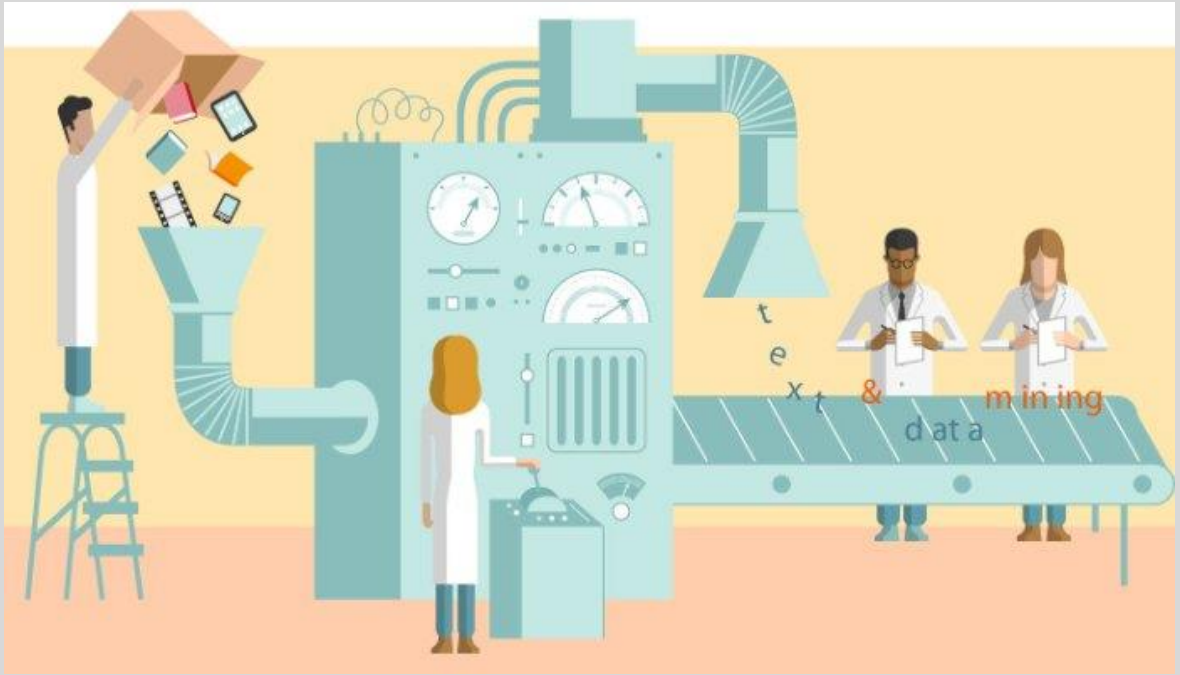


DigitalHouse > Coding School

DATA SCIENCE

Limpeza de dados com o Pandas

Julho de 2020

Limpeza de dados com o Pandas

OBJETIVOS DA AULA

1. Conhecer as operações envolvidas na limpeza de dados
2. Apresentar o problema dos valores faltantes
3. Conhecer e implementar as técnicas do “tidy data”
4. Ferramentas para a limpeza de dados: “apply()”, “value_counts()” e expressões lambda

Limpeza de dados

A limpeza de dados é uma etapa necessária em qualquer projeto de dados.

Podemos resumir o processo de limpeza de dados fazendo referência às seis tarefas a seguir:

1. **Resolver problemas de formato**: Por exemplo, ao passar de CSV para o Pandas, uma data não é importada corretamente. Ex.: 20090609231247 em vez de 2009-06-09 23:12:47

2. **Corrigir valores errados:** Por exemplo, um valor numérico ou inválido para descrever o gênero. Ou uma idade representada por um número negativo ou muito maior que 100.

3. **Padronizar categorias**: Quando os dados são coletados com um sistema que não tem os valores tipificados, valores que representam as mesmas categorias podem ser expressos de formas diferentes, por exemplo: Bra, BR e Brasil.

4. **Completar dados faltantes:** Os conjuntos de dados do mundo real geralmente vêm com dados faltantes que correspondem a informações perdidas ou nunca coletadas. Existem diversas técnicas para completar dados faltantes. O processo de completar dados faltantes se chama “imputação”.

Limpeza de dados

5. Atribuir os tipos corretos de dados: O formato em que os dados se encontram afeta a análise, por diversos motivos. Por exemplo, as operações que podem ser realizadas dependem do tipo de dados. Além disso, alguns tipos ocupam menos espaço na memória do que outros.

