ExerciciosdoLivro

February 24, 2025

[4]:
```
pip install pandas_datareader
```

```
Defaulting to user installation because normal site-packages is not writeable
Looking in links: /usr/share/pip-wheels
Collecting pandas_datareader
  Downloading pandas_datareader-0.10.0-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: lxml in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas_datareader) (4.9.3)
Requirement already satisfied: pandas>=0.23 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas_datareader) (2.1.4)
Requirement already satisfied: requests>=2.19.0 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas_datareader) (2.31.0)
Requirement already satisfied: numpy<2,>=1.22.4 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas>=0.23->pandas_datareader) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas>=0.23->pandas_datareader) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas>=0.23->pandas_datareader) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas>=0.23->pandas_datareader) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
requests>=2.19.0->pandas_datareader) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
requests>=2.19.0->pandas_datareader) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
requests>=2.19.0->pandas_datareader) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
```

```
requests>=2.19.0->pandas_datareader) (2024.2.2)
Requirement already satisfied: six>=1.5 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
python-dateutil>=2.8.2->pandas>=0.23->pandas_datareader) (1.16.0)
Downloading pandas_datareader-0.10.0-py3-none-any.whl (109 kB)
                           109.5/109.5
kB 1.0 MB/s eta 0:00:008 MB/s eta 0:00:01
Installing collected packages: pandas_datareader
Successfully installed pandas_datareader-0.10.0
Note: you may need to restart the kernel to use updated packages.
```

[16]:
```
import matplotlib.pyplot as fig
import datetime as dt
#import pandas_datareader.data as web
import pandas as pd
```

[161]:
```
bovespa = None

with open(file='./bcdata24026.csv', mode='r', encoding='utf8') as arquivo:
  bovespa = arquivo.read()
  print(bovespa)
```

```
"data";"valor"
"01/01/2002";"-212"
"01/04/2002";"0"
"01/07/2002";"0"
"01/10/2002";"105"
"01/01/2003";"-5"
"01/04/2003";"-285"
"01/07/2003";"-325"
"01/10/2003";"81"
"01/01/2004";"-78"
"01/04/2004";"6"
"01/07/2004";"-221"
"01/10/2004";"109"
"01/01/2005";"-33"
"01/04/2005";"-254"
"01/07/2005";"-332"
"01/10/2005";"119"
"01/01/2006";"-41"
"01/04/2006";"-152"
"01/07/2006";"-252"
"01/10/2006";"113"
"01/01/2007";"81"
"01/04/2007";"25"
"01/07/2007";"-50"
"01/10/2007";"142"
```

```
"01/01/2008";"211"
"01/04/2008";"158"
"01/07/2008";"99"
"01/10/2008";"609"
"01/01/2009";"360"
"01/04/2009";"315"
"01/07/2009";"302"
"01/10/2009";"426"
"01/01/2010";"406"
"01/04/2010";"352"
"01/07/2010";"353"
"01/10/2010";"797"
"01/01/2011";"722"
"01/04/2011";"660"
"01/07/2011";"807"
"01/10/2011";"668"
"01/01/2012";"495"
"01/04/2012";"725"
"01/07/2012";"829"
"01/10/2012";"555"
"01/01/2013";"716"
"01/04/2013";"624"
"01/07/2013";"670"
"01/10/2013";"647"
"01/01/2014";"650"
"01/04/2014";"615"
"01/07/2014";"555"
"01/10/2014";"594"
"01/01/2015";"607"
"01/04/2015";"591"
"01/07/2015";"603"
"01/10/2015";"680"
"01/01/2016";"718"
"01/04/2016";"959"
"01/07/2016";"794"
"01/10/2016";"730"
"01/01/2017";"777"
"01/04/2017";"627"
"01/07/2017";"785"
"01/10/2017";"1219"
"01/01/2018";"1334"
"01/04/2018";"1577"
"01/07/2018";"1608"
"01/10/2018";"2395"
"01/01/2019";"2117"
"01/04/2019";"1952"
"01/07/2019";"2004"
"01/10/2019";"1721"
```

```
"01/01/2020";"1787"
"01/04/2020";"1108"
"01/07/2020";"993"
"01/10/2020";"1382"
"01/01/2021";"1563"
"01/04/2021";"1563"
"01/07/2021";"1351"
"01/10/2021";"1429"
"01/01/2022";"2018"
"01/04/2022";"1424"
"01/07/2022";"1285"
"01/10/2022";"838"
"01/01/2023";"819"
"01/04/2023";"866"
"01/07/2023";"946"
"01/10/2023";"1148"
"01/01/2024";"874"
"01/04/2024";"873"
"01/07/2024";"874"
```

