ENGIN 604 Introducción a Python para las Finanzas
Análisis de Datos con Pandas

Profesor: *Gabriel E. Cabrera*
Ayudante: *Alex Den Braber*

La librería Pandas está pensada para análisis de datos del tipo tabular. No solo provee clases y funciones útiles, también permite aplicar funciones desde otras librerías como NumPy.

Existen dos tipos de estructuras fundamentales en Pandas:

Cuadro 1: Estructuras Fundamentales

Tipo de objeto	Descripción	Usado para
`DataFrame`	Objeto de 2-dimensiones con índice (`index`)	Datos tabulares organizados en columnas
`Series`	Objeto de 1-dimensión con índice (`index`)	Serie (de tiempo) de datos única

Para importar Pandas:

```
# se importa pandas
import pandas as pd
```

1. Series

Una Series es un *array* de una dimensión que contiene una secuencia de valores (como en NumPy) y una etiqueta (*label*) denominada índice (`index`). Para crear una Series:

```
# se crea Series
obj = pd.Series([4, 7, -5, 3],                 # se define los datos
                index = ['a', 'b', 'c', 'd'])  # se especifica el índice

# se verifica el obj
obj
```

```
## a    4
## b    7
## c   -5
## d    3
## dtype: int64
```

2. DataFrames

Un DataFrame es una estructura de 2-dimensiones con datos etiquetados (*labels*), índice en las filas como en las columnas. La columna potencialmente puede contener diferentes tipos de datos. Para crear un objeto `DataFrame`:

```
# se crea DataFrame
df = pd.DataFrame([10, 20, 30, 40],              # se define los datos
                  columns = ['numbers'],         # nombre de columna
                  index = ['a', 'b', 'c', 'd'])  # se especifica el índice

# se verifica df
df
```

```
##    numbers
## a       10
## b       20
## c       30
## d       40
```

Es importante notar que:

- Los datos están organizados en columna (puede tener nombres personalizados)
- Hay un índice que puede tomar diferentes formatos (e.g números, *strings*, etc).

Para acceder al índice:

```
# extrae columna
df.columns
```

```
## Index(['numbers'], dtype='object')
```

Se puede pasar una Series a un DataFrame:

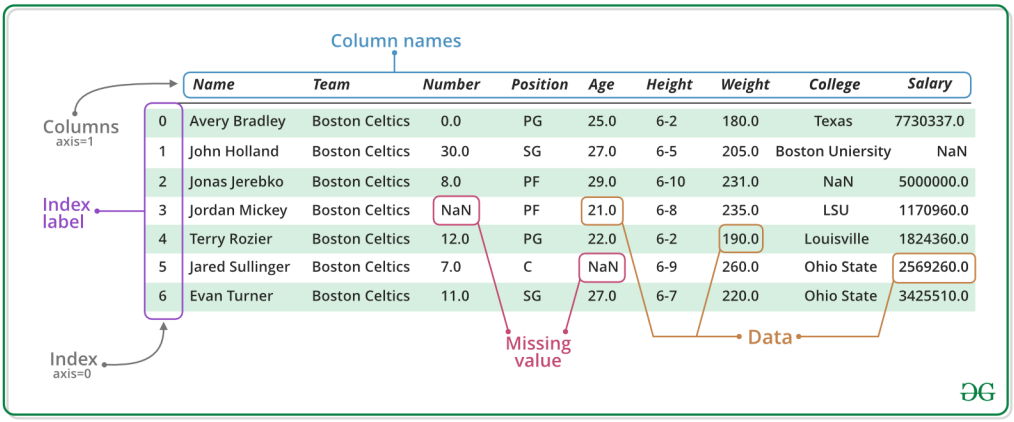
```
# serie a df
obj_to_df = obj.to_frame()

# se le asigna el nombre de la columna
obj_to_df.columns = ['numbers']

# verifica obj_to_df
obj_to_df
```

```
##    numbers
## a        4
## b        7
## c       -5
## d        3
```

En resumen la estructura de DataFrame:



Para ver la documentación de la librería Pandas ir a: https://pandas.pydata.org/docs/

3. Aplicación

1. El archivo `gapminder.xlsx` y `gapminder.dta` contiene un extracto del proyecto Gapminder sobre expectativa de vida (`lifeExp`), PIB per cápita (`gdpPercap`) y población (`pop`), según país (continente). Utilizando la librería Pandas, cargue a su espacio de trabajo ambas bases de datos. Nombre uno de los dos `DataFrame` como `gapminder`.
2. Muestre las 10 primeras y últimas observaciones de `gapminder`.
3. Genere un nuevo `DataFrame` que contenga solo los países del continente americano (`americas`) en el año 2007.
4. A partir del `DataFrame` generado en (3), muestre:
   - El país con mayor PIB per cápita en el año 2007
   - El país con menor PIB per cápita en el año 2007

   ¿Qué observa en el índice?
5. Reinicie el índice del `DataFrame` generado en (3). Luego elimine las variables `country` y `continent`.
6. Utilizando una *list comprehension*, renombre las columnas con su nombre original en minúscula.
7. Genere una variable que contenga la expectativa de vida (`lifeexp`) en meses. Realice definiendo una función y utilizando una función anónima.
8. Utilizando la base de datos original:
   a. Trabaje solo con los paises Europeos.
   b. Genere el crecimiento del PIB per cápita por país.
   c. Elimine los `NAs`.
   d. Construya una breve estadística descriptiva por país.
   e. Guarde las estadísticas descriptivas en un archivo con extensión .xlsx (excel).