

Proyecto de Simulación de Eventos Discretos

Raimel Romaguera Puig David Sanchez Iglesias
Yoel Enriquez Sena Manuel Alejandro Gamboa

June 2024

Introducción

Este proyecto consiste en simular los resultados futuros de un evento deportivo, en este caso será la lucha Greco-romana en las Olimpiadas de París que comienzan el 26 de Julio de 2024. Donde para cada categoría se escogerán los atletas clasificados, se hará una indagación en sus enfrentamientos dos a dos para hallar la probabilidad que tienen entre ellos de ganar o perder. Luego de tener esto se distribuirán los atletas en los cuadros de enfrentamientos siguiendo las reglas que rigen en la actualidad, aquí se verá una respectiva aleatoriedad, luego haciendo uso de la generación de variables aleatorias con el apoyo de la estadística obtendremos los resultados de los combates y del evento.

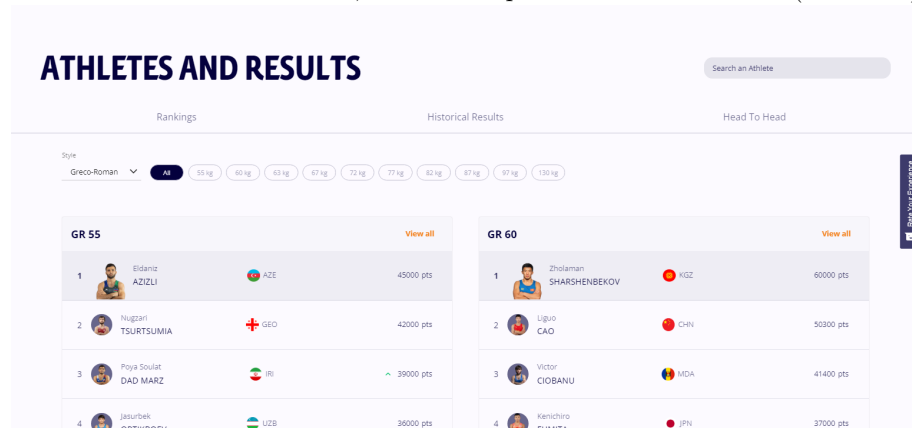
Es común que la gente quiera saber con anticipación los resultados de algún evento, y más aún, un evento deportivo como las Olimpiadas, pero más allá de eso, este trabajo tiene objetivo investigar la veracidad de que, en un enfrentamiento entre dos atletas, tiene relevancia el historial de encuentros anteriores entre los mismos, sería una desilusión saber de que un combate es totalmente aleatorio, o que siempre los que son número uno en el ranking van a obtener la medalla de oro. Para el final de este proyecto veremos que tanto acierto tiene la simulación, ya que simulará los encuentros de las olimpiadas de Tokio 2020 y se compararán resultados.

Las variables que se conocen en este evento son los combates dos a dos de todos los atletas clasificados, para cada combate se sabe, el nombre del atleta 1, el nombre del atleta 2, la categoría de peso, la clasificación de la victoria (esta está determinada en su mayoría por puntos, más adelante se describirá con detalle), el atleta ganador y la fecha. Las variables que se desconocen, y por tanto se van a generar son la distribución de los cuadros de enfrentamientos y los resultados de los enfrentamientos para cada par de atletas que les toque enfrentarse, luego el resultado que se obtendrá es el medallero del evento para cada categoría.

Detalles de Implementación

Obtención de datos: El programa se implementará en Python. La información de los atletas clasificados para las Olimpiadas París se obtuvo de [1], la información de los enfrentamientos de entre los atletas se obtuvo de la página oficial de la Federación Internacional de Lucha[2]. Primero se debe definir la estructura donde se almacenará esta información, para ello se hará uso de la biblioteca `sqlite3` para crear una base de datos local. Se creará una tabla para cada categoría de peso para guardar el nombre, país y ranking de cada atleta clasificado de esa categoría, también para cada categoría se creará una tabla donde se registrará todos los combates registrados que incluyen a dos de los clasificados, o sea los combates entre los atletas clasificados dos a dos.

En la página oficial de la UWW(United World Wrestling) hay una pestaña llamada Athletes and Results, en la cual aparecen todos los atletas(Picture 1).

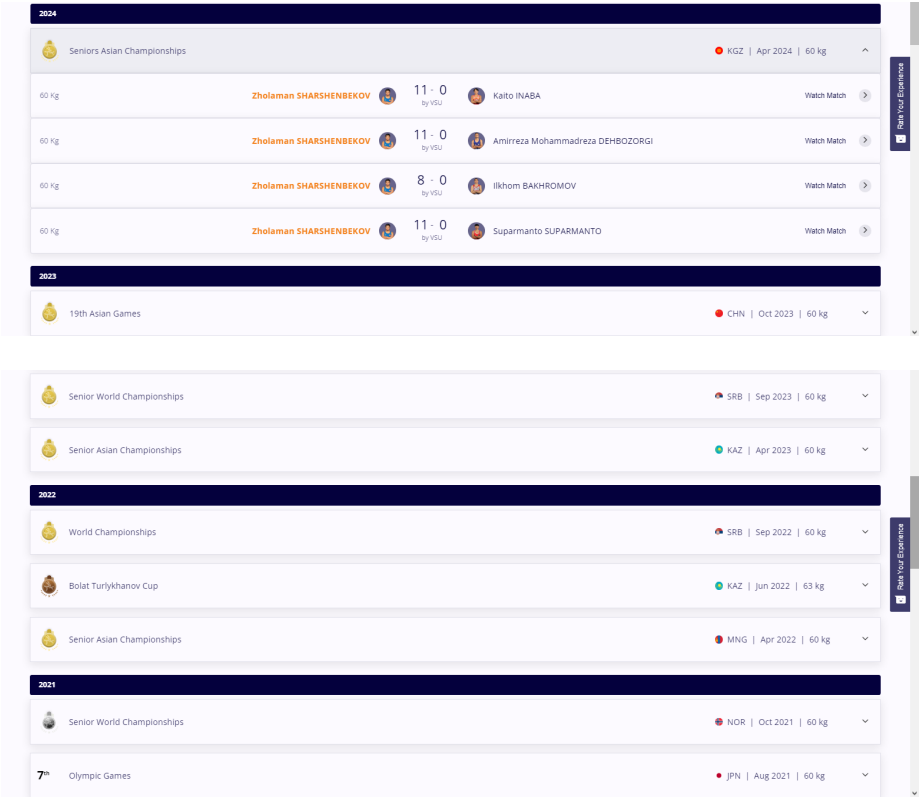


The screenshot shows the 'ATHLETES AND RESULTS' page on the UWW website. It features a search bar, tabs for 'Rankings', 'Historical Results', and 'Head To Head', and a filter for 'Style: Greco-Roman'. Two weight categories are visible: GR 55 and GR 60. Each category lists the top four athletes with their rank, name, country flag, and points.

Style	Rankings	Historical Results	Head To Head																																
Greco-Roman	55 kg, 60 kg, 63 kg, 67 kg, 70 kg, 73 kg, 82 kg, 87 kg, 97 kg, 130 kg																																		
GR 55	<table><tr><td>1</td><td>Eldaniz AZIZLI</td><td>AZE</td><td>45000 pts</td></tr><tr><td>2</td><td>Nugzar TSURTSUMIA</td><td>GEO</td><td>42000 pts</td></tr><tr><td>3</td><td>Poya Soulat DAD MARZ</td><td>IRI</td><td>39000 pts</td></tr><tr><td>4</td><td>Jasurbek ORTIKBOEV</td><td>UZB</td><td>36000 pts</td></tr></table>	1	Eldaniz AZIZLI	AZE	45000 pts	2	Nugzar TSURTSUMIA	GEO	42000 pts	3	Poya Soulat DAD MARZ	IRI	39000 pts	4	Jasurbek ORTIKBOEV	UZB	36000 pts	GR 60	<table><tr><td>1</td><td>Zholaman SHARSHENBEKOV</td><td>KGZ</td><td>60000 pts</td></tr><tr><td>2</td><td>Liguo CAO</td><td>CHN</td><td>50300 pts</td></tr><tr><td>3</td><td>Victor CIOBANU</td><td>MDA</td><td>41400 pts</td></tr><tr><td>4</td><td>Kienichiro FUMITA</td><td>JPN</td><td>37000 pts</td></tr></table>	1	Zholaman SHARSHENBEKOV	KGZ	60000 pts	2	Liguo CAO	CHN	50300 pts	3	Victor CIOBANU	MDA	41400 pts	4	Kienichiro FUMITA	JPN	37000 pts
1	Eldaniz AZIZLI	AZE	45000 pts																																
2	Nugzar TSURTSUMIA	GEO	42000 pts																																
3	Poya Soulat DAD MARZ	IRI	39000 pts																																
4	Jasurbek ORTIKBOEV	UZB	36000 pts																																
1	Zholaman SHARSHENBEKOV	KGZ	60000 pts																																
2	Liguo CAO	CHN	50300 pts																																
3	Victor CIOBANU	MDA	41400 pts																																
4	Kienichiro FUMITA	JPN	37000 pts																																

Figure 1

Cada tarjeta de atleta lleva a la página del atleta, que a su vez tiene una pestaña llamada Results, donde contiene todos sus combates en una competencia de alcance internacional (Figure 2)



4

La idea es para cada jugador clasificado ver todos sus combates, y luego almacenar solo los combates que incluyen otros jugadores clasificados. Para ello haremos uso del módulo `selenium` con el cual se podrá inicializar un webdriver para navegar e interactuar con las ventanas desplegables de los resultados de los atletas, para poder recolectar la información de manera dinámica, luego se procede a filtrar estos datos y a llenar las tablas de los enfrentamientos 2 a 2 de las diferentes categorías. Llegado a este punto se entrará a la construcción del

modelo. Primero se obtienen todos los atletas clasificados en una categoría dada haciendo una consulta a la base de datos pasándole como parámetro a la consulta una categoría dada. Luego se procede a calcular, para cada par de jugadores, la probabilidad que tienen de ganar o perder un enfrentamiento contra dicho oponente. Para ello se implementó la función `GetMatrixVersusPlayers(clashes, n)`, donde `clashes` son todos los enfrentamientos entre los atletas clasificados, y `n` es la cantidad de atletas. Se crea una matriz, en la cual se registrará un medidor de cuánto supera un atleta a otro. Este medidor está influenciado por las victorias, y el tipo de victoria de un oponente sobre otro. En lucha Greco-romana existen diferentes maneras de ganar, por ejemplo, Victoria por diferencia de puntos (VOP), esta es cuando al terminar el combate, el vencedor tiene una diferencia de puntos de entre 1 y 7 puntos sobre el perdedor sin que este haya anotado puntos, de manera similar es (VPO1) que es cuando el perdedor logra al menos un punto, otra variante de victoria es la Victoria por Superioridad Técnica (VSU), ocurre en el escenario donde el vencedor alcanza una diferencia de puntos de al menos 8, y el perdedor no ha logrado puntos, al igual que VSU1 que es cuando el perdedor logra alcanzar al menos un punto, cabe resaltar que hay una diferencia en una victoria por diferencia de puntos y una victoria por superioridad técnica [3]. Estos detalles están implementados en la función `GetPointsByWin(status)` donde `status` es la condición de victoria, ejemplo VPO o VSU.

También entra en escena el tiempo, ya que no se puede tomar en cuenta de igual manera un enfrentamiento del 2018 a uno del 2024, para ello, se definirá una fecha mínima, y una fecha máxima (la fecha actual), como fecha mínima se tomará Enero del 2014, hace diez años. Para definir un indicador que reduzca las fechas más antiguas, se hará como sigue:

$$index = \frac{c - min}{max - min} \quad (1)$$

donde c es la fecha del combate, esto dará un valor entre 0 y 1, donde mientras más reciente es el enfrentamiento más se acerca a 1, por lo tanto mayor valor alcanza dicho índice, y mientras más se aleje de la fecha actual más reduce los valores. Luego de tener este índice se suman todos los puntos multiplicados por el índice

$$m_{ij} = \sum_1^k p_l * index \quad (2)$$

donde p es la cantidad de puntos alcanzados por el atleta i contra el atleta j en el enfrentamiento l , donde k es la cantidad de enfrentamientos entre i y j . Luego de esto se suman los puntos que hizo i contra j , y j contra i para obtener un total, luego la probabilidad $P(i) = \frac{m_{ij}}{total}$, y $P(j) = 1 - P(i)$.

Ya se tiene la probabilidad de cada par de atleta bien definida en una matriz, ahora se distribuirán los atletas en los cuadros de enfrentamiento, pero no es completamente aleatorio, en las competencias para evitar que los favoritos se encuentren en las primeras rondas, se distribuyen los 4 primeros en el ranking en los cuatro cuadros de combates, también se tiene en cuenta que los atletas que son del mismo país no caigan en el mismo cuadro, luego de cumplir estos dos requisitos, el resto de los atletas se distribuyen de manera aleatoria por los restantes espacios, esta distribución está implementada en la función `DistCuadrosEnfrentamientos(athletes)`, donde `athletes` son los atletas clasificados en una categoría dada. Luego a la hora de enfrentar dos atletas se genera una variable aleatoria x entre 0 y 1, si $x < P(i)$ entonces se toma al atleta i como ganador del enfrentamiento, si $x > P(i)$ entonces se toma a i como el perdedor del enfrentamiento, luego se continúa con el evento hasta encontrar el oro, la plata, y los bronce, este proceso se repite una cantidad de veces para encontrar en promedio, quienes son los posibles atletas que alcancen medallas.

[1]https://en.wikipedia.org/wiki/Wrestling_at_the_2024_Summer_Olympics_

_Qualification

[2]<https://uww.org/athletes-results>