[159]:
```
df = pd.read_csv('bcdata24026.csv', sep=',')
df
```

[159]:
```
          data;"valor"
0   01/01/2002;"-212"
1      01/04/2002;"0"
2      01/07/2002;"0"
3    01/10/2002;"105"
4     01/01/2003;"-5"
..                 …
86   01/07/2023;"946"
87  01/10/2023;"1148"
88   01/01/2024;"874"
89   01/04/2024;"873"
90   01/07/2024;"874"

[91 rows x 1 columns]
```

[163]:
```
df.shape
```

[163]:
```
(91, 1)
```

[165]:
```
df.head(10)
```

[165]:
```
         data;"valor"
0  01/01/2002;"-212"
```

```
1     01/04/2002;"0"
2     01/07/2002;"0"
3    01/10/2002;"105"
4     01/01/2003;"-5"
5   01/04/2003;"-285"
6   01/07/2003;"-325"
7     01/10/2003;"81"
8    01/01/2004;"-78"
9     01/04/2004;"6"
```

[44]:
```
inicio = dt.datetime(2019,1,1)
fim = dt.datetime(2019,11,24)
print(inicio)
print(fim)
```

```
2019-01-01 00:00:00
2019-11-24 00:00:00
```

[167]:
```
df.columns
```

[167]:
```
Index(['data;"valor"'], dtype='object')
```

[179]:
```
df
```

[179]:
```
          data;"valor"
0   01/01/2002;"-212"
1      01/04/2002;"0"
2      01/07/2002;"0"
3    01/10/2002;"105"
4     01/01/2003;"-5"
..                 …
86   01/07/2023;"946"
87  01/10/2023;"1148"
88   01/01/2024;"874"
89   01/04/2024;"873"
90   01/07/2024;"874"

[91 rows x 1 columns]
```

[82]:
```
print(df['Ticker'].tail(5))
```

```
14972    NaN
14973    NaN
14974    NaN
14975    NaN
14976    NaN
Name: Ticker, dtype: object
```

[72]:
```
#inicio = dt.datetime(2019,1,1)
#fim = dt.datetime(2019,11,24)
#df = pd.read_csv('bovespa1.csv', sep=',')
#dta=(df['Close'])
#dta
#fig.grid()
#fig.title('Petrobrás (Jan-Nov)', fontsize=18,weight='bold')
```

[72]:
```
0        NaN
1        NaN
2        NaN
3        NaN
4        NaN
          ..
14972    NaN
14973    NaN
14974    NaN
14975    NaN
14976    NaN
Name: Close, Length: 14977, dtype: float64
```

[74]:
```
df['Close'].nunique()
```

[74]:
```
5
```

[76]:
```
df.shape
```

[76]:
```
(14977, 8)
```

[90]:
```
id_close = df['High'].value_counts()
id_close.head()
```

[90]:
```
High
8.0     2
1.0     1
16.0    1
12.0    1
20.0    1
Name: count, dtype: int64
```

[80]:
```
id_close.value_counts()
```

[80]:
```
count
1    4
2    1
Name: count, dtype: int64
```

[94]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14977 entries, 0 to 14976
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       14977 non-null  object 
 1   Ticker     6 non-null      object 
 2   Open       6 non-null      float64
 3   High       6 non-null      float64
 4   Low        6 non-null      float64
 5   Close      6 non-null      float64
 6   Volume     6 non-null      float64
 7   Adj Close  6 non-null      float64
dtypes: float64(6), object(2)
memory usage: 936.2+ KB
```

[97]:
```
df['Low'].head(10)
```

[97]:
```
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN
5   NaN
6   NaN
7   NaN
8   NaN
9   NaN
Name: Low, dtype: float64
```

[103]:
```
valnonull = df['Low'] == 'non-null'
valnonull[0:100]
```

[103]:
```
0     False
1     False
2     False
3     False
4     False
      …
95    False
96    False
97    False
98    False
99    False
Name: Low, Length: 100, dtype: bool
```

[137]:
```
df['Date'].describe()
```

[137]:
```
count        14977
unique       14976
top       7/9/2016
freq             2
Name: Date, dtype: object
```

[135]:
```
df[['Volume','High','Date']].describe()
```

[135]:
```
        Volume       High
count      6.0   6.000000
mean       0.0  10.833333
std        0.0   6.705719
min        0.0   1.000000
25%        0.0   8.000000
50%        0.0  10.000000
75%        0.0  15.000000
max        0.0  20.000000
```

[139]:
```
df.columns
```

[139]:
```
Index(['Date', 'Ticker', 'Open', 'High', 'Low', 'Close', 'Volume',
       'Adj Close'],
      dtype='object')
```

[143]:
```
df['Date'].tail()
```

[143]:
```
14972    5/10/2015,ELPL4,"10,68","10,82","10,44","10,64…
14973    2/10/2015,ELPL4,"10,51","10,81","10,26","10,52…
14974    1/10/2015,ELPL4,"10,94","11,2","10,52","10,57"…
14975    30/9/2015,ELPL4,"10,78","11,05","10,66","10,95…
14976    29/9/2015,ELPL4,"10,83","11,05","10,42","10,77…
Name: Date, dtype: object
```

[157]:
```
clos = df['Close']
clos.tail(5)
```

[157]:
```
14972   NaN
14973   NaN
14974   NaN
14975   NaN
14976   NaN
Name: Close, dtype: float64
```

[181]:
```
import math
```

[185]:
```
math.exp(5)
```

[185]: 148.4131591025766

[187]:
```
math.log2(3)
```

[187]: 1.584962500721156

[189]:
```
math.log10(4)
```

[189]: 0.6020599913279624

[195]:
```
math.exp(1)
```

[195]: 2.718281828459045

[197]:
```
math.pow(2,3)
```

[197]: 8.0

[200]:
```
math.sqrt(16)
```

[200]: 4.0

[202]:
```
import math as mt
```

[204]:
```
mt.cos(3)
```

[204]: -0.9899924966004454

[212]:
```
dados = [10,5,1,1,2,2,3,5,10,2,2,1,3,4,3]
```

[208]:
```
print(dados[0])
```

