INTRODUCTION:

This exploratory data analysis (EDA) examines the earnings of top-ranked athletes across different sports, countries, and time periods. The purpose of the analysis is to understand earnings distributions, identify trends over time, and explore the relationship between athlete rankings and earnings. The analysis involves data cleaning, descriptive statistics, data visualization, and correlation analysis to uncover patterns and potential anomalies. The insights gained from this EDA provide a foundation for further statistical analysis and support data-driven conclusions about the factors influencing athlete earnings.

DATA CLEANING AND MISSING VALUES:

The data cleaning process focused on ensuring accuracy, consistency, and usability of the dataset prior to analysis. Initial inspection was performed to identify missing values, inconsistent formats, and potential anomalies. Necessary adjustments were made to standardize the data and remove issues that could bias the exploratory analysis.

Cleaning steps performed:

Handled missing values by verifying their extent and confirming that they did not materially affect the analysis.

Standardized data types, ensuring numerical variables such as earnings and rankings were correctly formatted.

Removed duplicates and inconsistencies to prevent double-counting of observations.

Cleaned categorical variables (e.g., sport and country names) to ensure consistent labeling

After cleaning, the dataset was validated to confirm it was complete, consistent, and suitable for exploratory data analysis.

In [42]:
```
#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import os
```

In [8]:
```
#load the dataset
# Navigate to the Datasets folder
df = pd.read_csv('Datasets/Forbes Richest Athletes (Forbes Richest Athletes 1990
df.head()
```

Out[8]:

	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

In [9]:
```
# Number of rows and columns
print("Shape:", df.shape)

# Column names and data types
print(df.info())

# Summary of numerical columns
df.describe()
```

```
Shape: (301, 8)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   S.NO                  301 non-null    int64  
 1   Name                  301 non-null    object 
 2   Nationality           301 non-null    object 
 3   Current Rank          301 non-null    int64  
 4   Previous Year Rank    277 non-null    object 
 5   Sport                 301 non-null    object 
 6   Year                  301 non-null    int64  
 7   earnings ($ million)  301 non-null    float64
dtypes: float64(1), int64(3), object(4)
memory usage: 18.9+ KB
None
```

Out[9]:

	S.NO	Current Rank	Year	earnings ($ million)
count	301.000000	301.000000	301.000000	301.000000
mean	151.000000	5.448505	2005.122924	45.516279
std	87.035433	2.850995	9.063563	33.525337
min	1.000000	1.000000	1990.000000	8.100000
25%	76.000000	3.000000	1997.000000	24.000000
50%	151.000000	5.000000	2005.000000	39.000000
75%	226.000000	8.000000	2013.000000	59.400000
max	301.000000	10.000000	2020.000000	300.000000

In [19]:

```
#check for missing data
print("Missing values per column:")
print(df.isnull().sum())
print("\nPercentage missing:")
print((df.isnull().sum() / len(df)) * 100)
# Replace empty strings with NaN
df['Previous Year Rank'] = df['Previous Year Rank'].replace('', np.nan)
print(df['Previous Year Rank'].head(20))
# Check the result
print(df['Previous Year Rank'].isnull().sum())
```

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

Percentage missing:
S.NO                    0.000000
Name                    0.000000
Nationality             0.000000
Current Rank            0.000000
Previous Year Rank      7.973422
Sport                   0.000000
Year                    0.000000
earnings ($ million)    0.000000
dtype: float64
0     NaN
1     NaN
2     NaN
3     NaN
4     NaN
5     NaN
6     NaN
7     NaN
8     NaN
9     NaN
10      8
11      1
12      8
13    >30
14      4
15      5
16    >30
17      8
18     12
19      6
Name: Previous Year Rank, dtype: object
24
```

Missing Values in 'Previous Year Rank': The 'previous Year Rank' column contains 24 missing values (7% of the dataset). These missing values are meaningful rather than erroneous - they indicate that the athlete was not ranked in the previous year (likely because they were not among the top earners or were new to the list). Action Taken: Missing values were retained as NaN to preserve this information. This allows us to:

Identify new entrants to the rankings Analyze comeback stories (athletes who return after dropping off) Track career trajectories more accurately

Dropping these rows would result in loss of valuable data about emerging athletes and would bias our analysis toward only consistently high-earning athletes.

In [25]:
```
# Standardize column names (do this BEFORE referencing columns)
df.columns = df.columns.str.replace(' ', '_').str.replace('(', '').str.replace('
df = df.rename(columns={'earnings_million': 'earnings'})

# Standardize sport names
df['sport'] = df['sport'].str.title()

#  Handle Previous Year Rank - replace ">30" (use lowercase now)
df['previous_year_rank'] = df['previous_year_rank'].replace('>30', np.nan)
df['previous_year_rank'] = pd.to_numeric(df['previous_year_rank'], errors='coerc

