

Tarea Grande 1

Visualización de Información
Ayudantes: Uranía Donoso y Ricardo Schilling.
Profesor: Denis Parra
Enunciado: 5 de Abril de 2021

Indicaciones

- Fecha de entrega: 22 de Abril de 2021 hasta las 19:59.
 - Se debe entregar la tarea en el repositorio asignado a cada uno por *Github Classroom*. [Link](#).
 - La tarea es **en parejas**.
 - **La copia será sancionada con una nota 1.1 en la tarea, además de las sanciones disciplinarias correspondientes.**
-

Objetivo

El objetivo de esta tarea es que aprendan a:

- Utilizar **Python** para manipular datos.
- Utilizar datos para hacer gráficos utilizando una librería de su elección.
- Utilizar **HTML+CSS+JS** para construir una página web sencilla en la que se muestren las visualizaciones construidas.

Contexto

Reglas Generales

En la parte 1 de su tarea trabajarán con los datos entregados¹, con la finalidad de obtener lo necesario para la confección de un *dashboard*.

Para esto es importante que consideren lo siguiente:

- El entregable de la parte 1 es un *jupyter notebook* (extensión.ipynb), donde se encuentre el **código, gráficos realizados y explicaciones** que les parezcan pertinentes.
- Se deberá utilizar **obligatoriamente** la librería **Pandas** para el procesamiento de los datos.
- De no cumplir con alguno de los dos puntos anteriores, su nota máxima será un 4.
- Se espera que usen la librería *Pandas* de manera **eficiente**, por lo que no estará permitido usar *for* ni equivalentes como listas de mapas o generadores (excepto si se trata de alguna función nativa de *Pandas*).
- Para los gráficos, no se permitirá ningún tipo de **hardcoding**, ya sea de los mismos datos numéricos o de los animes.
- Los gráficos deben tener título y nombres para cada eje según corresponda.
- Esperamos que comenten los gráficos que generen, por un lado explicando las funciones usadas y sus argumentos y por otro los resultados obtenidos.
- El no cumplir con las condiciones anteriores implicará que no se asignará puntaje al gráfico correspondiente.
- No deben subir al repositorio archivos temporales como checkpoints de sus notebooks, archivos como .DS_STORE, u otro tipo de archivos de esa naturaleza.

Datos entregados

En el archivo .csv adjunto a esta tarea encontrarán la siguiente información de animes:

- **ID:** ID del dataset.

¹Valdivieso, H. (2021, Marzo). Anime Recommendation Database 2020, Version 5. Retrieved March 28, 2021 from <https://www.kaggle.com/hernan4444/anime-recommendation-database-2020>.

- **Name:** El nombre de la serie de anime.
- **Score:** Puntaje "promedio" dado por los usuarios.
- **Type:** TV, movie, OVA, etc.
- **Episodes:** Cantidad de episodios.
- **Watching:** Cantidad de usuarios que se encuentran viendo la serie.
- **Completed:** Cantidad de usuarios que terminaron de ver la serie completa.
- **Plan to watch:** Cantidad de usuarios que tienen planeado ver la serie.
- **Season:** Estación del año en el que se publicó el anime.
- **Year:** Año de producción de la serie.

Instrucciones

Parte 1: Manipulación y visualización de datos. (3 pts)

Pregunta 1 (0,5 pts)

Queremos analizar la distribución típica de los puntajes promedio de los animes (para ver, por ejemplo, qué nota tienden a poner los usuarios). Para lo anterior, deben crear un histograma para los scores de los animes, para luego estimar la distribución de la densidad de sus datos.

