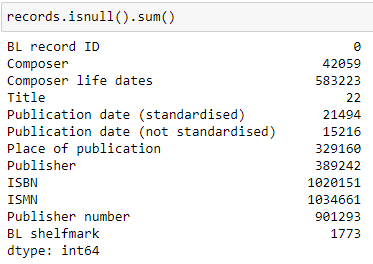
Com o intuito de treinar minhas habilidades de preparação de dados, fui até o Kaggle e escolhi uma biblioteca de dados aleatória para começar a treinar. Como gosto muito de música e acho que tenho um pouco mais de conhecimento sobre, escolhi o repositório *History of music (British Library)* mais especificamente o arquivo csv sobre as músicas em si (*records.csv*) que conta com 1.045.508 linhas na tabela. Criei um Jupyter notebook pela comand line do Anaconda e comecei a exploração.

Primeiramente importei todas as ferramentas que poderia usar, o arquivo csv e inspecionei suas primeiras linhas. Ao inspecionar suas colunas já percebi que todos os números de identificação, exceto *BL record ID* e *BL shelfmark*, eram muito inconsistentes, e isso se explica vendo de onde esses dados vieram, pois sabendo que são frutos de uma iniciativa de pesquisa da biblioteca britânica que faz sentido os identificadores relacionados a ela serem consistentes.

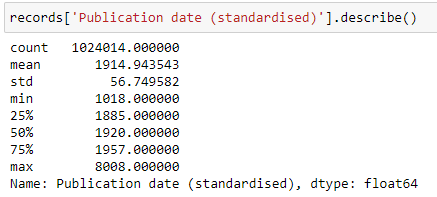
Continuando as etapas iniciais ao se explorar dados ( vendo os valores únicos de cada tabela, verificando variáveis categóricas, …), ao ver as instâncias nulas de cada coluna percebi como a parte das colunas referente aos números identificadores tinha muito valores nulos:



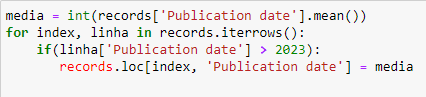
A coluna ISBN tendo 1.020.151 entradas nulas, o ISMN tendo 1.034.661 e Publisher number com 901293, mas como esses eram ID 's essas informações não tinha como se conseguir. Passando para as próximas colunas percebi como algumas coisas eu podia relacionar para conseguir mais informações:

* Se o atributo da data de publicação de uma linha for nulo mas a data não padronizada não for, eu posso tentar transformar a informação da data não padronizada em uma data de publicação comum
* Pelo lugar de publicação, pode se conseguir quem publicou por comparar com os que mais publicam naquele lugar nas linhas em que essas informações existem
* E sabendo o compositor tem como saber o período de vida dele

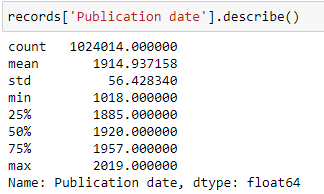
Primeiramente comecei pela etapa da data de publicação pois ela é uma variável numérica mais simples de se trabalhar e mais quantificável no processo de avaliação da eficácia do que seja lá o que eu for fazer. Vendo uma descrição da coluna é possível retirar mais algumas conclusões:



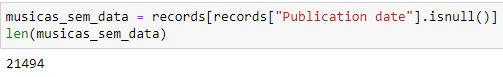
A média de ano de publicação é 1914, as músicas aqui guardadas vão de 1018 até o ano de…8008??? Assim já sabia por onde começar, retirando *outliers* Mas antes estava cansado de digitar esse nome de coluna gigante então mudei ele para 'Publication date'. Agora voltando ao processo anterior:



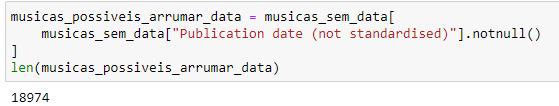
E checando novamente:



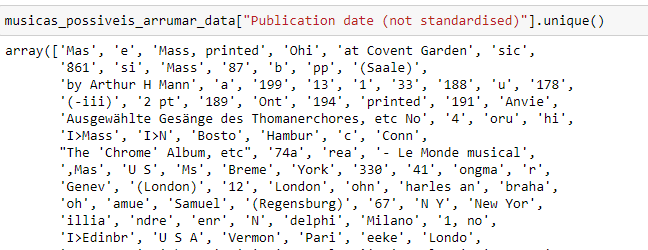
Com isso feito, queria trocar o tipo dessa coluna para um int64 já que não fazia sentido um ano fixo ser um float64, mas ao tentar fazer isso me lembrei dos valores nulos, e como não queria já substituir eles só para mudar seu tipo resolvi trabalhar com os valores nulos primeiro.



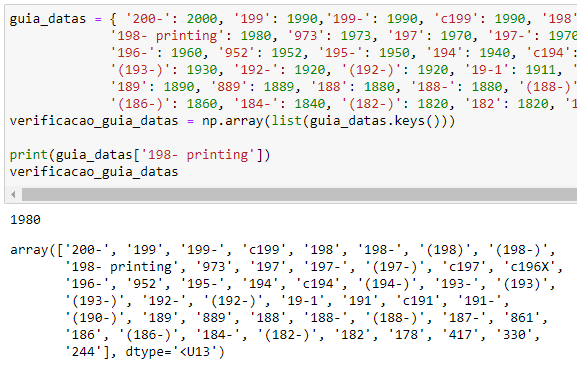
Primeiramente, separei as músicas que não possuem uma data de publicação, e como podemos ver na imagem, existem 21494 linhas sem essa informação.



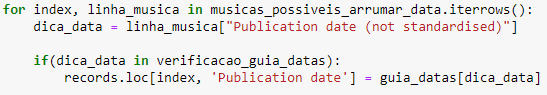
Dessas 21494 linhas, 18974 continham a data de publicação não padronizada, logo poderiam ter a informação da data de publicação adquirida dessa coluna. Logo fui ver quais eram as possíveis informações que estariam na coluna de datas não padronizadas:



Com isso descobri que haviam mais de 200 valores únicos e que boa parte desses valores não remeteram realmente a anos, mas como no meio vi alguns que poderiam ser convertidos, continuei.

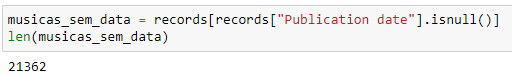


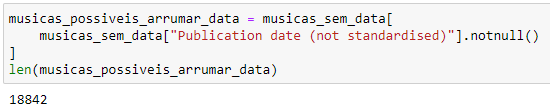
Analisando todas as possibilidades que poderiam ser anos, criei um dicionário, guia\_datas, para que quando passasse um valor das strings possíveis da coluna de data não padronizada, caso se remetesse à um ano, ele retornasse o ano correspondente àquela string. A variável vareificacao\_guia\_datas apenas servirá como validação para o processo das atualizações das datas, como demonstrado a seguir:



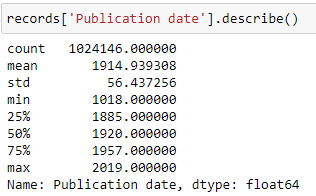
Fazendo um looping pelas linhas do DataFrame musicas\_possiveis\_arrumar\_data, eu separo a informação da data em dica\_data e vejo se aquela dica\_data é possível transformar em um ano, ou seja, verifico se ela é um valor que existe nas strings que haviam sido separadas em verificacao\_guia\_datas. Caso esteja na lista, o valor então é substituído no DataFrame original, records, sendo colocado o valor de retorno de guia\_datas, quando se passa a dica\_data para ela.

Rodando novamente as células das informações sobre as músicas com valores nulos na coluna de data de publicação podemos ver…





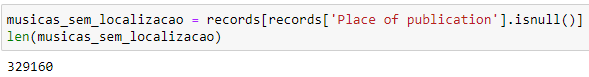
Que comparado ao resultado anterior, somente foram mudadas 132 linhas, que correspondem a apenas aproximadamente 0.01% do total das entradas, logo mudando…

