

Práctica de Programación en Python

Fecha de entrega: 6 de diciembre

Como práctica de evaluación del módulo de Programación en Python, se propone la realización de cuatro ejercicios, cada uno de ellos centrado en una parte del material teórico cubierto en clase: Python básico, NumPy y Pandas.

Para cada uno de los ejercicios se entregará un Notebook de Jupyter independiente en el que, además del código que resuelva cada problema, se deberá explicar el funcionamiento, las soluciones/decisiones propuestas (bien con comentarios bien con celdas *markdown*) y aportar ejemplos que demuestren su correcto funcionamiento.

Puedes usar cualquier recurso a tu alcance como apoyo para la resolución de los ejercicios (Google, StackOverflow, libros, etc.). Eso sí, recuerda hacer mención a tus fuentes en los comentarios sobre la resolución.

Se deberá entregar un fichero zip con tu nombre y apellidos (***miguel_angel_corella_python.zip***), en el que estén todos los archivos necesarios. No es necesario incluir el set de datos de los ejercicios de Pandas.

Python básico – Ejercicios cortos (2 puntos)

En línea con los ejercicios de Python básico realizados en clase, se deberá dar solución a los siguientes problemas:

1. Escribe un **programa** que solicite al usuario una lista de números positivos y/o negativos y, utilizando una comprensión de lista, genere otra con los números de la lista original que cumplen (A LA VEZ):
 - a. Son positivos.
 - b. Son divisibles entre 7.
 - c. No son múltiplos de 3.
2. Escribe una **función** llamada “comprimir” que reciba como parámetro una lista de elementos y devuelva como salida una lista de tuplas de la forma (elemento, número de repeticiones consecutivas). Por ejemplo:

```
comprimir([1, 1, 1, 2, 2, 3, 4, 4, 1, 1, 1]) = [(1,3), (2, 2), (3, 1), (4, 2), (1, 3)]
```

IMPORTANTE: La lista no tiene por qué contener sólo números.

3. Crea una **función** llamada “dame_personas” que reciba como parámetros una cadena de caracteres con el nombre de una ciudad y un diccionario de la forma {<persona1>: <ciudad1>, <persona2>:<ciudad2>, ...} y devuelva una lista de personas que pertenecen a esa ciudad ordenado alfabéticamente. Por ejemplo:

```
dame_personas('Alicante', {'Pedro': 'Alicante', 'Juan': 'Alicante'}) = ['Juan', 'Pedro']
```

4. Escribe una **función** “dame_palabras_veces” que reciba la ruta de un fichero de texto y un número entero y devuelva el listado de palabras del fichero cuya

frecuencia en el mismo es igual al número entero dado. Asume que las palabras del fichero siempre están separadas por espacios y haz que se ignore la diferencia entre mayúsculas y minúsculas. Por ejemplo:

Fichero <dummy.txt>

Esto es una prueba Prueba para validar validar, que funciona funciona.

```
dame_palabra_veces('dummy.txt', 2) = ['prueba', 'validar', 'funciona']
```

IMPORTANTE: Se debe incluir control de errores suficiente para chequear que el fichero indicado existe.

NumPy – Resolver Sudokus (2 puntos)

Escribe una función que reciba como parámetro el nombre de un fichero que contendrá un Sudoku completo (en forma de matriz), y devuelva si está bien solucionado o no.

A lo largo de la función, deberás ir comprobando los distintos requisitos que debe cumplir la solución de un Sudoku, pero además, también debes hacer comprobaciones más básicas, como que los números de cada celda están en el rango y tipo correcto o que la dimensión de la matriz es la adecuada para un Sudoku.

La función debe devolver un mensaje diciendo que la solución es correcta o, en caso contrario, el listado completo de motivos por el que la solución es incorrecta.

Puedes visitar esta página para repasar las reglas del Sudoku: <http://www.sudokumania.com.ar/metodos/reglas-del-sudoku>

Pandas - Explorando información de AirBnB (3 puntos)

Para evaluar los conocimientos adquiridos sobre la librería de análisis de datos Pandas, vamos a realizar un pequeño ejercicio de análisis exploratorio sobre un set de datos con información sobre AirBnB.

Más concretamente, junto con la práctica se entrega un fichero airbnb.csv que contiene información sobre alojamientos disponibles en AirBnB. Estos datos incluyen, entre otras cosas, información sobre ubicación, precios, reviews, ocupación, etc.

Con este set de datos, deberás preparar un Notebook donde se resuelvan las siguientes cuestiones.

1. Carga del set de datos en una variable llamada "airbnb".
2. Número de observaciones incluidas en el set de datos.
3. Listado de variables (y sus tipos) incluidas en el set de datos.
4. Estadísticos básicos **sólo de las variables numéricas** del set de datos.
5. Tablas de frecuencia **sólo de las variables categóricas** del set de dato.
6. Identificación de variables con *missing values* y proporción de los mismos.

Pandas - Analizando información de AirBnB (3 puntos)

Tras la exploración básica del set de datos del ejercicio anterior, debería ser sencillo obtener *insights* o conclusiones sobre los costes de los seguros informados.

El objetivo de este ejercicio, será utilizar las diferentes funcionalidades de **agrupación** y **agregación** que pone a nuestra disposición el módulo **pandas** de Python para responder a un pequeño conjunto de preguntas sobre las ofertas disponibles.

IMPORTANTE: los ejercicios se deberán utilizar las funcionalidades de **agrupación y agregación de Pandas** y **NO tablas pivote**.

El listado de preguntas que se deberá responder es el siguiente:

1. ¿Hay más ofertas de "Private room" o de "Entire home" en el dataset? ¿Cuántos más en porcentaje?
2. ¿Dónde está localizado el alojamiento más barato? Incluye en la respuesta localización, área y coordenadas.
3. ¿Cuál es el precio medio de las ofertas de cada área?
4. ¿Qué correlación hay entre el precio y el número de reviews?
5. ¿Cuál es el número total de ofertas por tipo y ubicación (location)?
6. ¿Cuáles son los 15 hosts con mayor número de ofertas disponibles? Incluye en el resultado únicamente el nombre del host y el número de ofertas disponibles.