



RELATORÍA 3

Santacruz Zambrano, Solvey Isleny ^{1a}

¹ Estudiante del programa curricular de Ingeniería Química, Universidad Nacional de Colombia sede Manizales.

^a ssantacruz@unal.edu.co

20 de noviembre del 2022

1. Resumen

En el presente documento se encuentra la recopilación de las clases comprendidas entre el 07 de octubre y el 18 de octubre del 2022. Para esta relatoría, se tratarán los temas de módulos y paquetes integrados de Python; estructuras para analizar datos como Numpy y Pandas y manipulación de datos por medio de gráficas, integrales y derivadas.

2. Notas de clase

2.1. Módulos y paquetes: octubre 07 del 2022

Un módulo es un conjunto de documentos o fichero con extensión .py, el cual puede contener variables, funciones, objetos o sentencias que puedan ser ejecutables. Existe un documento principal llamado control que será el responsable de llamar a funciones externas que se haya creado en el módulo.

Existen módulos integrados de Python que son:

- random: sirve para generar numeros aleatorios, con sus respectivas distribuciones estadísticas.
- math: este módulo contiene funciones matemáticas comunes y constantes numéricas como pi, Euler, etc.
- tqdm: ayuda a que el programador estime el tiempo de ejecución de un programa.
- pandas: permite realizar hojas de cálculo fáciles de manipular semejantes a Excel y estructuras sql.
- sklearn: son funciones y herramientas útiles en machine learning.
- matplotlib: generador de distintos tipos de gráficas, basado en MATLAB.
- numpy: provee estructuras de datos útiles.
- os: se usa para realizar operaciones en el terminal desde Python.
- sys: organiza la ubicación de las herramientas, módulos, funciones, o paquetes de Python.

2.2. Estructuras para analizar datos: octubre 18 del 2022

En Python existes estructuras versátiles para el manejo y almacenamiento de datos como son:

- Numpy: conformado por arreglos (arrays) tipo vectores, matrices o tensores, los cuales son las estructuras más básicas.
- Pandas: conformado por series tipo vectores y matrices o dataFrames que son archivos que funcionan como las hojas de Excel.

2.3. Arreglos Numpy: noviembre 02 del 2022

Los arrays de numpy son estructuras que dan soporte a los iterables de python. Estos poseen mayor cantidad funcionalidades, están optimizadas y son de gran utilidad para el cálculo numérico.

Mediante los arreglos, es posible crear estructuras tipo vectores, matrices y tensores de manera eficiente

Numpy posee muchas herramientas, entre ellas tenemos:

Funciones integradas de numpy

- Creacion de arreglos:
 - np.array(iterable): crea un arreglo de datos
 - np.arange(inicio,fin,salto): crea un arreglo de números entre límites definidos.
 - np.ones(filas,columnas): crea un arreglo de unos.
 - np.zeros(filas, columnas): crea un arreglo de ceros.
- Redimensionamiento:
 - np.reshape(filas, columnas): reacondiciona el tamaño de un arreglo.
- Apilamiento:
 - np.hstack(elemento1,elemento2): apila datos de forma horizontal.
 - np.vstack(elemento1, elemento2): apila datos de forma vertical.
- Indexado y slicing:
 - vector[columna]
 - matriz[fila,columna]
 - tensor[profundidad,fila,columna]
- Almacenamiento:
 - np.savetxt("ruta", arreglo): guarda el arreglo en formato txt.
 - np.loadtxt("ruta")

Métodos en numpy

- Valor mínimo: sirve para calcular el valor mínimo de un arreglo de datos.
`arreglo.min()`
- Valor máximo: sirve para calcular el valor máximo de un arreglo de datos.
`arreglo.max()`
- Promedio: calcula el promedio de un arreglo de datos.
`arreglo.mean()`
- Desviación estándar: sirve para calcular la desviación estándar de un arreglo de datos.
`arreglo.std()`
- Suma del arreglo: sirve para calcular la suma de los valores de un arreglo de datos.
`arreglo.sum()`

Otras funciones matemáticas:

- Funciones trigonométricas: calcula las funciones trigonométricas.

`np.sin(arreglo)`

`np.cos(arreglo)`

- Funciones matemáticas:

`np.exp(arreglo)`: calcula el exponente de un número.

`np.log(arreglo)`: calcula el logaritmo de un número.

`np.gcd(arreglo)`: máximo común divisor.

`np.lcm(arreglo)`: mínimo común múltiplo.

`np.dot(arreglo1, arreglo2)`: producto punto.

`np.linalg.det(arreglo)`: determinante.

`np.linalg.inv(arreglo)`: inversa de una matriz.

`np.linalg.norm(arreglo)`: Norma.

`np.linalg.solve(a,b)`: sirve para resolver sistema de ecuaciones.

2.4. Series Pandas: noviembre 04 del 2022

Es una estructura unidimensional que se construye sobre un arreglo numpy. Está compuesto por índices numéricos (0,1,2,3...).

índices claves: "Enero", "Febrero"

Para indexar, se pueden usar índices numéricos o índices claves.

Las herramientas de una serie son:

- Atributos:
 - serie.index : retorna los índices clave de un arreglo de datos.
 - serie.values : retorna la data de un arreglo de datos.
 - serie.shape : retorna el tamaño del arreglo de datos.
- Métodos:
 - serie.describe(): devuelve otra serie describiendo la serie original.
 - serie.mean(): devuelve la media de un arreglo de datos.
 - serie.std(): devuelve la desviación estándar.
 - serie.min(): devuelve el valor mínimo de un arreglo de datos.
 - serie.max(): devuelve el valor máximo de un arreglo de datos.
 - serie.idxmin(): devuelve la clave del mínimo de un arreglo de datos.
 - serie.idxmax(): devuelve la clave del máximo de un arreglo de datos.
 - serie.value_counts(): devuelve las frecuencias de la serie.
- Indexado:

Las series se pueden indexar utilizando el índice numérico o el índice clave.

2.5. DataFrames: noviembre 11 del 2022

Son estructuras de datos en 2D formadas por filas y columnas, imitando a los archivos excel, sql, csv.

Los DataFrames funcionan como las hojas de Excel. Para indexar trabajan igual que las listas y diccionarios. Se puede usar el indexado por índice numérico o por índice clave.

Para crear un DataFrame:

```
import pandas as pd

hoja1 = pd.DataFrame(data = arreglo 2D numpy,
                     columns = ["columna1", ..., "columnaN"],
                     index = ["fila1", ..., "filaN" ])
```

2.6. Manipulación de datos: noviembre 11 del 2022

Para visualizar los datos de mejor manera se puede usar el módulo para gráficas matplotlib de la siguiente manera:

```
import matplotlib.pyplot as plt
plt.figure()
plt.plot(x, y, style) #donde x, y contienen la data,
plt.show()           #style: el estilo y color de linea
```

Por otro lado, en el análisis de datos, la derivada y la integral son valores importantes para ciertos propósitos, por lo tanto debe calcularse de manera numérica:

- Método numérico para calcular derivadas

Se hace uso de su significado geométrico, teniendo en cuenta que una derivada es una pendiente de una función.

$$f'(x) = \frac{y_2 - y_1}{x_2 - x_1}$$

- Método numérico para calcular integrales

Se resuelve utilizando el método del trapecio (calculando áreas) y haciendo la correspondiente sumatoria.