

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Aula 5
Introdução ao
Pandas - Parte II

Introdução

- Nesta aula utilizaremos Pandas para:
 - Manipular os dados:
 - Adicionar colunas
 - Adicionar linhas
 - Alterar índices
 - Unir DataFrames
 - Filtrar e Tratar os dados
 - Sumarizar os dados
 - Conteúdo dos atributos
 - Estatísticas
 - Plotagem

Adicionando Colunas

- Pode-se adicionar novas colunas informando um novo índice no acesso e atribuindo à um *array*

```
[244] df1
      Nome          CPF  Curso
000.000.000-00  Rafael 000.000.000-00   SI
000.000.123-89  Ricardo 000.000.123-89 Crochê
000.000.456-12 Ronaldo 000.000.456-12   SI
```

```
[245] df1['Idade'] = np.random.randint(100, size=3)
```

```
[246] df1
      Nome          CPF  Curso  Idade
000.000.000-00  Rafael 000.000.000-00   SI    96
000.000.123-89  Ricardo 000.000.123-89 Crochê    24
000.000.456-12 Ronaldo 000.000.456-12   SI    47
```

```
[247] df1['Cidade'] = ['Porto Ferreira', 'Jales', 'Rinópolis']
```

```
df1
      Nome          CPF  Curso  Idade     Cidade
000.000.000-00  Rafael 000.000.000-00   SI    96  Porto Ferreira
000.000.123-89  Ricardo 000.000.123-89 Crochê    24      Jales
000.000.456-12 Ronaldo 000.000.456-12   SI    47      Rinópolis
```

Adicionando Colunas

- Pode-se adicionar novas colunas utilizando a função apply

```
[248] df1
```

	Nome	CPF	Curso	Idade	Cidade
000.000.000-00	Rafael	000.000.000-00	SI	96	Porto Ferreira
000.000.123-89	Ricardo	000.000.123-89	Crochê	24	Jales
000.000.456-12	Ronaldo	000.000.456-12	SI	47	Rinópolis

```
[251] df1['Ano_Nascimento'] = df1['Idade'].apply(lambda x : 2020 - x)
```

```
df1
```

	Nome	CPF	Curso	Idade	Cidade	Ano_Nascimento
000.000.000-00	Rafael	000.000.000-00	SI	96	Porto Ferreira	1924
000.000.123-89	Ricardo	000.000.123-89	Crochê	24	Jales	1996
000.000.456-12	Ronaldo	000.000.456-12	SI	47	Rinópolis	1973

Adicionando Colunas

- **OBSERVAÇÃO:** podemos aplicar funções para formatar as séries geradas antes de adicioná-las como colunas, como a `round()` para padronizar o número de casas decimais

	Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	[Rodas de liga, 'Traços elétricos', 'Pára...]	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	[Central multimídia, 'Teto panorâmico', 'Vid...]	106181.94
2	Crossfox	Motor Diesel V8	1990	37130.0	False	[Vidros autênticos, Controle de estabilidade...]	72832.16
3	D55	Motor 2.4 Turbo	2019	NaN	True	['Traços elétricos', '4 X 4', 'Vidros elétricos...]	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	[Rodas de liga, '4 X 4', 'Central multimídia,...]	92612.10
...
253	Phantom 2013	Motor V8	2014	27950.0	False	['Controle de estabilidade', 'Piloto automático...]	51759.58
254	Cadillac Ciel concept	Motor V8	1991	29980.0	False	['Bancos de couro', 'Painel digital', 'Sensor...]	51687.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	[Rodas de liga, Controle de traçador, 'Câmbio...]	69934.03
256	Aston Martin DBS	Motor Diesel V8	1996	7695.0	False	['Ar condicionado', '4 X 4', 'Câmbio automático...]	122110.90
257	Macan	Motor Diesel V8	1992	50188.0	False	[Central multimídia, 'Teto panorâmico', 'Vid...]	90381.47

256 rows × 8 columns

```
[319]: df3['Km_Média'] = df3[['Ano','Quilometragem']].apply(lambda x : x['Quilometragem'] / x['Ano'], axis=1).round(2)
```

	Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor	Km_Média
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	[Rodas de liga, 'Traços elétricos', 'Pára...]	88078.64	22.17
1	Passat	Motor Diesel	1991	5712.0	False	[Central multimídia, 'Teto panorâmico', 'Vid...]	106181.94	2.87
2	Crossfox	Motor Diesel V8	1990	37130.0	False	[Vidros autênticos, Controle de estabilidade...]	72832.16	18.65
3	D55	Motor 2.4 Turbo	2019	NaN	True	['Traços elétricos', '4 X 4', 'Vidros elétricos...]	124549.07	NaN
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	[Rodas de liga, '4 X 4', 'Central multimídia,...]	92612.10	12.84
...
253	Phantom 2013	Motor V8	2014	27950.0	False	['Controle de estabilidade', 'Piloto automático...]	51759.58	13.66
254	Cadillac Ciel concept	Motor V8	1991	29980.0	False	['Bancos de couro', 'Painel digital', 'Sensor...]	51687.06	15.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	[Rodas de liga, Controle de traçador, 'Câmbio...]	69934.03	26.29
256	Aston Martin DBS	Motor Diesel V8	1996	7695.0	False	['Ar condicionado', '4 X 4', 'Câmbio automático...]	122110.90	3.85
257	Macan	Motor Diesel V8	1992	50188.0	False	[Central multimídia, 'Teto panorâmico', 'Vid...]	90381.47	25.19

256 rows × 9 columns

Adicionando Colunas

- Pode-se adicionar novas colunas concatenando DataFrames por meio do método concat

```
[320] df11
```

	Nome	CPF	Curso	Idade	Cidade	Ano_Nascimento
0	Rafael	000.000.000-00	SI	78	Porto Ferreira	1942
1	Ricardo	000.000.123-89	Crochê	76	Jales	1944
2	Ronaldo	000.000.456-12	SI	75	Rinópolis	1945

```
[321] df10
```

	Nota
0	10
1	4
2	6

```
pd.concat([df11,df10],axis=1)
```

	Nome	CPF	Curso	Idade	Cidade	Ano_Nascimento	Nota
0	Rafael	000.000.000-00	SI	78	Porto Ferreira	1942	10
1	Ricardo	000.000.123-89	Crochê	76	Jales	1944	4
2	Ronaldo	000.000.456-12	SI	75	Rinópolis	1945	6



Adicionando Colunas

- Pode-se criar colunas também fazendo a união de informações contidas em diferentes DataFrames utilizando a função `merge`
- A diferença é que o `merge` faz a união de acordo com as chaves fornecidas, isto é, a informação de dois DataFrames serão unidas de acordo com os valores das chaves
- Existem diferentes opções para se fazer a união de maneira semelhante aos dos Bancos de Dados: `inner`, `left`, `right` e `outer`
- A união pode ser feita com base nos índices ou com base em colunas informadas
- Veja: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html>

Adicionando Colunas

```
[154] df1
```

		Nome	Ano_Nasc	CPF	Curso
	345.666.888-99	Rafael	1985	345.666.888-99	SI
	258.222.989-90	Ricardo	1980	258.222.989-90	ADM
	555.654.987-98	Ronaldo	1983	555.654.987-98	CC

```
[158] df_cursos = pd.DataFrame({'Sigla': ['ADM','CC','SI'],
                               'Nome_Curso': ['Administração','Ciência da Computação','Sistemas de Informação']})
```

```
[159] df_cursos
```

	Sigla	Nome_Curso
0	ADM	Administração
1	CC	Ciência da Computação
2	SI	Sistemas de Informação

```
pd.merge(df1, df_cursos, left_on='Curso', right_on='Sigla')
```

	Nome	Ano_Nasc	CPF	Curso	Sigla	Nome_Curso
0	Rafael	1985	345.666.888-99	SI	SI	Sistemas de Informação
1	Ricardo	1980	258.222.989-90	ADM	ADM	Administração
2	Ronaldo	1983	555.654.987-98	CC	CC	Ciência da Computação

Adicionando Linhas

- Para adicionar uma nova linha em um DataFrame, pode-se utilizar a função `append()`
- O primeiro parâmetro para a função `append` é um dicionário cujas chaves devem corresponder às colunas
- Caso não queira especificar um índice para um novo elemento, basta utilizar o parâmetro `ignore_index = True`
- OBSERVAÇÃO:** pode-se também utilizar uma lista de dicionários como parâmetro da função

Adicionando Linhas

```
[151] df3
```

	Nome	CPF	Curso
0	Rafael	000.000.000-11	SI
1	Ricardo	010.030.040.80	Crochê
2	Ronaldo	858.987.98-10	SI

```
[152] df3 = df3.append({'Nome': 'Zé', 'CPF': '012.321.878-96', 'Curso': 'Balé'}, ignore_index=True)
```

	Nome	CPF	Curso
0	Rafael	000.000.000-11	SI
1	Ricardo	010.030.040.80	Crochê
2	Ronaldo	858.987.98-10	SI
3	Zé	012.321.878-96	Balé

```
[153] df3.append([{'Nome': 'Pelé', 'CPF': '111.222.333-44', 'Curso': 'Administração'},  
{'Nome': 'Garrincha', 'CPF': '444.555.666-77', 'Curso': 'Contabilidade'}], ignore_index=True)
```

	Nome	CPF	Curso
0	Rafael	000.000.000-11	SI
1	Ricardo	010.030.040.80	Crochê
2	Ronaldo	858.987.98-10	SI
3	Zé	012.321.878-96	Balé
4	Pelé	111.222.333-44	Administração
5	Garrincha	444.555.666-77	Contabilidade



Adicionando Linhas

- Também pode-se utilizar Series ou uma lista de Series como entrada para a função append
- Porém, é necessário fazer com que os índices das séries sejam iguais às colunas do DataFrame



Adicionando Linhas

```
[152] df3
      Nome          CPF Curso
0  Rafael  000.000.000-11     SI
1  Ricardo  010.030.040.80 Crochê
2  Ronaldo  858.987.98-10     SI
3      Zé  012.321.878-96   Balé

[156] pd3 = df3.append(pd.Series(['Mané', '456.987.123-99', 'Ed. Física'], index=df3.columns), ignore_index=True)
pd3
      Nome          CPF Curso
0  Rafael  000.000.000-11     SI
1  Ricardo  010.030.040.80 Crochê
2  Ronaldo  858.987.98-10     SI
3      Zé  012.321.878-96   Balé
4  Mané  456.987.123-99 Ed. Física

❶ pd3 = df3.append([pd.Series(['Mané', '456.987.123-99', 'Letras'], index=df3.columns),
                  pd.Series(['Coutinho', '987.654.321-00', 'Direito'], index=df3.columns),
                  ], ignore_index=True)
pd3
      Nome          CPF Curso
0  Rafael  000.000.000-11     SI
1  Ricardo  010.030.040.80 Crochê
2  Ronaldo  858.987.98-10     SI
3      Zé  012.321.878-96   Balé
4  Mané  456.987.123-99 Letras
5  Coutinho  987.654.321-00 Direito
```



