Exploratory Data Analysis on the 'Forbes Richest Athletes (1990 - 2020)'Dataset

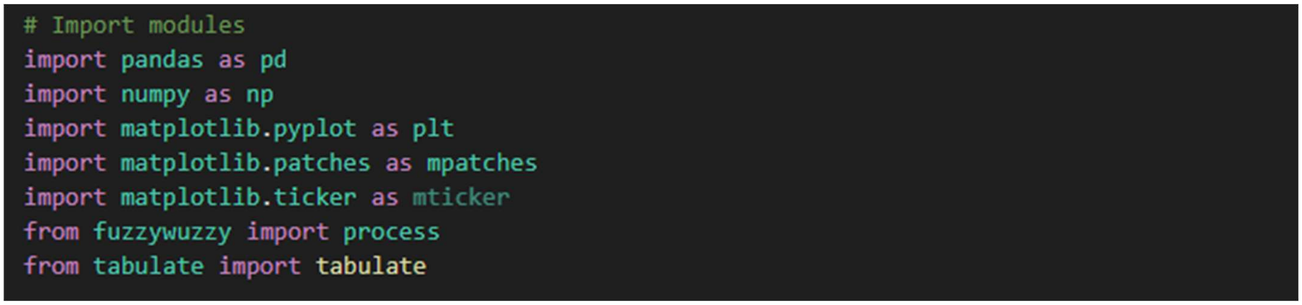
REPORT

Introduction

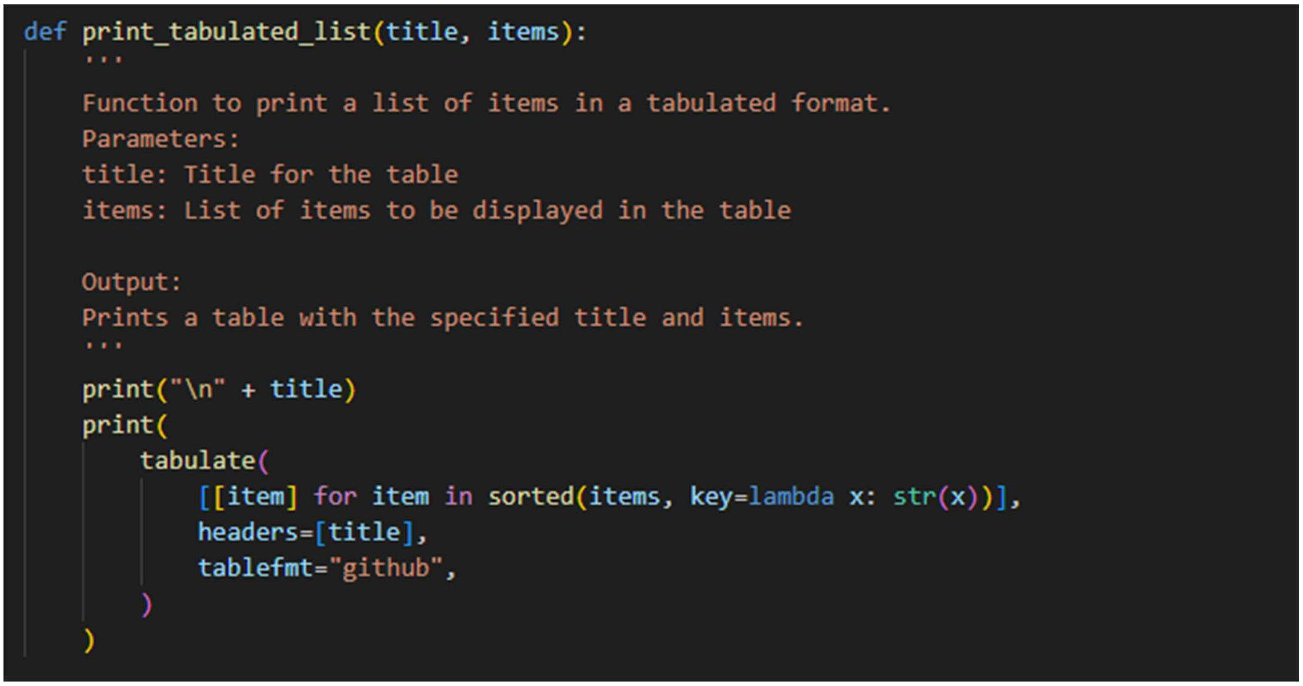
The dataset I have chosen is the 'Forbes Richest Athletes (1990 - 2020)', which shows the top ten athletes by gross annual earnings (USD) for each year from 1990 - 2020. For each athlete in each year, we are given their name, nationality, Current Rank (for that year), Previous Year Rank (if known), their sport, the year the data is for, and their earnings for the year (in millions USD). An analysis of this data would be able to explore general trends of which sports have the highest paid athletes over time, as well as the increase in pay across sports and years.

Data Cleaning

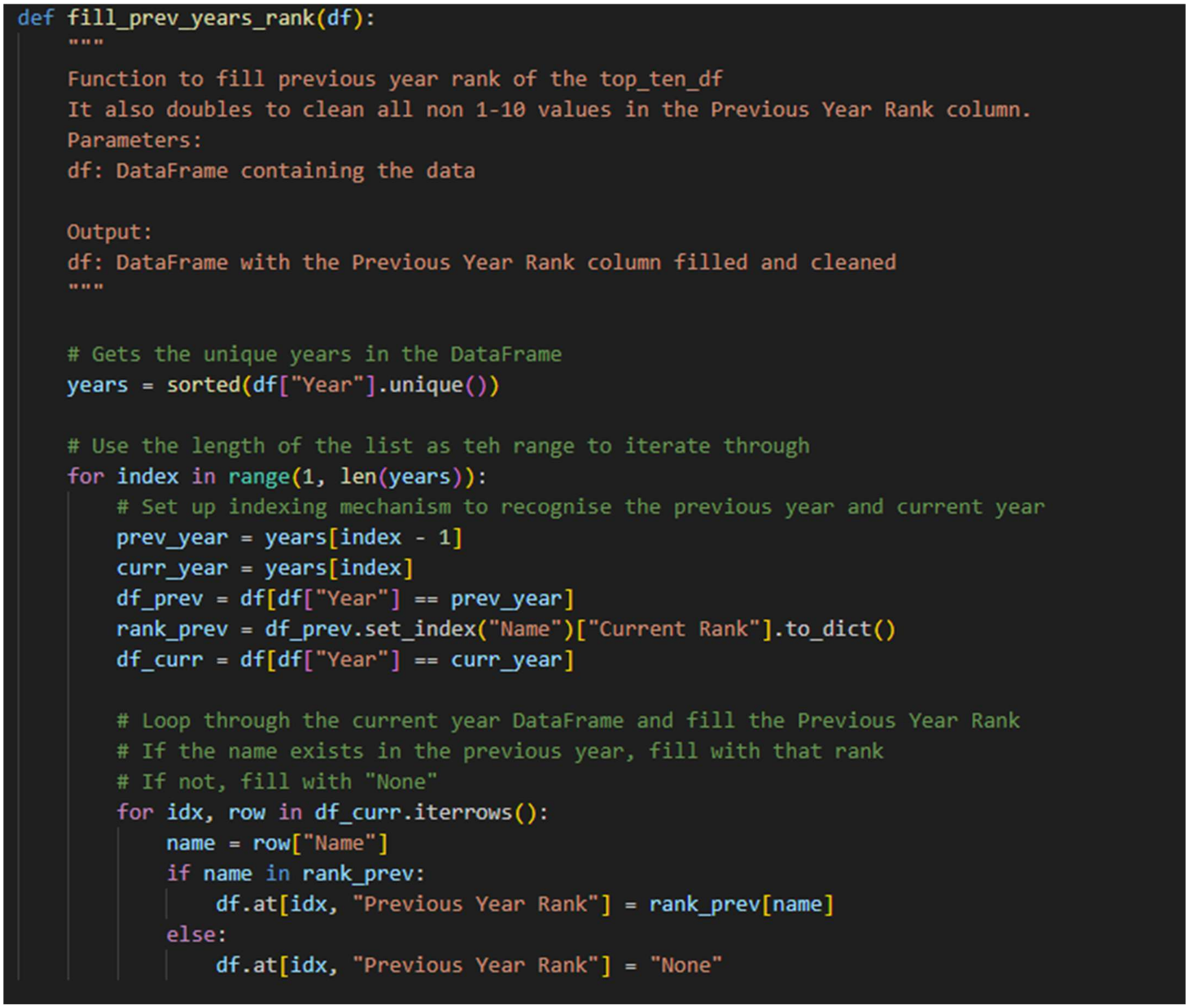
There was a bit of work to do cleaning the dataset, as there were some unique variable names that referred to the same country, as well as some missing data. To handle it effectively, I used a few imported modules and user-defined functions as seen below.



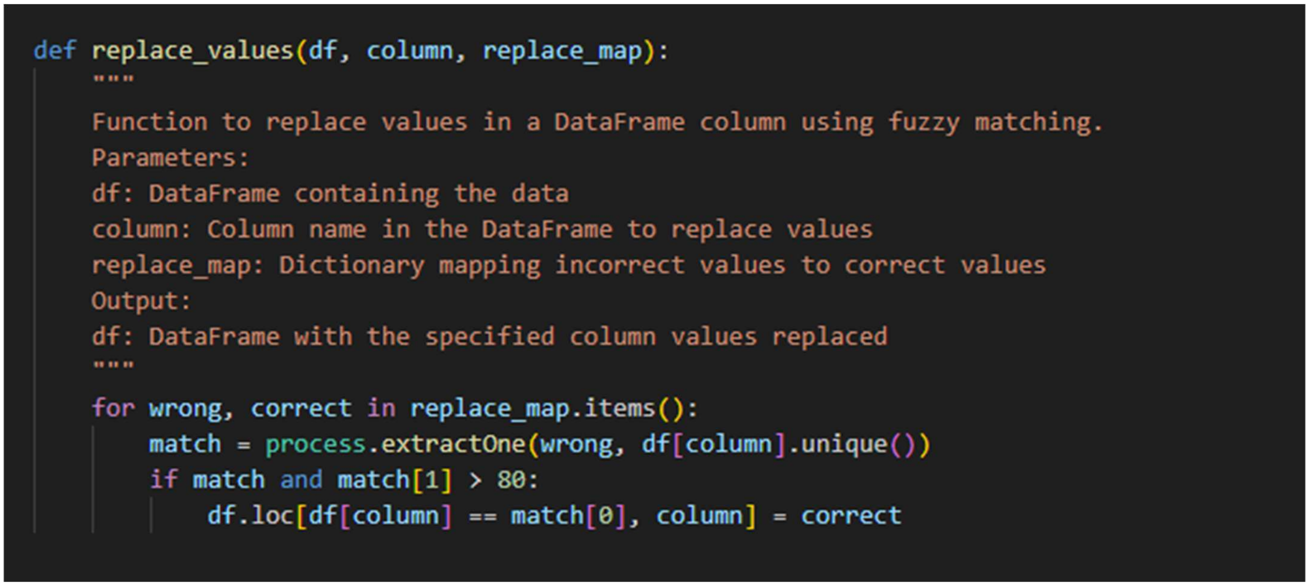
```python
# Import modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import matplotlib.ticker as mticker
from fuzzywuzzy import process
from tabulate import tabulate
```



```python
def print_tabulated_list(title, items):
    '''
    Function to print a list of items in a tabulated format.
    Parameters:
    title: Title for the table
    items: List of items to be displayed in the table

    Output:
    Prints a table with the specified title and items.
    '''
    print("\n" + title)
    print(
        tabulate(
            [[item] for item in sorted(items, key=lambda x: str(x))],
            headers=[title],
            tablefmt="github",
        )
    )
```

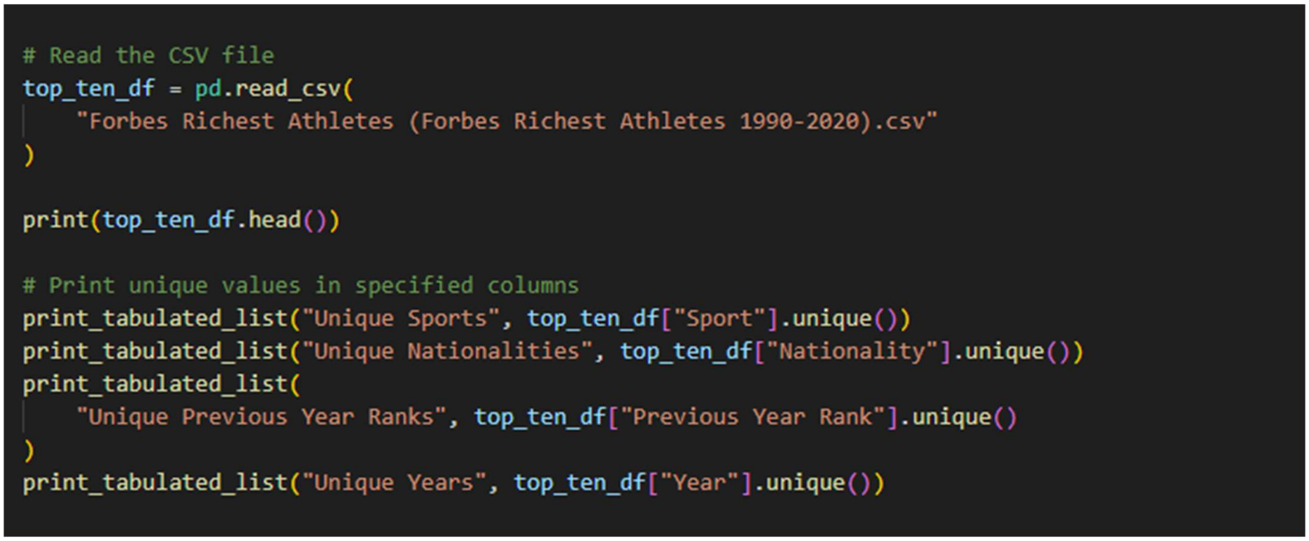


```python
def fill_prev_years_rank(df):
    """

    Function to fill previous year rank of the top_ten_df
    It also doubles to clean all non 1-10 values in the Previous Year Rank column.
    Parameters:
    df: DataFrame containing the data

    Output:
    df: DataFrame with the Previous Year Rank column filled and cleaned
    """

    # Gets the unique years in the DataFrame
    years = sorted(df["Year"].unique())

    # Use the length of the list as teh range to iterate through
    for index in range(1, len(years)):
        # Set up indexing mechanism to recognise the previous year and current year
        prev_year = years[index - 1]
        curr_year = years[index]
        df_prev = df[df["Year"] == prev_year]
        rank_prev = df_prev.set_index("Name")["Current Rank"].to_dict()
        df_curr = df[df["Year"] == curr_year]

        # Loop through the current year DataFrame and fill the Previous Year Rank
        # If the name exists in the previous year, fill with that rank
        # If not, fill with "None"
        for idx, row in df_curr.iterrows():
            name = row["Name"]
            if name in rank_prev:
                df.at[idx, "Previous Year Rank"] = rank_prev[name]
            else:
                df.at[idx, "Previous Year Rank"] = "None"
```



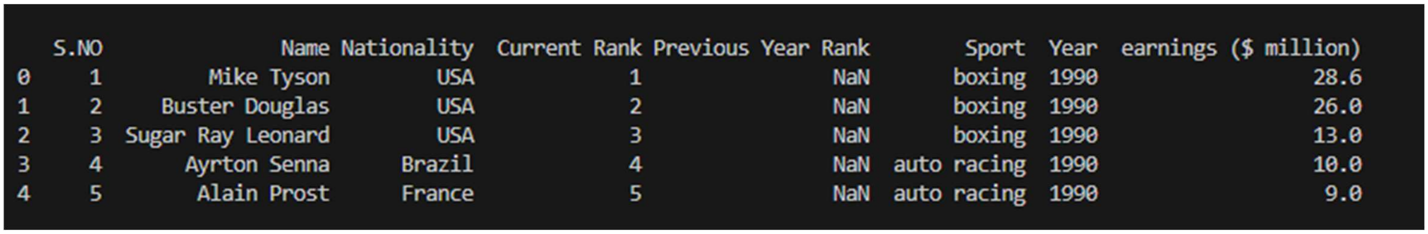
```python
def replace_values(df, column, replace_map):
    """

    Function to replace values in a DataFrame column using fuzzy matching.
    Parameters:
    df: DataFrame containing the data
    column: Column name in the DataFrame to replace values
    replace_map: Dictionary mapping incorrect values to correct values
    Output:
    df: DataFrame with the specified column values replaced
    """

    for wrong, correct in replace_map.items():
        match = process.extractOne(wrong, df[column].unique())
        if match and match[1] > 80:
            df.loc[df[column] == match[0], column] = correct
```

Let’s read in the data, print first 5 rows, and check columns for unique values to see what we are working with:

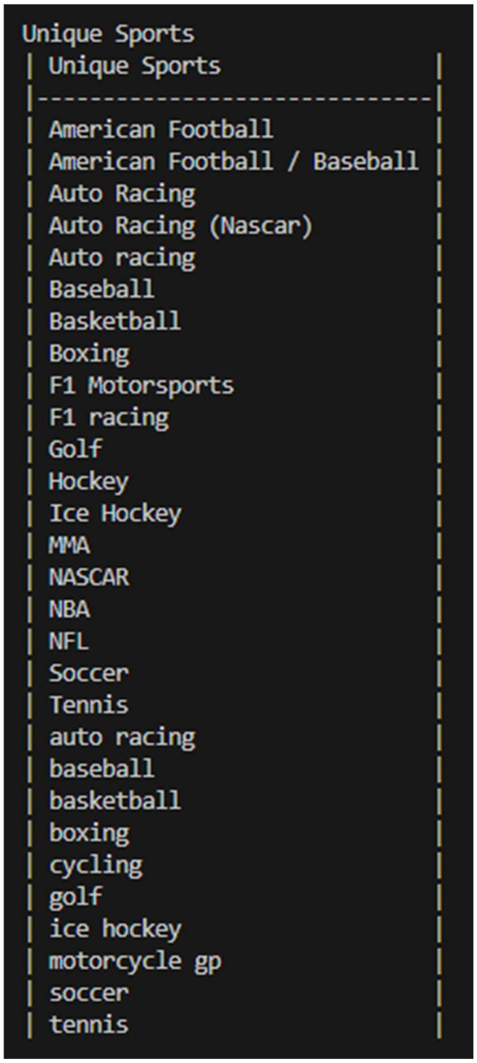


```
# Read the CSV file
top_ten_df = pd.read_csv(
    "Forbes Richest Athletes (Forbes Richest Athletes 1990-2020).csv"
)

print(top_ten_df.head())

# Print unique values in specified columns
print_tabulated_list("Unique Sports", top_ten_df["Sport"].unique())
print_tabulated_list("Unique Nationalities", top_ten_df["Nationality"].unique())
print_tabulated_list(
    "Unique Previous Year Ranks", top_ten_df["Previous Year Rank"].unique()
)
print_tabulated_list("Unique Years", top_ten_df["Year"].unique())
```

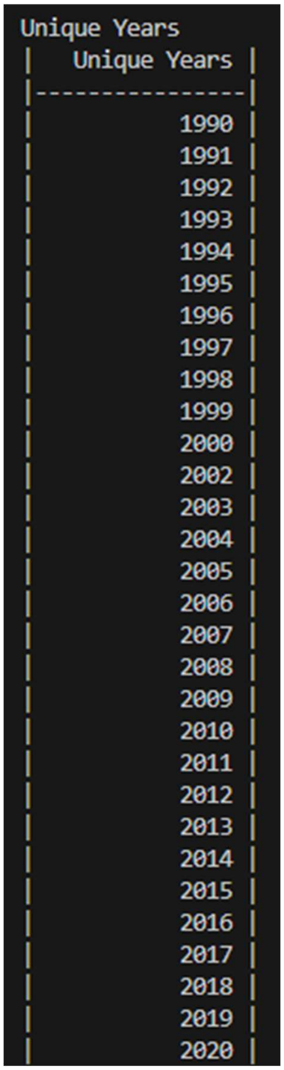
OUTPUTS:



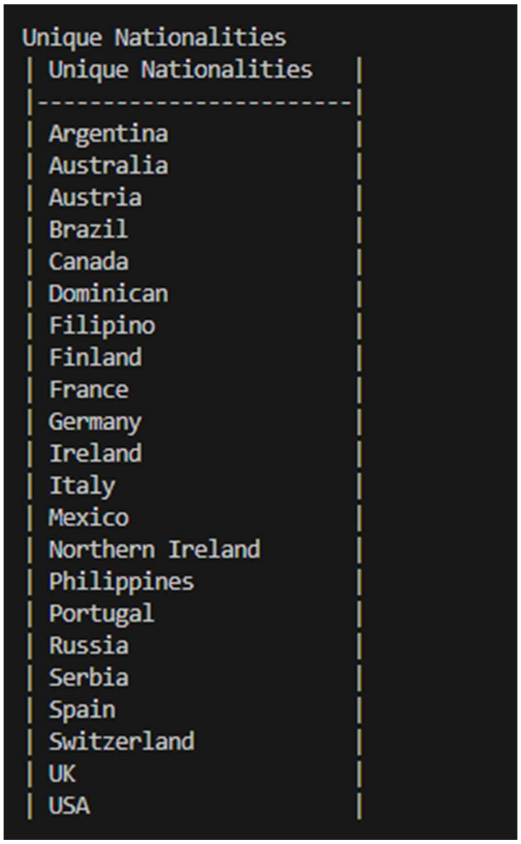
```
   S.NO              Name Nationality  Current Rank Previous Year Rank        Sport  Year  earnings ($ million)
0     1        Mike Tyson         USA             1                NaN       boxing  1990                  28.6
1     2    Buster Douglas         USA             2                NaN       boxing  1990                  26.0
2     3  Sugar Ray Leonard        USA             3                NaN       boxing  1990                  13.0
3     4      Ayrton Senna      Brazil             4                NaN  auto racing  1990                  10.0
4     5       Alain Prost      France             5                NaN  auto racing  1990                   9.0
```



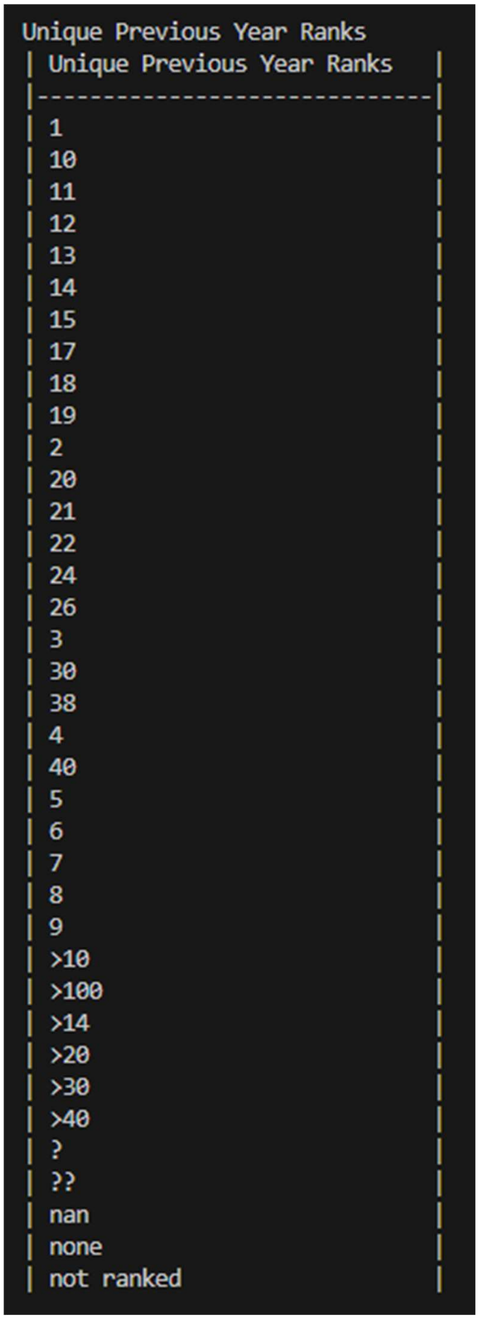
```
Unique Sports
| Unique Sports                 |
|-------------------------------|
| American Football             |
| American Football / Baseball  |
| Auto Racing                   |
| Auto Racing (Nascar)          |
| Auto racing                   |
| Baseball                      |
| Basketball                    |
| Boxing                        |
| F1 Motorsports                |
| F1 racing                     |
| Golf                          |
| Hockey                        |
| Ice Hockey                    |
| MMA                           |
| NASCAR                        |
| NBA                           |
| NFL                           |
| Soccer                        |
| Tennis                        |
| auto racing                   |
| baseball                      |
| basketball                    |
| boxing                        |
| cycling                       |
| golf                          |
| ice hockey                    |
| motorcycle gp                 |
| soccer                        |
| tennis                        |
```



```
Unique Years
|   Unique Years |
|----------------|
|           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 |
```

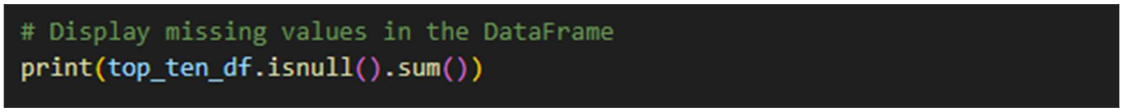


```
Unique Nationalities
| Unique Nationalities    |
|-------------------------|
| Argentina               |
| Australia               |
| Austria                 |
| Brazil                  |
| Canada                  |
| Dominican               |
| Filipino                |
| Finland                 |
| France                  |
| Germany                 |
| Ireland                 |
| Italy                   |
| Mexico                  |
| Northern Ireland        |
| Philippines             |
| Portugal                |
| Russia                  |
| Serbia                  |
| Spain                   |
| Switzerland             |
| UK                      |
| USA                     |
```



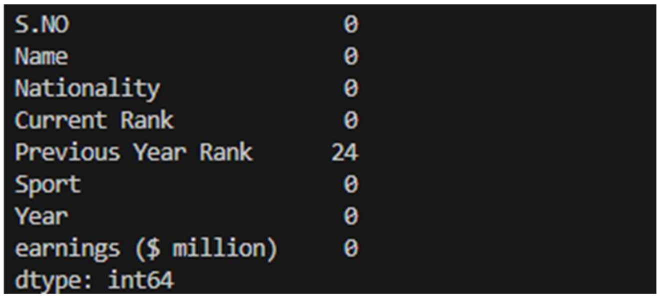
```
Unique Previous Year Ranks
 | Unique Previous Year Ranks   |
 |------------------------------|
 | 1                            |
 | 10                           |
 | 11                           |
 | 12                           |
 | 13                           |
 | 14                           |
 | 15                           |
 | 17                           |
 | 18                           |
 | 19                           |
 | 2                            |
 | 20                           |
 | 21                           |
 | 22                           |
 | 24                           |
 | 26                           |
 | 3                            |
 | 30                           |
 | 38                           |
 | 4                            |
 | 40                           |
 | 5                            |
 | 6                            |
 | 7                            |
 | 8                            |
 | 9                            |
 | >10                          |
 | >100                         |
 | >14                          |
 | >20                          |
 | >30                          |
 | >40                          |
 | ?                            |
 | ??                           |
 | nan                          |
 | none                         |
 | not ranked                   |
```

There are quite a few variable names we need to handle in the ‘Sport’ column, a few of which can be handled by the lower() function, and a couple we need to handle in the Nationalities column. The ‘Previous Year Ranked’ column is the worst, with lots of unique names for when athletes are not on the list. The year 2001 is also missing from the dataset and prompts a check of other potential missing data from the set.



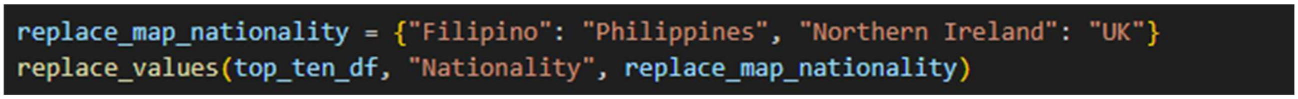
```python
# Display missing values in the DataFrame
print(top_ten_df.isnull().sum())
```

OUTPUTS:



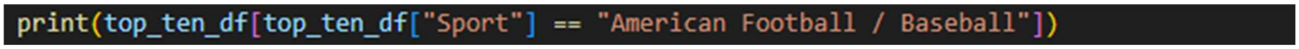
```
S.NO                     0
Name                     0
Nationality              0
Current Rank             0
Previous Year Rank      24
Sport                    0
Year                     0
earnings ($ million)     0
dtype: int64
```

We see that there are also 24 missing values in the ‘Previous Year Rank’ column. These, along with the missing 2001 athletes, will be investigated later in this report. For now, we make use of the replace_values function to fix up the few errant country names. It should be noted, that Northern Ireland is a part of the UK, not Ireland; a common mistake that should not be made.

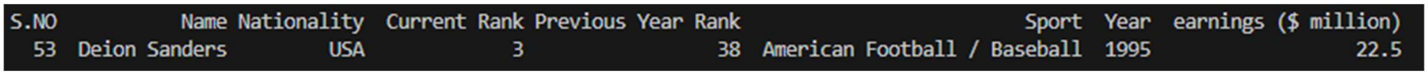


```
replace_map_nationality = {"Filipino": "Philippines", "Northern Ireland": "UK"}
replace_values(top_ten_df, "Nationality", replace_map_nationality)
```

Looking at the Sports column, a particular entry has both American football and baseball.



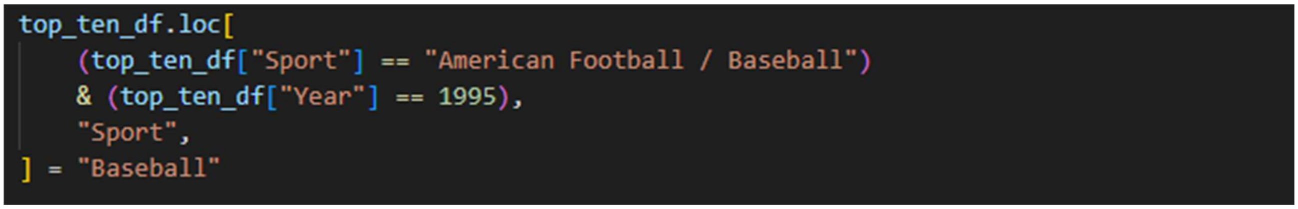
```
print(top_ten_df[top_ten_df["Sport"] == "American Football / Baseball"])
```



```
S.NO           Name Nationality  Current Rank Previous Year Rank                         Sport  Year  earnings ($ million)
  53  Deion Sanders         USA             3                 38  American Football / Baseball  1995                  22.5
```

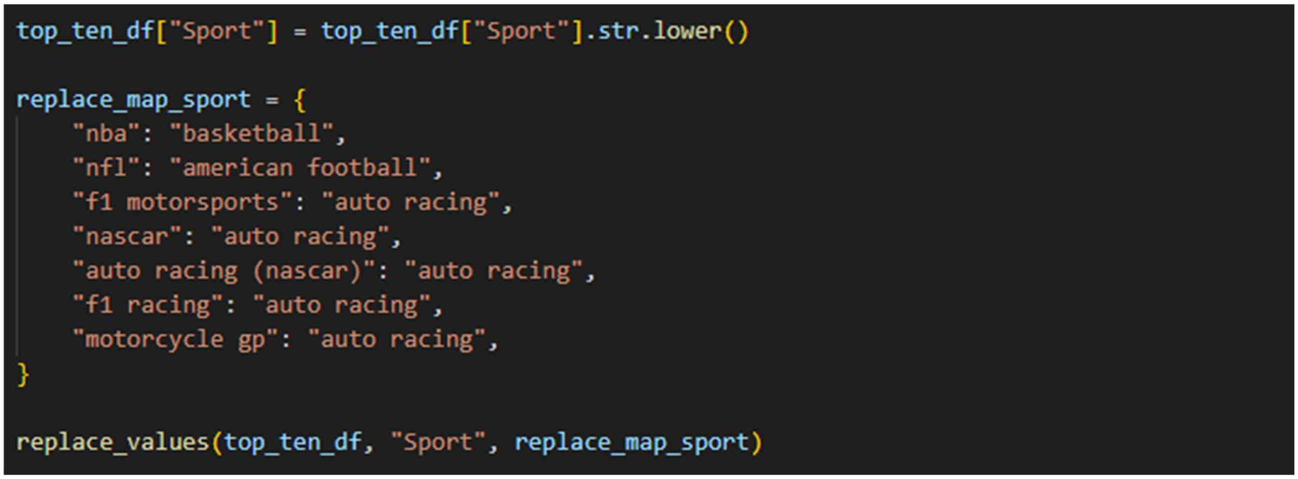
After some external research on Deion Sanders in 1995, we find that most of his money came from baseball that year, so for analysis sakes I will change his Sport to baseball.

Source: https://overthecap.com/player/deion-sanders/8578



```
top_ten_df.loc[
    (top_ten_df["Sport"] == "American Football / Baseball")
    & (top_ten_df["Year"] == 1995),
    "Sport",
] = "Baseball"
```

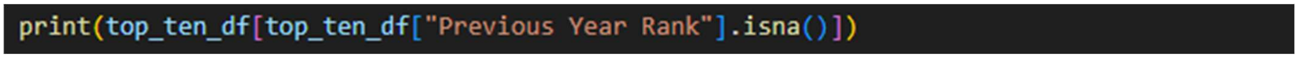
The sports column is still very messy, and we will use the same replace_values function to clean it up. Due to the lack of distinguishment in auto racing in the earlier years of data, we will classify all motorsports as “auto racing”. We will also use the ‘lower()’ function to remove some of the spelling duplicates.



```
top_ten_df["Sport"] = top_ten_df["Sport"].str.lower()

replace_map_sport = {
    "nba": "basketball",
    "nfl": "american football",
    "f1 motorsports": "auto racing",
    "nascar": "auto racing",
    "auto racing (nascar)": "auto racing",
    "f1 racing": "auto racing",
    "motorcycle gp": "auto racing",
}

replace_values(top_ten_df, "Sport", replace_map_sport)
```

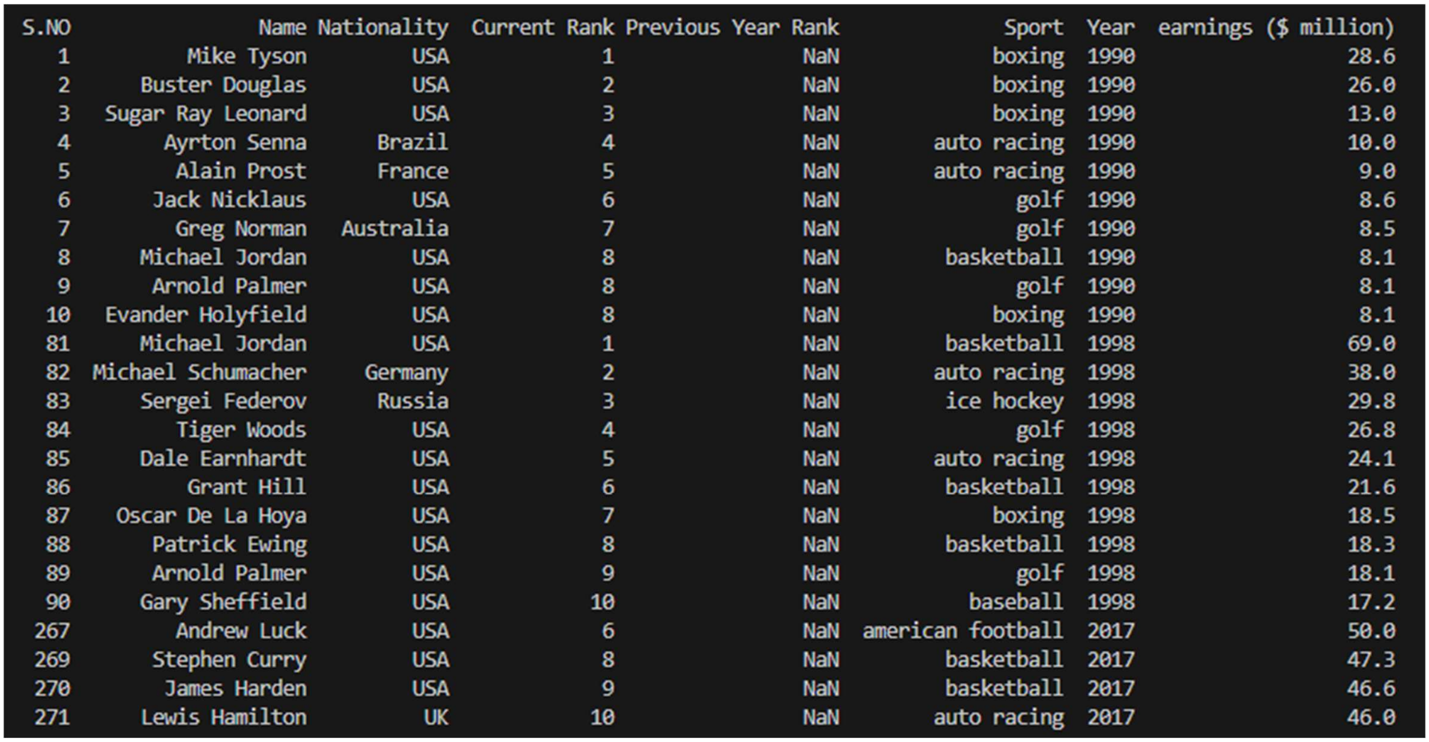
Missing Data

After successfully cleaning the Sport and Nationality column, the very messy ‘Previous Year Rank’ column is next. Not only does it have a variety of names for athletes not on the previous years Top Ten, but it has 24 missing values in the column.



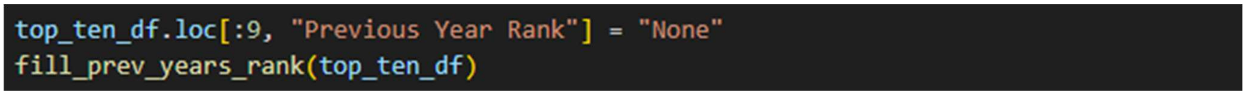
```
print(top_ten_df[top_ten_df["Previous Year Rank"].isna()])
```

As the missing values (see overpage) are directly related to the dataset (hence are Missing at Random), we can impute them ourselves with some logic. The user-defined fill_prev_years_rank function will scan the athletes in a current year, and fill in their “Current Rank” of the previous year, if it exists. Else, it will fill that column with “None”.



```
S.NO              Name Nationality  Current Rank Previous Year Rank              Sport  Year  earnings ($ million)
   1        Mike Tyson         USA             1                NaN             boxing  1990                  28.6
   2    Buster Douglas         USA             2                NaN             boxing  1990                  26.0
   3  Sugar Ray Leonard        USA             3                NaN             boxing  1990                  13.0
   4      Ayrton Senna      Brazil             4                NaN        auto racing  1990                  10.0
   5       Alain Prost      France             5                NaN        auto racing  1990                   9.0
   6      Jack Nicklaus        USA             6                NaN               golf  1990                   8.6
   7       Greg Norman   Australia             7                NaN               golf  1990                   8.5
   8     Michael Jordan        USA             8                NaN         basketball  1990                   8.1
   9      Arnold Palmer        USA             8                NaN               golf  1990                   8.1
  10  Evander Holyfield        USA             8                NaN             boxing  1990                   8.1
  81     Michael Jordan        USA             1                NaN         basketball  1998                  69.0
  82 Michael Schumacher    Germany             2                NaN        auto racing  1998                  38.0
  83     Sergei Federov     Russia             3                NaN         ice hockey  1998                  29.8
  84        Tiger Woods        USA             4                NaN               golf  1998                  26.8
  85     Dale Earnhardt        USA             5                NaN        auto racing  1998                  24.1
  86         Grant Hill        USA             6                NaN         basketball  1998                  21.6
  87   Oscar De La Hoya        USA             7                NaN             boxing  1998                  18.5
  88     Patrick Ewing         USA             8                NaN         basketball  1998                  18.3
  89      Arnold Palmer        USA             9                NaN               golf  1998                  18.1
  90     Gary Sheffield        USA            10                NaN           baseball  1998                  17.2
 267        Andrew Luck        USA             6                NaN  american football  2017                  50.0
 269      Stephen Curry        USA             8                NaN         basketball  2017                  47.3
 270       James Harden        USA             9                NaN         basketball  2017                  46.6
 271     Lewis Hamilton         UK            10                NaN        auto racing  2017                  46.0
```

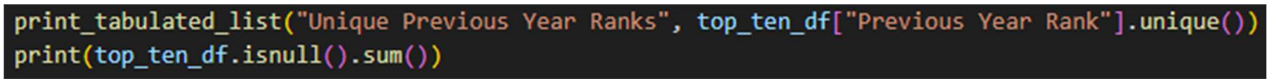
This has the wonderful double action of cleaning up the unique values in the column. As we are not concerned about the rank of the previous year, only whether they are there or not, we can fill all values not within 1-10 with the same value. However, we will need to manually change the first 10 rows, as there is no previous year to check from.



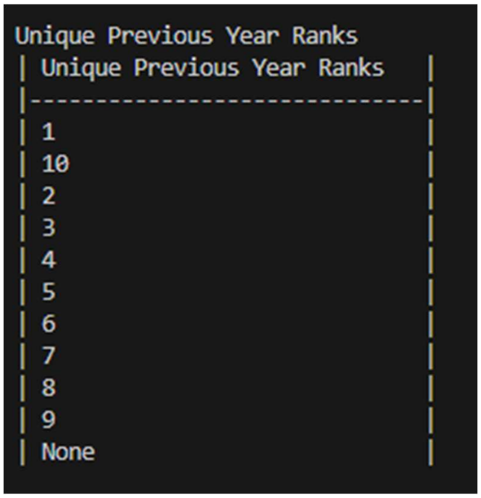
```
top_ten_df.loc[:9, "Previous Year Rank"] = "None"
fill_prev_years_rank(top_ten_df)
```

We now need to investigate the missing 2001 values. Looking online, the reporting period was changed from full calendar year to June-to-June in 2001, meaning that there was no consistent data. Therefore, this data is Missing not at Random (MNAR), and there is nothing we can do about it.

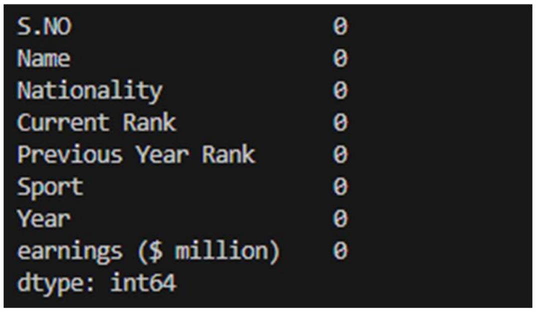
Source: https://www.topendsports.com/world/lists/earnings/forbes-index.htm



```
print_tabulated_list("Unique Previous Year Ranks", top_ten_df["Previous Year Rank"].unique())
print(top_ten_df.isnull().sum())
```



```
Unique Previous Year Ranks
| Unique Previous Year Ranks   |
|------------------------------|
| 1                            |
| 10                           |
| 2                            |
| 3                            |
| 4                            |
| 5                            |
| 6                            |
| 7                            |
| 8                            |
| 9                            |
| None                         |
```

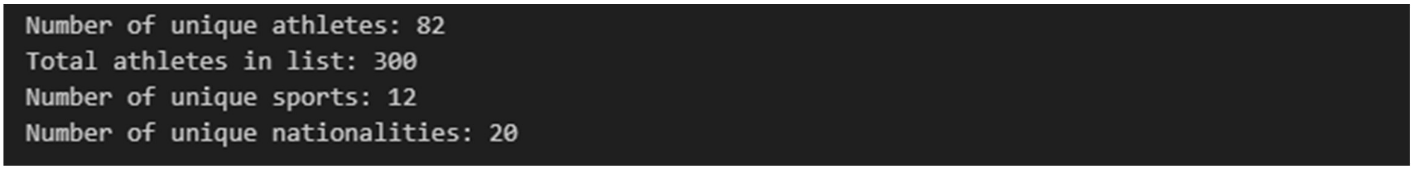


```
S.NO                    0
Name                    0
Nationality             0
Current Rank            0
Previous Year Rank      0
Sport                   0
Year                    0
earnings ($ million)    0
dtype: int64
```

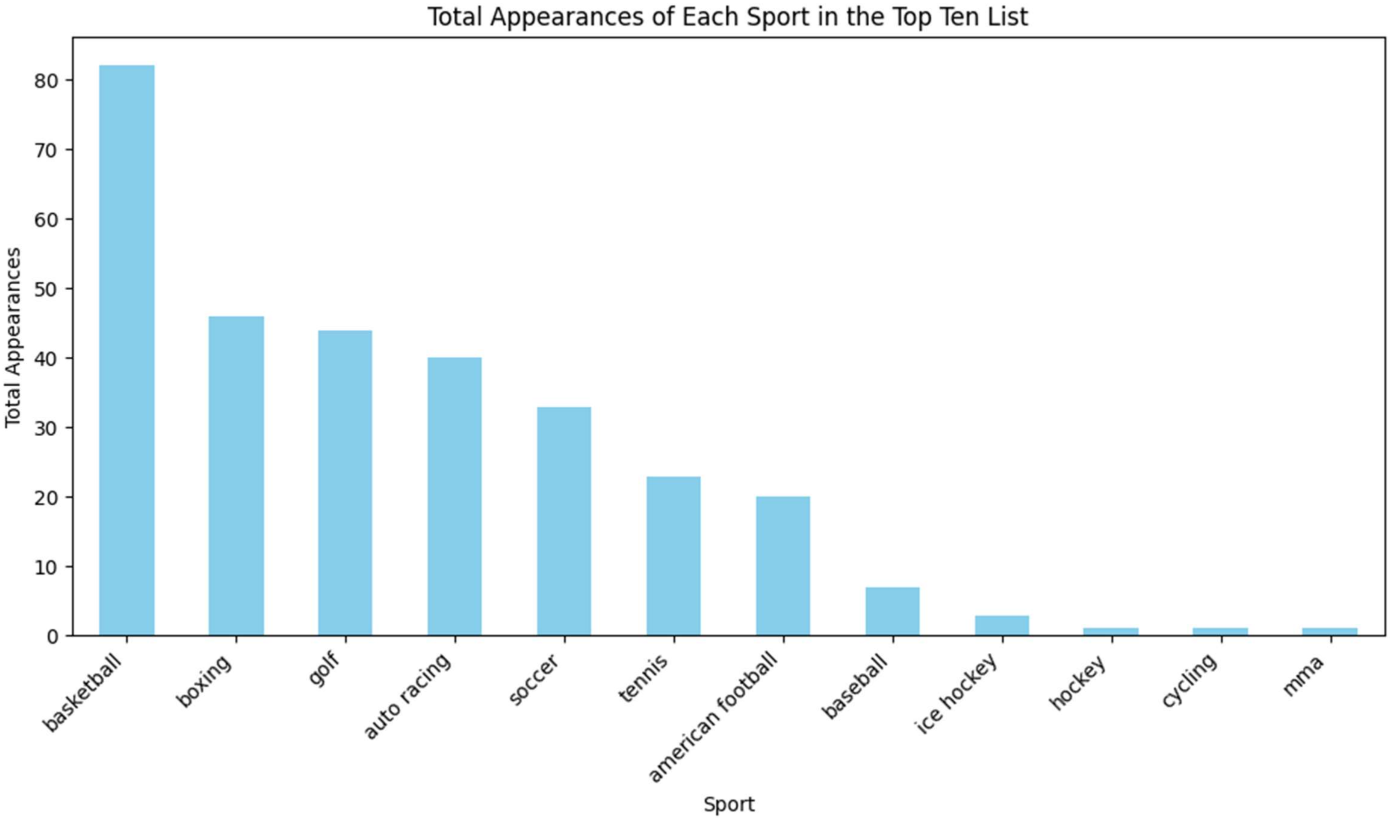
With no values missing, and acceptable unique values in key columns, we have a pre-processed dataset that is ready to be explored and analysed.

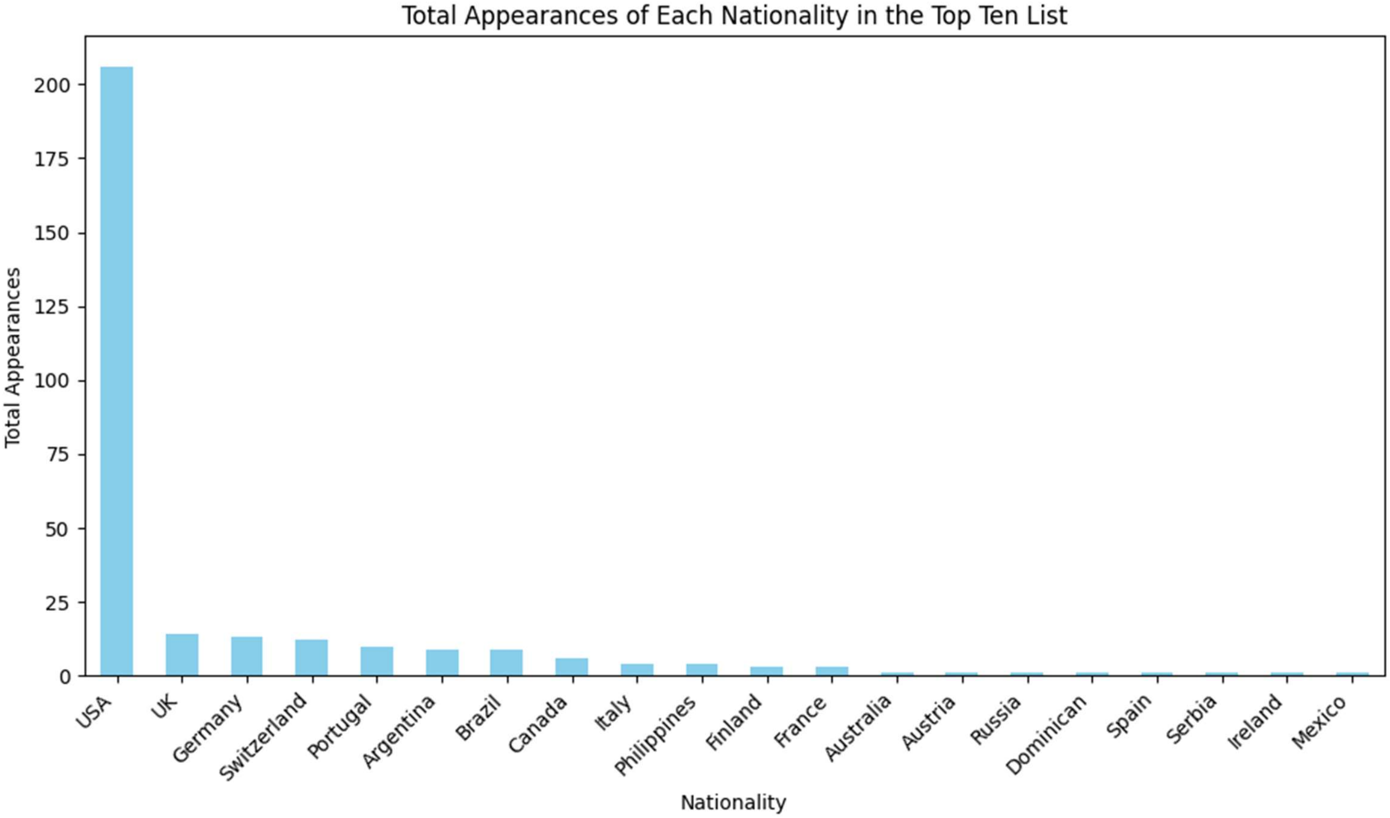
Data Stories and Visualisation

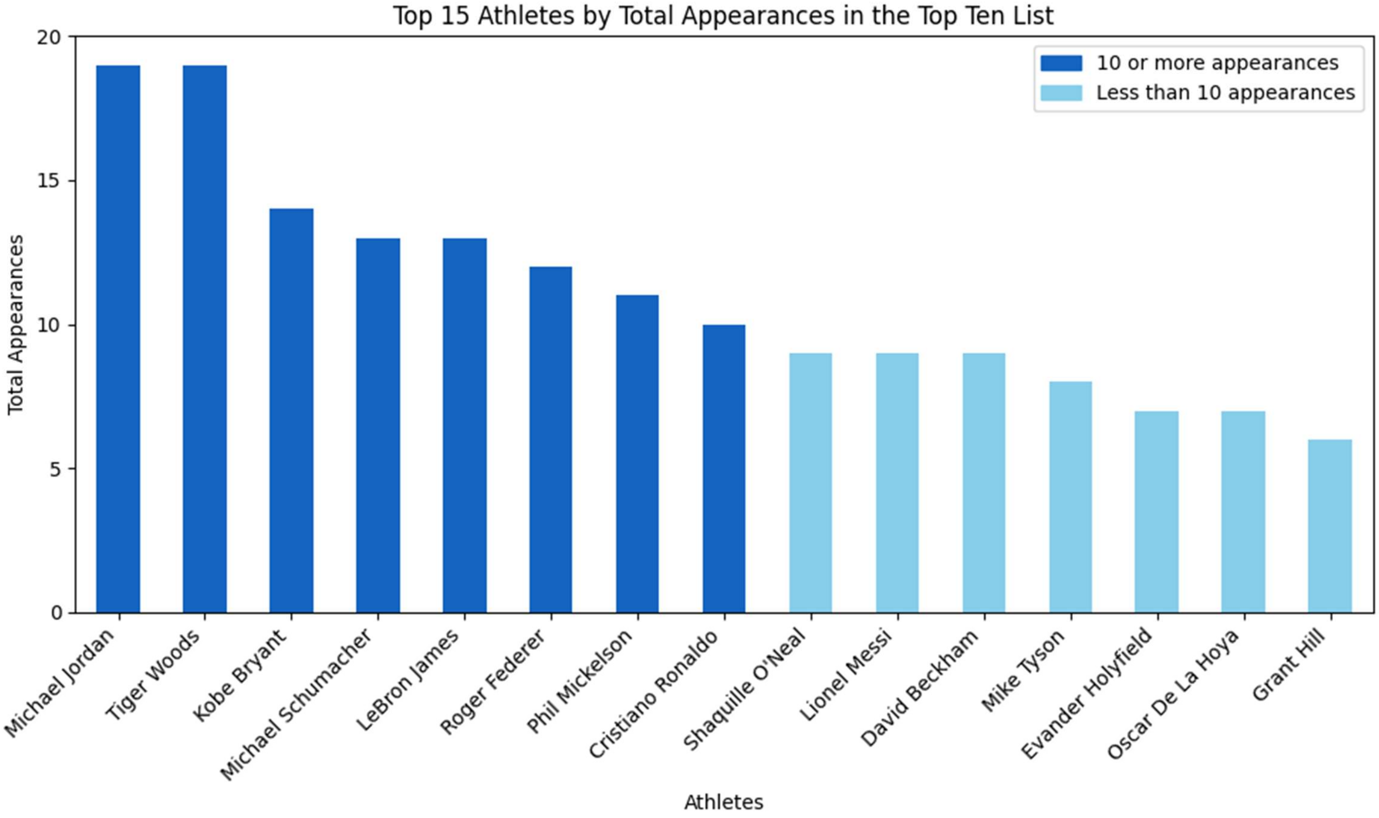
We should first start with a general overview of the athletes, nationalities and sports that make up our dataset.



```
Number of unique athletes: 82
Total athletes in list: 300
Number of unique sports: 12
Number of unique nationalities: 20
```



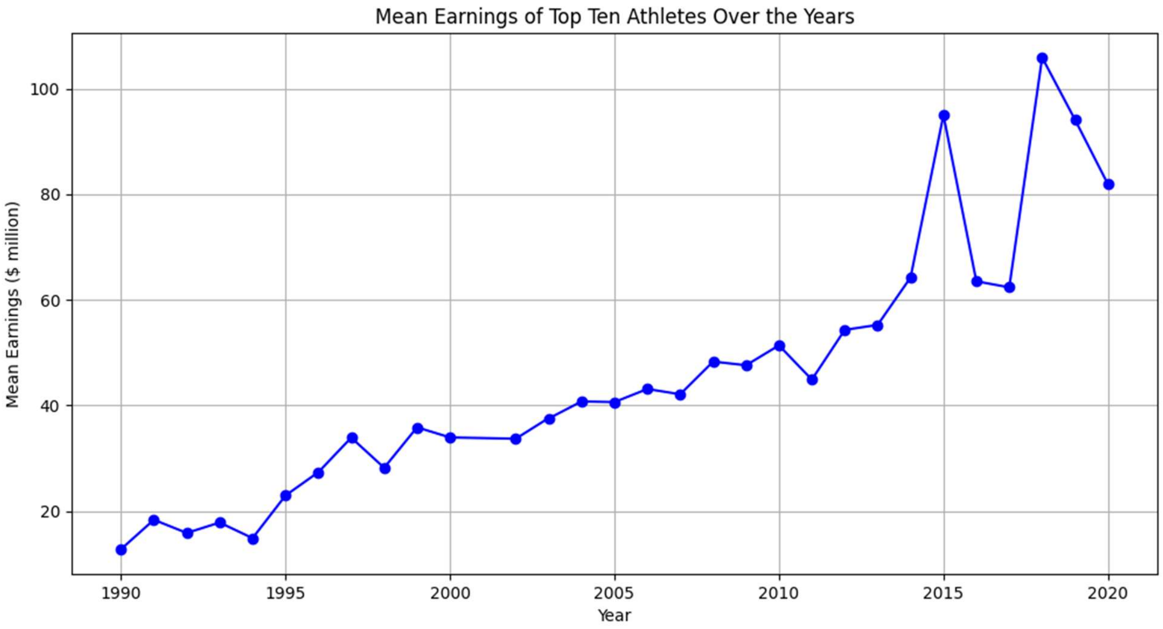




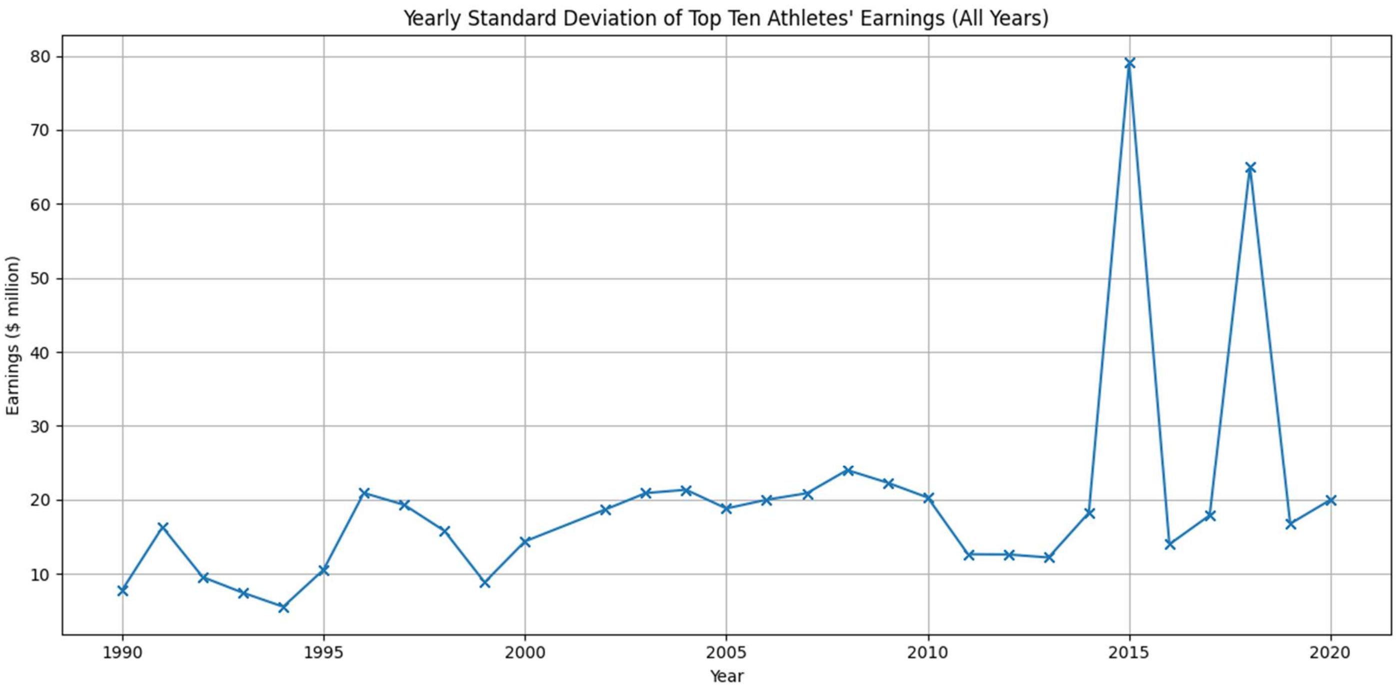
From these graphs, we see that the US dominates the nationalities of athletes on the list. Looking at the spread of sports across the list, basketball is clearly the sport that features the most, followed by a close group of boxing, golf and auto racing, and a steep drop-off after American football. Looking at the top 15 athletes by total appearances on the list, we see that 8 athletes have appeared on this list 10 or more times. This will most likely put these athletes amongst the highest earning athletes on the list, as longevity is an important factor in total earnings.

HOW ATHLETES EARN

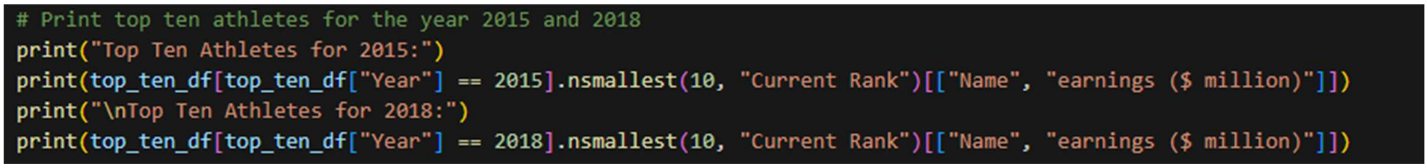
Calculating the mean and standard deviation of each years' group of athletes will show us two important elements. First, the mean will show us how much the total investment in sports has changed over the years. The standard deviation will show us the spread of the top ten athletes; if there's an even spread, or a global superstar out earning everyone.



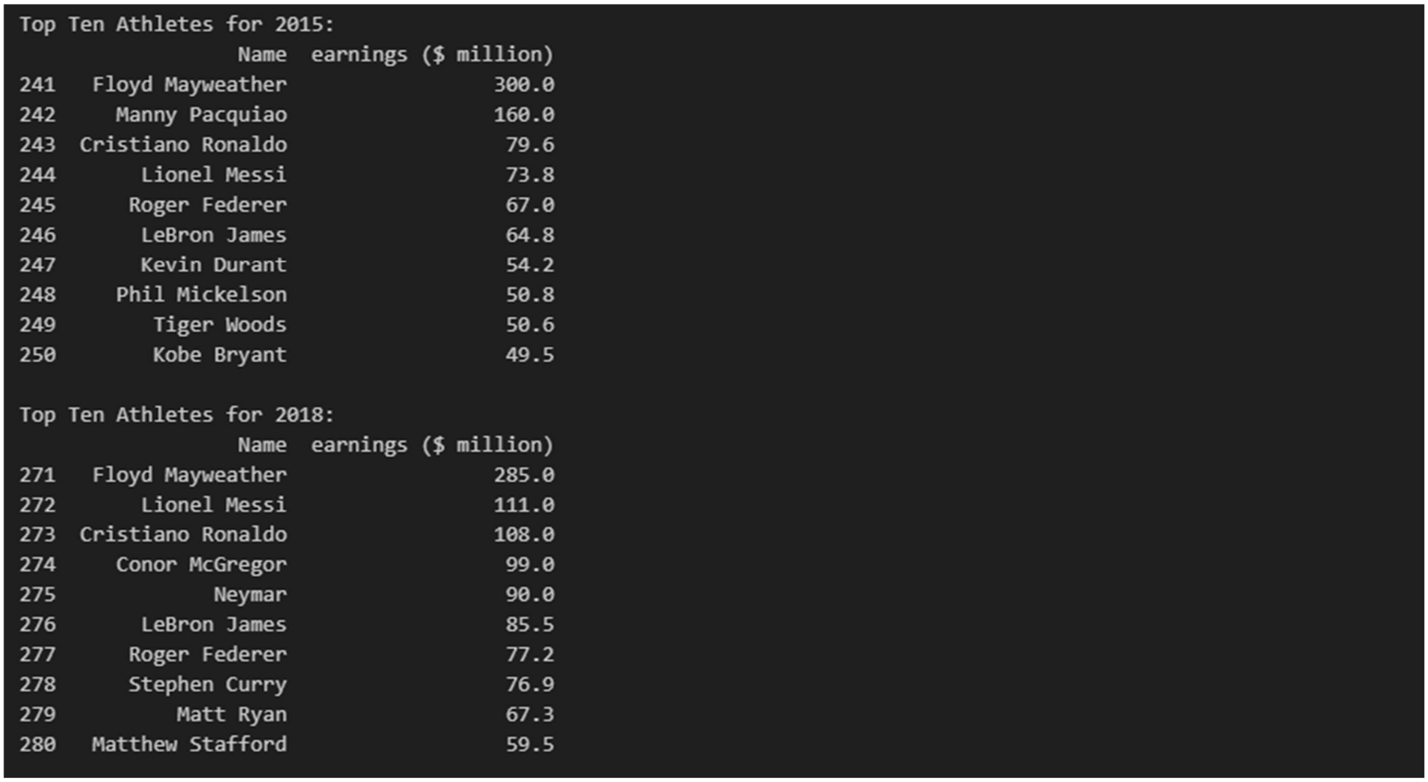
There is a distinct trend of more money being spent year on year, however there are also two major outliers in years 2015 and 2018.



Whilst the standard deviation meanders between 25 and 8 for all years in the dataset, except those same two years of 2015 and 2018. This could be indicative of marquee events this year, as only 1 or 2 athletes got paid enough to shift the mean earnings to extreme highs.



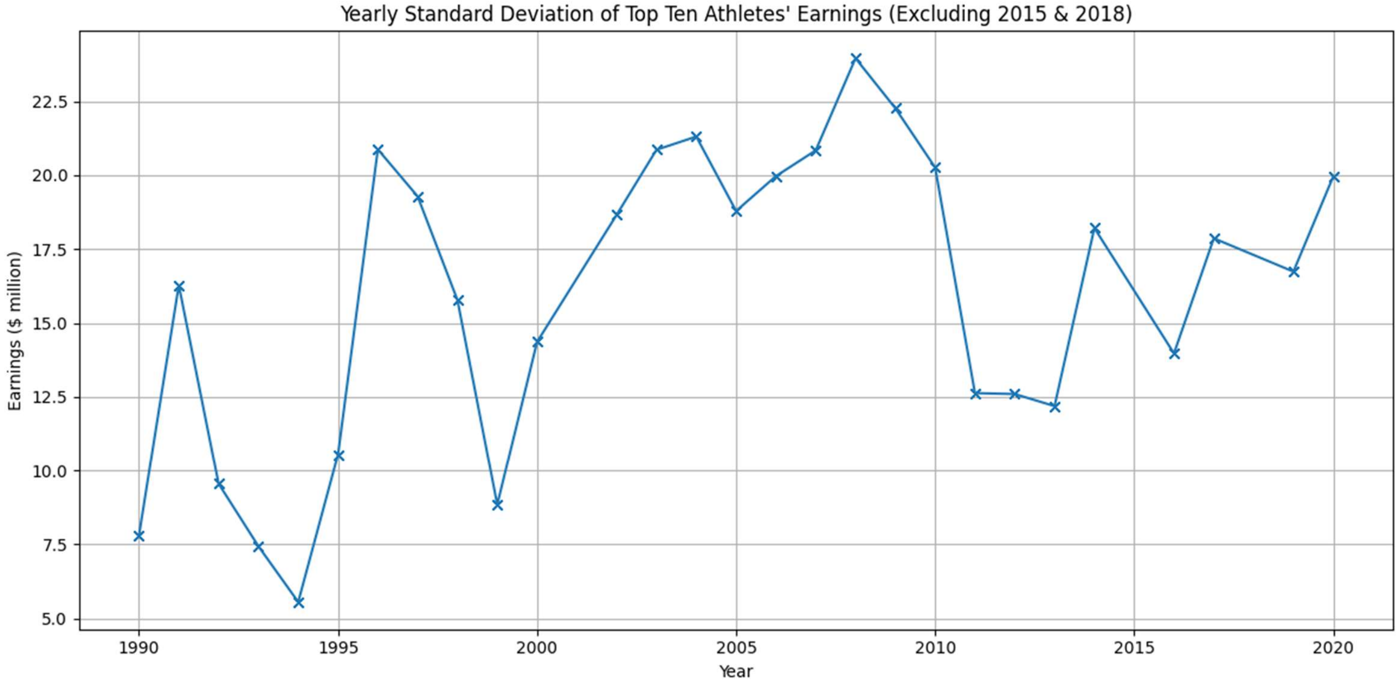
```
# Print top ten athletes for the year 2015 and 2018
print("Top Ten Athletes for 2015:")
print(top_ten_df[top_ten_df["Year"] == 2015].nsmallest(10, "Current Rank")[["Name", "earnings ($ million)"]])
print("\nTop Ten Athletes for 2018:")
print(top_ten_df[top_ten_df["Year"] == 2018].nsmallest(10, "Current Rank")[["Name", "earnings ($ million)"]])
```



```
Top Ten Athletes for 2015:
                  Name  earnings ($ million)
241    Floyd Mayweather                 300.0
242      Manny Pacquiao                 160.0
243   Cristiano Ronaldo                  79.6
244        Lionel Messi                  73.8
245       Roger Federer                  67.0
246        LeBron James                  64.8
247        Kevin Durant                  54.2
248       Phil Mickelson                 50.8
249         Tiger Woods                  50.6
250         Kobe Bryant                  49.5

Top Ten Athletes for 2018:
                  Name  earnings ($ million)
271    Floyd Mayweather                 285.0
272        Lionel Messi                 111.0
273   Cristiano Ronaldo                 108.0
274      Conor McGregor                  99.0
275              Neymar                  90.0
276        LeBron James                  85.5
277       Roger Federer                  77.2
278       Stephen Curry                  76.9
279           Matt Ryan                  67.3
280    Matthew Stafford                  59.5
```

We can see in those two years that Floyd Mayweather (boxing) earned 300 and 285 million – indicating that he held big fights. His opponents (Manny Pacquaio and Conor McGregor) earned 160 and 99 million these years, respectively, which explains the sharp increase in the standard deviation of the total athletes' earnings. The spike for 2018 is slightly lower as the money spent is not quite as extravagant, but it also appears that Cristiano Ronaldo, Lionel Messi and Neymar all earned big contracts that year too.

Whilst the trend for the mean is still rather evident, it is worthwhile removing the outlier years from the standard deviation graph to re-examine and see if there is a clearer trend.

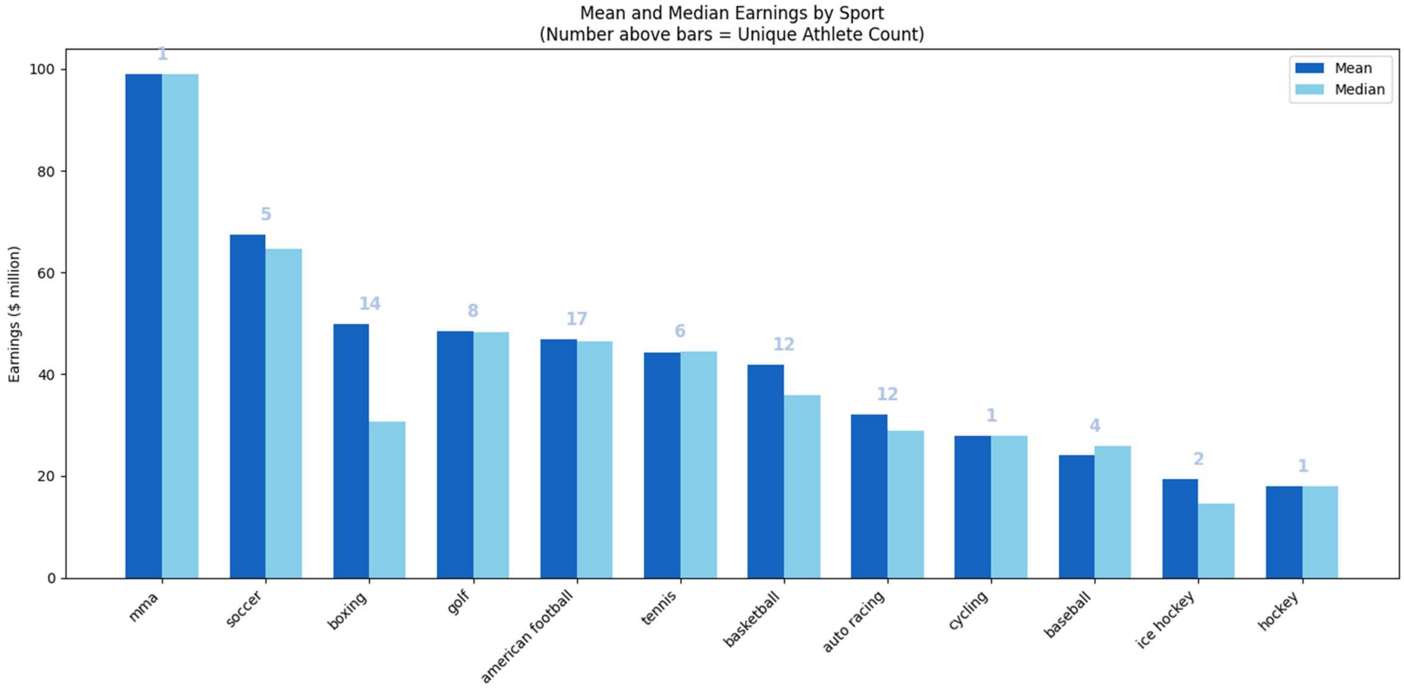

Yearly Standard Deviation of Top Ten Athletes' Earnings (Excluding 2015 & 2018)

Despite removing the outliers, no trend emerges from the standard deviation. There's a slight trend upwards towards the end, however the 2020 value is less than the 1996 value, but the spread of athletes within the top ten appears to be different each year.

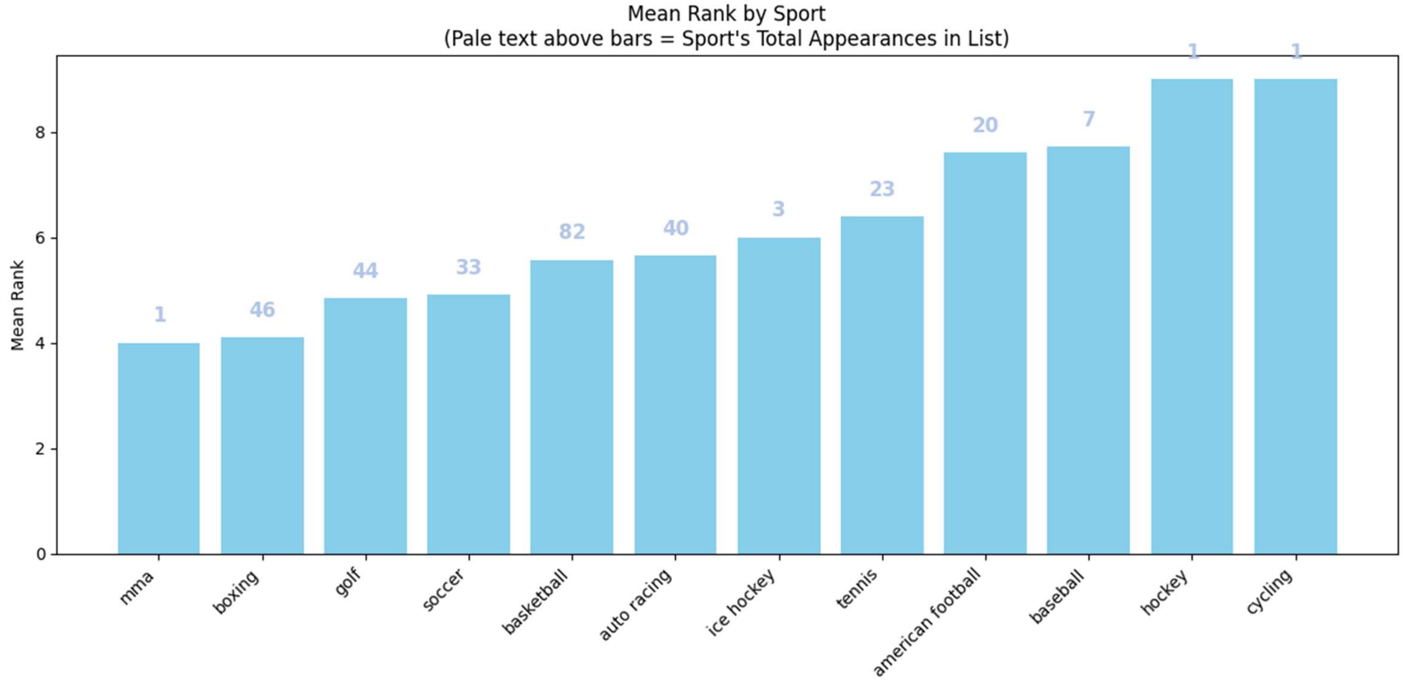
Overall, we see that more money has been invested into athletes over time. Marquee events can impact on the mean and std deviation of the data – it leads to an interesting idea that longevity may not be the best way to earn the most money in a sporting career, as in the years where large marquee events are removed, the spread of the top ten earnings does not seem to be related to the previous years data.

BEST PERFORMING SPORT

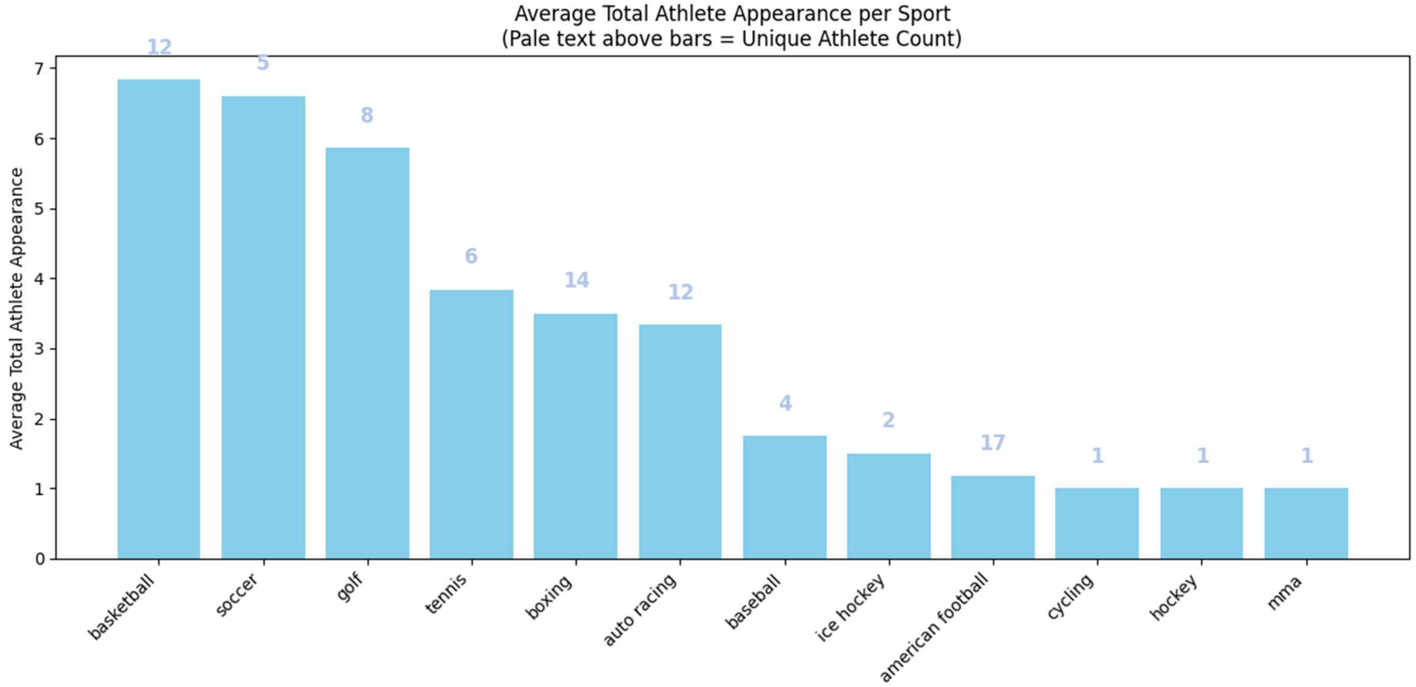
We now shift our focus to looking at how each sport performs in the list. This will take us through a view of their earnings, frequency and type of appearances on the list.



Here, each athlete's mean and median earnings is calculated, and then grouped into each sport. Each sport's mean and median earnings are calculated and presented, along with the number of athletes in each sport. We see that MMA, soccer and boxing have the highest means. MMA has one athlete, which was Conor McGregor in the previously mentioned outlier. There is also a large difference in the median and mean for boxing, which is most likely a result of the outlier Floyd Mayweather fights as well. Most other sports have even median and means – there would be an influence of when the athletes earned their money, as that would influence their mean earnings.



An examination of the mean rank per sport (and how many appearances on the list they get), with the lowest rank being better, we see that boxing and MMA have the lowest score. Golf and soccer are closely competing afterwards, and then it slowly trails higher from there. This data lends itself to the idea that boxing is the best paying sport, as it not only has the second highest number of athletes, but also the second-best mean ranking.



This chart measures each athletes' total appearances on the list, then groups them by sport. It measures the mean of each sport's athletes. For example, for MMA, we can expect that if a player makes the list, they will, on average, only spend one year on the list.

Basketball, soccer and golf's very high scores indicate that longevity at a 'superstar' peak of these sports is achievable and has been achieved between the years 1990 and 2020. The higher 'Unique Athlete Count' suggests that basketball has had more of these repeat athletes than soccer and golf; however, it could very well be one athlete staying for a long time.

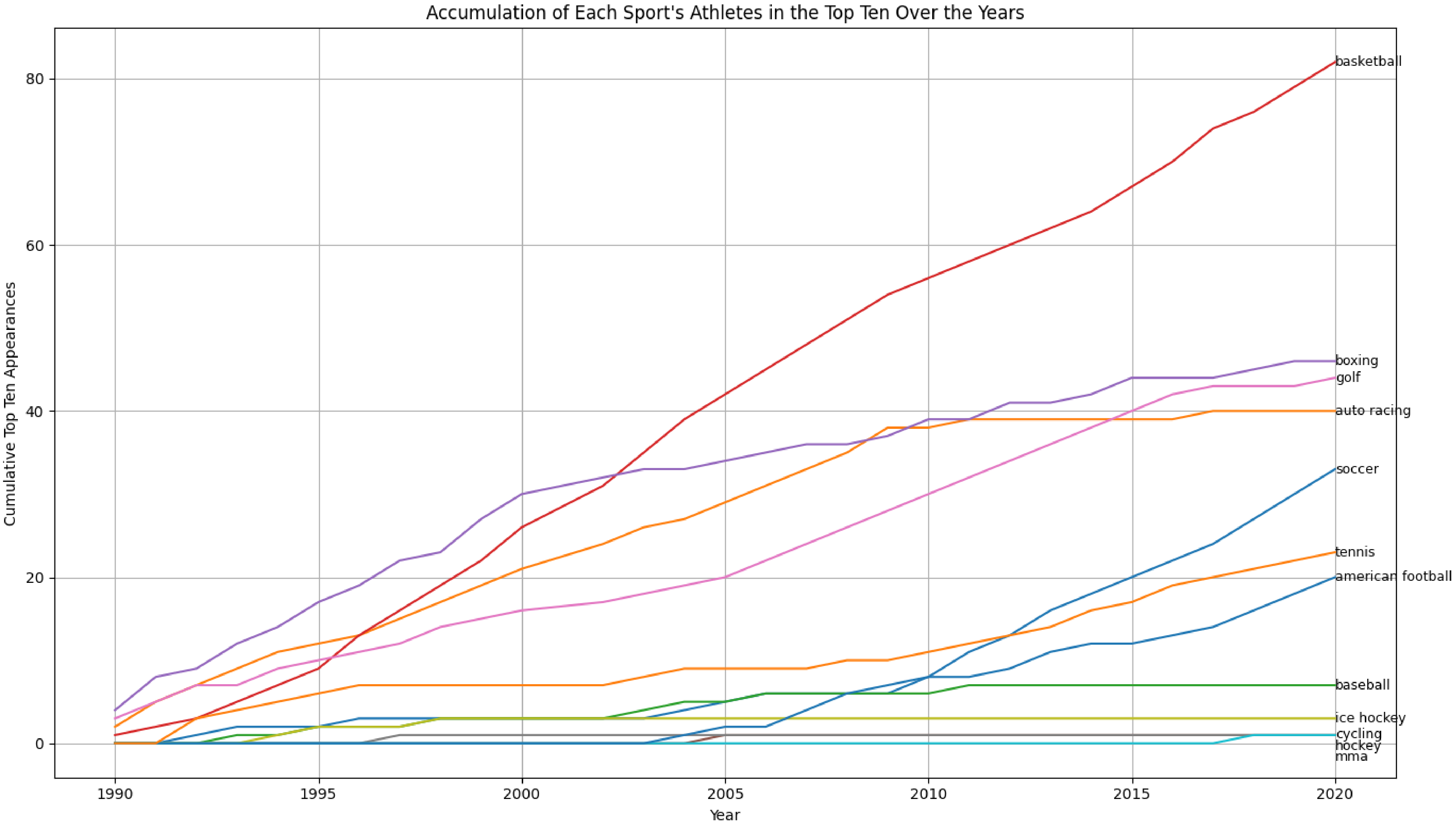
Boxing has a lower score, which make sense with the nature of the sport. Rather than have a few high-earning athletes for a long time, it needs a constant cycle of athletes for success. American Football also has a low score, but the highest number of unique athletes, which could show that the sport itself is not quite as lucrative for athletes, and longevity at the peak of the sport is not typical.

SEE OVERLEAF FOR GRAPH

The line graph on the previous page shows the running total of each sport's athlete's in the Top Ten list over the years, which shows the general trend of the sports performance over time. Basketball has consistently had athletes on the list, with constant growth since the data starts. Golf had a similar but smaller consistency until 2015 when it flattened out. Boxing and auto-racing were growing well over the first 10-15 years, before plateauing at the end. One hit wonders like MMA, cycling, ice-hockey and hockey are at the bottom.

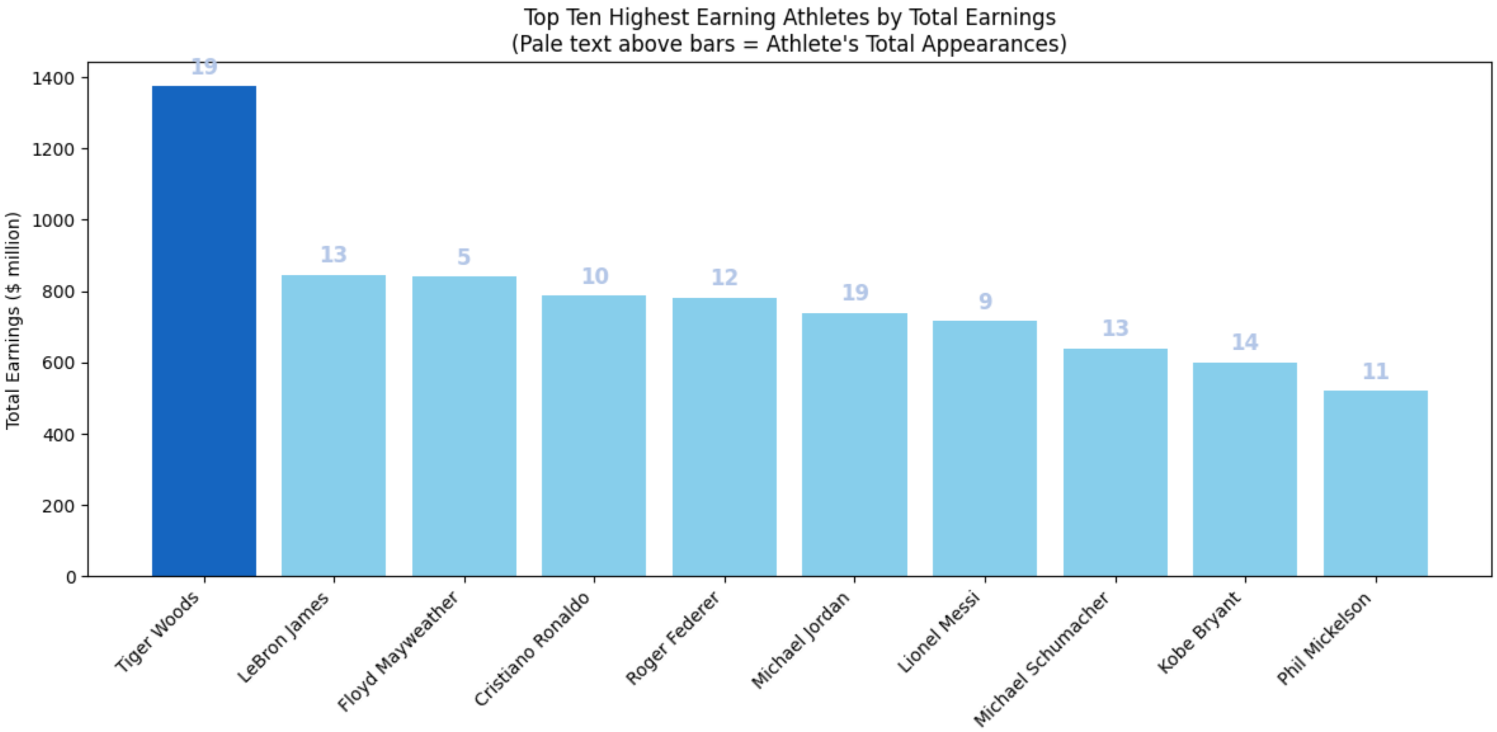
Soccer and American Football both experience larger growth right at the end, from years 2015-2020, potentially indicating increased popularity, larger salary caps or some superstars entering the sport.

Overall, basketball is the most consistently well-paying sport, with strong mean total earnings per athlete, and strong mean rank. It also has good opportunities for longevity. Boxing is perhaps the best sport for earning money, however it's not likely to have a long career at the top of the sport. Soccer and Golf are very similar to each other, with their top athletes being paid well and supported to have longevity at the top performance of their sport.

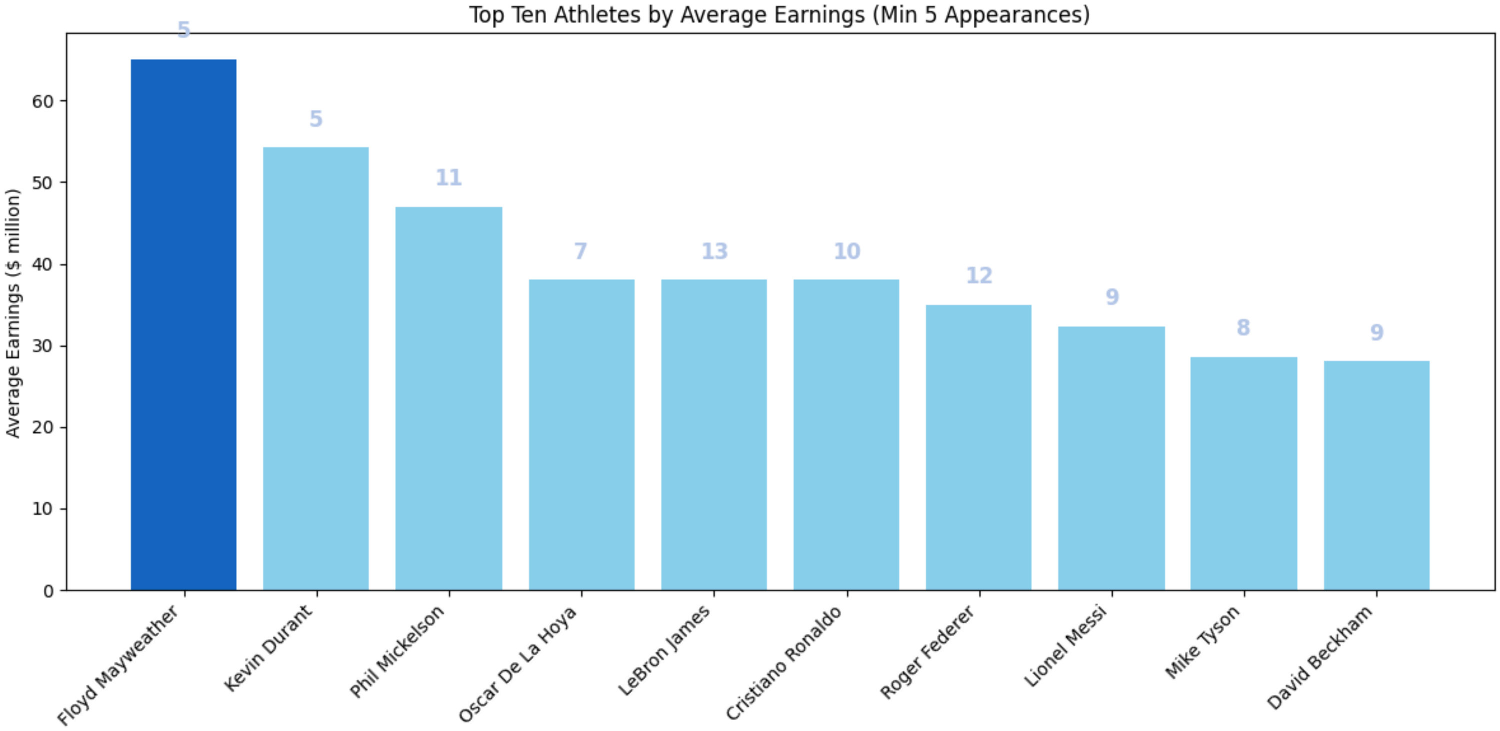

Accumulation of Each Sport's Athletes in the Top Ten Over the Years
Cumulative Top Ten Appearances
Year
0
20
40
60
80
1990
1995
2000
2005
2010
2015
2020
basketball
boxing
golf
auto racing
soccer
tennis
american football
baseball
ice hockey
cycling
hockey
mma

BEST ATHLETE

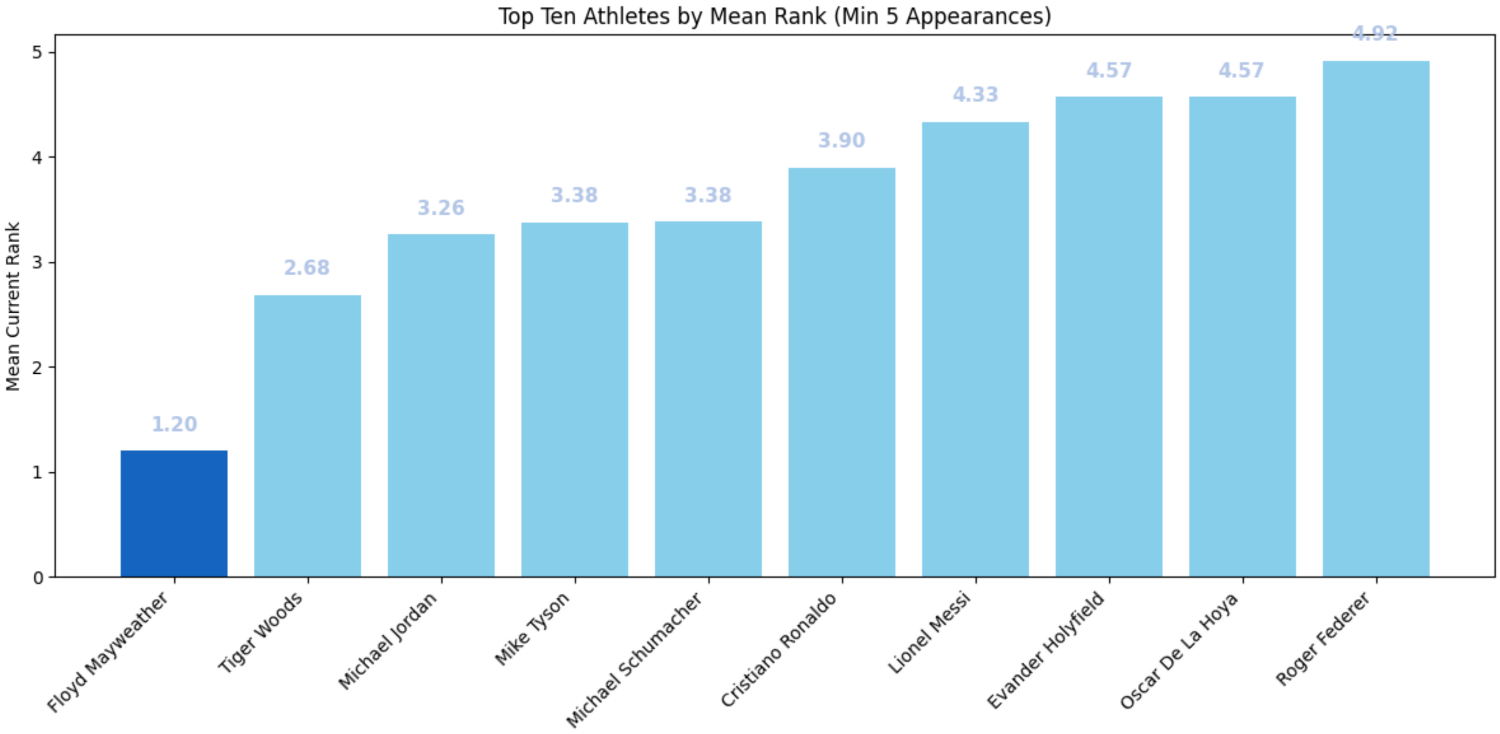
Now for the fun part – let's decide who the best athlete is! We will examine how much money they have made, as well as their mean rank and dominance within their sport. To remove outliers, all athletes must appear on the list 5 times to be considered for the graphs.



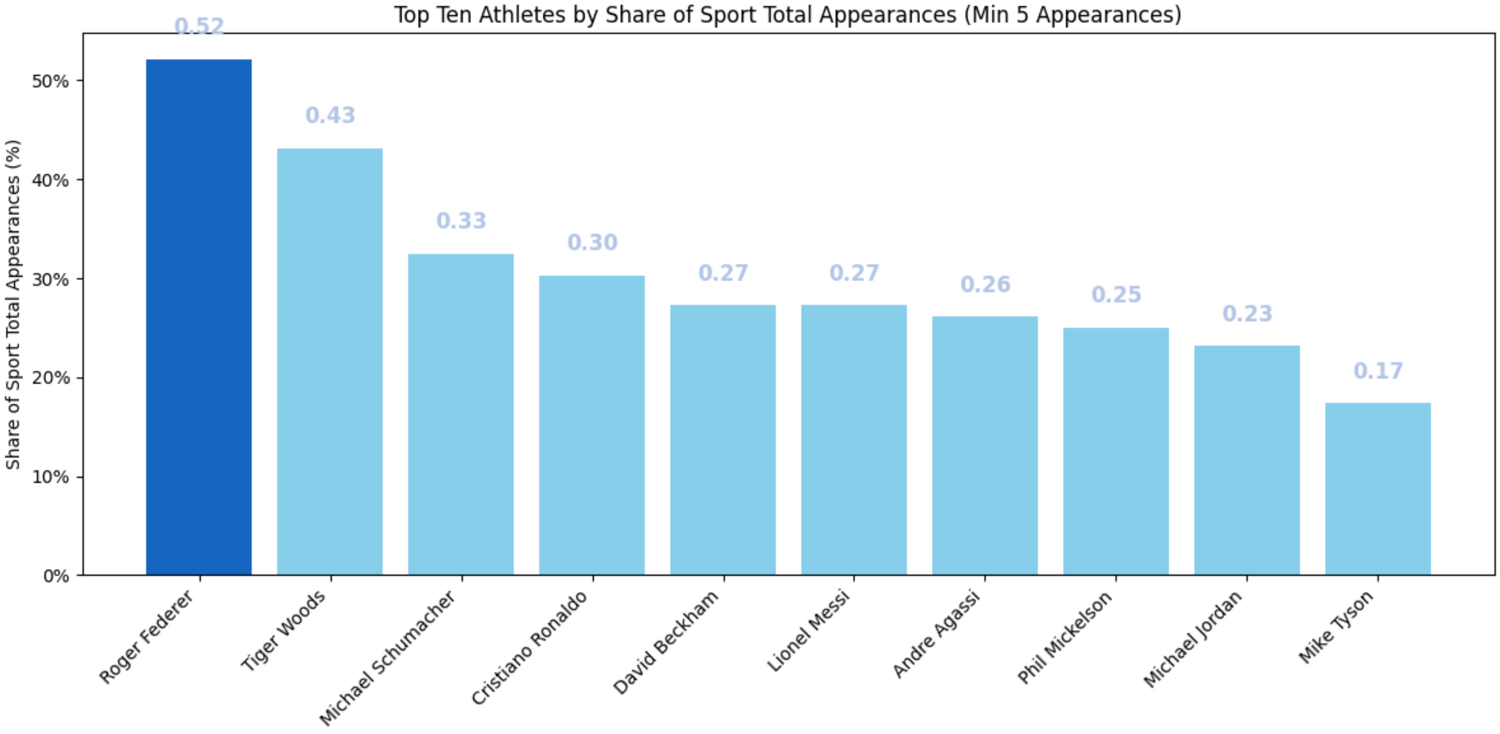
Here we can see that Tiger Woods has well and truly earned the most of any athlete on this list. A large part of this is due to his 19 appearances, however it takes a staggering career to be at the top for that long. It is interesting that Michael Jordan has the same number of appearances, and much less total earnings – s true showing of how the increasing investment in sports benefitted some athletes, and not others.



Looking at the highest average per athlete (using a qualifier of 5 appearances to remove outliers), it's not surprising that Floyd Mayweather is at the top. However, Kevin Durant and Phil Mickelson are also accustomed to large pay checks. Interestingly, Tiger Woods and Michael Jordan are not on this list at all – this begs the question, if they had similar careers a decade later, how much more would they have earned?



Looking at the athletes by their mean rank, we can examine their standing amongst the best of the best. We see that Floyd Mayweather once again tops our list – when he makes the list, he is as close to the top as possible. Having already discussed Tiger Woods and Michael Jordans longevity with 19 appearances each, their very low scores become even more impressive.



Here we look at how much of each sports total appearances an athlete is responsible for: a measure of dominance within their sport. We see that Roger Federer is responsible for more than half of tennis's appearances. Tiger Woods is the other notable athlete, contributing to nearly 43% of golf's appearances. It's also worth noting that soccer has Cristiano Ronaldo, Lionel Messi and David Beckham combining to secure 90% of their appearances.

Overall, for the title of best athlete, it's a discussion between Floyd Mayweather and Tiger Woods; Floyd earned the most money per appearance and had the highest mean ranking. However, Tiger's dominance within his sport, dominant total earnings and very low mean ranking (considering his longevity) places him in the discussion as well. Should money be taken out of the equation, or equalised across time periods, the answer may become clear, or a new athlete emerge.

This report was written by: Jem Herbert-Rice