



ANÁLISIS DE LOS ATLETAS MEJOR PAGADOS DEL MUNDO

INTRODUCCIÓN A LA CIENCIA DE DATOS

Emmanuel Aguilar Yepiz

Eduardo Alfonso Hernández López

Márquez Leonel Dulce Stephanie

INTRODUCCIÓN A LA CIENCIA DE DATOS

DEVF PROYECTO 1

Contents

1.OBJETIVO	3
2 INTRODUCCIÓN	4
Análisis de los atletas mejor pagados del mundo	4
3 PRIMEROS PASOS	5
3.1 Importación de librerías base:	5
3.2 Carga de la información:	5
4 DATA CLEANING	6
4.1 Transformación minúsculas-mayúsculas:	6
4.2 Alineación de elementos similares:	6
4.3 Limpieza de la columna Previous Year Rank:	6
4.4 Limpieza de la columna S.NO:	7
4.5 Agregamos una columna de tipo date time con formato año	7
4.6 Limpiamos la columna Name = Nombre del Jugador y “nacionalidad”	7
4.7 Integración de la data frame final	8
5 PROPÓSITO DEL DOCUMENTO	9
5.1Deportistas cuyo ranking ha subido al menos dos lugares entre 2010 y 2020.	9
5.2 Atleta con el menor número de apariciones y mayores ganancias.	10
5.3Deporte y país con mayor número de atletas no rankeados que entraron en la lista de atletas mejor pagados.	12
5.4 País con mayor número de deportes con atletas en el dataset.	13
5.5 ¿Cuántos atletas pertenecen a cada deporte?	14
5.6 Atletas con mayores ganancias por deporte por década.	15
5.6 Ganancia total por cada deporte por cada año	17
6 DE LOS RESULTADOS EN POWER BI	19



1.OBJETIVO

Utilizar las herramientas de pandas, numpy, seaborn, pandas y matplotlib en python para el análisis del ingreso de los deportistas mejor pagados del mundo en los últimos años; todo lo anterior clasificando, analizando, comparando e implementando las bibliotecas sugeridas.

2 INTRODUCCIÓN

Análisis de los atletas mejor pagados del mundo

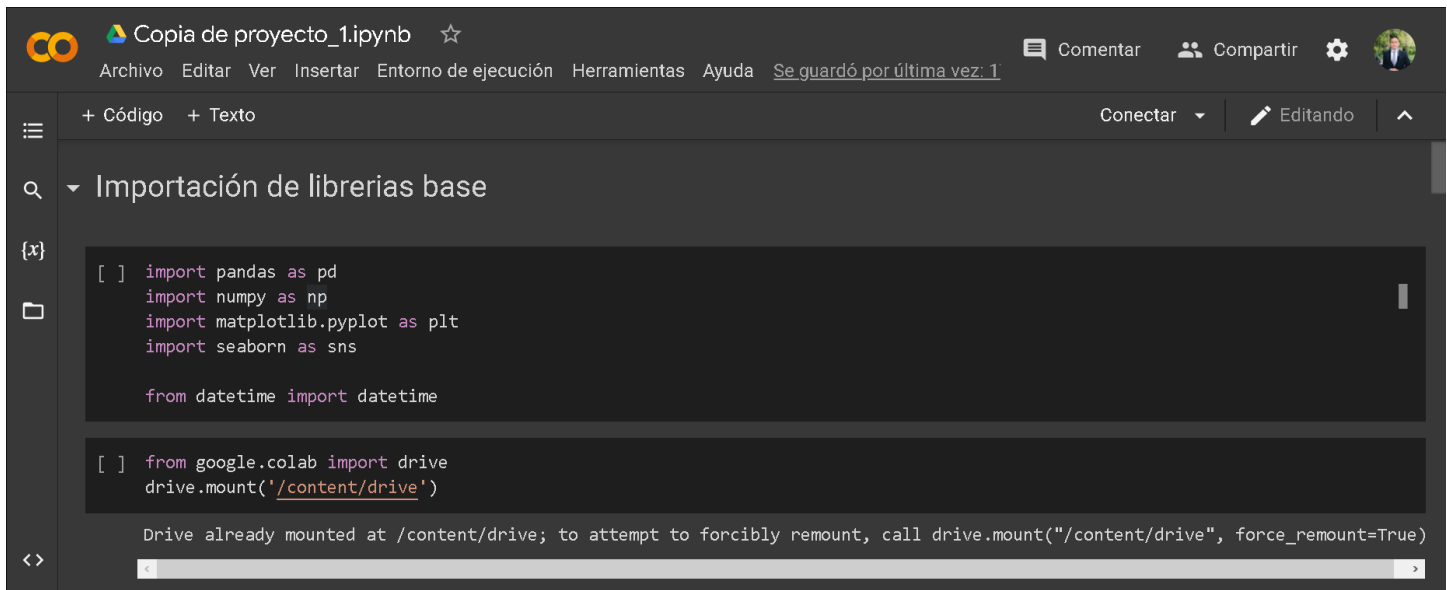
En base a las indicaciones del archivo del proyecto compartido, se tienen los siguientes datos:

- Tiger Woods domina la lista de los mejores clasificados recientemente mientras que antes lo era Michael Jordan.
- EE. UU. domina el mundo en lo que respecta a las ganancias.
- Monica Seles es la única mujer que figura en la lista de los 10 deportistas mejor pagados entre 1990 y 2020.
- Los 3 principales ganadores en 2020 son jugadores de fútbol.
- Los jugadores de baloncesto son los que más ganan, seguidos de Boxeo y Golf.

En este hacklab, se desarrollará un script en **Python** con la ayuda de **Google Colaboratory** para analizar los datos y luego usamos **Plotly** y **Matplotlib** para mejorar la visualización y obtener mejores aprendizajes sobre el dataset.

3 PRIMEROS PASOS

3.1 Importación de librerías base:



```
[ ] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

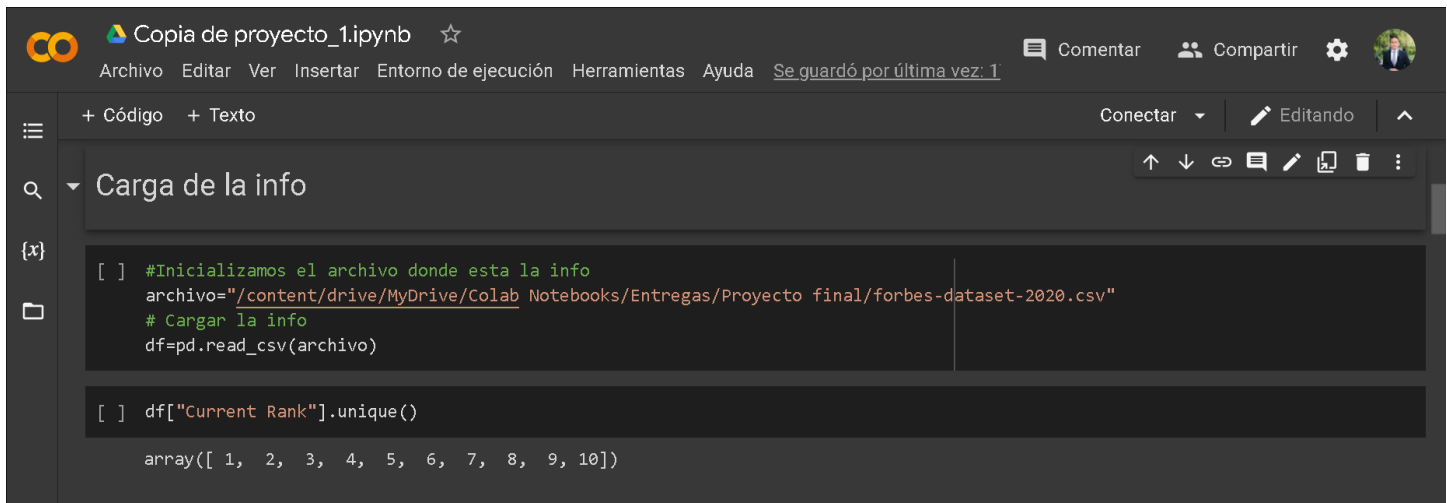
from datetime import datetime

[ ] from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

Se importan las herramientas necesarias para el manejo de la información: pandas, numpy, matplotlib y seaborn. Se crea el puente entre colab y drive.

3.2 Carga de la información:



