

**EXTRACCIÓN DE CONOCIMIENTOS EN BASES DE DATOS  
REPORTE DE CASO: NBA  
UNIDAD 2: PROCEDENCIA DE LOS DATOS**

**ARANDA GONZÁLEZ LUCÍA DEL CARMEN  
9ADGS**

**ESCUINAPA, SINALOA, 31 DE MAYO DE 2024**

## CONTEXTO DEL ANÁLISIS

El documento presente, muestra un informe o análisis relacionado / enfocado con datos de la NBA (National Basketball Association). El análisis se centra en la manipulación y transformación de un DataFrame proveniente de un archivo extensión CSV, que contiene datos sobre tiros de dos puntos (FGA), generando resultados gráficos en tablas.

El análisis se realiza utilizando el lenguaje de programación Python y su librería Pandas para manipulación y análisis de datos. Se trabaja con dos DataFrames diferentes, df1 y df2, que contienen datos sobre tiros de dos puntos (columna 'FGA') de la NBA.

El análisis de datos del data frame, busca generar estadísticas descriptivas, transformar los datos y crear nuevos DataFrames a partir de la columna 'FGA' de los DataFrames originales df1 y df2. Se utilizan métodos como .describe(), .to\_frame() y operaciones como eliminar y renombrar columnas.

## OBJETIVO DEL ANÁLISIS

Analizar datos del data frame NBADATOS, en la herramienta web JupyterLite, con las librerías de python pandas, matplotlib.pyplot, y numpy, para generar resultados estadísticos y exactos mostrando así tendencias o relaciones entre la información.

## HIPOTESIS

Los datos de tiros de dos puntos (FGA) de dos temporadas o equipos diferentes de la NBA muestran diferencias significativas en sus estadísticas descriptivas y distribución, lo que podría indicar tendencias o patrones interesantes al comparar los dos conjuntos de datos.

## DESARROLLO DEL ANÁLISIS

Celda 1:

Esta sentencia realiza la función de importar librerías de cálculos y estadística, la ruta del archivo que se explora, su lectura, y finalmente su impresión/consulta de visualización.

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
path='NBADATOS.csv'
df=pd.read_csv(path)
df
```

Celda 6:

Esta sentencia realiza la función de reemplazar el nombre de los campos 'GAME\_DATE' y 'TEAM\_NAME' por 'Fecha' y 'Equipo', posteriormente, se eliminan las dos columnas primeramente mencionadas del DataFrame df, se indica una operación de columna (axis=1), e indicando que los cambios se realizan en el DataFrame original. La última línea restablece el índice del DataFrame df a su valor por defecto (es decir, una secuencia de enteros de 0 a N-1, donde N es la longitud del DataFrame).

```
[6]: df['Fecha']=df['GAME_DATE']
      df['Equipo']=df['TEAM_NAME']
      df.drop('GAME_DATE', axis=1, inplace=True)
      df.drop('TEAM_NAME',axis=1,inplace=True)
      df.reset_index()
```

Celda 8:

Esta sentencia realiza la función de acomodar columnas del DataFrame con sus contenidos, organizados en un orden específico declarado por un arreglo. Después las muestra.

```
[8]: df=df[['Equipo', 'Fecha', 'PTS', 'FGM', 'FGA', 'FG3M', 'FG3A', 'Total Tiros',
      'Total Tiros Encestados']]
      df
```

Celda 9:

Esta sentencia realiza la función de ordenar los valores del DataFrame por fecha y de manera ascendente, después se divide el DataFrame en dos partes, cada una con un rango de fecha especificado, finalmente por separado, se imprimen los resultados de la parte correspondiente del DataFrame.

```
[9]: df.sort_values('Fecha', ascending=True)
      df1=df[df['Fecha']<'2016-01-01']
      df2=df[df['Fecha']>='2016-01-01']
      df1
      df2
```

Celda 10:

En la primera línea, estamos tomando la columna 'FGA' del DataFrame df1. El método .describe() proporciona las estadísticas resumidas para columnas numéricas y categóricas. Al especificar include='all', estamos solicitando que se incluyan estadísticas para todas las columnas, incluso las no numéricas (como las

de tipo objeto). Las estadísticas incluidas son: recuento (count), valores únicos (unique), valor más frecuente (top) y frecuencia del valor más frecuente (freq).

En la segunda línea, similar al caso anterior, estamos aplicando .describe() a la columna 'FGA' del DataFrame df2.

Nuevamente, se incluirán estadísticas para todas las columnas, independientemente de su tipo.

```
[10]: TirosDosP1=df1['FGA'].describe(include='all')
       TirosDosP2=df2['FGA'].describe(include='all')
```

Celda 11:

En la primera línea, se está convirtiendo un objeto (posiblemente una serie o un DataFrame) llamado TirosDosP1 en un DataFrame. La función .to\_frame() convierte una serie en un DataFrame de una sola columna.

En la segunda línea, se crea una nueva columna llamada 'FGA\_P1' en el DataFrame TirosDosP1. El contenido de esta columna es el mismo que la columna existente 'FGA'.

La tercera línea imprime el resultado.

```
[11]: TirosDosP1=TirosDosP1.to_frame()
       TirosDosP1['FGA_P1']=TirosDosP1['FGA']
       TirosDosP1
```

Celda 12:

La primera línea elimina la columna llamada 'FGA' del DataFrame TirosDosP1, con axis=1, Indica que estamos eliminando una columna (el valor 1 se refiere al eje de las columnas), e inplace=True significa que la operación se realizará directamente en el DataFrame TirosDosP1 sin crear un nuevo DataFrame. Finalmente se imprime el resultado.

```
[12]: TirosDosP1.drop('FGA', axis=1, inplace=True)
       TirosDosP1
```

Celda 13:

La primera línea crea un nuevo DataFrame llamado TirosDosP2 a partir de una columna existente. El método .to\_frame() convierte la columna en un DataFrame.

La segunda línea renombra la columna FGA a FGA\_P2 en el DataFrame TirosDosP2 e imprime los resultados.

La cuarta línea elimina la columna original FGA del DataFrame TirosDosP2, y nuevamente imprime los resultados.

```
[13]: TirosDosP2=TirosDosP2.to_frame()
TirosDosP2['FGA_P2']=TirosDosP2['FGA']
TirosDosP2
TirosDosP2.drop('FGA', axis=1, inplace=True)
TirosDosP2
```

Celda 14:

Se crea en la primera línea un nuevo DataFrame llamado TirosDos con dos columnas: 'FGA\_P1' y 'FGA\_P2'. Inicialmente, este DataFrame está vacío y se imprime.

Luego, en la tercera y cuarta línea, se asignan los valores de las columnas 'FGA\_P1' y 'FGA\_P2' a partir de otros DataFrames llamados TirosDosP1 y TirosDosP2, respectivamente.

Finalmente, en la quinta línea, se redondean los valores en el DataFrame TirosDos.

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

## RESULTADOS

Celda 1:

```
[1]:
```

	TEAM_ABBREVIATION	TEAM_NAME	GAME_DATE	MATCHUP	W	MIN	PTS	FGM	FGA	FG_PCT	... AST	STL	BLK	TOV	PF	PLUS_MINUS	Game Type	Season	GAME_NUM	Round	
0	DAL	Dallas Mavericks	2011-06-12	DAL @ MIA	W	240	105	41	82	0.500	...	19	11	1	14	24	10.0	Playoff	2010-11	6	NBA Finals
1	DAL	Dallas Mavericks	2011-06-09	DAL vs. MIA	W	241	112	39	69	0.565	...	23	8	3	11	20	9.0	Playoff	2010-11	5	NBA Finals
2	DAL	Dallas Mavericks	2011-06-07	DAL vs. MIA	W	240	86	29	73	0.397	...	13	7	2	11	18	3.0	Playoff	2010-11	4	NBA Finals
3	MIA	Miami Heat	2011-06-05	MIA @ DAL	W	239	88	34	78	0.436	...	20	8	5	10	27	2.0	Playoff	2010-11	3	NBA Finals
4	DAL	Dallas Mavericks	2011-06-02	DAL @ MIA	W	240	95	36	75	0.480	...	18	8	2	18	20	2.0	Playoff	2010-11	2	NBA Finals
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
15771	LAL	Los Angeles Lakers	2021-08-04	LAL @ SAC	W	240	84	27	67	0.403	...	11	11	5	14	13	4.0	Regular	22021	Regular	Regular
15772	MIA	Miami Heat	2021-08-04	MIA @ GSW	W	240	94	31	65	0.477	...	19	13	4	21	17	13.2	Regular	22021	Regular	Regular
15773	GSW	Golden State Warriors	2021-08-03	GSW @ SAC	W	265	89	33	76	0.434	...	18	4	6	16	20	1.0	Regular	22021	Regular	Regular
15774	MIA	Miami Heat	2021-08-03	MIA vs. LAL	W	241	80	33	71	0.465	...	18	10	4	19	21	2.0	Regular	22021	Regular	Regular
15775	MEM	Memphis Grizzlies	2021-08-03	MEM @ UTB	W	200	104	41	86	0.477	...	26	10	12	12	15	39.0	Regular	22021	Regular	Regular

Celda 6:

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

Celda 8:

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

Celda 9:

[9]:

		Equipo	Fecha	PTS	FGM	FGA	FG3M	FG3A	Total Tiros	Total Tiros Encestados
7147	Cleveland Cavaliers	2016-06-19	93	33	82	6	25	107	39	
7148	Cleveland Cavaliers	2016-06-16	115	40	77	10	27	104	50	
7149	Cleveland Cavaliers	2016-06-13	112	44	83	10	24	107	54	
7150	Golden State Warriors	2016-06-10	108	33	81	17	36	117	50	
7151	Cleveland Cavaliers	2016-06-08	120	48	91	12	25	116	60	
...	...	...	...	...	...	...	...	...	...	...
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	

Celda 11:

[11]:

	FGA	FGA_P1
<b>count</b>	7833.000000	7833.000000
<b>mean</b>	80.904762	80.904762
<b>std</b>	7.896620	7.896620
<b>min</b>	46.000000	46.000000
<b>25%</b>	76.000000	76.000000
<b>50%</b>	81.000000	81.000000
<b>75%</b>	86.000000	86.000000
<b>max</b>	124.000000	124.000000

Celda 12:

[12]:

### FGA\_P1

<b>count</b>	7833.000000
<b>mean</b>	80.904762
<b>std</b>	7.896620
<b>min</b>	46.000000
<b>25%</b>	76.000000
<b>50%</b>	81.000000
<b>75%</b>	86.000000
<b>max</b>	124.000000

Celda 13:

[13]:

### FGA\_P2

<b>count</b>	7943.000000
<b>mean</b>	86.016996
<b>std</b>	7.885455
<b>min</b>	50.000000
<b>25%</b>	81.000000
<b>50%</b>	86.000000
<b>75%</b>	91.000000
<b>max</b>	124.000000

Celda 14:

[14]:	FGA_P1	FGA_P2
<b>count</b>	7833.0	7943.0
<b>mean</b>	81.0	86.0
<b>std</b>	8.0	8.0
<b>min</b>	46.0	50.0
<b>25%</b>	76.0	81.0
<b>50%</b>	81.0	86.0
<b>75%</b>	86.0	91.0
<b>max</b>	124.0	124.0

## CONCLUSIONES

Se analizaron datos de tiros de dos puntos (FGA) de dos fuentes diferentes, utilizando los DataFrames df1 y df2 en Python con la librería Pandas, e igualmente se generaron estadísticas descriptivas para las columnas 'FGA' de ambos DataFrames utilizando el método .describe(), lo cual permitió obtener un resumen de características como el recuento, valores únicos, valor más frecuente y su frecuencia, y se logró la creación nuevos DataFrames (TirosDosP1 y TirosDosP2) a partir de las columnas 'FGA' de los DataFrames originales, realizando operaciones cómo renombrar columnas y eliminar columnas originales.

Finalmente, se combinaron los nuevos DataFrames TirosDosP1 y TirosDosP2 en un solo DataFrame llamado TirosDos, con las columnas 'FGA\_P1' y 'FGA\_P2' provenientes de las dos fuentes de datos originales.

Este análisis concluye en que permitió preparar y transformar los datos de tiros de dos puntos de las dos fuentes diferentes, creando un nuevo DataFrame consolidado (TirosDos) que podría ser utilizado para análisis posteriores, como comparar las estadísticas descriptivas o explorar patrones y tendencias entre las dos fuentes de datos.

El documento describe un proceso de manipulación y transformación de datos de tiros de dos puntos de la NBA utilizando Python y Pandas, con el fin de generar un conjunto de datos consolidado para su posterior análisis y comparación.