```
10
```

[214]:
```
print(dados[-1])
```

```
3
```

[218]:
```
print(len(dados))
```

```
15
```

[220]:
```
mercado=['açoes','opçoes','futuro','dolar','ouro','criptomoeda']
print(mercado)
```

```
['açoes', 'opçoes', 'futuro', 'dolar', 'ouro', 'criptomoeda']
```

[224]:
```
print(mercado[0:3])
```

```
['açoes', 'opçoes', 'futuro']
```

[226]:
```
'futuro' in mercado
```

[226]:
```
True
```

[228]:
```
'valeria' in mercado
```

[228]:
```
False
```

[234]:
```
print(" 'futuro' esta na lista?", fut)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[234], line 1
----> 1 print(" 'futuro' esta na lista?", fut)

NameError: name 'fut' is not defined
```

[238]:
```
mercado[2]='commodity'
```

[240]:
```
print(mercado)
```

```
['açoes', 'opçoes', 'commodity', 'dolar', 'ouro', 'criptomoeda']
```

[242]:
```
mercado[0:2]=['tesouro','tesouros']
```

[244]:
```
print(mercado)
```

```
['tesouro', 'tesouros', 'commodity', 'dolar', 'ouro', 'criptomoeda']
```

[252]:
```
mercado.append('dolar')
```

[254]:
```
print(mercado)
```

```
['tesouro', 'tesouros', 'commodity', 'dolar', 'ouro', 'criptomoeda', 'comprar',
'dolar']
```

[256]:
```
mercado.count('dolar')
```

[256]:
```
2
```

[258]:
```
mercado.extend(['Petrobras','BB','Vale'])
```

[260]:
```
print(mercado)
```

```
['tesouro', 'tesouros', 'commodity', 'dolar', 'ouro', 'criptomoeda', 'comprar',
'dolar', 'Petrobras', 'BB', 'Vale']
```

[262]:
```
mercado.sort()
```

[264]:
```
print(mercado)
```

```
['BB', 'Petrobras', 'Vale', 'commodity', 'comprar', 'criptomoeda', 'dolar',
'dolar', 'ouro', 'tesouro', 'tesouros']
```

[269]:
```
mercado.sort(key=str.casefolg)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[269], line 1
----> 1 mercado.sort(key=str.casefolg)

AttributeError: type object 'str' has no attribute 'casefolg'
```

[271]:
```
mercado.reverse()
```

[273]:
```
print(mercado)
```

```
['tesouros', 'tesouro', 'ouro', 'dolar', 'dolar', 'criptomoeda', 'comprar',
'commodity', 'Vale', 'Petrobras', 'BB']
```

[275]:
```
mercado.remove('ouro')
```

[277]:
```
print(mercado)
```

```
['tesouros', 'tesouro', 'dolar', 'dolar', 'criptomoeda', 'comprar', 'commodity',
'Vale', 'Petrobras', 'BB']
```

[279]:
```
mercado.index('Petrobras')
```

[279]:
```
8
```

[281]:
```
mercado.insert(2, 'Fundo de Investimento')
```

[283]:
```
print(mercado)
```

```
['tesouros', 'tesouro', 'Fundo de Investimento', 'dolar', 'dolar',
'criptomoeda', 'comprar', 'commodity', 'Vale', 'Petrobras', 'BB']
```

[286]:
```
listaderivados = ['imoveis', 'dolar', 'açoes', 'PGBL', 'VGBL']
```

[288]:
```
print(listaderivados)
```

