

**EXTRACCIÓN DE CONOCIMIENTOS DE BASES DE DATOS
ANÁLISIS DE DATOS NBA – GRÁFICAS**

**ARANDA GONZÁLEZ LUCÍA DEL CARMEN
9ADGS**

ESCUINAPA, SINALOA, 10 DE JUNIO DE 2024

TABLAS DE TIROS DE TRES PUNTOS

	FG3A_P1	FG3A_P2
count	7833.0	7943.0
mean	20.0	30.0
std	6.0	8.0
min	3.0	8.0
25%	16.0	25.0
50%	20.0	30.0
75%	25.0	35.0
max	46.0	68.0

TABLAS DE TIROS DE DOS PUNTOS

	FGA_P1	FGA_P2
count	7833.0	7943.0
mean	81.0	86.0
std	8.0	8.0
min	46.0	50.0
25%	76.0	81.0
50%	81.0	86.0
75%	86.0	91.0
max	124.0	124.0

UNIÓN DE TABLAS DE TRES Y DOS PUNTOS

	FG3A_P1	FG3A_P2	FGA_P1	FGA_P2
count	7833.000000	7943.000000	7833.000000	7943.000000
mean	20.396272	30.363842	80.904762	86.016996
std	6.411274	7.926368	7.896620	7.885455
min	3.000000	8.000000	46.000000	50.000000
25%	16.000000	25.000000	76.000000	81.000000
50%	20.000000	30.000000	81.000000	86.000000
75%	25.000000	35.000000	86.000000	91.000000
max	46.000000	68.000000	124.000000	124.000000

EXPLICACIÓN DEL PROCESO ANTERIOR

Para la creación y unión de las dos tablas de estadística, se crean principalmente dos DataFrames vacíos, TirosTres y TirosDos, con las columnas 'FG3A_P1', 'FG3A_P2':

```
TirosTres=pd.DataFrame(columns=['FG3A_P1', 'FG3A_P2'])
TirosTres
TirosTres['FG3A_P1']=TirosTresP1['FG3A_P1']
TirosTres['FG3A_P2']=TirosTresP2['FG3A_P2']
TirosTres.round()
```

Y 'FGA_P1', 'FGA_P2' respectivamente:

```
TirosDos=pd.DataFrame(columns=['FGA_P1', 'FGA_P2'])
TirosDos
TirosDos['FGA_P1']=TirosDosP1['FGA_P1']
TirosDos['FGA_P2']=TirosDosP2['FGA_P2']
TirosDos.round()
```

Posteriormente, se asignan los valores de las columnas 'FG3A_P1', 'FG3A_P2' en TirosTres y 'FGA_P1', 'FGA_P2' en TirosDos desde los DataFrames TirosTresP1, TirosTresP2, TirosDosP1 y TirosDosP2 respectivamente.

Luego, se redondean los valores en ambas tablas usando el método .round().

Finalmente, se unen las dos tablas TirosTres y TirosDos en una sola tabla resultado usando la función pd.concat(). El parámetro axis=1 indica que la unión se realiza a lo largo del eje de las columnas (es decir, las tablas se unen horizontalmente).

```
resultado = pd.concat([TirosTres, TirosDos], axis=1)
resultado
```

PROCESO PARA LA CONSTRUCCIÓN DE LA PRIMERA GRÁFICA (G1)

Para la construcción de la primera gráfica se implementaron las siguientes sentencias, donde se declara principalmente la información base de la gráfica.

Definimos el dataframe (df1, el primero) del que se extraerán los datos e imprimimos el resultado.

```
df1_circ=df1[['FGA', 'FG3A']]
df1_circ
```

	FGA	FG3A
0	82	26
1	69	19
2	73	19
3	78	19
4	75	17
...
8655	67	18
8656	72	16
8657	66	18
8658	63	12
8659	66	19

Posteriormente transponemos los valores previamente generados de manera horizontal, imprimiéndolos con la última línea.

```
df1_circ=df1_circ.T # T de transponer los valores verticales a horizontales
df1_circ
```

	0	1	2	3	4	5	6	7	8	9	...	8650	8651	8652	8653	8654	8655	8656	8657	8658	8659
FGA	82	69	73	78	75	80	66	78	75	81	...	73	77	65	70	66	67	72	66	63	66
FG3A	26	19	19	19	17	24	15	20	13	25	...	20	26	12	16	18	18	16	18	12	19

Ahora definimos “TotalTiros” de manera que nos muestre una columna nueva asignando valores de cada fila calculada, y axis suma todas las filas devolviendo una serie de “Pandas”.

```
df1_circ['TotalTiros']=df1_circ.sum(axis=1)
df1_circ
```

	0	1	2	3	4	5	6	7	8	9	...	8651	8652	8653	8654	8655	8656	8657	8658	8659	TotalTiros
FGA	82	69	73	78	75	80	66	78	75	81	...	77	65	70	66	67	72	66	63	66	633727
FG3A	26	19	19	19	17	24	15	20	13	25	...	26	12	16	18	18	16	18	12	19	159764

EXPLICACIÓN G1

Este código está utilizando la biblioteca de visualización de datos matplotlib para crear un gráfico circular (o gráfico de pastel) de la columna 'TotalTiros' del DataFrame df1_circ:

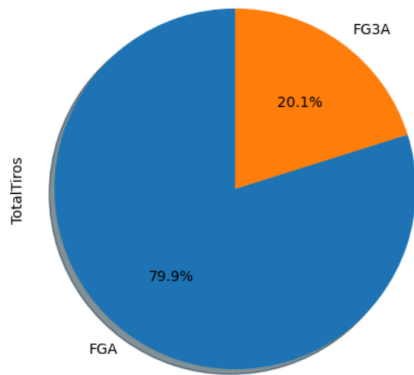
- `kind='pie'`: especifica que queremos un gráfico de pastel.
- `figsize=(5, 6)`: especifica el tamaño de la figura en pulgadas (ancho, alto).
- `autopct='%1.1f%%'`: añade etiquetas a las secciones del gráfico de pastel con el porcentaje que cada sección representa del total. El formato `'%1.1f%%'` significa que los porcentajes se mostrarán con un decimal.
- `startangle=90`: especifica el ángulo de inicio del gráfico de pastel. En este caso, el gráfico comenzará a las 12 en punto y continuará en sentido antihorario.
- `shadow=True`: añade una sombra al gráfico de pastel para un efecto tridimensional.

```
df1_circ['TotalTiros'].plot(kind='pie',
                             figsize=(5, 6),
                             autopct='%1.1f%%', # añadimos porcentajes,
                             startangle=90,      # ángulo de comienzo,
                             shadow=True,        # añadimos sombra,
                             ),
plt.title('Distribución de Tiros para el Periodo 1')
plt.axis('equal')

plt.show()
```

Y este es el resultado grafico que genera, mostrando un grafico de pastel con 20.1% de FG3A y 79.9% de FGA.

Distribución de Tiros para el Periodo 1



PROCESO PARA LA CONSTRUCCIÓN DE LA SEGUNDA GRÁFICA (G2)

Parecido a la grafica anterior, definimos el dataframe del que se extraerán los datos e imprimimos el resultado, solo que definimos que el dataframe que usaremos esta vez es el segundo (df2).

```
df2_circ=df2[['FGA','FG3A']]
df2_circ
```

	FGA	FG3A
7147	82	25
7148	77	27
7149	83	24
7150	81	36
7151	91	25
...
15771	67	22
15772	65	24
15773	76	33
15774	71	24
15775	86	26

Y similar al código previo, transponemos los valores previamente generados de manera horizontal, imprimiéndolos con la última línea.

```
df2_circ=df2_circ.T # T de transponer los valores verticales a horizontales
df2_circ
```

	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	...	15766	15767	15768	15769	15770	15771	15772	15773	15
FGA	82	77	83	81	91	81	87	85	88	74	...	80	81	80	83	77	67	65	76	
FG3A	25	27	24	36	25	33	27	37	45	31	...	35	35	33	27	34	22	24	33	

Ahora definimos “TotalTiros” de manera que nos muestre una columna nueva asignando valores de cada fila calculada, y axis suma todas las filas devolviendo una serie de “Pandas”, pero recordando que estamos utilizando el dataframe 2, por lo que se declara en cada sentencia que contenga “df” igualmente.

```
df2_circ['TotalTiros']=df2_circ.sum(axis=1)
df2_circ
```

	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	...	15768	15769	15770	15771	15772	15773	15774	15775	.
FGA	82	77	83	81	91	81	87	85	88	74	...	80	83	77	67	65	76	71	86	68
FG3A	25	27	24	36	25	33	27	37	45	31	...	33	27	34	22	24	33	24	26	24

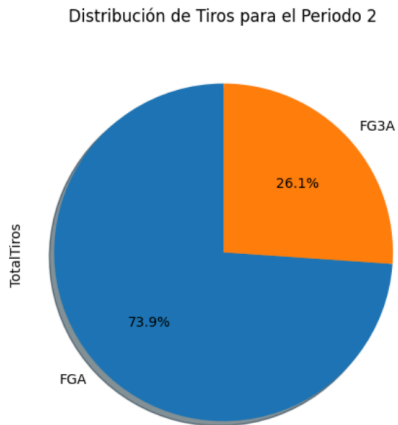
EXPLICACIÓN G2

En este código también se está utilizando la biblioteca de visualización de datos matplotlib para crear un gráfico circular (o gráfico de pastel) de la columna 'TotalTiros' del DataFrame df2_circ, e igual, especificamos el grafico de paste, el tamaño de la figura, etiquetas de sección del porcentaje, el ángulo del inicio del gráfico, y una sombra para agregar estética.

```
df2_circ['TotalTiros'].plot(kind='pie',
                             figsize=(5, 6),
                             autopct='%1.1f%%', # añadimos porcentajes,
                             startangle=90,      # ángulo de comienzo,
                             shadow=True,         # añadimos sombra,
                             ),
plt.title('Distribución de Tiros para el Periodo 2')
plt.axis('equal')

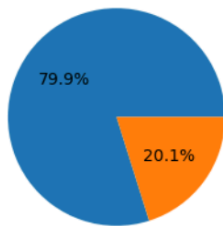
plt.show()
```

El resultado grafico que genera, muestra un gráfico de pastel con 26.1% de FG3A y 73.9% de FGA.

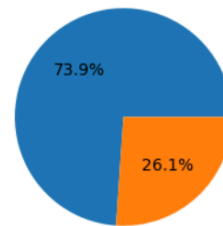


UNIÓN DE LAS DOS GRÁFICAS

Distribución de Tiros para el Periodo 1



Distribución de Tiros para el Periodo 2



EXPLICACIÓN DE LA UNIÓN DE LAS DOS GRÁFICAS

Este código genera lo que son los dos gráficos de pastel en una sola figura para comparar la distribución (de manera visual) de tiros en dos periodos diferentes. Primero, se crea una figura con dos subgráficos (ax1 y ax2) en una fila y comparten el eje y. Luego, en cada subgráfico, se dibuja un gráfico de pastel utilizando los datos de 'TotalTiros' de los DataFrames df1_circ y df2_circ respectivamente. Los porcentajes se muestran en cada sección del gráfico de pastel. Finalmente, se establecen los títulos de los subgráficos y se muestra la figura con "show()".

```
f, (ax1, ax2) = plt.subplots(1, 2, sharey=True, figsize=(15,3))
ax1.pie(df1_circ['TotalTiros'], autopct='%1.1f%%')
ax1.set_title('Distribución de Tiros para el Periodo 1')
ax2.pie(df2_circ['TotalTiros'], autopct='%1.1f%%')
ax2.set_title('Distribución de Tiros para el Periodo 2')

plt.show()
```

TABLA DE CABECERA

	Equipo	Fecha	PTS	FGM	FGA	FG3M	FG3A	Total Tiros	Total Tiros Encestados
0	Dallas Mavericks	2011-06-12	105	41	82	11	26	108	52
1	Dallas Mavericks	2011-06-09	112	39	69	13	19	88	52
2	Dallas Mavericks	2011-06-07	86	29	73	4	19	92	33
3	Miami Heat	2011-06-05	88	34	78	8	19	97	42
4	Dallas Mavericks	2011-06-02	95	36	75	6	17	92	42

EXPLICACIÓN DE LA TABLA DE CABECERA

Con la función `df.head()`, logamos mostrar las primeras 5 filas del dataframe que tenemos definido con las siglas 'df', meramente como análisis previo.

```
df.head()
```

TABLAS DEL DATAFRAME LINE

	Equipo	Fecha	PTS	FGM	FGA	FG3M	FG3A	Total Tiros	Total Tiros Encestados
0	Dallas Mavericks	2011-06-12	105	41	82	11	26	108	52
1	Dallas Mavericks	2011-06-09	112	39	69	13	19	88	52
2	Dallas Mavericks	2011-06-07	86	29	73	4	19	92	33
3	Miami Heat	2011-06-05	88	34	78	8	19	97	42
4	Dallas Mavericks	2011-06-02	95	36	75	6	17	92	42
...
15771	Los Angeles Lakers	2021-08-04	84	27	67	6	22	89	33
15772	Miami Heat	2021-08-04	94	31	65	9	24	89	40
15773	Golden State Warriors	2021-08-03	89	33	76	10	33	109	43
15774	Miami Heat	2021-08-03	80	33	71	7	24	95	40
15775	Memphis Grizzlies	2021-08-03	104	41	86	10	26	112	51

	Equipo	Fecha	PTS	FGM	FGA	FG3M	FG3A	Total Tiros	Total Tiros Encestados	Year
0	Dallas Mavericks	2011-06-12	105	41	82	11	26	108	52	2011
1	Dallas Mavericks	2011-06-09	112	39	69	13	19	88	52	2011
2	Dallas Mavericks	2011-06-07	86	29	73	4	19	92	33	2011
3	Miami Heat	2011-06-05	88	34	78	8	19	97	42	2011
4	Dallas Mavericks	2011-06-02	95	36	75	6	17	92	42	2011
...
15771	Los Angeles Lakers	2021-08-04	84	27	67	6	22	89	33	2021
15772	Miami Heat	2021-08-04	94	31	65	9	24	89	40	2021
15773	Golden State Warriors	2021-08-03	89	33	76	10	33	109	43	2021
15774	Miami Heat	2021-08-03	80	33	71	7	24	95	40	2021
15775	Memphis Grizzlies	2021-08-03	104	41	86	10	26	112	51	2021

EXPLICACIÓN DE TABLAS DEL DATAFRAME LINE

Posteriormente creamos una copia del dataframe bajo el nombre 'df_line', es decir, donde se almacenará dicha copia, con el fin de crear una grafica de línea.

```
df_line=df
df_line
```

Convertimos la columna 'fecha' del dataframe df_line al formato de fecha correspondiente a año, mes y día, luego, se crea la columna 'Year' en df_line extrayendo el año de la columna de 'Fecha', y la última línea imprime el resultado.

```
df_line['Fecha'] = pd.to_datetime(df_line.Fecha, format='%Y-%m-%d')
```

```
df_line['Year']=df_line['Fecha'].dt.year
df_line
```

COLUMNAS DEL DATAFRAME LINE

```
Index(['Equipo', 'Fecha', 'PTS', 'FGM', 'FGA', 'FG3M', 'FG3A', 'Total Tiros',
      'Total Tiros Encestados', 'Year'],
      dtype='object')
```

EXPLICACIÓN DE LAS COLUMNAS DEL DATAFRAME LINE

Con 'df_line.columns' mostramos los nombres de las columnas del dataframe df_line, para su posterior uso en un subconjunto de columnas.

```
df_line.columns
```

SUBCONJUNTO DE COLUMNAS

	Year	PTS	FGM	FGA	FG3M	FG3A	Total Tiros	Total Tiros Encestados
0	2011	105	41	82	11	26	108	52
1	2011	112	39	69	13	19	88	52
2	2011	86	29	73	4	19	92	33
3	2011	88	34	78	8	19	97	42
4	2011	95	36	75	6	17	92	42
...
15771	2021	84	27	67	6	22	89	33
15772	2021	94	31	65	9	24	89	40
15773	2021	89	33	76	10	33	109	43
15774	2021	80	33	71	7	24	95	40
15775	2021	104	41	86	10	26	112	51

	Year	PTS	FGM	FGA	FG3M	FG3A	Total Tiros	Total Tiros Encestados
0	2010	69002	25162	52614	4621	11782	64396	29783
1	2011	95314	35384	73598	6394	16609	90207	41778
2	2012	168430	63312	134542	11888	31360	165902	75200
3	2013	156571	58514	122689	12084	31126	153815	70598
4	2014	161985	59741	124843	12942	33557	158400	72683
5	2015	158561	58648	125441	13357	35330	160771	72005
6	2016	164506	60732	127595	14839	38497	166092	75571
7	2017	167182	61600	127670	16479	42337	170007	78079
8	2018	167683	61858	128756	17126	44809	173565	78984
9	2019	165903	60683	126560	18118	47556	174116	78801
10	2020	93087	33672	69179	10655	27305	96484	44327
11	2021	137653	50451	103473	16069	40676	144149	66520

EXPLICACIÓN DEL PROCEDIMIENTO DEL SUBCONJUNTO DE COLUMNAS

Seleccionamos dicho subconjunto de columnas del dataframe df y lo almacenamos en df_line, tomando en cuenta todos los valores obtenidos en la consulta de columnas, pero excluyendo las columnas equipo y fecha.

```
df_line=df[['Year','PTS', 'FGM', 'FGA', 'FG3M', 'FG3A', 'Total Tiros',
            'Total Tiros Encestados']]
df_line
```

Agrupamos el dataframe df_line por la columna de 'Year' y sumamos los valores de las demás columnas para cada año.

```
df_line=df_line.groupby(['Year'], as_index=False).sum()
df_line
```