```
[ ] #Inicializamos el archivo donde esta la info
archivo="/content/drive/MyDrive/Colab Notebooks/Entregas/Proyecto final/forbes-dataset-2020.csv"
# Cargar la info
df=pd.read_csv(archivo)

[ ] df["Current Rank"].unique()

array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

Leemos el archivo desde el drive

4 DATA CLEANING

4.1 Transformación minúsculas-mayúsculas:

```
Limpieza de la info

[ ] # Copiamos el dataframe para no alterar el original
    df2=df.copy()

[ ] #Pasamos a mayusculas para tratar de dejar iguales los datos
    df2["Sport"]=df["Sport"].str.upper()
    df2["Name"]=df["Name"].str.upper()
    df2["Nationality"]=df["Nationality"].str.upper()
```

4.2 Alineación de elementos similares:

```
Limpiamos los deportes

[ ] df2["Sport"].unique()

[ ] #Arrglamos los deportes
    df2["Sport"]=df2["Sport"].apply(lambda x: x.replace("NBA","BASKETBALL"))
    df2["Sport"]=df2["Sport"].apply(lambda x: x.replace("AUTO RACING (NASCAR)","AUTO RACING"))
    df2["Sport"]=df2["Sport"].apply(lambda x: x.replace("NFL","AMERICAN FOOTBALL"))
    df2["Sport"]=df2["Sport"].apply(lambda x: x.replace("ICE HOCKEY","HOCKEY"))
    df2["Sport"]=df2["Sport"].apply(lambda x: x.replace("ICE HOCKEY","HOCKEY"))
    df2["Sport"]=df2["Sport"].apply(lambda x: x.replace("AMERICAN FOOTBALL / BASEBALL","AMERICAN FOOTBALL"))

# como quedaron los deportes
df2["Sport"].unique()

Caso especial Deion Luwynn Sanders Sr. (9 de agosto de 1967) es un jugador y entrenador de fútbol americano, jugador de béisbol y comentarista deportivo estadounidense.1 Es el único deportista en disputar al mismo tiempo una super Bowl y una Serie Mundial.
```

4.3 Limpieza de la columna Previous Year Rank:

```
Limpiamos la columna Previous Year Rank

df2.info()

[ ] # Los NaN de la columna Previous Year Rank los convertimos en 0
    df2["Previous Year Rank"].fillna(0, inplace=True)

[ ] #Limpiamos valores no numericos con 0
    df2["Previous Year Rank"][df2["Previous Year Rank"]=="not ranked"]=0
    df2["Previous Year Rank"][df2["Previous Year Rank"]=="?"]=0
    df2["Previous Year Rank"][df2["Previous Year Rank"]=="?"]=0
    df2["Previous Year Rank"][df2["Previous Year Rank"]==">30"]=31
    df2["Previous Year Rank"][df2["Previous Year Rank"]==">40"]=41
    df2["Previous Year Rank"][df2["Previous Year Rank"]==">10"]=11
    df2["Previous Year Rank"][df2["Previous Year Rank"]==">20"]=21
    df2["Previous Year Rank"][df2["Previous Year Rank"]==">14"]=15
    df2["Previous Year Rank"][df2["Previous Year Rank"]==">100"]=101
    df2["Previous Year Rank"][df2["Previous Year Rank"]=="none"]=0

[ ] df2["Previous Year Rank"].unique()
```


4.4 Limpieza de la columna S.NO:

```
Limpiamos la columna S.NO

[15] # Borramos la columna S.NO que no tiene valor
df2.drop(columns=["S.NO"], inplace=True)
df2.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Name                   301 non-null   object
1   Nationality            301 non-null   object
2   Current Rank           301 non-null   int64
3   Previous Year Rank     301 non-null   object
4   Sport                  301 non-null   object
5   Year                   301 non-null   int64
6   earnings ($ million)   301 non-null   float64
dtypes: float64(1), int64(2), object(4)
memory usage: 16.6+ KB
```

Se retira la columna S.NO que contiene valores nulos

4.5 Agregamos una columna de tipo date time con formato año

```
Agregamos una columna de tipo datetime con formato año

[190] #Agregamos la columna year_dt como datetime64[ns] en formato de year_dt
df2['year_dt'] = pd.to_datetime(df2['Year'], format='%Y')

#Cambiamos el formato a año
df2["year_dt"] = pd.DatetimeIndex(df2["year_dt"]).year

[191] # Veamos los valores de la columna de fecha
df2["year_dt"].unique()

array([1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
       2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012,
       2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020])
```

Homologación de las fechas para mejor manejo de las series de tiempo

4.6 Limpiamos la columna Name = Nombre del Jugador y “nacionalidad”

```
Limpiamos la columna Name = Nombre del Jugador

[148] df2["Name"].sort_values(ascending=True).unique()

[149] # Arreglamos los nombres
df2["Name"] = df2["Name"].apply(lambda x: x.replace("AARON ROGERS", "AARON RODGERS"))

[150] df2["Name"].sort_values(ascending=True).unique()
```

Solo aparece un nombre escrito de diferente forma, teóricamente siendo de un mismo jugador

La columna nacionalidad

```
[187] df2["Nationality"].sort_values(ascending=True).unique()
# Todo en orden

array(['ARGENTINA', 'AUSTRALIA', 'AUSTRIA', 'BRAZIL', 'CANADA',
       'DOMINICAN', 'FILIPINO', 'FINLAND', 'FRANCE', 'GERMANY', 'IRELAND',
       'ITALY', 'MEXICO', 'NORTHERN IRELAND', 'PHILIPPINES', 'PORTUGAL',
       'RUSSIA', 'SERBIA', 'SPAIN', 'SWITZERLAND', 'UK', 'USA'],
      dtype=object)

[188] #Pasamos el rank a int
df2["Current Rank"]=df2["Current Rank"].astype('int64')
df2["Current Rank"]

[189] #Pasamos el rank anterior a entero
df2["Previous Year Rank"]=df2["Previous Year Rank"].astype('int64')
df2["Previous Year Rank"]

[154] # creamos la columna rank_improve , si =1 su rank mejoro
df2["rank_improve"]=np.where(
    (df2["Current Rank"]< df2["Previous Year Rank"]),
    1,
    0
)

[155] df2["c"]=1
```

Primero se ordenan los nombres de las nacionalidades (A-Z), se convierten valores de str-int y viceversa para las columnas “current rank” “previous year rank”

4.7 Integración de la data frame final

Copiamos al df final ya limpio

```
df3=df2.copy()
df3
```

	Name	Nationality	Current Rank	Previous Year Rank	Sport	Year	earnings (\$ million)	year_dt	rank_improve	c
0	MIKE TYSON	USA	1	0	BOXING	1990	28.6	1990	0	1
1	BUSTER DOUGLAS	USA	2	0	BOXING	1990	26.0	1990	0	1
2	SUGAR RAY LEONARD	USA	3	0	BOXING	1990	13.0	1990	0	1
3	AYRTON SENNA	BRAZIL	4	0	AUTO RACING	1990	10.0	1990	0	1
4	ALAIN PROST	FRANCE	5	0	AUTO RACING	1990	9.0	1990	0	1
...
296	STEPHEN CURRY	USA	6	9	BASKETBALL	2020	74.4	2020	1	1
297	KEVIN DURANT	USA	7	10	BASKETBALL	2020	63.9	2020	1	1
298	TIGER WOODS	USA	8	11	GOLF	2020	62.3	2020	1	1
299	KIRK COUSINS	USA	9	101	AMERICAN FOOTBALL	2020	60.5	2020	1	1
300	CARSON WENTZ	USA	10	101	AMERICAN FOOTBALL	2020	59.1	2020	1	1

301 rows x 10 columns

5 PROPÓSITO DEL DOCUMENTO

5.1Deportistas cuyo ranking ha subido al menos dos lugares entre 2010 y 2020.

Deportistas cuyo ranking ha subido al menos dos lugares entre 2010 y 2020.

```
[158] df2.to_csv("/content/drive/MyDrive/Colab Notebooks/Entregas/Proyecto final/datos_limpios.csv")

[159] players=df2["Name"].sort_values(ascending=True).unique()
      players

[160] def crecimiento_jugador(jugadores, datos):
      list_of_dic={}
      crecimiento=0
      #revisamos jugador por jugador
      for jugador in jugadores:
          # Iteramos en los datos del jugador en años mayores a 2010
          for index, row in datos[(datos["Name"]==jugador) & (datos["year_dt"]>=2010)].iterrows():
              current=int(row["Current Rank"])
              previous=int(row["Previous Year Rank"])
              #Si el rank actual es menos al pasado, subio de rank
              if current<previous:
                  crecimiento=crecimiento+1
              if crecimiento>2:
                  list_of_dic[jugador]=crecimiento
              crecimiento=0
      #creamos un dataframe
      data_frame=pd.DataFrame(list(list_of_dic.items()), columns=["jugador","crecimiento_rankin"])
      return data_frame
```

Se ordena una copia del dataframe ordenándose de forma descendente. Se crea una función que devuelva el crecimiento que ha tenido el jugador en el periodo señalado

```
[161] upRankin_df=crecimiento_jugador(players,df2)
      upRankin_df
      #Creación del data frame jugadores-ranking

[162] upRankin_df=upRankin_df.sort_values("crecimiento_rankin",ascending=True)
      upRankin_df
      #Ordenamos el nuevo dataframe en orden ascendente
```

	jugador	crecimiento_rankin
1	FLOYD MAYWEATHER	3
2	KEVIN DURANT	4
0	CRISTIANO RONALDO	5
3	LEBRON JAMES	5
4	LIONEL MESSI	6
5	ROGER FEDERER	7

```
[163] upRankin_df.to_csv("/content/drive/MyDrive/Colab Notebooks/Entregas/Proyecto final/crecimiento_jugadores.csv")
      #Guardamos el dataframe como csv
```

A continuación se asigna una nueva variable con el nombre del jugador y el crecimiento en el ranking

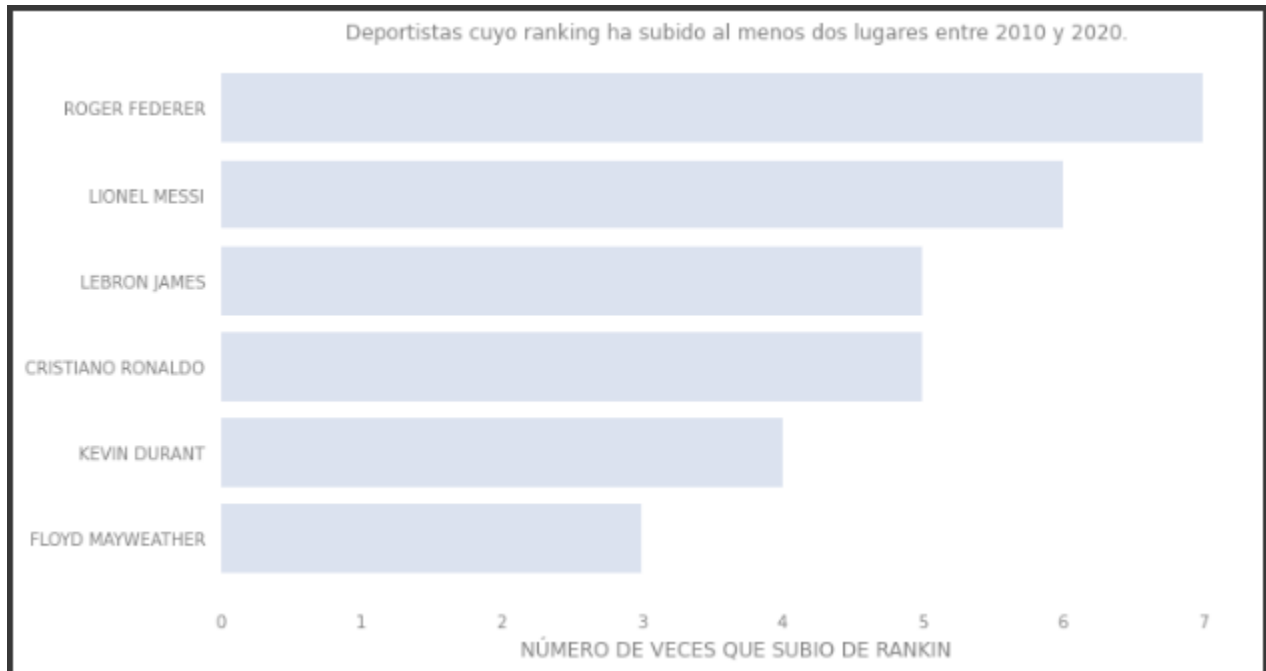
```
[164] #Creación del gráfico
fig = plt.figure(figsize=(11, 6))
plt.barh(upRankin_df["jugador"], upRankin_df["crecimiento_rankin"], align='center', alpha=0.2, edgecolor = "none")
plt.box(False)

#Títulos
plt.title("Deportistas cuyo ranking ha subido al menos dos lugares entre 2010 y 2020.", color="#808080")
plt.xlabel("NÚMERO DE VECES QUE SUBIO DE RANKIN", color="#808080")

#Etiquetas
plt.xticks(fontsize=10, color="#808080")
plt.yticks(fontsize=10, color="#808080")

plt.show()
```

Se crea el grafico con los deportistas que han ascendido al menos dos ocasiones en el ranking



5.2 Atleta con el menor número de apariciones y mayores ganancias.

▼ Atleta con el menor número de apariciones y mayores ganancias.

```
[165] df_agrupado = df2.groupby(["Name"])[["c", "earnings ($ million)"]].sum()
df_agrupado=df_agrupado.sort_values(by=["c","earnings ($ million)"], ascending=(True, False))
df_agrupado.reset_index(inplace=True)
```

```
[166] df_agrupado.head(10)
```

