

# Exercício 001

---

## Exercício 001

---

A rede oficial de apostas na loteria é gerenciada pela Caixa Econômica Federal e a mesma disponibiliza todos os dados dos sorteios para consulta pública em seu site

[Site da Caixa - LotoFácil](#)

A proposta do exercício é coletar esses dados e levantar as informações da LotoFácil

- Qual o número mais sorteado e o menos sorteado?
- Qual a média de números Pares(p) Ímpares(I) e Primos(R) saem por sorteio?

## Nivelando a instalação:

Para instalar o Python e o seu instalador de pacotes, o PIP, o site da Python Brasil tem um tutorial super claro para cada sistema operacional

- [Linux](#)
- [MacOS](#)
- [Windows](#)

## Executando

---

### Primeira coisa, VirtualEnv

Primeiramente vamos criar e ativar um ambiente virtual para isolar nossa aplicação do resto do sistema e para isso instalamos o virtualenv:

```
pip install virtualenv
```

ou o venv:

```
sudo apt install python3-venv
```

agora iniciamos nosso desenvolvimento criando o nosso ambiente virtual

```
virtualenv .venv
```

para o Virtualenv e para o venv:

```
python -m venv .venv
```

Agora para ambos é o mesmo sistema de ativação:

```
source .venv/bin/activate
pip install ipython
```

## Coletando os dados

Instalamos o pacote requests, ele é responsável por acionar um endpoint e coletar seu retorno da mesma forma que um navegador web, para instalar pacotes abrimos um terminal e após ativado nosso virtual env usamos o comando `pip install nome_do_pacote` no nosso caso:

```
pip install requests
```

Assim iniciamos nosso código com o import do pacote requests, criamos uma variável **url** que recebe o link da página que vamos consultar e usando a função get to pacote capturamos o retorno do site

```
import requests
url =
'http://loterias.caixa.gov.br/wps/portal/loterias/landing/lotofacil/!ut/p/
a1/04_Sj9CPykssy0xPLMnMz0vMAfGjz0LNDH0MPAzCdbz8vTxNDRy9_Y2NQ13CDA0sTIEKIoE
KnN0dPUzMFqWmDEwsjAw8XZw8XMwtfQ0MPM2I02-AAzgaENIfRh-
FqsQ9wBmoxN_FydLAGAgNTKEK8DkRrACPGwpyQyMMMj0VAcySpRM!/dL5/d5/L2dBISEvZ0FBI
S9nQSEh/pw/Z7_HGK818G0K85260Q50IRSC42046/res/id=historicoHTML/c=cacheLevel
Page=//'
r = requests.get(url)
```

podemos visualizar o retorno primeiro exibindo a variável r:

```
In [4]: r
Out[4]: <Response [200]>
```

O valor *Response [200]* indica que o servidor retornou ok para nossa requisição, agora para obter o valor usamos o comando `r.text` e ele vai exibir todo o código HTML da página

## Importando uma tabela HTML para um DataFrame Pandas

Aqui vamos precisar de dois novos pacotes, o pandas e o lxml que instalamos com o pip:

```
pip install pandas
pip install lxml
```

Assim adicionamos as seguintes linhas a nosso código:

```
r_html = r.text
df = pd.read_html(r_html)
```

```

### Identificando o tipo de variável
type(df)
type(df[0])

### coletando o primeiro dado da lista [0]
df = df[0].copy()
df

```

Analisando nosso DataFrame

```

In [5]:df
Out[5]: [      Concurso Data Sorteio  Bola1  ...  Acumulado_15_Números
Estimativa_Premio  Valor_Acumulado_Especial
0          1  29/09/2003    18  ...          000
000          000
1          1  29/09/2003    18  ...          000
000          000
2          1  29/09/2003    18  ...          000
000          000
3          1  29/09/2003    18  ...          000
000          000
4          1  29/09/2003    18  ...          000
000          000
...          ...          ...          ...          ...
...          ...          ...
8533      2141  25/01/2021    23  ...          000
1.500.00000          39.478.81010
8534      2142  26/01/2021    22  ...          000
1.500.00000          39.795.02202
8535      2142  26/01/2021    22  ...          000
1.500.00000          39.795.02202
8536      2143  27/01/2021    11  ...          1.252.97626
3.500.00000          40.098.16145
8537      2144  28/01/2021    20  ...          000
1.500.00000          40.538.17875

[8538 rows x 33 columns]]

