# PANDAS tutorials - https://youtu.be/CmorAWRsCAw?list=PLeo1K3hjS3uuASpe-1LjfG5f14Bnozjwy # Tutorial 02 - Dataframe Basics

In [1]:
```
import pandas as pd
# Importando o arquivo csv
# Caso o arquivo esteja na mesma pasta que o arquivo xxx.ipynb não é necessário descrever o caminho
# Observe que quando se cola o endereço do explorer é necessário inverter as barras
# As barras podem ser duplas como no comando abaixo similar ao Linux: pwd (present working directory)
# df = pd.read_csv("C:/Users/henri/Documents/Henrique/Trabalho/Python/Pandas/Tutoriais/sample_data_tutorial_02.csv")
df = pd.read_csv('C:\\Users\\henri\\Documents\\Henrique\\Trabalho\\Python\\Pandas\\Tutoriais\\sample_data_tutorial_02.csv')
```

In [2]:
```
pwd
```

Out[2]:
```
'C:\\Users\\henri\\Documents\\Henrique\\Trabalho\\Python\\Pandas\\Tutoriais'
```

In [3]:
```
df
```

Out[3]:

| | day | temperature | windspeed | event |
|---|---|---|---|---|
| **0** | 1/1/2017 | 32 | 6 | Rain |
| **1** | 1/2/2017 | 35 | 7 | Sunny |
| **2** | 1/3/2017 | 28 | 2 | Snow |
| **3** | 1/4/2017 | 24 | 7 | Snow |
| **4** | 1/5/2017 | 32 | 4 | Rain |
| **5** | 1/6/2017 | 32 | 2 | Sunny |

In [4]:
```
# O dataframe também poderia ser criado via dicionário
test_df = {
    'day': ['1','2','3'],
    'temp_C' : [30, 44, 52],
    'xwind' : [8, 3, 11]
}
df1 = pd.DataFrame(test_df)
df1
```

Out[4]:

| | day | temp_C | xwind |
|---|---|---|---|
| **0** | 1 | 30 | 8 |
| **1** | 2 | 44 | 3 |
| **2** | 3 | 52 | 11 |

In [5]:
```
# O dataframe também poderia ser criado via tuples
test1_df = [
    ('1','2','3'),
    (30, 44, 52),
    (8, 3, 11)
]
df2 = pd.DataFrame(test1_df, columns=['day','temp_C','xwind'])
df2
```

Out[5]:

| | day | temp_C | xwind |
|---|---|---|---|
| **0** | 1 | 2 | 3 |
| **1** | 30 | 44 | 52 |
| **2** | 8 | 3 | 11 |

In [6]:
```
df.shape
```

Out[6]: (6, 4)

In [7]:
```
rows, columns = df.shape
print (rows, columns)
```

```
6 4
```

In [8]:
```
df.head(2) # Imprime somente as 2 primeiras linhas incluindo o cabeçalho
```

Out[8]:

| | day | temperature | windspeed | event |
|---|---|---|---|---|
| **0** | 1/1/2017 | 32 | 6 | Rain |
| **1** | 1/2/2017 | 35 | 7 | Sunny |

In [9]:
```
df.tail(1) # Imprime somente a última linha incluindo o cabeçalho
```

Out[9]:

| | day | temperature | windspeed | event |
|---|---|---|---|---|
| **5** | 1/6/2017 | 32 | 2 | Sunny |

In [10]:
```
df[2:5] # Imprime da linha 2 a linha 4 (linha 5 não incluída)
# Para imprimir todo DataFrame poderia ser: df ou df[:]
```

Out[10]:

| | day | temperature | windspeed | event |
|---|---|---|---|---|
| **2** | 1/3/2017 | 28 | 2 | Snow |
| **3** | 1/4/2017 | 24 | 7 | Snow |
| **4** | 1/5/2017 | 32 | 4 | Rain |

In [11]:
```
df.columns # Imprime o cabeçalho somente
```

Out[11]: Index(['day', 'temperature', 'windspeed', 'event'], dtype='object')

In [12]:
```
df.temperature # Imprime a coluna 'temperature'. Este mesmo comando poderia ser: df
['temperature']
# O uso dos brackets é necessário quando há espaço nos nomes, ex. 'temp erature'
```

Out[12]:
```
0    32
1    35
2    28
3    24
4    32
5    32
Name: temperature, dtype: int64
```

In [13]:
```
type(df.event)
```

Out[13]:
```
pandas.core.series.Series
```

In [14]:
```
# Para visualizar apenas algumas colunas [[]]
df[['day','windspeed']]
```

Out[14]:

| | day | windspeed |
|---|---|---|
| **0** | 1/1/2017 | 6 |
| **1** | 1/2/2017 | 7 |
| **2** | 1/3/2017 | 2 |
| **3** | 1/4/2017 | 7 |
| **4** | 1/5/2017 | 4 |
| **5** | 1/6/2017 | 2 |

In [15]:
```
df.temperature.max() # outras funções: mean, min, std. Procurar na internet outras
 operações incluídas nos 'pandas'
```

Out[15]:
```
35
```

In [16]:
```
df.describe() # Imprime estatísticas das colunas com valores (float ou integers)
```

Out[16]:

| | temperature | windspeed |
|---|---|---|
| **count** | 6.000000 | 6.000000 |
| **mean** | 30.500000 | 4.666667 |
| **std** | 3.885872 | 2.338090 |
| **min** | 24.000000 | 2.000000 |
| **25%** | 29.000000 | 2.500000 |
| **50%** | 32.000000 | 5.000000 |
| **75%** | 32.000000 | 6.750000 |
| **max** | 35.000000 | 7.000000 |

In [17]:
```
df[df.temperature>=32] # Imprime df onde na coluna 'temperature' é maior ou igual 32
```

Out[17]:

| | day | temperature | windspeed | event |
|---|---|---|---|---|
| **0** | 1/1/2017 | 32 | 6 | Rain |
| **1** | 1/2/2017 | 35 | 7 | Sunny |
| **4** | 1/5/2017 | 32 | 4 | Rain |
| **5** | 1/6/2017 | 32 | 2 | Sunny |

In [18]:
```
df.windspeed[df.temperature>=32] # Imprime somente na coluna 'windspeed' os valores onde
# a coluna 'temperature' é maior ou igual 32
```

Out[18]:
```
0    6
1    7
4    4
5    2
Name: windspeed, dtype: int64
```

In [19]:
```
df[df.temperature==df['temperature'].max()] # Imprime todo df onde ocorre na coluna 'temperature' o valor máximo
```

Out[19]:

| | day | temperature | windspeed | event |
|---|---|---|---|---|
| **1** | 1/2/2017 | 35 | 7 | Sunny |

In [20]:
```
df['day'][df.temperature==df['temperature'].max()] # Imprime somente 'day'onde 'temperature' = valor máximo
# Para visualizar todas funções dos pandas google: "pandas series operations"
```

Out[20]:
```
1    1/2/2017
Name: day, dtype: object
```

In [21]:
```
# Imprime 'day' e 'temperature' onde 'temperature' = valor máximo
df[['day','temperature']][df.temperature==df['temperature'].max()]
```

Out[21]:

| | day | temperature |
|---|---|---|
| **1** | 1/2/2017 | 35 |

In [22]:
```
df.index # Indexes
```

Out[22]:
```
RangeIndex(start=0, stop=6, step=1)
```

In [23]:
```
# Sabemos que os 'pandas' colocam os indexes como 0, 1, 2 ...
# Caso se queira trocar o index usando a coluna 'day', fica:
df.set_index('day', inplace=True) # Caso não se coloque a opção 'inplace=True' o DataFrame não irá alterar-se!
```

In [24]:
```
df
```

Out[24]:

| | temperature | windspeed | event |
|---|---|---|---|
| **day** | | | |
| **1/1/2017** | 32 | 6 | Rain |
| **1/2/2017** | 35 | 7 | Sunny |
| **1/3/2017** | 28 | 2 | Snow |
| **1/4/2017** | 24 | 7 | Snow |
| **1/5/2017** | 32 | 4 | Rain |
| **1/6/2017** | 32 | 2 | Sunny |

In [25]:
```
df.loc['1/4/2017']
```

Out[25]:
```
temperature      24
windspeed         7
event          Snow
Name: 1/4/2017, dtype: object
```

In [26]:
```
# Para retornar o index original:
df.reset_index(inplace=True)
```

In [27]:
```
df
```

Out[27]:

| | day | temperature | windspeed | event |
|---|---|---|---|---|
| **0** | 1/1/2017 | 32 | 6 | Rain |
| **1** | 1/2/2017 | 35 | 7 | Sunny |
| **2** | 1/3/2017 | 28 | 2 | Snow |
| **3** | 1/4/2017 | 24 | 7 | Snow |
| **4** | 1/5/2017 | 32 | 4 | Rain |
| **5** | 1/6/2017 | 32 | 2 | Sunny |

In [28]:
```
df.set_index('event', inplace=True)
```

In [29]:
```
df # Observe que o 'event' contém linhas com mesmo valor
```

Out[29]:

| | day | temperature | windspeed |
|---|---|---|---|
| **event** | | | |
| **Rain** | 1/1/2017 | 32 | 6 |
| **Sunny** | 1/2/2017 | 35 | 7 |
| **Snow** | 1/3/2017 | 28 | 2 |
| **Snow** | 1/4/2017 | 24 | 7 |
| **Rain** | 1/5/2017 | 32 | 4 |
| **Sunny** | 1/6/2017 | 32 | 2 |

In [30]:
```
df.loc['Snow']
```

Out[30]:

| | day | temperature | windspeed |
|---|---|---|---|
| **event** | | | |
| **Snow** | 1/3/2017 | 28 | 2 |
| **Snow** | 1/4/2017 | 24 | 7 |