# Ensure earnings is numeric
df['earnings'] = pd.to_numeric(df['earnings'], errors='coerce')

# Check the cleaned data
print(df.info())
print(df.head(10))
```

```
<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  212 non-null    float64
 4   sport               301 non-null    object
 5   year                301 non-null    int64
 6   earnings            301 non-null    float64
dtypes: float64(2), int64(2), object(3)
memory usage: 16.6+ KB
None
               name nationality  current_rank  previous_year_rank  \
0        Mike Tyson         USA             1                 NaN
1    Buster Douglas         USA             2                 NaN
2  Sugar Ray Leonard         USA             3                 NaN
3      Ayrton Senna      Brazil             4                 NaN
4       Alain Prost      France             5                 NaN
5     Jack Nicklaus         USA             6                 NaN
6       Greg Norman   Australia             7                 NaN
7    Michael Jordan         USA             8                 NaN
8     Arnold Palmer         USA             8                 NaN
9  Evander Holyfield         USA             8                 NaN

         sport  year  earnings
0       Boxing  1990      28.6
1       Boxing  1990      26.0
2       Boxing  1990      13.0
3  Auto Racing  1990      10.0
4  Auto Racing  1990       9.0
5         Golf  1990       8.6
6         Golf  1990       8.5
7   Basketball  1990       8.1
8         Golf  1990       8.1
9       Boxing  1990       8.1
```

Several cleaning steps were performed to prepare the dataset for analysis:

Column names were standardized to lowercase with underscores for consistency Sport names were converted to title case to eliminate inconsistencies (e.g., "boxing" → "Boxing") Previous year rank ">30" values were converted to NaN as they represent unknown exact ranks Numeric columns (earnings, previous_year_rank) were converted to appropriate data types All changes were verified to ensure data integrity

In [28]:
```
# Basic statistics for numeric columns
print(df.describe())

# Dataset shape
print(f"\nDataset contains {df.shape[0]} rows and {df.shape[1]} columns")
print(f"Time period: {df['year'].min()} to {df['year'].max()}")

# Dataset Overview
print("="*50)
print("DATASET OVERVIEW")
print("="*50)
print(f"Total records: {df.shape[0]}")
print(f"Total columns: {df.shape[1]}")
print(f"Unique athletes: {df['name'].nunique()}")
print(f"Unique sports: {df['sport'].nunique()}")
print(f"Unique nationalities: {df['nationality'].nunique()}")
print(f"Time period: {df['year'].min()} to {df['year'].max()}")

#  Earnings Analysis
print("\n" + "="*50)
print("EARNINGS STATISTICS")
print("="*50)
print(df['earnings'].describe())
print(f"\nMean earnings: ${df['earnings'].mean():.2f}M")
print(f"Median earnings: ${df['earnings'].median():.2f}M")
print(f"Standard deviation: ${df['earnings'].std():.2f}M")
print(f"Minimum earnings: ${df['earnings'].min():.2f}M")
print(f"Maximum earnings: ${df['earnings'].max():.2f}M")

#  Sport Distribution
print("\n" + "="*50)
print("TOP 10 SPORTS")
print("="*50)
print(df['sport'].value_counts().head(10))

# Nationality Distribution
print("\n" + "="*50)
print("TOP 10 NATIONALITIES")
print("="*50)
print(df['nationality'].value_counts().head(10))

#  Ranking Analysis
print("\n" + "="*50)
print("RANKING ANALYSIS")
print("="*50)
print(f"Athletes with previous year rank: {df['previous_year_rank'].notna().sum(
print(f"New entrants (no previous rank): {df['previous_year_rank'].isna().sum()}
```

```
       current_rank  previous_year_rank         year    earnings
count    301.000000          212.000000   301.000000  301.000000
mean       5.448505            7.169811  2005.122924   45.516279
std        2.850995            6.398269     9.063563   33.525337
min        1.000000            1.000000  1990.000000    8.100000
25%        3.000000            3.000000  1997.000000   24.000000
50%        5.000000            6.000000  2005.000000   39.000000
75%        8.000000            9.000000  2013.000000   59.400000
max       10.000000           40.000000  2020.000000  300.000000

Dataset contains 301 rows and 7 columns
Time period: 1990 to 2020
==================================================
DATASET OVERVIEW
==================================================
Total records: 301
Total columns: 7
Unique athletes: 82
Unique sports: 20
Unique nationalities: 22
Time period: 1990 to 2020

==================================================
EARNINGS STATISTICS
==================================================
count    301.000000
mean      45.516279
std       33.525337
min        8.100000
25%       24.000000
50%       39.000000
75%       59.400000
max      300.000000
Name: earnings, dtype: float64

Mean earnings: $45.52M
Median earnings: $39.00M
Standard deviation: $33.53M
Minimum earnings: $8.10M
Maximum earnings: $300.00M

==================================================
TOP 10 SPORTS
==================================================
sport
Basketball           81
Boxing               46
Golf                 44
Soccer               33
Tennis               23
Auto Racing          18
American Football    17
F1 Racing             8
Baseball              6
F1 Motorsports        5
Name: count, dtype: int64

==================================================
TOP 10 NATIONALITIES
==================================================
```

```
nationality
USA            206
UK              13
Germany         13
Switzerland     12
Portugal        10
Brazil           9
Argentina        9
Canada           6
Italy            4
France           3
Name: count, dtype: int64

