



FACULTAD DE CIENCIAS EXACTAS Y NATURALES
Física y Astronomía

Tercer examen de fundamentación en computación

NOMBRE:

CÉDULA:

NOTA: *Entregue un solo notebook de Python con la solución de todos los puntos. La entrega se debe hacer a través de la asignación que se encuentra en el Classroom. Fecha de entrega: 11 de junio de 2023 a las 11:59 pm.*

1. (33 %) El fichero exoplanets contiene la base de datos de Exoplanetas, sacada de NASA Exoplanet Archive, la cual contiene más de 3900 exoplanetas confirmados con sus respectivos parámetros planetarios y estelares.
 - (a) Crear un DataFrame a partir del fichero anterior.
 - (b) Eliminar las filas con valores desconocidos y mostrar el número de filas del dataframe resultante.
 - (c) Hacer una gráfica de masa vs periodo, el color de los puntos debe representar el método de detección.
 - (d) Determine que porcentaje de descubrimientos se han hecho por cada uno de los métodos de detección. Adicionalmente, realice un gráfico que mejor muestre esta información.
 - (e) ¿Cuántos exoplanetas se han descubierto por año? Realiza un histograma que muestre cuántos exoplanetas se han descubierto por año.
 - (f) Conocer los planetas más extremos descubiertos. Mostrar la información de los exoplanetas que cumplan con las siguientes características:
 - El más lejano a la Tierra.
 - El más cercano a la Tierra.
 - El más grande.
 - El más pequeño.
 - (g) Los 5 exoplanetas más parecidos a la Tierra. Debe proponer un método para determinar cuáles son los 5 exoplanetas descubiertos más parecidos a la Tierra, y sus características. Justifique su método.
 - (h) Correlación entre propiedades de los exoplanetas. Realice una tabla de correlación y determine que parámetros se correlacionan con los otros y de qué manera lo hacen. Realice un análisis y determine a que se pueden deber estar correlaciones.
2. (33 %) El fichero coches.csv contiene información sobre los modelos de coches vendidos en USA un determinado año. Se pide:
 - (a) Crear un DataFrame a partir del fichero anterior.
 - (b) Eliminar las filas con valores desconocidos y mostrar el número de filas del dataframe resultante.
 - (c) Crear una columna con el precio en euros (cambio 1\$ = 0.94)
 - (d) Mostrar por pantalla las 10 últimas filas del DataFrame.
 - (e) Mostrar por pantalla el número de marcas que contiene el DataFrame.
 - (f) Mostrar por pantalla el número de modelos de cada marca que hay en el DataFrame, de mayor a menor frecuencia.

- (g) Mostrar por pantalla la marca y el modelo del coche más caro.
 - (h) Mostrar por pantalla el precio medio en euros de los coches agrupando por marca y ordenando de menor a mayor precio.
 - (i) Dibujar el diagrama de barras del porcentaje de modelos de cada marca.
 - (j) Dibujar el diagrama de dispersión de la potencia y el precio.
 - (k) Los gráficos deben guardarse en una carpeta con el nombre gráficos y deben tener un título adecuado.
3. (34 %) Los ficheros emisiones-2016.csv, emisiones-2017.csv, emisiones-2018.csv y emisiones-2019.csv, contienen datos sobre las emisiones contaminantes en la ciudad de Madrid en los años 2016, 2017, 2018 y 2019 respectivamente. Escribir un programa con los siguientes requisitos:
- (a) Generar un DataFrame con los datos de los cuatro ficheros.
 - (b) Filtrar las columnas del DataFrame para quedarse con las columnas ESTACION, MAGNITUD, AÑO, MES y las correspondientes a los días D01, D02, etc.
 - (c) Reestructurar el DataFrame para que los valores de los contaminantes de las columnas de los días aparezcan en una única columna.
 - (d) Añadir una columna con la fecha a partir de la concatenación del año, el mes y el día (usar el módulo datetime).
 - (e) Eliminar las filas con fechas no válidas (utilizar la función isnat del módulo numpy) y ordenar el DataFrame por estaciones contaminantes y fecha.
 - (f) Mostrar por pantalla las estaciones y los contaminantes disponibles en el DataFrame.
 - (g) Crear una función que reciba una estación, un contaminante y un rango de fechas y devuelva una serie con las emisiones del contaminante dado en la estación y rango de fechas dado.
 - (h) Mostrar un resumen descriptivo (mínimo, máximo, media, etc.) para cada contaminante.
 - (i) Mostrar un resumen descriptivo para cada contaminante por distritos.
 - (j) Crear una función que reciba una estación y un contaminante y devuelva un resumen descriptivo de las emisiones del contaminante indicado en la estación indicada.
 - (k) Crear una función que devuelva las emisiones medias mensuales de un contaminante y un año dados para todos las estaciones.
 - (l) Crear un función que reciba una estación de medición y devuelva un DataFrame con las medias mensuales de los distintos tipos de contaminantes.