

**AviClima**

Análisis y predicción de la distribución de aves en función del clima

Índice

1. Descripción del proyecto 2

2. Recuperación de datos 3

3. Tratamiento de datos 5

4. Análisis de datos 7

5. Almacenamiento en Hive 11

Descripción del proyecto

En el proyecto se analizan los datos de avistamientos de aves en Euskadi y se comparan con los datos climáticos para buscar las posibles relaciones entre ellos.

Para ello se han utilizado datos de las siguientes fuentes:

Laboratorio Cornell de ornitología ([ebird.org](http://www.ebird.org/))

Agencia Estatal de Meteorología ([aemet.es](https://www.aemet.es/))

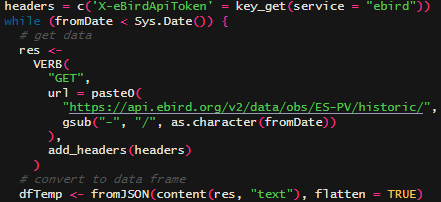
El proyecto se divide en tres apartados: la recuperación de los datos, el tratamiento y el análisis.

Recuperación de datos

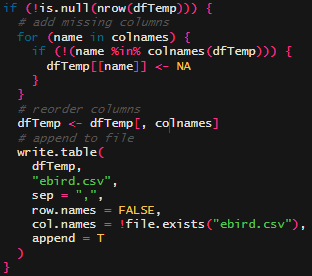
En este paso se han buscado datos en las APIs de ambas fuentes y se han almacenado en archivos csv para su posterior tratamiento y análisis.

En el caso de ebird.org las peticiones a la API se han realizado directamente a través de httr realizando consultas GET a sus URLs.

Por limitaciones de la API ha sido necesario pedir los datos de cada día de forma independiente.

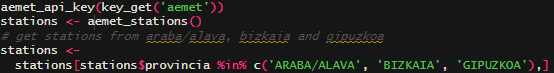


Como no todas las peticiones devuelven el mismo número de columnas es necesario crearlas y ordenarlas para asegurarnos de que el archivo final no tenga errores. Por último, se agregan los datos al archivo.



Para los datos de AEMET disponemos de la librería “climaemet” que nos facilita las peticiones.

Para poder filtrar los datos de Euskadi debemos buscar sus estaciones meteorológicas.



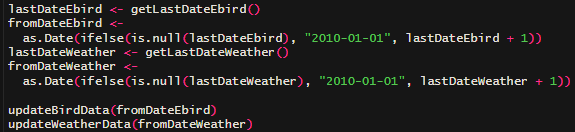
Después pediremos la información meteorológica para las fechas que nos interesen.



Y por último uniremos los datos con la función merge de forma que eliminaremos la información de los territorios que no nos interesan y agregaremos la localización de las predicciones.

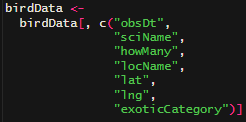


Para ahorrar tiempo en caso de querer actualizar todos estos datos se han creado unas funciones auxiliares que buscaran la última fecha almacenada en los archivos de forma que el código final a ejecutar seria el siguiente.



Tratamiento de datos

A la hora del tratamiento, empezaremos por eliminar columnas innecesarias.



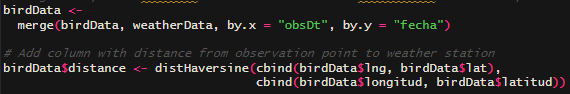
Para cruzar los datos utilizamos la fecha, como esta columna no es única en ninguno de los datasets el resultado de juntarlos multiplicara el número de filas, por cada avistamiento tendremos los datos climáticos de todas las estaciones. Para poder eliminar los resultados innecesarios primero tenemos que añadir identificador a los avistamientos.



Como no todos los resultados meteorológicos incluyen toda la información eliminaremos los que no tengan al menos la temperatura media del día.



Ahora ya podremos unir las tablas, tras lo cual agregaremos una columna con la distancia entre las coordenadas del avistamiento y las de la estación meteorológica, que nos servirá para saber qué datos filtrar más adelante.



La forma más sencilla de limpiar las filas que almacenan datos irrelevantes será ordenar los datos por la distancia que hemos calculado y eliminar las filas que contengan duplicados del identificador. De esta forma nos quedaremos únicamente con una entrada por avistamiento que incluirá la información meteorológica de la estación más cercana.