Adicionando Linhas

- Também é possível utilizar o método concat para concatenar dois DataFrames desde que as colunas sejam iguais

[171] df1

	Nome	CPF	Curso
000.000.000-00	Rafael	000.000.000-00	SI
000.000.123-89	Ricardo	000.000.123-89	Crochê
000.000.456-12	Ronaldo	000.000.456-12	SI

[173] df15

	Nome	CPF	Curso
000.000.000-00	Abelardo	147.258.369-96	Geografia
000.000.123-89	Alberto	369.258.147-77	História
000.000.456-12	Adalberto	258.147.369-22	Enfermagem

df16 = df1.append(df15)

df16

	Nome	CPF	Curso
000.000.000-00	Rafael	000.000.000-00	SI
000.000.123-89	Ricardo	000.000.123-89	Crochê
000.000.456-12	Ronaldo	000.000.456-12	SI
000.000.000-00	Abelardo	147.258.369-96	Geografia
000.000.123-89	Alberto	369.258.147-77	História
000.000.456-12	Adalberto	258.147.369-22	Enfermagem

Substituindo Linhas

- Para substituir o conteúdo de uma linha, basta acessá-la por meio dos métodos `loc` ou `iloc` e atribuir um novo valor a linha
- Esse novo valor pode ser fornecido por meio de uma lista ou uma `Series`

Substituindo Linhas

[160] pd3

	Nome	CPF	Curso
0	Rafael	000.000.000-11	SI
1	Ricardo	010.030.040.80	Crochê
2	Ronaldo	858.987.98-10	SI
3	Zé	012.321.878-96	Balé
4	Mané	456.987.123-99	Letras
5	Coutinho	987.654.321-00	Direito

▶ #Substituindo os dados de uma linha utilizando a função iloc e uma lista
pd3.iloc[5] = ['Messi', '456.789.321-55', 'Medicina']
pd3

	Nome	CPF	Curso
0	Rafael	000.000.000-11	SI
1	Ricardo	010.030.040.80	Crochê
2	Ronaldo	858.987.98-10	SI
3	Zé	012.321.878-96	Balé
4	Mané	456.987.123-99	Letras
5	Messi	456.789.321-55	Medicina

Substituindo Linhas

[167] pd3

C	Nome	CPF	Curso
0	Rafael	000.000.000-11	SI
1	Ricardo	010.030.040.80	Crochê
2	Ronaldo	858.987.98-10	SI
3	Salá	777.888.999-11	Contabilidade
4	Mané	456.987.123-99	Letras
5	Messi	456.789.321-55	Medicina

▶ #Substituindo os dados de uma linha utilizando a função iloc e uma lista
pd3.iloc[3] = pd.Series(['Salá', '777.888.999-11', 'Contabilidade'], index=pd3.columns)
pd3

C	Nome	CPF	Curso
0	Rafael	000.000.000-11	SI
1	Ricardo	010.030.040.80	Crochê
2	Ronaldo	858.987.98-10	SI
3	Salá	777.888.999-11	Contabilidade
4	Mané	456.987.123-99	Letras
5	Messi	456.789.321-55	Medicina

Apagando Linhas e Colunas

- Para apagar linhas ou colunas, pode-se fazer uso da função `drop()`
- Um dos usos da função `drop` permite informar diretamente se a intenção é apagar colunas fornecendo uma lista de nomes de colunas no parâmetro `columns`, ou apagar uma lista de linhas fornecendo uma lista de índices para o parâmetro `index`
- Pode-se também simplesmente fornecer uma lista de índices e especificar a qual eixo os índices pertencem por meio do parâmetro `axis`
- OBSERVAÇÃO:** para fazer com que o comando `drop` altere o próprio `DataFrame`, dever para uso do parâmetro `inplace=True`



Apagando Linhas e Colunas

```
[186] df16 = df1.append(df15)
      df16
```

C		Nome	CPF	Curso
	000.000.000-00	Rafael	000.000.000-00	SI
	000.000.123-89	Ricardo	000.000.123-89	Crochê
	000.000.456-12	Ronaldo	000.000.456-12	SI
	000.000.000-00	Abelardo	147.258.369-96	Geografia
	000.000.123-89	Alberto	369.258.147-77	História
	000.000.456-12	Adalberto	258.147.369-22	Enfermagem

```
[187] df16.drop(columns=['Curso'], inplace=True)
      df16
```

C		Nome	CPF
	000.000.000-00	Rafael	000.000.000-00
	000.000.123-89	Ricardo	000.000.123-89
	000.000.456-12	Ronaldo	000.000.456-12
	000.000.000-00	Abelardo	147.258.369-96
	000.000.123-89	Alberto	369.258.147-77
	000.000.456-12	Adalberto	258.147.369-22

Apagando Linhas e Colunas

df16

		Nome	CPF
	000.000.000-00	Rafael	000.000.000-00
	000.000.123-89	Ricardo	000.000.123-89
	000.000.456-12	Ronaldo	000.000.456-12
	000.000.000-00	Abelardo	147.258.369-96
	000.000.123-89	Alberto	369.258.147-77
	000.000.456-12	Adalberto	258.147.369-22

df16.drop(index='000.000.123-89', inplace=True)
df16

		Nome	CPF
	000.000.000-00	Rafael	000.000.000-00
	000.000.456-12	Ronaldo	000.000.456-12
	000.000.000-00	Abelardo	147.258.369-96
	000.000.456-12	Adalberto	258.147.369-22



Apagando Linhas e Colunas

```
[195] df16 = df1.append(df15)
df16
```

	None	CPF	Curso
000.000.000-00	Rafael	000.000.000-00	SI
000.000.123-89	Ricardo	000.000.123-89	Crochê
000.000.456-12	Ronaldo	000.000.456-12	SI
000.000.000-00	Abelardo	147.258.369-96	Geografia
000.000.123-89	Alberto	369.258.147-77	História
000.000.456-12	Adalberto	258.147.369-22	Enfermagem

```
[196] df16.drop(['CPF', 'Curso'], axis=1, inplace=True)
df16
```

	None
000.000.000-00	Rafael
000.000.123-89	Ricardo
000.000.456-12	Ronaldo
000.000.000-00	Abelardo
000.000.123-89	Alberto
000.000.456-12	Adalberto

```
⑥ df16.drop(['000.000.000-00'], axis=0, inplace=True)
df16
```

	None
000.000.123-89	Ricardo
000.000.456-12	Ronaldo
000.000.123-89	Alberto
000.000.456-12	Adalberto



Apagando Linhas e Colunas

- Também podemos utilizar as consultas para “apagar” dados utilizando algum critério de seleção com base nos valores
- Para isso, armazenar o resultado da consulta no próprio DataFrame

[134] df4.head(n=10)							
	Nome	Motor	Ano	Quilometragem	Zero_km	Accessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44400.0	False	[Rodas de liga, Travas elétricas, Piloto...]	89708.64
1	Passat	Motor Diesel	1991	5712.0	False	[Central multimídia, Teto panorâmico, Pil...]	106161.9
2	CrossFox	Motor Diesel V8	1990	37223.0	False	[Piloto automático, Controle de estabilidade...]	72832.16
3	DSS	Motor 2.4 Turbo	2019	Nan	True	[Travas elétricas, 4 X 4, Vidros elétricos...]	126569.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	[Rodas de liga, 4 X 4, Central multimídia,...]	93612.10
5	Polo Weekend	Motor 1.8 16v	2012	10728.0	False	[Sensor de estacionamento, Teto panorâmico...]	97497.73
6	AS	Motor 4.0 Turbo	2019	Nan	True	[Controle automático, Câmera de estacionament...]	56445.20
7	Série 3 Cabrio	Motor 1.0 16v	2009	77599.0	False	[Controle de estabilidade, Sensor crepuscula...]	112310.44
8	Dodge Journey	Motor 3.0 32v	2010	99370.0	False	[Vidros elétricos, Piloto automático, Teto...]	120716.27
9	Carrera	Motor 5.0 V8 Bi-Turbo	2011	37878.0	False	[Ar condicionado, Painel digital, Central...]	70566.49

[148] df4 = df4[df4['Nome'] == 'Passat']							
	Nome	Motor	Ano	Quilometragem	Zero_km	Accessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	[Rodas de liga, Travas elétricas, Piloto...]	89708.64
2	CrossFox	Motor Diesel V8	1990	37223.0	False	[Piloto automático, Controle de estabilidade...]	72832.16
3	DSS	Motor 2.4 Turbo	2019	Nan	True	[Travas elétricas, 4 X 4, Vidros elétricos...]	126569.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	[Rodas de liga, 4 X 4, Central multimídia,...]	93612.10
5	Polo Weekend	Motor 1.8 16v	2012	10728.0	False	[Sensor de estacionamento, Teto panorâmico...]	97497.73
6	AS	Motor 4.0 Turbo	2019	Nan	True	[Câmera automática, Câmera de estacionament...]	56445.20
7	Série 3 Cabrio	Motor 1.0 16v	2009	77599.0	False	[Controle de estabilidade, Sensor crepuscula...]	112310.44
8	Dodge Journey	Motor 3.0 32v	2010	99397.0	False	[Vidros elétricos, Piloto automático, Teto...]	120716.27
9	Carrera	Motor 5.0 V8 Bi-Turbo	2011	37878.0	False	[Ar condicionado, Painel digital, Central...]	70566.49
10	Ford Edge	Motor Diesel V6	2002	12899.0	False	[Sensor crepuscular, Rodas de liga, Teto...]	71647.99

Apagando Linhas e Colunas

- Pode-se também utilizar o operador `del` para apagar colunas

```
[324] df1
```

	Nome	Ano_Nasc	CPF	Curso
345.666.888-99	Rafael	1985	345.666.888-99	SI
258.222.989-90	Ricardo	1980	258.222.989-90	ADM
555.654.987-98	Ronaldo	1983	555.654.987-98	CC

```
[325] del df1['Curso']
```

```
df1
```

	Nome	Ano_Nasc	CPF
345.666.888-99	Rafael	1985	345.666.888-99
258.222.989-90	Ricardo	1980	258.222.989-90
555.654.987-98	Ronaldo	1983	555.654.987-98

Alterando Índices

- Muitas vezes é desejado alterar os valores dos índices e colunas por alguma facilidade de acesso ou programação
- Para isso, pode-se manipular diretamente os atributos dos índices ou utilizar métodos específicos para tal

Alterando Índices

- A função `reset_index` pode ser utilizada para atribuir índices de variam de 0 a (número de linhas - 1) às linhas de um DataFrame

```
[146] df4
```

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
3	DS5	Motor 2.4 Turbo	2019		NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico... 124549.07
8	Dodge Journey	Motor 3.0 32v	2010		99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet... 120716.27
9	Carens	Motor 5.0 V8 Bi-Turbo	2011		37978.0	False	['Ar condicionado', 'Painel digital', 'Central... 76566.49

```
df4 = df4.reset_index()
df4
```

	index	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	3	DS5	Motor 2.4 Turbo	2019		NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico... 124549.07
1	8	Dodge Journey	Motor 3.0 32v	2010		99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet... 120716.27
2	9	Carens	Motor 5.0 V8 Bi-Turbo	2011		37978.0	False	['Ar condicionado', 'Painel digital', 'Central... 76566.49

Alterando Índices

- Pode-se alterar diretamente os índices das linhas passando diretamente um lista ou array para o atributo index do DataFrame

```
[156] df4.index = [4, 6, 8]
```

df4

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
4	DS5	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
6	Dodge Journey	Motor 3.0 32v	2010	99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet...']	120716.27
8	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0	False	['Ar condicionado', 'Painel digital', 'Central...']	76566.49

Alterando Índices

- Pode-se utilizar a função `rename` para se alterar os índices de linhas e colunas
- Para alterar os índices das linhas, deve-se utilizar o atributo `index`
- Para alterar os índices das colunas, deve-se utilizar o atributos `columns`
- Para ambos, deve-se passar um dicionário cuja chave é o valor do índice atual, e o valor corresponde ao novo índice



Alterando Índices

Alterando o índice das linhas

```
[157] df4
```

C	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
4	DS5	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
6	Dodge Journey	Motor 3.0 32v	2010	99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet...']	120716.27
8	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0	False	['Ar condicionado', 'Painel digital', 'Central...']	76566.49

```
df4 = df4.rename(index={6:16, 8:18})  
df4
```

C	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
4	DS5	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
16	Dodge Journey	Motor 3.0 32v	2010	99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet...']	120716.27
18	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0	False	['Ar condicionado', 'Painel digital', 'Central...']	76566.49

Alterando Índices

Alterando o índice das colunas

[161] df4

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
4	DS5	Motor 2.4 Turbo	2019		NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico... 124549.07
6	Dodge Journey	Motor 3.0 32v	2010		99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet... 120716.27
8	Carens	Motor 5.0 V8 Bi-Turbo	2011		37978.0	False	['Ar condicionado', 'Painel digital', 'Central... 76566.49

```
df4 = df4.rename(columns={'Nome': 'Modelo', 'Quilometragem': 'KM', 'Valor': 'Preço'})  
df4
```

	Modelo	Motor	Ano	KM	Zero_km	Acessórios	Preço
4	DS5	Motor 2.4 Turbo	2019	NaN		True	['Travas elétricas', '4 X 4', 'Vidros elétrico... 124549.07
6	Dodge Journey	Motor 3.0 32v	2010	99197.0		False	['Vidros elétricos', 'Piloto automático', 'Tet... 120716.27
8	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0		False	['Ar condicionado', 'Painel digital', 'Central... 76566.49

Ordenação

- Pode-se ordenar os elementos de um DataFrame considerando os valores dos índices
- Para isso, pode-se utilizar a função `sort_index`, cujos principais atributos são:
 - `axis`: 0 para índices das linhas (padrão) e 1 para os índices das colunas
 - `ascending`: True para ordenação ascendente (padrão) e False para descendente
 - `inplace`: True para ordenar e manter o DataFrame ordenado e False para retornar um novo DataFrame ordenado (padrão)
 - `kind`: define o algoritmo de ordenação – 'quicksort' (padrão), 'mergesort', e 'heapsort'
 - `na_position`: define a ordem onde os valores com NaN aparecerão – 'first' para o começo, e 'last' para o final (padrão)



Ordenação

[259] df1

	Nome	Ano_Nasc	CPF	Curso
345.666.888-99	Rafael	1985	345.666.888-99	SI
258.222.989-90	Ricardo	1980	258.222.989-90	ADM
555.654.987-98	Ronaldo	1983	555.654.987-98	CC

[260] # Ordenando considerando os índices das linhas
df1.sort_index()

	Nome	Ano_Nasc	CPF	Curso
258.222.989-90	Ricardo	1980	258.222.989-90	ADM
345.666.888-99	Rafael	1985	345.666.888-99	SI
555.654.987-98	Ronaldo	1983	555.654.987-98	CC

▶ # Ordenando de maneira descendente considerando os índices das colunas
df1.sort_index(axis=1, ascending=False)

	Nome	Curso	CPF	Ano_Nasc
345.666.888-99	Rafael	SI	345.666.888-99	1985
258.222.989-90	Ricardo	ADM	258.222.989-90	1980
555.654.987-98	Ronaldo	CC	555.654.987-98	1983



Ordenação

- A função `sort_values` é utilizada para ordenar linhas com base em valores das colunas ou mesmo das próprias linhas
- Essa função possui basicamente os mesmos atributos do método `sort_index`
- Porém, aqui é necessário informar quais são os critérios de ordenação, por exemplo, uma coluna ou lista de colunas cujos valores serão considerado para a ordenação
- Esses critério são informados no parâmetro `by` (primeiro parâmetro da função)

Ordenação

```
[168] df4.head(n=10)
```

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilida...']	72832.16
3	DS5	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
5	Palio Weekend	Motor 1.8 16v	2012	10728.0	False	['Sensor de estacionamento', 'Teto panorâmico...']	97497.73
6	A5	Motor 4.0 Turbo	2019	NaN	True	['Câmbio automático', 'Câmera de estacionament...']	56445.20
7	Série 3 Cabrio	Motor 1.0 8v	2009	77599.0	False	['Controle de estabilidade', 'Sensor crepuscul...']	112310.44
8	Dodge Journey	Motor 3.0 32v	2010	99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet...']	120716.27
9	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0	False	['Ar condicionado', 'Painel digital', 'Central...']	76566.49

```
[169] df4.sort_values(by=['Ano','Valor'],ascending=False).head(n=10)
```

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
117	Lamborghini Ferruccio	Motor 3.0 32v	2019	NaN	True	['Vidros elétricos', 'Piloto automático', 'Tel...']	149157.17
142	Voyage	Motor 5.0 V8 Bi-Turbo	2019	NaN	True	['Ar condicionado', 'Sensor de estacionamento...']	148996.47
132	Aston Martin Cygnet	Motor 5.0 V8 Bi-Turbo	2019	NaN	True	['Painel digital', 'Ar condicionado', 'Sensor ...']	147770.87
158	308 CC	Motor 4.0 Turbo	2019	NaN	True	['Câmera de estacionamento', 'Bancos de couro...']	146405.79
78	Cielo Hatch	Motor Diesel	2019	NaN	True	['Painel digital', 'Central multimídia', 'Câme...']	145197.70
68	Lancer Evolution X	Motor Diesel V6	2019	NaN	True	['Vidros elétricos', 'Câmbio automático', 'Con...']	142392.75
124	Camry	Motor Diesel	2019	NaN	True	['Travas elétricas', 'Rodas de liga', 'Sensor ...']	138597.27
201	Classe B	Motor Diesel V8	2019	NaN	True	['4 X 4', 'Bancos de couro', 'Sensor crepuscul...']	135543.00
57	CLA	Motor 1.0 8v	2019	NaN	True	['Central multimídia', 'Sensor de chuva', 'Con...']	133329.19
58	Accord	Motor V8	2019	NaN	True	['Sensor de estacionamento', 'Ar condicionado...']	131961.43

Tratando os Dados

- Muitas vezes podemos ter valores NaN nas células de DataFrames e Series
- Esse dados correspondem à valores ausentes, isto é, não foram observados ou cadastrados na base
- Porém, fazer processamentos utilizando esses valores podem levar a resultados indesejados
- Por exemplo, se fizermos uma operação matemática utilizando o valor NaN, o resultado também será NaN



Tratando os Dados

- O Pandas provê funções para lidar com valores NaN de direntes formas:
 - Apagando registros com valores NaN
 - Substituindo valores NaN por um valor padrão
 - Substituindo valores NaN por valores inferidos

Tratando os Dados

- Podemos eliminar todas as instâncias com valores NaN invocando a função `dropna()` no DataFrame correspondente
- Caso queira apagar instâncias que contenham valores NaN em apenas algumas colunas, é possível informar essas colunas no parâmetros `subset`

Tratando os Dados

```
[286] df3.sort_values(by='Quilometragem',na_position='first')
```

```
# Retornando um DataFrame sem exemplos contendo o valor NaN na Quilometragem
df3.dropna(subset=['Quilometragem'])
```

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	[Rodas de liga], 'Travas elétricas', 'Páto...	88079.64
1	Passat	Motor Diesel V8	1991	5712.0	False	[Central multimídia], 'Teto panorâmico', 'Fre...	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	[Pistão automático], 'Controle de estabilida...	72832.16
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	[Rodas de liga], '4 X 4', 'Central multimídia',...	92612.10
5	Pálio Weekend	Motor 1.8 16v	2012	10728.0	False	[Sensor de estacionamento], 'Teto panorâmico'...	97497.73
...
253	Phantom 2013	Motor V8	2014	27505.0	False	[Controle de estabilidade], 'Pistão automático...',...	51759.56
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro'], 'Painel digital', 'Sensor...	51667.00
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	[Rodas de liga], 'Controle de estabilidade', 'Câmbi...	69384.00
256	Aston Martin DB5	Motor Diesel	1966	7695.0	False	['Ar condicionado'], '4 X 4', 'Câmbio automático',...	12110.90
257	Macan	Motor Diesel V6	1992	50188.0	False	[Central multimídia], 'Teto panorâmico', 'Vid...	90381.47

Tratando os Dados

		Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
3		D5S	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétricos...', 'Câmbio automático', 'Câmera de estacionamento...', 'Sensor de estacionamento', 'Sensor crepuscular', 'Painel digital', 'Rodas de liga...', 'Controle de estabilidade', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	124549.07
6		A5	Motor 4.0 Turbo	2019	NaN	True	['Câmbio automático', 'Câmera de estacionamento...', 'Sensor de estacionamento', 'Sensor crepuscular', 'Painel digital', 'Rodas de liga...', 'Controle de estabilidade', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	56445.20
13		25	Motor V8	2019	NaN	True	['Sensor crepuscular', 'Painel digital', 'Rodas de liga...', 'Controle de estabilidade', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	53183.38
17		A3	Motor 1.0 Bv	2019	NaN	True	['4 X 4', 'Piloto automático', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	88952.39
21		Série 1 M	Motor V8	2019	NaN	True	['Controle de estabilidade', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	94564.40
...	
75		CS	Motor 2.0 16v	1994	118236.0	False	['Painel digital', 'Teto panorâmico', 'Piloto automático', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	60818.92
62	New Fiesta Hatch		Motor Diesel	2017	118895.0	False	['Sensor de estacionamento', 'Travas elétricas...', 'Sensor crepuscular', 'Piloto automático', 'Teto panorâmico', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	66007.16
56	Aston Martin DB7 Vantage		Motor V8	2007	119513.0	False	['Sensor crepuscular', 'Piloto automático', 'Teto panorâmico', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	128984.23
91	207 SW		Motor 3.0 32v	2017	119760.0	False	['Controle de estabilidade', 'Travas elétricas...', 'Sensor de chuva', 'Piloto automático', 'Teto panorâmico', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	117709.67
121	Discovery 4		Motor 4.0 Turbo	2005	119945.0	False	['Freios ABS', 'Rodas de liga', 'Sensor de chuva', 'Piloto automático', 'Teto panorâmico', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	115784.95

258 rows × 7 columns

		Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant		Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto automático', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	88078.64
1	Passat		Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Freios ABS', 'Sensor de estacionamento', 'Painel digital', 'Rodas de liga...', 'Controle de estabilidade', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	106161.94
2	Crossfox		Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidade', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	72832.18
4	Aston Martin DB4		Motor 2.4 4 Turbo	2009	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	92612.10
5	Pão Weekend		Motor 1.8 16v	2012	10728.0	False	['Sensor de estacionamento', 'Teto panorâmico', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	97497.73
...
253	Phantom 2013		Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automático', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	51759.58
254	Cadillac Ciel concept		Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor de estacionamento', 'Piloto automático', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	51667.06
255	Classe GLK		Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio automático', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	68934.03
256	Aston Martin DB5		Motor Diesel	1996	7685.0	False	['Air condicionado', '4 X 4', 'Câmbio automático', 'Central multimídia...', 'Air condicionado', 'Teto panorâmico', 'Vidros t...']	122110.90
257	Macan		Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vidros t...']	90381.47

197 rows × 7 columns



Tratando os Dados

- Podemos substituir os valores NaN de todas as instâncias utilizando a função `fillna([valor_padrão])`
- Caso queira atribuir um valor padrão para cada coluna, basta passar como argumento um dicionário, cuja chave é o identificador da coluna e o valor corresponde ao valor que será utilizado no lugar dos NaNs da coluna



Tratando os Dados

[293] df3

	Nome	Motor	Ano	Kilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidade...']	72832.16
3	DSS	Motor 2.4 Turbo	2019	Nan	True	['Travas elétricas', '4 X 4', 'Vidros elétricos...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...', 'Câmbio automáti...']	92612.10
...
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automático...', 'Câmbio automáti...']	51759.58
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...']	51667.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio ...']	68934.03
256	Aston Martin DB5	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automáti...']	122110.90
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...']	90381.47

258 rows × 7 columns

❸ #Preenchendo todos os NaN da coluna Kilometragem com o valor padrão 0
df3.fillna({'Kilometragem' : 0})

	Nome	Motor	Ano	Kilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...']	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...']	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidade...']	72832.16
3	DSS	Motor 2.4 Turbo	2019	0.0	True	['Travas elétricas', '4 X 4', 'Vidros elétricos...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...', 'Câmbio automáti...']	92612.10
...
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automático...', 'Câmbio automáti...']	51759.58
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...']	51667.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio ...']	68934.03
256	Aston Martin DB5	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automáti...']	122110.90
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...']	90381.47

258 rows × 7 columns



Tratando os Dados

[293] df3

	None	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidade...']	72832.16
3	DSS	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétricos...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
...
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automático...']	51759.58
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...']	51667.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio...']	68934.03
256	Aston Martin DB5	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automático...']	122110.90
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...']	90381.47

258 rows × 7 columns

#Preenchendo todos os NaN com um valor padrão
df3.fillna(0)

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...']	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...']	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidade...']	72832.16
3	DSS	Motor 2.4 Turbo	2019	0.0	True	['Travas elétricas', '4 X 4', 'Vidros elétricos...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
...
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automático...']	51759.58
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...']	51667.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio...']	68934.03
256	Aston Martin DB5	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automático...']	122110.90
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...']	90381.47

258 rows × 7 columns

Tratando os Dados

- O Pandas também provê funcionalidades para eliminar instâncias duplicadas na base de dados
- Para isso, deve-se utilizar a função `drop_duplicates()` → irá eliminar instâncias cujos valores de todos os atributos sejam iguais
- Caso queira eliminar instâncias cujos valores de atributos em apenas algumas colunas sejam iguais, pode-se informar tais colunas em uma lista no parâmetro `subset`

Tratando os Dados

[304] df2

	Nome	Ano_Nasc	CPF	Curso	Idade
0	Rafael	1985	000.111.222-33	SI	35
1	Ricardo	1980	222.333.444-55	ADM	40
2	Ronaldo	1983	333.444.555-66	CC	37
3	Rafael	1985	000.111.222-33	SI	35
4	Rafael R.	1985	000.111.222-33	SI	35

[1] # Eliminando registros em que todos os campos são iguais
df2.drop_duplicates()

	Nome	Ano_Nasc	CPF	Curso	Idade
0	Rafael	1985	000.111.222-33	SI	35
1	Ricardo	1980	222.333.444-55	ADM	40
2	Ronaldo	1983	333.444.555-66	CC	37
4	Rafael R.	1985	000.111.222-33	SI	35

Tratando os Dados

[306] df2

	Nome	Ano_Nasc	CPF	Curso	Idade
0	Rafael	1985	000.111.222-33	SI	35
1	Ricardo	1980	222.333.444-55	ADM	40
2	Ronaldo	1983	333.444.555-66	CC	37
3	Rafael	1985	000.111.222-33	SI	35
4	Rafael R.	1985	000.111.222-33	SI	35

[307] # Eliminando registros em que todos os CPFs são iguais
df2.drop_duplicates(subset=['CPF'])

	Nome	Ano_Nasc	CPF	Curso	Idade
0	Rafael	1985	000.111.222-33	SI	35
1	Ricardo	1980	222.333.444-55	ADM	40
2	Ronaldo	1983	333.444.555-66	CC	37

Tratando os Dados

- Há métodos mais sofisticados para inputar valores em campos com o NaN, por exemplo, utilizando algoritmos estatísticos / aprendizado de máquina
- O Pandas provê o método `interpolate` para fazer a inferência de valores NaN
- Pode-se definir diferentes métodos de interpolação por meio do parâmetro `method`
- Pode ser aplicado tanto à `Series` quanto à `DataFrames`

Tratando os Dados

```
[412] df15 = pd.DataFrame({"Nome":["Rafael", "Ricardo", "Alexandre", "Ronaldo"],  
                           "Profissão": ["Professor", "Professor", "Diretor", "Professor"],  
                           "Idade": [34, 36, 50, 35],  
                           "Salário": [3500, None, 8000, 3600]})
```

```
[413] df15
```

	Nome	Profissão	Idade	Salário
0	Rafael	Professor	34	3500.0
1	Ricardo	Professor	36	NaN
2	Alexandre	Diretor	50	8000.0
3	Ronaldo	Professor	35	3600.0

```
[414] df15.interpolate(method='nearest')
```

	Nome	Profissão	Idade	Salário
0	Rafael	Professor	34	3500.0
1	Ricardo	Professor	36	3500.0
2	Alexandre	Diretor	50	8000.0
3	Ronaldo	Professor	35	3600.0

```
[415] df15.interpolate(method='linear')
```

	Nome	Profissão	Idade	Salário
0	Rafael	Professor	34	3500.0
1	Ricardo	Professor	36	5750.0
2	Alexandre	Diretor	50	8000.0
3	Ronaldo	Professor	35	3600.0

Tratando os Dados

Podemos utilizar o replace para fazer substituições e tratamento de valores, incluindo os NaNs



Tratando os Dados

```
[99] df6.sort_values(by=['Quilometragem'],na_position='first').head(n=10)
```

C.	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
3	DS5	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...',	124549.07
6	A5	Motor 4.0 Turbo	2019	NaN	True	['Câmbio automático', 'Câmera de estacionament...',	56445.20
13	J5	Motor V6	2019	NaN	True	['Sensor crepuscular', 'Painel digital', 'Roda...',	53183.38
17	A3	Motor 1.0 8v	2019	NaN	True	['4 X 4', 'Piloto automático', 'Central multim...',	88552.39
21	Série 1 M	Motor V8	2019	NaN	True	['Controle de estabilidade', 'Central multimidi...',	94564.40
24	Lamborghini Murciélagos	Motor 5.0 V8 Bi-Turbo	2019	NaN	True	['Freios ABS', 'Câmbio automático', 'Ar condic...',	121596.21
25	Up!	Motor Diesel V6	2019	NaN	True	['Sensor de estacionamento', 'Vidros elétricos...',	71367.10
27	Sandero Stepway	Motor 3.0 32v	2019	NaN	True	['Freios ABS', 'Ar condicionado', 'Teto panorâ...',	105432.84
36	Pajero TR4	Motor 2.4 Turbo	2019	NaN	True	['Controle de tração', 'Bancos de couro', 'Câm...',	51606.59
37	Polo Sedan	Motor V6	2019	NaN	True	['Sensor de chuva', 'Sensor crepuscular', 'Ar ...	123281.87

```
❸ df7 = df6.copy()
df7.replace({None : 0}, inplace=True)
df7.sort_values(by=['Quilometragem'],na_position='first').head(n=10)
```

C.	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
130	Classe CL	Motor V8	2019	0.0	True	['Central multimídia', 'Câmbio automático', '4...',	58525.41
116	Z4 Roadster	Motor 3.0 32v	2019	0.0	True	['Central multimídia', 'Câmbio automático', 'T...',	80382.24
37	Polo Sedan	Motor V6	2019	0.0	True	['Sensor de chuva', 'Sensor crepuscular', 'Ar ...	123281.87
39	S60	Motor 2.4 Turbo	2019	0.0	True	['Teto panorâmico', 'Freios ABS', '4 X 4', 'Ar...',	102609.54
132	Aston Martin Cygnet	Motor 5.0 V8 Bi-Turbo	2019	0.0	True	['Painel digital', 'Ar condicionado', 'Sensor ...	147770.87
42	Optima	Motor 1.8 16v	2019	0.0	True	['Central multimídia', 'Ar condicionado', 'Tet...',	86641.34
84	Hilux	Motor 5.0 V8 Bi-Turbo	2019	0.0	True	['4 X 4', 'Piloto automático', 'Teto panorâmico...',	64507.72
44	Range Rover Vogue	Motor 1.0 8v	2019	0.0	True	['4 X 4', 'Vidros elétricos', 'Rodas de liga...',	117626.73
114	Q3	Motor 3.0 32v	2019	0.0	True	['Controle de estabilidade', 'Bancos de couro',...	114727.68
189	Aston Martin Virage	Motor Diesel	2019	0.0	True	['Travas elétricas', 'Controle de tração', 'Câ...',	97290.18

Tratando os Dados

[104] df7.head(n=10)

