



Filtrando Dados

- podemos usar vários métodos para a filtragem de dados
- pegando o dataframe dados:

```
dados = pd.read_csv('.../dados/aluguel.csv', sep=';')
dados.head(10)
```

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	Quitinete	Copacabana	1	0	0	40	1700.0	500.0	60.0
1	Casa	Jardim Botânico	2	0	1	100	7000.0	NaN	NaN
2	Conjunto Comercial/Sala	Barra da Tijuca	0	4	0	150	5200.0	4020.0	1111.0
3	Apartamento	Centro	1	0	0	15	800.0	390.0	20.0
4	Apartamento	Higienópolis	1	0	0	48	800.0	230.0	NaN
5	Apartamento	Vista Alegre	3	1	0	70	1200.0	NaN	NaN
6	Apartamento	Cachambi	2	0	0	50	1300.0	301.0	17.0
7	Casa de Condomínio	Barra da Tijuca	5	4	5	750	22000.0	NaN	NaN
8	Casa de Condomínio	Ramos	2	2	0	65	1000.0	NaN	NaN
9	Conjunto Comercial/Sala	Centro	0	3	0	695	35000.0	19193.0	3030.0

Usando unique()

```
dados.Tipo.unique()

array(['Quitinete', 'Casa', 'Conjunto Comercial/Sala', 'Apartamento',
       'Casa de Condomínio', 'Prédio Inteiro', 'Flat', 'Loja/Salão',
       'Galpão/Depósito/Armazém', 'Casa Comercial', 'Casa de Vila',
       'Terreno Padrão', 'Box/Garagem', 'Loft',
       'Loja Shopping/ Ct Comercial', 'Chácara', 'Loteamento/Condomínio',
       'Sítio', 'Pousada/Chalé', 'Studio', 'Hotel', 'Indústria'],
      dtype=object)
```

Usando value_counts()

```
dados.Tipo.value_counts()
aptura Estangular
```

Apartamento	19532
Conjunto Comercial/Sala	6815
Loja/Salão	1426
Casa de Condomínio	996
Casa	967
Quitinete	836
Galpão/Depósito/Armazém	623
Flat	476
Prédio Inteiro	301
Casa Comercial	265
Casa de Vila	249
Loja Shopping/ Ct Comercial	247
Box/Garagem	82
Terreno Padrão	70
Loft	51
Sítio	10
Loteamento/Condomínio	5
Studio	4
Hotel	2
Indústria	1
Pousada/Chalé	1
Chácara	1
Name: Tipo, dtype: int64	

Usando drop_duplicates()

```
dados.Tipo.drop_duplicates()

0                 Quitinete
1                  Casa
2      Conjunto Comercial/Sala
3             Apartamento
7      Casa de Condomínio
16        Prédio Inteiro
17                  Flat
29            Loja/Salão
80  Galpão/Depósito/Armazém
83      Casa Comercial
117     Casa de Vila
159    Terreno Padrão
207      Box/Garagem
347                  Loft
589  Loja Shopping/ Ct Comercial
2157            Chácara
3354  Loteamento/Condomínio
4379                  Sítio
4721      Pousada/Chalé
6983            Studio
9687                  Hotel
23614            Indústria
Name: Tipo, dtype: object
```

Colocando drop_duplicates() em uma lista

```
list(dados.Tipo.drop_duplicates())  
['Quitinete',  
 'Casa',  
 'Conjunto Comercial/Sala',  
 'Apartamento',  
 'Casa de Condomínio',  
 'Prédio Inteiro',  
 'Flat',  
 'Loja/Salão',  
 'Galpão/Depósito/Armazém',  
 'Casa Comercial',  
 'Casa de Vila',  
 'Terreno Padrão',  
 'Box/Garagem',  
 'Loft',  
 'Loja Shopping/ Ct Comercial',  
 'Chácara',  
 'Loteamento/Condomínio',  
 'Sítio',  
 'Pousada/Chalé',  
 'Studio',  
 'Hotel',  
 'Indústria']
```

Criando uma Nova Lista

- pegamos a lista atual e descartamos os tipos que não queremos

```
residencial = ['Quitinete',  
 'Casa',  
 'Apartamento',  
 'Casa de Condomínio',  
 'Casa de Vila']  
residencial  
['Quitinete', 'Casa', 'Apartamento', 'Casa de Condomínio', 'Casa de Vila']
```

Método isin()

- compara a variável e o dataframe passados
- cria uma series booleana, identificando se o que está na variável, consta no dataframe

```
dados.Tipo.isin(residencial).head(10)
```

```
0      True
1      True
2     False
3      True
4      True
5      True
6      True
7      True
8      True
9     False
Name: Tipo, dtype: bool
```

- pode-se observar com os dois comandos acima que:
 - valores presentes na variável e no dataframe recebem True
 - valores não presentes recebem False

Criando um Novo Dataframe

- colocando o valor acima em uma variável

```
selecao = dados.Tipo.isin(residencial)
selecao
0      True
1      True
2     False
3      True
4      True
...
32955    True
32956    True
32957    True
32958    True
32959    False
Name: Tipo, Length: 32960, dtype: bool
```

- passando essa informação para um dataframe, ele vai manter apenas os registros que tiverem o valor True
- os valores com False são descartados

```
dados_residencial = dados[selecao]
dados_residencial
```

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	Quitinete	Copacabana	1	0	0	40	17000.0	500.0	60.0
1	Casa	Jardim Botânico	2	0	1	100	7000.0	NaN	NaN
3	Apartamento	Centro	1	0	0	15	800.0	390.0	20.0
4	Apartamento	Higienópolis	1	0	0	48	800.0	230.0	NaN
5	Apartamento	Vista Alegre	3	1	0	70	1200.0	NaN	NaN
...
32953	Apartamento	Méier	2	0	0	70	900.0	490.0	48.0
32955	Quitinete	Centro	0	0	0	27	800.0	350.0	25.0
32956	Apartamento	Jacarepaguá	3	1	2	78	1800.0	800.0	40.0
32957	Apartamento	São Francisco Xavier	2	1	0	48	1400.0	509.0	37.0
32958	Apartamento	Leblon	2	0	0	70	3000.0	760.0	NaN

22580 rows × 9 columns

Verificando se a Seleção foi Realizada Corretamente

```
list(dados_residencial.Tipo.drop_duplicates())
['Quitinete', 'Casa', 'Apartamento', 'Casa de Condomínio', 'Casa de Vila']
```

Comparando o Tamanho dos Dataframes

```
print(dados_residencial.shape[0])
print(dados.shape[0])
22580
32960
```

Reconstruindo o Índice

- atribuímos ao índice um range com o próprio tamanho dele

```
dados_residencial.index = range(dados_residencial.shape[0])
dados_residencial
```

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	Quitinete	Copacabana	1	0	0	40	1700.0	500.0	60.0
1	Casa	Jardim Botânico	2	0	1	100	7000.0	NaN	NaN
2	Apartamento	Centro	1	0	0	15	800.0	390.0	20.0
3	Apartamento	Higienópolis	1	0	0	48	800.0	230.0	NaN
4	Apartamento	Vista Alegre	3	1	0	70	1200.0	NaN	NaN
...
22575	Apartamento	Méier	2	0	0	70	900.0	490.0	48.0
22576	Quitinete	Centro	0	0	0	27	800.0	350.0	25.0
22577	Apartamento	Jacarepaguá	3	1	2	78	1800.0	800.0	40.0
22578	Apartamento	São Francisco Xavier	2	1	0	48	1400.0	509.0	37.0
22579	Apartamento	Leblon	2	0	0	70	3000.0	760.0	NaN

22580 rows × 9 columns