DATAFRAME FINAL

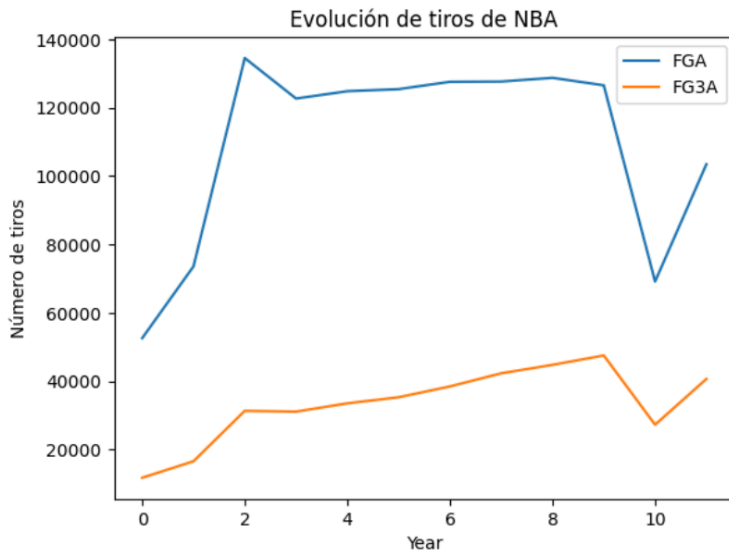
	FGA	FG3A
0	52614	11782
1	73598	16609
2	134542	31360
3	122689	31126
4	124843	33557
5	125441	35330
6	127595	38497
7	127670	42337
8	128756	44809
9	126560	47556
10	69179	27305
11	103473	40676

EXPLICACIÓN DEL DATAFRAME FINAL

Se crea un nuevo dataframe con el nombre de `df_line_final`, en donde se seleccionan solo las columnas 'FGA' y 'FG3A' de `df_line`, finalmente en la última línea se imprime el resultado.

```
df_line_final=df_line[['FGA','FG3A']]
df_line_final
```

GRÁFICA DE LINEAS



EXPLICACIÓN DE GRÁFICA DE LINEAS

Creamos un gráfico de líneas con los datos obtenidos de `df_line_final`, establecemos el título de la gráfica (Evolución de tiros de NBA), también asignamos la etiqueta del eje 'y' como 'Número de tiros', al igual que con la etiqueta del eje 'x' pero con el nombre de 'Year' o año en español. Finalizando con la función para mostrar la gráfica `show()`.

```
df_line_final.plot(kind='line')
plt.title('Evolución de tiros de NBA')
plt.ylabel('Número de tiros')
plt.xlabel('Year')
plt.show()
```

CONCLUSIÓN DEL ANÁLISIS DE LAS GRÁFICAS

En conclusión, la primera gráfica (G1) muestra que el 79.9% de los tiros intentados son tiros de dos puntos (FGA), mientras que solo el 20.1% son tiros de tres puntos (FG3A). Esto sugiere que, en el periodo de tiempo analizado, los equipos o jugadores prefirieron intentar más tiros de dos puntos que de tres puntos.

La segunda gráfica (G2) muestra una distribución ligeramente diferente, donde el 73.9% de los tiros intentados son tiros de dos puntos (FGA), y el 26.1% son tiros de tres puntos (FG3A). Aunque los tiros de dos puntos siguen siendo los más frecuentes, se observa un ligero aumento en el porcentaje de tiros de tres puntos intentados en comparación con la primera gráfica.

Al comparar ambas gráficas, se puede notar un pequeño cambio en la distribución de los tiros intentados. En la segunda gráfica, hay un incremento del 6% en el porcentaje de tiros de tres puntos intentados en comparación con la primera gráfica. Este cambio en la distribución de los tiros podría indicar una tendencia creciente hacia intentar más tiros de tres puntos en el deporte analizado (presumiblemente baloncesto NBA), lo cual podría reflejar cambios en las estrategias de juego o preferencias de los equipos y jugadores. Las gráficas muestran que, si bien los tiros de dos puntos son los más comunes en ambos períodos analizados, se observa un ligero aumento en la preferencia por intentar tiros de tres puntos en el segundo período en comparación con el primero. Posteriormente se realiza un análisis detallado de los datos de tiro en la NBA, donde de acuerdo con la selección de las columnas y sumas anuales muestra un gráfico de líneas donde la evolución de los tiros de campo intentados ('FGA') han incrementado más que los tiros de tres puntos intentados ('FG3A') a lo largo de los años, siendo superior. Este análisis brinda información valiosa sobre las tendencias y estrategias de juego en la NBA, además de entender a profundidad lo que son las tendencias en las estrategias de tiro.