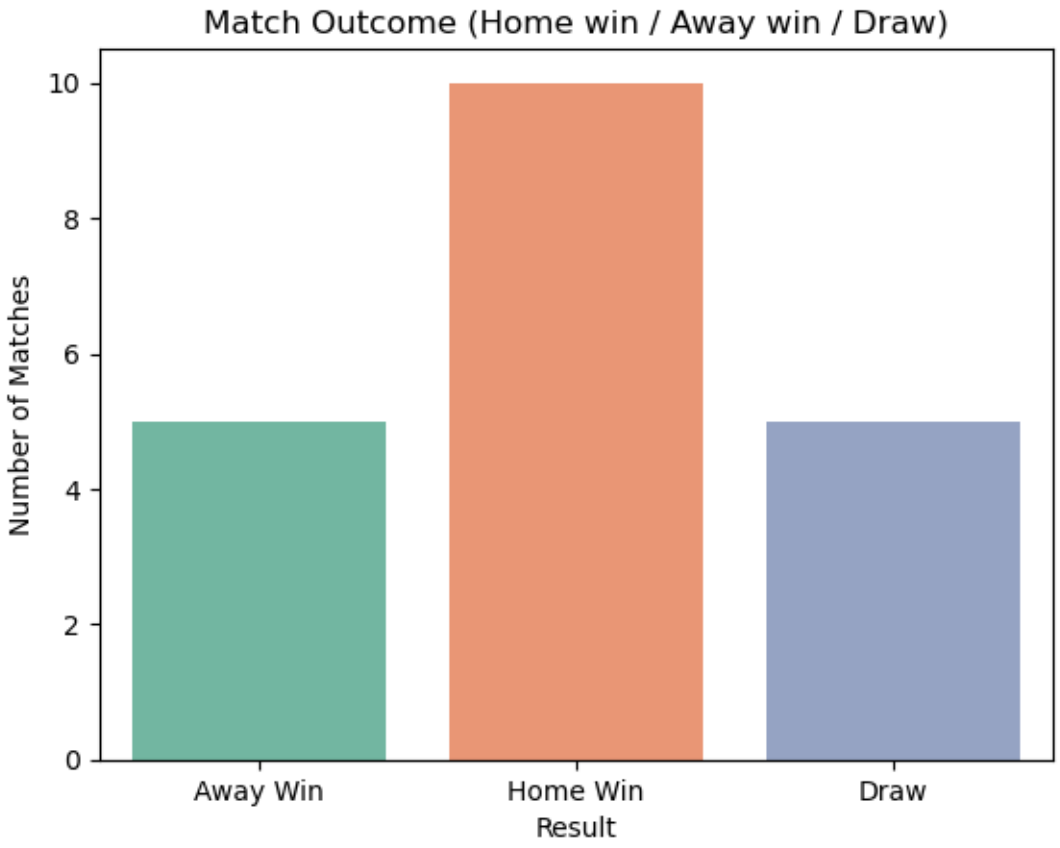
Football Matches Analysis

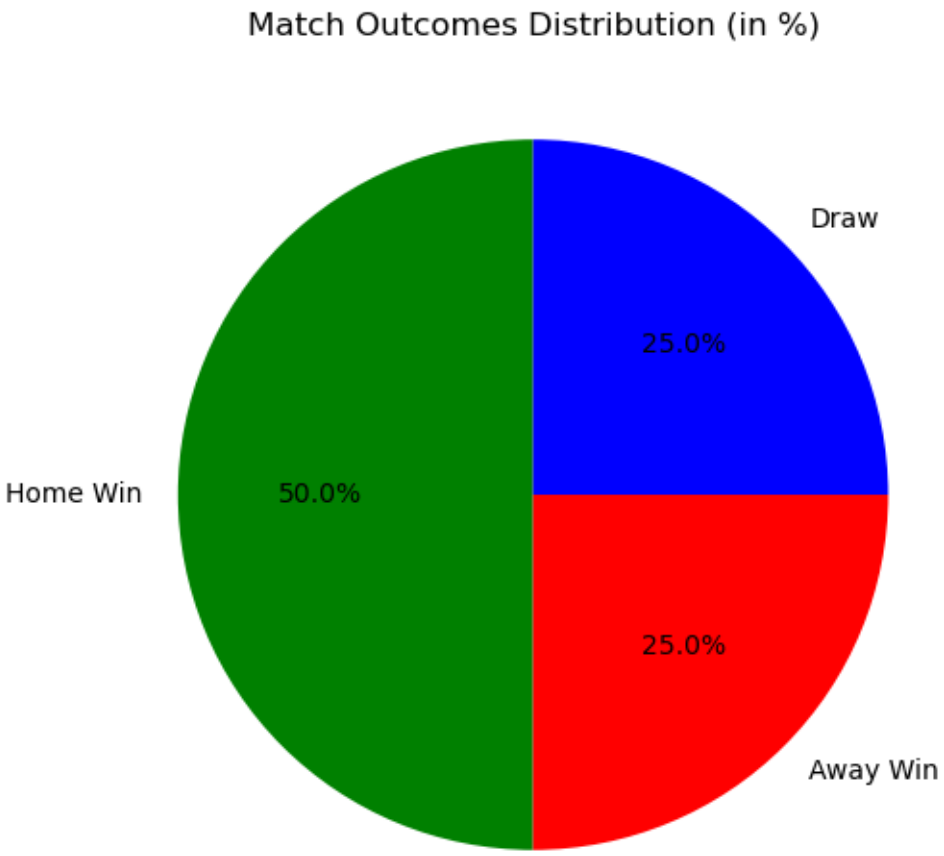
August 25, 2025

[46]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("football_matches.csv")
```

```
Result
Home Win    10
Away Win     5
Draw         5
Name: count, dtype: int64
```

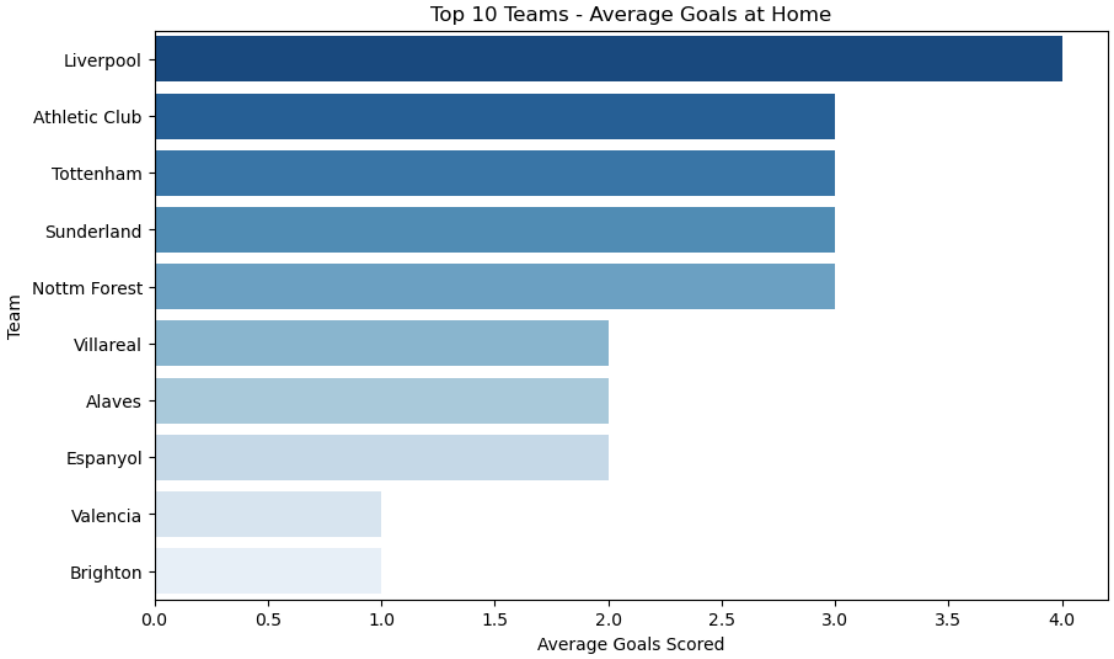


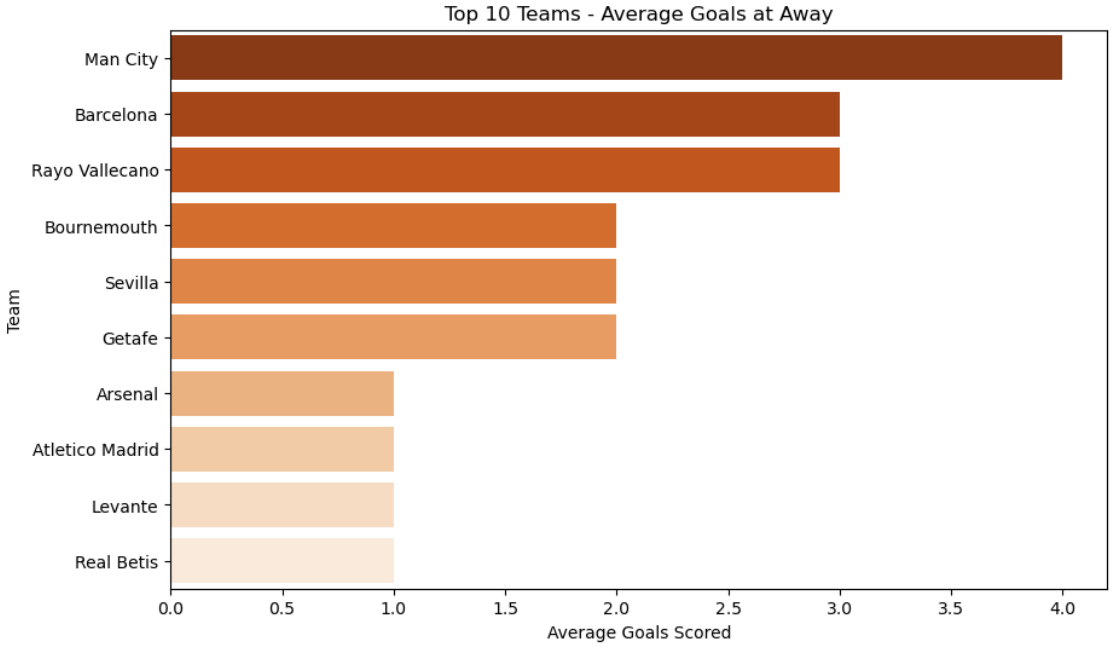
```
Percentages of Matches Outcomes
 Result
Home Win    50.0
Away Win    25.0
Draw        25.0
Name: count, dtype: float64
```



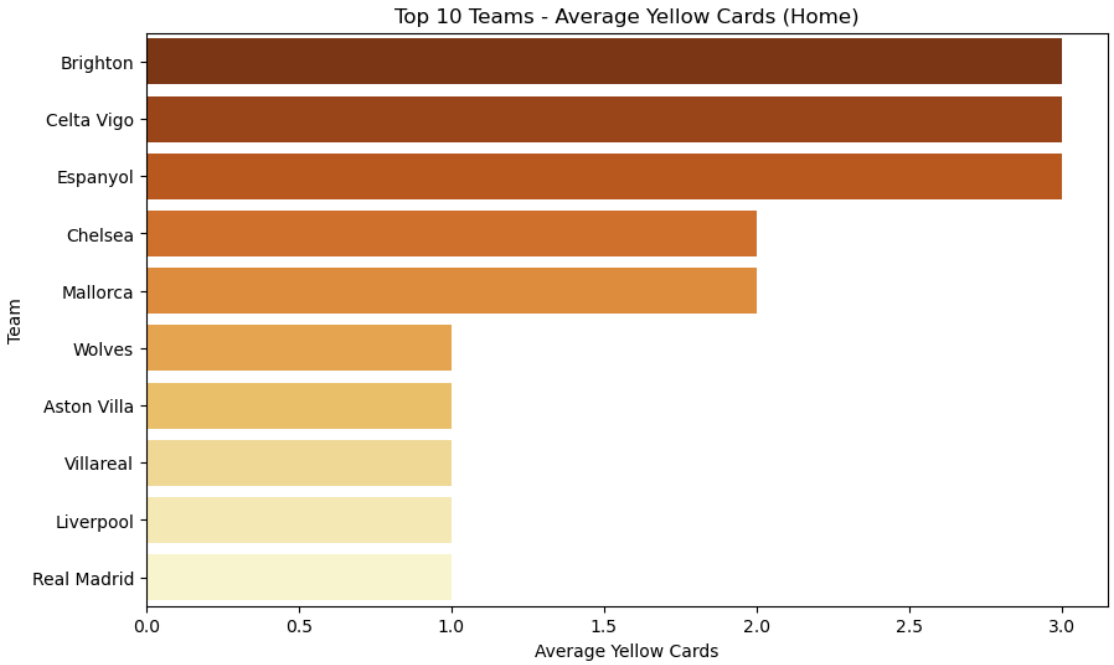