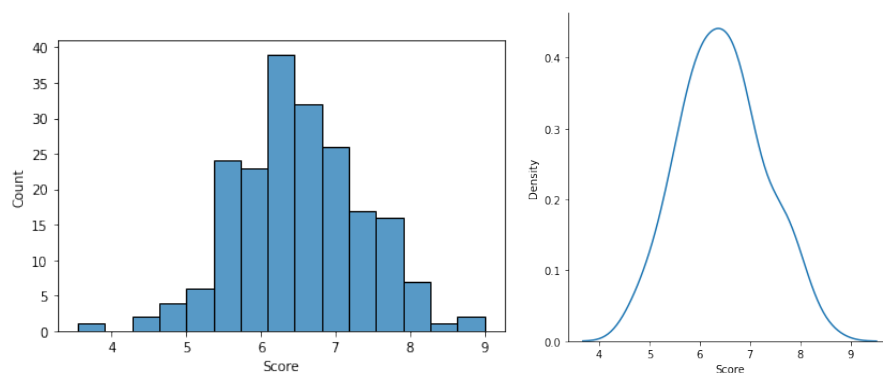


Figura 1: Ejemplo del histograma pedido y la distribución de densidad.

Pregunta 2 (0,5 pts)

Los gráficos anteriores aportan bastante información sobre la distribución de los puntajes generales, sin embargo, también nos interesa cómo el tipo de serie afecta a la distribución, por lo que deberán realizar los mismos gráficos de la pregunta anterior, pero separando por el tipo de serie.

Luego, es necesario que grafiquen todos los tipos de serie en el mismo histograma. Y lo mismo para sus distribuciones.

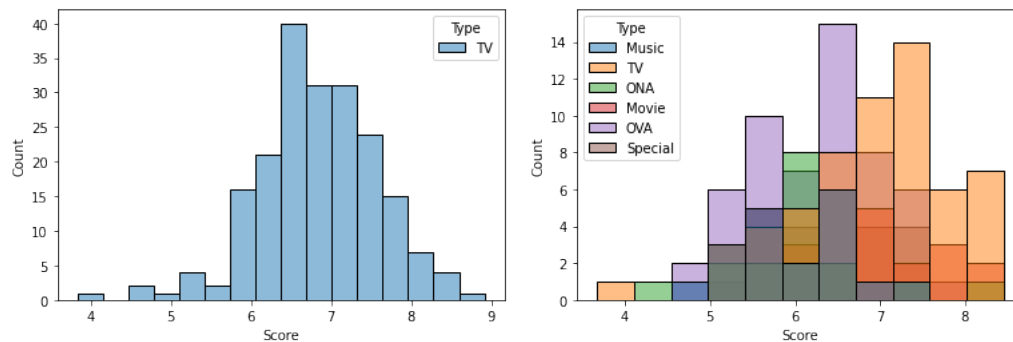


Figura 2: Ejemplo del histograma para un tipo de serie y para todos los tipos.

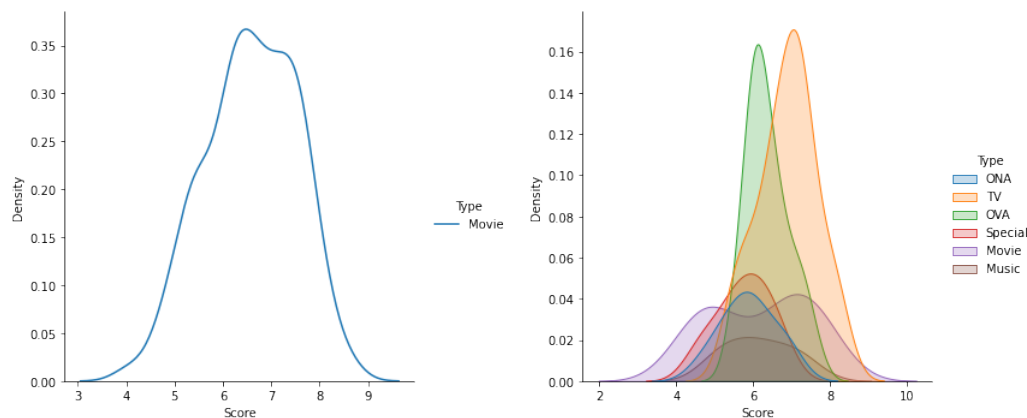


Figura 3: Ejemplo de distribuciones para un tipo de serie y para todos los tipos.

Pregunta 3 (0.5 pts)

Adicionalmente, queremos observar cómo se comportan las distribuciones de personas que están viendo, terminaron de ver o quieren ver los anime mejor recibidos.

Deben realizar tres gráficos de torta, donde se vean estos valores (Completed, Watching, Plan to Watch) para el top 5 de mejores anime.

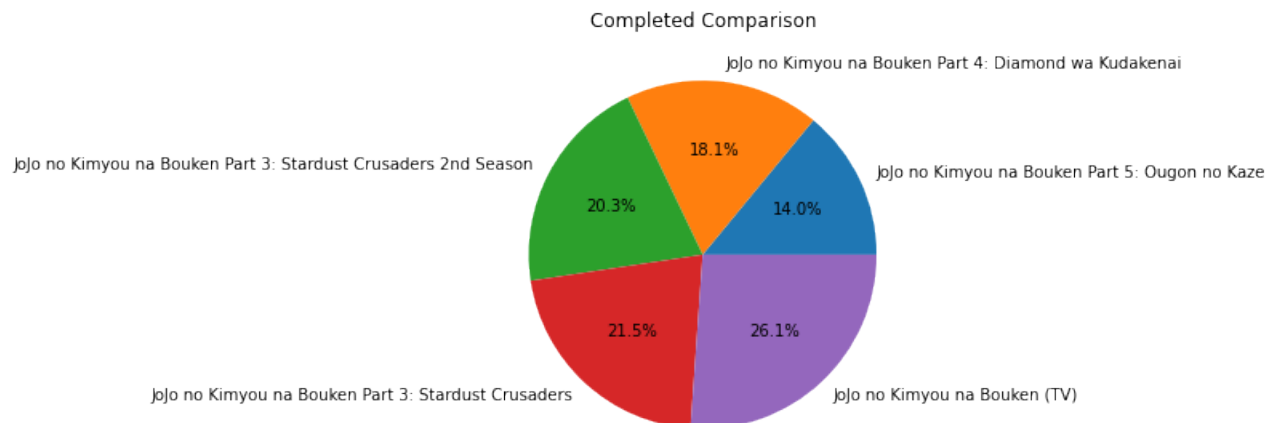


Figura 4: Ejemplo de gráfico de torta mostrando la distribución de personas que han terminado un determinado anime³.

Realizado lo anterior, queremos esos mismos datos, pero **agregados**, es decir, deben sumar los valores de cada columna para los 5 animes y graficarlos.

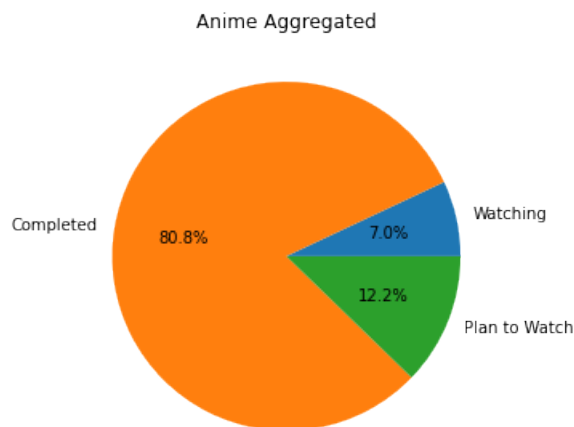


Figura 5: Ejemplo de gráfico de torta mostrando la distribución agregada.

³Lamentablemente, no podemos mostrar los datos para la mejor parte, Steel Ball Run, ya que aún no tiene anime...

Pregunta 4 (1 puntos)

Finalmente, nos gustaría saber como se comportan los scores y cantidad de interacciones (Completed, Watching, Plan to Watch) a lo largo del tiempo, partiendo desde el año 2000.

Para esto, es necesario que, para cada una de esas cuatro cantidades, muestres como se comportan en **promedio** cada año y muestres una **regresión** sobre esos datos ⁴.

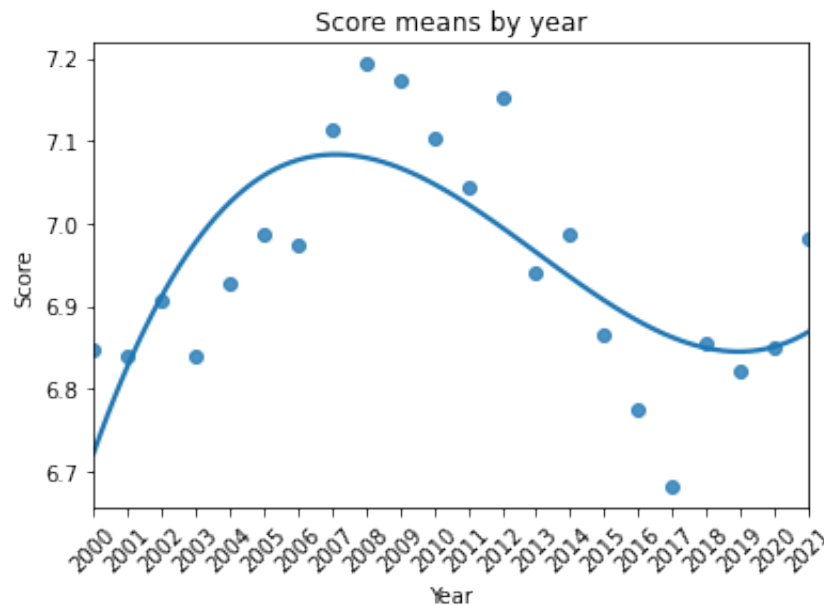


Figura 6: Ejemplo de gráfico que muestra como varía el promedio de puntajes en el tiempo.

Parte 2: Dashboard (1 punto)

Con las visualizaciones ya generadas, deben construir una página web que contenga cada uno de los gráficos mencionados en el enunciado.

Se recomienda revisar la siguiente guía de como funciona *Flexbox*.

Como la confección de los gráficos será evaluada en la parte 1, en esta parte se espera que cumplan con lo siguiente:

⁴No se preocupen, no tiene que hacer ningún cálculo por su cuenta pues todo lo pueden hacer con un regplot de seaborn, por ejemplo.

1. Gráficos de la pregunta 1: inicialmente la página debe mostrar los dos gráficos uno al lado del otro, los cuales deben estar centrados y alineados.

A continuación se detallan los requisitos de cada elemento *HTML* en la página:

- **El cuerpo:**

- Debe tener un margen del 3 % de la pantalla.
- Debe ser un contenedor *flex*.
- Debe tener el color hexadecimal #b0b0b0 de fondo.

- **El encabezado:**

- Debe tener una altura mínima 50px y máxima de 100px.
- Debe ser un contenedor *flex*.
- Los elementos dentro deben estar **centrados**.
- El color es libre.
- Debe tener un título a elección propia que tenga sentido con la página.

- **Los botones:**

- Debe tener los bordes redondos con 7px.

- **El contenedor que contiene las imágenes del inicio:**

- Debe ser un contenedor *flex*.
- Los elementos dentro deben estar **centrados**.
- Los elementos dentro deben saltar de línea cuando no hay suficiente espacio (Ayuda: ver las propiedades de los contenedores *flex* en la guía de *Flexbox* y anexo del enunciado).

- **Los contenedores de imágenes(aplica para todos los gráficos):**

- Deben tener un margen de 10px.
- Altura 400px y ancho de 500px.
- El fondo del contenedor debe ser de color blanco.

- **Las imágenes:**

- Se deben cargar las imágenes desde la carpeta *images/*.
- Deben agregar un texto alternativo que sea igual al título de cada gráfico (**¿Por qué agregar esto?**).
- Todas las imágenes deben tener el mismo tamaño (Ojo que deben cumplir con este requisito usando reglas de *CSS* y no editando las imágenes con un editor de imágenes).

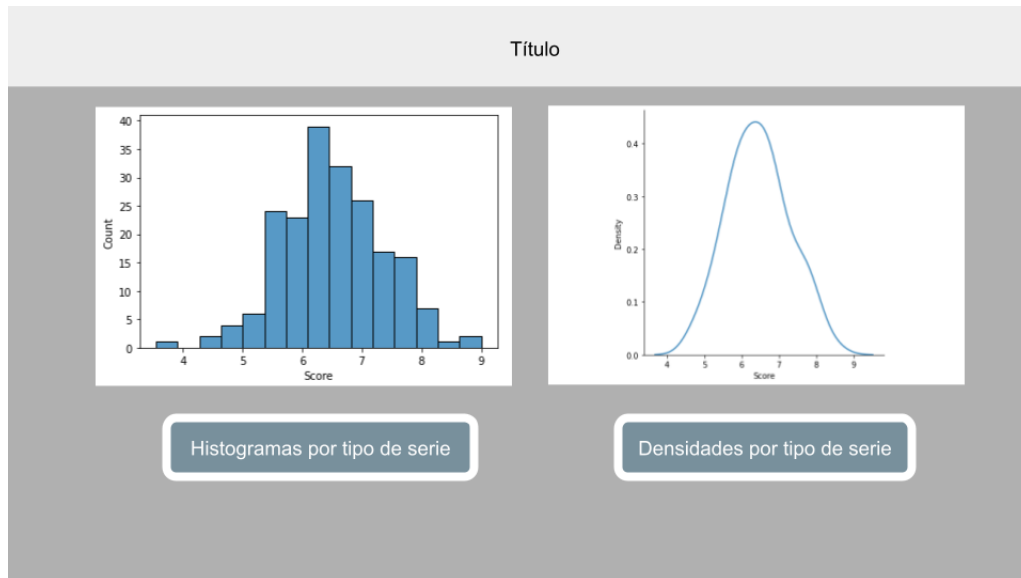


Figura 7: Página referencial de como se debería ver la visualización web.

Parte 3: Interactividad (1 punto)

1. Abajo de los gráficos de la pregunta 1, deberán agregar dos botones con nombres 'Histogramas por tipo de serie' y 'Densidades por tipo de serie'(Figura 8 y 9). Al presionar uno, se deben mostrar los 7 gráficos (el que incluye a todas las series y el de cada tipo de serie) centrados y alineados uno abajo del otro. Al bajar, los gráficos irán apareciendo con un delay de 500 milisegundos, esto usando *Javascript* y su funcionalidad de *ScrollReveal*.
2. Para los gráficos de las preguntas 3 y 4, se espera que usen *Javascript* para crear un *Drop-Down* (uno para los gráficos de la pregunta 3 y otro diferente para los de la pregunta 4) que despliegue los títulos de los gráficos y que al momento de seleccionar uno, este se muestre en la pantalla, permitiéndonos así cambiar de gráfico de manera interactiva. Ustedes deciden dónde colocar estas interacciones, sin embargo, deben ser claramente visibles.

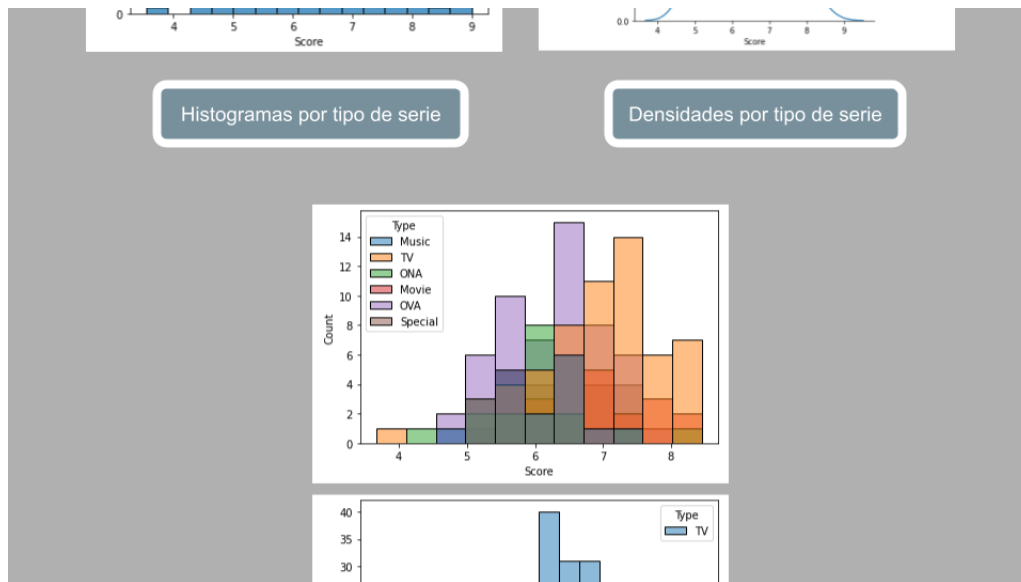


Figura 8: Página referencial de como se debería ver al presionar el botón "Histogramas por tipo de serie".

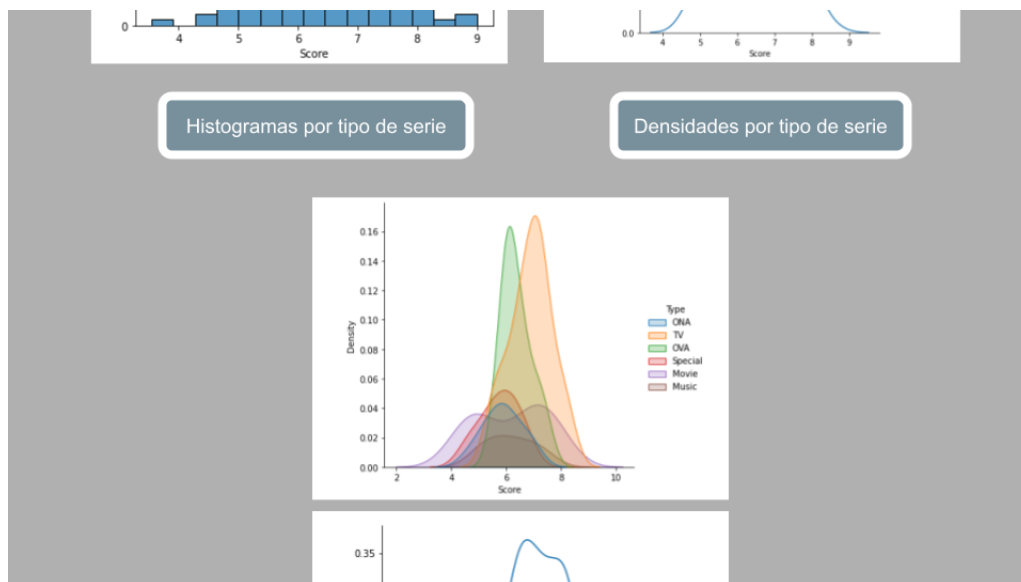


Figura 9: Página referencial de como se debería ver al presionar el botón "Densidades por tipo de serie".

Parte 4: *Fullmetal Alchemist: Brotherhood Mode* (1 punto)

1. ¿Qué pasó con el botón 'DarkMode'? Hoy en día se ha vuelto popular tener la opción de convertir una página o *app* en modo oscuro. Sin embargo, nosotros iremos un paso más allá y cambiaremos este modo a uno con temática del animé

con mayor score: "Fullmetal Alchemist: Brotherhood". Esta sección está destinada para que jueguen un poco con *Javascript* y le den ese comportamiento a su página. Al presionar el botón '*Fullmetal Alchemist: Brotherhood Mode*', debe ocurrir lo siguiente:

- El cuerpo debe cambiar a tener una imagen de la serie, esta la escogen ustedes.
- El encabezado debe cambiar de color a **rojo** .
- El botón debe cambiar su texto a '*NormalMode*'.

Después queremos poder volver a nuestro modo normal de la página (Sin recargarla, pruébenlo si quieren). Al presionar en el botón '*NormalMode*', debe ocurrir lo siguiente:

- El cuerpo debe cambiar de color a #b0b0b0.
- El encabezado debe volver a su color.
- El botón debe cambiar su texto a '*Fullmetal Alchemist: Brotherhood Mode*'.

Nota: Para poder asegurarnos de que no tengan ningún problema al agregar *JS* a la página, deberán abrir la consola (CMD) en **la carpeta donde esté el archivo *HTML* de la página** y correr el siguiente comando '*python -m http.server*', el cual les entregará el puerto en que están corriendo el servidor de python que acaban de crear.

Luego, en su navegador deben colocar la URL '*localhost:XXXX/YYYYY.html*', donde '*XXXX*' es el puerto del servidor y '*YYYYY*' es el nombre del archivo *HTML*.

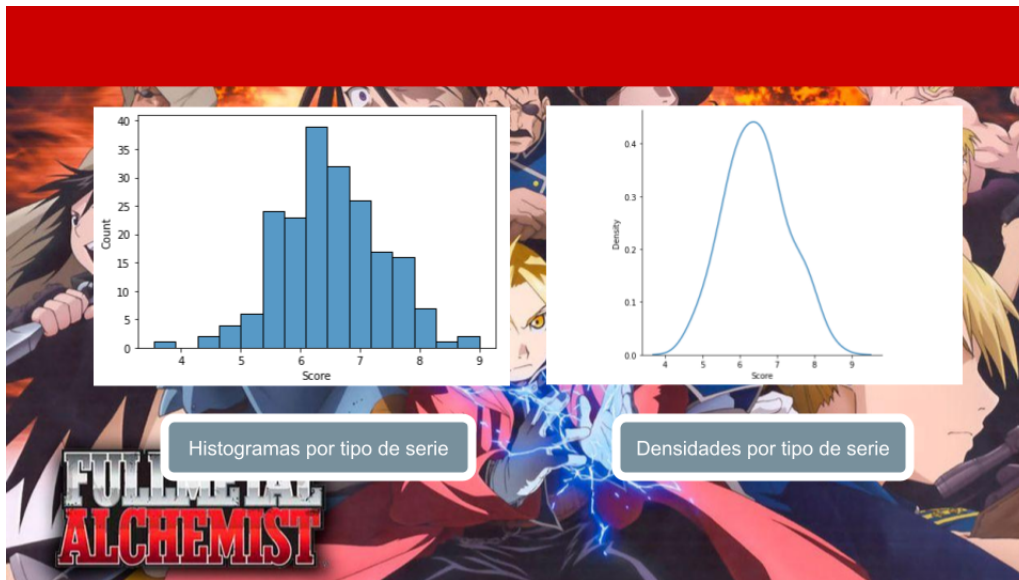


Figura 10: Página referencial de pagina en modo oscuro.

Parte 5: Bonus Concurso de Memes (0.3 puntos)

1. Para cada tarea, a partir de esta, tendrán una bonificación de dos décimas aquellos alumnos que elaboren y envíen los 3 mejores memes dentro del curso. Para hacer envío del meme, junto a su tarea, deben entregar el archivo en formato JPG o PNG o GIF para luego ser evaluado por nuestro comité especializado en memes. Este archivo debe tener el siguiente nombre: meme_TG1_apellidos.gif
2. Nuestro comité de memes tendrá en mente la siguiente rúbrica para la evaluación:
 - Nivel de creatividad.
 - Relación con los contenidos del curso.
 - Sentido del humor.
 - Respeto con los compañeros/ayudantes/profesores al usar imágenes de otras personas, siempre verificar la autorización del otro para evitar ofensas.
3. Las desiciones del comité no son apelables.

Formato de entrega.

Para evaluar con facilidad la parte 1 de esta tarea, deberán realizarla en un *jupyter notebook* y subir este archivo de extensión `.ipynb` a su repositorio asignado.

En cuanto a la parte 2, deben incluir **todos** los archivos que permitan visualizar su página correctamente. En el repositorio base de la tarea, encontrarán una estructura propuesta que les ayudará a cumplir con el formato esperado.

Entregas Atrasadas

Si así lo desea, existe la posibilidad de entregar la tarea fuera de plazo, con hasta 3 horas de atraso.

Cualquier entrega realizada pasada la hora estipulada de entrega será considerada como atrasada, **sin excepciones**. Cada hora o fracción implicará un descuento de **1.5 puntos** a su nota final.

Debido a lo anterior, recomendamos fuertemente enviar su tarea con anticipación, realizando commits intermedios de ser necesario.