

Programación de Sistemas Aeroportuarios

Práctica 9. Manipulación y Análisis de Datos de Vuelos

Objetivos: En esta práctica tendrás que desarrollar en Python un programa para analizar datos de vuelos.

Conocimientos teóricos previos: Conocimientos básicos de Python.

Tiempo de la práctica: El tiempo estimado para la realización de esta práctica es de 10 horas.

Fecha de entrega de la práctica: Esta práctica SÍ requiere ser entregada. La fecha límite para su entrega es el 8 de enero a las 23:59.

Introducción

- Implementa la aplicación en Python usando varios ficheros.
- Utiliza funciones correctamente para descomponer la aplicación y reutilizar código.
- Utiliza correctamente los tipos de datos vistos en clase (listas, tuplas, diccionarios)
- Controla todas las posibles excepciones de tu código.

La entrega se realizará en el GitLab de la Escuela, creando un proyecto que se llame `psa23-p2-adsb`.

Ejercicio 1. Descargar, descomprimir e introducir en base de datos

Debes leer el fichero CSV, que encontrarás en la siguiente dirección: <https://gsyc.urjc.es/~grex/psa/index.html>

Para ello deberás hacer web scraping con BeautifulSoup, y buscar por el atributo `href` del penúltimo enlace (etiqueta `a`) que contiene la página web `index.html`.

El fichero CSV, aún comprimido, tiene un tamaño considerable, por lo que desde tu código Python tendrás que comprobar primero si ya lo descargaste anteriormente (si existe el fichero). Sólo se descargará si no se encuentra en la ruta especificada. El fichero se encuentra comprimido, por lo que tendrás que descomprimirlo.

El fichero contiene información de los mensajes ADS-B con geo-localización de los aviones detectados desde el sensor en un día dado.

Convierte los campos de fecha correctamente para realizar futuras consultas, en particular, que sea string en mongoDB y que Pandas lo transforme en datetime.

Guarda correctamente las fechas y horas para poder hacer consultas. (Utiliza solo la fecha de creación del mensaje y crear un campo para introducir fecha y hora).

Ejercicio 2. Análisis

Utiliza pandas para parsear y visualizar la base de datos

Analiza las columnas, comprende su significado (utiliza las diapositivas del tema de ADS-B). Presta atención a las unidades. Si alguno de los campos prefieres utilizar otras unidades, ahora es el momento de realizar conversión.

Decide en este momento, si prefieres tener una columna en el DataFrame para la hora y otra para el tiempo, o por el contrario todo junto.

Genera los siguientes análisis de datos (imprimiendo los resultados por pantalla) y figuras asociadas:

1. Números de mensajes totales
2. Número de aviones distintos vistos
3. Número de mensajes recibidos por el sensor cada hora
4. Número de aviones distintos vistos por el sensor cada hora
5. Seleccionando un avión en concreto (el avión en concreto ha de fijarse como una constante de nombre FLIGHT al principio del fichero Python), calcula y muestra (en subgráficas):
 - Mensajes totales.
 - Número de mensajes cada 5 minutos.
 - Cómo varía la altitud.
6. Obtén los 10 aviones de los que más mensajes recibe el sensor
7. Obtén los 10 aviones que durante más tiempo están en el rango del sensor. Nótese que para identificar estos aviones, los mensajes sucesivos enviados por los aviones no han de haberse enviado en un intervalo mayor de 30 segundos.
8. Muestra la distancia máxima/mínima/media de los mensajes que recibe el sensor. Para ello calcula la distancia (geodésica) entre las coordenadas de los mensajes recibidos y las coordenadas del sensor de la URJC (40.283205, -3.821476). Consulta <https://geopy.readthedocs.io/> para calcular las distancias (no hace falta tener en cuenta la altitud).

Ejercicio 3. MongoDB

Lee el fichero CSV e importa todos los datos a una base de datos MongoDB. Crea una colección donde vayas introduciendo los datos del CSV para todos los vuelos que tengan “AA” en el código. Haz una captura de pantalla de Atlas donde se vea el número total de registro. Esa captura de pantalla la has de subir al repositorio con el nombre `captura-atlas.png`.