```
Top 5 Teams - Average Goals at Home:
 Home_Team
Liverpool        4.0
Athletic Club    3.0
Tottenham        3.0
Sunderland       3.0
Nottm Forest     3.0
Name: Home_Team_Score, dtype: float64
Top 5 Teams - Average Goals at Away:
```

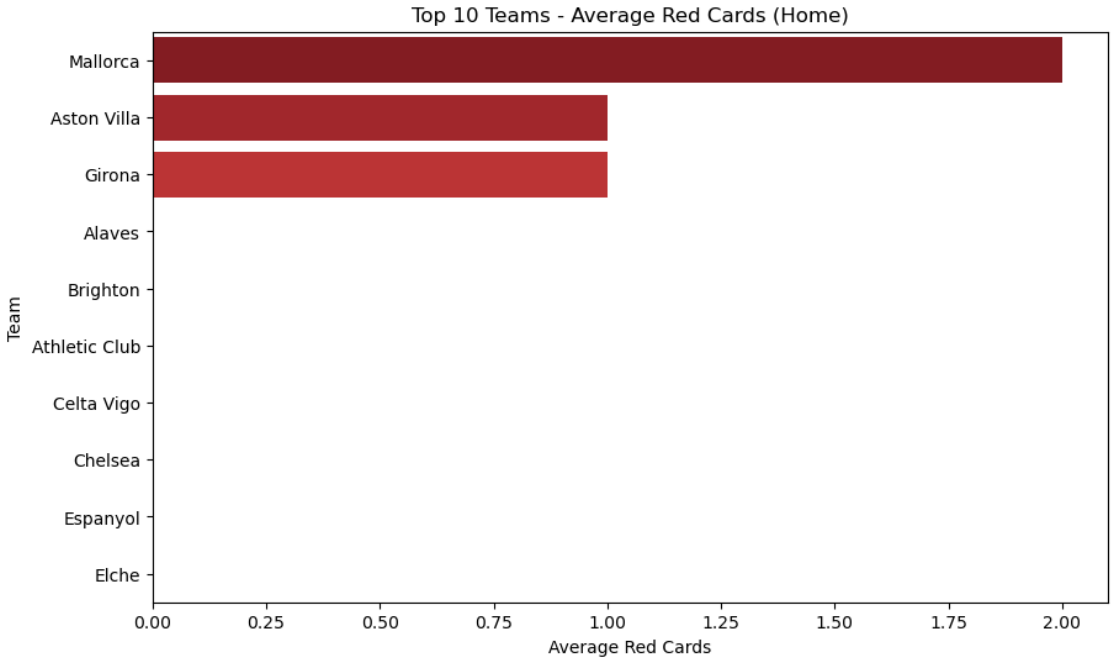
```
 Away_Team
Man City          4.0
Barcelona         3.0
Rayo Vallecano    3.0
Bournemouth       2.0
Sevilla           2.0
Name: Away_Team_Score, dtype: float64
```

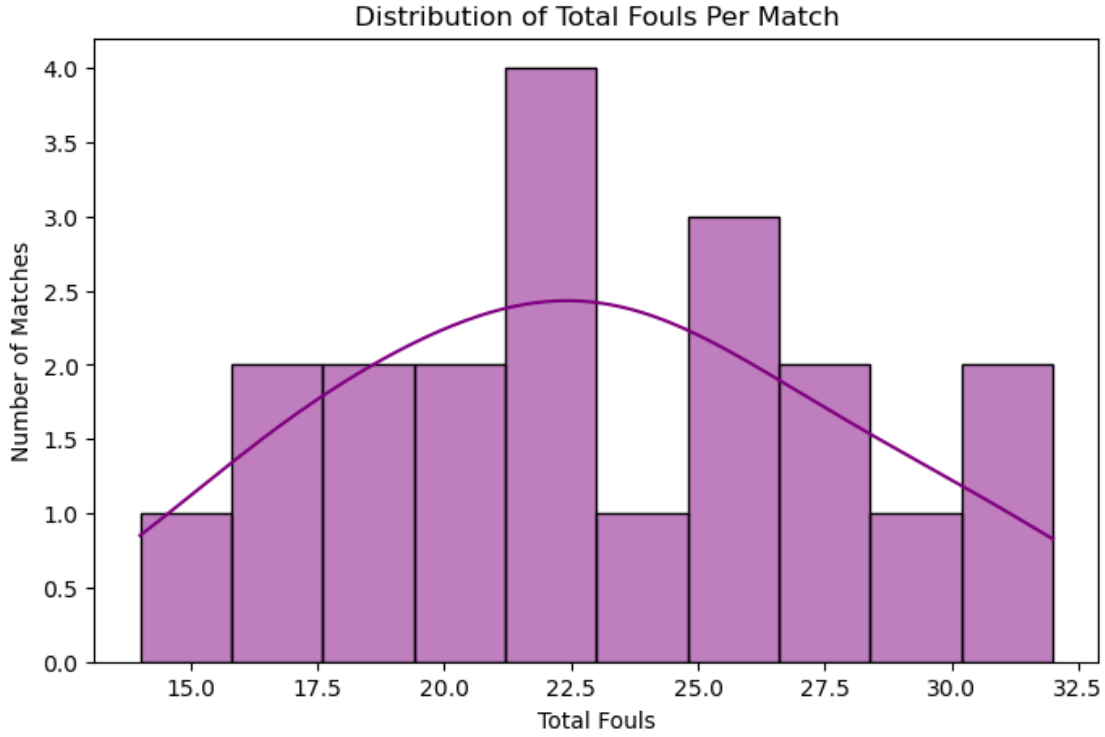




```
Top 5 Teams - Average Yellow Cards (Home):
            Home_Team_Yellowcard
Home_Team
Brighton                     3.0
Celta Vigo                   3.0
Espanyol                     3.0
Chelsea                      2.0
Mallorca                     2.0
Top 5 Teams - Average Red Cards (Home):
             Home_Team_Redcard
Home_Team
Mallorca                   2.0
Aston Villa                1.0
Girona                     1.0
Alaves                     0.0
Brighton                   0.0
```







[47]:
```
print(df.head())
```

```
   ID        Date    Home_Team       Away_Team  Home_Team_Score  \
0   1  15-08-2025       Girona  Rayo Vallecano                1
1   2  16-08-2025    Villareal          Oviedo                2
2   3  16-08-2025     Mallorca       Barcelona                0
3   4  16-08-2025    Liverpool     Bournemouth                4
4   5  16-08-2025  Aston Villa       Newcastle                0

   Away_Team_Score  Home_Team_Yellowcard  Away_Team_Yellowcard  \
0                3                     0                     1
1                0                     1                     0
2                3                     2                     1
3                2                     1                     2
4                0                     1                     1

   Home_Team_Redcard  Away_Team_Redcard  …  Away_Team_Passes  \
0                  1                  0  …               481
1                  0                  1  …               326
2                  2                  0  …               570
3                  0                  0  …               299
4                  1                  0  …               451

   Home_Team_Fouls  Away_Team_Fouls  Home_Team_Offside  Away_Team_Offside  \
0                8               17                  1                  1
1               10                8                  0                  1
2                8               17                  2                  3
3                7               10                  2                  2
4               13               11                  2                  1

   Home_Team_Corner  Away_Team_Corner          League    Result  Total_Fouls
0                 2                 4         La Liga  Away Win           25
1                10                 1         La Liga  Home Win           18
2                 3                 6         La Liga  Away Win           25
3                 6                 7  Premier League  Home Win           17
4                 3                 6  Premier League      Draw           24

[5 rows x 27 columns]
```

[48]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 27 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID                      20 non-null     int64
 1   Date                    20 non-null     object
 2   Home_Team               20 non-null     object
 3   Away_Team               20 non-null     object
```

```
 4   Home_Team_Score          20 non-null     int64
 5   Away_Team_Score          20 non-null     int64
 6   Home_Team_Yellowcard     20 non-null     int64
 7   Away_Team_Yellowcard     20 non-null     int64
 8   Home_Team_Redcard        20 non-null     int64
 9   Away_Team_Redcard        20 non-null     int64
 10  Home_Team_Shots          20 non-null     int64
 11  Away_Team_Shots          20 non-null     int64
 12  Home_Team_ShotsonTarget  20 non-null     int64
 13  Away_Team_ShotsonTarget  20 non-null     int64
 14  Home_Team_Possession     20 non-null     int64
 15  Away_Team_Possession     20 non-null     int64
 16  Home_Team_Passes         20 non-null     int64
 17  Away_Team_Passes         20 non-null     int64
 18  Home_Team_Fouls          20 non-null     int64
 19  Away_Team_Fouls          20 non-null     int64
 20  Home_Team_Offside        20 non-null     int64
 21  Away_Team_Offside        20 non-null     int64
 22  Home_Team_Corner         20 non-null     int64
 23  Away_Team_Corner         20 non-null     int64
 24  League                   20 non-null     object
 25  Result                   20 non-null     object
 26  Total_Fouls              20 non-null     int64
dtypes: int64(22), object(5)
memory usage: 4.3+ KB
None
```

[49]:
```
print(df.describe())
```

```
              ID  Home_Team_Score  Away_Team_Score  Home_Team_Yellowcard  \
count  20.00000        20.000000        20.000000             20.000000
mean   10.50000         1.400000         1.150000              1.050000
std     5.91608         1.273206         1.182103              1.050063
min     1.00000         0.000000         0.000000              0.000000
25%     5.75000         0.000000         0.000000              0.000000
50%    10.50000         1.000000         1.000000              1.000000
75%    15.25000         2.250000         2.000000              1.250000
max    20.00000         4.000000         4.000000              3.000000

       Away_Team_Yellowcard  Home_Team_Redcard  Away_Team_Redcard  \
count             20.000000          20.000000          20.000000
mean               1.650000           0.200000           0.100000
std                1.089423           0.523148           0.307794
min                0.000000           0.000000           0.000000
25%                1.000000           0.000000           0.000000
50%                2.000000           0.000000           0.000000
75%                2.000000           0.000000           0.000000
max                4.000000           2.000000           1.000000
```

```
       Home_Team_Shots  Away_Team_Shots  Home_Team_ShotsonTarget  …  \
count        20.000000        20.000000                20.000000  …
mean         13.350000        11.350000                 4.300000  …
std           6.268342         4.793362                 2.451637  …
min           3.000000         2.000000                 1.000000  …
25%           9.000000         8.500000                 3.000000  …
50%          12.500000        11.500000                 3.500000  …
75%          18.250000        14.250000                 5.000000  …
max          25.000000        24.000000                10.000000  …

       Away_Team_Possession  Home_Team_Passes  Away_Team_Passes  \
count             20.000000         20.000000         20.000000
mean              45.500000        464.600000        389.800000
std               13.449711        128.285619        120.080938
min               28.000000        225.000000        226.000000
25%               32.000000        392.250000        298.250000
50%               45.000000        449.000000        343.000000
75%               56.750000        525.250000        478.000000
max               72.000000        721.000000        618.000000

       Home_Team_Fouls  Away_Team_Fouls  Home_Team_Offside  Away_Team_Offside  \
count        20.000000          20.0000          20.000000          20.000000
mean         10.550000          12.3000           1.300000           1.600000
std           3.410124           3.4504           1.301821           1.930367
min           5.000000           7.0000           0.000000           0.000000
25%           8.000000          10.0000           0.000000           0.000000
50%          10.000000          12.0000           1.000000           1.000000
75%          14.000000          15.0000           2.000000           2.000000
max          16.000000          19.0000           4.000000           8.000000

       Home_Team_Corner  Away_Team_Corner  Total_Fouls
count         20.000000         20.000000    20.000000
mean           5.750000          3.750000    22.850000
std            3.160197          2.197487     4.987089
min            1.000000          0.000000    14.000000
25%            3.000000          2.000000    19.500000
50%            5.500000          3.500000    22.000000
75%            8.250000          5.250000    25.500000
max           11.000000          7.000000    32.000000

[8 rows x 22 columns]
```