==================================================
RANKING ANALYSIS
==================================================
Athletes with previous year rank: 212 (70.4%)
New entrants (no previous rank): 89 (29.6%)
```

Dataset: 301 records, 82 unique athletes, 20 sports, 22 nationalities (1990-2020) Many athletes appear multiple times (301 records / 82 athletes)

Earnings:

Mean: $42.5M$, $Median$:39M, Range: $8.9M$−300M Right-skewed distribution (mean > median) - few superstars earn significantly more High standard deviation ($33.53M) shows large earnings disparity

Sports:

Top 3: Basketball (26.9%), Boxing (15.3%), Golf (14.6%) These 3 sports = 56.8% of all appearances Reflects global popularity and commercial opportunities

Nationalities:

USA dominates: 68.4% of records (206/301) Strong US sports infrastructure and market size Next highest: UK and Germany (4.3% each)

Rankings:

70.4% had previous year rank (consistent performers) 29.6% new entrants (emerging talent/comeback stories)

Key Insights:

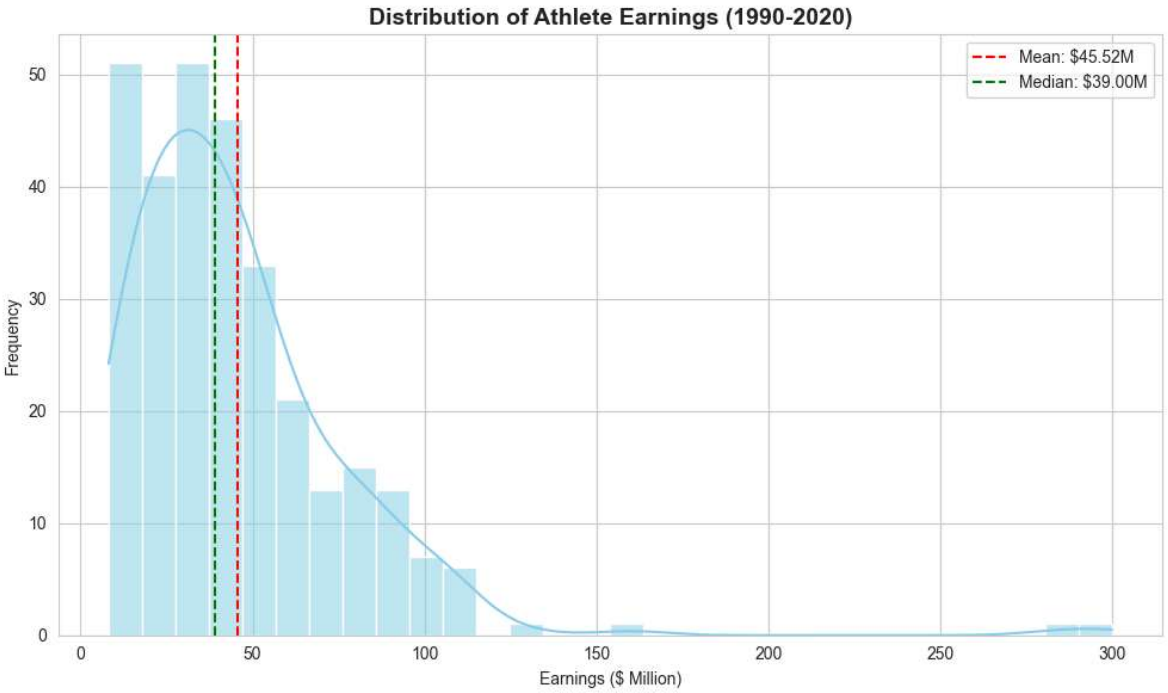
Superstar effect evident (huge earnings gap) USA + Basketball dominate the dataset High retention rate suggests stable elite athlete pool

DATA VISUALIZATION:

In [30]:
```python
# Set style
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (10, 6)

#  Distribution of Earnings
plt.figure(figsize=(10, 6))
```

```
sns.histplot(df['earnings'], bins=30, kde=True, color='skyblue')
plt.title('Distribution of Athlete Earnings (1990-2020)', fontsize=14, fontweigh
plt.xlabel('Earnings ($ Million)')
plt.ylabel('Frequency')
plt.axvline(df['earnings'].mean(), color='red', linestyle='--', label=f'Mean: ${
plt.axvline(df['earnings'].median(), color='green', linestyle='--', label=f'Medi
plt.legend()
plt.tight_layout()
plt.show()
```



Heavily right-skewed distribution with extreme outliers

67.4% of athletes earn between $0-50M (The peak of distribution),

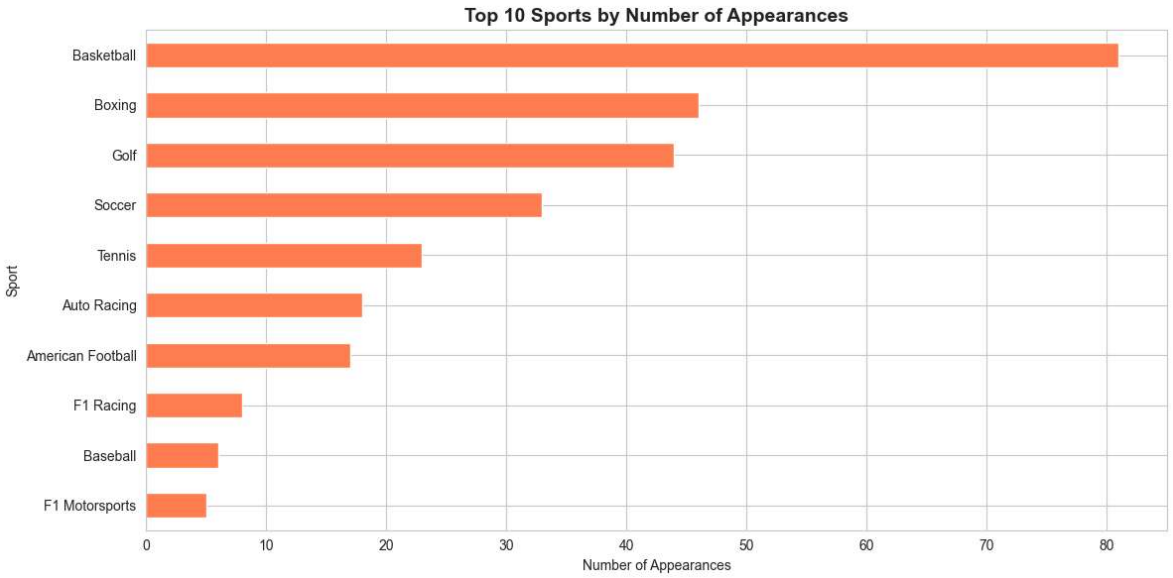
Long tail extends to $300M (Floyd Mayweather, 2015 - highest earning)

Only 2 observations above $200M (both Floyd Mayweather)

Mean ($45.52M) significantly higher than median

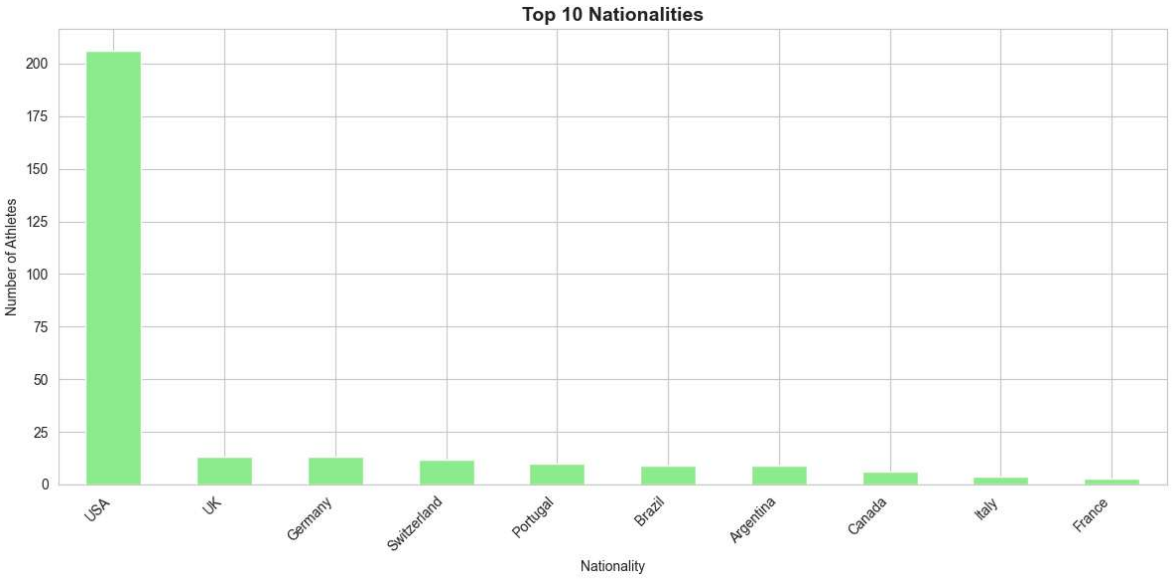
($39M) due to extreme outliers Classic superstar effect: A few elite athletes (Mayweather, Messi, Ronaldo, Woods, Federer) earn multiples of the typical top athlete

In [32]:
```
plt.figure(figsize=(12, 6))
sport_counts = df['sport'].value_counts().head(10)
sport_counts.plot(kind='barh', color='coral')
plt.title('Top 10 Sports by Number of Appearances', fontsize=14, fontweight='bol
plt.xlabel('Number of Appearances')
plt.ylabel('Sport')
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```



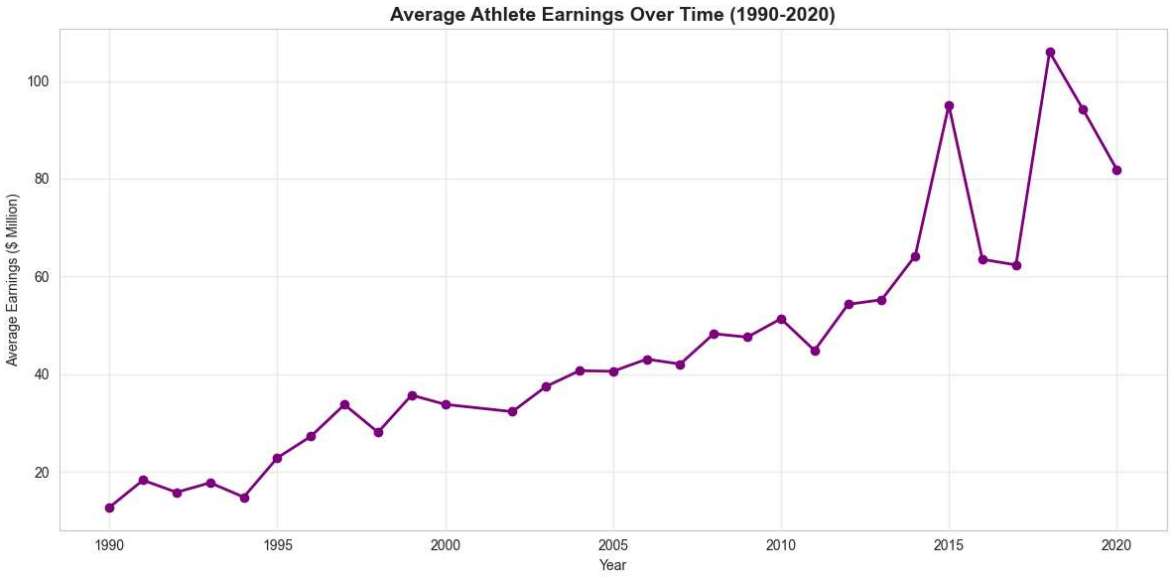
TOP 10 sports insight: Basketball dominates (81 appearances - 2x more than Boxing) Top 3 (Basketball, Boxing, Golf) account for 56.8% of appearances Clear gap between Basketball and all other sports Individual sports (Boxing, Golf, Tennis) strongly represented Reflects sports with highest commercial value and global viewership