6. Organizar o conjunto de dados corretamente: É importante estruturar as linhas e colunas da forma mais conveniente. Para fazer isso, as regras de "tidy data" podem ser aplicadas.

Valores faltantes

De modo geral, todo conjunto de dados possui dados faltantes. Seja porque esses dados não foram coletados, seja porque nunca existiram.

- Devemos conseguir detectar, preencher e eliminar os dados faltantes
- É preciso ter conhecimento do domínio para definir quais dados faltantes serão preenchidos e como.
- O Pandas oferece maneiras de fazer isso.

Representação de dados faltantes

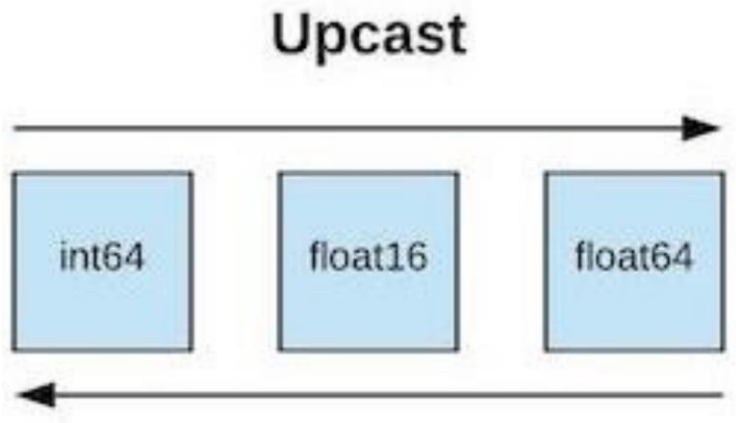
- Os conjuntos de dados do mundo real sempre têm **dados faltantes.**
- Cada linguagem/framework tem sua própria maneira de lidar com eles.
- O Pandas utiliza os valores **None, NaN ou NaT**, porque ele se baseia em Numpy:
    - **None:** objeto de Python que representa ausência de dados.
    - **NaN:** (Not A Number) definição de valor faltante de floats.
    - **NaT:** (Not A Timestamp) utilizado para valores faltantes do tipo Timestamp.

None VS np.nan

```
In [1]: import numpy as np
        import pandas as pd

In [2]: vals1 = np.array([1, None, 3, 4])
        vals1

Out[2]: array([1, None, 3, 4], dtype=object)

In [5]: vals2 = np.array([1, np.nan, 3, 4])
        vals2.dtype

Out[5]: dtype('float64')

In [3]: for dtype in ['object', 'int']:
            print("dtype =", dtype)
            %timeit np.arange(1E6, dtype=dtype).sum()
            print()

dtype = object
10 loops, best of 3: 78.2 ms per loop

dtype = int
100 loops, best of 3: 3.06 ms per loop
```

Quando temos um objeto **None** incluído em uma série, o "upcasting" de Numpy devolve como resultado "object".

Quando temos um np.nan, conservamos uma coluna de tipo float e podemos continuar fazendo operações com eficiência.



Processos de geração de dados perdidos

A abordagem escolhida frente a dados faltantes é importante, há diferentes mecanismos para descrever as possíveis relações à propensão dos dados a sua ausência e os valores desses dados, tanto ausentes, quanto observados.

- **Missing Completely at Random (MCAR)**
  - A probabilidade de o registro ter um valor perdido na variável Y não está relacionada nem com os valores de Y, nem com outros valores da matriz de dados (X).
  - Os valores perdidos são uma subamostra ao acaso dos valores totais.
  - Não há uma relação sistemática
  - Esta suposição é invalidada se:
    - Algum grupo ou subgrupo tiver maior probabilidade de apresentar dados perdidos na variável Y e/ou.
    - Se algum dos valores de Y tiver maior probabilidade de apresentar dados perdidos.

Processos de geração de dados perdidos

- **Missing at Random (MAR)**
  - Há uma relação sistemática entre a propensão dos dados faltantes e os dados observados, mas não os dados faltantes.
  - A chance de uma observação ser faltante não tem a ver com seu valor, mas tem a ver com os valores dos indivíduos observados. (Ex: Homens têm menos problema em contar seu peso do que mulheres, peso é um MAR).

