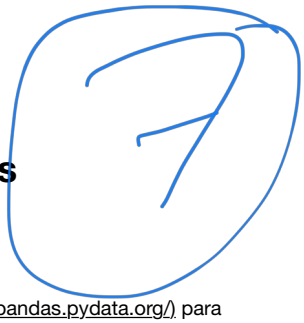

Reporte de Práctica 2: Lectura y manipulación de datos

Objetivo

En esta práctica, se cargan los datos en marcos de datos (inglés: data frame) con la librería pandas (https://pandas.pydata.org/) para poder iniciar su procesamiento.

Los objetivos de esta práctica son:

- Leer data frames de archivos
- Escribir data frames en CSV
- Agregar columnas en data frames
- Combinar dos o más data frames
- Filtrar renglones de un data frame En la práctica anterior decidí descargar un convertidor, ya que tengo archivos de excel y los convertí a .csv para tratamiento de datos, pero esa conversión me trajo muchos errores y tenía que guardar cada hoja individualmente, lo cual era muy tedioso. Después de investigar descubrí que la librería pandas también puede leer los archivos de excel, por lo que decidí volver a tomar los archivos originales para realizar esta prueba, esperando mejores resultados al tratar los archivos con pandas.

Lectura de datos

Para empezar a tratar los datos, use el orden por año, al utilizar los datos de '2015.xlsx', me lanzó este error ya que faltaban instalar librerías para el tratamiento de los archivos de excel:

In [ ]:
```
Traceback (most recent call last):
  File "/usr/local/lib/python3.6/dist-packages/pandas/io/excel.py", line 391, in __init__
    import xlrd
ModuleNotFoundError: No module named 'xlrd'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/lib/python3.6/dist-packages/pandas/util/_decorators.py", line 188, in wrapper
    return func(*args, **kwargs)
  File "/usr/local/lib/python3.6/dist-packages/pandas/util/_decorators.py", line 188, in wrapper
    return func(*args, **kwargs)
  File "/usr/local/lib/python3.6/dist-packages/pandas/io/excel.py", line 350, in read_excel
    io = ExcelFile(io, engine=engine)
  File "/usr/local/lib/python3.6/dist-packages/pandas/io/excel.py", line 653, in __init__
    self._reader = self._engines[engine](self._io)
  File "/usr/local/lib/python3.6/dist-packages/pandas/io/excel.py", line 393, in __init__
    raise ImportError(err_msg)
ImportError: Install xlrd >= 1.0.0 for Excel support
```

Así que usé el siguiente comando, para instalar la libreria que me recomendó pandas:

In [ ]:
```
$ sudo python3 -m pip install xlrd
```

Después de instalar todo correctamente usé un constructor específico para importar mis archivos a data frames.

In [ ]:
```
import pandas as pd
df2015 = pd.read_excel('2015.xlsx', index_col=None, header=0, sheet_name=0)
df2016 = pd.read_excel('2016.xlsx', index_col=0, header=0, sheet_name=[0,1,2,3,4])
df2017 = pd.read_excel('2017.xlsx', index_col=0, header=0, sheet_name=[0,1,2,3,4])
df2018 = pd.read_excel('2018.xlsx', index_col=None, header=0, sheet_name=0)
df2018mx = pd.read_excel('2018mx.xlsx', index_col=None, header=0, sheet_name=0)
```

Los parámetros que modifique en cada instancia son:

- index_col: indica cual de las columnas tiene un identificador para las filas, si no el archivo no tiene se ponen None.
- header: indica si los archivos tienen fila de encabezados, si no tiene se pone None.
- sheet_name: indica cuales son las hojas del documento que deseo importar, por defecto se agrega la hoja 0 pero si se quieren agregar mas se inserta un arreglo con el nombre o número de hoja.

El parámetro **sheet_name** me ayudó a importar todas las hojas de los excel de los años 2016 y 2017 ya que en estos años se capturó la información en diferentes hojas. Al hacer esto, el objeto receptor se volvió un diccionario de data frames, por lo que tuve que combinar los data frames, lo cual lo explicaré más adelante.

Escribir data frames en CSV.

Una vez cargados los datos en data frames, busqué la forma de exportarlos a CSV. Debido a que los datos tiene un campo llamado "Sinópsis" el cual contiene muchos de los caracteres especiales para hacer la separación de los valores decidí usar el separador "'\'" (slash invertido), se agregan 2 porque uno es el de fuga y el otro es el que se imprime.

Para exportar a CSV simplemente usamos el comando:

In [ ]:
```
df2015.to_csv("2015.csv", sep='\\', index=False)
df2016[0].to_csv("2016-0.csv", sep='\\', index=False)
df2016[1].to_csv("2016-1.csv", sep='\\', index=False)
df2016[2].to_csv("2016-2.csv", sep='\\', index=False)
df2016[3].to_csv("2016-3.csv", sep='\\', index=False)
df2016[4].to_csv("2016-4.csv", sep='\\', index=False)
df2017[0].to_csv("2017-0.csv", sep='\\', index=False)
df2017[1].to_csv("2017-1.csv", sep='\\', index=False)
df2017[2].to_csv("2017-2.csv", sep='\\', index=False)
df2017[3].to_csv("2017-3.csv", sep='\\', index=False)
df2017[4].to_csv("2017-4.csv", sep='\\', index=False)
df2018.to_csv("2018.csv", sep='\\', index=False)
df2018mx.to_csv("2018mx.csv", sep='\\', index=False)
```

Omití el index ya que no es información relevante para tratar nuestros datos. No hay una continuidad de los Id, ni una herencia de id de años anteriores.