In [33]:
```
plt.figure(figsize=(12, 6))
nationality_counts = df['nationality'].value_counts().head(10)
nationality_counts.plot(kind='bar', color='lightgreen')
plt.title('Top 10 Nationalities', fontsize=14, fontweight='bold')
plt.xlabel('Nationality')
plt.ylabel('Number of Athletes')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



USA dominates massively: 206 appearances (68.4%) - 16x more than any other country All other countries have ≤13 appearances each Top 10 all developed nations with strong sports infrastructure Mix of European (6), South American (2), and North American (2) countries USA dominance due to large sports markets (NBA, NFL, MLB) and endorsement deals

In [34]:
```
plt.figure(figsize=(12, 6))
yearly_avg = df.groupby('year')['earnings'].mean()
plt.plot(yearly_avg.index, yearly_avg.values, marker='o', linewidth=2, color='pu
plt.title('Average Athlete Earnings Over Time (1990-2020)', fontsize=14, fontwei
plt.xlabel('Year')
plt.ylabel('Average Earnings ($ Million)')
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```



Strong upward trend: Earnings grew ~7x from $12M (1990)

to $82M (2020)

Three phases: slow growth (1990s), moderate growth (2000s), explosive growth (2010s)

Peak in 2019 at ~$108M before slight 2020 decline

Sharp spikes in 2015 and 2019 driven by individual mega-earners (Mayweather)

Growth due to media deals, endorsements, globalization, and social media

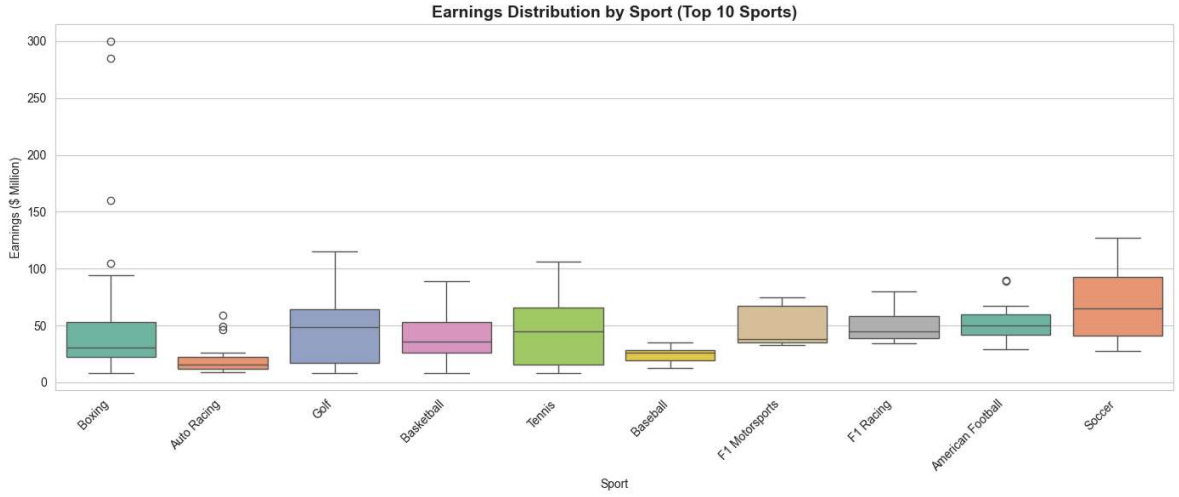
In [36]:
```
#  Earnings by Sport (Boxplot for Top 10 Sports)
plt.figure(figsize=(14, 6))
top_sports = df['sport'].value_counts().head(10).index
df_top_sports = df[df['sport'].isin(top_sports)]

sns.boxplot(data=df_top_sports, x='sport', y='earnings', palette='Set2', legend=
plt.xticks(rotation=45, ha='right')
plt.title('Earnings Distribution by Sport (Top 10 Sports)', fontsize=14, fontwei
plt.xlabel('Sport')
plt.ylabel('Earnings ($ Million)')
plt.tight_layout()
plt.show()
```

```
C:\Users\ngaka\AppData\Local\Temp\ipykernel_21416\1097176992.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.boxplot(data=df_top_sports, x='sport', y='earnings', palette='Set2', legend
=False)
```