```
['imoveis', 'dolar', 'açoes', 'PGBL', 'VGBL']
```

[292]:
```
listaderivados.index('dolar')
```

[292]:
```
1
```

[296]:
```
listaderivados.insert(3, 'poupança')
```

[298]:
```
print(listaderivados)
```

```
['imoveis', 'dolar', 'açoes', 'poupança', 'PGBL', 'VGBL']
```

[300]:
```
print(listaderivados[0:4])
```

```
['imoveis', 'dolar', 'açoes', 'poupança']
```

[304]:
```
import statistics as st
```

[308]:
```
prec = [10,11,11,10,10,10,8,8,9,7,11,12,13,8,9]
```

[311]:
```
print(prec)
```

```
[10, 11, 11, 10, 10, 10, 8, 8, 9, 7, 11, 12, 13, 8, 9]
```

[313]:
```
st.mean(prec)
```

[313]:
```
9.8
```

[315]:
```
st.median(prec)
```

[315]:
```
10
```

[317]:
```
st.mode(prec)
```

[317]:
```
10
```

[319]:
```
st.stdev(prec)
```

[319]:
```
1.65615734242165
```

[321]:
```
import matplotlib.pyplot as fig
import numpy as ny
```

[329]:
```
x = ny.arange(1, 20)
print(x)
```

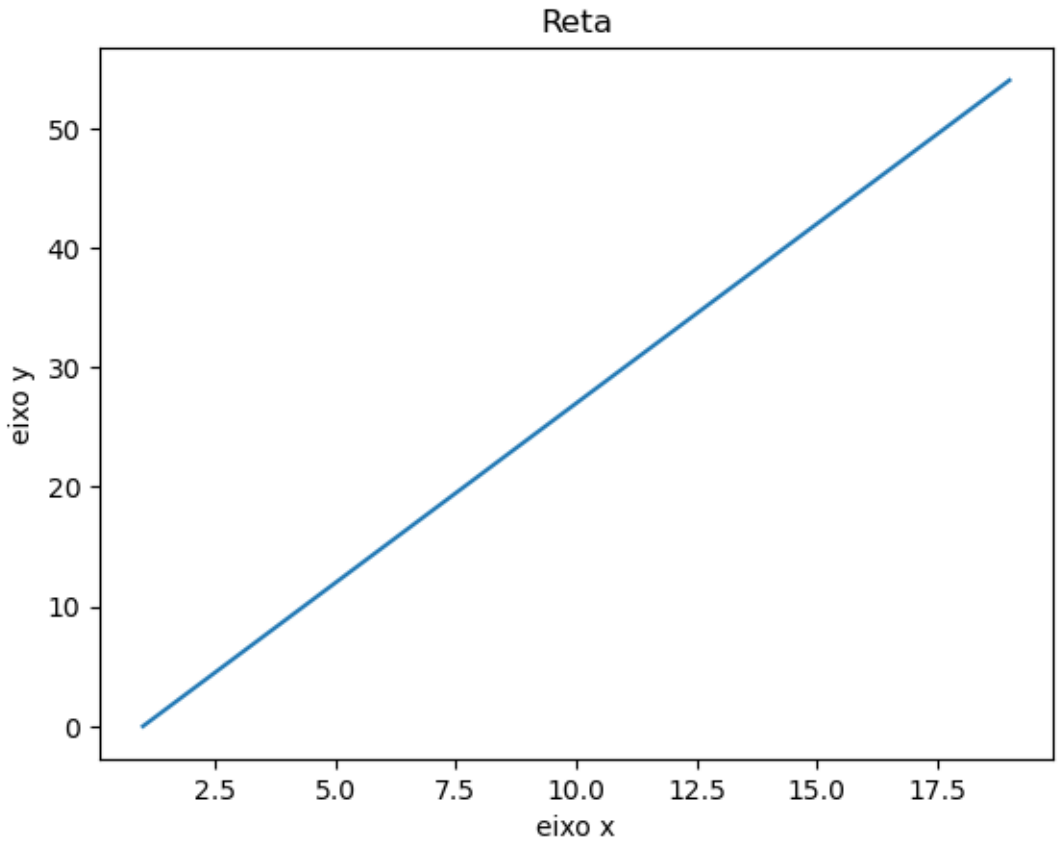
```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
```

[331]:
```
y = 3*x-3
print(y)
```

```
[ 0  3  6  9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54]
```

[335]:
```
fig.plot(x,y)
fig.title('Reta')
fig.xlabel('eixo x');fig.ylabel('eixo y')
```

[335]:
```
Text(0, 0.5, 'eixo y')
```



[337]:
```
import matplotlib.pyplot as fig
import numpy as ny
```

[339]:
```
x = ny.arange(1,30)
```

[343]:
```
y = 3*x-3
```

[9]:
```
import pandas as pd
```

[59]:
```
df = pd.read_excel('planalug.xlsx', sheet_name='plan1')
df
```

[59]:

	Unnamed: 0	Custo de Aluguel	Retorno
0	NaN	3000.0	1.31
1	NaN	2200.0	1.34
2	NaN	1800.0	1.42
3	NaN	800.0	1.40
4	NaN	680.0	1.42
5	NaN	1200.0	1.40
6	NaN	3200.0	1.47
7	NaN	2200.0	1.45
8	NaN	1500.0	1.48
9	NaN	1300.0	1.42
10	NaN	850.0	1.34
11	NaN	650.0	1.34
12	NaN	700.0	1.35
13	NaN	5300.0	1.35
14	NaN	NaN	1.36
15	NaN	NaN	1.32
16	NaN	NaN	1.24
17	NaN	NaN	1.22
18	NaN	NaN	1.27
19	NaN	NaN	1.26

[61]:
```
df.head()
```

[61]:

	Unnamed: 0	Custo de Aluguel	Retorno
0	NaN	3000.0	1.31
1	NaN	2200.0	1.34
2	NaN	1800.0	1.42
3	NaN	800.0	1.40
4	NaN	680.0	1.42

[28]:
```
import matplotlib.pyplot as fig
import numpy as ny
```

[34]:
```
t = ny.arange(0,2)
print(t)
```

```
[0 1]
```

[36]:
```
import matplotlib.pyplot as fig
import numpy as ny
```

[42]:
```
t = ny.arange(0,3)
x = 4.333333
```

```
print(0,3)
```

```
0 3
```

[55]:
```
#fig.plot(t,x)
```

[46]:
```
x = [10.0, -2.0, 5.0]
```

[49]:
```
import numpy as ny
```

[51]:
```
vetor = ny.array(x)
vetor
```

[51]:
```
array([10., -2.,  5.])
```

[53]:
```
print(vetor)
```

```
[10. -2.  5.]
```

[91]:
```
#   skiprows=1,
df = pd.read_excel('planalug.xlsx',
                   sheet_name='plan1',
                  usecols=['Retorno'])
df
```

[91]:

	Retorno
0	1.31
1	1.34
2	1.42
3	1.40
4	1.42
5	1.40
6	1.47
7	1.45
8	1.48
9	1.42
10	1.34
11	1.34
12	1.35
13	1.35
14	1.36
15	1.32
16	1.24
17	1.22
18	1.27
19	1.26

[69]:
```
#df = dropna()
```

[69]: 'plan1'

[89]:
```
#df = df.set_index('Retorno')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/tmp/ipykernel_458/1045192473.py in ?()
----> 1 df = df.set_index('Retorno')
      2 df

/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/pandas/core
 ↪frame.py in ?(self, keys, drop, append, inplace, verify_integrity)
   5866                     if not found:
   5867                         missing.append(col)
   5868
   5869         if missing:
-> 5870             raise KeyError(f"None of {missing} are in the columns")
   5871
   5872         if inplace:
   5873             frame = self

KeyError: "None of ['Retorno'] are in the columns"
```

[87]:
```
#df.to_excel('teste1.xlxs')
```

```
---------------------------------------------------------------------------
OptionError                               Traceback (most recent call last)
File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/pandas
 ↪io/excel/_base.py:1153, in ExcelWriter.__new__(cls, path, engine, date_format
 ↪datetime_format, mode, storage_options, if_sheet_exists, engine_kwargs)
   1152 try:
-> 1153     engine = config.get_option(f"io.excel.{ext}.writer", silent=True)
   1154     if engine == "auto":

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/pandas
 ↪_config/config.py:272, in CallableDynamicDoc.__call__(self, *args, **kwds)
    271 def __call__(self, *args, **kwds) -> T:
--> 272     return self.__func__(*args, **kwds)

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/pandas
 ↪_config/config.py:146, in _get_option(pat, silent)
    145 def _get_option(pat: str, silent: bool = False) -> Any:
--> 146     key = _get_single_key(pat, silent)
    148     # walk the nested dict
```

```
File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/pandas
 ↪_config/config.py:132, in _get_single_key(pat, silent)
    131         _warn_if_deprecated(pat)
--> 132     raise OptionError(f"No such keys(s): {repr(pat)}")
    133 if len(keys) > 1:

OptionError: No such keys(s): 'io.excel.xlxs.writer'

The above exception was the direct cause of the following exception:

ValueError                                Traceback (most recent call last)
Cell In[87], line 1
----> 1 df.to_excel('teste1.xlxs')

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/pandas
 ↪core/generic.py:2345, in NDFrame.to_excel(self, excel_writer, sheet_name,␣
 ↪na_rep, float_format, columns, header, index, index_label, startrow, startcol␣
 ↪engine, merge_cells, inf_rep, freeze_panes, storage_options, engine_kwargs)
   2332 from pandas.io.formats.excel import ExcelFormatter
   2334 formatter = ExcelFormatter(
   2335     df,
   2336     na_rep=na_rep,
   (…)
   2343     inf_rep=inf_rep,
   2344 )
-> 2345 formatter.write(
   2346     excel_writer,
   2347     sheet_name=sheet_name,
   2348     startrow=startrow,
   2349     startcol=startcol,
   2350     freeze_panes=freeze_panes,
   2351     engine=engine,
   2352     storage_options=storage_options,
   2353     engine_kwargs=engine_kwargs,
   2354 )

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/pandas
 ↪io/formats/excel.py:946, in ExcelFormatter.write(self, writer, sheet_name,␣
 ↪startrow, startcol, freeze_panes, engine, storage_options, engine_kwargs)
    942     need_save = False
    943 else:
    944     # error: Cannot instantiate abstract class 'ExcelWriter' with␣
 ↪abstract
    945     # attributes 'engine', 'save', 'supported_extensions' and␣
 ↪'write_cells'
--> 946     writer = ExcelWriter(  # type: ignore[abstract]
    947         writer,
    948         engine=engine,
```

```
    949             storage_options=storage_options,
    950             engine_kwargs=engine_kwargs,
    951         )
    952     need_save = True
    954 try:

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/pandas
 ↪io/excel/_base.py:1157, in ExcelWriter.__new__(cls, path, engine, date_format
 ↪datetime_format, mode, storage_options, if_sheet_exists, engine_kwargs)
   1155             engine = get_default_engine(ext, mode="writer")
   1156     except KeyError as err:
-> 1157         raise ValueError(f"No engine for filetype: '{ext}'") from err
   1159 # for mypy
   1160 assert engine is not None

ValueError: No engine for filetype: 'xlxs'
```

[93]:
```
df
```

[93]:

	Retorno
0	1.31
1	1.34
2	1.42
3	1.40
4	1.42
5	1.40
6	1.47
7	1.45
8	1.48
9	1.42
10	1.34
11	1.34
12	1.35
13	1.35
14	1.36
15	1.32
16	1.24
17	1.22
18	1.27
19	1.26

[107]:
```
x = df
x
```

[107]:

	Retorno
0	1.31
1	1.34

```
2      1.42
3      1.40
4      1.42
5      1.40
6      1.47
7      1.45
8      1.48
9      1.42
10     1.34
11     1.34
12     1.35
13     1.35
14     1.36
15     1.32
16     1.24
17     1.22
18     1.27
19     1.26
```

[109]:
```
import numpy as ny
vetor = ny.array(x)

print(vetor)
```

```
[[1.31]
 [1.34]
 [1.42]
 [1.4 ]
 [1.42]
 [1.4 ]
 [1.47]
 [1.45]
 [1.48]
 [1.42]
 [1.34]
 [1.34]
 [1.35]
 [1.35]
 [1.36]
 [1.32]
 [1.24]
 [1.22]
 [1.27]
 [1.26]]
```

[115]:
```
retorno = (vetor[1:20]-vetor[0:19])/vetor[0:19]
```

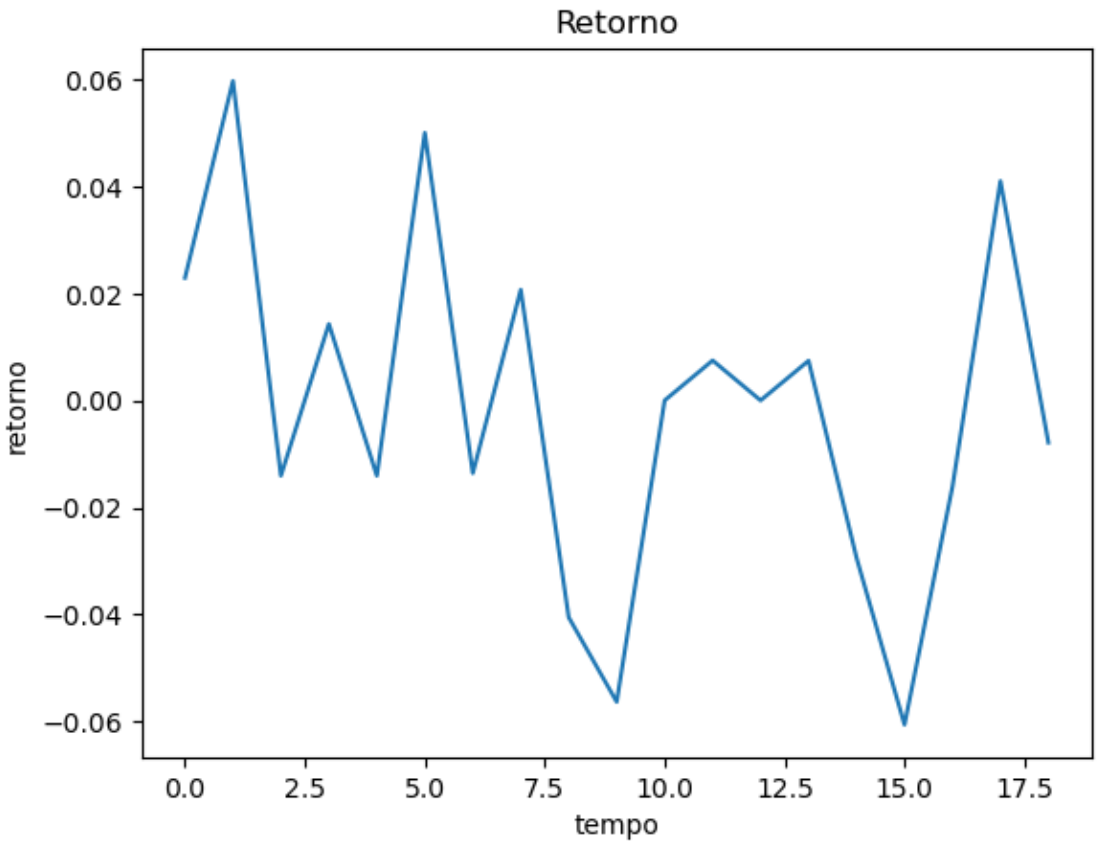
[117]:
```
print(retorno)
```

```
[[ 0.02290076]
 [ 0.05970149]
 [-0.01408451]
 [ 0.01428571]
 [-0.01408451]
 [ 0.05      ]
 [-0.01360544]
 [ 0.02068966]
 [-0.04054054]
 [-0.05633803]
 [ 0.        ]
 [ 0.00746269]
 [ 0.        ]
 [ 0.00740741]
 [-0.02941176]
 [-0.06060606]
 [-0.01612903]
 [ 0.04098361]
 [-0.00787402]]
```

[121]:
```
t = ny.arange(0,19)
```

[135]:
```
fig.plot(t, retorno)
fig.title('Retorno')
fig.xlabel('tempo')
fig.ylabel('retorno')
```

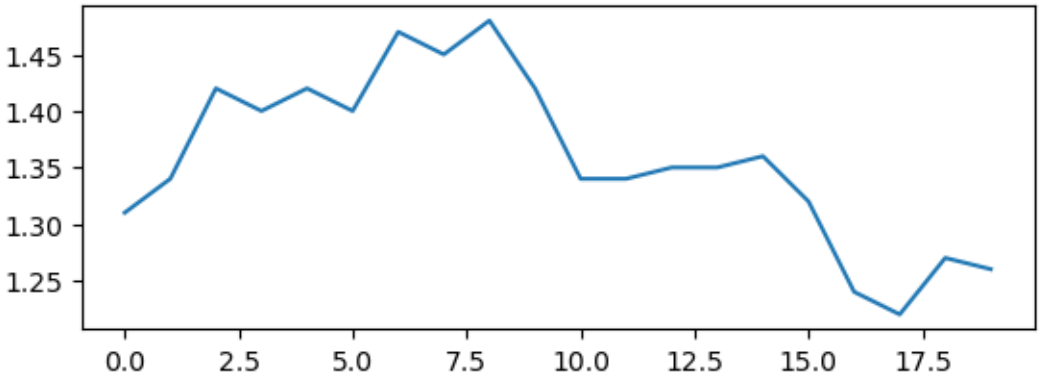
[135]:
```
Text(0, 0.5, 'retorno')
```



[140]:
```
tempo = ny.arange(0,20)
```

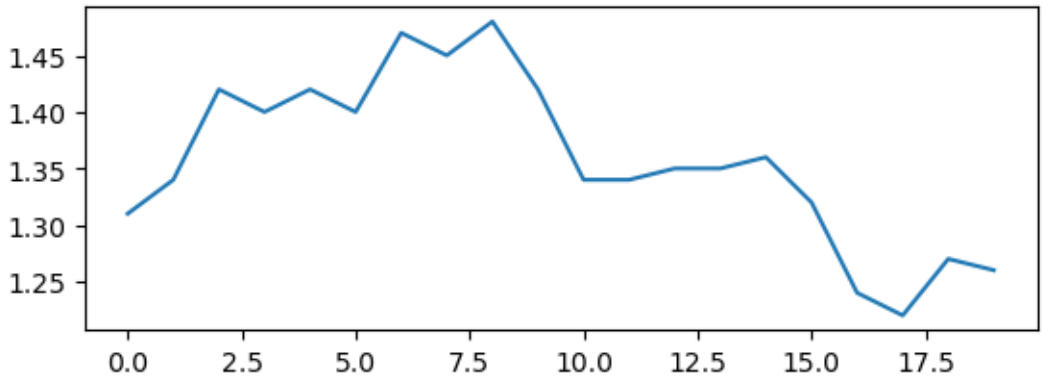
[142]:
```
fig.subplot(211); fig.plot(tempo, vetor)
```

[142]: [<matplotlib.lines.Line2D at 0x7069ddc8f8b0>]



[150]:
```
fig.subplot(212); fig.plot(tempo, vetor)
```

[150]:
```
[<matplotlib.lines.Line2D at 0x7069ddbc93c0>]
```



[7]:
```
import math

raio=float(input('raio = '))
area = math.pi*raio**2
print('area = ',area)
```

```
raio =  3.5
```

```
area =  38.48451000647496
```

[17]:
```
import math

x = float(input('x = '))
y = float(input('y = '))
z = math.sqrt(x**2+y**2)-math.log(x)+math.exp(y)

print('valor de z  = ', ret_medio)
```

```
x =  100
y =  90
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[17], line 7
      4 y = float(input('y = '))
      5 z = math.sqrt(x**2+y**2)-math.log(x)+math.exp(y)
----> 7 print('valor de z  = ', ret_medio)

NameError: name 'ret_medio' is not defined
```

[21]:
```
r1 = float(input('retorno 1 = '))
r2 = float(input('retorno 2 = '))
r3 = float(input('retorno 3 = '))

p1 = float(input('probabilidade 1 = '))
p2 = float(input('probabilidade 2 = '))
p3 = float(input('probabilidade 3 = '))

ret_medio = r1*p1 + r2*p2 + r3*p3

print('retorno medio = ', ret_medio)
```

```
retorno 1 =  100
retorno 2 =  90
retorno 3 =  120
probabilidade 1 =  0.2
probabilidade 2 =  0.7
probabilidade 3 =  0.1
```

```
retorno medio =  95.0
```

[25]:
```
p1 = float(input('resultado do primeiro dia = '))
p2 = float(input('resultado do segundo dia = '))
retorno = p2 - p1

if retorno >=0:
    print('lucro')
else:
    print('prejuizo')
```

```
resultado do primeiro dia =  10
resultado do segundo dia =  5
```

```
prejuizo
```

[29]:
```
p1 = float(input('resultado do pimeiro dia = '))
p2 = float(input('rersultado do segundo dia = '))
retorno = p2 - p1
if retorno >=0:
    print('lucro')
    print('fiquei feliz')
else:
    print('prejuizo')
('fiquei triste')
```

```
resultado do pimeiro dia =  10
rersultado do segundo dia =  20
```

```
lucro
```

```
fiquei feliz
```

[29]:
```
'fiquei triste'
```

[33]:
```
p1 = float(input('preco 1 = '))
if p1 <=18:
    print('barato')
elif (p1>18) and (p1<=25):
    print('adequado')
elif (p1>25) and (p1<=32):
    print('caro')
else:
    print('exremamente caro')
```

```
preco 1 =  40
exremamente caro
```

[39]:
```
perg1 = str(input('Temer sai?'))
if perg1=='s':
    perg2=str(input('Maia assume a presidência?'))
    if perg2 == 's':
            perg3 = str=(input('Antecipa a eleição'))
            if perg3 == 's':
                print('comprar Petrobras')
            else:
                print('comprar Usiminas')
    else:
            perg3=str(input('Antecipa a eleicao'))
            if perg3 == 's':
                print('Comprar Gerdau')
            else:
                print('Comprar Vale')
else:
    perg2=str(input('Greve de caminhoneiros?'))
    if perg2 == 's':
        print('Comprar dolar')
    else:
        print('Comprar ouro')
```

```
Temer sai? s
Maia assume a presidência? s
Antecipa a eleição n
comprar Usiminas
```

[47]:
```
n = int(input('qte de números = '))
cont = 1
s = 0
while cont<=n:
    s = s+cont
    cont = cont + 1
    print(s)
```

```
qte de números =  5
1
3
6
10
15
```

[53]:
```
n = int(input('qtde de numeros =  '))
cont = 1
s = 0
while cont<=n:
        s = s + cont
        cont+=1
print(s)
```

```
qtde de numeros =   6
21
```

[2]:
```
cont = 1
x=0
while x<20:
    x=cont*5
    cont=cont + 1
    print(x)
```

```
5
10
15
20
```

[ ]:
```
n=int(input('qte de numeros = '))
cont=
```

[ ]:
```
n=int(input('qte de numeros = '))
cont=1
s=0
while cont<=n:
    s=s+cont
```

```
    cont=cont+1
    print(s)
```

[ ]: