte maneira: “ O agente realiza uma ação num dado ambiente, alterando seu estado inicial, o que gera uma recompensa ao agente” . Dessa forma, o paradigma desse tipo de aprendizagem pressupõe que se conheça todas as regras, mas não se saiba a melhor sequência de ações.

**3 Principais Ferramentas para exploração, limpeza e modelagem de dados:**

**3.1 Numpy**

A biblioteca Numpy é um conjunto de ferramentas em python que oferece um código que aglutina estrutura de dados, algoritmos, e outras bibliotecas essenciais para a grande parte das aplicações científicas que envolva dados numéricos em python. Seu nome é a abreviatura para *numerical python*  e contêm uma série de recursos como arrays multidimensionais, funções para processamento entre arrays[[1]](#footnote-0), ferramentas para ler e gravar conjuntos de dados baseados em arrays, operações de álgebra linear, transformadas de Fourier, geração de números aleatórios, dentre outros (McKINNEY).

**3.2 Pandas**

Biblioteca que oferece estruturas de dados de alto nível e funções visando o trabalho eficiente com dados tabulares. Possui dois tipos principais de objetos o *DataFrame (* estrutura de dados tabular orientada a colunas com rótulos) e as *Series* ( um array unidimensional com rótulo). De forma geral, o Pandas combina o processamento de arrays da NumPy com a manipulação de dados relacionais, disponibilizando funcionalidades de indexação a fim de facilitar a reformatação, manipulação, agregações e seleção de subconjuntos de dados. Entre as principais funcionalidades do Pandas temos: Estrutura de dados de eixos nomeados, séries temporais integradas, operações aritméticas e reduções que preservem metadados, tratamento para dados ausentes, combinações com outras operações relacionais. O nome Pandas é uma derivação de *Panel data* (dados em painel), além de ser uma brincadeira com *Python data analysis* (McKINNEY).

**3.3 Matplotlib**

Originalmente criada por John D.Hunter, a biblioteca Matplotlib é ferramenta python mais popular para visualização e plotagem de dados bidimensionais e possui uma boa integração com o ecossistema python (McKINNEY). A biblioteca Matplotlib ainda possui o módulo pyplot que é uma coleção de funções que tem forte inspiração no MATLAB ( Matrix Laboratory)[[2]](#footnote-1) .

**3.4 IPython e Jupyter**

Por mais que não oferece nenhuma ferramenta de análise de dados por si só, o IPython é uma das ferramentas mais importantes para a ciência de dados com python atualmente. Projetado para maximizar a produtividade, o IPython é um interpretador interativo que auxilia no processo de exploração de dados sem a necessidade de edição e compilação de código. Oferecendo acesso interativo com o shell do sistema operacional e do sistema de arquivos, ajuda a fazer testes e executar tarefas de forma mais rápida e produtiva. De forma análoga ao IPython, o projeto Jupyter levou a interatividade para a plataforma web com integração com linguagens de marcação e blocos de código executável o que faz do Jupyter uma ótima ferramenta para a documentação (McKINNEY).

**3.5 SciPy**

É uma coleção de pacotes que tem como objetivo a resolução de diversos problemas padrões no processamento científico, tais como: integração numérica e solucionadores de equações diferenciais, álgebra linear e decomposição de matrizes, otimizadores de funções e algoritmos para encontrar raízes, ferramentas de processamentos de sinais, solucionadores de matrizes esparsas e sistemas lineares esparsos, dentre outros (McKINNEY).

**3.6 Scikit-learn**

Criado em 2010, o scikit-learn se transformou no principal kit de ferramentas de aprendizado de máquina python e possuí submódulos focados em diversas funcionalidades do aprendizado de máquina, tais como: classificação com ferramentas de *random forest* (floresta aleatória), regressão logística, regressão de Lasso, regressão de ridge, k-means, clustering espectral, fatoração de matrizes, busca em grade, pré-processamento com extração de atributos, normalização, entre outros (McKINNEY).

**4 Data Wrangling: *Preparação e limpeza de dados sobre Covid19 disponibilizados pelo Governo Brasileiro***

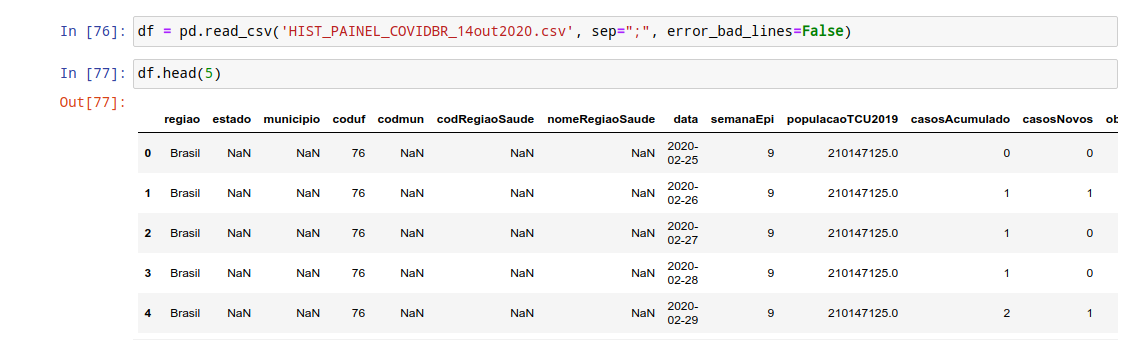
O primeiro passo para o trabalho com dados brutos é o processo comumente chamado na Ciência de dados de *Data Wrangling*(manipulação de dados). Nesse processo os dados são transformado para que possam entrar em modelos de predição e visualização. Esse processo é importante para a limpeza de dados faltantes e possíveis *outliers[[3]](#footnote-2).*

Será manipulada uma amostra diária da base de dados disponibilizada pelo governo no site covid.saude.gov.br, base essa que é atualizada diariamente e possui um painel interativo onde é possível dados detalhados da doença no país em um cenário tanto nacional como regional, assim o dataset disponibilizado possuí dados da maioria dos municípios nacionais além de estar separado por regiões.

Nosso objetivo é fazer com que esse dataset possa ser mais facilmente interpretado, retirando dele dados faltantes ou discrepantes, além de fornecer uma visão resumida das semanas epidemiológicas[[4]](#footnote-3) nas quais eles se encontra organizado. Esse processo é importante para a criação de futuros modelos de *machine learning* ( aprendizado de máquina) como forma de predição de cenários futuros da doença no país.

Dentro da plataforma disponibilizada pelo Governo, consegue-se facilmente baixar uma cópia dos dados no formato csv (*comma separated values -* valores separados por vírgulas) muito embora os valores venham separados por ponto e vírgula. Usando o *Jupyter Notebook* fazemos a importação da base de dados para dentro *kernel python* do nosso sistema:

Figura 1: importação de dados.

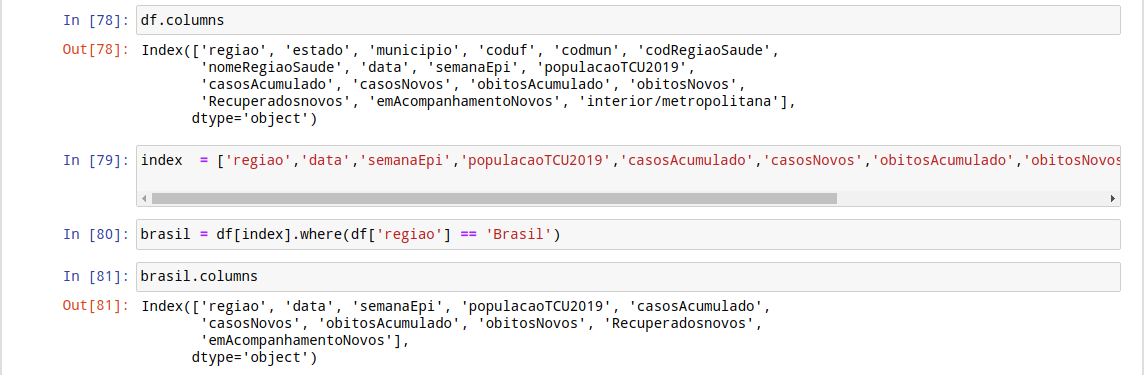


Fonte: elaborado pelo autor.

Como é possível perceber, temos uma estrutura de linhas e colunas. Na coluna “regiao” temos os dados divididos por todas as regiões do país e também em um contexto geral. Nesse primeiro momento, importa trabalhar apenas com os dados que envolvem a região Brasil por se tratar do somatório dos dados. Também é possível notar a existência massiva de dados inconclusos denominados “NaN” do inglês *not a number*, trata-se de uma definição genérica para dados que não foram possíveis de ser recuperados de alguma forma.

Em um segundo momento, interessa retirar não somente os dados faltantes como também colunas inteiras de informações que não dizem parte à visão macro das informações sobre a epidemia. Em outras palavras: se olharmos para os dados disponíveis, existem colunas inteiras de informações que não cabem a região Brasil, o que pode ser visto da descrição detalhada das colunas presentes do *dataset* discriminadas na figura.

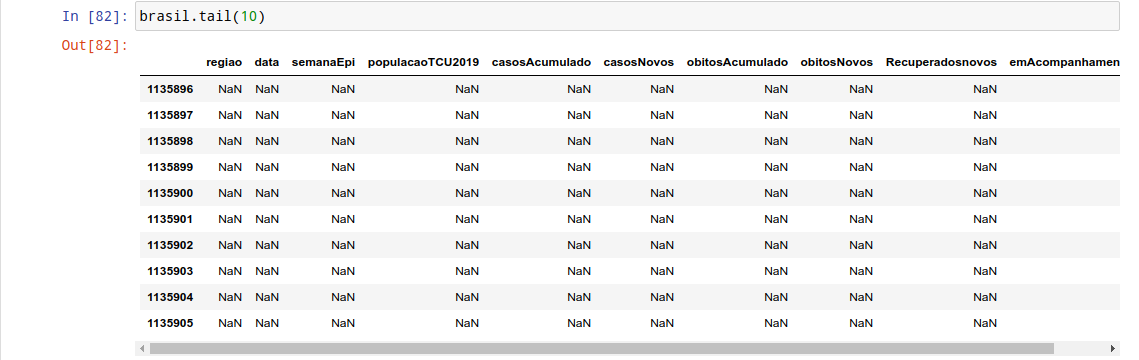
Figura 2: Exibição de colunas da base de dados.



Fonte: elaborado pelo autor.

Podemos ver que informações como “estado, municipio, coduf, populcaoTCU2019” entre outros são irrelevantes para nossa compreensão dos dados como um todo por dizerem respeito a regiões específicas. Assim, selecionamos apenas as colunas que são importantes para uma visão macro e retirando informações desnecessárias. Entretanto, ainda assim um volume muito grande de informações faltantes são retornadas por esse processo.

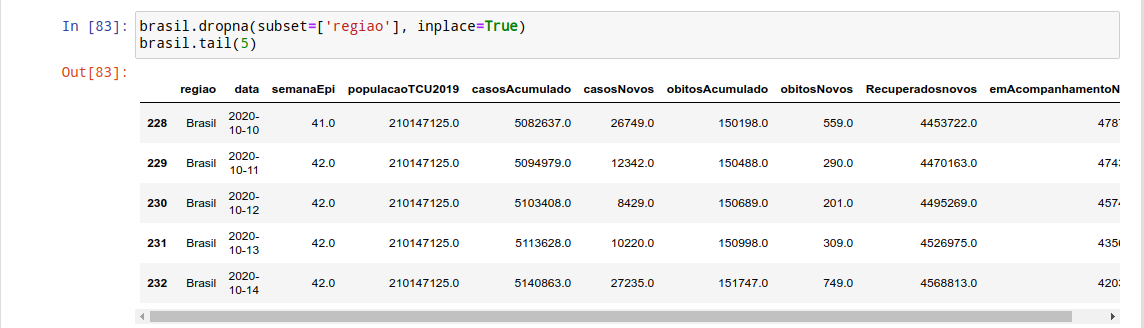
Figura 3: Retorno de dados faltantes.



Fonte: elaborado pelo autor.

Dito isso, podemos retirar todos dos dados inconclusos de nosso *dataset* deixando apenas as linhas que possuem dados completos, e printando as últimas 5 linhas de todo a base de dados.

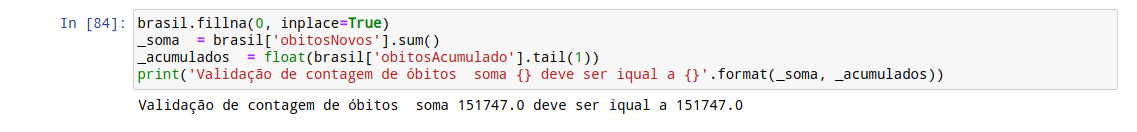
Figura 4: Filtragem de dados faltantes.



Fonte: elaborado pelo autor.

Entretanto, como podemos saber se em alguma parte do processo dados foram perdidos ou desorganizados? Resolvemos essa questão fazendo uma checagem simples de valores. Caso a base de dados não tenha sido desorganizada, o somatório do número de “ óbitos novos ” deve ser exatamente igual ao último registro da contagem geral de “óbitos acumulados”

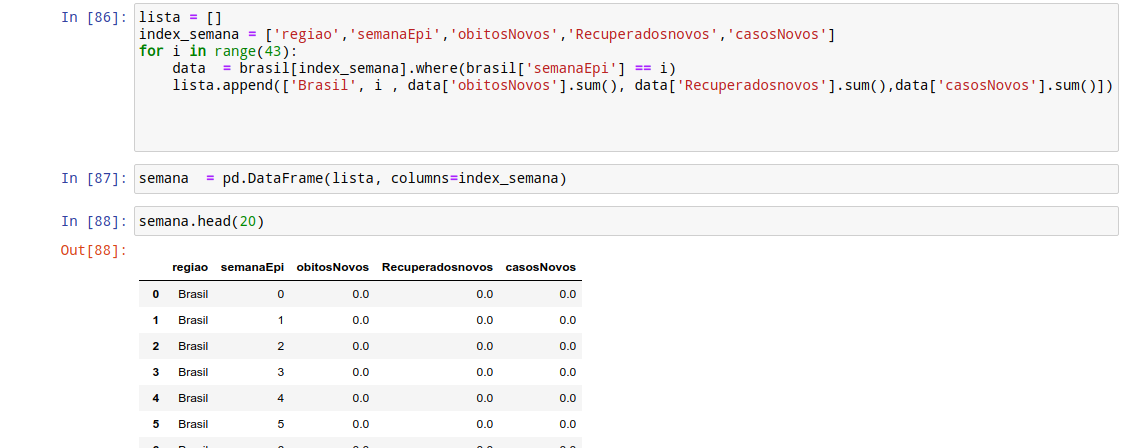
Figura 5: Teste de integridade do *dataset.*



Fonte: elaborado pelo autor.

Um último passo para a finalizarmos a formatação de nosso *dataset*  é unir os dias em semanas epidemiológicas. Podemos fazer isso a partir de um laço de repetição que irá iterar uma lista com a somatória de cada semana em formato de uma *Serie* do Pandas, posteriormente, unimos todos esses dados em um novo *Data Frame*  que chamaremos de “semanas” . Assim esse será nosso resultado final contendo dados específicos somente do Brasil como um todo e retornando as semanas epidemiológicas com os dados de todos os dias somados.

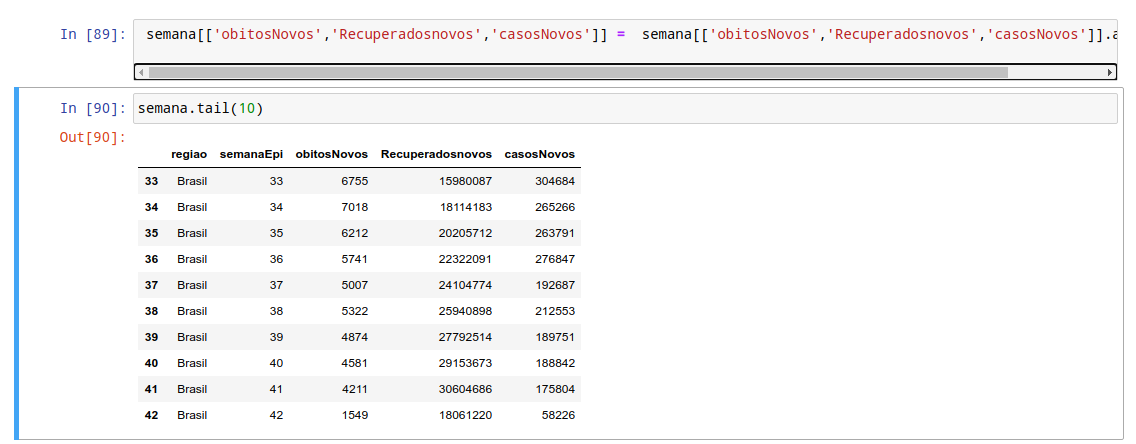
Figura 6: Organização dos dados pela somatória dos dias em semanas epidemiológicas.



Fonte: elaborado pelo autor.

Como resultado temos apenas 5 colunas em nosso *data frame* inicial com a semana epidemiológica representada em uma coluna sequencial que vai de 0 até 42 referente a data em que nossa base de dados foi coletada, dia 14 de outubro de 2020. Entretanto, para uma análise aprimorada de nossos dados, é importante que todos eles tenham o mesmo tipo, para que eventuais conflitos não ocorram. Assim, é possível observar pela figura anterior que os dados “obitosNovos”, “Recuperadosnovos” e “casosNovos” seguem o padrão de números de ponto flutuante. Contudo, é impossível que haja uma contagem de óbitos ou casos novos que ocorra em números não inteiros. Dessa forma, é recomendável que transformemos nossos dados para números inteiros.

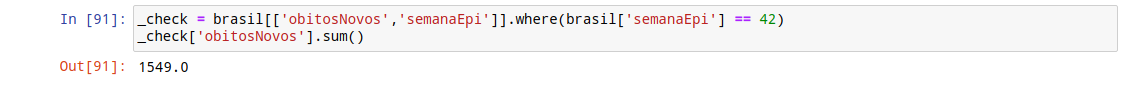
Figura 7: Conversão dados de ponto flutuante em inteiros.



Fonte: elaborado pelo autor.

Novamente, é recomendável que checamos se o processo manteve a integridade dos dados. Fazemos isso retornando a soma da última semana epidemiológica de nosso *dataframe* original. Esse valor deve ser correspondente com o último valor da coluna “obitosNovos”. Note que a saída que temos ainda é em ponto flutuante, isso porque vem exatamente de um conjunto de dados que não foi tratado.

Figura 8: Teste de integridade de dados.

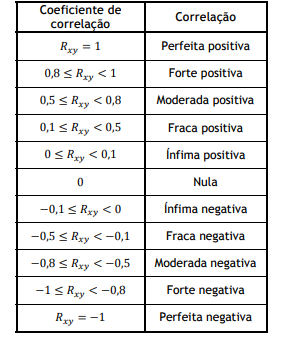


Fonte: elaborado pelo autor.

**5 Correlação e visualização de dados**

Tendo certeza da boa formatação de nossos dados e da sua integridade, podemos estabelecer uma série de correlações entre nossas diferentes variáveis. O ponto aqui é descobrir qual o nível de relacionamento existente entre diferentes variáveis e como o comportamento de uma afeta diretamente ou não a outra. De forma obvia a contagem de semanas epidemiológicas interfere diretamente na contagem dos dados e sua correlação com todas as outras variáveis é considerada forte positiva conforme vemos em Rodriques que explana os valores tabelados para o coeficiente de correlação de Pearson.

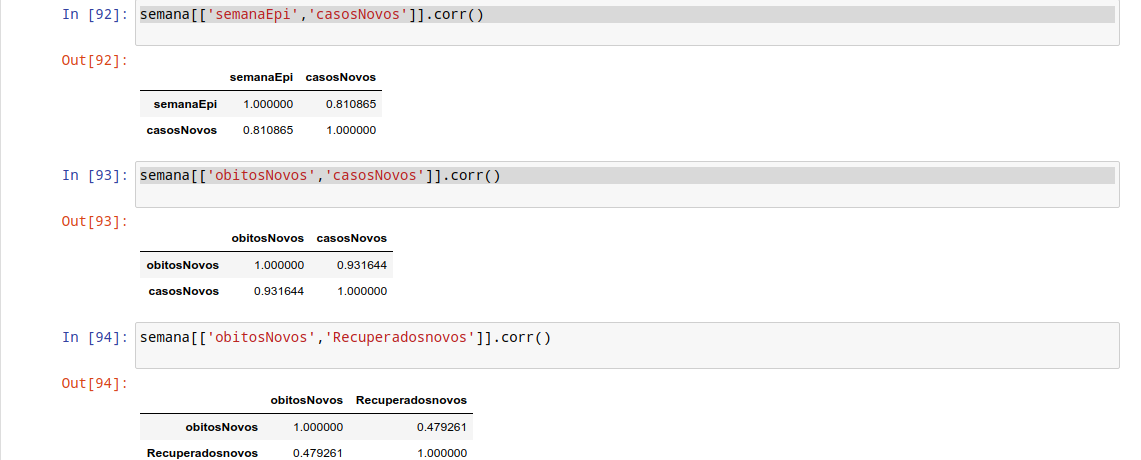
Figura 9: Tabela de correlação.



Fonte: RODRIGUES, 2012.

Por sua vez, podemos fazer uso de métodos da biblioteca Pandas para estabelecer o coeficiente correlacional de Pearson de nossas variáveis. O método “corr” tem como método padrão de correlação o método de Pearson, ao passo que também poderíamos obter as correlações de Kendall e Spearman alterando seus parâmetros. Assim, podemos perceber que há uma correlação forte positiva para “obitosNovos” em relação a “casosNovos” como tambem entre “semanaEpi” e “casosNovos” , mas de forma óbvia, temos uma relação fraca positiva para “obitosNovos” e “Recuperadosnovos”.

Figura 10: Teste de correlação entre dados.

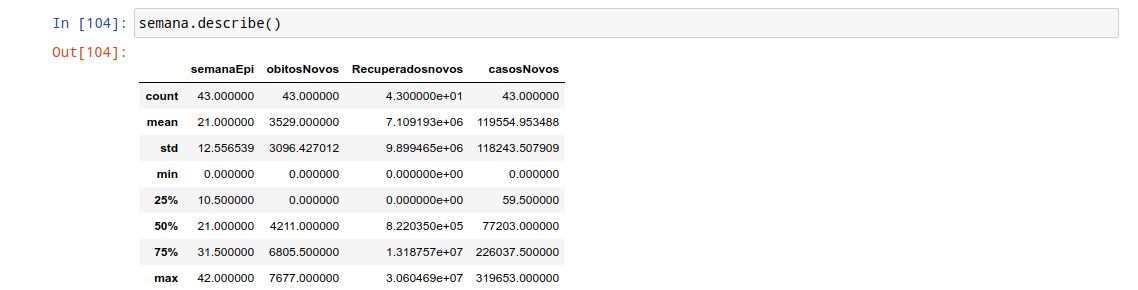


Fonte: elaborado pelo autor.

É extremamente importante levantar o nível de correlação porque será exatamente entre as variáveis com correlação mais forte que será possível a criação de um modelo de aprendizado de máquina ou da simples aplicação de regressão linear que será capaz de lançar previsões sobre cenários futuros. Dessa forma, como há uma relação intrínseca entre a semana epidemiológica e o número de casos novos é bem provável que um modelo treinado com esses dados como entrada poderá predizer com um bom nível de assertividade o número aproximado de novos casos conforme o tempo, isso levando em conta apenas os dados que temos e tomando apenas como preceito as informações que temos em mãos. De outra maneira, também podemos prever com um ótimo grau de assertividade o número de novos óbitos a partir do volume de contágio.

Dando continuidade a exploração dos dados, podemos ter em mãos uma série de dados descritivos de nosso dataset como média, máxima e mínima, e desvio padrão[[5]](#footnote-4) (representado aqui por std).

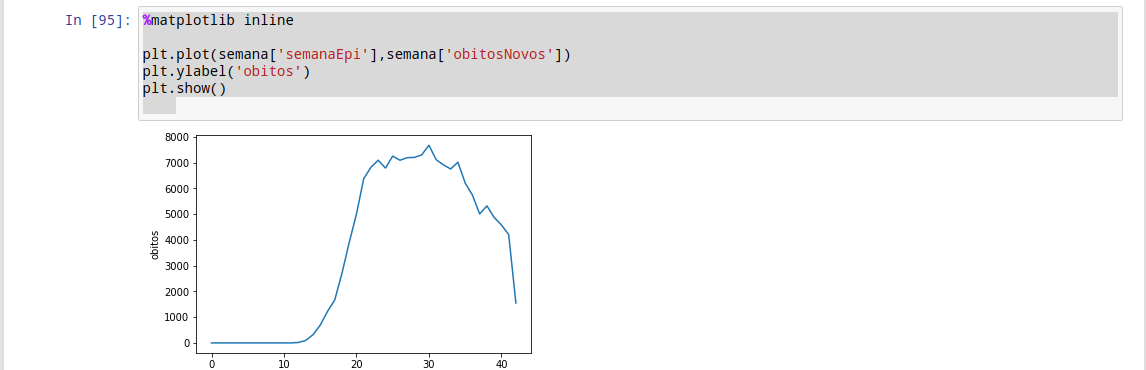
Figura 11: Descrição do *dataset* final.



Fonte: elaborado pelo autor.

O que significa que no estado atual da pandemia no Brasil temos uma média de quase 4 mil mortos por semana epidemiológica, com uma máxima de 7 mil mortos na trigésima semana. Entretanto, se olharmos para um gráfico de dispersão dos óbitos percebemos que há uma queda crescente no número de mortes e também no número de novos casos. Contudo, é necessário ressalvar que no dia da coleta dos dados não havia se passado uma semana inteira, assim a linha decrescente a partir da quadragésima semana não condiz necessariamente com os números totais. Ainda assim, vemos claramente uma queda massiva a partir do pico da semana 30 onde houve o maior número de mortes, entretanto em uma visão pouco otimista podemos pensar na subnotificação dos casos está afetando nossa visualização, pois parece pouco provável que essa queda tenha acontecido de forma tão repentina.

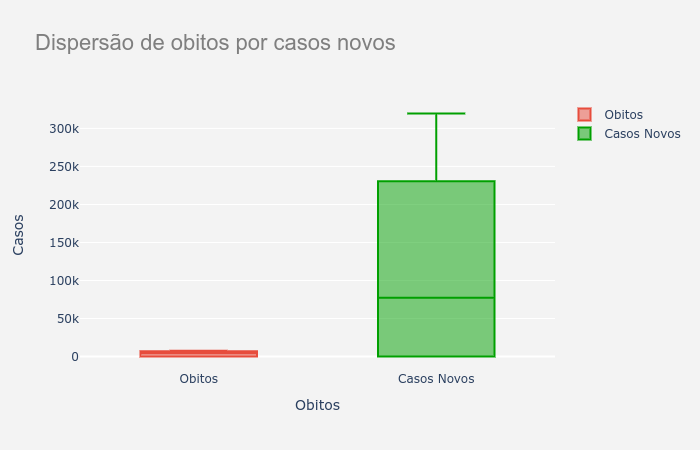
Figura 12: Gráfico de dispersão de óbitos por semana epidemiológica.



Fonte: elaborado pelo autor.

Outra importante forma de se visualizar nossos dados é fazendo uso de *boxplots*, pois a partir desse modelo de gráfico é possível perceber a presença *outliers* que extrapolam a média e podem prejudicar a construção de modelos de predição. Dessa forma, observando um gráfico de dispersão que relaciona os óbitos com os casos novos, podemos perceber de forma bem clara a taxa de letalidade do vírus na aproximação de 3% em relação a taxa de contágio expressa pelos casos novos. Não foi constatada a presença de *outliers* nos dados referentes aos casos novos e tão pouco nos dados de óbitos.

Figura 13: *boxplot* de óbitos e casos novos.



Fonte: elaborado pelo autor.

1. Trata-se de um tipo de representação de vetor de valores básicos: caracteres, inteiros, números de ponto flutuante. São tipos de sequência semelhantes a listas. (https://docs.python.org/pt-br/3/library/array.html) [↑](#footnote-ref-0)
2. Software interativo voltado para o cálculo numérico que integra a análise numérica, cálculo de matrizes, processamento de sinais e construção de gráficos.

   (https://pt.wikipedia.org/wiki/MATLAB) [↑](#footnote-ref-1)
3. trata-se de dados que são discrepantes da maioria dos outros dados presentes. [↑](#footnote-ref-2)
4. Por convenção internacional as semanas epidemiológicas são contadas de domingo a sábado. A primeira semana do ano é aquela que contém o maior número de dias de janeiro e a última a que contém o maior número de dias de dezembro. [↑](#footnote-ref-3)
5. representa a taxa de dispersão em torno da variável média. [↑](#footnote-ref-4)