



Ejercicios:

1. Para un sistema matricial de la forma $Ax=b$, donde A es una matriz de coeficientes constantes de $N \times N$, x un vector de incógnitas de $N \times 1$ y b un vector constantes de $N \times 1$, realice una función en Python que devuelva el vector de solución x usando el método de Gauss-Jordan con *pivote parcial (por filas)*. La función debe devolver igualmente la matriz inversa A^{-1} . Pruebe la función con una matriz A aleatoria (`rand`) de 3×3 y un vector b aleatorio de 3×1 . Compare la solución encontrada con las funciones de la librería `numpy` del paquete *linalg*.

2. Para una regresión lineal de la forma $\hat{y}(x) = a_1 x + a_0$, demuestre que los coeficientes a_1 y a_0 están dados por:

$$a_0 = \bar{y} - a_1 \bar{x} \quad \text{y} \quad a_1 = \frac{\sum_{i=1}^N [(x_i - \bar{x})(y_i - \bar{y})]}{\sum_{i=1}^N (x_i - \bar{x})^2}, \text{ donde el numerador de } a_1 \text{ corresponde a la covarianza}$$

entre x y y , y el denominador a la varianza de x , N al número total de pares de puntos (x, y) y \bar{x} y \bar{y} a la media (promedio) de x y y respectivamente.

3. Realice una función en Python que a partir de un conjunto de datos (x, y) , devuelva los coeficientes de la regresión lineal que más se ajusta al conjunto de puntos usando el método de los mínimos cuadrados. Utilice la forma matricial de la regresión lineal para hallar los coeficientes, y use el método de Gauss-Jordan implementado en el punto 1 para hallar la solución. Para validar el método se proveen los archivos `Lab-Reg-X.bin` y `Lab-Reg-Y.bin` los cuales contienen una serie de puntos (x, y) , almacenados en formato binario de tipo 'double'. Halle los coeficientes de la regresión lineal de esta serie de puntos usando la función desarrollada y compare la solución con las funciones *polyfit* de `numpy`. Realice una gráfica que contenga los puntos (x, y) y la recta encontrada. Puede usar la función *polyval* de `numpy`.

4. Del libro King and Mody, 2010 (referencia 2 del programa), realizar el problema 2.15. Presentar los resultados y figuras respectivas en Python. Utilice la función desarrollada en el punto 3 y compare los resultados con las funciones nativas de `numpy` (*polyfit* y *polyval*).