```

**Definindo constantes do jogo:**

```
nr_pop = list(range(1, 25))
nr_pares = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]
nr_impares = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25]
nr_primos = [2, 3, 5, 7, 11, 13, 17, 19, 23]
```

### Iniciando as variáveis de respostas:

```
comb = []
v_01 = 0
v_02 = 0
v_03 = 0
v_04 = 0
v_05 = 0
v_06 = 0
v_07 = 0
v_08 = 0
v_09 = 0
v_10 = 0
v_11 = 0
v_12 = 0
v_13 = 0
v_14 = 0
v_15 = 0
v_16 = 0
v_17 = 0
v_18 = 0
v_19 = 0
v_20 = 0
v_21 = 0
v_22 = 0
v_23 = 0
v_24 = 0
v_25 = 0
```

### Variáveis de interação:

```
lst_campos = ['Bola1', 'Bola2', 'Bola3', 'Bola4', 'Bola5',
              'Bola6', 'Bola7', 'Bola8', 'Bola9', 'Bola10', 'Bola11', 'Bola12',
              'Bola13', 'Bola14', 'Bola15']
```

### Loop

Precisamos percorrer nosso DF linha a linha, assim:

```
for index, row in df.iterrows():
```

em cada linha precisamos levantar os dados de frequencia dos grupos (p,i,np) e dos numeros sorteados(1...25)

assim primeira inicialisamos onde salvaremos os grupos:

```
for index, row in df.iterrows():  
    v_pares = 0  
    v_impares = 0  
    v_primos = 0
```

e agora em cada linha extraida (variável *row* do nosso for) testamos os campos:

```
for campo in lst_campos:  
    if row[campo] in nr_pares:  
        v_pares += 1  
    if row[campo] in nr_impares:  
        v_impares += 1  
    if row[campo] in nr_primos:  
        v_primos += 1  
    if row[campo] == 1:  
        v_01 += 1  
    if row[campo] == 2:  
        v_02 += 1  
    if row[campo] == 3:  
        v_03 += 1  
    if row[campo] == 4:  
        v_04 += 1  
    if row[campo] == 5:  
        v_05 += 1  
    if row[campo] == 6:  
        v_06 += 1  
    if row[campo] == 7:  
        v_07 += 1  
    if row[campo] == 8:  
        v_08 += 1  
    if row[campo] == 9:  
        v_09 += 1  
    if row[campo] == 10:  
        v_10 += 1  
    if row[campo] == 11:  
        v_11 += 1  
    if row[campo] == 12:
```

```

        v_12 += 1
    if row[campo] == 13:
        v_13 += 1
    if row[campo] == 14:
        v_14 += 1
    if row[campo] == 15:
        v_15 += 1
    if row[campo] == 16:
        v_16 += 1
    if row[campo] == 17:
        v_17 += 1
    if row[campo] == 18:
        v_18 += 1
    if row[campo] == 19:
        v_19 += 1
    if row[campo] == 20:
        v_20 += 1
    if row[campo] == 21:
        v_21 += 1
    if row[campo] == 22:
        v_22 += 1
    if row[campo] == 23:
        v_23 += 1
    if row[campo] == 24:
        v_24 += 1
    if row[campo] == 25:
        v_25 += 1

```

e então montamos a combinação da linha:

```

comb.append(str(v_pares) + 'p-' + str(v_impares) +
            'i-' + str(v_primos) + 'np')

```

Para facilitar, o loop completo:

```

for index, row in df.iterrows():
    v_pares = 0
    v_impares = 0
    v_primos = 0

    for campo in lst_campos:
        if row[campo] in nr_pares:
            v_pares += 1

```

```
if row[campo] in nr_impares:
    v_impares += 1
if row[campo] in nr_primos:
    v_primos += 1
if row[campo] == 1:
    v_01 += 1
if row[campo] == 2:
    v_02 += 1
if row[campo] == 3:
    v_03 += 1
if row[campo] == 4:
    v_04 += 1
if row[campo] == 5:
    v_05 += 1
if row[campo] == 6:
    v_06 += 1
if row[campo] == 7:
    v_07 += 1
if row[campo] == 8:
    v_08 += 1
if row[campo] == 9:
    v_09 += 1
if row[campo] == 10:
    v_10 += 1
if row[campo] == 11:
    v_11 += 1
if row[campo] == 12:
    v_12 += 1
if row[campo] == 13:
    v_13 += 1
if row[campo] == 14:
    v_14 += 1
if row[campo] == 15:
    v_15 += 1
if row[campo] == 16:
    v_16 += 1
if row[campo] == 17:
    v_17 += 1
if row[campo] == 18:
    v_18 += 1
if row[campo] == 19:
    v_19 += 1
if row[campo] == 20:
```

```

        v_20 += 1
    if row[campo] == 21:
        v_21 += 1
    if row[campo] == 22:
        v_22 += 1
    if row[campo] == 23:
        v_23 += 1
    if row[campo] == 24:
        v_24 += 1
    if row[campo] == 25:
        v_25 += 1
    comb.append(str(v_pares) + 'p-' + str(v_impares) +
                'i-'+str(v_primos)+'np')

```

Agora precisamos das frequencias dos numeros em uma lista unica:

```

freq_nr = [
    [1, v_01],
    [2, v_02],
    [3, v_03],
    [4, v_04],
    [5, v_05],
    [6, v_06],
    [7, v_07],
    [8, v_08],
    [9, v_09],
    [10, v_10],
    [11, v_11],
    [12, v_12],
    [13, v_13],
    [14, v_14],
    [15, v_15],
    [16, v_16],
    [17, v_17],
    [18, v_18],
    [19, v_19],
    [20, v_20],
    [21, v_21],
    [22, v_22],
    [23, v_23],
    [24, v_24],
    [25, v_25]
]

```

]



E finalmente podemos tratar e apresentar nossos resultados para os numeros mais frequentes:

```
freq_nr
freq_nr.sort(key=lambda tup: tup[1])
freq_nr[0] # primeiro
freq_nr[-1] # ultimo
```

e para as combinações de tipos de numeros:

```
counter = collections.Counter(comb)
resultado = pd.DataFrame(counter.items(), columns=['Combinacao',
'Frequencia'])
resultado['p_freq'] = resultado['Frequencia'] / \
    resultado['Frequencia'].sum()*100
resultado = resultado.sort_values(by='p_freq')
```

## O que temos de variáveis de entrada reais aqui?

- url de busca dos dados

sendo assim podemos transformar isso tudo em um script mais simples para executar:

Adicionamos:

```
import sys
```

Alteramos:

```
url = '...'
```

para:

```
url = sys.argv[1]
```

assim usamos via terminal o comando:

```
python main.py
'http://loterias.caixa.gov.br/wps/portal/loterias/landing/lotofacil/!ut/p/
a1/04_Sj9CPykssy0xPLMnMz0vMAfGjz0LNDH0MPAzCdbz8vTxNDRy9_Y2NQ13CDA0sTIEKIoE
KnN0dPUzMfQwMDEwsjAw8XZw8XMwtfQ0
...: MPM2I02-AAzgaENIfrh-
FqsQ9wBmoxN_FydLAGAgNTKEK8DkRrACPGwpyQyMMMj0VAcySprM!/dl5/d5/L2dBISEvZ0FBI
S9nQSEh/pw/Z7_HGK818G0K85260Q50IRSC42046/res/id=historicoHTML/c=cacheLevel
Page/=/'
```

e adicionamos nossa saída no final do arquivo:

```
print(
    '''
O número mais frequente é o: {}
O número menos frequente é o: {}
A combinação de Pares, Ímpares e Primos mais frequente é: {} com
frequencia de: {}%
    '''.format(freq_nr[-1][0], freq_nr[0][0],
resultado['Combinacao'].values[-1],
int(resultado['p_freq'].values[-1]*100)/100)
)
```