

# Evaluación de fuerza de cartas de los superamigos por simulación

Las cartas de los superamigos son un juego en el que mediante selección de ciertas características de las cartas, se realizan combates. La carta con el valor más alto es la ganadora. La pregunta es ¿Cuáles son las mejores cartas?

Una posibilidad para contestar esta pregunta es, si conocemos como es la distribución de cada variable, se podría rankear a cada carta. Como resultado obtendríamos que tan buena es cada carta en una "disciplina" determinada. Con estos rankings podríamos construir un ranking general como combinación de todas ellas.

```
In [5]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import copy
import random
from IPython.display import Image
plt.rcParams['figure.figsize'] = [15, 10]
%matplotlib inline
```

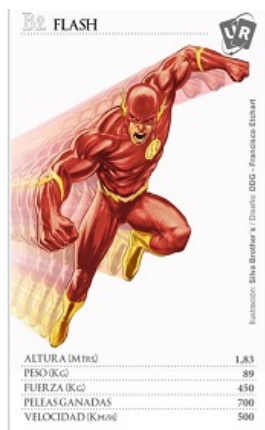
```
In [6]: f = pd.read_csv('/Users/dbikiel/Dropbox/proyectos de programacion/cartas superamigos/listado_DC.csv')
```

La lista de las cartas se puede ver a continuación

```
In [10]: Image("DC1.jpg")
```

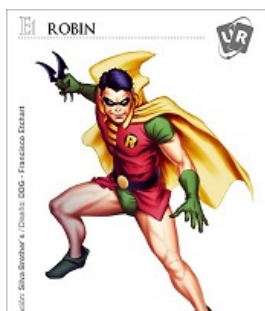
```
Out[10]:
```





In [11]: Image("DC2.jpg")

Out[11]:




|                  |      |
|------------------|------|
| Bat              |      |
| ALTURA (Mtrs)    | 1.72 |
| PESO (Kg)        | 68   |
| FUERZA (Kg)      | 350  |
| PELEAS GANADAS   | 615  |
| VELOCIDAD (Km/h) | 65   |

|                  |      |
|------------------|------|
| G                |      |
| ALTURA (Mtrs)    | 1.72 |
| PESO (Kg)        | 76   |
| FUERZA (Kg)      | 250  |
| PELEAS GANADAS   | 60   |
| VELOCIDAD (Km/h) | 28   |

|                  |      |
|------------------|------|
| Barbas           |      |
| ALTURA (Mtrs)    | 1.58 |
| PESO (Kg)        | 99   |
| FUERZA (Kg)      | 80   |
| PELEAS GANADAS   | 176  |
| VELOCIDAD (Km/h) | 10   |

|                  |      |
|------------------|------|
| Barbas           |      |
| ALTURA (Mtrs)    | 2.10 |
| PESO (Kg)        | 180  |
| FUERZA (Kg)      | 650  |
| PELEAS GANADAS   | 300  |
| VELOCIDAD (Km/h) | 70   |

|   |      |
|---|------|
| H BATICHICA   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                    |      |
| ALTURA (Mtrs)   | 1.76 |
| PESO (Kg)   | 55   |
| FUERZA (Kg)   | 300  |
| PELEAS GANADAS  | 513  |
| VELOCIDAD (Km/h)  | 60   |

|   |      |
|---|------|
| H GORILLA GRODD   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                    |      |
| ALTURA (Mtrs)   | 251  |
| PESO (Kg)   | 365  |
| FUERZA (Kg)   | 1200 |
| PELEAS GANADAS  | 301  |
| VELOCIDAD (Km/h)  | 60   |

|  |      |
|--|------|
| H BLACK ADAM   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                     |      |
| ALTURA (Mtrs)  | 1.83 |
| PESO (Kg)  | 114  |
| FUERZA (Kg)  | 1800 |
| PELEAS GANADAS   | 500  |
| VELOCIDAD (Km/h)   | 300  |

|   |      |
|---|------|
| H BRAINIAC  |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                      |      |
| ALTURA (Mtrs)   | 1.98 |
| PESO (Kg)   | 285  |
| FUERZA (Kg)   | 800  |
| PELEAS GANADAS  | 215  |
| VELOCIDAD (Km/h)  | 55   |

|  |      |
|--|------|
| G DR. FATE   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                     |      |
| ALTURA (Mtrs)  | 2.00 |
| PESO (Kg)  | 100  |
| FUERZA (Kg)  | 600  |
| PELEAS GANADAS   | 750  |
| VELOCIDAD (Km/h)   | 100  |


|  |      |
|--|------|
| G TORNADO ROJO   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                     |      |
| ALTURA (Mtrs)  | 1.90 |
| PESO (Kg)  | 295  |
| FUERZA (Kg)  | 800  |
| PELEAS GANADAS   | 716  |
| VELOCIDAD (Km/h)   | 200  |

|   |      |
|---|------|
| G MUJER HALCON  |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                      |      |
| ALTURA (Mtrs)   | 1.77 |
| PESO (Kg)   | 65   |
| FUERZA (Kg)   | 650  |
| PELEAS GANADAS  | 540  |
| VELOCIDAD (Km/h)  | 100  |

|  |      |
|--|------|
| G SINIESTRO  |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                       |      |
| ALTURA (Mtrs)  | 1.83 |
| PESO (Kg)  | 93   |
| FUERZA (Kg)  | 830  |
| PELEAS GANADAS   | 351  |
| VELOCIDAD (Km/h)   | 170  |

|   |      |
|---|------|
| H CANARIO NEGRO   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                      |      |
| ALTURA (Mtrs)   | 1.70 |
| PESO (Kg)   | 55   |
| FUERZA (Kg)   | 400  |
| PELEAS GANADAS  | 616  |
| VELOCIDAD (Km/h)  | 65   |

|   |      |
|---|------|
| H ZATANNA   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                      |      |
| ALTURA (Mtrs)   | 1.73 |
| PESO (Kg)   | 62   |
| FUERZA (Kg)   | 850  |
| PELEAS GANADAS  | 725  |
| VELOCIDAD (Km/h)  | 53   |

|  |      |
|--|------|
| H3 ATOMO   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                       |      |
| ALTURA (Mtrs)  | 1.82 |
| PESO (Kg)  | 82   |
| FUERZA (Kg)  | 340  |
| PELEAS GANADAS   | 480  |
| VELOCIDAD (Km/h)   | 50   |

|   |      |
|---|------|
| H4 CYBORG   |      |
|  |      |
| Ilustración: Silva Brothers y Claudio DDO - Francisco Echazart                        |      |
| ALTURA (Mtrs)   | 2.00 |
| PESO (Kg)   | 135  |
| FUERZA (Kg)   | 700  |
| PELEAS GANADAS  | 716  |
| VELOCIDAD (Km/h)  | 85   |

In [183]: f

Out[183]:

| LETRA | NUMERO | NOMBRE | ALTURA | PESO | FUERZA | PELEAS | VELOCIDAD |
|-------|--------|--------|--------|------|--------|--------|-----------|
|-------|--------|--------|--------|------|--------|--------|-----------|

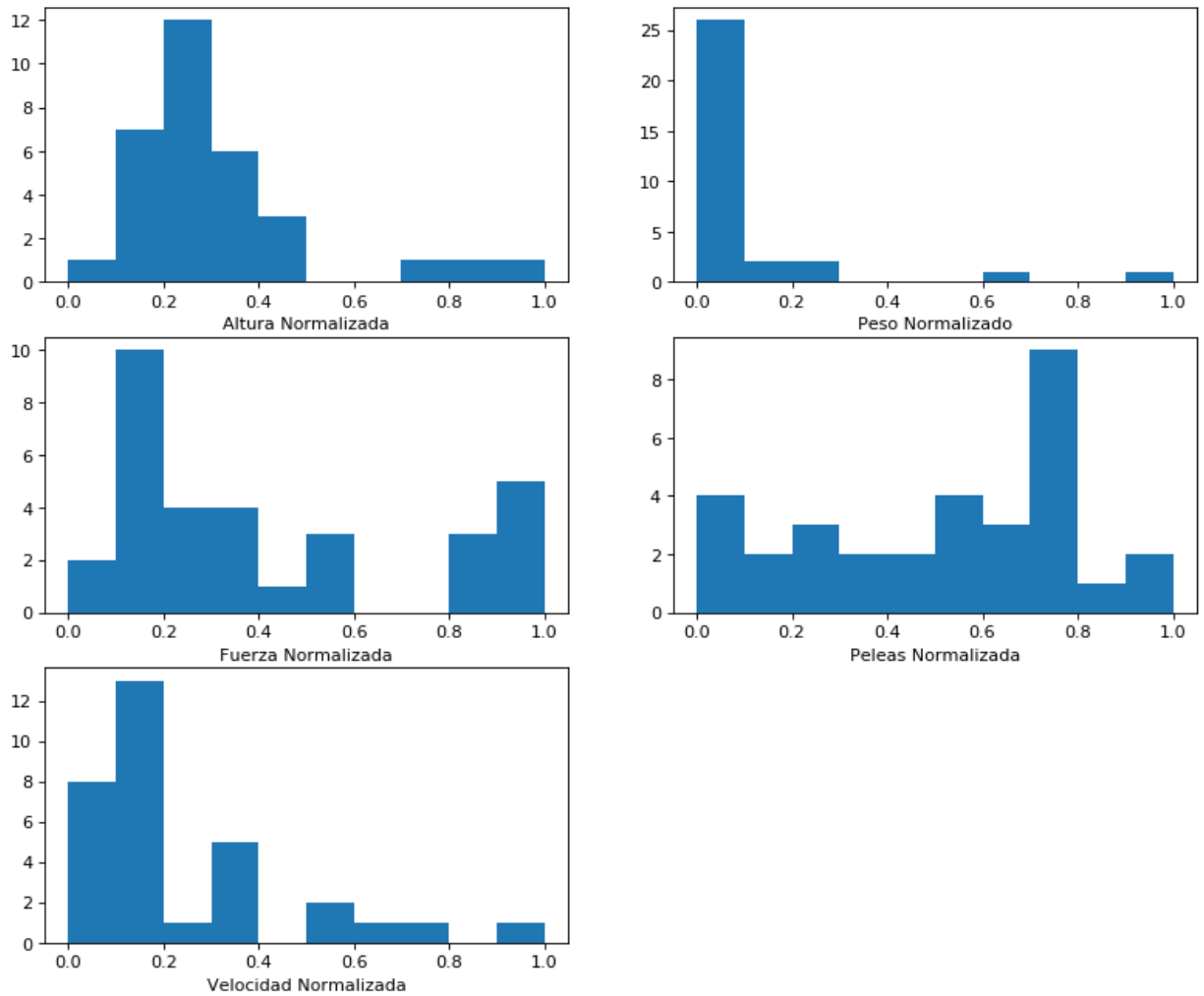
|           |   |   |                   |      |      |      |     |     |
|-----------|---|---|-------------------|------|------|------|-----|-----|
| <b>0</b>  | A | 1 | Superman          | 1.92 | 102  | 2000 | 990 | 400 |
| <b>1</b>  | A | 2 | Shazam            | 1.95 | 110  | 2000 | 700 | 300 |
| <b>2</b>  | A | 3 | Aquaman           | 1.86 | 92   | 1000 | 725 | 110 |
| <b>3</b>  | A | 4 | Martian Manhunter | 2.04 | 136  | 1900 | 750 | 200 |
| <b>4</b>  | B | 1 | Batman            | 1.89 | 95   | 500  | 910 | 70  |
| <b>5</b>  | B | 2 | Flash             | 1.83 | 89   | 450  | 700 | 500 |
| <b>6</b>  | B | 3 | Flecha Verde      | 1.89 | 89   | 450  | 780 | 55  |
| <b>7</b>  | B | 4 | Lobo              | 2.05 | 139  | 1750 | 525 | 100 |
| <b>8</b>  | C | 1 | Mujer Maravilla   | 1.88 | 89   | 1800 | 890 | 100 |
| <b>9</b>  | C | 2 | Hombre Halcon     | 1.95 | 114  | 1100 | 750 | 100 |
| <b>10</b> | C | 3 | Linterna Verde    | 1.83 | 91   | 1200 | 730 | 170 |
| <b>11</b> | C | 4 | Gatubela          | 1.80 | 61   | 300  | 250 | 60  |
| <b>12</b> | D | 1 | Joker             | 1.83 | 73   | 300  | 150 | 40  |
| <b>13</b> | D | 2 | Lex Luthor        | 1.88 | 95   | 400  | 110 | 42  |
| <b>14</b> | D | 3 | Doomsday          | 2.47 | 416  | 2000 | 125 | 350 |
| <b>15</b> | D | 4 | Darkseid          | 2.71 | 825  | 2000 | 425 | 200 |
| <b>16</b> | E | 1 | Robin             | 1.72 | 68   | 350  | 615 | 65  |
| <b>17</b> | E | 2 | Acertijo          | 1.72 | 76   | 250  | 60  | 28  |
| <b>18</b> | E | 3 | Pingüino          | 1.58 | 99   | 80   | 176 | 10  |
| <b>19</b> | E | 4 | Bane              | 2.10 | 180  | 650  | 300 | 70  |
| <b>20</b> | F | 1 | Batichica         | 1.76 | 55   | 300  | 513 | 60  |
| <b>21</b> | F | 2 | Gorilla Grodd     | 2.51 | 365  | 1200 | 301 | 60  |
| <b>22</b> | F | 3 | Black Adam        | 1.83 | 114  | 1800 | 500 | 300 |
| <b>23</b> | F | 4 | Brainiac          | 1.98 | 285  | 800  | 215 | 55  |
| <b>24</b> | G | 1 | Dr. Fate          | 2.00 | 100  | 600  | 750 | 100 |
| <b>25</b> | G | 2 | Tornado Rojo      | 1.90 | 295  | 800  | 716 | 200 |
| <b>26</b> | G | 3 | Mujer Halcon      | 1.77 | 65   | 650  | 540 | 100 |
| <b>27</b> | G | 4 | Siniestro         | 1.83 | 93   | 830  | 354 | 170 |
| <b>28</b> | H | 1 | Canario Negro     | 1.70 | 55   | 400  | 616 | 65  |
| <b>29</b> | H | 2 | Zatana            | 1.73 | 62   | 350  | 725 | 53  |
| <b>30</b> | H | 3 | Atomo             | 1.82 | 82   | 340  | 680 | 50  |
| <b>31</b> | H | 4 | Cyborg            | 2.00 | 1335 | 700  | 716 | 85  |



Para construir el ranking de cada variable y el ranking general, normalizamos todas las variables a través de los valores máximos y mínimos. De esta forma cada variable tiene un rango que va de 0.0 para la peor carta en esa variable a 1.0 para la mejor.

```
In [184]: f_norm = copy.copy(f)
f_norm.ALTURA = (f.ALTURA - np.min(f.ALTURA))/(np.max(f.ALTURA - np.min(f.ALTURA)))
f_norm.PESO = (f.PESO - np.min(f.PESO))/(np.max(f.PESO - np.min(f.PESO)))
f_norm.FUERZA = (f.FUERZA - np.min(f.FUERZA))/(np.max(f.FUERZA - np.min(f.FUERZA)))
f_norm.PELEAS = (f.PELEAS - np.min(f.PELEAS))/(np.max(f.PELEAS - np.min(f.PELEAS)))
f_norm.VELOCIDAD = (f.VELOCIDAD - np.min(f.VELOCIDAD))/(np.max(f.VELOCIDAD - np.min(f.VELOCIDAD)))
```

```
In [185]: fig = plt.figure(figsize=(12, 10), dpi= 80, facecolor='w', edgecolor='k')
plt.subplot(3,2,1)
plt.hist(f_norm.ALTURA)
plt.xlabel('Altura Normalizada')
plt.subplot(3,2,2)
plt.hist(f_norm.PESO)
plt.xlabel('Peso Normalizado')
plt.subplot(3,2,3)
plt.hist(f_norm.FUERZA)
plt.xlabel('Fuerza Normalizada')
plt.subplot(3,2,4)
plt.hist(f_norm.PELEAS)
plt.xlabel('Peleas Normalizada')
plt.subplot(3,2,5)
plt.hist(f_norm.VELOCIDAD)
plt.xlabel('Velocidad Normalizada')
plt.show()
```



Se puede observar que en cada variable hay algunas cartas con características superiores a las demás. Se puede obtener una idea general con estos rankings? Por ejemplo, que pasaría si construimos una nueva variable llamada SUMA que sea la suma de los 5 rankings.

```
In [186]: SUMA = f_norm.ALTURA + f_norm.PESO + f_norm.FUERZA + f_norm.PELEAS + f_norm.VELOCIDAD
```

```
In [187]: f_norm['SUMA'] = SUMA
```

```
In [188]: f_norm.sort_values(by= 'SUMA', ascending= False)
```

Out[188]:

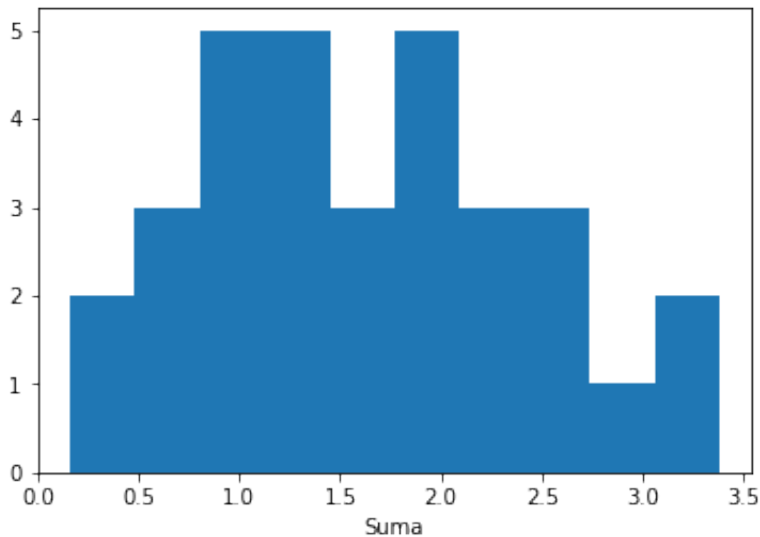
|    | LETRA | NUMERO | NOMBRE   | ALTURA   | PESO     | FUERZA   | PELEAS   | VELOCIDAD | SUM    |
|----|-------|--------|----------|----------|----------|----------|----------|-----------|--------|
| 15 | D     | 4      | Darkseid | 1.000000 | 0.601562 | 1.000000 | 0.392473 | 0.387755  | 3.3817 |
| 0  | A     | 1      | Superman | 0.300885 | 0.036719 | 1.000000 | 1.000000 | 0.795918  | 3.1335 |
| 14 | D     | 3      | Doomsday | 0.787611 | 0.282031 | 1.000000 | 0.069892 | 0.693878  | 2.8334 |

|           |   |   |                      |          |          |          |          |          |        |
|-----------|---|---|----------------------|----------|----------|----------|----------|----------|--------|
| <b>1</b>  | A | 2 | Shazam               | 0.327434 | 0.042969 | 1.000000 | 0.688172 | 0.591837 | 2.6504 |
| <b>31</b> | H | 4 | Cyborg               | 0.371681 | 1.000000 | 0.322917 | 0.705376 | 0.153061 | 2.5530 |
| <b>3</b>  | A | 4 | Martian<br>Manhunter | 0.407080 | 0.063281 | 0.947917 | 0.741935 | 0.387755 | 2.5479 |
| <b>8</b>  | C | 1 | Mujer<br>Maravilla   | 0.265487 | 0.026562 | 0.895833 | 0.892473 | 0.183673 | 2.2640 |
| <b>22</b> | F | 3 | Black<br>Adam        | 0.221239 | 0.046094 | 0.895833 | 0.473118 | 0.591837 | 2.2281 |
| <b>5</b>  | B | 2 | Flash                | 0.221239 | 0.026562 | 0.192708 | 0.688172 | 1.000000 | 2.1286 |
| <b>7</b>  | B | 4 | Lobo                 | 0.415929 | 0.065625 | 0.869792 | 0.500000 | 0.183673 | 2.0350 |
| <b>21</b> | F | 2 | Gorilla<br>Grodd     | 0.823009 | 0.242188 | 0.583333 | 0.259140 | 0.102041 | 2.0097 |
| <b>25</b> | G | 2 | Tornado<br>Rojo      | 0.283186 | 0.187500 | 0.375000 | 0.705376 | 0.387755 | 1.9388 |
| <b>10</b> | C | 3 | Linterna<br>Verde    | 0.221239 | 0.028125 | 0.583333 | 0.720430 | 0.326531 | 1.8796 |
| <b>9</b>  | C | 2 | Hombre<br>Halcon     | 0.327434 | 0.046094 | 0.531250 | 0.741935 | 0.183673 | 1.8303 |
| <b>2</b>  | A | 3 | Aquaman              | 0.247788 | 0.028906 | 0.479167 | 0.715054 | 0.204082 | 1.6749 |
| <b>24</b> | G | 1 | Dr. Fate             | 0.371681 | 0.035156 | 0.270833 | 0.741935 | 0.183673 | 1.6032 |
| <b>4</b>  | B | 1 | Batman               | 0.274336 | 0.031250 | 0.218750 | 0.913978 | 0.122449 | 1.5607 |
| <b>6</b>  | B | 3 | Flecha<br>Verde      | 0.274336 | 0.026562 | 0.192708 | 0.774194 | 0.091837 | 1.3596 |
| <b>27</b> | G | 4 | Siniestro            | 0.221239 | 0.029687 | 0.390625 | 0.316129 | 0.326531 | 1.2842 |
| <b>19</b> | E | 4 | Bane                 | 0.460177 | 0.097656 | 0.296875 | 0.258065 | 0.122449 | 1.2352 |
| <b>26</b> | G | 3 | Mujer<br>Halcon      | 0.168142 | 0.007812 | 0.296875 | 0.516129 | 0.183673 | 1.1726 |
| <b>23</b> | F | 4 | Brainiac             | 0.353982 | 0.179688 | 0.375000 | 0.166667 | 0.091837 | 1.1671 |
| <b>30</b> | H | 3 | Atomo                | 0.212389 | 0.021094 | 0.135417 | 0.666667 | 0.081633 | 1.1171 |
| <b>29</b> | H | 2 | Zatana               | 0.132743 | 0.005469 | 0.140625 | 0.715054 | 0.087755 | 1.0816 |
| <b>16</b> | E | 1 | Robin                | 0.123894 | 0.010156 | 0.140625 | 0.596774 | 0.112245 | 0.9836 |
| <b>28</b> | H | 1 | Canario<br>Negro     | 0.106195 | 0.000000 | 0.166667 | 0.597849 | 0.112245 | 0.9829 |
| <b>20</b> | F | 1 | Batichica            | 0.159292 | 0.000000 | 0.114583 | 0.487097 | 0.102041 | 0.8630 |
| <b>11</b> | C | 4 | Gatubela             | 0.194690 | 0.004687 | 0.114583 | 0.204301 | 0.102041 | 0.6203 |
| <b>13</b> | D | 2 | Lex Luthor           | 0.265487 | 0.031250 | 0.166667 | 0.053763 | 0.065306 | 0.5824 |

|    |   |   |          |          |          |          |          |          |        |
|----|---|---|----------|----------|----------|----------|----------|----------|--------|
| 12 | D | 1 | Joker    | 0.221239 | 0.014063 | 0.114583 | 0.096774 | 0.061224 | 0.5078 |
| 17 | E | 2 | Acertijo | 0.123894 | 0.016406 | 0.088542 | 0.000000 | 0.036735 | 0.2655 |
| 18 | E | 3 | Pingüino | 0.000000 | 0.034375 | 0.000000 | 0.124731 | 0.000000 | 0.1591 |

Con este nuevo ranking podemos ordenar las cartas. ¿Cómo es la distribución de SUMA?

```
In [189]: plt.hist(f_norm.SUMA)
plt.xlabel('Suma')
plt.show()
```

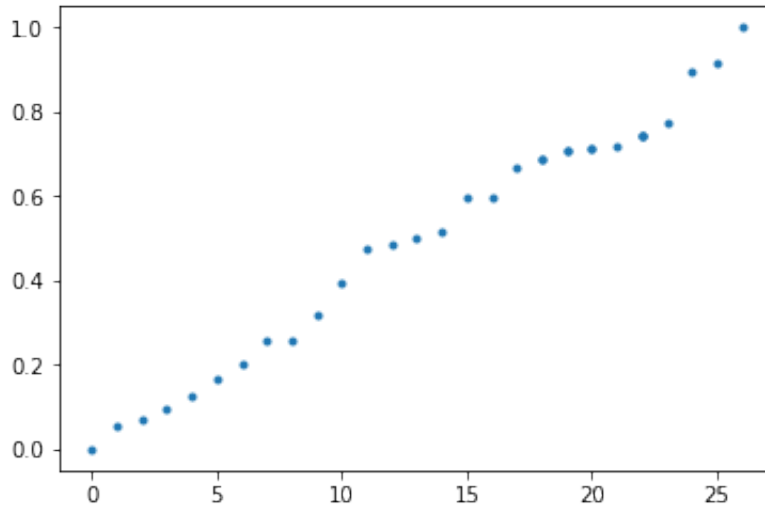


El problema de usar suma como indicador de la fuerza total de una de las cartas es que en un combate en particular se utiliza una única de las variables. Más aun, no importa el valor de la misma, sólo cual es mayor. Dicho esto, en realidad sólo importa la posición en el ranking normalizado y no el valor. Por ejemplo, es claro que Superman y Darkseid son cartas poderosas porque en dos de las categorías tienen los valores más altos y por lo tanto le ganan a todas las cartas en 2 de 5 variables. Por otro lado, el pingüino es la peor carta porque en dos de las 5 variables es la peor carta.

Una cosa que podríamos hacer es en vez de sumar el valor normalizado, sumar la posición del ranking. Por ejemplo, si ordenamos peleas:



```
In [190]: peleas = list(f_norm['PELEAS'])
peleas_ranking = list(map({j: i for i, j in enumerate(sorted(set(peleas)))).get, peleas))
plt.plot(peleas_ranking, peleas, '.')
plt.show()
```



```
In [191]: altura = list(f_norm['ALTURA'])
altura_ranking = list(map({j: i for i, j in enumerate(sorted(set(altura)))).get, altura))
peso = list(f_norm['PESO'])
peso_ranking = list(map({j: i for i, j in enumerate(sorted(set(peso)))).get, peso))
fuerza = list(f_norm['FUERZA'])
fuerza_ranking = list(map({j: i for i, j in enumerate(sorted(set(fuerza)))).get, fuerza))
velocidad = list(f_norm['VELOCIDAD'])
velocidad_ranking = list(map({j: i for i, j in enumerate(sorted(set(velocidad)))).get, velocidad))
ranking = np.array(altura_ranking) + np.array(peso_ranking) + np.array(fuerza_ranking) + np.array(peleas_ranking) + np.array(velocidad_ranking)

f_norm['RANKING'] = ranking
```

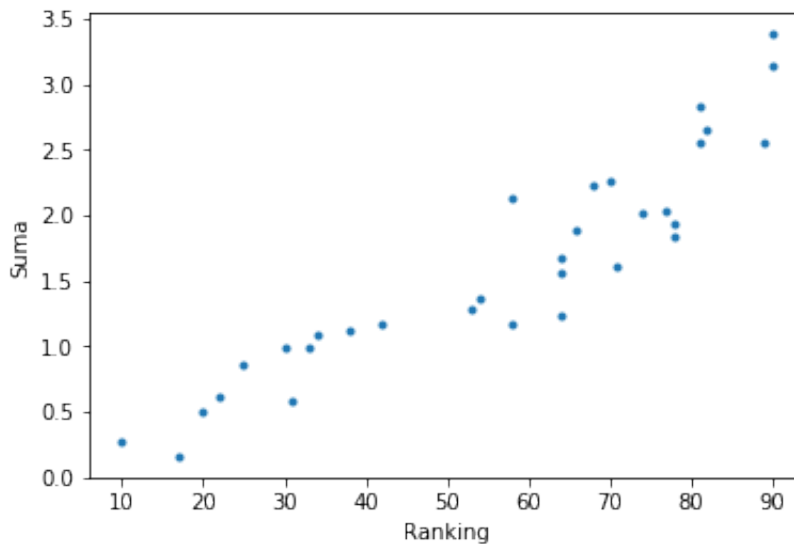
Usando la nueva variable RANKING, podemos ver como se relaciona con SUMA. Nuevamente, cuanto mayor es la variable RANKING, mayor es la fuerza de la carta.

```
In [192]: plt.plot(f_norm.RANKING, f_norm.SUMA, '.')
```

```
plt.xlabel('Ranking')
```

```
plt.ylabel('Suma')
```

```
plt.show()
```



Otra forma de evaluar la fuerza de una carta, es simular el combate de las mismas. Para ello, construimos una función `fight`, que dada la base de datos de las cartas y dos cartas al azar, elige al azar una de las variables de las cartas y devuelve `[1.0, 0.0]` si la primera carta es la ganadora, `[0.0, 1.0]` si es la segunda o `[0.5, 0.5]` si hay empate.

```
In [193]: def fight(database, card1, card2):
```

```
    variable = random.choice(['ALTURA', 'PESO', 'FUERZA', 'PELEAS', 'VELOC
```

```
    IDAD'])
```

```
    v1 = database.loc[card1, variable]
```

```
    v2 = database.loc[card2, variable]
```

```
    if v1 > v2:
```

```
        return [1, 0]
```

```
    elif v1 < v2:
```

```
        return [0, 1]
```

```
    else:
```

```
        return [0.5, 0.5]
```

```
    return v1, v2
```

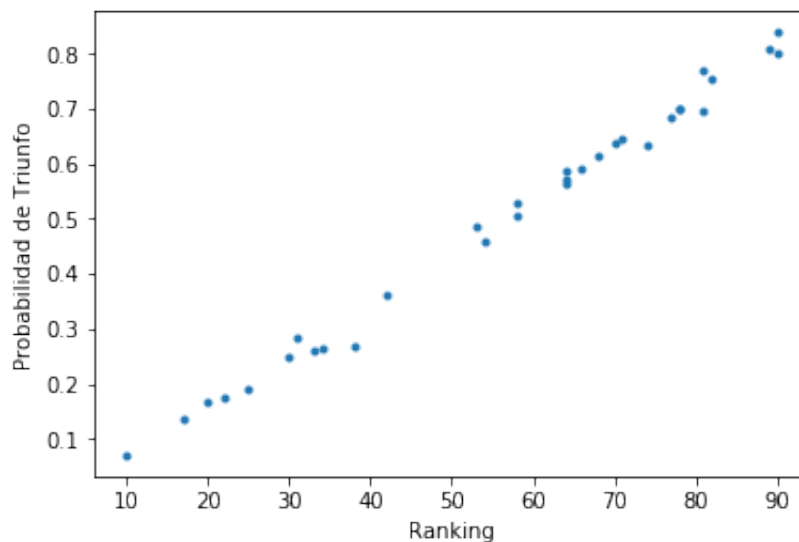
Para simular los combates y que haya igual combates para todas las cartas, dividimos el mazo en dos en forma aleatoria y repetimos los 16 posibles combates `reps = 10000` veces. Cada combate modifica el valor de una variable `POINTS` en la base de datos. Al finalizar las repeticiones, divide `POINTS` por la cantidad de combates realizados, con lo que `POINTS` pasa a ser una especie de probabilidad de triunfo de cada carta.

```
In [194]: f_norm['POINTS'] = 0.0
reps = 10000
for i in range(reps):
    cartas = np.arange(32)
    random.shuffle(cartas)
    partida = cartas.reshape(16,2)
    for j in range(16):
        v1, v2 = fight(f_norm,partida[j][0],partida[j][1])
        f_norm.loc[partida[j][0],'POINTS'] = f_norm.loc[partida[j][0],
'POINTS'] + v1
        f_norm.loc[partida[j][1],'POINTS'] = f_norm.loc[partida[j][1],
'POINTS'] + v2

f_norm['POINTS'] = f_norm['POINTS']/reps
```

Podemos ver si nuestra función de ranking refleja lo obtenido en las simulaciones

```
In [195]: plt.plot(f_norm.RANKING, f_norm.POINTS, '.')
plt.xlabel('Ranking')
plt.ylabel('Probabilidad de Triunfo')
plt.show()
```



Como puede observarse, la función de ranking describe bastante bien la probabilidad de triunfo de cada carta.

```
In [196]: f_norm.sort_values(by = 'POINTS',ascending= False)
```

Out[196]:

| LETRA | NUMERO | NOMBRE | ALTURA | PESO | FUERZA | PELEAS | VELOCIDAD | SUM |
|-------|--------|--------|--------|------|--------|--------|-----------|-----|
|-------|--------|--------|--------|------|--------|--------|-----------|-----|

|    |   |   |                      |          |          |          |          |          |        |
|----|---|---|----------------------|----------|----------|----------|----------|----------|--------|
| 0  | A | 1 | Superman             | 0.300885 | 0.036719 | 1.000000 | 1.000000 | 0.795918 | 3.1335 |
| 3  | A | 4 | Martian<br>Manhunter | 0.407080 | 0.063281 | 0.947917 | 0.741935 | 0.387755 | 2.5479 |
| 15 | D | 4 | Darkseid             | 1.000000 | 0.601562 | 1.000000 | 0.392473 | 0.387755 | 3.3817 |
| 14 | D | 3 | Doomsday             | 0.787611 | 0.282031 | 1.000000 | 0.069892 | 0.693878 | 2.8334 |
| 1  | A | 2 | Shazam               | 0.327434 | 0.042969 | 1.000000 | 0.688172 | 0.591837 | 2.6504 |
| 25 | G | 2 | Tornado<br>Rojo      | 0.283186 | 0.187500 | 0.375000 | 0.705376 | 0.387755 | 1.9388 |
| 9  | C | 2 | Hombre<br>Halcon     | 0.327434 | 0.046094 | 0.531250 | 0.741935 | 0.183673 | 1.8303 |
| 31 | H | 4 | Cyborg               | 0.371681 | 1.000000 | 0.322917 | 0.705376 | 0.153061 | 2.5530 |
| 7  | B | 4 | Lobo                 | 0.415929 | 0.065625 | 0.869792 | 0.500000 | 0.183673 | 2.0350 |
| 24 | G | 1 | Dr. Fate             | 0.371681 | 0.035156 | 0.270833 | 0.741935 | 0.183673 | 1.6032 |
| 8  | C | 1 | Mujer<br>Maravilla   | 0.265487 | 0.026562 | 0.895833 | 0.892473 | 0.183673 | 2.2640 |
| 21 | F | 2 | Gorilla<br>Grodd     | 0.823009 | 0.242188 | 0.583333 | 0.259140 | 0.102041 | 2.0097 |
| 22 | F | 3 | Black<br>Adam        | 0.221239 | 0.046094 | 0.895833 | 0.473118 | 0.591837 | 2.2281 |
| 10 | C | 3 | Linterna<br>Verde    | 0.221239 | 0.028125 | 0.583333 | 0.720430 | 0.326531 | 1.8796 |
| 2  | A | 3 | Aquaman              | 0.247788 | 0.028906 | 0.479167 | 0.715054 | 0.204082 | 1.6749 |
| 19 | E | 4 | Bane                 | 0.460177 | 0.097656 | 0.296875 | 0.258065 | 0.122449 | 1.2352 |
| 4  | B | 1 | Batman               | 0.274336 | 0.031250 | 0.218750 | 0.913978 | 0.122449 | 1.5607 |
| 5  | B | 2 | Flash                | 0.221239 | 0.026562 | 0.192708 | 0.688172 | 1.000000 | 2.1286 |
| 23 | F | 4 | Brainiac             | 0.353982 | 0.179688 | 0.375000 | 0.166667 | 0.091837 | 1.1671 |
| 27 | G | 4 | Siniestro            | 0.221239 | 0.029687 | 0.390625 | 0.316129 | 0.326531 | 1.2842 |
| 6  | B | 3 | Flecha<br>Verde      | 0.274336 | 0.026562 | 0.192708 | 0.774194 | 0.091837 | 1.3596 |
| 26 | G | 3 | Mujer<br>Halcon      | 0.168142 | 0.007812 | 0.296875 | 0.516129 | 0.183673 | 1.1726 |
| 13 | D | 2 | Lex Luthor           | 0.265487 | 0.031250 | 0.166667 | 0.053763 | 0.065306 | 0.5824 |
| 30 | H | 3 | Atomo                | 0.212389 | 0.021094 | 0.135417 | 0.666667 | 0.081633 | 1.1171 |
| 29 | H | 2 | Zatana               | 0.132743 | 0.005469 | 0.140625 | 0.715054 | 0.087755 | 1.0816 |
| 16 | E | 1 | Robin                | 0.123894 | 0.010156 | 0.140625 | 0.596774 | 0.112245 | 0.9836 |
| 28 | H | 1 | Canario<br>Negro     | 0.106195 | 0.000000 | 0.166667 | 0.597849 | 0.112245 | 0.9829 |

|    |   |   |           |          |          |          |          |          |        |
|----|---|---|-----------|----------|----------|----------|----------|----------|--------|
| 20 | F | 1 | Batichica | 0.159292 | 0.000000 | 0.114583 | 0.487097 | 0.102041 | 0.8630 |
| 11 | C | 4 | Gatubela  | 0.194690 | 0.004687 | 0.114583 | 0.204301 | 0.102041 | 0.6203 |
| 12 | D | 1 | Joker     | 0.221239 | 0.014063 | 0.114583 | 0.096774 | 0.061224 | 0.5078 |
| 18 | E | 3 | Pingüino  | 0.000000 | 0.034375 | 0.000000 | 0.124731 | 0.000000 | 0.1591 |
| 17 | E | 2 | Acertijo  | 0.123894 | 0.016406 | 0.088542 | 0.000000 | 0.036735 | 0.2655 |

Ahora vamos a construir una función para simular un partido entre dos jugadores. Para ello, mezclaremos las cartas y repartiremos la mitad a cada jugador. El jugador que tenga A1 comenzará y elegirá (al azar) una de las variables. Si gana, se lleva ambas cartas y las pone en el fondo de su mazo. El partido dura hasta que uno de los jugadores se queda sin cartas. Si hay empate, se elige un ganador al azar.

```
In [18]: def mezclar(n):
    cartas = np.arange(n)
    random.shuffle(cartas)
    return cartas.reshape(int(n/2),2)
```

Dado que el juego es completamente al azar, vamos a evaluar la ventaja de tener la carta A1. En este caso, el jugador con A1 se convierte en el jugador 1.

```
In [161]: def partido(empieza_A1):
    mazos = mezclar(32)

    if empieza_A1:
        if 0 in mazos[:,0]:
            mazo1 = list(mazos[:,0])
            mazo2 = list(mazos[:,1])
        else:
            mazo1 = list(mazos[:,1])
            mazo2 = list(mazos[:,0])
    else:
        if random.random() < 0.5:
            mazo1 = list(mazos[:,0])
            mazo2 = list(mazos[:,1])
        else:
            mazo1 = list(mazos[:,1])
            mazo2 = list(mazos[:,0])

    fin_partida = len(mazo1)*len(mazo2)

    while fin_partida != 0:
        carta1 = mazo1.pop()
        carta2 = mazo2.pop()
```

```

    v1, v2 = fight(f_norm, carta1, carta2)

    if v1 > v2:
        mazo1 = [carta1, carta2] + mazo1
    elif v1 < v2:
        mazo2 = [carta1, carta2] + mazo2
    else:
        if random.random() < 0.5:
            mazo1 = [carta1, carta2] + mazo1
        else:
            mazo2 = [carta1, carta2] + mazo2

    fin_partida = len(mazo1)*len(mazo2)
    if fin_partida == 0:
        break
    if len(mazo1) == 0:
        return [0,1]
    else:
        return [1,0]

rep = 10000
juegos = []
for i in range(rep):
    juegos.append(partido(True))
print('Empieza el jugador 1 y tiene A1:', list(np.sum(np.array(juegos), axis=0)/rep))

juegos = []
for i in range(rep):
    juegos.append(partido(False))
print('Empieza el jugador 1:', list(np.sum(np.array(juegos), axis=0)/rep))

```

Empieza el jugador 1 y tiene A1: [0.5535, 0.4465]

Empieza el jugador 1: [0.4965, 0.5035]

Dado que los jugadores eligen aleatoriamente la variable, empezar no debería dar una ventaja. Al parecer, tener A1 sí da una pequeña ventaja jugando aleatoriamente.

Construyamos un jugador que juegue óptimamente, en vez de al azar. Para ello, este jugador sabe exactamente cuál de las 5 variables es la mejor para cada carta.



```
In [197]: variables = ['ALTURA', 'PESO', 'FUERZA', 'PELEAS', 'VELOCIDAD']
MEJOR = []
for i in range(32):
    tmp = np.array(f_norm.loc[i][3:8])
    MEJOR.append(variables[np.argmax(tmp)])

#Superman es el mejor absoluto en peleas y Darkseid en altura
MEJOR[0] = 'PELEAS'
MEJOR[15] = 'ALTURA'
f_norm['MEJOR'] = MEJOR
```

```
In [198]: f_norm.sort_values(by = 'POINTS', ascending= False)
```

Out[198]:

|    | LETRA | NUMERO | NOMBRE               | ALTURA   | PESO     | FUERZA   | PELEAS   | VELOCIDAD | SUM    |
|----|-------|--------|----------------------|----------|----------|----------|----------|-----------|--------|
| 0  | A     | 1      | Superman             | 0.300885 | 0.036719 | 1.000000 | 1.000000 | 0.795918  | 3.1335 |
| 3  | A     | 4      | Martian<br>Manhunter | 0.407080 | 0.063281 | 0.947917 | 0.741935 | 0.387755  | 2.5479 |
| 15 | D     | 4      | Darkseid             | 1.000000 | 0.601562 | 1.000000 | 0.392473 | 0.387755  | 3.3817 |
| 14 | D     | 3      | Doomsday             | 0.787611 | 0.282031 | 1.000000 | 0.069892 | 0.693878  | 2.8334 |
| 1  | A     | 2      | Shazam               | 0.327434 | 0.042969 | 1.000000 | 0.688172 | 0.591837  | 2.6504 |
| 25 | G     | 2      | Tornado<br>Rojo      | 0.283186 | 0.187500 | 0.375000 | 0.705376 | 0.387755  | 1.9388 |
| 9  | C     | 2      | Hombre<br>Halcon     | 0.327434 | 0.046094 | 0.531250 | 0.741935 | 0.183673  | 1.8303 |
| 31 | H     | 4      | Cyborg               | 0.371681 | 1.000000 | 0.322917 | 0.705376 | 0.153061  | 2.5530 |
| 7  | B     | 4      | Lobo                 | 0.415929 | 0.065625 | 0.869792 | 0.500000 | 0.183673  | 2.0350 |
| 24 | G     | 1      | Dr. Fate             | 0.371681 | 0.035156 | 0.270833 | 0.741935 | 0.183673  | 1.6032 |
| 8  | C     | 1      | Mujer<br>Maravilla   | 0.265487 | 0.026562 | 0.895833 | 0.892473 | 0.183673  | 2.2640 |
| 21 | F     | 2      | Gorilla<br>Grodd     | 0.823009 | 0.242188 | 0.583333 | 0.259140 | 0.102041  | 2.0097 |
| 22 | F     | 3      | Black<br>Adam        | 0.221239 | 0.046094 | 0.895833 | 0.473118 | 0.591837  | 2.2281 |
| 10 | C     | 3      | Linterna<br>Verde    | 0.221239 | 0.028125 | 0.583333 | 0.720430 | 0.326531  | 1.8796 |
| 2  | A     | 3      | Aquaman              | 0.247788 | 0.028906 | 0.479167 | 0.715054 | 0.204082  | 1.6749 |
| 19 | E     | 4      | Bane                 | 0.460177 | 0.097656 | 0.296875 | 0.258065 | 0.122449  | 1.2352 |
| 4  | B     | 1      | Batman               | 0.274336 | 0.031250 | 0.218750 | 0.913978 | 0.122449  | 1.5607 |
| 5  | B     | 2      | Flash                | 0.221239 | 0.026562 | 0.192708 | 0.688172 | 1.000000  | 2.1286 |

|    |   |   |               |          |          |          |          |          |        |
|----|---|---|---------------|----------|----------|----------|----------|----------|--------|
| 23 | F | 4 | Brainiac      | 0.353982 | 0.179688 | 0.375000 | 0.166667 | 0.091837 | 1.1671 |
| 27 | G | 4 | Siniestro     | 0.221239 | 0.029687 | 0.390625 | 0.316129 | 0.326531 | 1.2842 |
| 6  | B | 3 | Flecha Verde  | 0.274336 | 0.026562 | 0.192708 | 0.774194 | 0.091837 | 1.3596 |
| 26 | G | 3 | Mujer Halcon  | 0.168142 | 0.007812 | 0.296875 | 0.516129 | 0.183673 | 1.1726 |
| 13 | D | 2 | Lex Luthor    | 0.265487 | 0.031250 | 0.166667 | 0.053763 | 0.065306 | 0.5824 |
| 30 | H | 3 | Atomo         | 0.212389 | 0.021094 | 0.135417 | 0.666667 | 0.081633 | 1.1171 |
| 29 | H | 2 | Zatana        | 0.132743 | 0.005469 | 0.140625 | 0.715054 | 0.087755 | 1.0816 |
| 16 | E | 1 | Robin         | 0.123894 | 0.010156 | 0.140625 | 0.596774 | 0.112245 | 0.9836 |
| 28 | H | 1 | Canario Negro | 0.106195 | 0.000000 | 0.166667 | 0.597849 | 0.112245 | 0.9829 |
| 20 | F | 1 | Batichica     | 0.159292 | 0.000000 | 0.114583 | 0.487097 | 0.102041 | 0.8630 |
| 11 | C | 4 | Gatubela      | 0.194690 | 0.004687 | 0.114583 | 0.204301 | 0.102041 | 0.6203 |
| 12 | D | 1 | Joker         | 0.221239 | 0.014063 | 0.114583 | 0.096774 | 0.061224 | 0.5078 |
| 18 | E | 3 | Pingüino      | 0.000000 | 0.034375 | 0.000000 | 0.124731 | 0.000000 | 0.1591 |
| 17 | E | 2 | Acertijo      | 0.123894 | 0.016406 | 0.088542 | 0.000000 | 0.036735 | 0.2655 |

```
In [129]: def fight_optimo(database, card1, card2, jugador):
    if jugador == 0:
        variable = database.loc[card1, 'MEJOR']
    else:
        variable = database.loc[card2, 'MEJOR']

    v1 = database.loc[card1, variable]
    v2 = database.loc[card2, variable]
    if v1 > v2:
        return [1, 0]
    elif v1 < v2:
        return [0, 1]
    else:
        return [0.5, 0.5]
    return v1, v2
```

```
In [163]: def partido_optimo_vs_azar():
    mazos = mezclar(32)

    if 0 in mazos[:, 0]:
        mazo1 = list(mazos[:, 0])
        mazo2 = list(mazos[:, 1])
    else:
        mazo1 = list(mazos[:, 1])
```

```

        mazo2 = list(mazos[:,0])

    fin_partida = len(mazo1)*len(mazo2)

    #Empieza el jugador 1, que juega optimamente
    turno = 0
    while fin_partida != 0:
        carta1 = mazo1.pop()
        carta2 = mazo2.pop()

        if turno == 0:
            v1, v2 = fight_optimo(f_norm, carta1, carta2, turno)
        else:
            v1, v2 = fight(f_norm, carta1, carta2)

        if v1 > v2:
            mazo1 = [carta1, carta2] + mazo1
            turno = 0
        elif v1 < v2:
            mazo2 = [carta1, carta2] + mazo2
            turno = 1
        else:
            if random.random() < 0.5:
                mazo1 = [carta1, carta2] + mazo1
                turno = 0
            else:
                mazo2 = [carta1, carta2] + mazo2
                turno = 1
        fin_partida = len(mazo1)*len(mazo2)
        if fin_partida == 0:
            break
    if len(mazo1) == 0:
        return [0,1]
    else:
        return [1,0]

rep = 10000
juegos = []
for i in range(rep):
    juegos.append(partido_optimo_vs_azar())
print('Empieza el jugador 1 con A1 optimamente :', list(np.sum(np.array(juegos), axis=0)/rep))

```

Empieza el jugador 1 con A1 optimamente : [0.9993, 0.0007]

Si el jugador que empieza juega óptimamente, gana el 99.9% de las veces.

In [164]: `def partido_optimo_vs_azar2():`

```

mazos = mezclar(32)

if 0 in mazos[:,0]:
    mazo1 = list(mazos[:,0])
    mazo2 = list(mazos[:,1])
else:
    mazo1 = list(mazos[:,1])
    mazo2 = list(mazos[:,0])

fin_partida = len(mazo1)*len(mazo2)

#Empieza el jugador 2, que juega al azar
turno = 0

while fin_partida != 0:
    carta1 = mazo1.pop()
    carta2 = mazo2.pop()

    if turno == 0:
        v1, v2 = fight(f_norm, carta1, carta2)
    else:
        v1, v2 = fight_optimo(f_norm, carta1, carta2, turno)

    if v1 > v2:
        mazo1 = [carta1, carta2] + mazo1
        turno = 0
    elif v1 < v2:
        mazo2 = [carta1, carta2] + mazo2
        turno = 1
    else:
        if random.random() < 0.5:
            mazo1 = [carta1, carta2] + mazo1
            turno = 0
        else:
            mazo2 = [carta1, carta2] + mazo2
            turno = 1
    fin_partida = len(mazo1)*len(mazo2)
    if fin_partida == 0:
        break
if len(mazo1) == 0:
    return [0,1]
else:
    return [1,0]

rep = 10000
juegos = []
for i in range(rep):
    juegos.append(partido_optimo_vs_azar2())
print('Empieza el jugador 1 con A1 al azar :', list(np.sum(np.array(jue
gos), axis=0)/rep))

```

Empieza el jugador 1 con A1 al azar : [0.0042, 0.9958]

La ventaja de tener A1 es mínima, si el jugador juega al azar contra otro que juega óptimamente.

```
In [169]: def partido_optimo_vs_optimo(empieza_A1):
    mazos = mezclar(32)

    if empieza_A1:
        if 0 in mazos[:,0]:
            mazo1 = list(mazos[:,0])
            mazo2 = list(mazos[:,1])
        else:
            mazo1 = list(mazos[:,1])
            mazo2 = list(mazos[:,0])
    else:
        if random.random() < 0.5:
            mazo1 = list(mazos[:,0])
            mazo2 = list(mazos[:,1])
        else:
            mazo1 = list(mazos[:,1])
            mazo2 = list(mazos[:,0])

    fin_partida = len(mazo1)*len(mazo2)

    #Empieza el jugador 1, que juega optimamente
    turno = 0
    while fin_partida != 0:
        carta1 = mazo1.pop()
        carta2 = mazo2.pop()

        v1, v2 = fight_optimo(f_norm, carta1, carta2, turno)

        if v1 > v2:
            mazo1 = [carta1, carta2] + mazo1
            turno = 0
        elif v1 < v2:
            mazo2 = [carta1, carta2] + mazo2
            turno = 1
        else:
            if random.random() < 0.5:
                mazo1 = [carta1, carta2] + mazo1
                turno = 0
            else:
                mazo2 = [carta1, carta2] + mazo2
                turno = 1
        fin_partida = len(mazo1)*len(mazo2)
    if fin_partida == 0:
        break
```

```

    if len(mazo1) == 0:
        return [0,1]
    else:
        return [1,0]

rep = 10000
juegos = []
for i in range(rep):
    juegos.append(partido_optimo_vs_optimo(True))
print('Empieza el jugador 1 y tiene A1: ', list(np.sum(np.array(juegos), axis=0)/rep))

juegos = []
for i in range(rep):
    juegos.append(partido_optimo_vs_optimo(False))
print('Empieza el Jugador 1: ', list(np.sum(np.array(juegos), axis=0)/rep))

```

Empieza el jugador 1 y tiene A1: [0.6952, 0.3048]

Empieza el Jugador 1: [0.5623, 0.4377]

Si ambos juegan óptimamente, el jugador que tiene la carta A1 tiene ventaja (porque es la mejor y porque empieza). Jugando óptimamente, empezar otorga una ventaja. Si además se tiene la mejor carta, la ventaja es mayor.

## Matrices de victorias por variable

Otra forma de evaluar cada carta es determinando la cantidad de cartas a las que les gana, pierde o empata en cada variable.



```
In [316]: def victorias_empates_derrotas(database,card,variable):
    v = 0
    e = 0
    d = 0
    for i in range(32):
        if database.iloc[i][variable] != database.iloc[card][variable]:
            if database.iloc[i][variable] < database.iloc[card][variable]:
                v += 1
            elif database.iloc[i][variable] > database.iloc[card][variable]:
                d += 1
            else:
                e += 1
    return [v,e,d]

def indice_variable(ved):
    return (ved[0] - ved[1] + 0.5*ved[2])/31
```

```
In [319]: indice_altura = []
indice_peso = []
indice_fuerza = []
indice_peleas = []
indice_velocidad = []
for i in range(32):
    indice_altura.append(indice_variable(victorias_empates_derrotas(f_norm,i, 'ALTURA'))))
    indice_peso.append(indice_variable(victorias_empates_derrotas(f_norm,i, 'PESO'))))
    indice_fuerza.append(indice_variable(victorias_empates_derrotas(f_norm,i, 'FUERZA'))))
    indice_peleas.append(indice_variable(victorias_empates_derrotas(f_norm,i, 'PELEAS'))))
    indice_velocidad.append(indice_variable(victorias_empates_derrotas(f_norm,i, 'VELOCIDAD'))))
```

```
In [320]: f_norm['I_ALTURA'] = indice_altura
f_norm['I_PESO'] = indice_peso
f_norm['I_FUERZA'] = indice_fuerza
f_norm['I_PELEAS'] = indice_peleas
f_norm['I_VELOCIDAD'] = indice_velocidad
```

```
In [322]: f_norm.sort_values(by = 'I_PELEAS', ascending=False)
```

Out[322]:

| LETRA | NUMERO | NOMBRE | ALTURA | PESO | FUERZA | PELEAS | VELOCIDAD | SUM |
|-------|--------|--------|--------|------|--------|--------|-----------|-----|
|-------|--------|--------|--------|------|--------|--------|-----------|-----|

|    |   |   |                   |          |          |          |          |          |        |
|----|---|---|-------------------|----------|----------|----------|----------|----------|--------|
| 0  | A | 1 | Superman          | 0.300885 | 0.036719 | 1.000000 | 1.000000 | 0.795918 | 3.1335 |
| 4  | B | 1 | Batman            | 0.274336 | 0.031250 | 0.218750 | 0.913978 | 0.122449 | 1.5607 |
| 8  | C | 1 | Mujer Maravilla   | 0.265487 | 0.026562 | 0.895833 | 0.892473 | 0.183673 | 2.2640 |
| 6  | B | 3 | Flecha Verde      | 0.274336 | 0.026562 | 0.192708 | 0.774194 | 0.091837 | 1.3596 |
| 10 | C | 3 | Linterna Verde    | 0.221239 | 0.028125 | 0.583333 | 0.720430 | 0.326531 | 1.8796 |
| 24 | G | 1 | Dr. Fate          | 0.371681 | 0.035156 | 0.270833 | 0.741935 | 0.183673 | 1.6032 |
| 3  | A | 4 | Martian Manhunter | 0.407080 | 0.063281 | 0.947917 | 0.741935 | 0.387755 | 2.5479 |
| 9  | C | 2 | Hombre Halcon     | 0.327434 | 0.046094 | 0.531250 | 0.741935 | 0.183673 | 1.8303 |
| 2  | A | 3 | Aquaman           | 0.247788 | 0.028906 | 0.479167 | 0.715054 | 0.204082 | 1.6749 |
| 29 | H | 2 | Zatana            | 0.132743 | 0.005469 | 0.140625 | 0.715054 | 0.087755 | 1.0816 |
| 25 | G | 2 | Tornado Rojo      | 0.283186 | 0.187500 | 0.375000 | 0.705376 | 0.387755 | 1.9388 |
| 31 | H | 4 | Cyborg            | 0.371681 | 1.000000 | 0.322917 | 0.705376 | 0.153061 | 2.5530 |
| 30 | H | 3 | Atomo             | 0.212389 | 0.021094 | 0.135417 | 0.666667 | 0.081633 | 1.1171 |
| 5  | B | 2 | Flash             | 0.221239 | 0.026562 | 0.192708 | 0.688172 | 1.000000 | 2.1286 |
| 1  | A | 2 | Shazam            | 0.327434 | 0.042969 | 1.000000 | 0.688172 | 0.591837 | 2.6504 |
| 28 | H | 1 | Canario Negro     | 0.106195 | 0.000000 | 0.166667 | 0.597849 | 0.112245 | 0.9829 |
| 16 | E | 1 | Robin             | 0.123894 | 0.010156 | 0.140625 | 0.596774 | 0.112245 | 0.9836 |
| 26 | G | 3 | Mujer Halcon      | 0.168142 | 0.007812 | 0.296875 | 0.516129 | 0.183673 | 1.1726 |
| 7  | B | 4 | Lobo              | 0.415929 | 0.065625 | 0.869792 | 0.500000 | 0.183673 | 2.0350 |
| 20 | F | 1 | Batichica         | 0.159292 | 0.000000 | 0.114583 | 0.487097 | 0.102041 | 0.8630 |
| 22 | F | 3 | Black Adam        | 0.221239 | 0.046094 | 0.895833 | 0.473118 | 0.591837 | 2.2281 |
| 15 | D | 4 | Darkseid          | 1.000000 | 0.601562 | 1.000000 | 0.392473 | 0.387755 | 3.3817 |
| 27 | G | 4 | Siniestro         | 0.221239 | 0.029687 | 0.390625 | 0.316129 | 0.326531 | 1.2842 |
| 21 | F | 2 | Gorilla Grodd     | 0.823009 | 0.242188 | 0.583333 | 0.259140 | 0.102041 | 2.0097 |
| 19 | E | 4 | Bane              | 0.460177 | 0.097656 | 0.296875 | 0.258065 | 0.122449 | 1.2352 |
| 11 | C | 4 | Gatubela          | 0.194690 | 0.004687 | 0.114583 | 0.204301 | 0.102041 | 0.6203 |
| 23 | F | 4 | Brainiac          | 0.353982 | 0.179688 | 0.375000 | 0.166667 | 0.091837 | 1.1671 |

|    |   |   |            |          |          |          |          |          |        |
|----|---|---|------------|----------|----------|----------|----------|----------|--------|
| 18 | E | 3 | Pingüino   | 0.000000 | 0.034375 | 0.000000 | 0.124731 | 0.000000 | 0.1591 |
| 12 | D | 1 | Joker      | 0.221239 | 0.014063 | 0.114583 | 0.096774 | 0.061224 | 0.5078 |
| 14 | D | 3 | Doomsday   | 0.787611 | 0.282031 | 1.000000 | 0.069892 | 0.693878 | 2.8334 |
| 13 | D | 2 | Lex Luthor | 0.265487 | 0.031250 | 0.166667 | 0.053763 | 0.065306 | 0.5824 |
| 17 | E | 2 | Acertijo   | 0.123894 | 0.016406 | 0.088542 | 0.000000 | 0.036735 | 0.2655 |

Se puede ver, que el ordenamiento de que variable es la mejor, es mucho más clara con este nuevo índice. Por ejemplo, la mujer maravilla es mejor en peleas que en fuerza, mientras que usando solo la normalización nos daba al revés.

```
In [331]: variables = ['ALTURA', 'PESO', 'FUERZA', 'PELEAS', 'VELOCIDAD']
I_MEJOR = []
for i in range(32):
    tmp = np.array(f_norm.loc[i][['I_ALTURA', 'I_PESO', 'I_FUERZA', 'I_PE
LEAS', 'I_VELOCIDAD']])
    I_MEJOR.append(variables[np.argmax(tmp)])

f_norm['I_MEJOR'] = I_MEJOR
```

```
In [332]: f_norm
```

```
Out[332]:
```

|    | LETRA | NUMERO | NOMBRE               | ALTURA   | PESO     | FUERZA   | PELEAS   | VELOCIDAD | SUM    |
|----|-------|--------|----------------------|----------|----------|----------|----------|-----------|--------|
| 0  | A     | 1      | Superman             | 0.300885 | 0.036719 | 1.000000 | 1.000000 | 0.795918  | 3.1335 |
| 1  | A     | 2      | Shazam               | 0.327434 | 0.042969 | 1.000000 | 0.688172 | 0.591837  | 2.6504 |
| 2  | A     | 3      | Aquaman              | 0.247788 | 0.028906 | 0.479167 | 0.715054 | 0.204082  | 1.6749 |
| 3  | A     | 4      | Martian<br>Manhunter | 0.407080 | 0.063281 | 0.947917 | 0.741935 | 0.387755  | 2.5479 |
| 4  | B     | 1      | Batman               | 0.274336 | 0.031250 | 0.218750 | 0.913978 | 0.122449  | 1.5607 |
| 5  | B     | 2      | Flash                | 0.221239 | 0.026562 | 0.192708 | 0.688172 | 1.000000  | 2.1286 |
| 6  | B     | 3      | Flecha<br>Verde      | 0.274336 | 0.026562 | 0.192708 | 0.774194 | 0.091837  | 1.3596 |
| 7  | B     | 4      | Lobo                 | 0.415929 | 0.065625 | 0.869792 | 0.500000 | 0.183673  | 2.0350 |
| 8  | C     | 1      | Mujer<br>Maravilla   | 0.265487 | 0.026562 | 0.895833 | 0.892473 | 0.183673  | 2.2640 |
| 9  | C     | 2      | Hombre<br>Halcon     | 0.327434 | 0.046094 | 0.531250 | 0.741935 | 0.183673  | 1.8303 |
| 10 | C     | 3      | Linterna<br>Verde    | 0.221239 | 0.028125 | 0.583333 | 0.720430 | 0.326531  | 1.8796 |

|    |   |   |               |          |          |          |          |          |        |
|----|---|---|---------------|----------|----------|----------|----------|----------|--------|
| 11 | C | 4 | Gatubela      | 0.194690 | 0.004687 | 0.114583 | 0.204301 | 0.102041 | 0.6203 |
| 12 | D | 1 | Joker         | 0.221239 | 0.014063 | 0.114583 | 0.096774 | 0.061224 | 0.5078 |
| 13 | D | 2 | Lex Luthor    | 0.265487 | 0.031250 | 0.166667 | 0.053763 | 0.065306 | 0.5824 |
| 14 | D | 3 | Doomsday      | 0.787611 | 0.282031 | 1.000000 | 0.069892 | 0.693878 | 2.8334 |
| 15 | D | 4 | Darkseid      | 1.000000 | 0.601562 | 1.000000 | 0.392473 | 0.387755 | 3.3817 |
| 16 | E | 1 | Robin         | 0.123894 | 0.010156 | 0.140625 | 0.596774 | 0.112245 | 0.9836 |
| 17 | E | 2 | Acertijo      | 0.123894 | 0.016406 | 0.088542 | 0.000000 | 0.036735 | 0.2655 |
| 18 | E | 3 | Pingüino      | 0.000000 | 0.034375 | 0.000000 | 0.124731 | 0.000000 | 0.1591 |
| 19 | E | 4 | Bane          | 0.460177 | 0.097656 | 0.296875 | 0.258065 | 0.122449 | 1.2352 |
| 20 | F | 1 | Batichica     | 0.159292 | 0.000000 | 0.114583 | 0.487097 | 0.102041 | 0.8630 |
| 21 | F | 2 | Gorilla Grodd | 0.823009 | 0.242188 | 0.583333 | 0.259140 | 0.102041 | 2.0097 |
| 22 | F | 3 | Black Adam    | 0.221239 | 0.046094 | 0.895833 | 0.473118 | 0.591837 | 2.2281 |
| 23 | F | 4 | Brainiac      | 0.353982 | 0.179688 | 0.375000 | 0.166667 | 0.091837 | 1.1671 |
| 24 | G | 1 | Dr. Fate      | 0.371681 | 0.035156 | 0.270833 | 0.741935 | 0.183673 | 1.6032 |
| 25 | G | 2 | Tornado Rojo  | 0.283186 | 0.187500 | 0.375000 | 0.705376 | 0.387755 | 1.9388 |
| 26 | G | 3 | Mujer Halcon  | 0.168142 | 0.007812 | 0.296875 | 0.516129 | 0.183673 | 1.1726 |
| 27 | G | 4 | Siniestro     | 0.221239 | 0.029687 | 0.390625 | 0.316129 | 0.326531 | 1.2842 |
| 28 | H | 1 | Canario Negro | 0.106195 | 0.000000 | 0.166667 | 0.597849 | 0.112245 | 0.9829 |
| 29 | H | 2 | Zatana        | 0.132743 | 0.005469 | 0.140625 | 0.715054 | 0.087755 | 1.0816 |
| 30 | H | 3 | Atomo         | 0.212389 | 0.021094 | 0.135417 | 0.666667 | 0.081633 | 1.1171 |
| 31 | H | 4 | Cyborg        | 0.371681 | 1.000000 | 0.322917 | 0.705376 | 0.153061 | 2.5530 |

Simulemos nuevamente. Jugador óptimo vs azar:

```
In [335]: def fight_optimo2(database, card1, card2, jugador):
            if jugador == 0:
                variable = database.loc[card1, 'I_MEJOR']
            else:
                variable = database.loc[card2, 'I_MEJOR']

            v1 = database.loc[card1, variable]
            v2 = database.loc[card2, variable]
```

```

    if v1 > v2:
        return [1,0]
    elif v1 < v2:
        return [0,1]
    else:
        return [0.5, 0.5]
    return v1, v2

def partido_optimo_vs_azar():
    mazos = mezclar(32)

    if 0 in mazos[:,0]:
        mazo1 = list(mazos[:,0])
        mazo2 = list(mazos[:,1])
    else:
        mazo1 = list(mazos[:,1])
        mazo2 = list(mazos[:,0])

    fin_partida = len(mazo1)*len(mazo2)

    #Empieza el jugador 1, que juega optimamente
    turno = 0
    while fin_partida != 0:
        carta1 = mazo1.pop()
        carta2 = mazo2.pop()

        if turno == 0:
            v1, v2 = fight_optimo2(f_norm, carta1, carta2, turno)
        else:
            v1, v2 = fight(f_norm, carta1, carta2)

        if v1 > v2:
            mazo1 = [carta1, carta2] + mazo1
            turno = 0
        elif v1 < v2:
            mazo2 = [carta1, carta2] + mazo2
            turno = 1
        else:
            if random.random() < 0.5:
                mazo1 = [carta1, carta2] + mazo1
                turno = 0
            else:
                mazo2 = [carta1, carta2] + mazo2
                turno = 1
        fin_partida = len(mazo1)*len(mazo2)
        if fin_partida == 0:
            break
    if len(mazo1) == 0:
        return [0,1]
    else:

```

```

        return [1,0]

rep = 10000
juegos = []
for i in range(rep):
    juegos.append(partido_optimo_vs_azar())
print('Empieza el jugador 1 con A1 optimamente :', list(np.sum(np.array(juegos), axis=0)/rep))

```

Empieza el jugador 1 con A1 optimamente : [0.9994, 0.0006]

```

In [336]: def partido_optimo_vs_azar2():
    mazos = mezclar(32)

    if 0 in mazos[:,0]:
        mazo1 = list(mazos[:,0])
        mazo2 = list(mazos[:,1])
    else:
        mazo1 = list(mazos[:,1])
        mazo2 = list(mazos[:,0])

    fin_partida = len(mazo1)*len(mazo2)

    #Empieza el jugador azaroso
    turno = 0

    while fin_partida != 0:
        carta1 = mazo1.pop()
        carta2 = mazo2.pop()

        if turno == 0:
            v1, v2 = fight(f_norm, carta1, carta2)
        else:
            v1, v2 = fight_optimo2(f_norm, carta1, carta2, turno)

        if v1 > v2:
            mazo1 = [carta1, carta2] + mazo1
            turno = 0
        elif v1 < v2:
            mazo2 = [carta1, carta2] + mazo2
            turno = 1
        else:
            if random.random() < 0.5:
                mazo1 = [carta1, carta2] + mazo1
                turno = 0
            else:
                mazo2 = [carta1, carta2] + mazo2
                turno = 1
        fin_partida = len(mazo1)*len(mazo2)
    if fin_partida == 0:

```



```

        break
    if len(mazo1) == 0:
        return [0,1]
    else:
        return [1,0]

rep = 10000
juegos = []
for i in range(rep):
    juegos.append(partido_optimo_vs_azar2())
print('Empieza el jugador 1 con A1 (al azar):',list(np.sum(np.array(ju
egos), axis=0)/rep))

```

Empieza el jugador 1 con A1 al azar : [0.0024, 0.9976]

```

In [337]: def partido_optimo_vs_optimo(empieza_A1):
    mazos = mezclar(32)

    if empieza_A1:
        if 0 in mazos[:,0]:
            mazo1 = list(mazos[:,0])
            mazo2 = list(mazos[:,1])
        else:
            mazo1 = list(mazos[:,1])
            mazo2 = list(mazos[:,0])
    else:
        if random.random() < 0.5:
            mazo1 = list(mazos[:,0])
            mazo2 = list(mazos[:,1])
        else:
            mazo1 = list(mazos[:,1])
            mazo2 = list(mazos[:,0])

    fin_partida = len(mazo1)*len(mazo2)

    #Empieza el jugador 1, que juega optimamente
    turno = 0
    while fin_partida != 0:
        carta1 = mazo1.pop()
        carta2 = mazo2.pop()

        v1, v2 = fight_optimo2(f_norm, carta1, carta2, turno)

        if v1 > v2:
            mazo1 = [carta1, carta2] + mazo1
            turno = 0
        elif v1 < v2:
            mazo2 = [carta1, carta2] + mazo2
            turno = 1
        else:

```

```

        if random.random() < 0.5:
            mazo1 = [carta1, carta2] + mazo1
            turno = 0
        else:
            mazo2 = [carta1, carta2] + mazo2
            turno = 1
    fin_partida = len(mazo1)*len(mazo2)
    if fin_partida == 0:
        break
    if len(mazo1) == 0:
        return [0,1]
    else:
        return [1,0]

rep = 10000
juegos = []
for i in range(rep):
    juegos.append(partido_optimo_vs_optimo(True))
print('Empieza el jugador 1 y tiene A1: ', list(np.sum(np.array(juegos), axis=0)/rep))

juegos = []
for i in range(rep):
    juegos.append(partido_optimo_vs_optimo(False))
print('Empieza el Jugador 1: ', list(np.sum(np.array(juegos), axis=0)/rep))

Empieza el jugador 1 y tiene A1: [0.6497, 0.3503]
Empieza el Jugador 1: [0.5524, 0.4476]

```

Asumiendo que conocemos cual es la mejor condicion para atacar (I\_MEJOR), el índice de defensa es un promedio de todas sus características. En otras palabras, el indice defensivo es aleatorio, el ofensivo es óptimo (asumiendo que desconocemos el contrincante)

```

In [345]: I_ATAQUE = []
I_DEFENSA = []
for i in range(32):
    tmp = np.max(f_norm.loc[i][['I_ALTURA', 'I_PESO', 'I_FUERZA', 'I_PEL
AS', 'I_VELOCIDAD']])
    I_ATAQUE.append(tmp)
    tmp = np.mean(f_norm.loc[i][['I_ALTURA', 'I_PESO', 'I_FUERZA', 'I_PEL
EAS', 'I_VELOCIDAD']])
    I_DEFENSA.append(tmp)

f_norm['I_ATAQUE'] = I_ATAQUE
f_norm['I_DEFENSA'] = I_DEFENSA

```

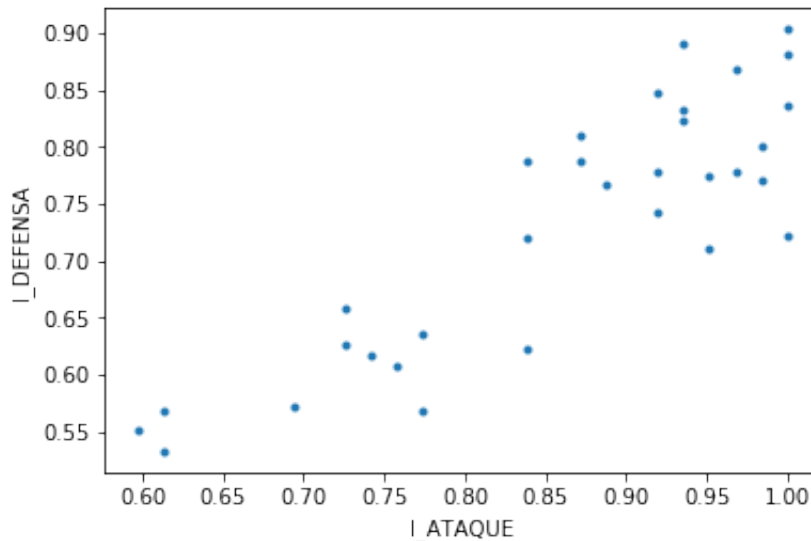
In [346]: f\_norm

Out[346]:

|    | LETRA | NUMERO | NOMBRE               | ALTURA   | PESO     | FUERZA   | PELEAS   | VELOCIDAD | SUM    |
|----|-------|--------|----------------------|----------|----------|----------|----------|-----------|--------|
| 0  | A     | 1      | Superman             | 0.300885 | 0.036719 | 1.000000 | 1.000000 | 0.795918  | 3.1335 |
| 1  | A     | 2      | Shazam               | 0.327434 | 0.042969 | 1.000000 | 0.688172 | 0.591837  | 2.6504 |
| 2  | A     | 3      | Aquaman              | 0.247788 | 0.028906 | 0.479167 | 0.715054 | 0.204082  | 1.6749 |
| 3  | A     | 4      | Martian<br>Manhunter | 0.407080 | 0.063281 | 0.947917 | 0.741935 | 0.387755  | 2.5479 |
| 4  | B     | 1      | Batman               | 0.274336 | 0.031250 | 0.218750 | 0.913978 | 0.122449  | 1.5607 |
| 5  | B     | 2      | Flash                | 0.221239 | 0.026562 | 0.192708 | 0.688172 | 1.000000  | 2.1286 |
| 6  | B     | 3      | Flecha<br>Verde      | 0.274336 | 0.026562 | 0.192708 | 0.774194 | 0.091837  | 1.3596 |
| 7  | B     | 4      | Lobo                 | 0.415929 | 0.065625 | 0.869792 | 0.500000 | 0.183673  | 2.0350 |
| 8  | C     | 1      | Mujer<br>Maravilla   | 0.265487 | 0.026562 | 0.895833 | 0.892473 | 0.183673  | 2.2640 |
| 9  | C     | 2      | Hombre<br>Halcon     | 0.327434 | 0.046094 | 0.531250 | 0.741935 | 0.183673  | 1.8303 |
| 10 | C     | 3      | Linterna<br>Verde    | 0.221239 | 0.028125 | 0.583333 | 0.720430 | 0.326531  | 1.8796 |
| 11 | C     | 4      | Gatubela             | 0.194690 | 0.004687 | 0.114583 | 0.204301 | 0.102041  | 0.6203 |
| 12 | D     | 1      | Joker                | 0.221239 | 0.014063 | 0.114583 | 0.096774 | 0.061224  | 0.5078 |
| 13 | D     | 2      | Lex Luthor           | 0.265487 | 0.031250 | 0.166667 | 0.053763 | 0.065306  | 0.5824 |
| 14 | D     | 3      | Doomsday             | 0.787611 | 0.282031 | 1.000000 | 0.069892 | 0.693878  | 2.8334 |
| 15 | D     | 4      | Darkseid             | 1.000000 | 0.601562 | 1.000000 | 0.392473 | 0.387755  | 3.3817 |
| 16 | E     | 1      | Robin                | 0.123894 | 0.010156 | 0.140625 | 0.596774 | 0.112245  | 0.9836 |
| 17 | E     | 2      | Acertijo             | 0.123894 | 0.016406 | 0.088542 | 0.000000 | 0.036735  | 0.2655 |
| 18 | E     | 3      | Pingüino             | 0.000000 | 0.034375 | 0.000000 | 0.124731 | 0.000000  | 0.1591 |
| 19 | E     | 4      | Bane                 | 0.460177 | 0.097656 | 0.296875 | 0.258065 | 0.122449  | 1.2352 |
| 20 | F     | 1      | Batichica            | 0.159292 | 0.000000 | 0.114583 | 0.487097 | 0.102041  | 0.8630 |
| 21 | F     | 2      | Gorilla<br>Grodd     | 0.823009 | 0.242188 | 0.583333 | 0.259140 | 0.102041  | 2.0097 |
| 22 | F     | 3      | Black<br>Adam        | 0.221239 | 0.046094 | 0.895833 | 0.473118 | 0.591837  | 2.2281 |
| 23 | F     | 4      | Brainiac             | 0.353982 | 0.179688 | 0.375000 | 0.166667 | 0.091837  | 1.1671 |
| 24 | G     | 1      | Dr. Fate             | 0.371681 | 0.035156 | 0.270833 | 0.741935 | 0.183673  | 1.6032 |

|           |   |   |               |          |          |          |          |          |        |
|-----------|---|---|---------------|----------|----------|----------|----------|----------|--------|
| <b>25</b> | G | 2 | Tornado Rojo  | 0.283186 | 0.187500 | 0.375000 | 0.705376 | 0.387755 | 1.9388 |
| <b>26</b> | G | 3 | Mujer Halcon  | 0.168142 | 0.007812 | 0.296875 | 0.516129 | 0.183673 | 1.1726 |
| <b>27</b> | G | 4 | Siniestro     | 0.221239 | 0.029687 | 0.390625 | 0.316129 | 0.326531 | 1.2842 |
| <b>28</b> | H | 1 | Canario Negro | 0.106195 | 0.000000 | 0.166667 | 0.597849 | 0.112245 | 0.9829 |
| <b>29</b> | H | 2 | Zatana        | 0.132743 | 0.005469 | 0.140625 | 0.715054 | 0.087755 | 1.0816 |
| <b>30</b> | H | 3 | Atomo         | 0.212389 | 0.021094 | 0.135417 | 0.666667 | 0.081633 | 1.1171 |
| <b>31</b> | H | 4 | Cyborg        | 0.371681 | 1.000000 | 0.322917 | 0.705376 | 0.153061 | 2.5530 |

```
In [348]: plt.plot(f_norm.I_ATAQUE,f_norm.I_DEFENSA,'.')
plt.xlabel('I_ATAQUE')
plt.ylabel('I_DEFENSA')
plt.show()
```

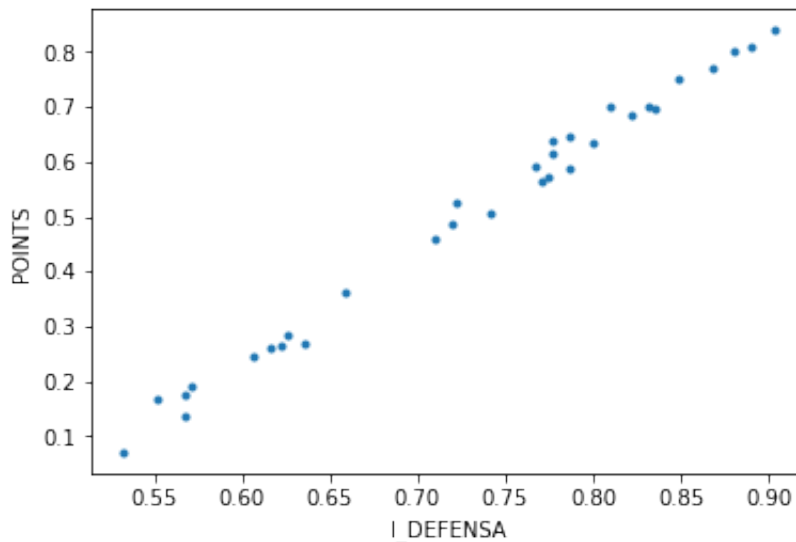


```
In [353]: plt.plot(f_norm.I_DEFENSA,f_norm.POINTS, '.')
```

```
plt.xlabel('I_DEFENSA')
```

```
plt.ylabel('POINTS')
```

```
plt.show()
```



## Evaluación de la probabilidad de triunfo de cada combinación de cartas

Dado que tenemos pocas cartas y solo 5 variables, si una carta es mayor que la otra en las 5, ganará el 100% de las veces. En vez de simular, podemos calcular la probabilidad de triunfo directamente:

```
In [355]: def triunfo_empate_derrota(database,card1,card2):
```

```
    triunfo = 0
```

```
    empate = 0
```

```
    derrota = 0
```

```
    for i in ['ALTURA', 'PESO', 'FUERZA', 'PELEAS', 'VELOCIDAD']:
```

```
        if database.iloc[card1][i] > database.iloc[card2][i]:
```

```
            triunfo += 1
```

```
        elif database.iloc[card1][i] < database.iloc[card2][i]:
```

```
            derrota += 1
```

```
        else:
```

```
            empate += 1
```

```
    return [triunfo/5.0, empate/5.0, derrota/5.0]
```

```

In [390]: I_TRIUNFO = []
I_EMPATE = []
for i in range(32):
    i_triunfo = 0.0
    i_empate = 0.0
    for j in range(32):
        if i != j:
            t, e, d = triunfo_empate_derrota(f_norm,i,j)
            i_triunfo += t
            i_empate += e
    I_TRIUNFO.append(i_triunfo)
    I_EMPATE.append(i_empate)

I_TRIUNFO = np.array(I_TRIUNFO)/31.0
I_EMPATE = np.array(I_EMPATE)/31.0

f_norm['I_TRIUNFO'] = I_TRIUNFO
f_norm['I_EMPATE'] = I_EMPATE

```

```

In [397]: f_norm

```

```

Out[397]:

```

|    | LETRA | NUMERO | NOMBRE               | ALTURA   | PESO     | FUERZA   | PELEAS   | VELOCIDAD | SUM    |
|----|-------|--------|----------------------|----------|----------|----------|----------|-----------|--------|
| 0  | A     | 1      | Superman             | 0.300885 | 0.036719 | 1.000000 | 1.000000 | 0.795918  | 3.1335 |
| 1  | A     | 2      | Shazam               | 0.327434 | 0.042969 | 1.000000 | 0.688172 | 0.591837  | 2.6504 |
| 2  | A     | 3      | Aquaman              | 0.247788 | 0.028906 | 0.479167 | 0.715054 | 0.204082  | 1.6749 |
| 3  | A     | 4      | Martian<br>Manhunter | 0.407080 | 0.063281 | 0.947917 | 0.741935 | 0.387755  | 2.5479 |
| 4  | B     | 1      | Batman               | 0.274336 | 0.031250 | 0.218750 | 0.913978 | 0.122449  | 1.5607 |
| 5  | B     | 2      | Flash                | 0.221239 | 0.026562 | 0.192708 | 0.688172 | 1.000000  | 2.1286 |
| 6  | B     | 3      | Flecha<br>Verde      | 0.274336 | 0.026562 | 0.192708 | 0.774194 | 0.091837  | 1.3596 |
| 7  | B     | 4      | Lobo                 | 0.415929 | 0.065625 | 0.869792 | 0.500000 | 0.183673  | 2.0350 |
| 8  | C     | 1      | Mujer<br>Maravilla   | 0.265487 | 0.026562 | 0.895833 | 0.892473 | 0.183673  | 2.2640 |
| 9  | C     | 2      | Hombre<br>Halcon     | 0.327434 | 0.046094 | 0.531250 | 0.741935 | 0.183673  | 1.8303 |
| 10 | C     | 3      | Linterna<br>Verde    | 0.221239 | 0.028125 | 0.583333 | 0.720430 | 0.326531  | 1.8796 |
| 11 | C     | 4      | Gatubela             | 0.194690 | 0.004687 | 0.114583 | 0.204301 | 0.102041  | 0.6203 |
| 12 | D     | 1      | Joker                | 0.221239 | 0.014063 | 0.114583 | 0.096774 | 0.061224  | 0.5078 |
| 13 | D     | 2      | Lex Luthor           | 0.265487 | 0.031250 | 0.166667 | 0.053763 | 0.065306  | 0.5824 |



|           |   |   |               |          |          |          |          |          |        |
|-----------|---|---|---------------|----------|----------|----------|----------|----------|--------|
| <b>14</b> | D | 3 | Doomsday      | 0.787611 | 0.282031 | 1.000000 | 0.069892 | 0.693878 | 2.8334 |
| <b>15</b> | D | 4 | Darkseid      | 1.000000 | 0.601562 | 1.000000 | 0.392473 | 0.387755 | 3.3817 |
| <b>16</b> | E | 1 | Robin         | 0.123894 | 0.010156 | 0.140625 | 0.596774 | 0.112245 | 0.9836 |
| <b>17</b> | E | 2 | Acertijo      | 0.123894 | 0.016406 | 0.088542 | 0.000000 | 0.036735 | 0.2655 |
| <b>18</b> | E | 3 | Pingüino      | 0.000000 | 0.034375 | 0.000000 | 0.124731 | 0.000000 | 0.1591 |
| <b>19</b> | E | 4 | Bane          | 0.460177 | 0.097656 | 0.296875 | 0.258065 | 0.122449 | 1.2352 |
| <b>20</b> | F | 1 | Batichica     | 0.159292 | 0.000000 | 0.114583 | 0.487097 | 0.102041 | 0.8630 |
| <b>21</b> | F | 2 | Gorilla Grodd | 0.823009 | 0.242188 | 0.583333 | 0.259140 | 0.102041 | 2.0097 |
| <b>22</b> | F | 3 | Black Adam    | 0.221239 | 0.046094 | 0.895833 | 0.473118 | 0.591837 | 2.2281 |
| <b>23</b> | F | 4 | Brainiac      | 0.353982 | 0.179688 | 0.375000 | 0.166667 | 0.091837 | 1.1671 |
| <b>24</b> | G | 1 | Dr. Fate      | 0.371681 | 0.035156 | 0.270833 | 0.741935 | 0.183673 | 1.6032 |
| <b>25</b> | G | 2 | Tornado Rojo  | 0.283186 | 0.187500 | 0.375000 | 0.705376 | 0.387755 | 1.9388 |
| <b>26</b> | G | 3 | Mujer Halcon  | 0.168142 | 0.007812 | 0.296875 | 0.516129 | 0.183673 | 1.1726 |
| <b>27</b> | G | 4 | Siniestro     | 0.221239 | 0.029687 | 0.390625 | 0.316129 | 0.326531 | 1.2842 |
| <b>28</b> | H | 1 | Canario Negro | 0.106195 | 0.000000 | 0.166667 | 0.597849 | 0.112245 | 0.9829 |
| <b>29</b> | H | 2 | Zatana        | 0.132743 | 0.005469 | 0.140625 | 0.715054 | 0.087755 | 1.0816 |
| <b>30</b> | H | 3 | Atomo         | 0.212389 | 0.021094 | 0.135417 | 0.666667 | 0.081633 | 1.1171 |
| <b>31</b> | H | 4 | Cyborg        | 0.371681 | 1.000000 | 0.322917 | 0.705376 | 0.153061 | 2.5530 |

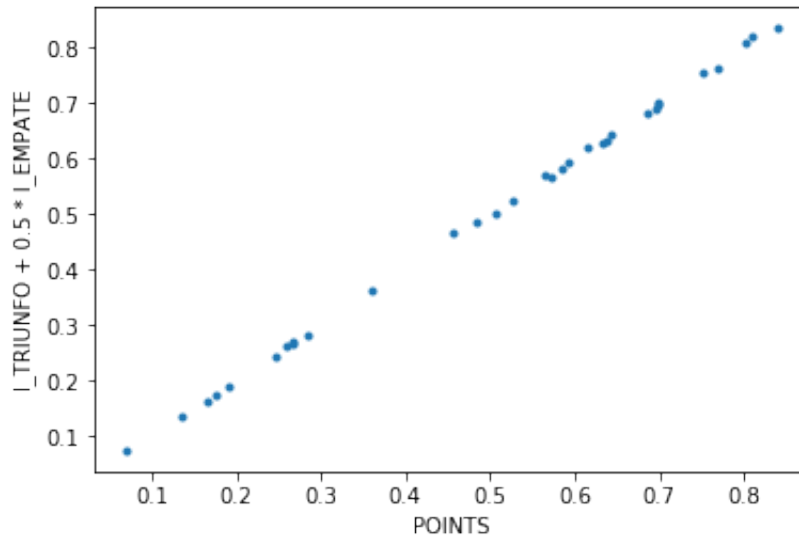
32 rows × 22 columns

```
In [394]: plt.plot(f_norm.POINTS,f_norm.I_TRIUNFO+0.5*f_norm.I_EMPATE, '.')
```

```
plt.xlabel('POINTS')
```

```
plt.ylabel('I_TRIUNFO + 0.5 * I_EMPATE')
```

```
plt.show()
```



```
In [396]: f_norm.sort_values(by = 'I_TRIUNFO',ascending=False)
```

Out[396]:

|    | LETRA | NUMERO | NOMBRE               | ALTURA   | PESO     | FUERZA   | PELEAS   | VELOCIDAD | SUM    |
|----|-------|--------|----------------------|----------|----------|----------|----------|-----------|--------|
| 0  | A     | 1      | Superman             | 0.300885 | 0.036719 | 1.000000 | 1.000000 | 0.795918  | 3.1335 |
| 3  | A     | 4      | Martian<br>Manhunter | 0.407080 | 0.063281 | 0.947917 | 0.741935 | 0.387755  | 2.5479 |
| 15 | D     | 4      | Darkseid             | 1.000000 | 0.601562 | 1.000000 | 0.392473 | 0.387755  | 3.3817 |
| 14 | D     | 3      | Doomsday             | 0.787611 | 0.282031 | 1.000000 | 0.069892 | 0.693878  | 2.8334 |
| 1  | A     | 2      | Shazam               | 0.327434 | 0.042969 | 1.000000 | 0.688172 | 0.591837  | 2.6504 |
| 25 | G     | 2      | Tornado<br>Rojo      | 0.283186 | 0.187500 | 0.375000 | 0.705376 | 0.387755  | 1.9388 |
| 31 | H     | 4      | Cyborg               | 0.371681 | 1.000000 | 0.322917 | 0.705376 | 0.153061  | 2.5530 |
| 7  | B     | 4      | Lobo                 | 0.415929 | 0.065625 | 0.869792 | 0.500000 | 0.183673  | 2.0350 |
| 9  | C     | 2      | Hombre<br>Halcon     | 0.327434 | 0.046094 | 0.531250 | 0.741935 | 0.183673  | 1.8303 |
| 21 | F     | 2      | Gorilla<br>Grodd     | 0.823009 | 0.242188 | 0.583333 | 0.259140 | 0.102041  | 2.0097 |
| 24 | G     | 1      | Dr. Fate             | 0.371681 | 0.035156 | 0.270833 | 0.741935 | 0.183673  | 1.6032 |
| 8  | C     | 1      | Mujer<br>Maravilla   | 0.265487 | 0.026562 | 0.895833 | 0.892473 | 0.183673  | 2.2640 |

|           |   |   |                |          |          |          |          |          |        |
|-----------|---|---|----------------|----------|----------|----------|----------|----------|--------|
| <b>22</b> | F | 3 | Black Adam     | 0.221239 | 0.046094 | 0.895833 | 0.473118 | 0.591837 | 2.2281 |
| <b>2</b>  | A | 3 | Aquaman        | 0.247788 | 0.028906 | 0.479167 | 0.715054 | 0.204082 | 1.6749 |
| <b>10</b> | C | 3 | Linterna Verde | 0.221239 | 0.028125 | 0.583333 | 0.720430 | 0.326531 | 1.8796 |
| <b>19</b> | E | 4 | Bane           | 0.460177 | 0.097656 | 0.296875 | 0.258065 | 0.122449 | 1.2352 |
| <b>4</b>  | B | 1 | Batman         | 0.274336 | 0.031250 | 0.218750 | 0.913978 | 0.122449 | 1.5607 |
| <b>5</b>  | B | 2 | Flash          | 0.221239 | 0.026562 | 0.192708 | 0.688172 | 1.000000 | 2.1286 |
| <b>23</b> | F | 4 | Brainiac       | 0.353982 | 0.179688 | 0.375000 | 0.166667 | 0.091837 | 1.1671 |
| <b>27</b> | G | 4 | Siniestro      | 0.221239 | 0.029687 | 0.390625 | 0.316129 | 0.326531 | 1.2842 |
| <b>6</b>  | B | 3 | Flecha Verde   | 0.274336 | 0.026562 | 0.192708 | 0.774194 | 0.091837 | 1.3596 |
| <b>26</b> | G | 3 | Mujer Halcon   | 0.168142 | 0.007812 | 0.296875 | 0.516129 | 0.183673 | 1.1726 |
| <b>30</b> | H | 3 | Atomo          | 0.212389 | 0.021094 | 0.135417 | 0.666667 | 0.081633 | 1.1171 |
| <b>13</b> | D | 2 | Lex Luthor     | 0.265487 | 0.031250 | 0.166667 | 0.053763 | 0.065306 | 0.5824 |
| <b>29</b> | H | 2 | Zatana         | 0.132743 | 0.005469 | 0.140625 | 0.715054 | 0.087755 | 1.0816 |
| <b>16</b> | E | 1 | Robin          | 0.123894 | 0.010156 | 0.140625 | 0.596774 | 0.112245 | 0.9836 |
| <b>28</b> | H | 1 | Canario Negro  | 0.106195 | 0.000000 | 0.166667 | 0.597849 | 0.112245 | 0.9829 |
| <b>20</b> | F | 1 | Batichica      | 0.159292 | 0.000000 | 0.114583 | 0.487097 | 0.102041 | 0.8630 |
| <b>11</b> | C | 4 | Gatubela       | 0.194690 | 0.004687 | 0.114583 | 0.204301 | 0.102041 | 0.6203 |
| <b>12</b> | D | 1 | Joker          | 0.221239 | 0.014063 | 0.114583 | 0.096774 | 0.061224 | 0.5078 |
| <b>18</b> | E | 3 | Pingüino       | 0.000000 | 0.034375 | 0.000000 | 0.124731 | 0.000000 | 0.1591 |
| <b>17</b> | E | 2 | Acertijo       | 0.123894 | 0.016406 | 0.088542 | 0.000000 | 0.036735 | 0.2655 |

32 rows × 22 columns

In [ ]: