AAMNA NAZ AWAN

BIG DATA ANALYTICS

FINAL EXAM

DATASET: **Football data from transfer markt**

ABOUT DATA SET:

Clean, structured and automatically updated football data from Transfermarkt, including

60,000+ games from many seasons on all major competitions

400+ clubs from those competitions

30,000+ players from those clubs

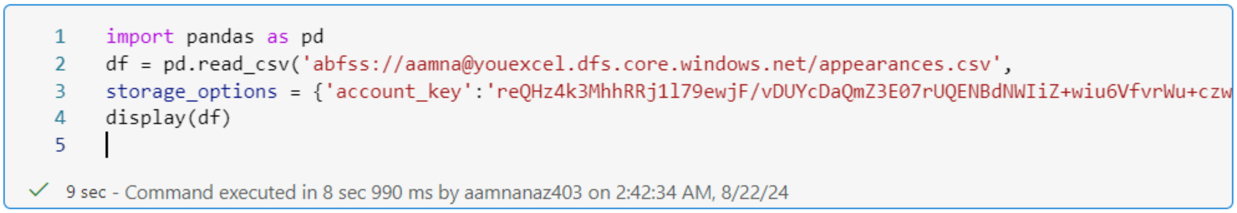
400,000+ player market valuations historical records

1,200,000+ player appearance records from all games

FILE NAMES:

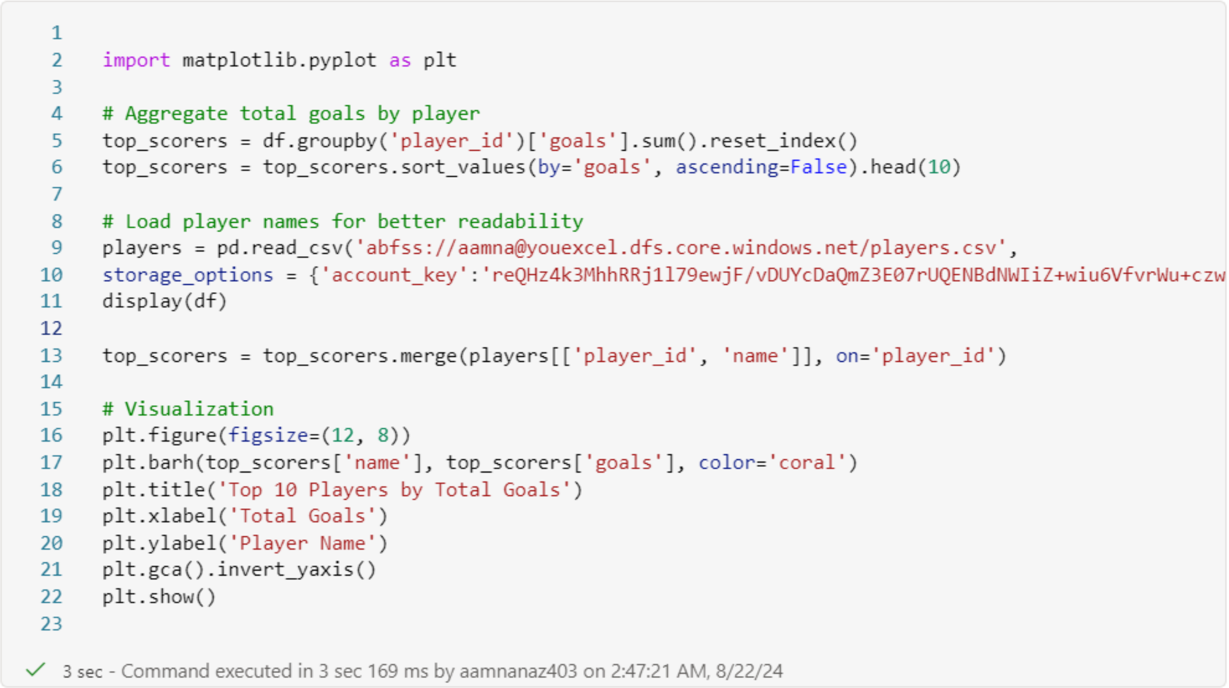
1. appearances.csv
2. club_games.csv
3. clubs.csv
4. competitions.csv
5. game_events.csv
6. game_lineups.csv
7. games.csv
8. player_valuations.csv
9. players.csv
10. transfers.csv

QUERY NO:1 **Top Players by Total Goals**



```python
import pandas as pd
df = pd.read_csv('abfss://aamna@youexcel.dfs.core.windows.net/appearances.csv',
storage_options = {'account_key':'reQHz4k3MhhRRj1179ewjF/vDUYcDaQmZ3E07rUQENBdNWIiZ+wiu6VfvrWu+czw
display(df)

```

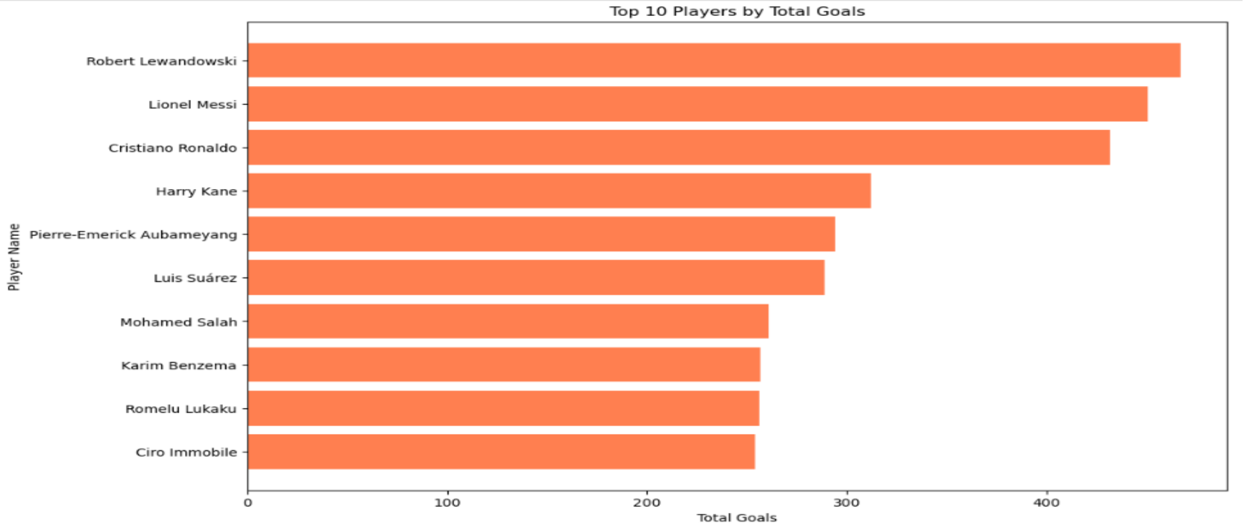
✓ 9 sec - Command executed in 8 sec 990 ms by aamnanaz403 on 2:42:34 AM, 8/22/24



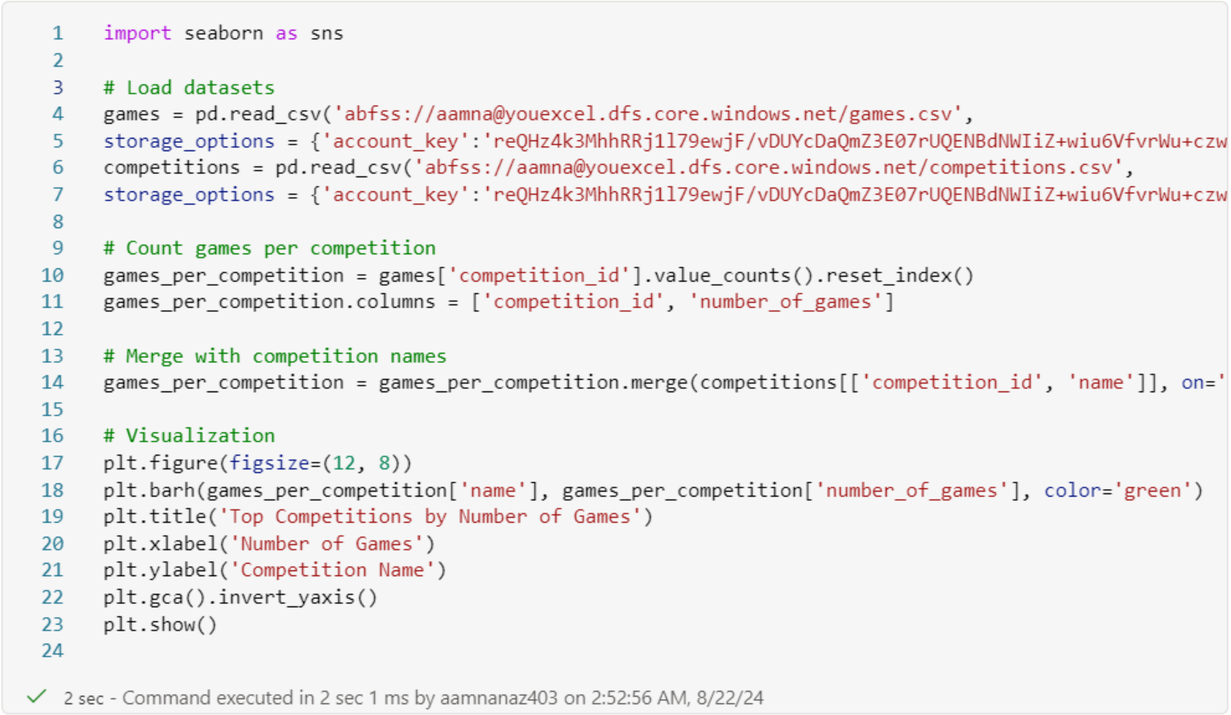
```python

import matplotlib.pyplot as plt

# Aggregate total goals by player
top_scorers = df.groupby('player_id')['goals'].sum().reset_index()
top_scorers = top_scorers.sort_values(by='goals', ascending=False).head(10)

# Load player names for better readability
players = pd.read_csv('abfss://aamna@youexcel.dfs.core.windows.net/players.csv',
storage_options = {'account_key':'reQHz4k3MhhRRj1179ewjF/vDUYcDaQmZ3E07rUQENBdNWIiZ+wiu6VfvrWu+czw
display(df)

top_scorers = top_scorers.merge(players[['player_id', 'name']], on='player_id')

# Visualization
plt.figure(figsize=(12, 8))
plt.barh(top_scorers['name'], top_scorers['goals'], color='coral')
plt.title('Top 10 Players by Total Goals')
plt.xlabel('Total Goals')
plt.ylabel('Player Name')
plt.gca().invert_yaxis()
plt.show()