Por ultimo volveremos a eliminar las columnas auxiliares que hemos creado.

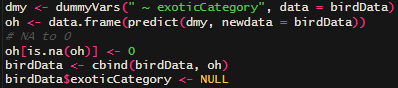


En este punto ya tenemos nuestro dataset unificado y podemos prepararlo para su posterior análisis y uso en sistemas predictivos de inteligencia artificial.

Dividimos la fecha en tres columnas numéricas.



Como la columna exoticCategory es un dato categórico con tres posibles opciones lo transformaremos en tres columnas diferentes con datos binarios.



Como algunas columnas contienen valores nulos tendremos que reemplazarlos, en el caso de howMany los rellenaremos con 1s.



Para la columna velmedia utilizaremos la mediana



Y para prec utilizaremos 0s tanto para los valores nulos como para los marcados como Ip (esto lo hacemos siguiendo las explicaciones de la API de AEMET), además transformaremos toda la columna en datos numéricos.



Por ultimo hemos visto que algunas localizaciones incluían caracteres invisibles que rompían el formato del archivo csv, los reemplazaremos por espacios simples antes de guardar el archivo.

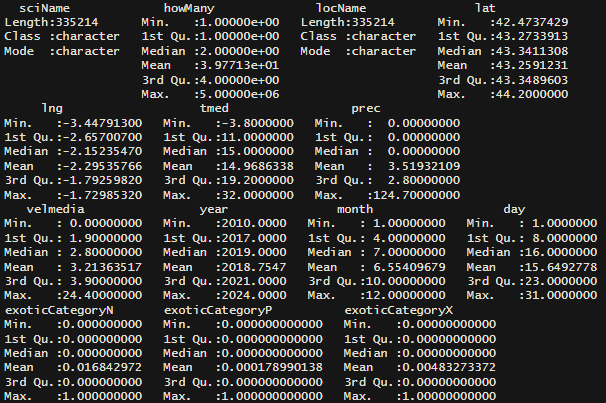


Análisis de datos

En este paso utilizaremos el archivo tratado para buscar información relevante, primero cargaremos el archivo y mostraremos un resumen de sus datos.

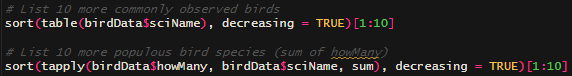


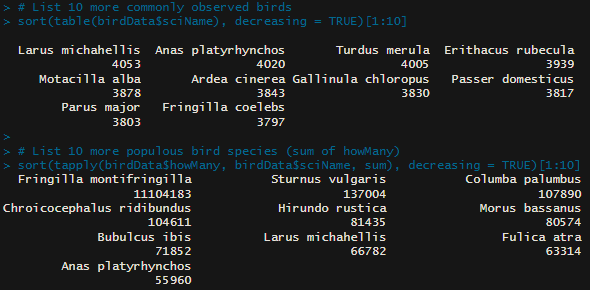
Esto nos dará un primer vistazo con el que podremos ver la cantidad de datos de los que disponemos y su distribución, incluyendo máximos, mínimos, medias, medianas y cuartiles.



Por ejemplo, podremos ver como las columnas howMany y prec tienen una gran diferencia entre sus medias, medianas y máximos, más adelante podremos ver como esto se debe a que, aunque la mayoría de avistamientos tienen datos muy bajos de ambos, hay unos pocos con datos muy desproporcionados que generan la distancia entre media y mediana.

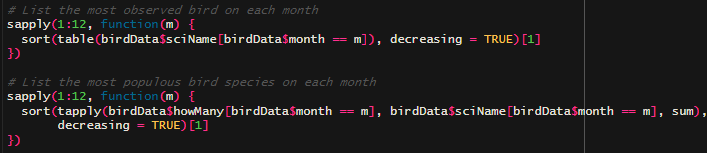
Tras esto hemos listado las 10 especies que más veces se han avistado y las 10 con mayor población.