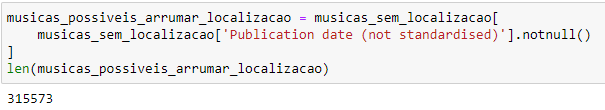


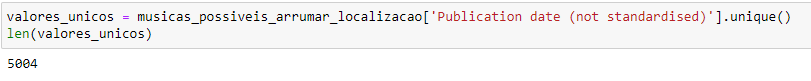
Nada do contexto geral dessa coluna. E com isso aprendi a validar minhas ações antes de executá-las, pois se tivesse visto que apenas 132 músicas das 18974 que “poderiam ser arrumadas” continham realmente uma informação de ano, não teria gastado tempo nesse processo, e se tivesse percebido que mesmo que eu mudasse 18974 linhas isso não representaria nem 2% das linhas totais do Dataframe, nem havia começado. Mas pelo menos com isso consegui uma nova informação, a coluna de data de publicação não padronizada não continha somente valores de ano, mas também de lugar. Então uma nova tarefa surgiu após a primeira etapa estipulada. Atualizando tarefas:

1. ~~Se o atributo da data de publicação de uma linha for nulo mas a data não padronizada não for, eu posso tentar transformar a informação da data não padronizada em uma data de publicação comum;~~
2. Se o valor de lugar de publicação for nulo mas houver a informação de data não padronizada, há como conseguir a informação do lugar de lá;
3. Pelo lugar de publicação, pode se conseguir quem publicou por comparar com os que mais publicam naquele lugar nas linhas em que essas informações existem;
4. E sabendo o compositor tem como saber o período de vida dele;

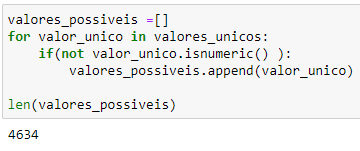
Agora com minha lição aprendida, primeiro separei as linhas que se encaixam na descrição dessa atividade e vi se o número delas faria uma diferença no Dataframe caso sejam mudadas.



  
 Com a verificação da quantidade, 315573 foi considerado um número qualificado para se trabalhar. Agora indo para a validação do impacto que as mudanças fariam no Dataframe. Peguei os valores únicos de data de publicação não padronizada dentro de musicas\_possiveis\_arrumar\_localizacao:



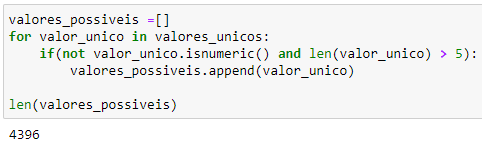
Contendo 5004 entradas diferentes resolvi filtrá-las um pouco pois possivelmente haviam informações inúteis ali, mas como queria ser cauteloso em quais iria retirar comecei por partes. Primeiramente queria retirar os valores numéricos que estavam lá, e após isso ainda sobraram 4634 valores únicos.



Verificando se isso ainda continuaria válido, é notado que ainda existem 287974 linhas válidas com as colunas já filtradas.



Anteriormente inspecionando a tabela vi que a menor string que indicava localidade registrada ali era ‘London’ então filtrei pelo tamanho da string também, resultando ainda em 4396 valores únicos.



E para validar as linhas ainda dentro desse grupo verifiquei novamente a célula anterior:



Que agora só tinha 10784 valores correspondentes, ou seja beirando o 1% das entradas totais do Data Frame, e assim inutilizando o resultado desse processo. Mesmo que essas mudanças fizessem alguma diferença caso esses dados fossem usados para alguma coisa, estou apenas realizando esses processos para refinar os dados utilizados, e assim refinar minhas habilidades de preparação de dados, e eu julguei com base no baixo impacto dessa ação que ela não irá influenciar em nada o contexto geral de minhas atividades. E assim mais um processo “concluído”.

1. ~~Se o atributo da data de publicação de uma linha for nulo mas a data não padronizada não for, eu posso tentar transformar a informação da data não padronizada em uma data de publicação comum;~~
2. ~~Se o valor de lugar de publicação for nulo mas houver a informação de data não padronizada, há como conseguir a informação do lugar de lá;~~
3. Pelo lugar de publicação, pode se conseguir quem publicou por comparar com os que mais publicam naquele lugar nas linhas em que essas informações existem;
4. E sabendo o compositor tem como saber o período de vida dele;

Começando o processo de conseguir os compositores pelo lugar de publicação resolvi ver quantas

duas coisa que não considerei:

* Se um lugar tiver mais de um compositor como sua moda
* Se o lugar de uma linha que falta um compositor for único, significando que não haveria valores do compositor comum naquele lugar pois ele só aparece naquela instância