Agregar columnas

Debido a que en la práctica anterior definimos los datos a tomar en cuenta:

- Año (Columna agregada)
- Categoría
- País
- Género
- ¿Cómo se enteró?
- Referencia celular

Es necesario agregar la columna Año a todos los data frames para identificar el año de procedencia al unir todos los datos. Además de esto, se tomaron solo columnas que tenían la información necesaria y se descartaron las demás.

Para completar la información de los data frames usé los siguientes comandos:

In [ ]:
```
df2015['Año']='2015'
df2016[0]['Año']='2016'
df2016[0]['Categoría']='Infantil'
df2016[1]['Año']='2016'
df2016[1]['Categoría']='Juvenil'
df2016[2]['Año']='2016'
df2016[2]['Categoría']='Aficionado'
df2016[3]['Año']='2016'
df2016[3]['Categoría']='Profesional'
df2016[4]['Año']='2016'
df2016[4]['Categoría']='SmarTIC'
df2017[0]['Año']='2017'
df2017[0]['Categoría']='Aficionado'
df2017[1]['Año']='2017'
df2017[1]['Categoría']='Juvenil'
df2017[2]['Año']='2017'
df2017[2]['Categoría']='Profesional'
df2017[3]['Año']='2017'
df2017[3]['Categoría']='SmarTIC'
df2017[4]['Año']='2017'
df2017[4]['Categoría']='Reto al guión'
df2018['Año']='2018'
df2018mx['Año']='2018'
```

Agregué las categorías a 2016 y 2017 ya que en esos años las categorías estaban separadas por hoja en la información original. Aproveché esta oportunidad para filtrar las columnas que necesitaba por data frame y sobreescribí los data frames, considerando que solo me iba a quedar con la información que necesitaba.

In [ ]:
```
df2015 = df2015[['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2016[0] = df2016[0][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2016[1] = df2016[1][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2016[2] = df2016[2][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2016[3] = df2016[3][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2016[4] = df2016[4][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2017[0] = df2017[0][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2017[1] = df2017[1][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2017[2] = df2017[2][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2017[3] = df2017[3][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2017[4] = df2017[4][['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2018 = df2018[['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
df2018mx = df2018mx[['Año','Categoría','País','Género','Como se enteró','Referencia Celular']]
```

Combinar 2 o más data frames

Dado que ya trate los datos y los filtré las columnas que solo quería llegó el momento de combinar los data frames más tediosos de toda la actividad, los años 2016 y 2017, cabe mencionar que hasta este punto fusioné los data frames porque además de estar resolviendo las actividades en el patrón sugerido, también hacía falta limpiar los datos, hasta este punto las columnas están ordenadas en el mismo orden y ya se agregaron y llenaron los campos que faltaban, como año y categoría.

Para combinar todos los data frames del año 2016 y 2017 con sus correspondientes años y además también combiné df2018 y 2018mx en df2018, usé los siguientes comandos:

In [ ]:
```
df2016 = pd.concat([df2016[0],df2016[1],df2016[2],df2016[3],df2016[4]])
df2017 = pd.concat([df2017[0],df2017[1],df2017[2],df2017[3],df2017[4]])
df2018 = pd.concat([df2018,df2018mx])
```

Por último se concatenaron todas los data frames en uno solo donde ya estarían ordenados.

In [ ]:
```
Concurso_cine = pd.concat([df2015,df2016,df2017,df2018])
Concurso_cine.to_csv('datosLimpiosCine.csv', sep='\\', index=False)
```

Filtrar renglones de data frame

Para filtrar renglones use pandas, la forma de hacerlo es muy sencilla, primero se introduce una función lógica entre corchetes al data frame y nos arroja los renglones que cumplen con el resultado verdadero de la función lógica. Por ejemplo, si queremos saber ¿Cuántos concursantes de Argentina participaron en 2016?, usamos el siguiente comando:

In [ ]:
```
df2016[df2016['Pais'] == 'Argentina']
```

In [ ]:
```
        Año    Categoría Cómo se enteró       Pais Referencia del celular
ID
45.0   2016     infantil            NaN  Argentina                    NaN
105.0  2016      juvenil          redes  Argentina                iPhone 5
108.0  2016  profesional       internet  Argentina                  iphone
```

Conclusiónes

A pesar de que ya había trabajado antes con los archivos en la práctica anterior, ahora me ahorre mucho tiempo trabajando con las múltiples hojas de los archivos 2016 y 2017, ya conocía pandas, lo utilicé como un reemplazo de R studio ya que estaba más familiarizado con Python que con R.

Por fín pude unir todos los datos en un solo archivo, en la práctica anterior lo veía imposible y muy tedioso, pero esta herramienta me ayudó a tratar los datos solo variando el índice de la hoja del archivo original.

Aún y cuando en la práctica 2 que explica la Dra. Elisa, yo no usé bash en esta ocasión debido a que no fue necesario limpieza extra o tratamiento fuera de pandas, ya que es una herramienta muy completa.

--11 de Febrero 2019-- Luis Angel Gutierrez Rodriguez 1484412 (tel:1484412)