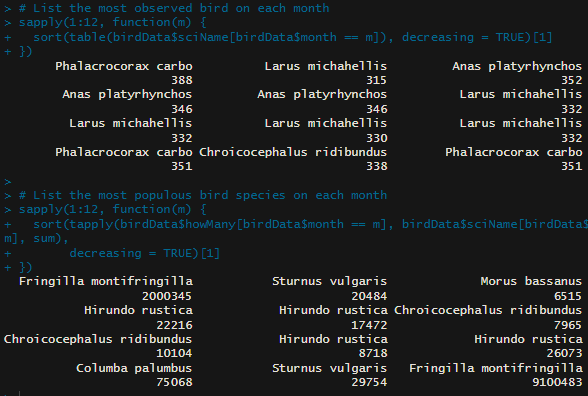




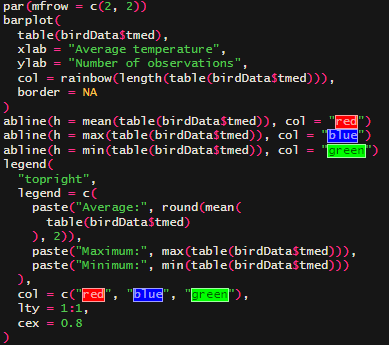
En estos resultados podemos ver como únicamente la especie con más avistamientos aparece en la segunda lista, esto puede deberse a que la mayoría de avistamientos pertenecen a aficionados que únicamente reportan las aves menos comunes mientras unos pocos pertenecen a grupos que realizan recuentos de población y llegan a reportar millones en una sola entrada.

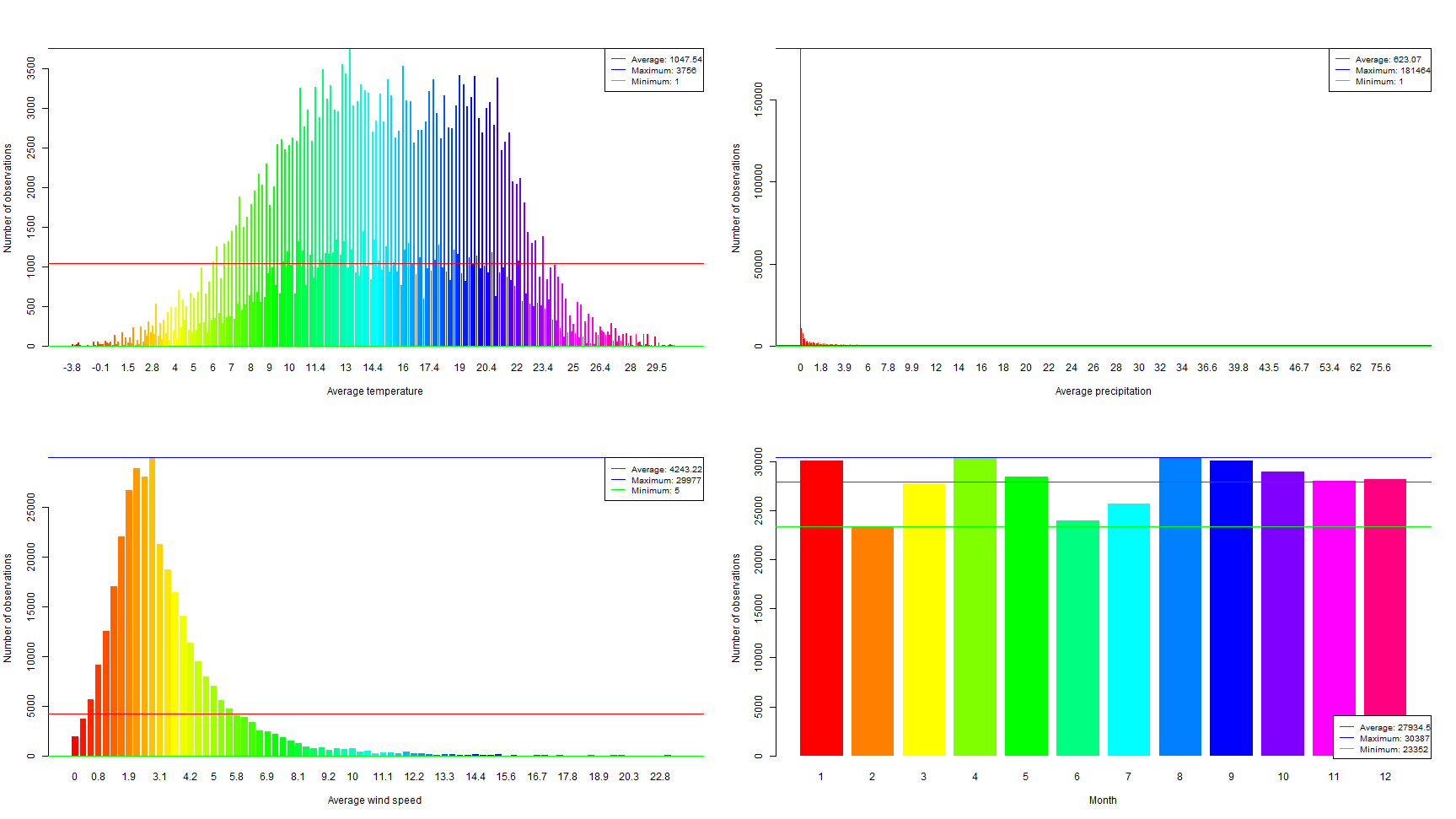
También hemos realizado el listado de la especie más vista y con mayor población de cada mes para comprobar si las migraciones afectan en gran medida a los datos.





Para buscar relaciones entre los avistamientos y el clima se han realizado tres graficas de barras en las que se comparan el número de avistamientos con la temperatura, las precipitaciones y la velocidad del viento respectivamente, finalmente se ha añadido otra grafica que muestra los avistamientos mensuales. El código corresponde al primero.



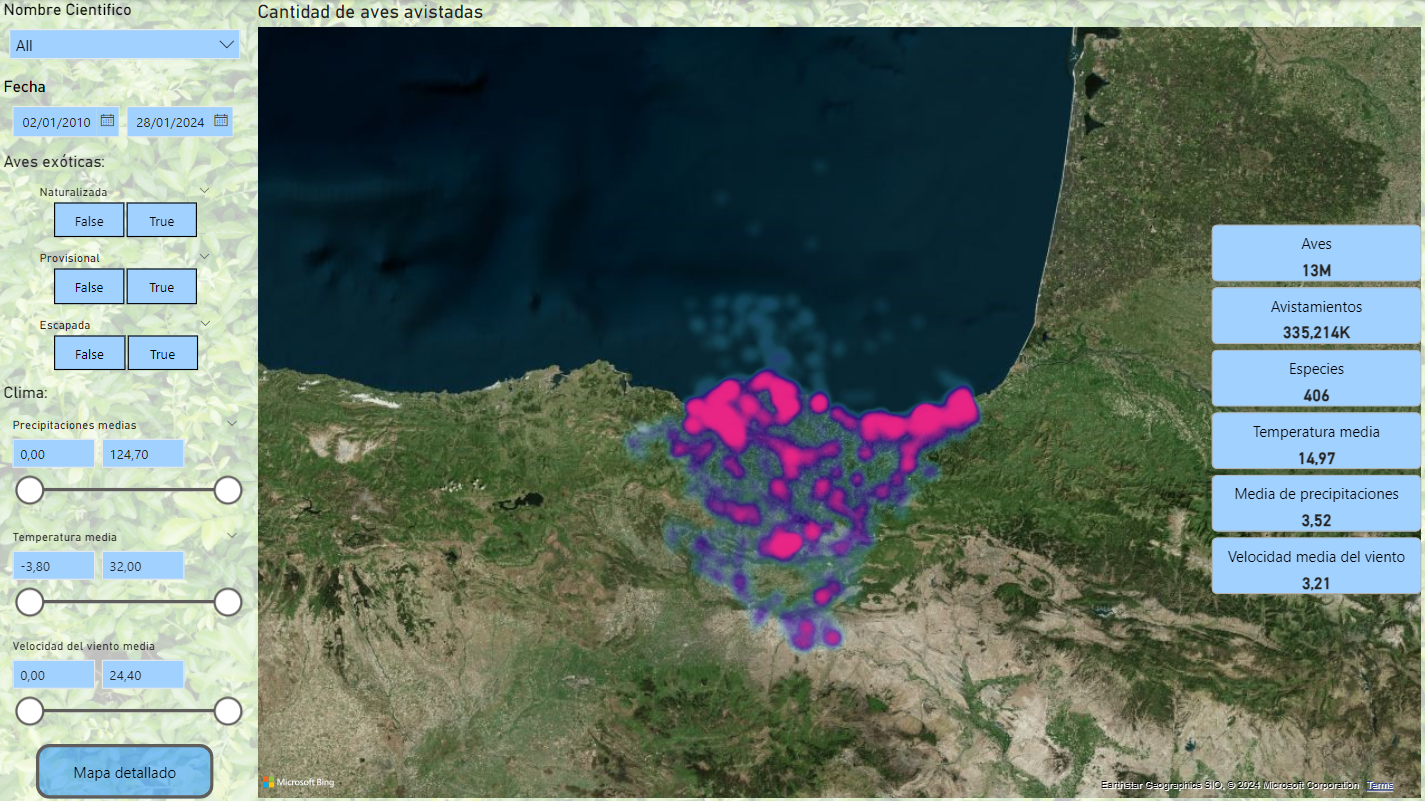


En la primera grafica podemos ver una distribución normal de los avistamientos con respecto a la temperatura media del día.