```

✓ 3 sec - Command executed in 3 sec 169 ms by aamnanaz403 on 2:47:21 AM, 8/22/24

OUTPUT:-



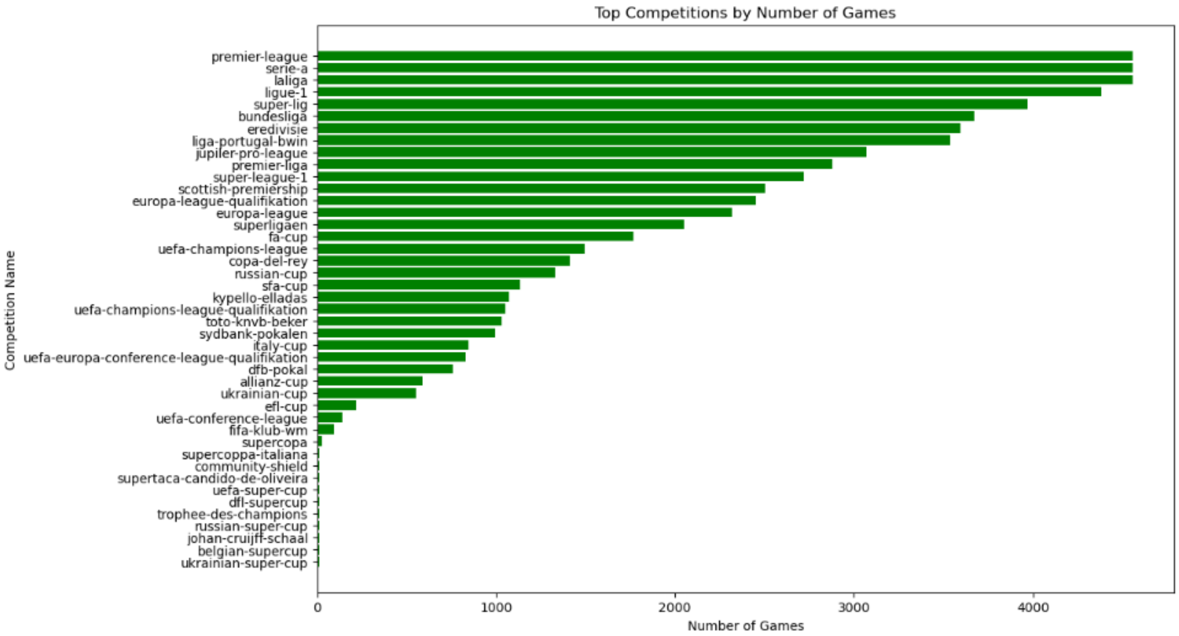
QUERY NO:2 **Top Competitions by Number of Games**



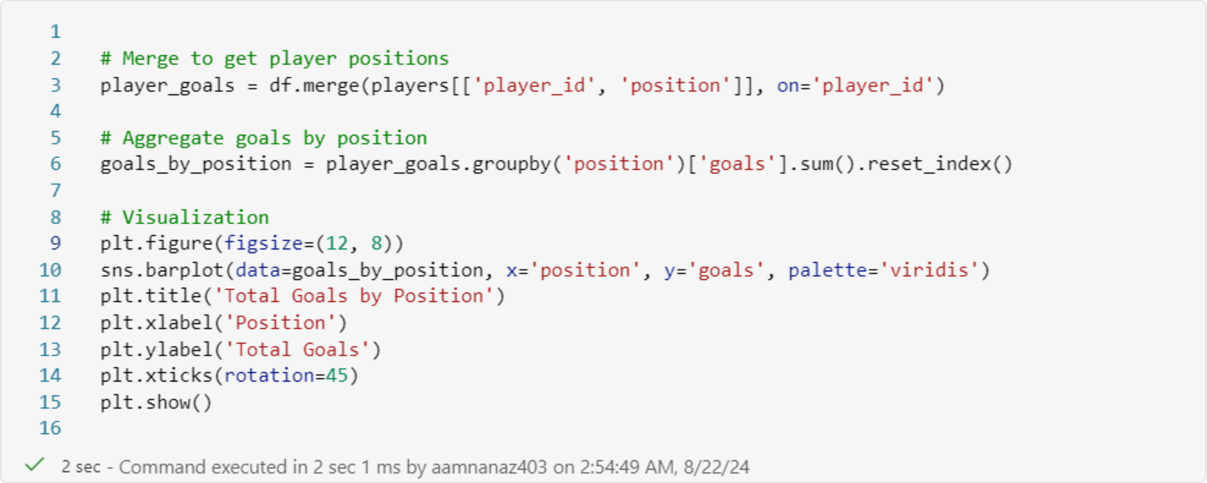
```python
import seaborn as sns

# Load datasets
games = pd.read_csv('abfss://aamna@youexcel.dfs.core.windows.net/games.csv',
storage_options = {'account_key':'reQHz4k3MhhRRj1l79ewjF/vDUYcDaQmZ3E07rUQENBdNWIiZ+wiu6VfvrWu+czw
competitions = pd.read_csv('abfss://aamna@youexcel.dfs.core.windows.net/competitions.csv',
storage_options = {'account_key':'reQHz4k3MhhRRj1l79ewjF/vDUYcDaQmZ3E07rUQENBdNWIiZ+wiu6VfvrWu+czw

# Count games per competition
games_per_competition = games['competition_id'].value_counts().reset_index()
games_per_competition.columns = ['competition_id', 'number_of_games']

# Merge with competition names
games_per_competition = games_per_competition.merge(competitions[['competition_id', 'name']], on='

# Visualization
plt.figure(figsize=(12, 8))
plt.barh(games_per_competition['name'], games_per_competition['number_of_games'], color='green')
plt.title('Top Competitions by Number of Games')
plt.xlabel('Number of Games')
plt.ylabel('Competition Name')
plt.gca().invert_yaxis()
plt.show()