	Name	c	earnings (\$ million)
0	CONOR MCGREGOR	1	99.0
1	CANELO ALVAREZ	1	94.0
2	RUSSELL WILSON	1	89.5
3	KIRK COUSINS	1	60.5
4	MATTHEW STAFFORD	1	59.5
5	CARSON WENTZ	1	59.1
6	NOVAK DJOKOVIC	1	55.8
7	MUHAMMAD ALI	1	55.0
8	CAM NEWTON	1	53.1
9	JORDAN SPIETH	1	52.8

Se agrupa un data frame con la suma de las ganancias, nombre de los jugadores tomando en cuenta el número de ocasiones que aparecen en el ranking (c=1)

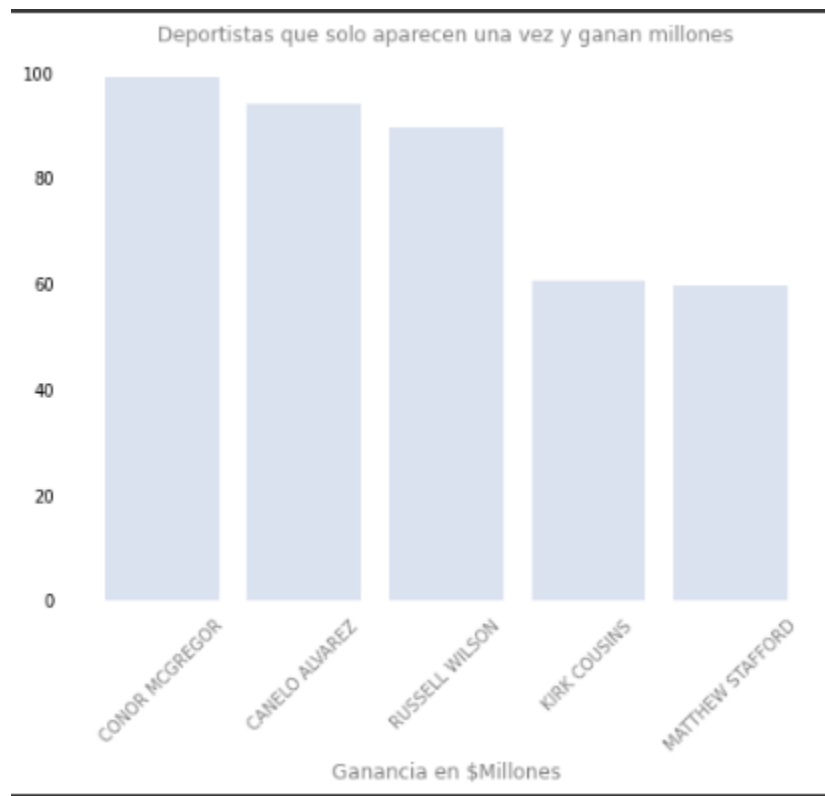
Generamos el gráfico que describe las ganancias por unidad de aparición::

```
fig = plt.figure(figsize=(8, 6))
plt.bar(df_agrupado["Name"].head(5), df_agrupado["earnings ($ million)"].head(5), align='center', alpha=0.2, edgecolor="none")
plt.box(False)

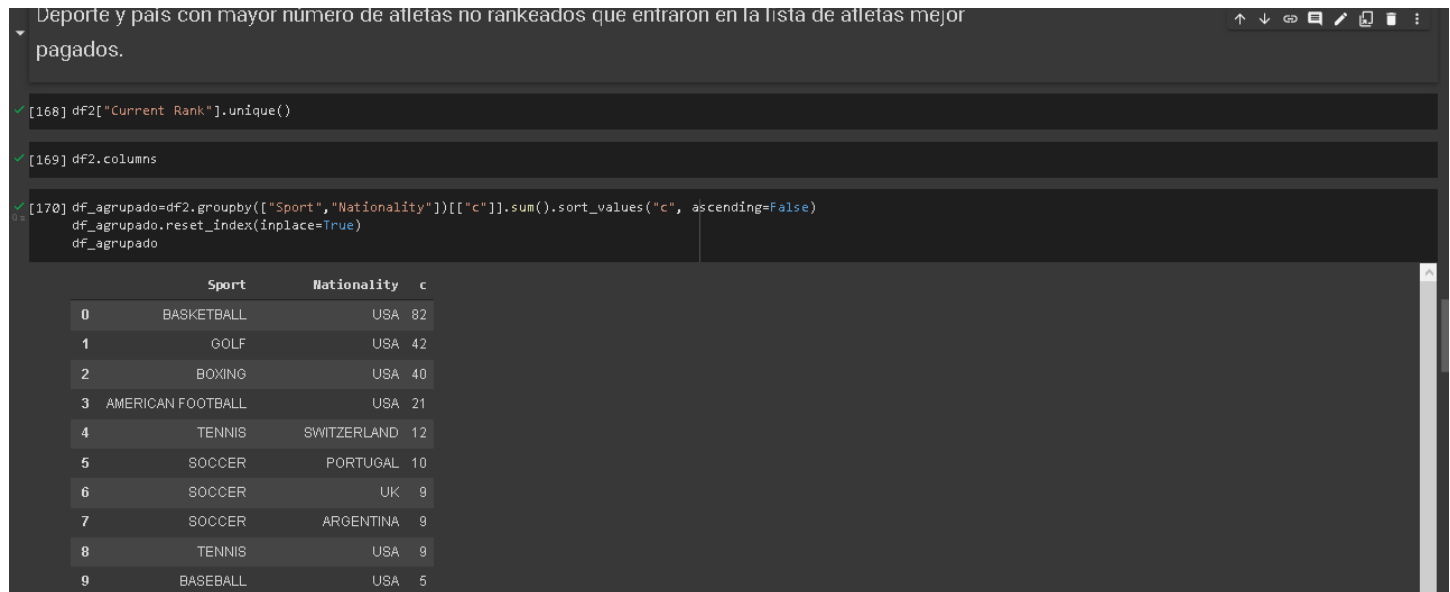
#Títulos
plt.title("Deportistas que solo aparecen una vez y ganan millones", color="#808080")
plt.xlabel("Ganancia en $Millones", color="#808080")

#Etiquetas
plt.xticks(fontsize=10, color="#808080", rotation=45)
plt.yticks(fontsize=10, color="#000000")

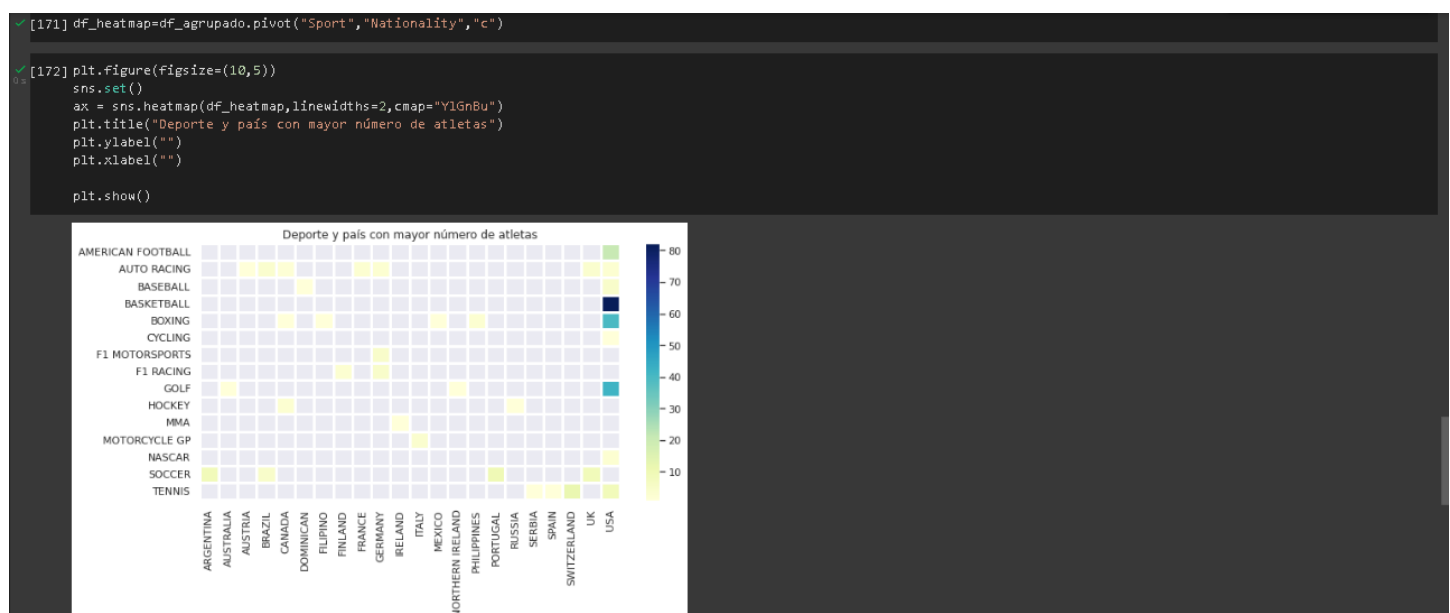
plt.show()
```