- **Missing Not at Random (MNAR)**
  - A uma relação entre a propensão de uma observação ser faltante e o valor da observação.
  - Não ignorável, o mecanismo causador de dados faltantes deve ser incorporado na modelagem do problema, é necessário incluir uma motivação para os dados faltantes no modelagem final.
  - Dados que retratam realidades sociais distintas, escolaridade, doenças.

Processos de geração de dados perdidos

DigitalHouse > Coding School

MCAR

X	Y
0	366,44
0	181,67
0	NR
1	682,61
1	542,28
2	NR
2	577,22
2	219,45
3	209,86
3	NR
3	372,89
3	330,52
4	NR
4	534,75
4	691,69
4	NR
4	629,45

MAR

X	Y
0	28
0	13,69
0	181,67
1	219,45
1	209,86
2	366,44
2	372,89
2	330,52
3	NR
3	534,75
3	387,01
3	NR
4	629,45
4	757,86
4	NR
4	691,69
4	NR

MNAR

X	Y
0	13,69
0	28
0	181,67
1	209,86
1	219,45
2	330,52
2	366,44
2	372,89
3	387,01
3	416,11
3	534,75
3	542,28
4	NR
4	629,45
4	NR
4	NR
4	NR

Processos de geração de dados perdidos

DigitalHouse > Coding School

MCAR

X	Y
0	366,44
0	181,67
0	NR
1	682,61
1	542,28
2	NR
2	577,22
2	219,45
3	209,86
3	NR
3	372,89
3	330,52
4	NR
4	534,75
4	691,69
4	NR
4	629,45

MAR

X	Y
0	28
0	13,69
0	181,67
1	219,45
1	209,86
2	366,44
2	372,89
2	330,52
3	NR
3	534,75
3	387,01
3	NR
4	629,45
4	757,86
4	NR
4	691,69
4	NR

MNAR

X	Y
0	13,69
0	28
0	181,67
1	209,86
1	219,45
2	330,52
2	366,44
2	372,89
3	387,01
3	416,11
3	534,75
3	542,28
4	NR
4	629,45
4	NR
4	NR
4	NR

Processos de geração de dados perdidos

DigitalHouse > Coding School

MCAR

X	Y
0	366,44
0	181,67
0	NR
1	682,61
1	542,28
2	NR
2	577,22
2	219,45
3	209,86
3	NR
3	372,89
3	330,52
4	NR
4	534,75
4	691,69
4	NR
4	629,45

MAR

X	Y
0	28
0	13,69
0	181,67
1	219,45
1	209,86
2	366,44
2	372,89
2	330,52
3	NR
3	534,75
3	387,01
3	NR
4	629,45
4	757,86
4	NR
4	691,69
4	NR

MNAR

X	Y
0	13,69
0	28
0	181,67
1	209,86
1	219,45
2	330,52
2	366,44
2	372,89
3	387,01
3	416,11
3	534,75
3	542,28
4	NR
4	629,45
4	NR
4	NR
4	NR

Processos de geração de dados perdidos

DigitalHouse > Coding School

MCAR

X	Y
0	366,44
0	181,67
0	NR
1	682,61
1	542,28
2	NR
2	577,22
2	219,45
3	209,86
3	NR
3	372,89
3	330,52
4	NR
4	534,75
4	691,69
4	NR
4	629,45

MAR

X	Y
0	28
0	13,69
0	181,67
1	219,45
1	209,86
2	366,44
2	372,89
2	330,52
3	NR
3	534,75
3	387,01
3	NR
4	629,45
4	757,86
4	NR
4	691,69
4	NR

MNAR

X	Y
0	13,69
0	28
0	181,67
1	209,86
1	219,45
2	330,52
2	366,44
2	372,89
3	387,01
3	416,11
3	534,75
3	542,28
4	NR
4	629,45
4	NR
4	NR
4	NR

Abandono de atributos

O abandono de atributos é uma aproximação usada com o objetivo de reduzir o número de colunas de um dataset. Isso facilita a manipulação dos dados e torna o processo de limpeza mais eficiente.

- Realizar uma análise de correlação entre os atributos.
- 90% de dados faltantes é um motivo para abandono do atributo.
- Considere a natureza do atributo, avalie se é praticamente útil fazer uso de um determinado atributo. Caso o atributo não seja, realmente, útil, você pode abandoná-lo.

À prática guiada.

Métodos de imputação

Os valores faltantes podem ser preenchidos substituindo-os **pela média** da série ou **pela** **média condicionada** para determinada categoria. Por exemplo, na ausência de um valor de estatura para uma mulher, substituí-lo pela média de estatura das mulheres.