```

✓ 2 sec - Command executed in 2 sec 1 ms by aamnanaz403 on 2:52:56 AM, 8/22/24

OUTPUT:-



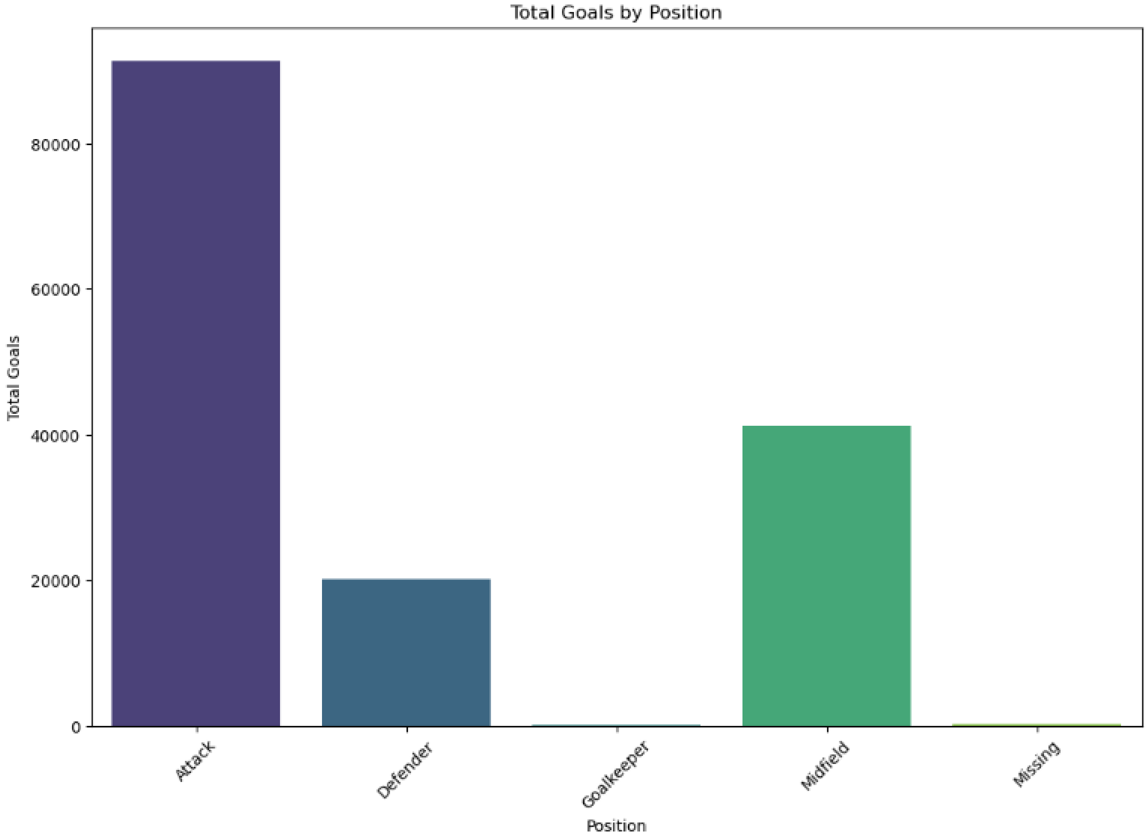
QUERY : 3 **Goal Distribution by Position**



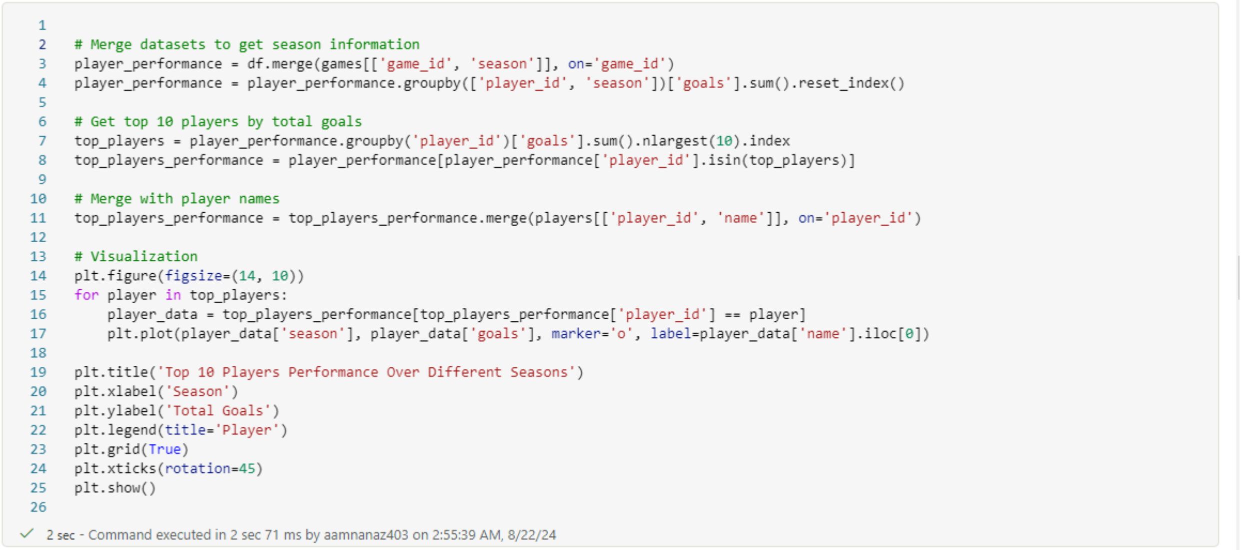
```python

# Merge to get player positions
player_goals = df.merge(players[['player_id', 'position']], on='player_id')

# Aggregate goals by position
goals_by_position = player_goals.groupby('position')['goals'].sum().reset_index()

# Visualization
plt.figure(figsize=(12, 8))
sns.barplot(data=goals_by_position, x='position', y='goals', palette='viridis')
plt.title('Total Goals by Position')
plt.xlabel('Position')
plt.ylabel('Total Goals')
plt.xticks(rotation=45)
plt.show()

```

✓ 2 sec - Command executed in 2 sec 1 ms by aamnanaz403 on 2:54:49 AM, 8/22/24

OUTPUT:-



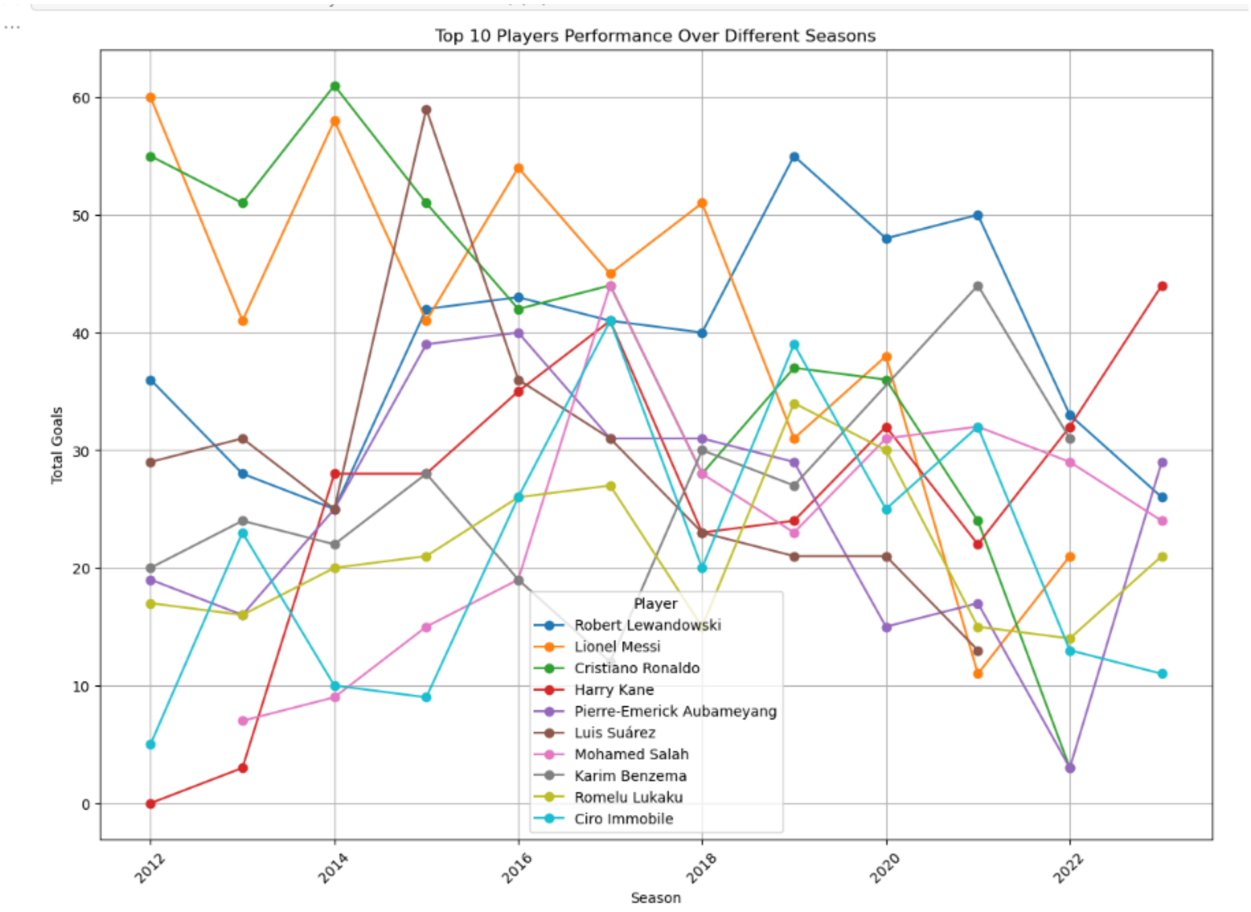
QUERY : 4 **Player Performance Over Different Seasons**



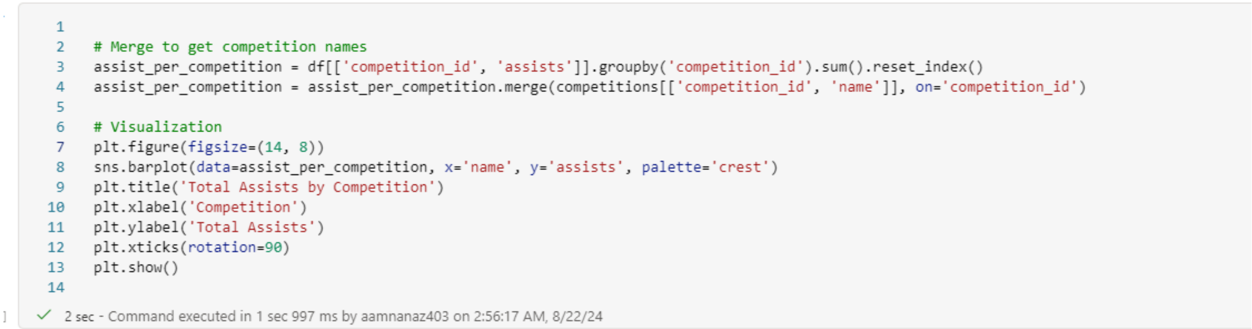
```
# Merge datasets to get season information
player_performance = df.merge(games[['game_id', 'season']], on='game_id')
player_performance = player_performance.groupby(['player_id', 'season'])['goals'].sum().reset_index()

# Get top 10 players by total goals
top_players = player_performance.groupby('player_id')['goals'].sum().nlargest(10).index
top_players_performance = player_performance[player_performance['player_id'].isin(top_players)]

# Merge with player names
top_players_performance = top_players_performance.merge(players[['player_id', 'name']], on='player_id')

# Visualization
plt.figure(figsize=(14, 10))
for player in top_players:
    player_data = top_players_performance[top_players_performance['player_id'] == player]
    plt.plot(player_data['season'], player_data['goals'], marker='o', label=player_data['name'].iloc[0])

plt.title('Top 10 Players Performance Over Different Seasons')
plt.xlabel('Season')
plt.ylabel('Total Goals')
plt.legend(title='Player')
plt.grid(True)
plt.xticks(rotation=45)
plt.show()

```

✓ 2 sec - Command executed in 2 sec 71 ms by aamnanaz403 on 2:55:39 AM, 8/22/24

OUTPUT:-



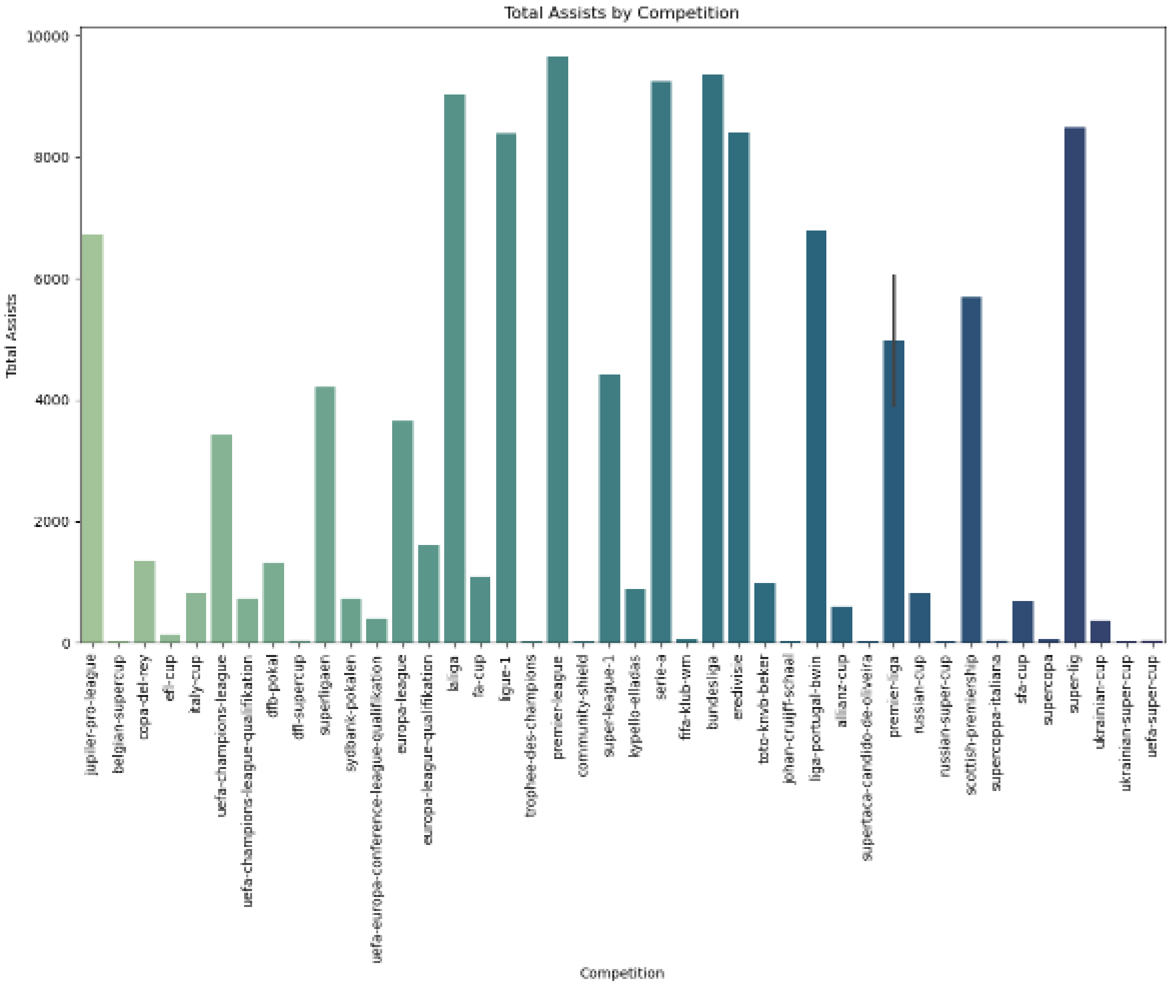
QUERY: 5 **Comparison of Total Assists by Competition**



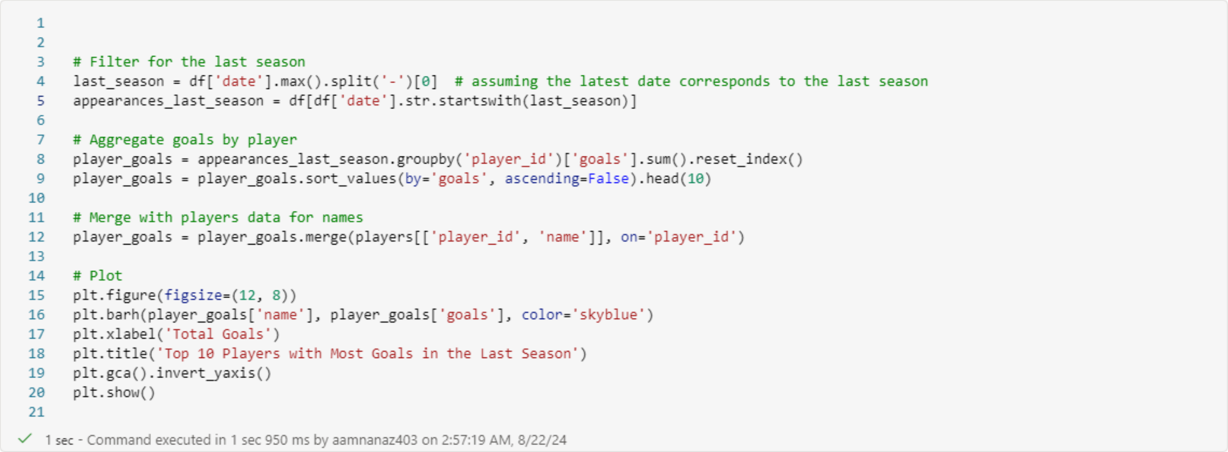
```

# Merge to get competition names
assist_per_competition = df[['competition_id', 'assists']].groupby('competition_id').sum().reset_index()
assist_per_competition = assist_per_competition.merge(competitions[['competition_id', 'name']], on='competition_id')

# Visualization
plt.figure(figsize=(14, 8))
sns.barplot(data=assist_per_competition, x='name', y='assists', palette='crest')
plt.title('Total Assists by Competition')
plt.xlabel('Competition')
plt.ylabel('Total Assists')
plt.xticks(rotation=90)
plt.show()

```

✓ 2 sec - Command executed in 1 sec 997 ms by aamnanaz403 on 2:56:17 AM, 8/22/24

OUTPUT:-



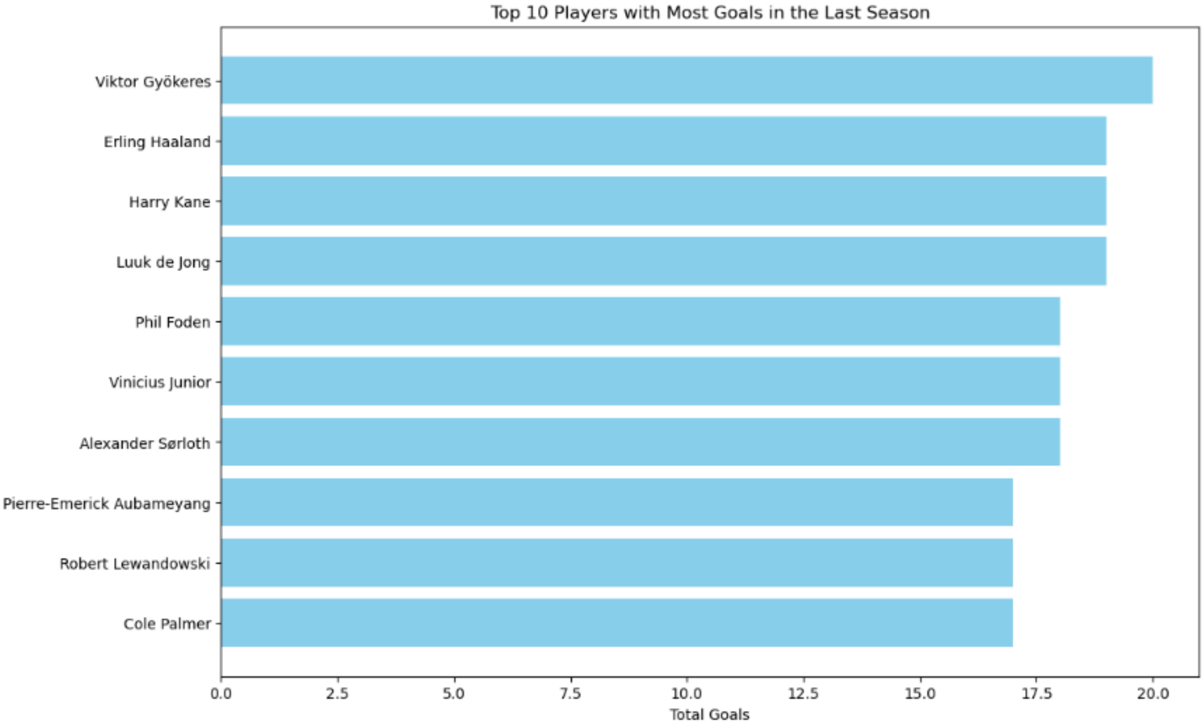
QUERY: 6 **Top 10 Players with Most Goals in the Last Season**



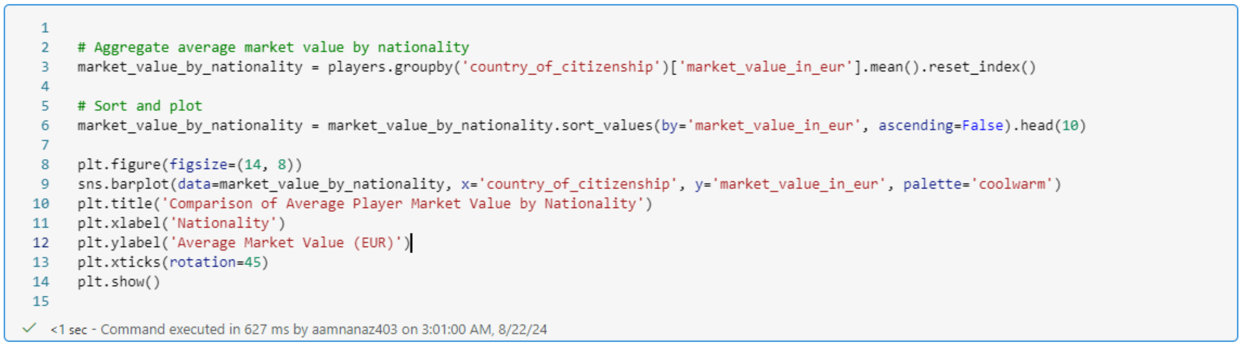
```


# Filter for the last season
last_season = df['date'].max().split('-')[0]  # assuming the latest date corresponds to the last season
appearances_last_season = df[df['date'].str.startswith(last_season)]

# Aggregate goals by player
player_goals = appearances_last_season.groupby('player_id')['goals'].sum().reset_index()
player_goals = player_goals.sort_values(by='goals', ascending=False).head(10)

# Merge with players data for names
player_goals = player_goals.merge(players[['player_id', 'name']], on='player_id')

# Plot
plt.figure(figsize=(12, 8))
plt.barh(player_goals['name'], player_goals['goals'], color='skyblue')
plt.xlabel('Total Goals')
plt.title('Top 10 Players with Most Goals in the Last Season')
plt.gca().invert_yaxis()
plt.show()

```

✓ 1 sec - Command executed in 1 sec 950 ms by aamnanaz403 on 2:57:19 AM, 8/22/24

OUTPUT:-



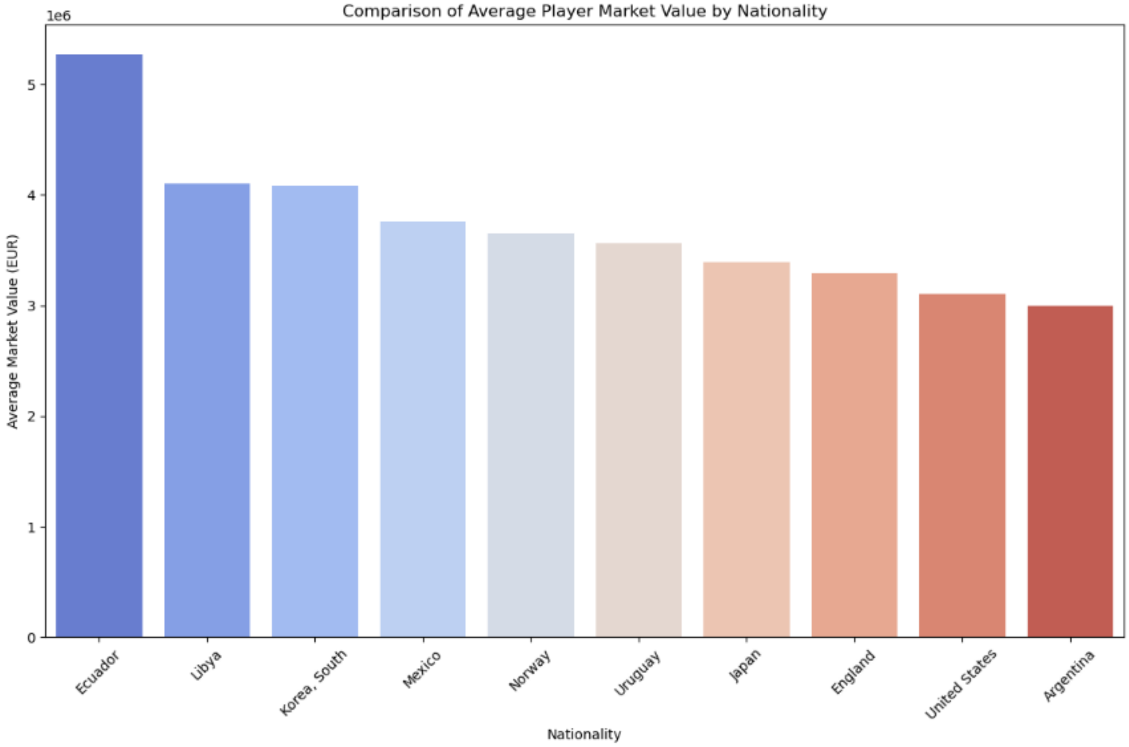
QUERY: 7 **Comparison of Average Player Market Value by Nationality**



```

# Aggregate average market value by nationality
market_value_by_nationality = players.groupby('country_of_citizenship')['market_value_in_eur'].mean().reset_index()

# Sort and plot
market_value_by_nationality = market_value_by_nationality.sort_values(by='market_value_in_eur', ascending=False).head(10)

plt.figure(figsize=(14, 8))
sns.barplot(data=market_value_by_nationality, x='country_of_citizenship', y='market_value_in_eur', palette='coolwarm')
plt.title('Comparison of Average Player Market Value by Nationality')
plt.xlabel('Nationality')
plt.ylabel('Average Market Value (EUR)')
plt.xticks(rotation=45)
plt.show()

```

✓ <1 sec - Command executed in 627 ms by aamnanaz403 on 3:01:00 AM, 8/22/24

OUTPUT:-


Comparison of Average Player Market Value by Nationality
1e6
Average Market Value (EUR)
Ecuador
Libya
Korea, South
Mexico
Norway
Uruguay
Japan
England
United States
Argentina
Nationality