5.3Deporte y país con mayor número de atletas no rankeados que entraron en la lista de atletas mejor pagados.



Se genera un data frame agrupado por deporte, nacionalidad y numero de deportistas para esas columnas



Se genera un mapa de calor con la información del nuevo dataframe, donde se observa que la mayor cantidad de jugadores mejor pagados pertenecen al ramo del basketball boxeo y golf, los cuales se desempeñan en Estados Unidos.

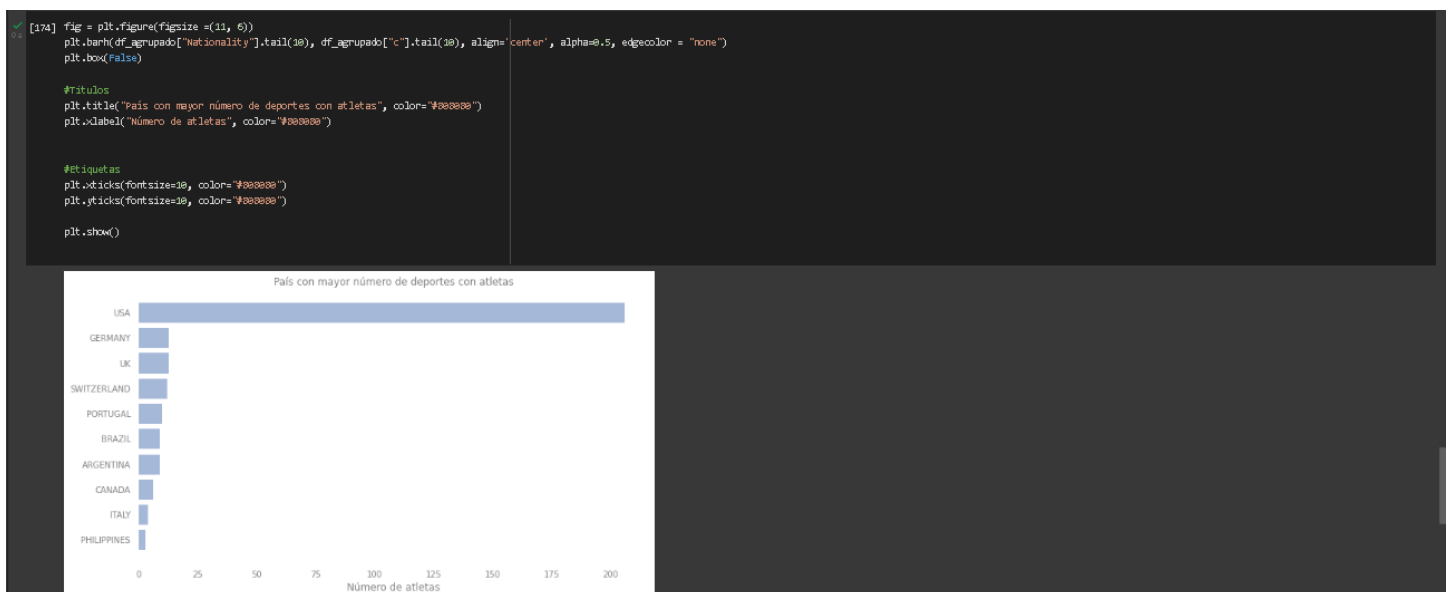
5.4 País con mayor número de deportes con atletas en el dataset.

```
▼ País con mayor número de deportes con atletas en el dataset.

df_agrupado=df2.groupby(["Nationality"])[["c"].sum().sort_values(ascending=True).to_frame()
df_agrupado.reset_index(inplace=True)
df_agrupado
#df_agrupado=df_agrupado.sort_values("c", ascending=True)
```

	Nationality	c
0	IRELAND	1
1	AUSTRALIA	1
2	AUSTRIA	1
3	SPAIN	1
4	DOMINICAN	1
5	FILIPINO	1
6	SERBIA	1

Se agrupa el dataset por nacionalidad y la suma de los jugadores por país



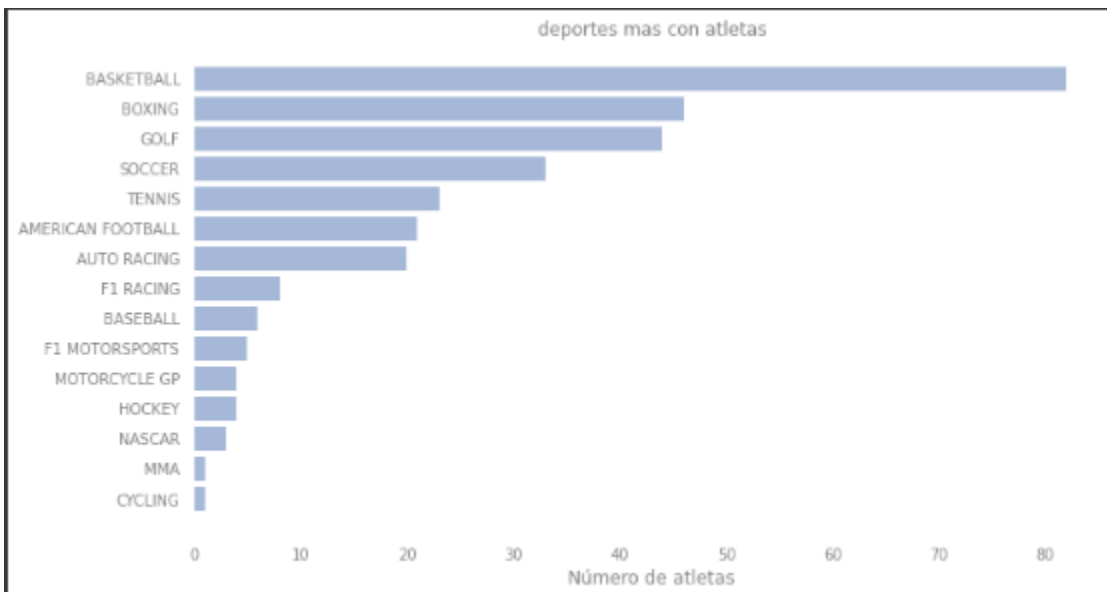
El gráfico muestra que los países con mayor número de atletas dentro del ranking es Estados Unidos, seguido de Alemania y el Reino Unido

