

Guía de Laboratorio: Programación en Python

Procesamiento Digital de Señales

Paula Pérez, Alejandro Escobar, Cristian Ríos, Pablo Alvarado

2023-2

1. Introducción

En este laboratorio se pretende que el estudiante repase conceptos básicos de programación y profundice en las bases de programación en Python que se usarán durante el curso.

2. Ejercicios de Python

- Escriba un programa que encuentre el valor aproximado del número de euler en base a la suma infinita de la Ecuación 1. El usuario debe ingresar el número de elementos usados en la aproximación. Por ejemplo, si se ingresa 3 $e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} = 2,5$

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \dots \quad (1)$$

Nota: El formato de salida debe ser: e es aproximadamente: 2.5

- Implemente el anterior ejercicio en una función.
- Pida en un bucle *while* un nombre y una edad, almacénelos en una lista de duplas, hasta que el usuario diga no más. Ahora recorra esa lista mediante un ciclo *for*, imprimiendo nombre y edad, e indicando si es mayor de edad o no.
- Implemente el anterior ejercicio en varias funciones. Una debe retornar la dupla de nombre y edad. Otra debe recibir una edad y retornar un valor Booleano indicando si es mayor de edad o no.

3. Ejercicios con señales y numpy

- Cierta sistema necesita que usted calcule una métrica. Sea $\mathbf{C} \in \mathbb{R}^{n \times n}$ una matriz $n \times n$ de números reales. Genere una matriz aleatoria \mathbf{C} donde n es ingresado por el usuario (n es un número entero). Calcule la métrica de acuerdo a la Ecuación 2, donde $\text{tr}(\mathbf{C})$ es la traza de la matriz \mathbf{C} .

$$\text{métrica} = \frac{\text{tr}(\mathbf{C})}{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} c_{ij}}. \quad (2)$$

Nota: La traza de una matriz es la sumatoria de los elementos de su diagonal principal.

- El siguiente código genera una señal aleatoria de longitud variable (de acuerdo a su edad):

```
import numpy as np
edad = # AQUI SU EDAD

senal=np.empty((0,))
matriz=np.empty((0,100))
for i in range(edad):
    ventana=np.random.rand()*np.random.randn(100)+np.random.rand()
    senal=np.hstack((senal,ventana)) # Para concatenar en un vector
    matriz=np.vstack((matriz,ventana)) # Para concatenar en una ...
    matriz
```

Grafique la señal (en la variable *senal*). Nombre los ejes, utilice la cuadrícula. Recuerde utilizar el comando `%matplotlib inline` para que la gráfica se incruste en el notebook.

Ahora utilice un ciclo *for* para recorrer cada una de las filas de la matriz con las ventanas, y en cada iteración calcule e imprima la media y la desviación estándar. ¿Cómo se relacionan estos valores con la gráfica que generó anteriormente?

- Implemente en dos funciones el anterior ejemplo, la primera debe recibir su edad (o el número de iteraciones del ciclo) y devolver la señal y la matriz de ventanas. La segunda debe recibir estas dos variables, graficar la señal y devolver un vector con las medias y desviaciones estándar por ventana.