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidad...']	72832.16
3	DS5	Motor 2.4 Turbo	2019	0.0	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
5	Palio Weekend	Motor 1.8 16v	2012	10728.0	False	['Sensor de estacionamento', 'Teto panorâmico...']	97497.73
6	A5	Motor 4.0 Turbo	2019	0.0	True	['Câmbio automático', 'Câmera de estacionament...']	56445.20
7	Série 3 Cabrio	Motor 1.0 8v	2009	77599.0	False	['Controle de estabilidade', 'Sensor crepuscul...']	112310.44
8	Dodge Journey	Motor 3.0 32v	2010	99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tel...']	120716.27
9	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0	False	['Ar condicionado', 'Painel digital', 'Central...']	76566.49

df7.replace(['Jetta Variant','Passat','Crossfox'],'WV').head(n=10)

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	WV	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...']	88078.64
1	WV	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...']	106161.94
2	WV	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidad...']	72832.16
3	DS5	Motor 2.4 Turbo	2019	0.0	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
5	Palio Weekend	Motor 1.8 16v	2012	10728.0	False	['Sensor de estacionamento', 'Teto panorâmico...']	97497.73
6	A5	Motor 4.0 Turbo	2019	0.0	True	['Câmbio automático', 'Câmera de estacionament...']	56445.20
7	Série 3 Cabrio	Motor 1.0 8v	2009	77599.0	False	['Controle de estabilidade', 'Sensor crepuscul...']	112310.44
8	Dodge Journey	Motor 3.0 32v	2010	99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tel...']	120716.27
9	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0	False	['Ar condicionado', 'Painel digital', 'Central...']	76566.49

Tratando os Dados

[108] df7.head(n=10)

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidad...']	72832.16
3	DSS	Motor 2.4 Turbo	2019	0.0	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
5	Pallo Weekend	Motor 1.8 16v	2012	10728.0	False	['Sensor de estacionamento', 'Teto panorâmico...']	97497.73
6	A5	Motor 4.0 Turbo	2019	0.0	True	['Câmbio automático', 'Câmera de estacionament...']	56445.20
7	Série 3 Cabrio	Motor 1.0 8v	2009	77599.0	False	['Controle de estabilidade', 'Sensor crepuscul...']	112310.44
8	Dodge Journey	Motor 3.0 32v	2010	99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet...']	120716.27
9	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0	False	['Ar condicionado', 'Painel digital', 'Central...']	76566.49

df7.replace(regex='Palio.*', value='Palio').head(n=10)

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...']	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...']	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidad...']	72832.16
3	DSS	Motor 2.4 Turbo	2019	0.0	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
5	Palio	Motor 1.8 16v	2012	10728.0	False	['Sensor de estacionamento', 'Teto panorâmico...']	97497.73
6	A5	Motor 4.0 Turbo	2019	0.0	True	['Câmbio automático', 'Câmera de estacionament...']	56445.20
7	Série 3 Cabrio	Motor 1.0 8v	2009	77599.0	False	['Controle de estabilidade', 'Sensor crepuscul...']	112310.44
8	Dodge Journey	Motor 3.0 32v	2010	99197.0	False	['Vidros elétricos', 'Piloto automático', 'Tet...']	120716.27
9	Carens	Motor 5.0 V8 Bi-Turbo	2011	37978.0	False	['Ar condicionado', 'Painel digital', 'Central...']	76566.49



Filtrando Dados

- Além dos métodos para recuperar tuplas de um DataFrame quando atingidos um critério, como por meio da função `query`, é possível utilizar outros métodos para selecionar / filtrar dados no Pandas
- Um deles é o método `isin`, o qual recebe uma lista de valores e retornará uma lista de *booleans*, na qual os valores `True` correspondem aos elementos pertencentes à lista e `False` aos que não pertencem

Introdução
 Manipulando Linhas e Colunas
Filtrando e Tratando os Dados
 Extraindo Informações dos Dados
 Plotanto Dados
 Material Complementar

Tratando os Dados
Filtrando os Dados

Filtrando Dados

	Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44610.0	False	[Rodas de liga, Travas elétricas, Piloto...]	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	[Central multimídia, Teto panorâmico, Fre...]	101618.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	[Piloto automático, Controle de estabilida...]	72832.16
3	D55	Motor 2.4 Turbo	2019	Nan	True	[Travas elétricas, 4 X 4, Vidro elétrico...]	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	[Rodas de liga, 4 X 4, Central multimídia,...]	92612.10
..
253	Phantom 2013	Motor V8	2014	27950.0	False	[Controle de estabilidade, Piloto automáti...]	51791.59
254	Cadillac Ciel concept	Motor V8	1991	29901.0	False	[Bancos de couro, Painel digital, Sensor...]	51987.06
255	Classe GLK, Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	[Rodas de liga, Controle de tração, Câmbio...]	69934.03	
256	Aston Martin DBS	Motor Diesel	1999	7695.0	False	[Ar condicionado, 4 X 4, Câmbio automático,...]	122110.00
257	Macan	Motor Diesel V8	1992	50188.0	False	[Central multimídia, Ralo parcenélico, Vol...]	90381.47

258 rows × 7 columns

```
[429] selecao = df3["Motor"].isin(["Motor Diesel V8", "Motor V8", "Motor 1.0 4x4"])
```

```
selecao
0    False
1    False
2     True
3    False
4    False
..
253   True
254   True
255   False
256   False
257   False
Name: Motor, Length: 258, dtype: bool
```

```
df3[selecao]
```

	Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
2	Crossfox	Motor Diesel V8	1990	37123.0	False	[Piloto automático, Controle de estabilidade,...]	72832.16
7	Série C Cobrio	Motor 1.0 4v	2009	77990.0	False	[Controle de estabilidade, Sensor de pressão...]	112350.44
11	Cougar	Motor Diesel V8	2007	8052.0	False	[Central multimídia, 4 X 4, Volante multifunç...]	73915.53
15	RAM	Motor Diesel V8	2018	119907.0	False	[Sensor de pressão, Câmbio automático, ...]	59910.40
17	A3	Motor 1.0 4v	2019	Nan	True	[4 X 4, Piloto automático, Central multimídia,...]	85552.39
..
253	Lamborghini Sesto Elemento	Motor V8	2007	85384.0	False	[Sensor de pressão, Freios ABS, Ar condic...]	55085.99
254	Lamborghini Huracán	Motor V8	1994	98108.0	False	[Sensor de chuva, Sensor de estacionamento,...]	118826.44
255	A7	Motor 1.0 4v	2007	71280.0	False	[Sensor de chuva, Vidro elétrico, Sensor...]	137827.62
256	Phantom 2013	Motor V8	2014	27950.0	False	[Controle de estabilidade, Piloto automáti...]	51791.59
257	Cadillac Ciel concept	Motor V8	1991	29901.0	False	[Bancos de couro, Painel digital, Sensor...]	51987.06

72 rows × 7 columns

Filtrando Dados

- Pode-se utilizar a função `filter` para filtrar linhas ou colunas utilizando diferentes critérios
 - `items`: lista contendo os índices que devem ser filtrado
 - `like`: *string* contendo parte do conteúdo que os índices devem ter
 - `regex`: *string* contendo uma expressão regular que será utilizada como filtro
- Deve-se especificar o eixo dos índices utilizando o parâmetro `axis`



Filtrando Dados

```
[111] df1
```

	Nome	Ano_Nasc	CPF	Curso
345.666.888-99	Rafael	1985	345.666.888-99	SI
258.222.989-90	Ricardo	1980	258.222.989-90	ADM
555.654.987-98	Ronaldo	1983	555.654.987-98	CC

```
[112] # Filtrando utilizando uma lista  
df1.filter(items=['Nome','CPF'])
```

	Nome	CPF
345.666.888-99	Rafael	345.666.888-99
258.222.989-90	Ricardo	258.222.989-90
555.654.987-98	Ronaldo	555.654.987-98

```
[114] # Filtrando utilizando o parâmetro like (equivalente ao contains)  
df1.filter(like='on')
```

	Nome
345.666.888-99	Rafael
258.222.989-90	Ricardo
555.654.987-98	Ronaldo

```
[1] # Filtrando utilizando uma expressão regular  
df1.filter(regex="C[A-za-z]+")
```

	CPF	Curso
345.666.888-99	345.666.888-99	SI
258.222.989-90	258.222.989-90	ADM
555.654.987-98	555.654.987-98	CC



Extraindo Informações dos Dados

- Podemos extrair diversas informações das Series e DataFrames:
 - Valores únicos
 - Contagem de Valores
 - Estatísticas dos atributos
 - Estatísticas de grupos de instâncias

Valores Únicos

- Para retornar os valores únicos de uma coluna, deve-se fazer uso da função `unique()`

[345] df3

C	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidad...']	72832.16
3	DS5	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
...
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automátic...']	51759.58
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...']	51667.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio...']	68934.03
256	Aston Martin DBS	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automátic...']	122110.90
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...']	90381.47

258 rows × 7 columns

▶ # Obtendo os diferentes valores para Motor no DataFrame
`df3['Motor'].unique()`

▶ array(['Motor 4.0 Turbo', 'Motor Diesel', 'Motor Diesel V8',
 'Motor 2.4 Turbo', 'Motor 1.8 16v', 'Motor 1.8 8v',
 'Motor 3.0 32v', 'Motor 5.0 V8 Bi-Turbo', 'Motor Diesel V6',
 'Motor V6', 'Motor V8', 'Motor 2.0 16v'], dtype=object)



Contagem de Valores

- Para contar a frequência de cada valor em uma coluna, deve-se fazer uso da função `value_counts()`

[345] df3

	Nome	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Motor automático', 'Controle de estabilidade...']	72832.16
3	DS5	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétricos...']	124549.07
4	Aston Martin DB4	Motor 2.4 Bi-Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
...
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automático...']	51759.58
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...']	51667.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio...']	68934.03
256	Aston Martin DBS	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automático...']	122110.90
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...']	90381.47

258 rows × 7 columns

➊ # Contando a quantidades de cada tipo de motor
df3[‘Motor’].value_counts()

➋ Motor 3.0 32v 27
Motor V8 26
Motor Diesel 26
Motor 1.8 V4 25
Motor 4.0 Turbo 25
Motor Diesel V8 21
Motor 5.0 V8 BI-Turbo 21
Motor Diesel V6 21
Motor V6 20
Motor 2.0 16v 18
Motor 1.8 16v 18
Motor 2.4 Turbo 10
Name: Motor, dtype: int64

Estatísticas

- Pode-se extrair estatísticas dos valores de uma coluna ou conjunto de colunas, utilizando, por exemplo, as funções, `min()`, `max()` e `mean()`

[349] df3

	Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor A.D Turbo	2003	44410.0	False	[Rodas de liga, "Travas elétricas", "Pôsito...]	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	[Central multimídia, "Telé panorâmico", Fre...]	106161.94
2	CrossFox	Motor Diesel V8	1990	37123.0	False	[Pôsto automático, "Controle de estabilidade...]	72832.16
3	D50	Motor 2.4 Turbo	2019	Nan	True	[Travas elétricas, 4 X 4, "Vidros elétricos...]	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	[Rodas de liga, 4 X 4, Central multimídia...]	92612.10
...
253	Phantom 2013	Motor V8	2014	27905.0	False	[Controle de estabilidade, "Pôsto automátic...]	51739.58
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	[Bancos de couro, "Painel digital", "Sensor ...]	51867.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	[Rodas de liga, "Controle de tração", "Câmbio...]	68934.03
256	Aston Martin DB5	Motor Diesel	1960	7685.0	False	[Ar condicionado, 4 X 4, "Câmbio automátic...]	122110.90
257	Macan	Motor Diesel V8	1992	50188.0	False	[Central multimídia, "Telé panorâmico", "Vid...]	90381.47

258 rows × 7 columns

```
[350] # Obtendo o carro mais antigo
df3['Ano'].min()

D: 1990

[352] # Obtendo o carro mais novo
df3['Ano'].max()

D: 2019

[353] # Extraindo a média das idades dos veículos
df3['Ano'].mean()

D: 2007.5116279069769

④ # Extraindo a média das idades dos veículos e dos valores dos veículos
df3[['Ano', 'Valor']].mean()

D: Ano    2007.511628
      Valor   98990.513101
      dtype: float64
```

Estatísticas

- A função `describe()` extrai várias estatísticas dos atributos numéricos de um DataFrame

[355]: df3

		None	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64	
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94	
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidad...]	72832.16	
3	D55	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétricos...]	124549.07	
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...]	92612.10	
...	
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automátic...]	51759.58	
254	Cadillac Ciel concept	Motor V8	1991	29881.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...]	51667.06	
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio...]	68934.03	
256	Aston Martin DB5	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automátic...]	122110.90	
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...]	90381.47	

258 rows × 7 columns

df3.describe()

	Ano	Quilometragem	Valor
count	258.000000	197.000000	258.000000
mean	2007.511628	58278.421320	98960.513101
std	9.725906	35836.733259	29811.932305
min	1990.000000	107.000000	50742.100000
25%	1999.000000	27503.000000	70743.512500
50%	2008.000000	55083.000000	97724.380000
75%	2018.000000	90495.000000	124633.302500
max	2019.000000	119945.000000	149489.920000

Estatísticas

- A função `corr()` extrai a correlação entre todos os atributos de um DataFrame

[369] df3

		Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64	
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94	
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidade...']	72832.16	
3	D55	Motor 2.4 Turbo	2019	NaN	True	['Travas elétricas', '4 X 4', 'Vidros elétricos...']	124549.07	
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10	
...	
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automáti...']	51759.58	
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...']	51667.06	
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbio...']	68934.03	
256	Aston Martin DB5	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automátic...']	122110.90	
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...']	90381.47	

258 rows x 7 columns

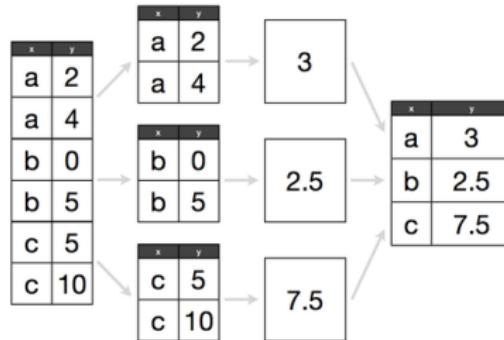
df3.fillna(0).corr()

	Ano	Quilometragem	Zero_km	Valor
Ano	1.000000	-0.455654	0.658572	0.062673
Quilometragem	-0.455654	1.000000	-0.621228	0.058250
Zero_km	0.658572	-0.621228	1.000000	0.027366
Valor	0.062673	0.058250	0.027366	1.000000

Groupby

- Também é possível extraír extratísticas por grupos de valores, isto é, dado um atributo (ou coluna), para cada valor desta coluna, selecionam-se as instâncias que contêm este valor e extraem-se as medidas estatísticas destas instâncias

Exemplo da extração da média por grupos considerando a coluna x



https://miro.medium.com/max/865/1*f5niM71W-uTRJkSSGhW7QQ.png

Groupby

- Para facilitar a extração das estatísticas por grupos, pode-se utilizar a função groupby
- A função groupby recebe um coluna ou uma lista de coluna as quais serão consideradas para fazer o agrupamento
- Os grupos obtidos podem ser acessados pelo atributo groups
 - Um dicionário cuja chave corresponde aos valores das colunas ou à uma tupla contendo a combinação dos valores das colunas
 - Valores dos dicionário correspondem aos índices das instancias em que os valor(es) da(s) chave(s) ocorreu(rão)



Groupby

Objeto retornado pela função groupby é um DataFrameGroupBy

```
[134] df3
```

	Name	Motor	Ano	Quilometragem	Zero_km	Acessórios	Valor
0	Jetta Variant	Motor 4.0 Turbo	2003	44410.0	False	['Rodas de liga', 'Travas elétricas', 'Piloto ...	88078.64
1	Passat	Motor Diesel	1991	5712.0	False	['Central multimídia', 'Teto panorâmico', 'Fre...]	106161.94
2	Crossfox	Motor Diesel V8	1990	37123.0	False	['Piloto automático', 'Controle de estabilidad...']	72832.16
3	DS5	Motor 2.4 Turbo	2019	0.0	True	['Travas elétricas', '4 X 4', 'Vidros elétrico...']	124549.07
4	Aston Martin DB4	Motor 2.4 Turbo	2006	25757.0	False	['Rodas de liga', '4 X 4', 'Central multimídia...']	92612.10
...
253	Phantom 2013	Motor V8	2014	27505.0	False	['Controle de estabilidade', 'Piloto automátic...']	51759.58
254	Cadillac Ciel concept	Motor V8	1991	29981.0	False	['Bancos de couro', 'Painel digital', 'Sensor ...']	51667.06
255	Classe GLK	Motor 5.0 V8 Bi-Turbo	2002	52637.0	False	['Rodas de liga', 'Controle de tração', 'Câmbi...']	68934.03
256	Aston Martin DBS	Motor Diesel	1996	7685.0	False	['Ar condicionado', '4 X 4', 'Câmbio automátic...']	122110.90
257	Macan	Motor Diesel V6	1992	50188.0	False	['Central multimídia', 'Teto panorâmico', 'Vid...']	90381.47

258 rows × 7 columns

```
[135] grupos = df3.groupby(by='Motor')
```

```
type(grupos)
```

```
pandas.core.groupby.generic.DataFrameGroupBy
```

Groupby

Exemplo do dicionário armazenado no atributo groups

```
grupos.groups
C: {'Motor 1.0 8v': Int64Index([ 7, 17, 18, 20, 30, 32, 38, 44, 49, 51, 57, 67, 71,
      95, 144, 162, 170, 176, 197, 199, 206, 214, 228, 229, 252],
      dtype='int64'),
     'Motor 1.8 16v': Int64Index([ 5, 42, 72, 88, 94, 105, 109, 113, 118, 134, 140, 143, 147,
      174, 190, 212, 223, 230],
      dtype='int64'),
     'Motor 2.0 16v': Int64Index([ 31, 33, 40, 59, 64, 75, 79, 90, 102, 138, 165, 171, 178,
      196, 225, 236, 239, 244],
      dtype='int64'),
     'Motor 2.4 Turbo': Int64Index([3, 4, 36, 39, 53, 98, 135, 139, 216, 222], dtype='int64'),
     'Motor 3.0 32v': Int64Index([ 8, 16, 19, 27, 28, 46, 50, 61, 73, 91, 93, 100, 103,
      108, 114, 116, 117, 137, 156, 163, 164, 168, 177, 185, 186, 220,
      245],
      dtype='int64'),
     'Motor 4.0 Turbo': Int64Index([ 0, 6, 26, 29, 43, 47, 87, 96, 112, 121, 126, 133, 152,
      158, 166, 167, 175, 179, 188, 203, 213, 215, 218, 238, 248],
      dtype='int64'),
     'Motor 5.0 V8 BI-Turbo': Int64Index([ 9, 14, 24, 45, 54, 55, 63, 65, 80, 84, 106, 129, 132,
      142, 151, 153, 160, 181, 195, 209, 255],
      dtype='int64'),
     'Motor Diesel': Int64Index([ 1, 23, 60, 62, 69, 74, 78, 86, 89, 111, 119, 124, 145,
      148, 149, 157, 173, 189, 208, 210, 211, 226, 233, 237, 240, 256],
      dtype='int64'),
     'Motor Diesel V6': Int64Index([ 10, 22, 25, 68, 70, 82, 97, 123, 125, 141, 146, 150, 169,
      180, 183, 184, 205, 219, 234, 243, 257],
      dtype='int64'),
     'Motor Diesel V8': Int64Index([ 2, 11, 15, 48, 85, 110, 120, 122, 127, 128, 159, 172, 182,
      187, 191, 193, 201, 202, 227, 231, 249],
      dtype='int64'),
     'Motor V6': Int64Index([ 12, 13, 37, 41, 56, 81, 83, 92, 101, 104, 131, 136, 155,
      200, 217, 224, 235, 241, 246, 247],
      dtype='int64'),
     'Motor V8': Int64Index([ 21, 34, 35, 52, 58, 66, 76, 77, 99, 107, 115, 130, 154,
      161, 192, 194, 198, 204, 207, 221, 232, 242, 250, 251, 253, 254],
      dtype='int64'))}
```

Groupby

Extraindo a média dos grupos

grupos.mean()

	Ano	Quilometragem	Zero_km	Valor
Motor				
Motor 1.0 8v	2004.920000	47155.960000	0.120000	97832.979200
Motor 1.8 16v	2010.388889	30994.611111	0.222222	83872.455556
Motor 2.0 16v	2005.000000	55719.944444	0.166667	94041.291111
Motor 2.4 Turbo	2010.700000	52708.500000	0.400000	99104.985000
Motor 3.0 32v	2011.629630	37720.851852	0.370370	93772.808148
Motor 4.0 Turbo	2008.080000	52085.360000	0.240000	102303.932000
Motor 5.0 V8 Bi-Turbo	2006.857143	31951.476190	0.190476	93346.201905
Motor Diesel	2007.384615	47597.961538	0.192308	107179.946154
Motor Diesel V6	2001.428571	48309.000000	0.142857	104054.461429
Motor Diesel V8	2008.952381	44182.571429	0.285714	103580.674762
Motor V6	2007.050000	46701.350000	0.200000	107471.725000
Motor V8	2008.461538	42635.692308	0.346154	97934.818846

Groupby

Extraindo a média dos grupos considerando colunas específicas

▶ `grupos[['Ano','Quilometragem','Valor']].mean()`

	Ano	Quilometragem	Valor
Nome			
207 SW	2017	119760.0	117709.67
207 Sedan	2010	83559.0	89511.96
308 CC	2019	0.0	146405.79
500 Abarth	1994	56944.0	90345.05
A1	1994	81007.0	70846.03
...
XKR	1996	85127.0	54263.16
Xsara Picasso	2013	74867.0	142603.84
Z4 Roadster	2019	0.0	80382.24
i30	2019	0.0	86224.10
i30 CW	2015	8497.0	73311.75

258 rows × 3 columns



Groupby

Utilizando o `describe` para extrair várias estatísticas dos grupos

D		Ano												Quilometragem												Valor											
		count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max				
		Motor												Avaliação												Preço											
	Motor 1.0 8v	25.0	2004.920000	9.806877	1991.0	1997.00	2003.0	2014.00	2019.0	25.0	47155.960000	36372.562298	0.0	14046.00	41496.00	77599.00	115930.0	25.0	97832.979200	30298.841182	52805.91	70424.1100	97951.030	123911.7800	146203.34												
	Motor 1.6 16v	18.0	2010.398889	8.776231	1995.0	2005.25	2012.5	2018.00	2019.0	18.0	30994.611111	36008.117697	0.0	5902.25	14716.5	48558.75	116002.0	18.0	63672.455594	2348.235978	53894.61	64179.3625	95838.190	99111.6950	128991.16												
	Motor 2.0 16v	18.0	2005.000000	10.652500	1990.0	1995.25	2000.5	2013.75	2019.0	18.0	59719.944444	44962.971412	0.0	12299.00	55854.5	99037.75	116298.0	18.0	94041.291111	11658.837881	50742.10	68707.5675	11812.390	119520.6325	142923.02												
	Motor 2.4 Turbo	10.0	2010.700000	9.930260	1995.0	2000.00	2013.5	2019.00	2019.0	10.0	57708.500000	50669.026932	0.0	0.00	58878.0	99255.50	110966.0	10.0	99104.960000	23325.476369	51008.59	80470.0575	101080.995	118566.9320	124549.07												
	Motor 3.0 32v	27.0	2011.629650	7.959504	1993.0	2008.00	2013.0	2019.00	2019.0	27.0	37720.851852	36522.507672	0.0	0.00	67731.0	67696.00	119760.0	27.0	95772.808148	23932.239753	53620.31	66625.2950	90065.380	118870.3550	149157.17												
	Motor 4.0 Turbo	25.0	2008.080000	9.141207	1992.0	2003.00	2008.0	2016.00	2019.0	25.0	53035.300000	43714.162579	0.0	2395.00	45071.0	85519.00	110485.0	25.0	10203.030300	23458.616653	56485.43	88078.6400	109517.7150	12812.5700	160325.79												
	Motor V6 8V Turbo	21.0	2008.871548	9.834209	1992.0	1999.00	2008.0	2015.00	2019.0	21.0	31951.471150	28664.059429	0.0	8848.00	32017.0	47503.00	95649.0	21.0	85348.201905	34348.505591	56208.70	66433.8200	73038.530	127486.4200	148999.47												
	Motor Diesel	26.0	2007.384615	10.203077	1990.0	1998.50	2011.0	2015.75	2019.0	26.0	47977.961158	42866.107259	0.0	7886.00	32647.0	90181.75	118895.0	26.0	101719.946154	30813.501942	54263.18	78777.6100	111740.470	137205.6025	146718.91												
	Motor Diesel V6	21.0	2001.428571	10.220243	1990.0	1993.00	1998.0	2008.00	2019.0	21.0	48309.000000	35264.150417	0.0	23860.00	49467.0	74866.00	116482.0	21.0	104054.461423	39982.288787	52027.45	77897.0650	113090.325	125396.1100	147927.81												
	Motor Diesel V8	21.0	2008.951281	9.457870	1990.0	2003.00	2008.0	2019.00	2019.0	21.0	44182.574249	40077.509950	0.0	0.00	52185.0	73516.00	115667.0	21.0	105980.674761	32250.478000	56648.34	73019.5300	102988.700	135543.0000	148493.92												
	Motor V6	20.0	2007.050000	9.327238	1993.0	2000.00	2007.0	2016.50	2019.0	20.0	46701.350000	36680.400938	0.0	6845.50	45764.5	72682.50	116163.0	20.0	107471.725000	29111.613498	53181.38	88256.7900	112438.895	130329.4950	141982.93												
	Motor V8	26.0	2008.461538	10.161904	1991.0	2000.25	2009.0	2019.00	2019.0	26.0	42835.692306	43731.551590	0.0	0.00	28745.0	87621.50	118908.0	26.0	97934.818948	32497.832809	51697.08	62999.3900	104935.490	127418.5050	139961.48												

Groupby

Aplicando groupby considerando mais de uma coluna

```
[141]: grupos_compostos = df3.groupby(by=['Motor','Zero_km'])
```

grupos_compostos.groups

```
dict([(True, Int64Index([17, 44, 57], dtype='int64'), ('Motor 1.0 8v', True)), (False, Int64Index([117, 44, 57], dtype='int64'), ('Motor 1.0 8v', False)), (True, Int64Index([55, 72, 105, 109, 113, 118, 134, 147, 174, 174, 200, 212, 223, 238], dtype='int64'), ('Motor 1.0 16v', True)), (False, Int64Index([42, 88, 140, 143], dtype='int64'), ('Motor 1.0 16v', False)), (True, Int64Index([131, 33, 46, 64, 75, 98, 130, 165, 171, 178, 196, 225, 236, 239, 244], dtype='int64'), ('Motor 2.0 16v', True)), (False, Int64Index([159, 79, 182], dtype='int64'), ('Motor 2.0 16v', False)), (True, Int64Index([1, 27, 61, 114, 116, 117, 137, 177, 185, 186, 245], dtype='int64'), ('Motor 2.4 Turbo', True)), (False, Int64Index([9, 26, 29, 43, 47, 87, 96, 112, 121, 131, 152, 166, 167, 213, 215, 218], dtype='int64'), ('Motor 2.4 Turbo', False)), (True, Int64Index([16, 19, 26, 46, 50, 73, 91, 93, 100, 104, 126, 163, 164, 209, 259, 260], dtype='int64'), ('Motor 3.0 32v', True)), (False, Int64Index([18, 120, 156, 175, 230, 248], dtype='int64'), ('Motor 3.0 32v', False)), (True, Int64Index([14, 45, 54, 55, 63, 65, 80, 106, 129, 151, 153, 168, 181, 195, 209, 259, 260], dtype='int64'), ('Motor 5.0 V8 R1-Turbo', True)), (False, Int64Index([24, 64, 132, 142], dtype='int64'), ('Motor 5.0 V8 R1-Turbo', False)), (True, Int64Index([1, 23, 62, 69, 74, 86, 89, 111, 119, 145, 148, 149, 157, 173, 202, 210, 211, 220, 233, 240, 256], dtype='int64'), ('Motor 4.0 Turbo', True)), (False, Int64Index([60, 78, 124, 189, 237], dtype='int64'), ('Motor 4.0 Turbo', False)), (True, Int64Index([18, 22, 79, 82, 97, 123, 125, 141, 146, 158, 169, 180, 183, 184, 205, 219, 234, 257], dtype='int64'), ('Motor Diesel 1.0', True)), (False, Int64Index([25, 68, 243], dtype='int64'), ('Motor Diesel 1.0', False)), (True, Int64Index([2, 11, 15, 110, 120, 122, 127, 128, 172, 182, 187, 193, 227, 231], dtype='int64'), ('Motor Diesel 1.5', True)), (False, Int64Index([48, 85, 159, 191, 281, 282], dtype='int64'), ('Motor Diesel 1.5', False)), (True, Int64Index([12, 41, 56, 61, 83, 92, 101, 104, 131, 136, 155, 217, 224, 241, 246, 247], dtype='int64'), ('Motor Diesel 2.0', True)), (False, Int64Index([113, 37, 208, 235], dtype='int64'), ('Motor Diesel 2.0', False)), (True, Int64Index([34, 35, 32, 76, 77, 107, 115, 154, 161, 192, 194, 198, 232, 256, 251, 253, 254], dtype='int64'), ('Motor V6', True)), (False, Int64Index([21, 58, 66, 99, 130, 204, 207, 221, 242], dtype='int64'), ('Motor V6', False))])
```

Introdução
Manipulando Linhas e Colunas
Filtrando e Tratando os Dados
Extraindo Informações dos Dados
Plotando Dados
Material Complementar

Valores e Contagem de Valores
Estatísticas
Groupby

Groupby

Utilizando o describe para extrair várias estatísticas dos grupos

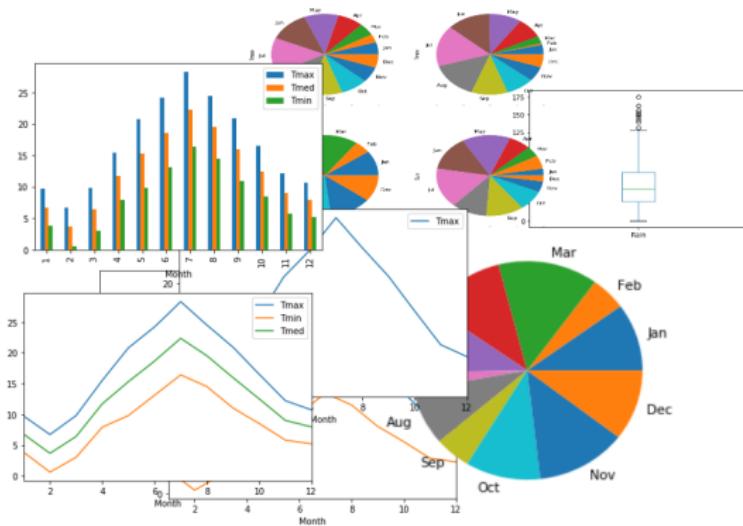
D	Ase	quilometragem										Valor													
		count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
Motor 1.0 I4v	False	22.0	2003.000000	5.953489	1991.0	1997.00	2002.5	2008.50	2018.0	22.0	52956.318182	2025.952929	4339.0	20843.75	53117.0	79661.50	115920.0	22.0	95741.644093	31011.092929	52055.91	49933.6625	89306.213	123054.2700	148203.34
Motor 1.0 I4v	True	3.0	2019.000000	0.000000	2019.0	2019.00	2019.0	2019.0	2019.0	3.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	3.0	123189.036667	22711.773793	60554.39	103089.5600	117623.730	125477.9900	133329.19
Motor 1.8 16v	False	14.0	2007.323571	6.352349	1993.0	2001.25	2009.0	2015.25	2018.0	14.0	39850.214286	38649.021502	4821.0	11807.25	18157.5	52395.25	116000.0	14.0	83963.033571	24304.445718	53054.81	82762.8300	81887.875	97901.0290	112928.11
Motor 1.8 16v	True	4.0	2019.000000	0.000000	2019.0	2019.00	2019.0	2019.0	2019.0	4.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	4.0	83948.432560	36949.984872	60468.61	79995.0719	8483.7220	98076.8790	105086.28
Motor 2.0 16v	False	15.0	2002.200000	5.389994	1990.0	1994.50	2001.0	2012.50	2015.0	15.0	66963.933333	36882.964880	3293.0	3131.50	50996.50	104966.50	118286.0	15.0	93863.430000	16109.384498	60818.92	6872.9750	75263.240	122001.4400	142823.02
Motor 2.0 16v	True	3.0	2019.000000	0.000000	2019.0	2019.00	2019.0	2019.00	2019.0	3.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	3.0	84480.596667	3952.582412	50742.10	8187.6650	112033.270	117042.8450	122018.42
Motor 2.4 Turbo	False	6.0	2005.199887	8.183316	1995.0	1999.25	2006.0	2008.00	2017.0	6.0	87887.500000	31174.762033	25757.0	92487.50	97486.0	102798.00	110966.0	6.0	94673.456000	16101.328861	79762.02	8347.0755	99662.275	116114.125	122975.63
Motor 3.0 32V	False	4.0	2012.000000	0.000000	2013.0	2019.00	2019.0	2019.00	2019.0	4.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	4.0	83212.255000	23101.530001	51605.39	88558.0205	108570.840	117036.0725	124549.07
Motor 3.0 32V	True	17.0	2007.294118	7.069988	1993.0	2004.00	2008.0	2013.00	2017.0	17.0	59900.588235	35176.339952	2827.0	28834.00	98546.0	89594.00	119760.0	17.0	9108.872941	29954.318591	53630.31	611218.5900	84222.850	117708.7000	141852.89
Motor 4.0 Turbo	False	10.0	2019.000000	0.000000	2019.0	2019.00	2019.0	2019.0	2019.0	10.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	10.0	93814.988000	2304.713591	56453.94	8288.9500	10000.944	112033.7000	149157.17
Motor 4.0 Turbo	True	19.0	2004.815799	7.698998	1990.0	2001.50	2006.50	2009.50	2018.0	19.0	66933.384421	36651.025348	2395.0	42802.00	85959.0	84754.50	119945.0	19.0	109443.426826	2012.918441	60968.33	91224.6000	107342.680	126022.8500	141561.06
Motor 5.0 V8 Bi-Turbo	False	17.0	2004.000000	7.920382	1982.0	1996.00	2002.0	2011.00	2015.0	17.0	39493.470588	26408.778237	3173.0	25356.00	25325.0	49500.00	95648.0	17.0	89052.021765	30023.884781	56208.70	66433.0200	70808.030	124661.3000	149034.00
Motor Diesel	False	21.0	2004.819458	9.307136	1990.0	1996.00	2004.0	2014.00	2017.0	21.0	59930.809324	41264.118372	5712.0	19998.0	50204.0	99002.0	118895.0	21.0	20749.737948	30790.398890	54053.18	77838.2000	112293.820	133030.0000	149731.81
Motor Diesel VII	False	5.0	2013.000000	0.000000	2019.0	2019.00	2019.0	2019.00	2019.0	5.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	5.0	10904.886000	34495.778584	61598.79	81998.3000	97290.180	138587.2700	145317.70
Motor Diesel VII	True	19.0	1995.500000	7.702117	1990.0	1992.25	1996.5	2002.00	2018.0	19.0	59690.500000	31374.905544	4420.0	32495.75	51496.6	79981.50	114942.0	19.0	10230.886858	26290.315187	52027.45	78955.7025	109343.250	122881.7500	147872.81
Motor Diesel VII	True	3.0	2019.000000	0.000000	2019.0	2019.00	2019.0	2019.00	2019.0	3.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	3.0	114294.020000	27710.579877	71367.10	101019.0500	12982.210	15622.4800	14288.75
Motor Diesel VII	False	15.0	2004.819333	8.196161	1990.0	2002.50	2007.00	2017.00	2017.0	15.0	61855.805000	30516.139896	307.0	44856.00	58212.50	90704.50	115607.0	15.0	107283.026667	30065.014512	56063.83	73375.0000	99708.180	142240.0000	141849.00
Motor VII	False	6.0	2019.000000	0.000000	2019.0	2019.00	2019.0	2019.00	2019.0	6.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	6.0	105577.045500	39527.045500	64701.00	90128.1425	114755.281	120500.4500	155743.00
Motor VII	True	16.0	2004.005290	7.811376	1993.0	1996.75	2004.0	2017.00	2018.0	16.0	59376.887930	34484.425131	2314.0	54202.50	54202.50	82146.50	119513.0	16.0	108738.689375	36460.062833	54325.87	80258.0000	112438.89	138137.4000	141982.03
Motor VII	False	4.0	2015.000000	0.000000	2019.0	2019.00	2019.0	2019.00	2019.0	4.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	4.0	8847.867950	34405.589511	53153.38	89915.0475	106711.870	124127.5900	132704.75
Motor VII	True	17.0	2002.882393	8.035729	1991.0	1995.00	2001.00	2006.00	2017.0	17.0	65207.529412	37740.000990	810.0	29981.00	75677.8	90604.00	118806.0	17.0	98423.970000	34675.314491	51607.00	61783.0000	115005.5000	127610.0000	139661.40
Motor VII	True	9.0	2019.000000	0.000000	2019.0	2019.00	2019.0	2019.00	2019.0	9.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	9.0	97103.066667	30303.790575	51866.00	64526.4400	126247.0000	131161.0000	

Plotanto Dados

- O Pandas provê funcionalidades para plotagem de gráficos de forma a não necessitar recorrer à outras bibliotecas, pelo menos para alguns modelos básicos de gráficos
- A plotagem de gráficos do pandas utiliza a biblioteca Matplotlib
- Portanto, caso queira manipular as propriedades dos gráficos, como o tamanho, ou para salvar os gráficos, é necessário importar a biblioteca Matplotlib → `import matplotlib.pyplot as plt`

Plotanto Dados

Exemplos de gráficos que podem ser plotados pelo Pandas



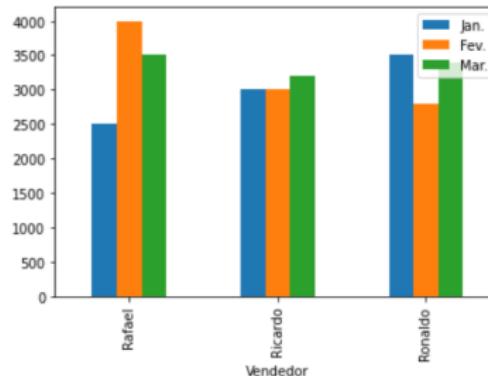
https://miro.medium.com/max/841/0*cMAY3_cWwEo1hTgi.png

Gráfico de Barras

```
[115] df16 = pd.DataFrame({ 'Vendedor': ['Rafael','Ricardo','Ronaldo'],  
                           'Jan.': [2500,3000,3500],  
                           'Fev.': [4000, 3000, 2800],  
                           'Mar.': [3500, 3200, 3400]  
                         })  
df16.index = df16['Vendedor']
```

```
[116] df16.plot(kind='bar')
```

```
Out[116]: <matplotlib.axes._subplots.AxesSubplot at 0x7f481feedb70>
```



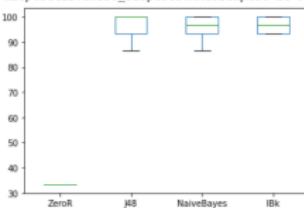
Box

[134] df17

	ZeroR	J48	NaiveBayes	IBk
0	33.333333	93.333333	100.000000	93.333333
1	33.333333	100.000000	100.000000	100.000000
2	33.333333	100.000000	100.000000	100.000000
3	33.333333	100.000000	100.000000	100.000000
4	33.333333	93.333333	93.333333	100.000000
5	33.333333	100.000000	93.333333	93.333333
6	33.333333	86.666667	93.333333	93.333333
7	33.333333	86.666667	86.666667	93.333333
8	33.333333	100.000000	93.333333	93.333333
9	33.333333	100.000000	100.000000	100.000000

df17.plot.box()

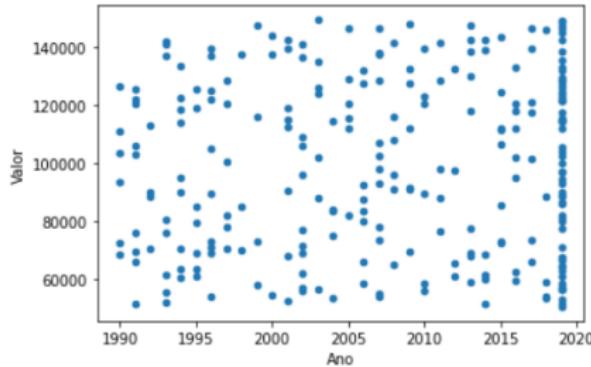
<matplotlib.axes._subplots.AxesSubplot at 0x7f481ee662b0>



Scatter

```
▶ df3.plot.scatter(x='Ano', y='Valor')
```

```
▷ <matplotlib.axes._subplots.AxesSubplot at 0x7f481f008128>
```

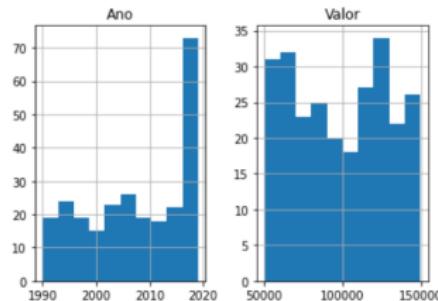


Histograma



```
df3[['Ano', 'Valor']].hist()
```

```
[  
    array([ [             <matplotlib.axes._subplots.AxesSubplot object at 0x7f481f5f29e8>] ],  
          dtype=object)
```



Material Complementar

- User guide

https://pandas.pydata.org/pandas-docs/stable/user_guide

- Pandas: Sort rows or columns in Dataframe based on values using Dataframe.sort_values()

<https://thispointer.com/>

[pandas-sort-rows-or-columns-in-dataframe-based-on-values-using-dataframe-sort_values/](https://thispointer.com/pandas-sort-rows-or-columns-in-dataframe-based-on-values-using-dataframe-sort_values/)

- PYTHON: 10 WAYS TO FILTER PANDAS DATAFRAME

<https://www.listendata.com/2019/07/how-to-filter-pandas-dataframe.html>

- Visualization

https://pandas.pydata.org/docs/user_guide/visualization.html#visualization



Material Complementar

- Merge, join, concatenate and compare

https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html

- Scaling to large datasets

https://pandas.pydata.org/pandas-docs/stable/user_guide/scale.html

- Bar Plots in Python using Pandas DataFrames

<https://www.shanelynn.ie/bar-plots-in-python-using-pandas-dataframes/>

Material Complementar



Tópicos em Inteligência Artificial

<http://ava.ufms.br/>

Rafael Geraldeli Rossi
rafael.g.rossi@ufms.br

Slides baseados nos cursos