5.5 ¿Cuántos atletas pertenecen a cada deporte?

```
¿Cuántos atletas por deporte tiene cada país?  
[175] df_agrupado=df2.groupby(["sport"])[["c"].sum().sort_values(False).to_frame()  
df_agrupado.reset_index(inplace=True)  
df_agrupado  
df_agrupado=df_agrupado.sort_values(["c"], ascending=True)  
df_agrupado
```

	Nationality	c
0	IRELAND	1
1	AUSTRALIA	1
2	AUSTRIA	1
3	SPAIN	1
4	DOMINICAN	1
5	FILIPINO	1
6	SERBIA	1
7	RUSSIA	1
8	MEXICO	1
9	NORTHERN IRELAND	1
11	FRANCE	3
12	PHILIPPINES	3
10	FINLAND	3
13	ITALY	4
14	CANADA	6

Se agrupan los datos por nacionalidad y numero de deportistas, se crea el gráfico con la información del número de atletas en el ranking para cada deporte



El gráfico muestra que el basketball, boxeo y golf encabezan el número de atletas.

5.6 Atletas con mayores ganancias por deporte por década

```
Atleta con mayores ganancias por deporte por década

[186] #creamos los bins
df2['Decada'] = pd.cut(x=df2['Year'],
                      bins=[1988,1999,2000,2010,2020,2030],
                      labels=["1988-1999", "1999-1999", "2000-2009", "2010-2020", "2020-2030"])

[179] #Agrupamos las ganancias por década, deportista y deporte
df_agrupado=df2.groupby(["Name","Sport","Decada"])[["earnings ($ million)"]].sum().to_frame()
#Quitamos los valores en 0
df_agrupado=df_agrupado[df_agrupado["earnings ($ million)"]!=0]
df_agrupado.reset_index(inplace=True)
df_agrupado=df_agrupado.sort_values(["earnings ($ million)","Sport","Name","Decada"], ascending=(False,True, True, True))
df_agrupado.reset_index(inplace=True)
#df_agrupado

[181] #Obtenemos los valores unicos para década y deporte
decadas=df_agrupado["Decada"].unique().sort_values(ascending=True).unique()
deportes=df_agrupado["Sport"].sort_values(ascending=True).unique()
deportes

[181] #Armamos un data frame con los mejores pagados por década, deporte y año
best_dude=pd.DataFrame()
for deporte in deportes:
    for decada in decadas:
        #Por cada deporte de cada década sacamos el deportista mejor pagado y lo agregamos a un dataframe
        best_dude=pd.DataFrame.append((df_agrupado[(df_agrupado["Sport"]==deporte)&(df_agrupado["Decada"]==decada)].sort_values(by=["earnings ($ million)","Decada"],ascending=(False,True)).head(1)),other=best_dude)

[182] #Ordenamos el data frame
best_dude=best_dude.sort_values(["Sport","Decada"], ascending=(True,True))
best_dude
```

En resumen se crean los intervalos de años en décadas, se agrupan los datos de deportistas para los periodos señalados, se retiran los valores nulos. A continuación se crea un dataframe con los mejores pagados por década, deporte y año; se ordena el dataframe

Rankings by decade, sports y año, 66 ordena of data frame							
index		Name		Sport	Decada	earnings (\$ million)	
91	21	DEION SANDERS	AMERICAN FOOTBALL	1990-1999		22.5	
71	81	PEYTON MANNING	AMERICAN FOOTBALL	2000-2009		42.0	
24	0	AARON RODGERS	AMERICAN FOOTBALL	2010-2020		138.3	
28	65	MICHAEL SCHUMACHER	AUTO RACING	1990-1999		123.0	

A continuación, se crea una función que nos proporcione los mejores pagados por deporte en base a las ganancias obtenidas por década

```
[285] def grafica_bestDude(datos, deporte):

    def addLabels(x,y):
        for i in range(len(x)):
            plt.text(i, y[i], x[i], ha = 'center')

    fig = plt.figure(figsize =(6, 3))
    bar_plot =plt.bar(datos["Decada"], datos["earnings ($ million)"], align='center', alpha=0.2, edgecolor = "none")
    plt.box(True)

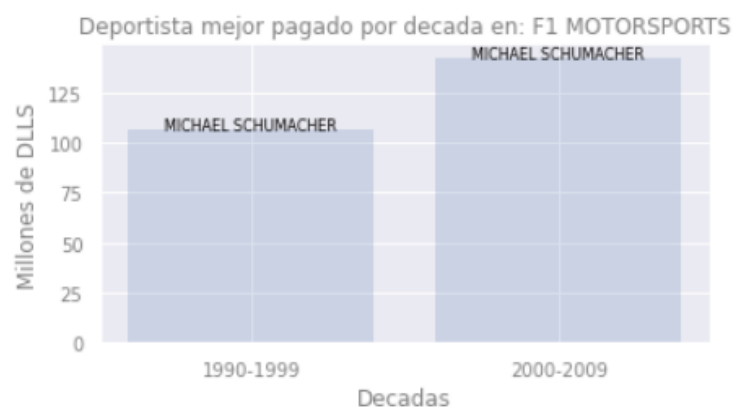
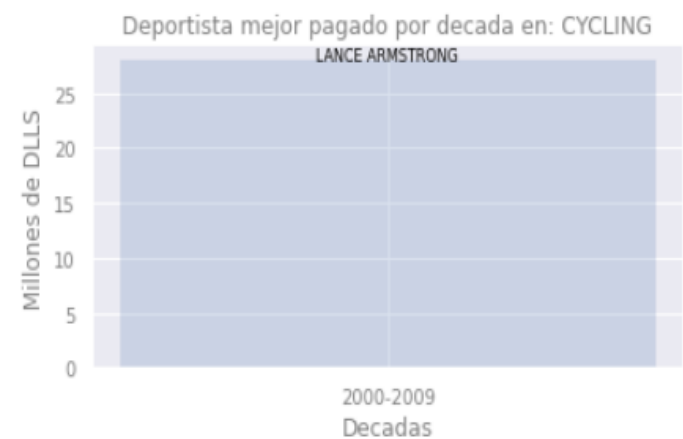
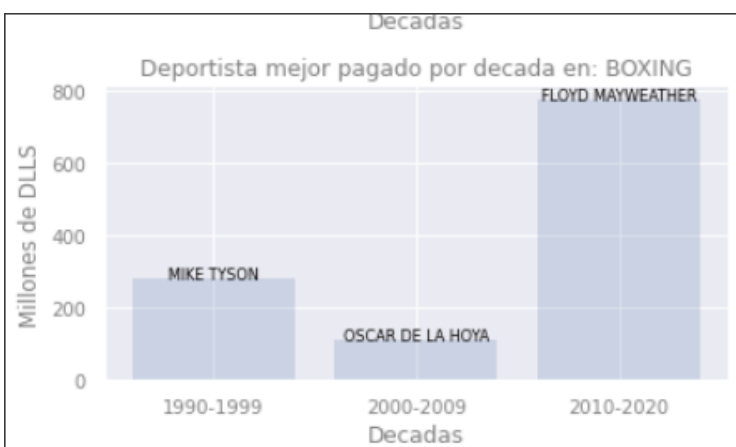
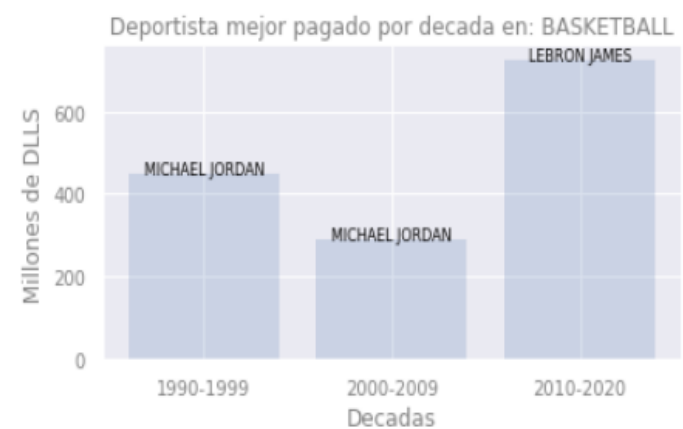
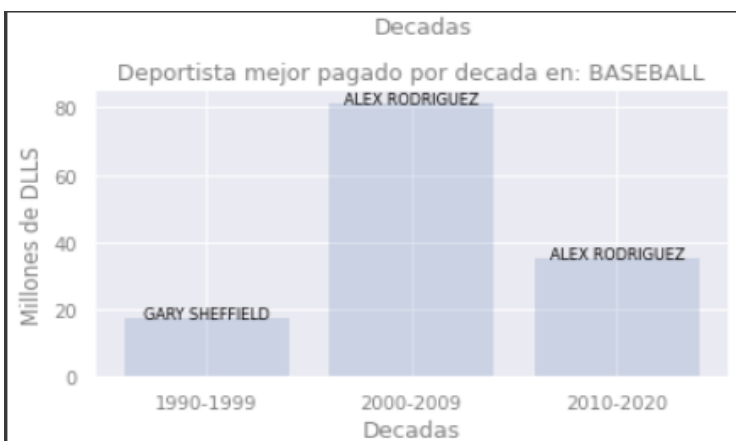
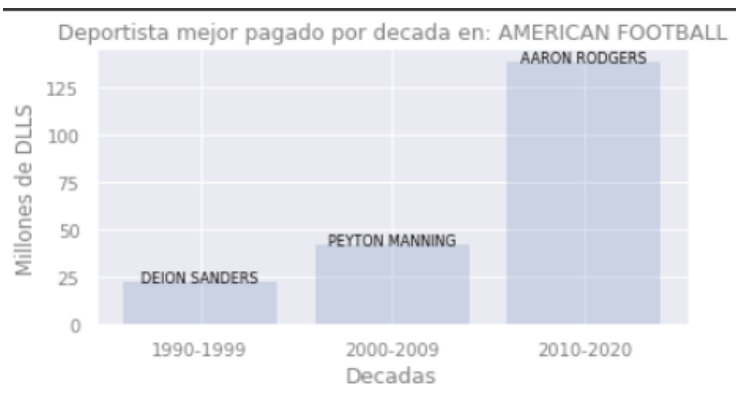
    #Títulos
    plt.title('deportista mejor pagado por decada en: '+ deporte, color='000000')
    plt.xlabel('Decadas', color='000000')
    plt.ylabel('millones de Dóls', color='000000')

    for ind in datos.index:
        plt.text(datos["Decada"][ind],datos["earnings ($ million)"][ind],datos["Name"][ind], ha = 'center', color='000000', fontsize=8)

    #Etiquetas
    plt.xticks(fontsize=10, color='000000')
    plt.yticks(fontsize=10, color='000000')

    plt.show()

for deporte in deportes:
    grafica_bestDude(best_dude[best_dude["Sport"]==deporte],deporte)
```

5.6 Ganancia total por cada deporte por cada año

Ganancia total por cada deporte por cada año

```
[ ] df2.columns

[ ] df_agrupado=df2.groupby(["Sport","year_dt"])["earnings ($ million)"].sum().to_frame()
df_agrupado.reset_index(inplace=True)

[ ] df_agrupado

[ ] deportes=df_agrupado["Sport"].unique()
```

```
] def grafica_deportes(df, deporte):

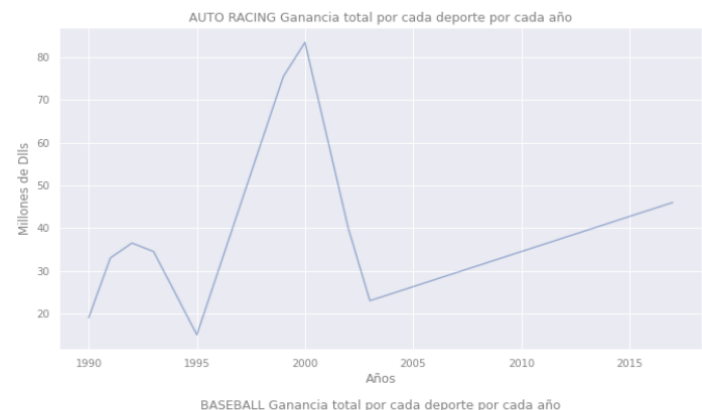
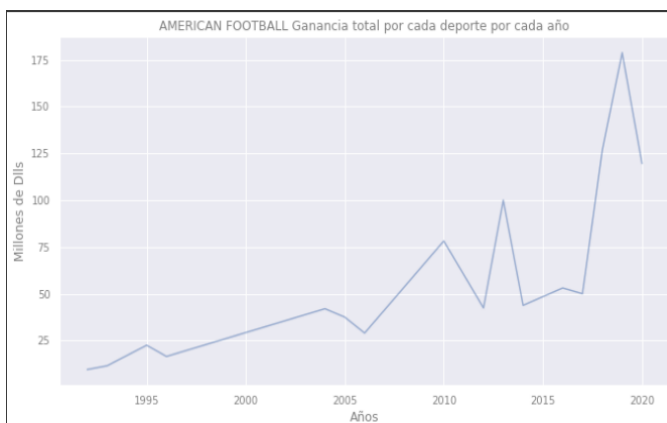
    df_filtro=df[df_agrupado["Sport"]==deporte]
    fig = plt.figure(figsize =(11, 6))
    plt.plot(df_filtro["year_dt"], df_filtro["earnings ($ million)"], alpha=0.5,)
    plt.box(True)
    plt.grid(True)

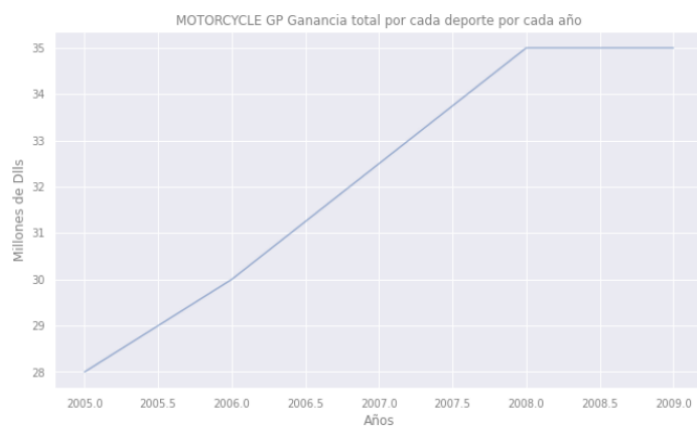
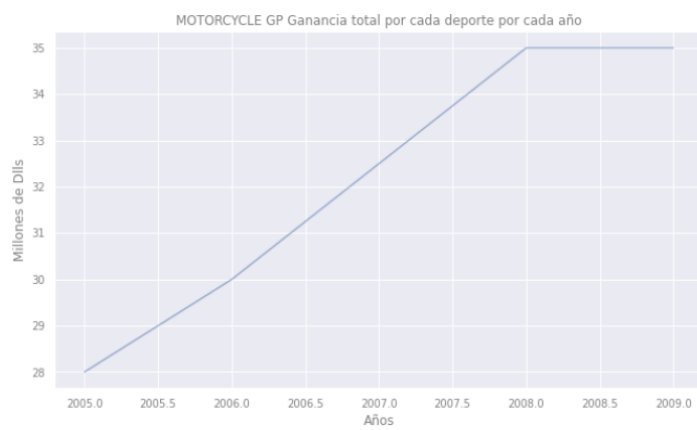
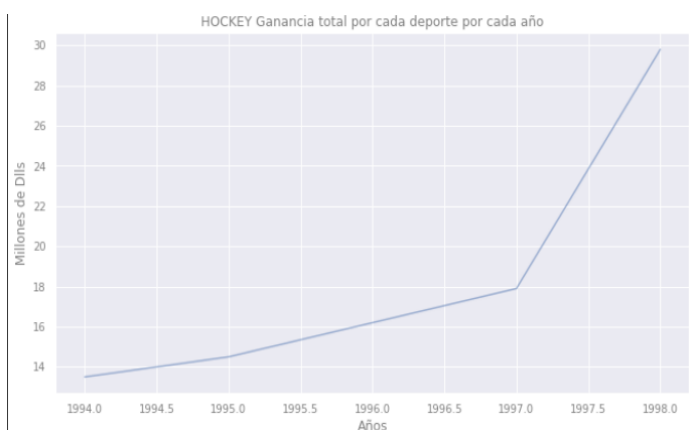
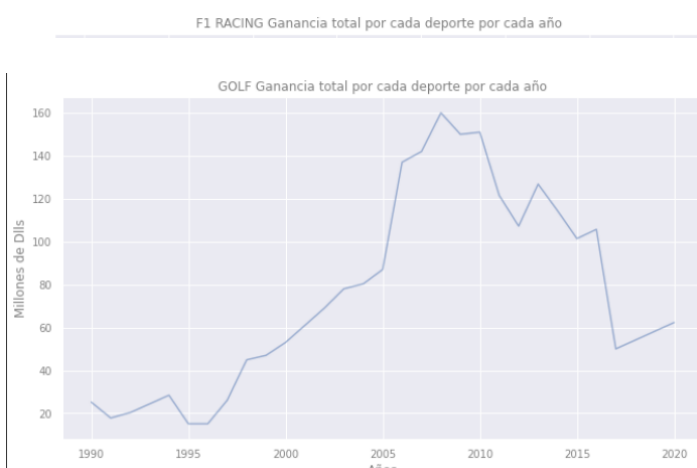
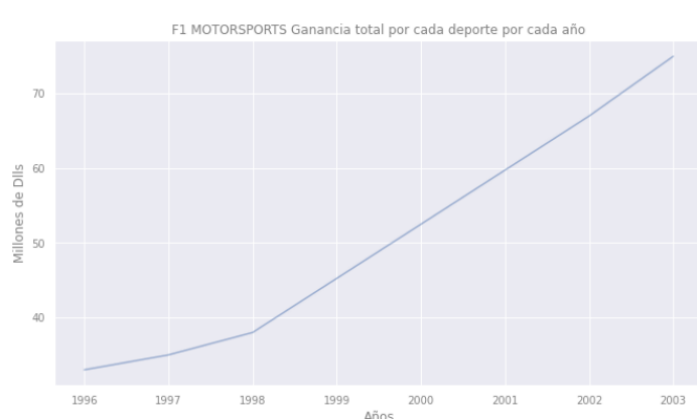
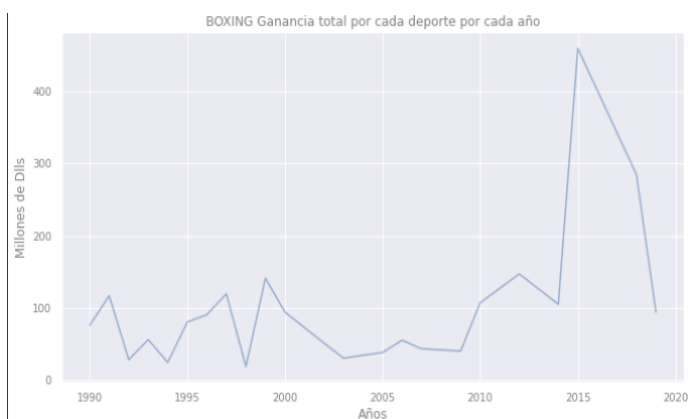
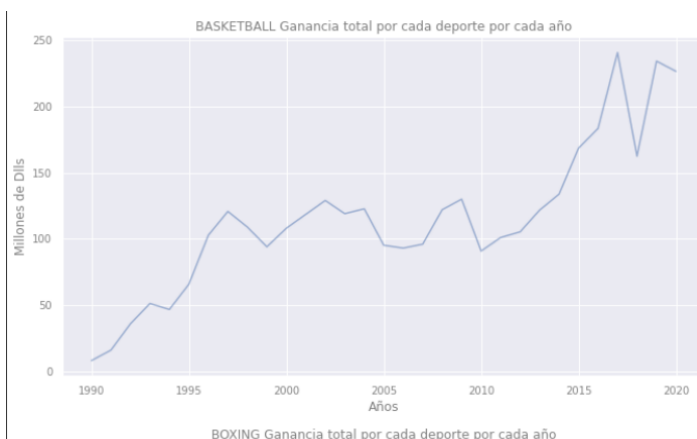
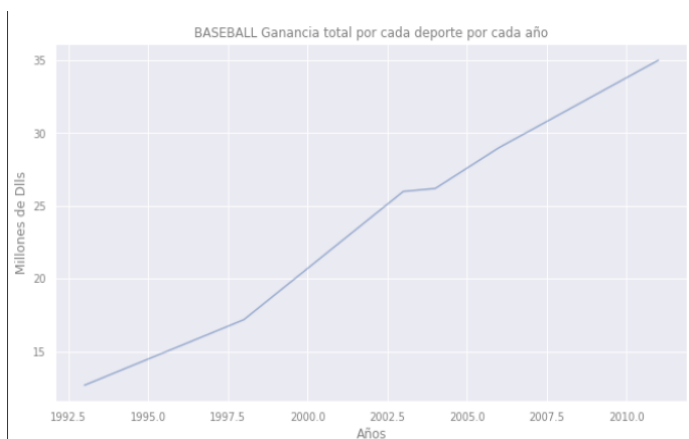
    #Titulos
    plt.title(deporte+" Ganancia total por cada deporte por cada año", color="#808080")
    plt.xlabel("Años", color="#808080")
    plt.ylabel("Millones de Dls", color="#808080")

    #Etiquetas
    plt.xticks(fontsize=10, color="#808080")
    plt.yticks(fontsize=10, color="#808080")

    plt.show()
```

Se agrupan las variables “sport”, “year_dt” y la suma de las ganancias. Luego se genera una función que nos permita realizar un gráfico que muestre las ganancias por cada deporte de cada año





6 DE LOS RESULTADOS EN POWER BI