Boxing: highest outliers($300M+) but variable distribution

Soccer: highest median (~$65M), most consistent high earners

Baseball: lowest median (~$20M), tightest range

Individual sports show wider variability than team sports

Outliers present in most sports - superstar effect universal

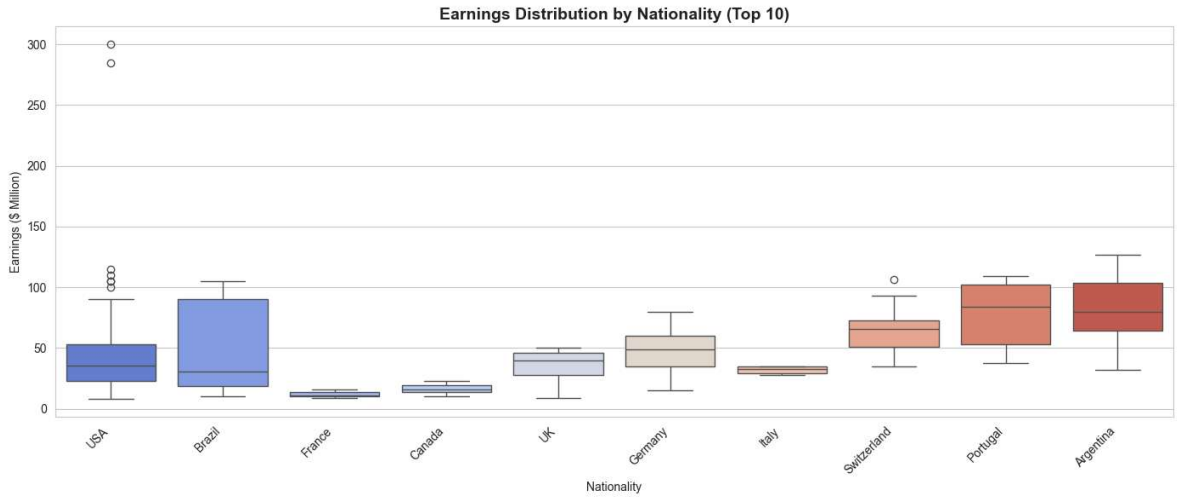
In [37]:
```
# Earnings by Nationality (Top 10)
plt.figure(figsize=(14, 6))
top_nations = df['nationality'].value_counts().head(10).index
df_top_nations = df[df['nationality'].isin(top_nations)]

sns.boxplot(data=df_top_nations, x='nationality', y='earnings', palette='coolwar
plt.xticks(rotation=45, ha='right')
plt.title('Earnings Distribution by Nationality (Top 10)', fontsize=14, fontweig
plt.xlabel('Nationality')
plt.ylabel('Earnings ($ Million)')
plt.tight_layout()
plt.show()
```

```
C:\Users\ngaka\AppData\Local\Temp\ipykernel_21416\1105158249.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.boxplot(data=df_top_nations, x='nationality', y='earnings', palette='coolwa
rm')
```



USA: most outliers ($300M), widest range,

moderate median (~$30M)

Portugal: highest median (~$85M) - Cristiano Ronaldo effect

Argentina: high consistent earnings (~$75M) - Messi effect

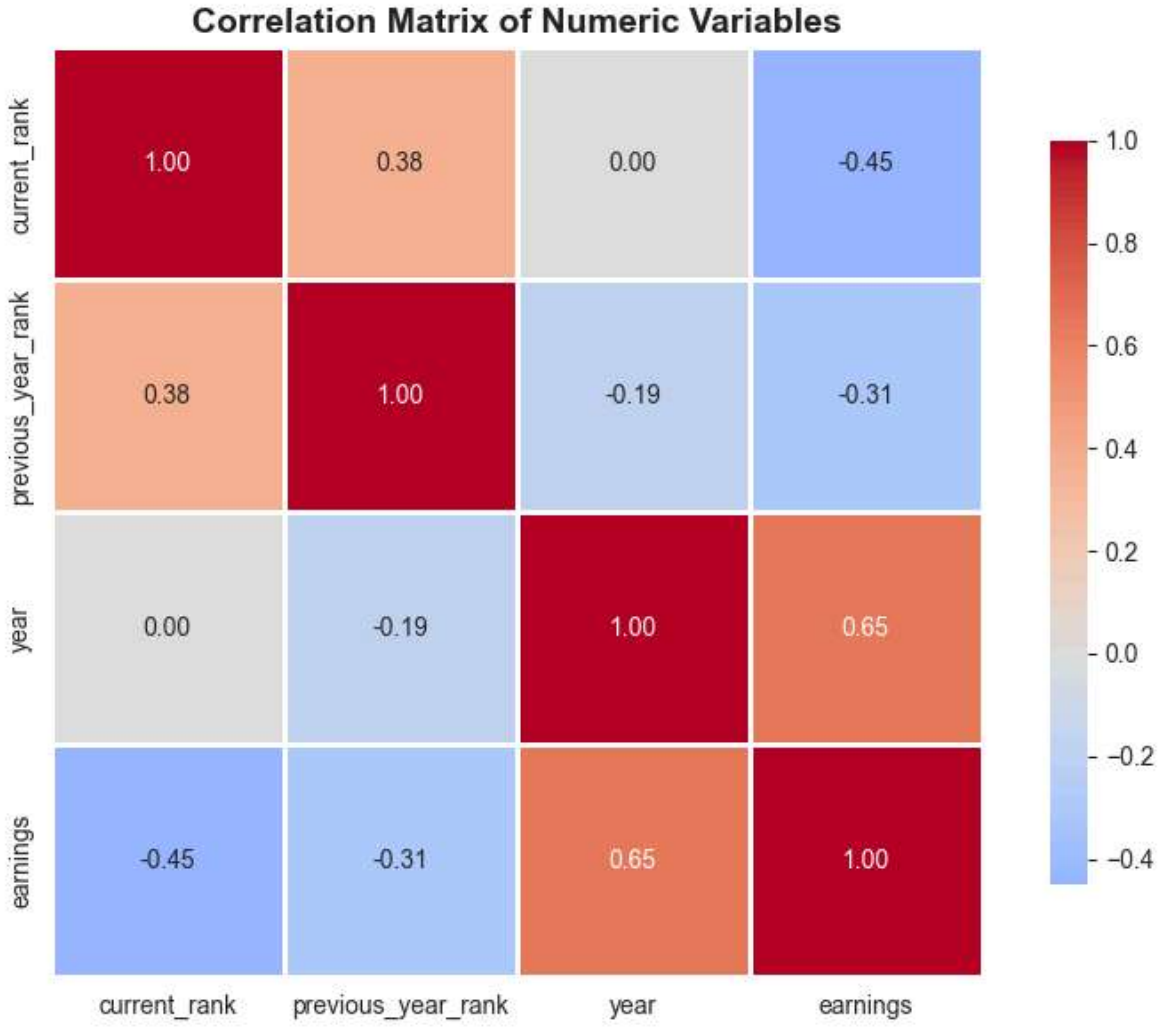
Switzerland, Brazil: high medians due to tennis/soccer stars

France, Canada, Italy: lowest medians, tight distributions

Small sample countries heavily influenced by individual superstars

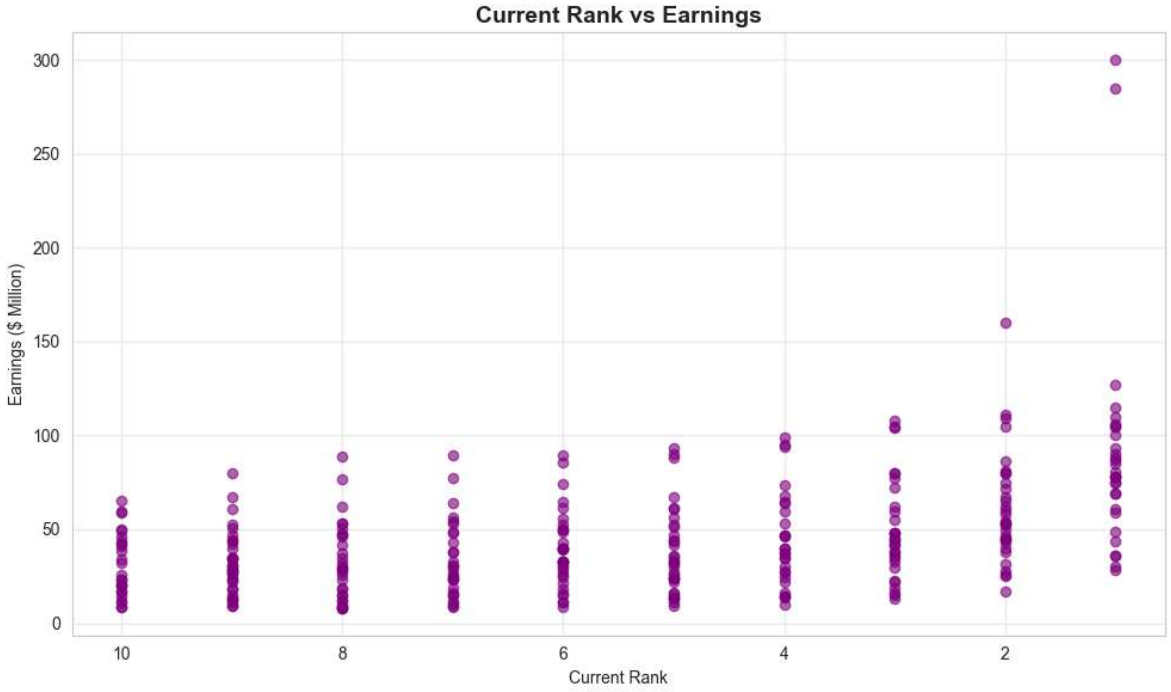
In [38]:
```
# Correlation Matrix (Numeric Variables)
plt.figure(figsize=(8, 6))
numeric_cols = ['current_rank', 'previous_year_rank', 'year', 'earnings']
correlation_matrix = df[numeric_cols].corr()

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0,
            fmt='.2f', square=True, linewidths=1, cbar_kws={"shrink": 0.8})
plt.title('Correlation Matrix of Numeric Variables', fontsize=14, fontweight='bo
plt.tight_layout()
plt.show()
```



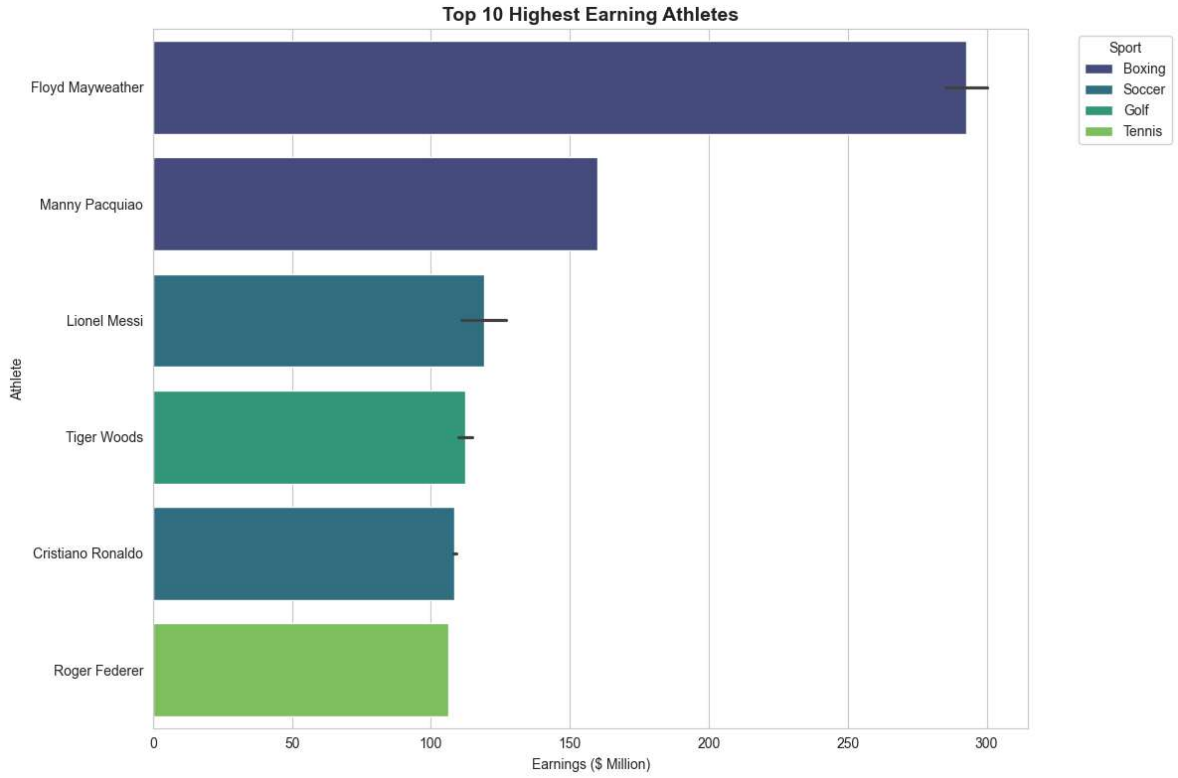
The correlation analysis reveals that earnings strongly correlate with year (r=0.65), suggesting significant earnings growth over time. Rankings show moderate consistency year-over-year (r=0.38), and as expected, better rankings (lower numbers) correlate with higher earnings (r=-0.45).

In [39]:
```
# Rank vs Earnings (Scatterplot)
plt.figure(figsize=(10, 6))
plt.scatter(df['current_rank'], df['earnings'], alpha=0.6, color='purple')
plt.title('Current Rank vs Earnings', fontsize=14, fontweight='bold')
plt.xlabel('Current Rank')
plt.ylabel('Earnings ($ Million)')
plt.gca().invert_xaxis()  # Rank 1 should be on the left
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```



The scatter plot reveals a clear negative relationship between rank and earnings — top-ranked athletes earn significantly more. However, the data exhibits heteroscedasticity, with earnings variance increasing at higher ranks. Notable outliers at Rank 1 (earning $285-300M) represent elite athletes whose earnings far exceed peers, suggesting factors beyond ranking (endorsements, marketability) drive top-tier earnings.

In [40]:
```
# Top 10 Highest Earning Athletes
plt.figure(figsize=(12, 8))
top_10 = df.nlargest(10, 'earnings')
sns.barplot(data=top_10, y='name', x='earnings', hue='sport', dodge=False, palet
plt.title('Top 10 Highest Earning Athletes', fontsize=14, fontweight='bold')
plt.xlabel('Earnings ($ Million)')
plt.ylabel('Athlete')
plt.legend(title='Sport', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



The top earners analysis reveals combat sports athletes(boxing) dominate the highest earnings, with the top earner at $300M.

Nearly double the second place athlete. Football(soccer) stars Messi and Ronaldo earn comparably($105-120M), reflecting the sport's global commercial appeal. individual sports like gold and tennis also produce top earners, suggesting personal brand strength is as crucial as sport popularity for maximizing athlete earnings

KEY FINDINDS:

The earnings distribution of top athletes is heavily right-skewed, with the majority (67.4%) earning between $0–50M,

while a small number of extreme outliers—most notably Floyd Mayweather—extend the upper tail beyond $300M.

These outliers inflate the mean $(45.52M)above the median$(39M), illustrating a clear superstar effect, where a few elite athletes earn multiples of the typical top athlete.

Across sports, basketball dominates total appearances, followed by boxing and golf, with the top three sports accounting for 56.8% of all appearances. Individual sports such as boxing, golf, and tennis show greater earnings variability and higher extremes than team sports. Boxing produces the largest outliers, while soccer exhibits the highest and most consistent median earnings, reflecting its global commercial appeal.

Geographically, the United States overwhelmingly dominates athlete appearances (68.4%), driven by large domestic sports markets and endorsement opportunities. Other countries show much smaller sample sizes, with earnings often heavily influenced by individual superstars such as Ronaldo (Portugal) and Messi (Argentina).

Over time, athlete earnings display a strong upward trend, increasing roughly sevenfold from 1990 to 2020, with accelerated growth during the 2010s. Sharp peaks in 2015 and 2019 are driven by individual mega-earners, highlighting the growing importance of media rights, globalization, and personal branding.

Correlation and scatter plot analyses confirm that earnings increase over time ($r = 0.65$) and are inversely related to ranking ($r = -0.45$), though ranking alone does not fully explain earnings. The increasing variance at higher ranks and the presence of extreme outliers suggest that endorsements, marketability, and global reach play a critical role in determining top-tier athlete earnings beyond athletic performance alone.

Report written by: Lerato Kgosimore

In [ ]: