

Python

Estatística Descritiva - Pandas

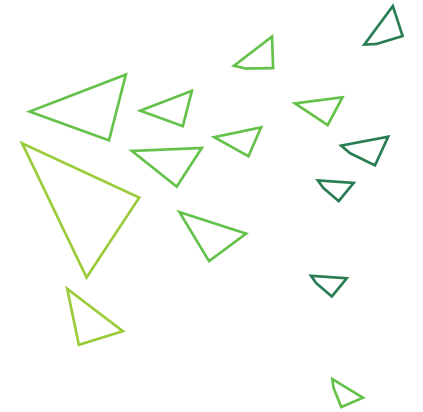
Eduardo Silva – easilva91@gmail.com

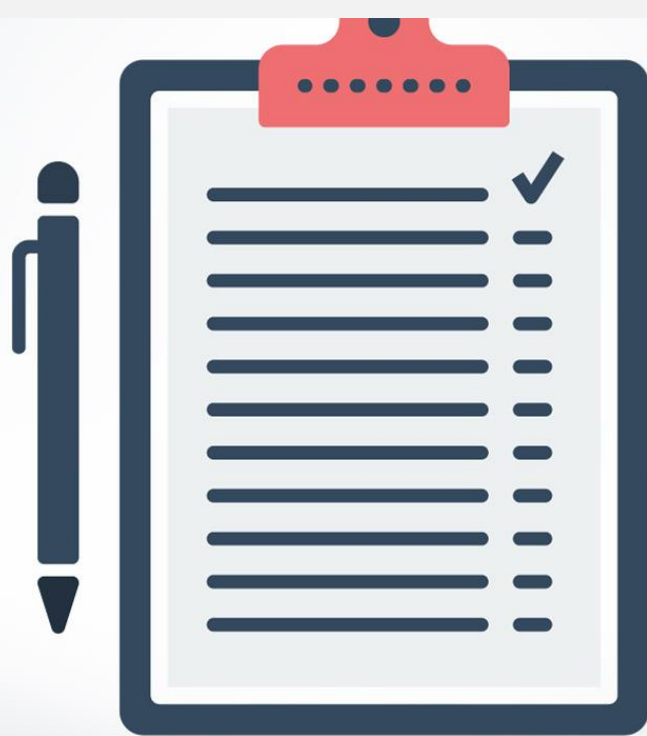
```
31 def __init__(self, path):
32     self.file = None
33     self.fingerprints = set()
34     self.logdups = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, 'requests.log'),
39                         'a')
40         self.file.seek(0)
41         self.fingerprints.update(re.findall(r'(?P<ip>[0-9.]+)', self.file.read()))
42
43 @classmethod
44 def from_settings(cls, settings):
45     debug = settings.getbool('SUPERLITE_DEBUG')
46     return cls(job_dir(settings), debug)
47
48 def request_seen(self, request):
49     fp = self.request_fingerprint(request)
50     if fp in self.fingerprints:
51         return True
52     self.fingerprints.add(fp)
53     if self.file:
54         self.file.write(fp + os.linesep)
55
56 def request_fingerprint(self, request):
57     return request_fingerprint(request)
```




O que é o Pandas?

Pandas é uma lib/módulo do Python comumente utilizada para se trabalhar com dados estruturados. Normalmente utilizada para gerar dataframes e armazenar dados em tabela.





Agenda

Hello Pandas

1. Lendo e Escrevendo com Pandas

- I. O que é o Pandas?
- II. Import do Pandas
- III. Ler e Escrever
 - I. Arquivos CSV
 - II. Outros Arquivos

2. Estrutura de dados com Pandas

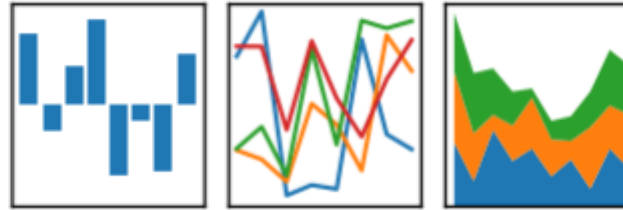
3. Estatística descritiva com Python

4. Quick Tips

O Que é o Pandas?

pandas

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



pandas é uma biblioteca de software escrita para a linguagem de programação Python para manipulação e análise de dados. Em particular, oferece estruturas de dados e operações para manipular tabelas numéricas e séries temporais. É um software livre lançado sob a licença BSD de três cláusulas.

O nome é derivado do termo "painel de dados", um termo econométrico para conjuntos de dados que incluem observações em vários períodos de tempo para os mesmos indivíduos.

[https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software))

Import do Pandas

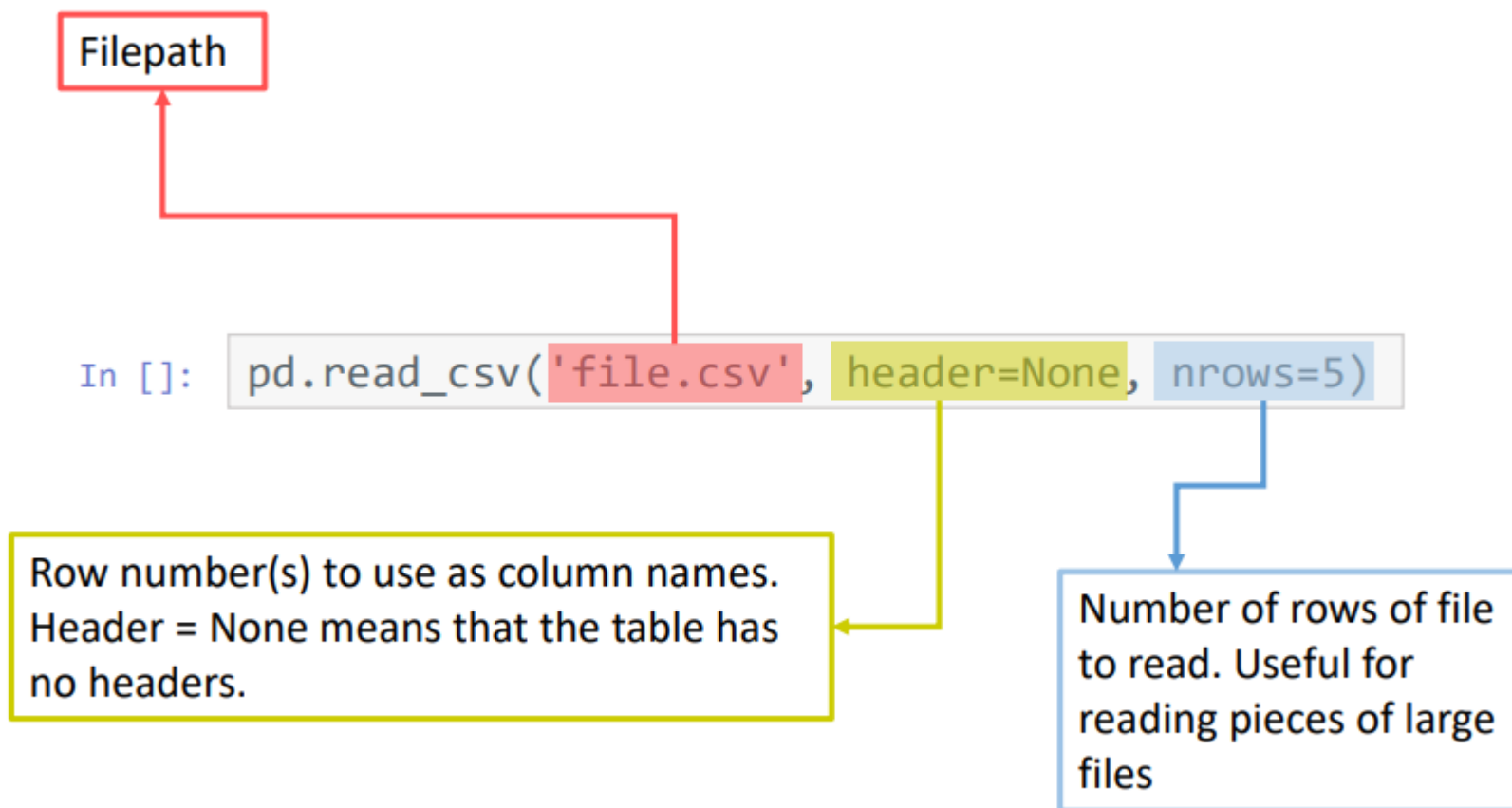
Assim como outros módulos/libs do Python a importação do pandas é bem simples, como demonstra a figura abaixo, o que ocorre é que com bastante frequência se usa uma abreviatura no nome da lib, de forma a facilitar o seu uso e não ter que repetir várias vezes o nome “pandas” dentro do código.

É importante lembrar que o pandas, reconhece os dados como dataframes, o que em suma seria uma estrutura tabular, como a que estamos acostumados a ver no excel.

```
In []: import pandas as pd
```

Ler e Escrever Arquivos em CSV

Leitura de arquivos.

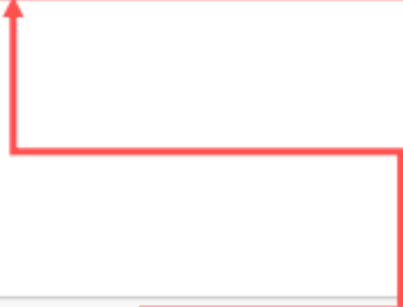


Ler e Escrever Arquivos em CSV

Escrita de arquivos.

Name of the file to be saved

```
In []: df.to_csv('myDataFrame.csv')
```

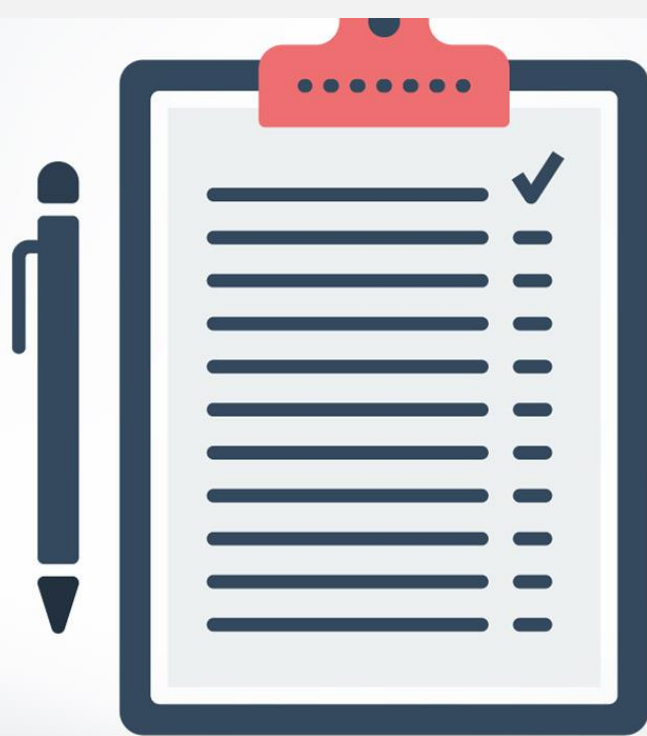


Ler e Escrever Outros Arquivos



Formato	Descrição dos dados	Leitura	Escrita
text	<u>CSV</u>	<u>read_csv</u>	<u>to_csv</u>
text	<u>JSON</u>	<u>read_json</u>	<u>to_json</u>
text	<u>HTML</u>	<u>read_html</u>	<u>to_html</u>
text	Local clipboard	<u>read_clipboard</u>	<u>to_clipboard</u>
binary	<u>MS Excel</u>	<u>read_excel</u>	<u>to_excel</u>
binary	<u>HDF5 Format</u>	<u>read_hdf</u>	<u>to_hdf</u>
binary	<u>Feather Format</u>	<u>read_feather</u>	<u>to_feather</u>
binary	<u>Msgpack</u>	<u>read_msgpack</u>	<u>to_msgpack</u>
binary	<u>Stata</u>	<u>read_stata</u>	<u>to_stata</u>
binary	<u>SAS</u>	<u>read_sas</u>	
binary	<u>Python Pickle Format</u>	<u>read_pickle</u>	<u>to_pickle</u>
SQL	<u>SQL</u>	<u>read_sql</u>	<u>to_sql</u>
SQL	<u>Google Big Query</u>	<u>read_gbq</u>	<u>to_gbq</u>

<http://pandas.pydata.org/pandas-docs/version/0.20/io.html>



Agenda

Hello Pandas

1. Lendo e Escrevendo com Pandas
2. Estrutura de Dados com Pandas
 - I. Series
 - II. DataFrame
3. Estatística Descritiva com Python
4. Quick Tips

Series

- Uma série é uma matriz rotulada unidimensional capaz de manter qualquer tipo de dados.

https://pandas.pydata.org/pandas-docs/stable/getting_started/dsintro.html

```
In []: s = pd.Series([3, -5, 7], index=['a', 'b', 'c'])
```

a	3
b	-5
c	7

DataFrame

- Um DataFrame é uma estrutura de dados rotulada bidimensional com colunas de tipos de dados potencialmente diferentes.

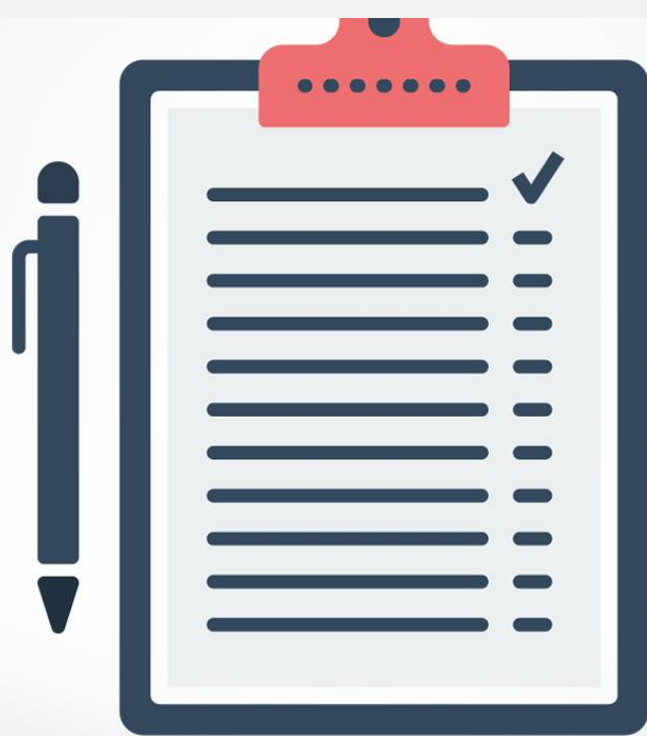
https://pandas.pydata.org/pandas-docs/stable/getting_started/dsintro.html

In []:

```
data = {'Movie Title': ['Se7en', 'Inception', 'The Dark Knight'],  
        'Director': ['David Fincher', 'Christopher Nolan', 'Christopher Nolan'],  
        'IMDB Rating': [8.6, 8.8, 9]}  
df = pd.DataFrame(data, columns=['Movie Title', 'Director', 'IMDB Rating'])
```

The diagram shows a table representing a DataFrame. The columns are labeled 'Movie Title', 'Director', and 'IMDB Rating'. The rows are indexed 0, 1, and 2. A yellow bracket on the left points to the index values, labeled 'Index'. A yellow bracket at the top points to the column headers, labeled 'columns'.

	Movie Title	Director	IMDB Rating
0	Se7en	David Fincher	8.6
1	Inception	Christopher Nolan	8.8
2	The Dark Knight	Christopher Nolan	9



Agenda

Hello Pandas

1. Lendo e Escrevendo com Pandas
2. Estrutura de Dados com Pandas
3. **Estatística Descritiva com Python**
 - I. Recuperando Informações de Serie/DataFrame
 - I. Informações básicas
 - II. Sumários
 - III. Set_index()
 - IV. loc()
 - V. iloc()
4. Quick Tips

Recuperando Informações de Serie/DataFrame

Dados para explorar: <https://www.kaggle.com/danielgrijalvas/movies/version/2>

Os dados podem ser obtidos nesse link: <https://github.com/eduardo2s/Aulas-L3P/blob/master/movies.csv>

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

```
In [5]: df = pd.read_csv('movies.csv', encoding = "ISO-8859-1")
```

```
In [6]: df.head()
```

Out[6]:

	budget	company	country	director	genre	gross	name	rating	released	runtime	score	star	votes	writer	year
0	8000000.0	Columbia Pictures Corporation	USA	Rob Reiner	Adventure	52287414.0	Stand by Me	R	1986-08-22	89	8.1	Wil Wheaton	299174	Stephen King	1986
1	6000000.0	Paramount Pictures	USA	John Hughes	Comedy	70136369.0	Ferris Bueller's Day Off	PG-13	1986-06-11	103	7.8	Matthew Broderick	264740	John Hughes	1986
2	15000000.0	Paramount Pictures	USA	Tony Scott	Action	179800601.0	Top Gun	PG	1986-05-16	110	6.9	Tom Cruise	236909	Jim Cash	1986
3	18500000.0	Twentieth Century Fox Film Corporation	USA	James Cameron	Action	85160248.0	Aliens	R	1986-07-18	137	8.4	Sigourney Weaver	540152	James Cameron	1986
4	9000000.0	Walt Disney Pictures	USA	Randal Kleiser	Adventure	18564613.0	Flight of the Navigator	PG	1986-08-01	90	6.9	Joey Cramer	36636	Mark H. Baker	1986

Recuperando Informações de Serie/DataFrame

Informações Básicas

- Número de linhas e Colunas

```
In [7]: df.shape
```

```
Out[7]: (6820, 15)
```

- Descrever o Index e as Colunas

```
In [8]: df.index
```

```
Out[8]: RangeIndex(start=0, stop=6820, step=1)
```

```
In [9]: df.columns
```

```
Out[9]: Index(['budget', 'company', 'country', 'director', 'genre', 'gross', 'name',  
              'rating', 'released', 'runtime', 'score', 'star', 'votes', 'writer',  
              'year'],  
              dtype='object')
```

Recuperando Informações de Serie/DataFrame

Informações Básicas

- Informações do DataFrame

In [10]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6820 entries, 0 to 6819
Data columns (total 15 columns):
budget      6820 non-null float64
company     6820 non-null object
country     6820 non-null object
director    6820 non-null object
genre       6820 non-null object
gross       6820 non-null float64
name        6820 non-null object
rating      6820 non-null object
released    6820 non-null object
runtime     6820 non-null int64
score       6820 non-null float64
star        6820 non-null object
votes       6820 non-null int64
writer      6820 non-null object
year        6820 non-null int64
dtypes: float64(3), int64(3), object(9)
memory usage: 799.3+ KB
```

Recuperando Informações de Serie/DataFrame

Informações Básicas

- Verificando Valores nulos ou falta de valores

```
In [11]: df.count()
```

```
Out[11]: budget      6820  
company      6820  
country      6820  
director     6820  
genre        6820  
gross        6820  
name         6820  
rating       6820  
released     6820  
runtime      6820  
score        6820  
star         6820  
votes        6820  
writer       6820  
year         6820  
dtype: int64
```

Recuperando Informações de Serie/DataFrame

Sumários

- Soma de Valores

```
In [12]: df.sum()
```

```
Out[12]: budget                1.67643e+11  
company      Columbia Pictures CorporationParamount Picture...  
country      USAUSAUSAUSAUSAUKUKUSAUSAUSAUSAustraliaUKUSAUSA...  
director      Rob ReinerJohn HughesTony ScottJames CameronRa...  
genre      AdventureComedyActionActionAdventureDramaAdven...  
gross                2.28455e+11  
name      Stand by MeFerris Bueller's Day OffTop GunAlie...  
rating      RPG-13PGRPGRPGRPG-13RPG-13RPG-13PG-13RRRPG-13P...  
released      1986-08-221986-06-111986-05-161986-07-181986-0...  
runtime                726680  
score                43476.8  
star      Wil WheatonMatthew BroderickTom CruiseSigourne...  
votes                485717144  
writer      Stephen KingJohn HughesJim CashJames CameronMa...  
year                13646822  
dtype: object
```

Recuperando Informações de Serie/DataFrame

Sumários

- Soma de Valores de Colunas específicas

```
In [18]: df.loc[:, 'gross'].sum()
```

```
Out[18]: 228455191158.0
```

- Soma cumulativa de valores

```
In [27]: df2.loc[:, 'gross'].cumsum()
```

```
Out[27]: company
Columbia Pictures Corporation    5.228741e+07
Paramount Pictures              1.224238e+08
Paramount Pictures              3.022244e+08
Twentieth Century Fox Film Corporation  3.873846e+08
Walt Disney Pictures            4.059492e+08
Hemdale                         5.444798e+08
Henson Associates (HA)          5.572097e+08
De Laurentiis Entertainment Group (DEG) 5.657610e+08
Paramount Pictures              6.062326e+08
SLM Production Group            6.466892e+08
Rimfire Films                   8.213242e+08
Thorn EMI Screen Entertainment    8.272242e+08
Twentieth Century Fox Film Corporation  8.354242e+08
Twentieth Century Fox Film Corporation  8.465242e+08
De Laurentiis Entertainment Group (DEG) 8.551451e+08
Producers Sales Organization (PSO)    8.618800e+08
De Laurentiis Entertainment Group (DEG) 8.693136e+08
Geffen Company, The              9.080610e+08
```


Recuperando Informações de Serie/DataFrame

Sumários

- Valores Mínimos

```
In []: thriller_movies_DF.min()
```

```
Out[]: Gross                100125340.0  
       RT No. Ratings       466323.0  
       IMDB No. Ratings     982734.0  
       RT Average Rating      8.2  
       IMDB Average Rating    8.6  
       dtype: float64
```

- Valores Máximos

```
In []: thriller_movies_DF.max()
```

```
Out[]: Gross                448130642.0  
       RT No. Ratings       1827436.0  
       IMDB No. Ratings     1828227.0  
       RT Average Rating      8.8  
       IMDB Average Rating    9.0  
       dtype: float64
```

Recuperando Informações de Serie/DataFrame

Sumários

- Sumário de Estatísticas

```
In [30]: df.describe().round(2)
```

Out[30]:

	budget	gross	runtime	score	votes	year
count	6.820000e+03	6.820000e+03	6820.00	6820.00	6820.00	6820.00
mean	2.458113e+07	3.349783e+07	106.55	6.37	71219.52	2001.00
std	3.702254e+07	5.819760e+07	18.03	1.00	130517.63	8.94
min	0.000000e+00	7.000000e+01	50.00	1.50	27.00	1986.00
25%	0.000000e+00	1.515839e+06	95.00	5.80	7665.25	1993.00
50%	1.100000e+07	1.213568e+07	102.00	6.40	25892.50	2001.00
75%	3.200000e+07	4.006534e+07	115.00	7.10	75812.25	2009.00
max	3.000000e+08	9.366622e+08	366.00	9.30	1861666.00	2016.00

```
In [31]: df.describe().transpose()
```

Out[31]:

	count	mean	std	min	25%	50%	75%	max
budget	6820.0	2.458113e+07	3.702254e+07	0.0	0.00	11000000.0	32000000.00	300000000.0
gross	6820.0	3.349783e+07	5.819760e+07	70.0	1515839.00	12135679.0	40065340.50	936662225.0
runtime	6820.0	1.065513e+02	1.802818e+01	50.0	95.00	102.0	115.00	366.0
score	6820.0	6.374897e+00	1.003142e+00	1.5	5.80	6.4	7.10	9.3
votes	6820.0	7.121952e+04	1.305176e+05	27.0	7665.25	25892.5	75812.25	1861666.0
year	6820.0	2.001000e+03	8.944501e+00	1986.0	1993.00	2001.0	2009.00	2016.0

Recuperando Informações de Serie/DataFrame

Sumários

- Média

```
In [32]: df.mean()
```

```
Out[32]: budget      2.458113e+07  
gross        3.349783e+07  
runtime      1.065513e+02  
score        6.374897e+00  
votes        7.121952e+04  
year         2.001000e+03  
dtype: float64
```

- Mediana

```
In [33]: df.median()
```

```
Out[33]: budget      11000000.0  
gross        12135679.0  
runtime         102.0  
score           6.4  
votes          25892.5  
year           2001.0  
dtype: float64
```

Recuperando Informações de Serie/DataFrame

.loc e .iloc

O loc e iloc servem como uma forma de busca e recuperação de informação dentro do dataframe.

```
In [36]: #podemos chamar uma linha pelo seu índice  
df.loc[5]
```

```
Out[36]: budget          6e+06  
company          Hemdale  
country           UK  
director    Oliver Stone  
genre          Drama  
gross          1.38531e+08  
name          Platoon  
rating           R  
released    1987-02-06  
runtime          120  
score           8.1  
star      Charlie Sheen  
votes          317585  
writer    Oliver Stone  
year           1986  
Name: 5, dtype: object
```

```
In [37]: #ou com um array de índices  
df.loc[[0,1,2]]
```

```
Out[37]:
```

	budget	company	country	director	genre	gross	name	rating	released	runtime	score	star	votes	writer	year
0	8000000.0	Columbia Pictures Corporation	USA	Rob Reiner	Adventure	52287414.0	Stand by Me	R	1986-08-22	89	8.1	Wil Wheaton	299174	Stephen King	1986
1	6000000.0	Paramount Pictures	USA	John Hughes	Comedy	70136369.0	Ferris Bueller's Day Off	PG-13	1986-06-11	103	7.8	Matthew Broderick	264740	John Hughes	1986
2	15000000.0	Paramount Pictures	USA	Tony Scott	Action	179800601.0	Top Gun	PG	1986-05-16	110	6.9	Tom Cruise	236909	Jim Cash	1986

Recuperando Informações de Serie/DataFrame

set_index()

Uma última observação, durante a apresentação é possível ver os DataFrames: df e df2, a diferença entre eles é o seu índice ou índice, por padrão o pandas gera um índice número de 0 até o último item, no entanto, por vezes é interessante ter determinado conteúdo no índice, para tal basta usar o `set_index()` para fazer essa alteração.

In [28]: df

Out[28]:

	INDEX	budget	company	country	director	genre	gross	name	rating	released	runtime	score	star	votes
0		8000000.0	Columbia Pictures Corporation	USA	Rob Reiner	Adventure	52287414.0	Stand by Me	R	1986-08-22	89	8.1	Wil Wheaton	299174
1		6000000.0	Paramount Pictures	USA	John Hughes	Comedy	70136369.0	Ferris Bueller's Day Off	PG-13	1986-06-11	103	7.8	Matthew Broderick	264740
2		15000000.0	Paramount Pictures	USA	Tony Scott	Action	179800601.0	Top Gun	PG	1986-05-16	110	6.9	Tom Cruise	236909

In [24]: df2 = df.set_index('company')

In [42]: df2.head()

Out[42]:

	INDEX	budget	country	director	genre	gross	name	rating	released	runtime	score	star	votes	writer	year
	company														
	Columbia Pictures Corporation	8000000.0	USA	Rob Reiner	Adventure	52287414.0	Stand by Me	R	1986-08-22	89	8.1	Wil Wheaton	299174	Stephen King	1986
	Paramount Pictures	6000000.0	USA	John Hughes	Comedy	70136369.0	Ferris Bueller's Day Off	PG-13	1986-06-11	103	7.8	Matthew Broderick	264740	John Hughes	1986
	Paramount Pictures	15000000.0	USA	Tony Scott	Action	179800601.0	Top Gun	PG	1986-05-16	110	6.9	Tom Cruise	236909	Jim Cash	1986

Recuperando Informações de Serie/DataFrame

.loc e .iloc

Para está etapa foi criada uma cópia do primeida DataFrame “df” chamada “df2”. A diferença se encontra no índice, que passou de numérico para a coluna “company”. Para tal foi utilizado o set_index(), que foi visto no slide anterior.

```
In [38]: # tambem podemos chamar diretamente pela linha  
df2.loc['Paramount Pictures']
```

```
[38]:
```

	budget	country	director	genre	gross	name	rating	released	runtime	score	star	votes	writer	year
company														
Paramount Pictures	6000000.0	USA	John Hughes	Comedy	70136369.0	Ferris Bueller's Day Off	PG-13	1986-06-11	103	7.8	Matthew Broderick	264740	John Hughes	1986
Paramount Pictures	15000000.0	USA	Tony Scott	Action	179800601.0	Top Gun	PG	1986-05-16	110	6.9	Tom Cruise	236909	Jim Cash	1986
Paramount Pictures	9000000.0	USA	Howard Deutch	Comedy	40471663.0	Pretty in Pink	PG-13	1986-02-28	96	6.8	Molly Ringwald	60565	John Hughes	1986
Paramount Pictures	25000000.0	USA	Leonard Nimoy	Adventure	109713132.0	Star Trek IV: The Voyage Home	PG	1986-11-26	119	7.3	William Shatner	66366	Gene Roddenberry	1986
Paramount Pictures	25000000.0	USA	Michael Ritchie	Action	79817937.0	The Golden Child	PG-13	1986-12-12	94	5.9	Eddie Murphy	42997	Dennis Feldman	1986
Paramount Pictures	3000000.0	USA	Tom McLoughlin	Horror	19472057.0	Jason Lives: Friday the 13th Part 5	R	1986-08-01	86	5.9	Thom Mathews	28310	Tom McLoughlin	1986

Recuperando Informações de Serie/DataFrame

.loc e .iloc

```
In [39]: # Selecionar um nome somente com as colunas de interesse  
df2.loc[['Paramount Pictures'],['country','score','star']]
```

Out[39]:

	country	score	star
company			
Paramount Pictures	USA	7.8	Matthew Broderick
Paramount Pictures	USA	6.9	Tom Cruise
Paramount Pictures	USA	6.8	Molly Ringwald
Paramount Pictures	USA	7.3	William Shatner
Paramount Pictures	USA	5.9	Eddie Murphy
Paramount Pictures	USA	5.9	Thom Mathews
Paramount Pictures	USA	6.1	Meryl Streep
Paramount Pictures	USA	6.2	Michael Keaton
Paramount Pictures	USA	7.2	William Hurt
Paramount Pictures	USA	6.2	Deborah Foreman

Recuperando Informações de Serie/DataFrame

.loc e .iloc

Usando operadores lógicos (todos os filmes com duração maior que 90 min)

```
In [40]: df.loc[(df['runtime']) >= 90]
```

Out[40]:

	budget	company	country	director	genre	gross	name	rating	released	runtime	score	star	votes	writer	year
1	6000000.0	Paramount Pictures	USA	John Hughes	Comedy	70136369.0	Ferris Bueller's Day Off	PG-13	1986-06-11	103	7.8	Matthew Broderick	264740	John Hughes	1986
2	15000000.0	Paramount Pictures	USA	Tony Scott	Action	179800601.0	Top Gun	PG	1986-05-16	110	6.9	Tom Cruise	236909	Jim Cash	1986
3	18500000.0	Twentieth Century Fox Film Corporation	USA	James Cameron	Action	85160248.0	Aliens	R	1986-07-18	137	8.4	Sigourney Weaver	540152	James Cameron	1986
4	9000000.0	Walt Disney Pictures	USA	Randal Kleiser	Adventure	18564613.0	Flight of the Navigator	PG	1986-08-01	90	6.9	Joey Cramer	36636	Mark H. Baker	1986
5	6000000.0	Hemdale	UK	Oliver Stone	Drama	138530565.0	Platoon	R	1987-02-06	120	8.1	Charlie Sheen	317585	Oliver Stone	1986
6	25000000.0	Henson Associates (HA)	UK	Jim Henson	Adventure	12729917.0	Labyrinth	PG	1986-06-27	101	7.4	David Bowie	102879	Dennis Lee	1986

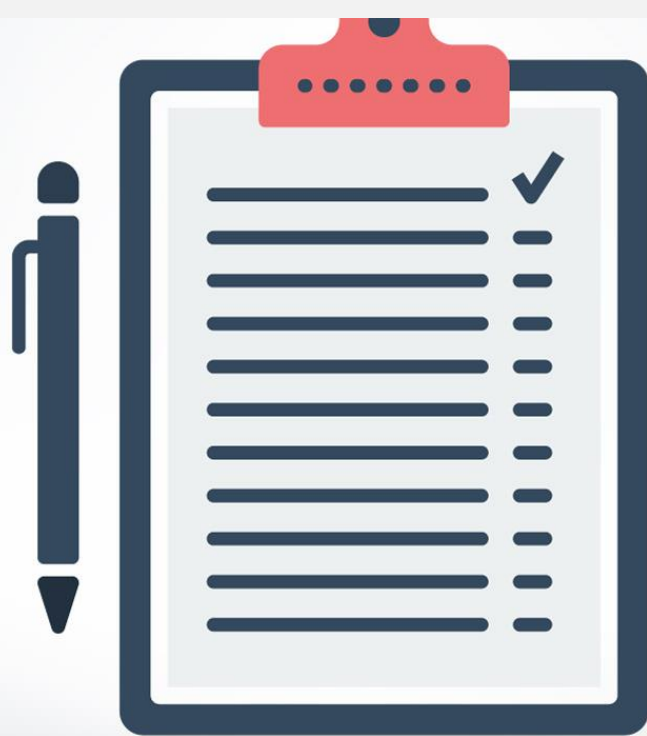
Recuperando Informações de Serie/DataFrame

.loc e .iloc

O iloc é mais simples que o loc, uma vez que usa dados numéricos para a busca, como no exemplo abaixo ele retorna os valores de 1 a 5 da última coluna.

```
In [41]: df.iloc[0:5, -1]
```

```
Out[41]: 0    1986  
         1    1986  
         2    1986  
         3    1986  
         4    1986  
         Name: year, dtype: int64
```



Agenda

Hello Pandas

1. Lendo e Escrevendo com Pandas
2. Estrutura de Dados com Pandas
3. Estatística Descritiva com Python
4. Quick Tips

Quick Tips

Como instalar libs/módulos no Python:

```
In [ ]: !pip install pandas ou pip install pandas
```

Com importar uma lib para utilizar no código:

```
In [ ]: import pandas as pd  
df = pd.read_csv('movies.csv', encoding = "ISO-8859-1")
```

Algumas Notas

O conteúdo que vimos nessa aula trata da lib pandas, que se trata de uma lib do Python, utilizada para se trabalhar com dados estruturados ou seja tabelas, bancos de dados, entre outros, todo o conteúdo aqui apresentado tenta demonstrar como é possível usar algumas bases de estatística descritiva utilizando essa lib.

Com ela é possível ter uma breve descrição dos dados (`.describe()`), se encontrar os mínimos e máximos de uma determinada coluna ou de todos os dados (`.min()` `.max()`), além da possibilidade de informações mais específicas utilizando o `.loc()` e o `.iloc()`.

No decorrer das aulas iremos utilizar muito o pandas e a estrutura de DataFrame, será uma lib com bastante importância.



Real Python