



Tratamento de Dados Faltantes

- dados faltantes ⇒ também conhecido como **missings**

Métodos de Seleção de Dados Nulos

`isnull()`

- tal método irá gerar um DataFrame booleano, em que a observação marcada como True caracteriza um dado nulo, isto é, um missing

```
dados.isnull()
```

```
dados.isnull()
```

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	True	True
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	True
4	False	False	False	False	False	False	False	True	True
...
22575	False	False	False	False	False	False	False	False	False
22576	False	False	False	False	False	False	False	False	False
22577	False	False	False	False	False	False	False	False	False
22578	False	False	False	False	False	False	False	False	False
22579	False	False	False	False	False	False	False	False	True
22580 rows × 9 columns									

notnull()

- funciona exatamente da maneira inversa ao isnull()
- se a informação for nula, será utilizada a notação False

```
dados.notnull()
```

```
dados.notnull()
```

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	False	False
2	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	False
4	True	True	True	True	True	True	True	False	False
...
22575	True	True	True	True	True	True	True	True	True
22576	True	True	True	True	True	True	True	True	True
22577	True	True	True	True	True	True	True	True	True
22578	True	True	True	True	True	True	True	True	True
22579	True	True	True	True	True	True	True	True	False
22580 rows × 9 columns									

Verificando mais informações do dataframe

```
# mostra a quantidade de registros que o dataframe possui  
# mostra a quantidade de registros não nulos de cada variável  
# podemos então identificar quais variáveis possuem registros nulos  
dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22580 entries, 0 to 22579
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Tipo         22580 non-null   object  
 1   Bairro       22580 non-null   object  
 2   Quartos     22580 non-null   int64  
 3   Vagas        22580 non-null   int64  
 4   Suites       22580 non-null   int64  
 5   Area          22580 non-null   int64  
 6   Valor         22571 non-null   float64 
 7   Condominio   20765 non-null   float64 
 8   IPTU          15795 non-null   float64 
dtypes: float64(3), int64(4), object(2)
memory usage: 1.6+ MB
```

- no caso acima, podemos ver que além das variáveis Condomínio e IPTU, a variável Valor, também possui alguns registros nulos

Verificando apenas os valores nulos de Valor

```
# passando a seleção direto para o dataframe
dados[dados['Valor'].isnull()]
```

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
58	Apartamento	Barra da Tijuca	2	1	1	70	NaN	970.0	68.0
1492	Apartamento	Leme	2	0	0	75	NaN	878.0	NaN
1683	Casa	Campo Grande	3	4	3	363	NaN	NaN	NaN
2012	Apartamento	Botafogo	2	0	0	95	NaN	1010.0	170.0
2034	Apartamento	Copacabana	2	0	0	72	NaN	850.0	NaN
4941	Casa	Campo Grande	3	2	1	100	NaN	NaN	NaN
8568	Apartamento	Leme	2	0	1	75	NaN	878.0	NaN
8947	Apartamento	Glória	3	0	1	135	NaN	910.0	228.0
9149	Apartamento	Gávea	3	1	1	105	NaN	880.0	221.0

- a variável valor é a mais importante desse banco de dados
- portanto, não interessa manter esses registros nulos e serão removidos

Removendo registros nulos

- para melhor visualização, serão criadas variáveis para verificar o tamanho do df antes e depois da remoção

```

A = dados.shape[0]
# recebe o tamanho do df antes da remoção
dados.dropna(subset = ['Valor'], inplace=True)
# remove valores nulos de uma variável ou lista de variáveis
# é necessário o inplace para execução no df
B = dados.shape[0]
# recebe o tamnho do df depois da remoção
A - B
# mostra a diferença para saber o tanto de registros removidos

```