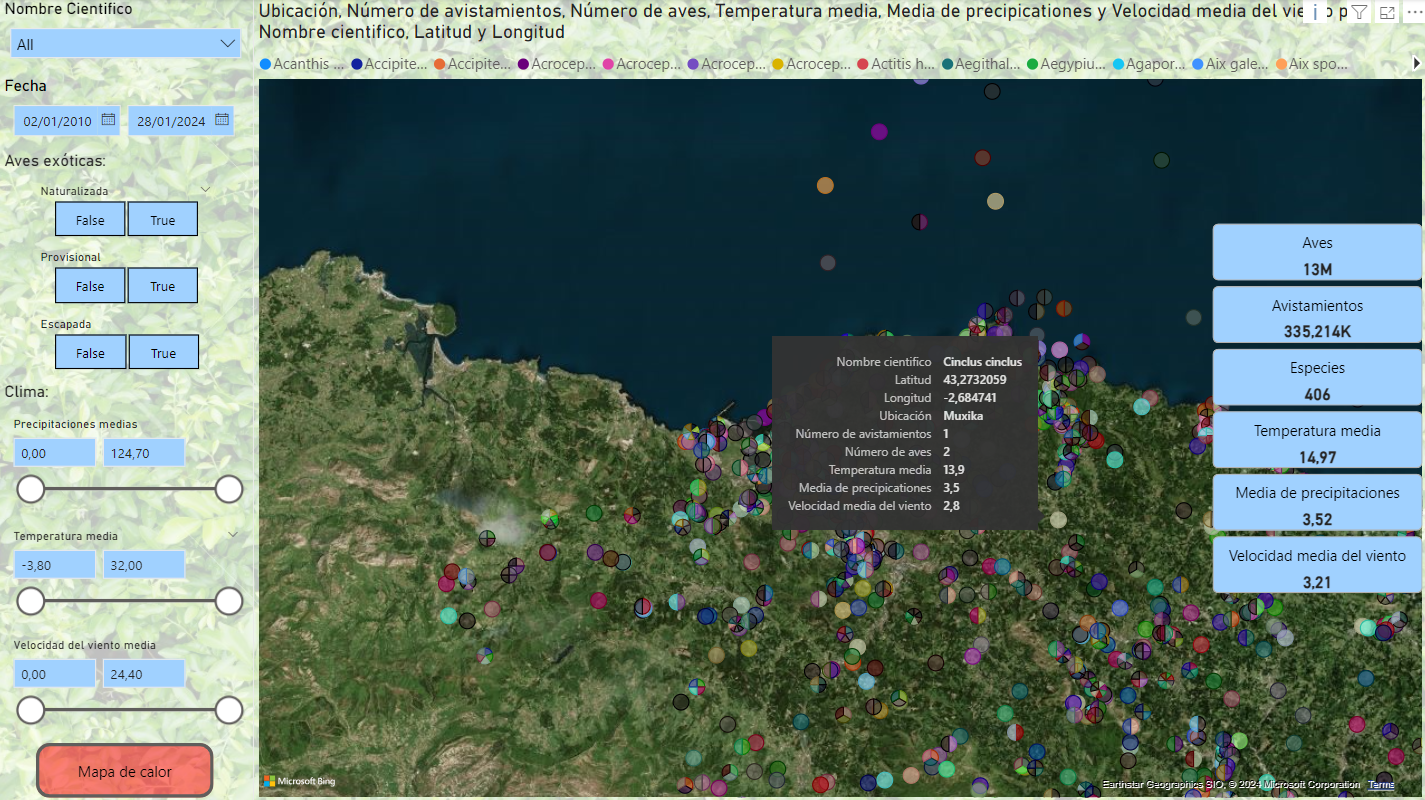
La segunda muestra como apenas se realizan avistamientos con precipitaciones ya que más de la mitad de ellos están en 0

Sobre la velocidad del viento vemos una distribución escorada a la izquierda que nos indica como la mayoría de avistamientos se realizan en días con poco viento.

Para terminar, se ha realizado una presentación con PowerBI en la que se puede visualizar toda la información de forma visual en un mapa interactivo.



Como se ve en la imagen disponemos de filtros para cada parámetro del dataset a excepción de la cantidad de aves que será la utilizada para decidir el color del mapa de calor.



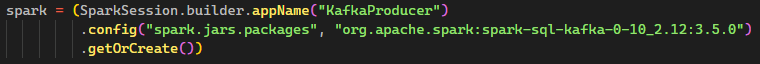
También tenemos un mapa detallado en el que podremos ver los datos específicos de cada punto.

Almacenamiento en Hive

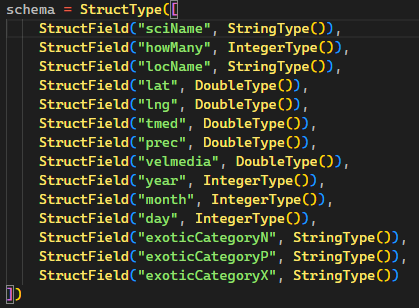
Por ultimo almacenaremos los datos en una base de datos Hive.

Para esto comenzaremos creando un productor de Kafka que enviara los datos de nuestro equipo local a un servidor remoto.

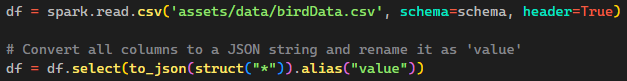
Utilizará una sesión de Spark configurada con las librerías necesarias de Kafka.



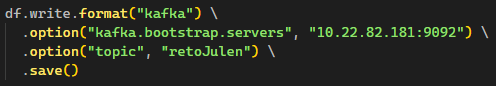
También necesitaremos definir la estructura de nuestros datos de la siguiente manera.



Entonces podremos leer el archivo y transformarlo a un dataframe con una colección de json almacenados en la columna value, también podríamos asignarles identificadores con una columna key.

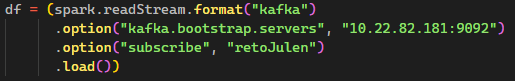


Por ultimo realizaremos el stream contra el servidor.

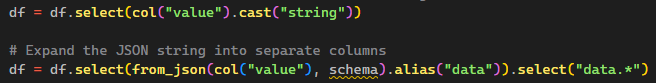


Después haremos un consumidor que ejecutaremos desde nuestra máquina de Hadoop para recibir la información.

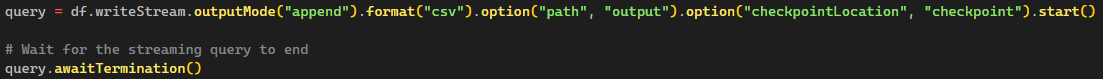
Para ello utilizaremos una sesión de Spark y estructura de datos idénticas a las del productor y leeremos el streaming con la siguiente función.



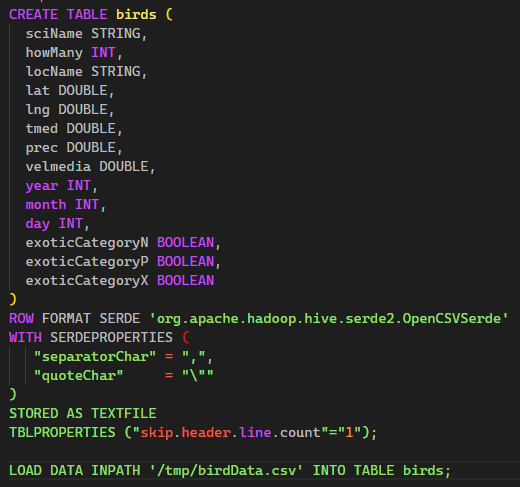
Ahora necesitamos convertir los atributos de los json en columnas.



Por último crearemos el archivo.



Ahora necesitaremos subir esta información al sistema HDFS para finalmente cargar el archivo en la base de datos Hive mediante el siguiente script.



En la configuración de la propia tabla elegimos el formato del csv, separado por comas, con los textos entrecomillados y con las cabeceras en la primera línea.