##

Métodos de carga de datos

Para cargar datos en python se requiere ser consciente de la ubicación, formato y nombre del archivo que se desea almacenar

1. Librerías

```
import pandas as pd, numpy as np, os
```

2. Carga de datos en formato csv

2.1. Función general

```
# Ubicación y nombre del archivo
path = r'C:\Users\DELL\OneDrive\Formación\Python\GITHUB - Repositorios\Datasets'
file = 'surveys.csv'

# Función de carga de datos
def custom_load(path,file):
    csv_path = os.path.join(path,file)
    return pd.read_csv(filepath_or_buffer = csv_path, # Ubicación del archivo
                       sep = ',', # Tipo de limitador entre columnas
                       skip_blank_lines = True, # Omitir filas en blanco
                       encoding = 'utf-8', # Codificación de caracteres
                       header = 'infer', # Inferir los títulos de columna, puede usarse un
                       index_col = None, # Columna a usar como índice del dataset. Ej. 're
                       na_filter = True, # Detecta strings vacíos como NaN (Default: True)
                       skiprows = None, # Omitir una fila del dataset. El conteno inicia e
                       dtype = {'month': np.float64} # Asigna un tipo de variable a las co
                      )


# Resultados
df = custom_load(path,file)

#print(df.shape)
#print(df.dtypes)
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35549 entries, 0 to 35548
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   record_id        35549 non-null  int64
 1   month            35549 non-null  float64
 2   day              35549 non-null  int64
 3   year             35549 non-null  int64
 4   plot_id          35549 non-null  int64
 5   species_id       34786 non-null  object
 6   sex              33038 non-null  object
 7   hindfoot_length  31438 non-null  float64
 8   weight           32283 non-null  float64
dtypes: float64(3), int64(4), object(2)
memory usage: 2.4+ MB
```

	record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
0	1	7.0	16	1977	2	NL	M	32.0	NaN
1	2	7.0	16	1977	3	NL	M	33.0	NaN
2	3	7.0	16	1977	2	DM	F	37.0	NaN
3	4	7.0	16	1977	7	DM	M	36.0	NaN
4	5	7.0	16	1977	3	DM	M	35.0	NaN

2.2. Cargar filas y columnas específicas

```python
# Ubicación y nombre del archivo
path = r'C:\Users\DELL\OneDrive\Formación\Python\GITHUB - Repositorios\Datasets'
file = 'surveys.csv'

# Función de carga de datos
def custom_load(path,file):
    csv_path = os.path.join(path,file)
    return pd.read_csv(filepath_or_buffer = csv_path, # Ubicación del archivo
                       sep = ',', # Tipo de limitador entre columnas
                       #usecols = [0,1,5] # Usar la 1ra, 2da y 6ta columna,
                       usecols = ['record_id','month','species_id'], # Usar la 1ra, 2da y
                       #nrows = 10, # Número de filas a cargar. Útil para evaluar framento
                       nrows = 20 # Se pude usar el parámetro skiprows y nrow para cargar
                      )
```

```
# Resultados
df = custom_load(path,file)

#print(df.shape)
#print(df.dtypes)
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   record_id   20 non-null     int64 
 1   month       20 non-null     int64 
 2   species_id  20 non-null     object
dtypes: int64(2), object(1)
memory usage: 608.0+ bytes
```

	record_id	month	species_id
0	1	7	NL
1	2	7	NL
2	3	7	DM
3	4	7	DM
4	5	7	DM

3. Carga de datos desde GITHUB

Para este ejemplo se usará una función lambda para aislar cierto tipo de columnas en el dataset.

Nótese que al final del link del archivo sebe agregarse el texto "?raw=true"

```
path = 'https://github.com/charliebrown007/Datasets/blob/main/'
file = 'Flinders_converted.csv'

f = lambda x: 'No' not in x and 'TOTAL' not in x
```

```
df = pd.read_csv(filepath_or_buffer = os.path.join(path,file + '?raw=true'),sep = ';', use
df.head()
```

	edad	sexo	v1	v2	v3	v4	v5	v6	v7	v8	...	v22	v23	v24	v25	v26	v27	v28	v29
0	21.0	1.0	2.0	2.0	4.0	2.0	1.0	2.0	3.0	2.0	...	2.0	3.0	4.0	3.0	3.0	3.0	2.0	3.0
1	23.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0	1.0	1.0	...	2.0	1.0	2.0	2.0	2.0	1.0	4.0	2.0
2	20.0	1.0	2.0	1.0	4.0	2.0	2.0	3.0	3.0	1.0	...	3.0	1.0	4.0	3.0	3.0	2.0	2.0	3.0
3	21.0	1.0	2.0	3.0	1.0	2.0	1.0	2.0	1.0	1.0	...	1.0	1.0	4.0	2.0	4.0	1.0	3.0	2.0
4	21.0	1.0	2.0	3.0	3.0	3.0	1.0	2.0	4.0	2.0	...	2.0	3.0	4.0	3.0	4.0	2.0	2.0	3.0

```
# Profundizando la función lambda
path = 'https://github.com/charliebrown007/Datasets/blob/main/'
file = 'Flinders_converted.csv'

df_cols = pd.read_csv(filepath_or_buffer = os.path.join(path,file + '?raw=true'),sep = ';'

f = lambda x : not ('v' in x or 'sexo' in x)
df_cols_2 = list(filter(f, df_cols))
df_cols_2
```

```
['No', 'edad', 'FDMQTOTAL']
```

4. Carga de datos desde una URL

5. Carga de archivos EXCEL (.xlsx)

6. Carga de datos desde One Drive

El URL de un archivo compartido en One Drive no puede usarse directamente porque nos envía a una página HTML para descargar el archivo posteriormente. Lo que se necesita es tranformar este link a uno de descarga directa para que pueda ser recnocido en Python mediante una API de One Drive.

```
import base64
def create_onedrive_directdownload (onedrive_link):
    data_bytes64 = base64.b64encode(bytes(onedrive_link, 'utf-8'))
    data_bytes64_String = data_bytes64.decode('utf-8').replace('/','_').replace('+','-').r
    resultUrl = f"https://api.onedrive.com/v1.0/shares/u!{data_bytes64_String}/root/conten
    return resultUrl
```

```
onedrive_link = 'https://1drv.ms/u/s!AneKqxx3Qjofh8BEzAxj441t-v4C4g'
onedrive_direct_link = create_onedrive_directdownload(onedrive_link)
df = pd.read_csv(filepath_or_buffer = onedrive_direct_link)
print(onedrive_link)
print(onedrive_direct_link)
df.head()
```

```
https://1drv.ms/u/s!AneKqxx3Qjofh8BEzAxj441t-v4C4g
https://api.onedrive.com/v1.0/shares/u!aHR0cHM6Ly8xZHJ2Lm1zL3UvcyFBbmVLcXh4M1Fqb2ZoOEJFekF4a
```

	Year	City	Sport	Discipline	NOC	Event	Event gender	Medal
0	1924	Chamonix	Skating	Figure skating	AUT	individual	M	Silver
1	1924	Chamonix	Skating	Figure skating	AUT	individual	W	Gold
2	1924	Chamonix	Skating	Figure skating	AUT	pairs	X	Gold
3	1924	Chamonix	Bobsleigh	Bobsleigh	BEL	four-man	M	Bronze
4	1924	Chamonix	Ice Hockey	Ice Hockey	CAN	ice hockey	M	Gold

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2311 entries, 0 to 2310
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Year          2311 non-null   int64 
 1   City          2311 non-null   object
 2   Sport         2311 non-null   object
 3   Discipline    2311 non-null   object
 4   NOC           2311 non-null   object
 5   Event         2311 non-null   object
 6   Event gender  2311 non-null   object
 7   Medal         2311 non-null   object
dtypes: int64(1), object(7)
memory usage: 144.6+ KB
```

7. Carga de datos provenientes de SPSS (.sav)