



ENGIN 604 INTRODUCCIÓN A PYTHON PARA LAS FINANZAS — OTOÑO 2021

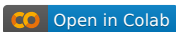
TAREA 2 - PAUTA

Entrega: 11:59pm, Sábado, Abril 3

Enviar a: engin604assignments@gmail.com

Límite máximo de páginas: 5 páginas

Ver políticas de tareas en <https://docenciaweb.fen.uchile.cl>



El archivo `returns_portfolios.pkl` contiene una lista que almacena el retorno mensual de dos portafolios ($j=1,2$) desde 2015-01-01 hasta 2020-12-01. A partir de este archivo responda:

1. Importe el archivo `returns_portfolios.pkl` como se muestra a continuación:

```
# importa la librería pandas
import pandas as pd

# lee pickle
returns_portfolios = pd.read_pickle('returns_portfolios.pkl')
```

2. Genere una función que permita calcular el promedio de una serie de datos almacenados en una lista.

$$r_j = \frac{\sum_{i=1}^n x_i}{n}$$

```
def media_aritmetica(x):
    """
    Parameters
    -----
    x : list
        Lista con los retornos.

    Returns
    -----
    Media aritmética

    """

    output = sum(x) / len(x)

    return(output)
```

```
port_a = returns_portfolios[0][1]
port_b = returns_portfolios[1][1]
```

¿Cual es el promedio de cada portafolio?

```
media_aritmetica(port_a)
```

```
## 0.02504936690460206
```

```
media_aritmetica(port_b)
```

```
## 0.02466073807150548
```

3. Genere una función que permita calcular la desviación estandar de una serie de datos almacenados en una lista.

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^n (x_i - r_j)^2}{n - 1}}$$

```
def desv_std(x):
    """
    Parameters
    -----
    x : list
        Lista con los retornos.

    Returns
    -----
    Desviación estandar.

    """

    desv_squared = 0

    mean_j = media_aritmetica(x)

    for row in x:
        desv_squared += (row - mean_j) ** 2

    output = (desv_squared / (len(x) - 1)) ** (1/2)

    return(output)
```

¿Cual es la desviación estandar de cada portafolio?

```
desv_std(port_a)
```

```
## 0.06566030506240997
```

```
desv_std(port_b)
```

```
## 0.06322762683884962
```

4. Genere una funcion que permita calcular la *Kurtosis* de una serie de datos almacenados en una lista.
La *Kurtosis* se define como:

$$\text{Kurtosis}_j = \frac{\frac{\sum_{i=1}^n (x_i - r_j)^4}{n}}{\left[\frac{\sum_{i=1}^n (x_i - r_j)^2}{n} \right]^2}$$

```
def kurtosis(x):
    """
    Parameters
    -----
    x : list
        Lista con los retornos.

    Returns
    -----
    Kurtosis.

    """

    # numerador
    desv4 = 0

    mean_j = media_aritmetica(x)

    for row in x:
        desv4 += (row - mean_j) ** 4

    above = desv4 / len(x)

    # denominador
    desv2 = 0

    for row1 in x:
        desv2 += (row1 - mean_j) ** 2

    below = desv2 / len(x)

    output = above / (below ** 2)

    return(output)
```

¿Cual es la *Kurtosis* de cada portafolio?

```
kurtosis(port_a)
```

```
## 3.3000545189533392
```

```
kurtosis(port_b)
```

```
## 2.612713317448421
```

5. Genere una funcion que permita calcular la *Skewness* de una serie de datos almacenados en una lista.
La *Skewness* se define como:

$$\text{Skewness}_j = \frac{\frac{\sum_{i=1}^n (x_i - r_j)^3}{n}}{\left[\frac{\sum_{i=1}^n (x_i - r_j)^2}{n} \right]^{1,5}}$$

```
def skewness(x):
    """
    Parameters
    -----
    x : list
        Lista con los retornos.

    Returns
    -----
    Skewness

    """

    # numerador
    desv3 = 0

    mean_j = media_aritmetica(x)

    for row in x:
        desv3 += (row - mean_j) ** 3

    above = desv3 / len(x)

    # denominador
    desv2 = 0

    for row1 in x:
        desv2 += (row1 - mean_j) ** 2

    below = desv2 / len(x)

    output = above / (below ** (1.5))

    return(output)
```

¿Cual es la *Skewness* de cada portafolio?

```
skewness(port_a)
```

```
## 0.10967498828025245
```

```
skewness(port_b)
```

```
## -0.12317820666332488
```

6. Un bono bullet es aquel en donde el emisor pagará al tenedor del bono cupones (pago de interés) correspondiente a cada periodo y al vencimiento (último periodo) recibirá el cupón más el principal (valor nominal del bono). La formula matemática sería:

$$P_B = \sum_{t=1}^{T-1} \frac{C}{(1+r)^t} + \frac{C + \text{Valor Nominal}}{(1+r)^T}$$

Donde P_B es el precio del bono, C el pago de intereses o cupones, T número de periodos y r la tasa de descuento. Considerando lo anterior construya una función que le permita calcular el precio de cualquier bono bullet. Para probar su función utilice un bono con pago de cupon anual, madurez de 25 años, tasa cupón 6,5 %, tasa de descuento 6,9 % y valor nominal de 100.

$$P_B = \frac{6,5}{(1+0,069)^1} + \dots + \frac{6,5}{(1+0,069)^{24}} + \frac{6.5 + 100}{(1+0,069)^{25}} = 95,29627$$

```
def precio_bono_bullet(vn, t, tc, r):
    """
    Parameters
    -----
    vn : float
        Valor nominal del bono bullet.
    t : float
        Número de periodos.
    tc : float
        Interes del cupón.
    r : float
        Tasa de descuento.

    Returns
    -----
    Precio de un bono bullet.

    """

    pb = 0

    for i in range(1,t+1):
        if i == t:
            pb += (vn * tc + vn) / (1 + r) ** i
        else:
            pb += (vn * tc) / (1 + r) ** i

    return(pb)

precio_bono_bullet(100, 25, 0.065, 0.069)

## 95.2962713625338
```