Esta abordagem tem vantagens e desvantagens:

- Vantagem: É muito provável que se aproxime do valor real do dado faltante.

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

Métodos de imputação

- Desvantagens:
    - Reduz artificialmente a variabilidade e aleatoriedade dos dados, o que pode levar a conclusões erradas.
    - Se havia correlação entre essa variável e outras, esse valor pode ser afetado.

Para resolver esses problemas, existem os métodos de **imputação múltipla** que tentam conservar as relações observadas entre as variáveis do conjunto de dados, garantindo que haja aleatoriedade nessas relações.

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

mean()

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

Métodos de imputação

O método ***fillna()*** do Pandas permite vários tipos de imputação, que podem ser especificados no parâmetro "method":

- Preencher os dados com um valor escalar ***(method = None).***
- Preencher os dados faltantes com o valor anterior ou o seguinte (ideal para séries temporais) ***(method = bfill) ou (method = ffill).***

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

df.fillna(0)

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	0.0
1	9	0.0	9.0	0	7.0
2	19	17.0	0.0	9	0.0

Métodos de imputação

O método fillna() também permite receber um dataframe onde os índices dos dados faltantes estejam associados a algum valor:

- Preencher com a média, moda ou mediana. ***df.fillna(df.mean())***

Outra possibilidade é eliminar os casos que contenham algum dado faltante (complete case deletion). Este método é ideal quando os dados faltantes são poucos e a ausência se dá totalmente ao acaso. ***df.dropna()***

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

df.fillna(0)

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	0.0
1	9	0.0	9.0	0	7.0
2	19	17.0	0.0	9	0.0

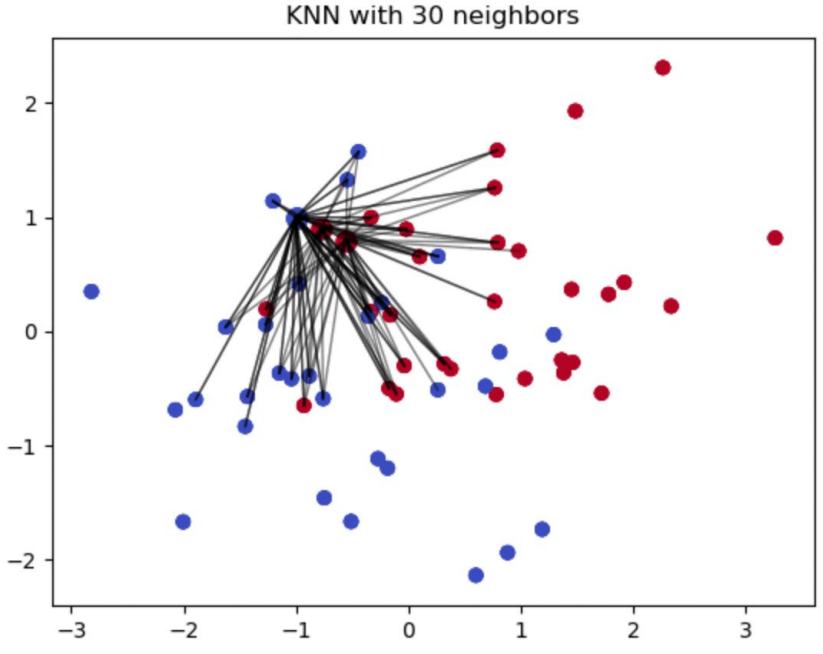
Métodos de imputação

O método de imputação por kNN (k-Nearest neighbor) usa a similaridade de atributos (feature similarity) para encontrar valores mais realistas para substituirem valores faltantes.

Usa input inicial para construir um rede de kNN, calcula a média ponderada da distância de vizinhos.

- **Pró**: Mais preciso que medias, medianas e outros métodos.
- **Con:** Computacionalmente custoso. Sensível à outliers.

```
import sys
from impyute.imputation.cs import fast_knn
sys.setrecursionlimit(100000) #Increase the recursion limit of the OS

# start the KNN training
imputed_training=fast_knn(train.values, k=30)
```



À prática guiada.

As regras do “tidy data”

O tidy data é um padrão de mapeamento do significado de um conjunto de dados de acordo com sua estrutura.

- Dizemos que um conjunto de dados está ordenado quando:
  - Cada variável é uma coluna.
  - Cada observação é uma linha.
  - Cada tipo de unidade observacional forma uma tabela.
  - Cada valor pertence a uma linha e uma coluna.
- Algumas definições:
  - Variável: É a medição de um atributo, por exemplo, peso, altura, etc.
  - Valor: É a medida que uma variável aceita para uma observação.
  - Observação: Todas as observações aceitam o mesmo tipo de valores para cada variável.

Tidy data: exemplos

Messy data

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

Tidy data

name	trt	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Wickham, H. JSS Vol 59 Issue 10 (2014).

Tidy data: exemplos

name	age	eye_color	height
Jake	34	Other	6'1"
Alice	55	Blue	5'9"
Tim	76	Brown	5'7"
Denise	19	Other	5'1"

Tidy data: exemplos

name	age	eye_color	height
Jake	34	Other	6'1"
Alice	55	Blue	5'9"
Tim	76	Brown	5'7"
Denise	19	Other	5'1"

Observação

Tidy data: exemplos

name	age	eye_color	height
Jake	34	Other	6'1"
Alice	55	Blue	5'9"
Tim	76	Brown	5'7"
Denise	19	Other	5'1"

Variável, Atributo

À prática guiada.

Ferramentas para a limpeza de dados

A função apply() no Pandas

- O método **apply()** permite aplicar qualquer função aos elementos de um Dataframe. Podem ser utilizados:
  - Por coluna: **df.apply(mi_funcion, axis = 0)**
  - Por linha **df.apply(my_function, axis = 1)**
  - Elemento por elemento **df.applymap(my_function)**
- A função **apply()** também pode ser utilizado sobre uma Serie, elemento por elemento.
- O que permite aplicar as operações vetorizadas de Numpy.

Expressões lambda

- Vamos recordar rapidamente como construir uma função usando def:
  - Esta maneira de declarar funções serve, entre outras coisas, para reduzir códigos duplicados e para modularizar o código.

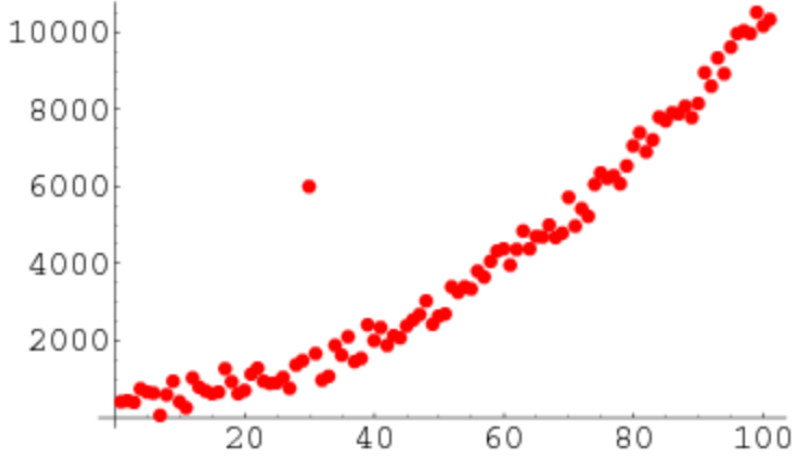
```
>>> def square(x): return x**2
>>> square(10)
100
```

- Faz sentido definir uma função dessa maneira se vamos invocar tal função apenas uma vez?
  - As funções lambda podem ser diretamente definidas no código que vai utilizá-las, sem necessidade de dar um nome para elas (são funções anônimas).
  - Aceitam uma única expressão e não contêm a expressão **return**.
  - É definida em uma única linha.

```
>>> square = lambda x: x**2
>>> square(10)
100
```

Anomalias e remoção de outliers

- Anomalias são pontos que não pertencem ao corpo da distribuição.
- Você deve ter motivos legítimos para remover um outlier.
- Deve ocorrer durante a fase explanatória.



MENÇÕES VÁLIDAS

- Preenchimento de dados faltantes com media/mediana,
- Correção de assimetria e
- Heterocedasticidade & Homocedasticidade.

MENÇÕES VÁLIDAS

- Estratégias mais avançadas de preenchimento incluem:
- **Regressão estocástica**: Que tenta prever os valor faltantes a partir de uma regressão de variáveis relacionadas.
- **Extrapolação e interpolação:** Tenta estimar o valor faltante através de outras observações uma série limitada de pontos.

À prática guiada.

Prática independente