



ENFIN761

## Business Intelligence para las Finanzas

### AYUDANTÍA 3

**Profesor:** *David Díaz S.*

**Ayudantes:** *Gabriel Cabrera G.*<sup>1</sup>

*28 agosto 2019*

### Concatenar y Merge

1. Construya los tres DataFrames que se encuentran en el apéndice.
2. Utilizando la función `pd.concat()`, concatene los DataFrames `df_a` y `df_b` por columna. Denomine a este nuevo objeto `df_nueva`.
3. Utilizando la función `pd.merge()`, fusione los DataFrames `df_nueva` y `df_n`.
4. Encuentre:
  - a. La intersección entre `df_a` y `df_b`.
  - b. La union entre `df_a` y `df_b`.
5. Fusione los DataFrames `df_a` y `df_b`:
  - a. Por la derecha.
  - b. Por la izquierda
6. Cambie el nombre de las columnas generadas en (5.b) usando sufijos.
7. Fusione los DataFrames `df_a` y `df_b`, a través de los índices (*index*).

### Trabajando con Datos Financieros

1. Utilizando la librería `pandas_datareader`, construya una función que le permita descargar (la fuente debe ser *Yahoo Finance*) los índices accionarios de Microsoft (**MSFT**), Nvidia (**NVDA**), AMD (**AMD**) y Apple (**AAPL**) con frecuencia, diaria desde el 1 de Enero del 2010 hasta hoy.
2. Seleccione el precio al cierre (*Close*) y luego construya una función para calcular los retornos de cada índice accionario. El retorno logarítmico se define como:

---

<sup>1</sup>✉: [gcabrera@fen.uchile.cl](mailto:gcabrera@fen.uchile.cl)

$$r_t = \log(1 + R_t) = \log\left(\frac{P_t}{P_{t-1}}\right) = p_t - p_{t-1}$$

3. Utilice la función `pd.pivot()` de manera que cada columna (4) contenga el retorno correspondiente a cada índice accionario. El nombre de las columnas deben corresponder al nombre del nemotécnico del índice.
4. Construya los retornos acumulados y luego gráfique utilizando la librería `matplotlib`. El color de cada serie como las “capas” del gráfico quedan a su elección.

## Complejidad computacional

1. Genere una función que entregue la secuencia de **Fibonacci**.

$$x_n = x_{n-1} + x_{n-2}$$

La secuencia es una sucesión infinita de números naturales 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,  $\infty$ . Por ejemplo, si su función la define como `fibonacci`, debería esperarse que `fibonacci(1) = 0`, `fibonacci(2) = 1`, `fibonacci(3) = 1` y así sucesivamente.

2. Utilizando la función generada en (1), demuestre que la complejidad computacional en términos de tiempo para la secuencia de Fibonacci es  $\mathcal{O}(2^n)$ .
3. Utilizando la función generada en (1), demuestre que la complejidad computacional en términos de espacio para cada  $n$  en una iteración es  $\mathcal{O}(1)$ .

## Apéndice

Table 1: DataFrame df\_a

id	nombre	apellido
1	Tom	Cruise
2	Will	Smith
3	Tom	Hanks
4	Jennifer	Aniston
5	Charlize	Theron

Table 2: DataFrame df\_b

id	nombre	apellido
4	Julia	Roberts
5	Nicole	Kidman
6	Emma	Watson
7	George	Clooney
8	Al	Pacino

Table 3: DataFrame df\_n

id	test_id
1	51
2	15
3	15
4	61
5	16
7	14
8	15
9	1
10	61
11	16