[50]:
```
print(df.isnull().sum())
```

```
ID                      0
Date                    0
Home_Team               0
```

```
Away_Team                  0
Home_Team_Score            0
Away_Team_Score            0
Home_Team_Yellowcard       0
Away_Team_Yellowcard       0
Home_Team_Redcard          0
Away_Team_Redcard          0
Home_Team_Shots            0
Away_Team_Shots            0
Home_Team_ShotsonTarget    0
Away_Team_ShotsonTarget    0
Home_Team_Possession       0
Away_Team_Possession       0
Home_Team_Passes           0
Away_Team_Passes           0
Home_Team_Fouls            0
Away_Team_Fouls            0
Home_Team_Offside          0
Away_Team_Offside          0
Home_Team_Corner           0
Away_Team_Corner           0
League                     0
Result                     0
Total_Fouls                0
dtype: int64
```

[51]:
```
print(df.duplicated().sum())
```

```
0
```

[52]:
```
print(df.dtypes)
```

```
ID                          int64
Date                       object
Home_Team                  object
Away_Team                  object
Home_Team_Score             int64
Away_Team_Score             int64
Home_Team_Yellowcard        int64
Away_Team_Yellowcard        int64
Home_Team_Redcard           int64
Away_Team_Redcard           int64
Home_Team_Shots             int64
Away_Team_Shots             int64
Home_Team_ShotsonTarget     int64
Away_Team_ShotsonTarget     int64
Home_Team_Possession        int64
Away_Team_Possession        int64
```

```
Home_Team_Passes            int64
Away_Team_Passes            int64
Home_Team_Fouls             int64
Away_Team_Fouls             int64
Home_Team_Offside           int64
Away_Team_Offside           int64
Home_Team_Corner            int64
Away_Team_Corner            int64
League                     object
Result                     object
Total_Fouls                 int64
dtype: object
```

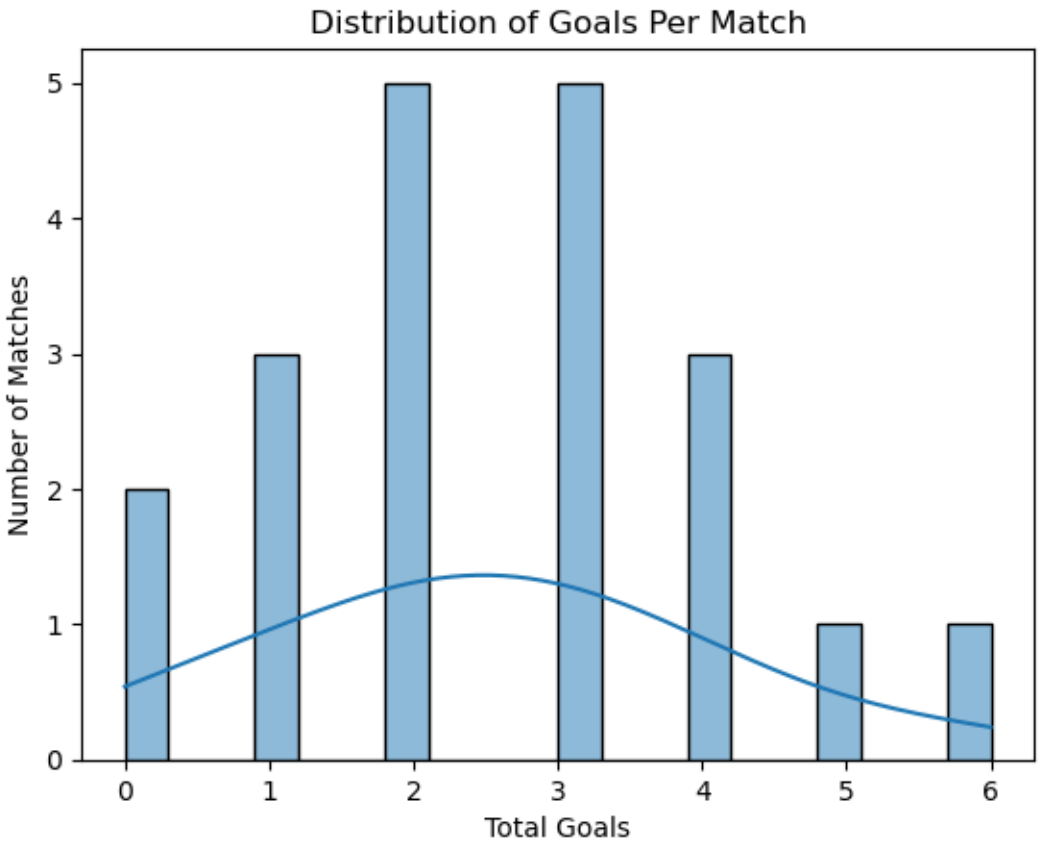
[53]:
```
print(df.columns)
```

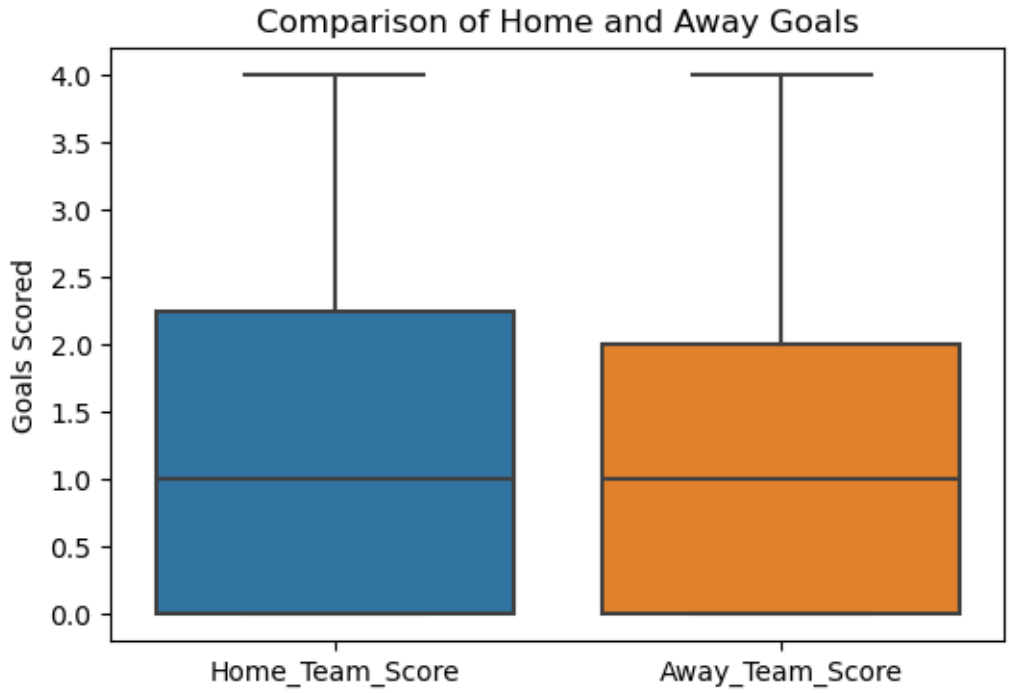
```
Index(['ID', 'Date', 'Home_Team', 'Away_Team', 'Home_Team_Score',
       'Away_Team_Score', 'Home_Team_Yellowcard', 'Away_Team_Yellowcard',
       'Home_Team_Redcard', 'Away_Team_Redcard', 'Home_Team_Shots',
       'Away_Team_Shots', 'Home_Team_ShotsonTarget', 'Away_Team_ShotsonTarget',
       'Home_Team_Possession', 'Away_Team_Possession', 'Home_Team_Passes',
       'Away_Team_Passes', 'Home_Team_Fouls', 'Away_Team_Fouls',
       'Home_Team_Offside', 'Away_Team_Offside', 'Home_Team_Corner',
       'Away_Team_Corner', 'League', 'Result', 'Total_Fouls'],
      dtype='object')
```

[54]:
```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce', dayfirst=True)
print(df.dtypes)
print(df['Date'].head())

df['Total_Goals'] = df['Home_Team_Score'] + df['Away_Team_Score']

sns.histplot(df['Total_Goals'], bins=20, kde=True)
plt.title("Distribution of Goals Per Match")
plt.xlabel("Total Goals")
plt.ylabel("Number of Matches")
plt.show()

plt.figure(figsize=(6,4))
sns.boxplot(data=df[["Home_Team_Score", "Away_Team_Score"]])
plt.title("Comparison of Home and Away Goals")
plt.ylabel("Goals Scored")
plt.show()
```

```
ID                                int64
Date                     datetime64[ns]
Home_Team                        object
Away_Team                        object
Home_Team_Score                   int64
```

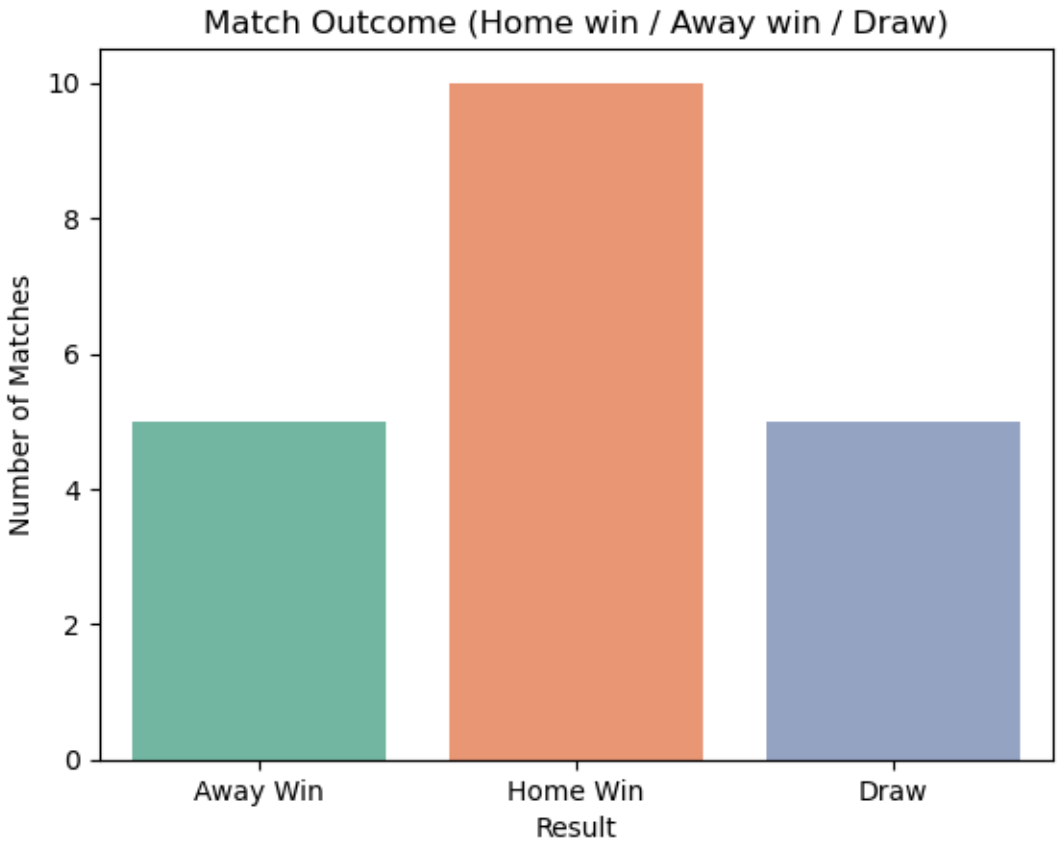
```
Away_Team_Score                    int64
Home_Team_Yellowcard               int64
Away_Team_Yellowcard               int64
Home_Team_Redcard                  int64
Away_Team_Redcard                  int64
Home_Team_Shots                    int64
Away_Team_Shots                    int64
Home_Team_ShotsonTarget            int64
Away_Team_ShotsonTarget            int64
Home_Team_Possession               int64
Away_Team_Possession               int64
Home_Team_Passes                   int64
Away_Team_Passes                   int64
Home_Team_Fouls                    int64
Away_Team_Fouls                    int64
Home_Team_Offside                  int64
Away_Team_Offside                  int64
Home_Team_Corner                   int64
Away_Team_Corner                   int64
League                            object
Result                            object
Total_Fouls                        int64
dtype: object
0   2025-08-15
1   2025-08-16
2   2025-08-16
3   2025-08-16
4   2025-08-16
Name: Date, dtype: datetime64[ns]
```


Distribution of Goals Per Match
Number of Matches
Total Goals
0
1
2
3
4
5
6


Comparison of Home and Away Goals
Goals Scored
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Home_Team_Score
Away_Team_Score

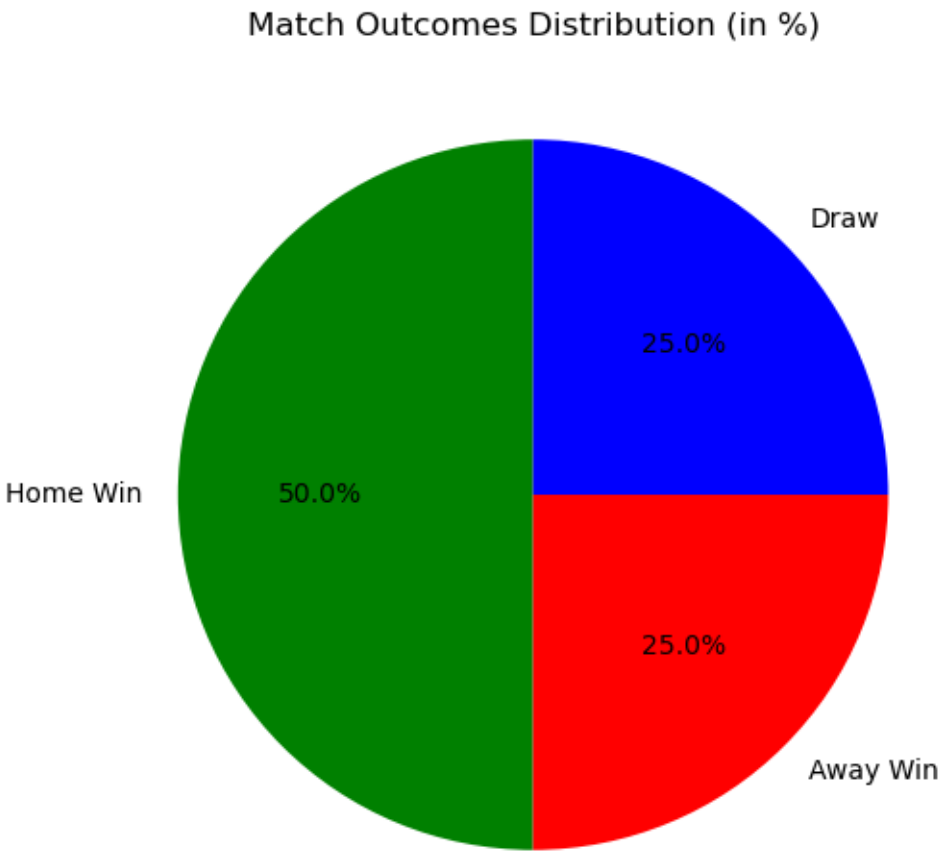
[78]:
```
def get_result(row):
    if row['Home_Team_Score'] > row['Away_Team_Score']:
        return "Home Win"
    elif row['Home_Team_Score'] < row['Away_Team_Score']:
        return "Away Win"
    else:
        return "Draw"

df['Result'] = df.apply(get_result, axis=1)
print(df['Result'].value_counts())

sns.countplot(x='Result', data=df, palette='Set2')
plt.title("Match Outcome (Home win / Away win / Draw)")
plt.xlabel("Result")
plt.ylabel("Number of Matches")
plt.show()
```

```
Result
Home Win    10
Away Win     5
Draw         5
Name: count, dtype: int64
```



[79]:
```
result_count = df['Result'].value_counts()
result_percentages = (result_count / len(df)) * 100
print("Percentages of Matches Outcomes\n", result_percentages)

plt.figure(figsize=(6,6))
plt.pie(result_count, labels=result_count.index, autopct='%.1f%%',
        startangle=90, colors=['green', 'red', 'blue'])
plt.title("Match Outcomes Distribution (in %)")
plt.show()
```

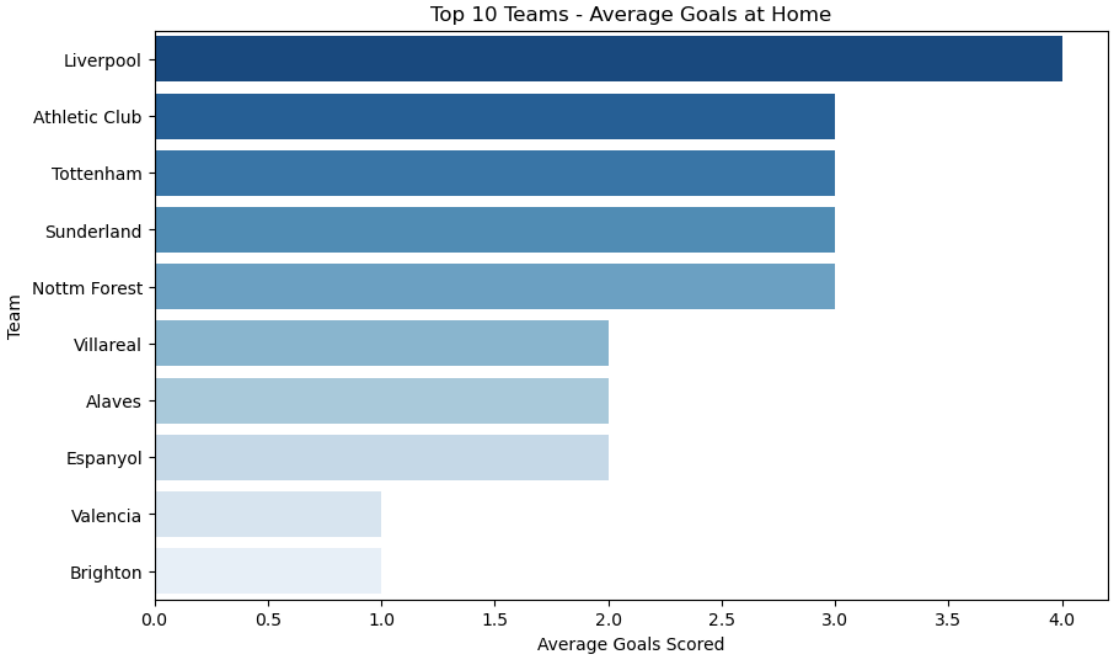
```
Percentages of Matches Outcomes
 Result
Home Win    50.0
Away Win    25.0
Draw        25.0
Name: count, dtype: float64
```



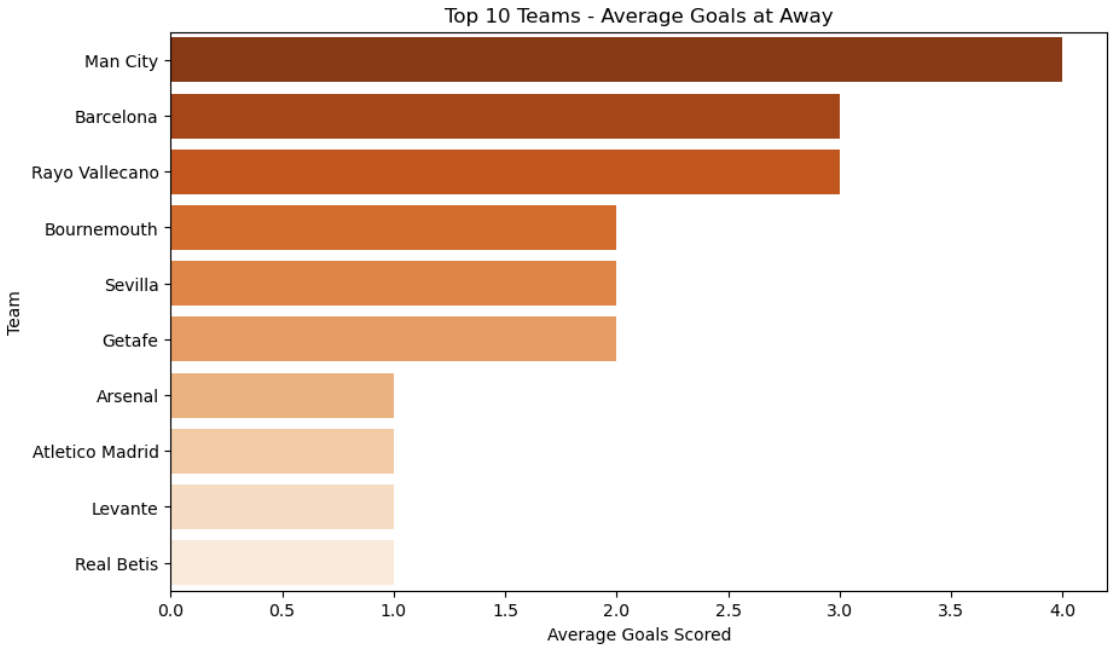
[80]:
```
home_goals = df.groupby("Home_Team")["Home_Team_Score"].mean().
 ↪sort_values(ascending=False)
away_goals = df.groupby("Away_Team")["Away_Team_Score"].mean().
 ↪sort_values(ascending=False)

print("Top 5 Teams - Average Goals at Home:\n", home_goals.head())
print("Top 5 Teams - Average Goals at Away:\n", away_goals.head())

plt.figure(figsize=(10,6))
sns.barplot(x=home_goals.head(10).values, y=home_goals.head(10).index,␣
 ↪palette="Blues_r")
plt.title("Top 10 Teams - Average Goals at Home")
plt.xlabel("Average Goals Scored")
plt.ylabel("Team")
plt.show()
```

```
Top 5 Teams - Average Goals at Home:
```

```
 Home_Team
Liverpool        4.0
Athletic Club    3.0
Tottenham        3.0
Sunderland       3.0
Nottm Forest     3.0
Name: Home_Team_Score, dtype: float64
Top 5 Teams - Average Goals at Away:
 Away_Team
Man City          4.0
Barcelona         3.0
Rayo Vallecano    3.0
Bournemouth       2.0
Sevilla           2.0
Name: Away_Team_Score, dtype: float64
```



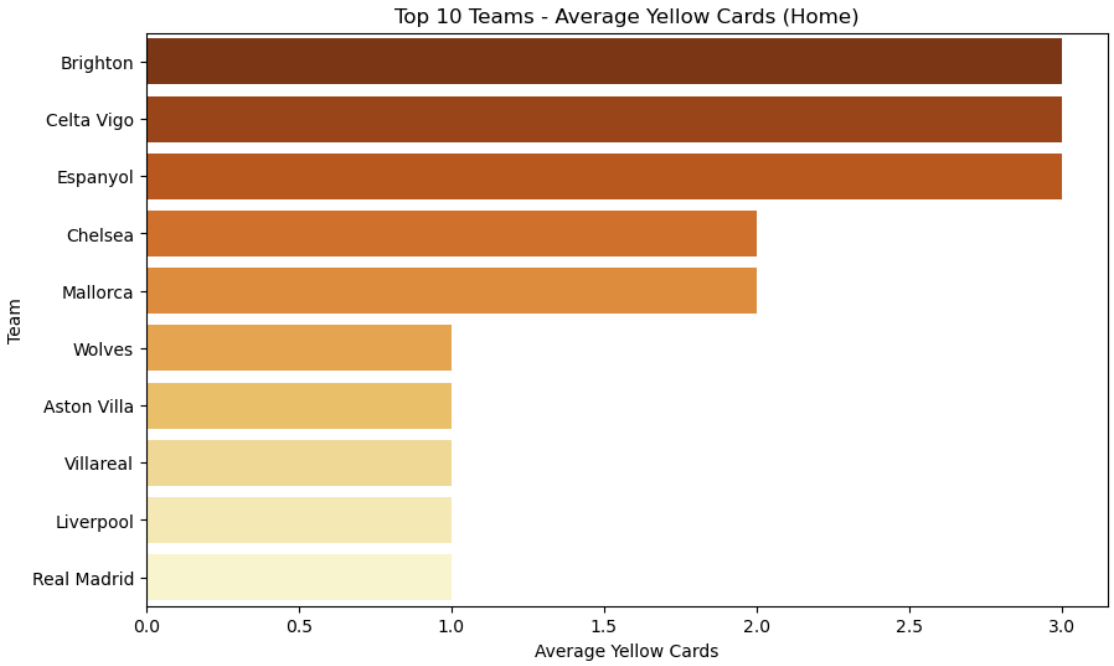
[81]:
```
plt.figure(figsize=(10,6))
sns.barplot(x=away_goals.head(10).values, y=away_goals.head(10).index,␣
 ↪palette="Oranges_r")
plt.title("Top 10 Teams - Average Goals at Away")
plt.xlabel("Average Goals Scored")
plt.ylabel("Team")
plt.show()
```



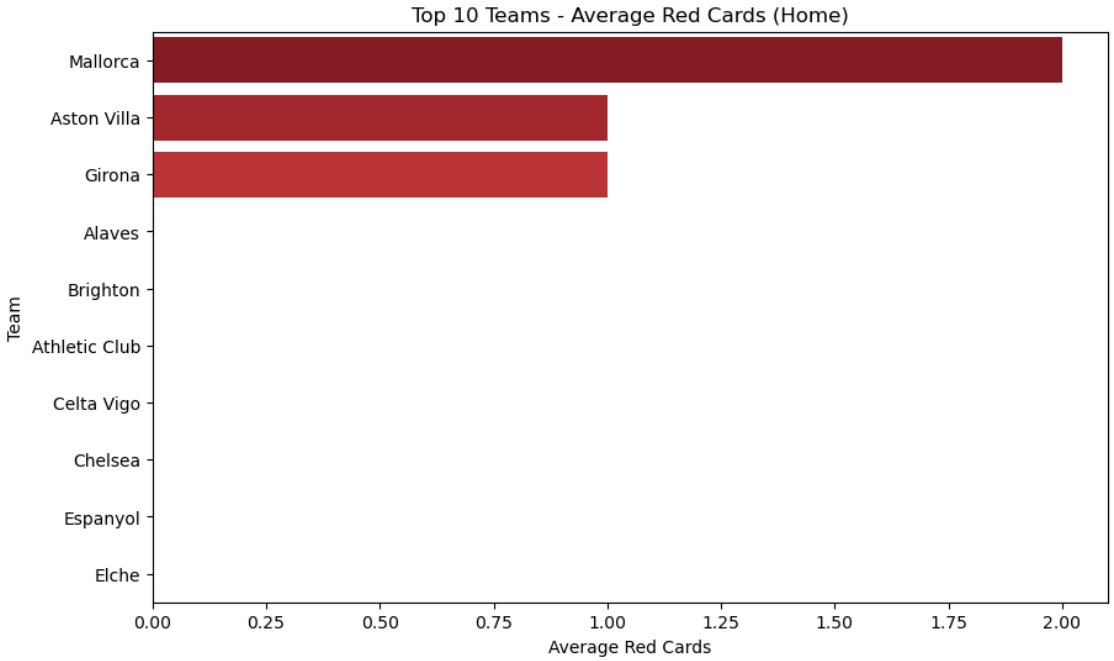
[82]:
```
avg_yellow_cards = df.groupby("Home_Team")[['Home_Team_Yellowcard']].mean().
 ↪sort_values(
    by="Home_Team_Yellowcard", ascending=False)
avg_red_cards = df.groupby("Home_Team")[['Home_Team_Redcard']].mean().
 ↪sort_values(
    by="Home_Team_Redcard", ascending=False)

print("Top 5 Teams - Average Yellow Cards (Home):\n", avg_yellow_cards.head())
print("Top 5 Teams - Average Red Cards (Home):\n", avg_red_cards.head())

plt.figure(figsize=(10,6))
sns.barplot(
    x=avg_yellow_cards.head(10).values.flatten(),
    y=avg_yellow_cards.head(10).index,
    palette="YlOrBr_r"
)
plt.title("Top 10 Teams - Average Yellow Cards (Home)")
plt.xlabel("Average Yellow Cards")
plt.ylabel("Team")
plt.show()
```

```
Top 5 Teams - Average Yellow Cards (Home):
             Home_Team_Yellowcard
Home_Team
Brighton                      3.0
Celta Vigo                    3.0
```

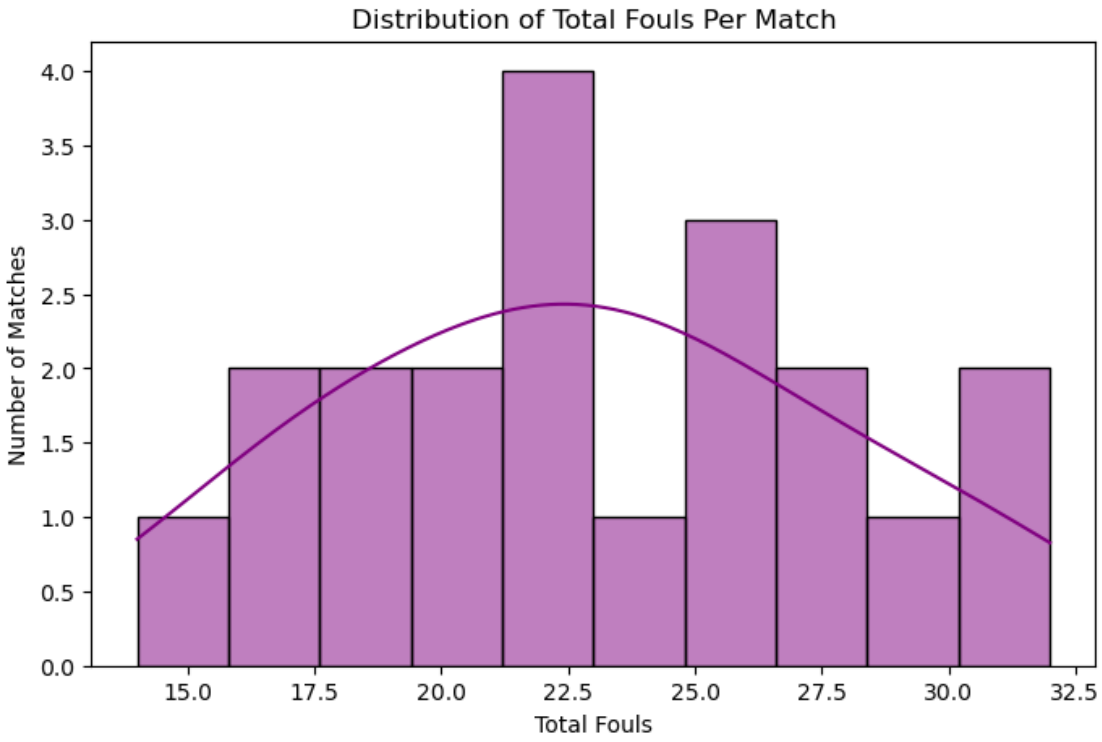
```
Espanyol                    3.0
Chelsea                     2.0
Mallorca                    2.0
Top 5 Teams - Average Red Cards (Home):
             Home_Team_Redcard
Home_Team
Mallorca                   2.0
Aston Villa                1.0
Girona                     1.0
Alaves                     0.0
Brighton                   0.0
```



[83]:
```
plt.figure(figsize=(10,6))
sns.barplot(
    x=avg_red_cards.head(10).values.flatten(),
    y=avg_red_cards.head(10).index,
    palette="Reds_r"
)
plt.title("Top 10 Teams - Average Red Cards (Home)")
plt.xlabel("Average Red Cards")
plt.ylabel("Team")
plt.show()
```



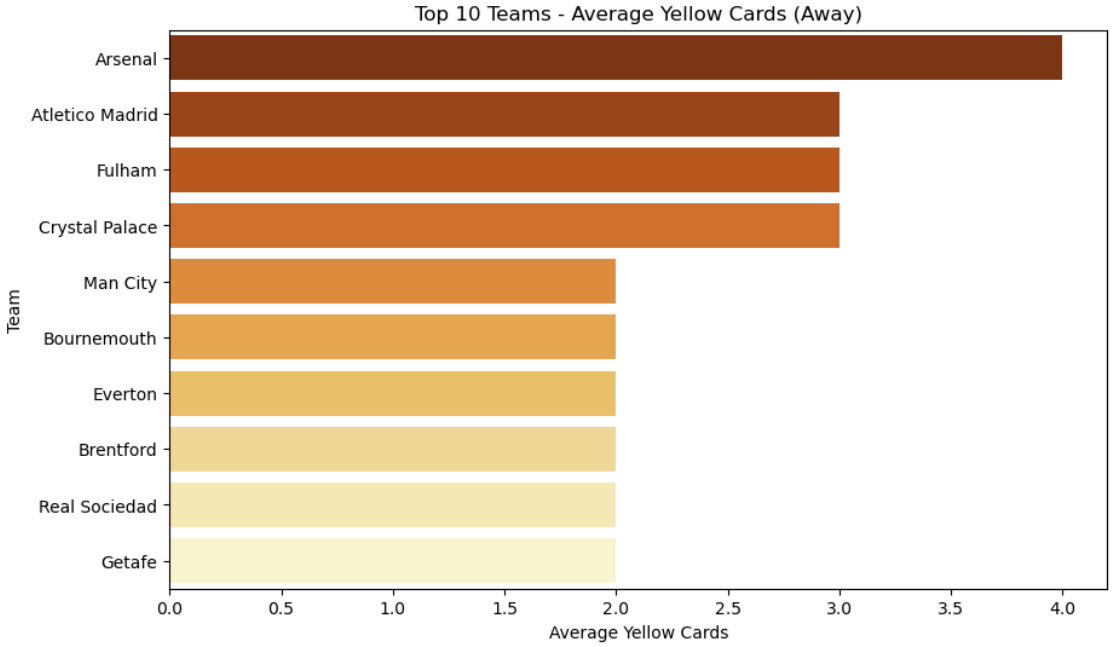
[84]:
```
df["Total_Fouls"] = df["Home_Team_Fouls"] + df["Away_Team_Fouls"]

plt.figure(figsize=(8,5))
sns.histplot(df["Total_Fouls"], bins=10, kde=True, color="purple")
plt.title("Distribution of Total Fouls Per Match")
plt.xlabel("Total Fouls")
plt.ylabel("Number of Matches")
plt.show()
```

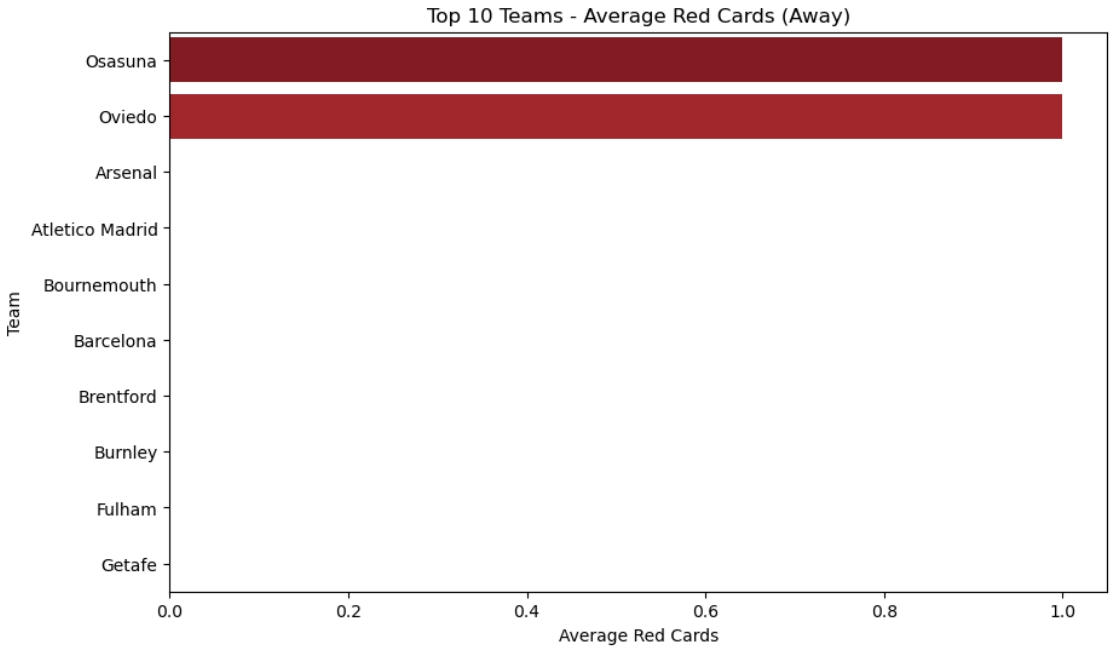


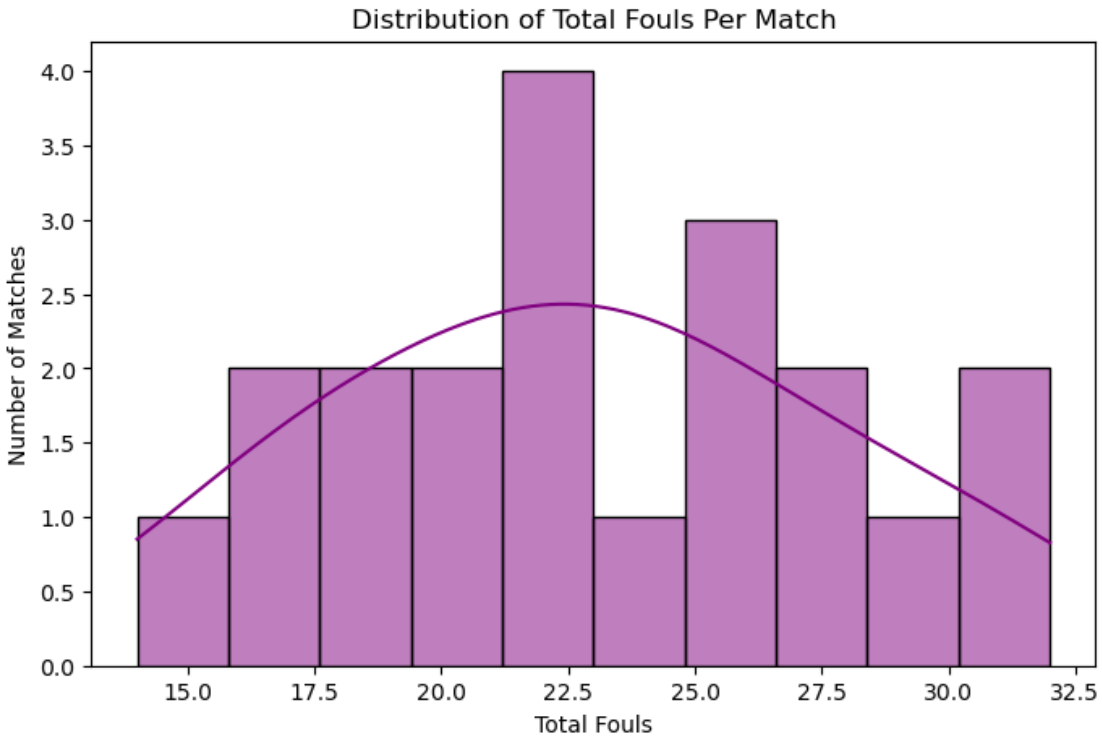
[55]:
```
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)

avg_yellow_cards = df.groupby("Away_Team")[['Away_Team_Yellowcard']].mean().
 ↪sort_values(
    by="Away_Team_Yellowcard", ascending=False)
avg_red_cards = df.groupby("Away_Team")[['Away_Team_Redcard']].mean().
 ↪sort_values(
    by="Away_Team_Redcard", ascending=False)

print("Top 5 Teams - Average Yellow Cards (Away):\n", avg_yellow_cards.head())
print("Top 5 Teams - Average Red Cards (Away):\n", avg_red_cards.head())

plt.figure(figsize=(10,6))
sns.barplot(
    x=avg_yellow_cards.head(10).values.flatten(),
    y=avg_yellow_cards.head(10).index,
    palette="YlOrBr_r"
)
plt.title("Top 10 Teams - Average Yellow Cards (Away)")
plt.xlabel("Average Yellow Cards")
plt.ylabel("Team")
plt.show()
```

```
plt.figure(figsize=(10,6))
sns.barplot(
    x=avg_red_cards.head(10).values.flatten(),
    y=avg_red_cards.head(10).index,
    palette="Reds_r"
)
plt.title("Top 10 Teams - Average Red Cards (Away)")
plt.xlabel("Average Red Cards")
plt.ylabel("Team")
plt.show()

df["Total_Fouls"] = df["Home_Team_Fouls"] + df["Away_Team_Fouls"]

plt.figure(figsize=(8,5))
sns.histplot(df["Total_Fouls"], bins=10, kde=True, color="purple")
plt.title("Distribution of Total Fouls Per Match")
plt.xlabel("Total Fouls")
plt.ylabel("Number of Matches")
plt.show()
```

```
Top 5 Teams - Average Yellow Cards (Away):
                Away_Team_Yellowcard
Away_Team
Arsenal                          4.0
Atletico Madrid                  3.0
Fulham                           3.0
Crystal Palace                   3.0
Man City                         2.0
Top 5 Teams - Average Red Cards (Away):
                Away_Team_Redcard
Away_Team
Osasuna                       1.0
Oviedo                        1.0
Arsenal                       0.0
Atletico Madrid               0.0
Bournemouth                   0.0
```


Top 10 Teams - Average Yellow Cards (Away)
Arsenal
Atletico Madrid
Fulham
Crystal Palace
Man City
Bournemouth
Everton
Brentford
Real Sociedad
Getafe
Team
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Average Yellow Cards

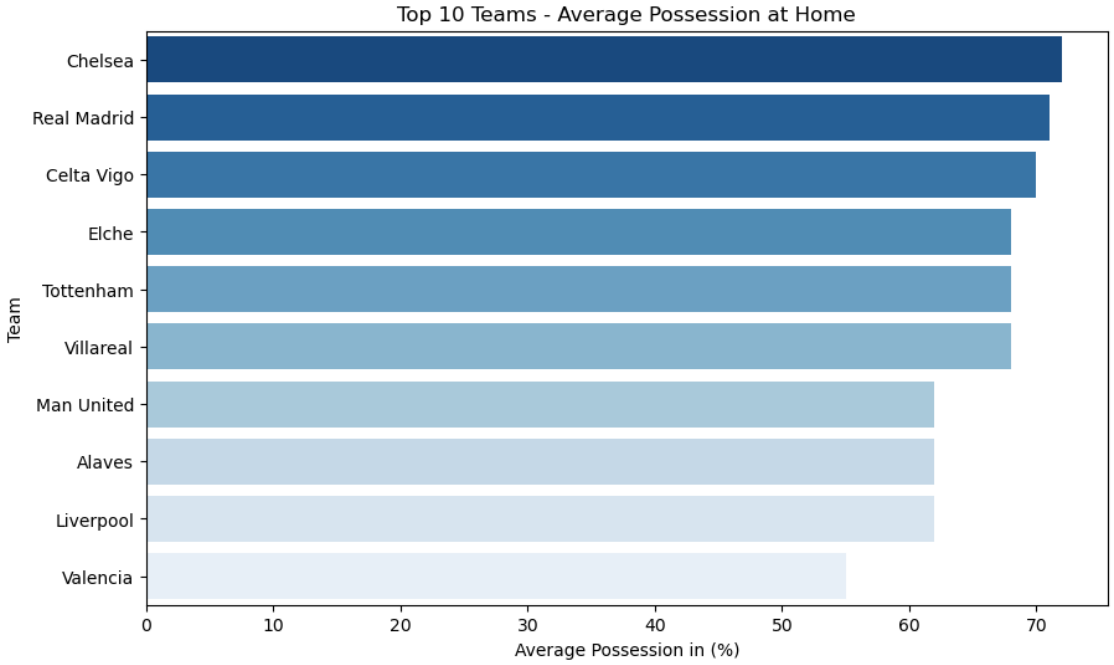

Top 10 Teams - Average Red Cards (Away)
Osasuna
Oviedo
Arsenal
Atletico Madrid
Bournemouth
Barcelona
Brentford
Burnley
Fulham
Getafe
Team
0.0
0.2
0.4
0.6
0.8
1.0
Average Red Cards



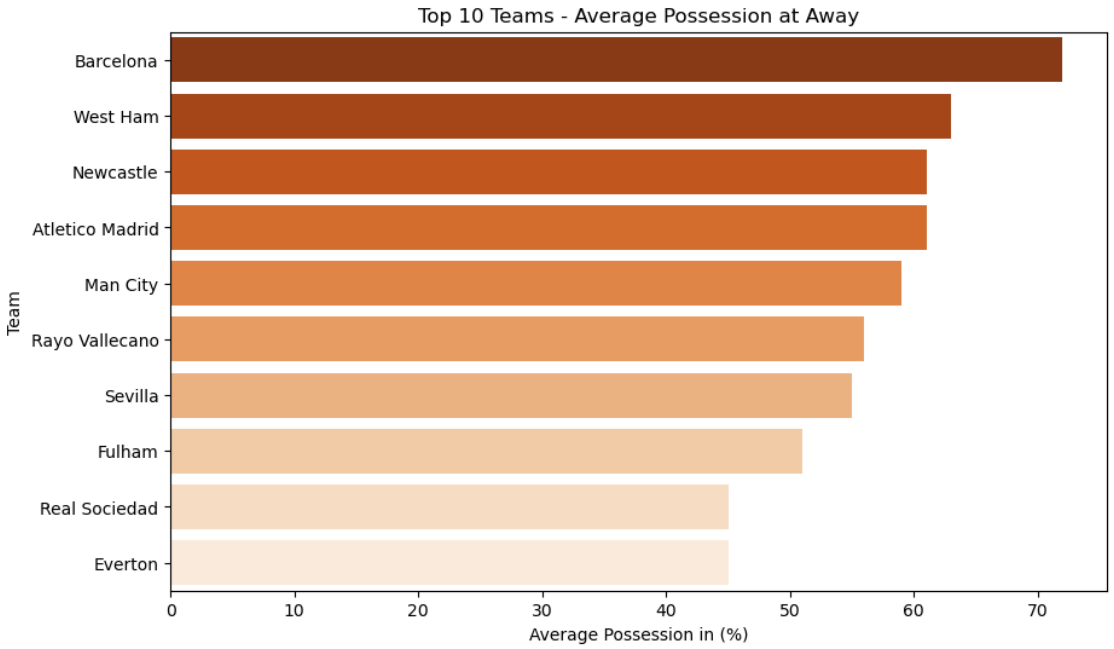
[56]:
```
home_possession = df.groupby("Home_Team")["Home_Team_Possession"].mean().
  ↪sort_values(ascending = False)
away_possession = df.groupby("Away_Team")["Away_Team_Possession"].mean().
  ↪sort_values(ascending = False)

print("Top 5 Teams - Average Possession at Home: \n", home_possession.head())
print("Top 5 Teams - Average Possession at Away: \n", away_possession.head())

plt.figure(figsize=(10,6))
sns.barplot(x=home_possession.head(10).values, y=home_possession.head(10).
  ↪index, palette="Blues_r")
plt.title("Top 10 Teams - Average Possession at Home")
plt.xlabel("Average Possession in (%)")
plt.ylabel("Team")
plt.show()
```

```
Top 5 Teams - Average Possession at Home:
 Home_Team
Chelsea        72.0
Real Madrid    71.0
Celta Vigo     70.0
Elche          68.0
Tottenham      68.0
```

```
Name: Home_Team_Possession, dtype: float64
Top 5 Teams - Average Possession at Away:
 Away_Team
Barcelona          72.0
West Ham           63.0
Newcastle          61.0
Atletico Madrid    61.0
Man City           59.0
Name: Away_Team_Possession, dtype: float64
```



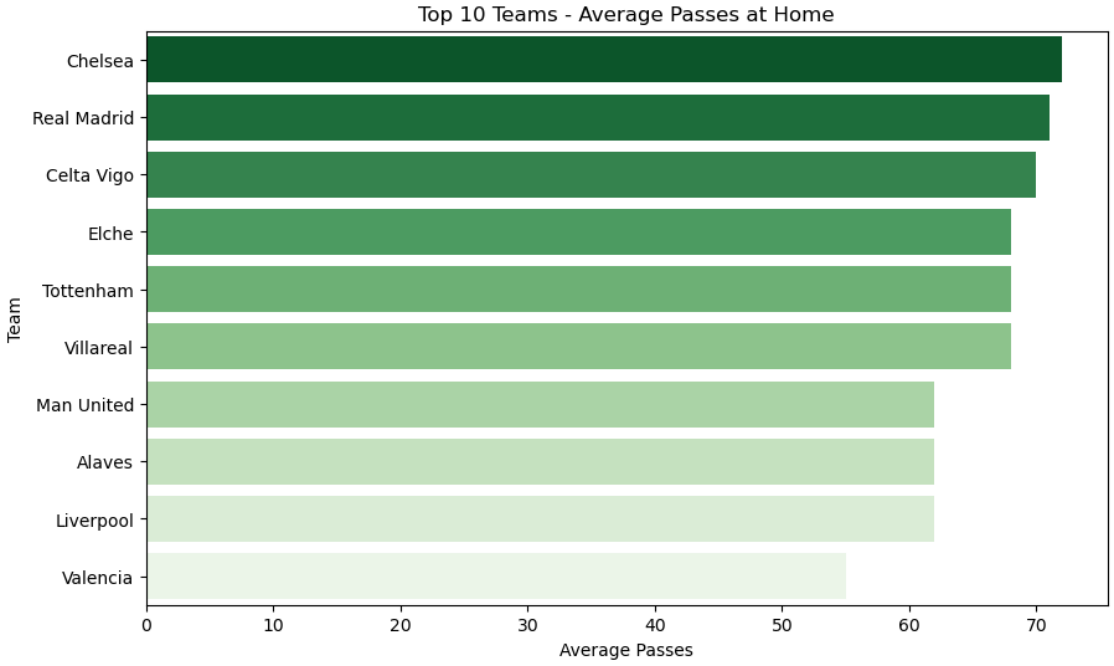
[57]:
```
plt.figure(figsize=(10,6))
sns.barplot(x=away_possession.head(10).values, y=away_possession.head(10).
  ↪index, palette="Oranges_r")
plt.title("Top 10 Teams - Average Possession at Away")
plt.xlabel("Average Possession in (%)")
plt.ylabel("Team")
plt.show()
```



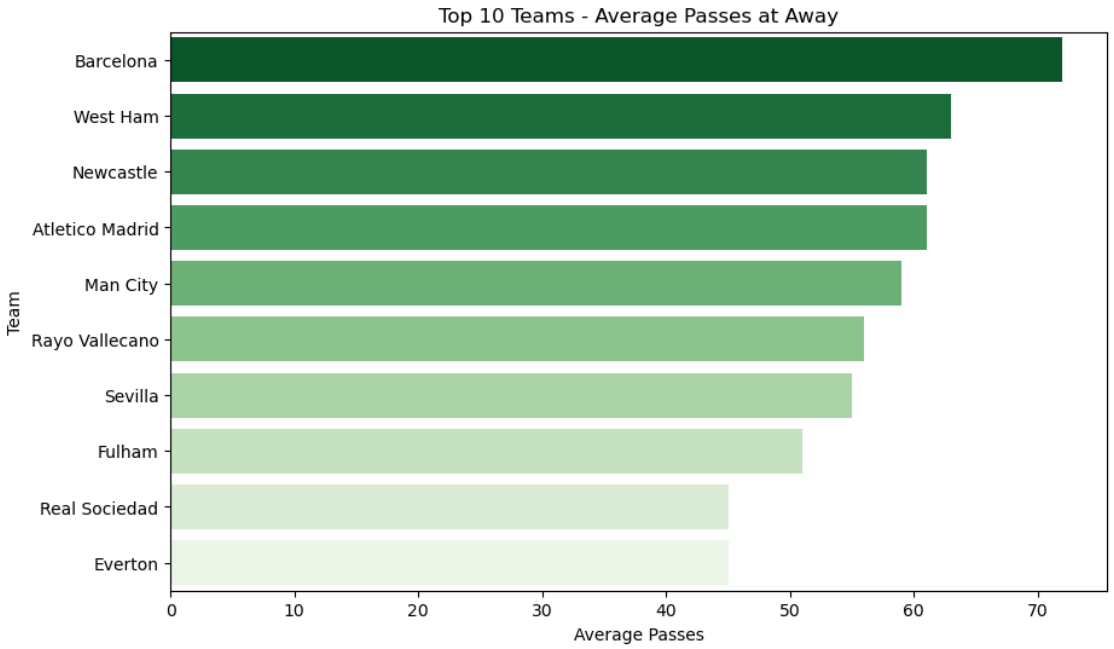
[58]:
```
home_passes = df.groupby("Home_Team")["Home_Team_Passes"].mean().
  ↪sort_values(ascending = False)
away_passes = df.groupby("Away_Team")["Away_Team_Passes"].mean().
  ↪sort_values(ascending = False)

print("Top 5 Teams - Average Passes at Home: \n", home_possession.head())
print("Top 5 Teams - Average Passes at Away: \n", away_possession.head())

plt.figure(figsize=(10,6))
sns.barplot(x=home_possession.head(10).values, y=home_possession.head(10).
  ↪index, palette="Greens_r")
plt.title("Top 10 Teams - Average Passes at Home")
plt.xlabel("Average Passes")
plt.ylabel("Team")
plt.show()
```

```
Top 5 Teams - Average Passes at Home:
 Home_Team
Chelsea        72.0
Real Madrid    71.0
Celta Vigo     70.0
Elche          68.0
Tottenham      68.0
Name: Home_Team_Possession, dtype: float64
Top 5 Teams - Average Passes at Away:
 Away_Team
```

```
Barcelona          72.0
West Ham           63.0
Newcastle          61.0
Atletico Madrid    61.0
Man City           59.0
Name: Away_Team_Possession, dtype: float64
```

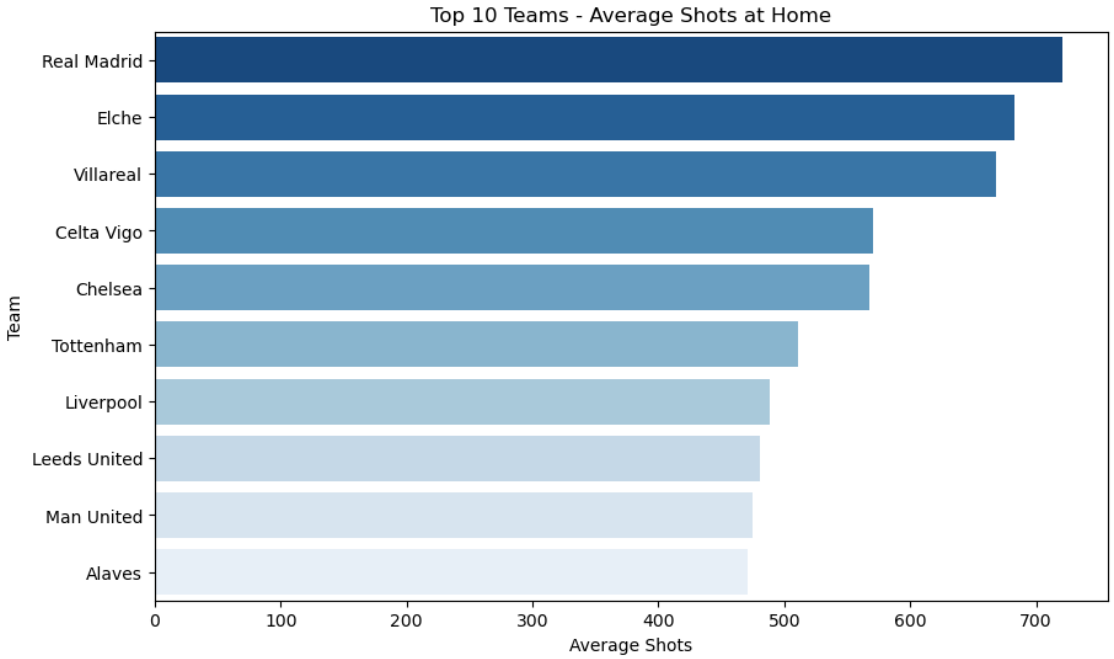


[59]:
```
plt.figure(figsize=(10,6))
sns.barplot(x=away_possession.head(10).values, y=away_possession.head(10).
  ↪index, palette="Greens_r")
plt.title("Top 10 Teams - Average Passes at Away")
plt.xlabel("Average Passes")
plt.ylabel("Team")
plt.show()
```

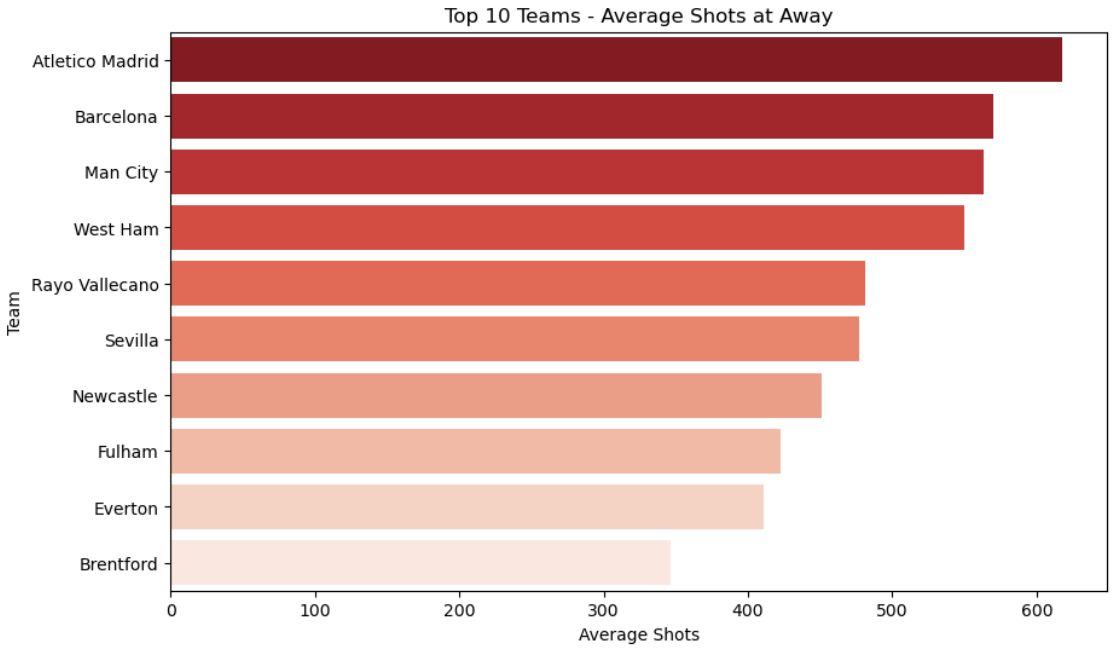


[60]:
```
home_shots = df.groupby("Home_Team")["Home_Team_Passes"].mean().
 ↪sort_values(ascending = False)
away_shots = df.groupby("Away_Team")["Away_Team_Passes"].mean().
 ↪sort_values(ascending = False)


home_shots_on_target = df.groupby("Home_Team")["Home_Team_Passes"].mean().
 ↪sort_values(ascending = False)
away_shots_on_target = df.groupby("Away_Team")["Away_Team_Passes"].mean().
 ↪sort_values(ascending = False)

print("Top 5 Teams - Average Shots at Home: \n", home_shots.head())
print("Top 5 Teams - Average Shots at Away: \n", away_shots.head())
print("Top 5 Teams - Average Shots On Target at Home: \n", home_shots_on_target.
 ↪head())
print("Top 5 Teams - Average Shots On Target at Away: \n", away_shots_on_target.
 ↪head())

plt.figure(figsize=(10,6))
sns.barplot(x=home_shots.head(10).values, y=home_shots.head(10).index,␣
 ↪palette="Blues_r")
plt.title("Top 10 Teams - Average Shots at Home")
plt.xlabel("Average Shots")
plt.ylabel("Team")
plt.show()
```

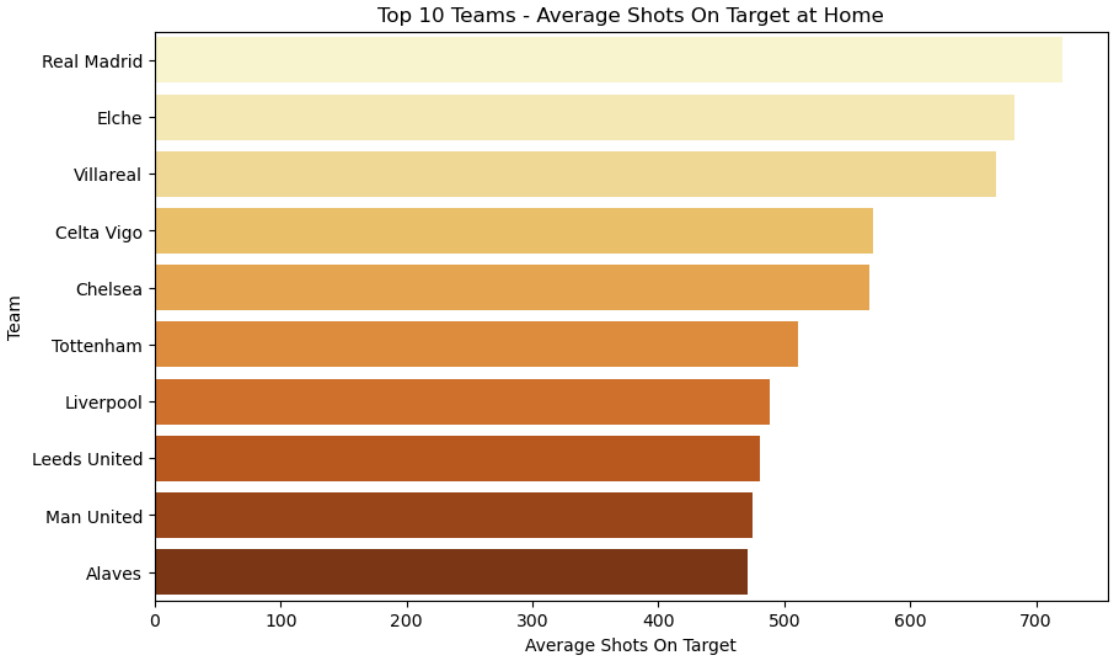
```
Top 5 Teams - Average Shots at Home:
 Home_Team
Real Madrid    721.0
Elche          683.0
Villareal      668.0
Celta Vigo     571.0
Chelsea        568.0
Name: Home_Team_Passes, dtype: float64
Top 5 Teams - Average Shots at Away:
 Away_Team
Atletico Madrid    618.0
Barcelona          570.0
Man City           563.0
West Ham           550.0
Rayo Vallecano     481.0
Name: Away_Team_Passes, dtype: float64
Top 5 Teams - Average Shots On Target at Home:
 Home_Team
Real Madrid    721.0
Elche          683.0
Villareal      668.0
Celta Vigo     571.0
Chelsea        568.0
Name: Home_Team_Passes, dtype: float64
Top 5 Teams - Average Shots On Target at Away:
 Away_Team
Atletico Madrid    618.0
Barcelona          570.0
Man City           563.0
West Ham           550.0
Rayo Vallecano     481.0
Name: Away_Team_Passes, dtype: float64
```



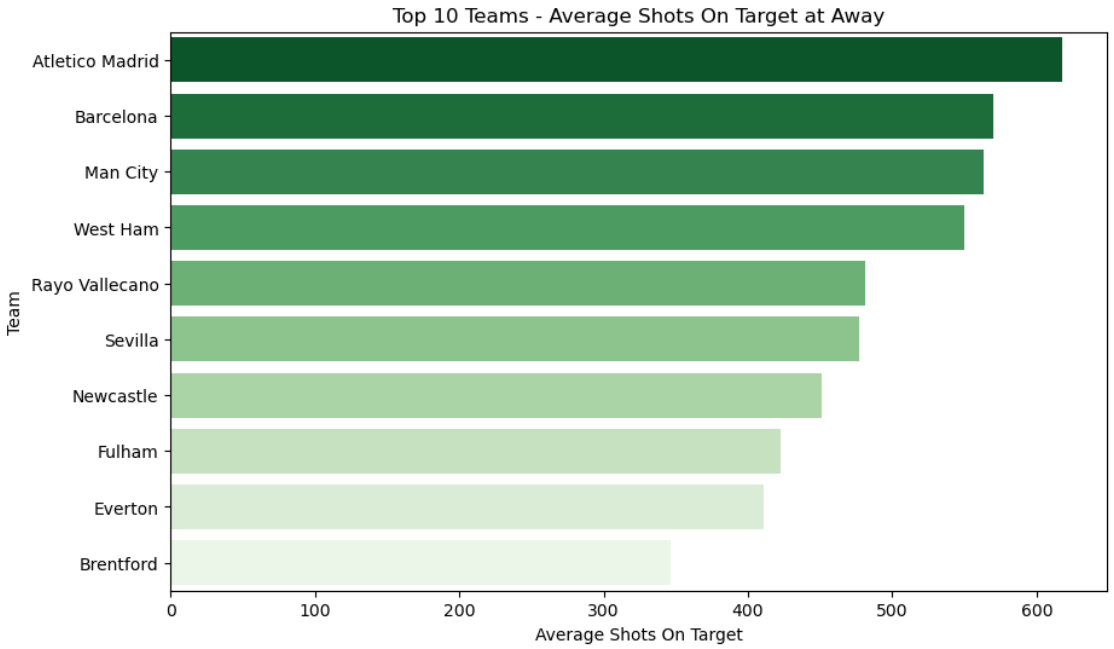
[61]:
```
plt.figure(figsize=(10,6))
sns.barplot(x=away_shots.head(10).values, y=away_shots.head(10).index,␣
 ↪palette="Reds_r")
plt.title("Top 10 Teams - Average Shots at Away")
plt.xlabel("Average Shots")
plt.ylabel("Team")
plt.show()
```



[62]:
```
plt.figure(figsize=(10,6))
sns.barplot(x=home_shots_on_target.head(10).values, y=home_shots_on_target.
 ↪head(10).index, palette="YlOrBr")
plt.title("Top 10 Teams - Average Shots On Target at Home")
plt.xlabel("Average Shots On Target")
plt.ylabel("Team")
plt.show()
```



[63]:
```
plt.figure(figsize=(10,6))
sns.barplot(x=away_shots_on_target.head(10).values, y=away_shots_on_target.
 ↪head(10).index, palette="Greens_r")
plt.title("Top 10 Teams - Average Shots On Target at Away")
plt.xlabel("Average Shots On Target")
plt.ylabel("Team")
plt.show()
```



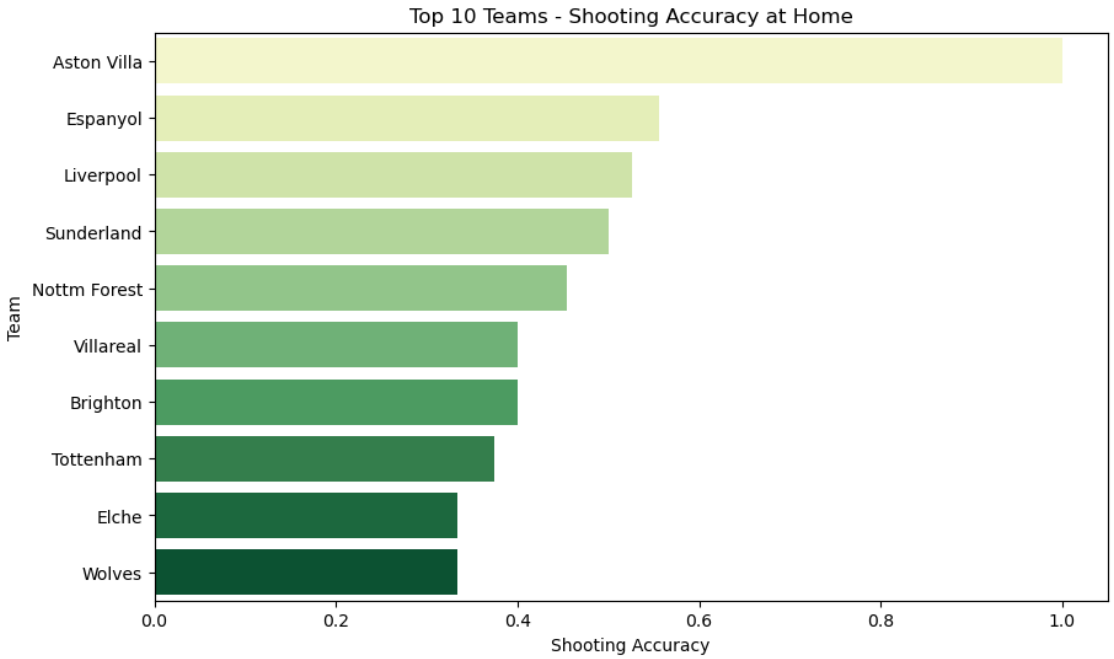
[64]:
```
df["Home_Shooting_Accuracy"] = df["Home_Team_ShotsonTarget"] /␣
  ↪df["Home_Team_Shots"]
df["Away_Shooting_Accuracy"] = df["Away_Team_ShotsonTarget"] /␣
  ↪df["Away_Team_Shots"]

print("Average Shooting Accuracy (Home):", df["Home_Shooting_Accuracy"].mean())
print("Average Shooting Accuracy (Away):", df["Away_Shooting_Accuracy"].mean())
```

```
Average Shooting Accuracy (Home): 0.35876499962026276
Average Shooting Accuracy (Away): 0.3151964701964702
```

[65]:
```
home_accuracy = df.groupby("Home_Team") ["Home_Shooting_Accuracy"].mean().
  ↪sort_values(ascending = False)
away_accuracy = df.groupby("Away_Team") ["Away_Shooting_Accuracy"].mean().
  ↪sort_values(ascending = False)

print("Top 5 Teams - Shooting Accuracy at Home:\n", home_accuracy.head())
print("Top 5 Teams - Shooting Accuracy at Away:\n", away_accuracy.head())

plt.figure(figsize=(10,6))
sns.barplot(x=home_accuracy.head(10).values, y=home_accuracy.head(10).index,␣
  ↪palette="YlGn")
plt.title("Top 10 Teams - Shooting Accuracy at Home")
plt.xlabel("Shooting Accuracy")
plt.ylabel("Team")
```

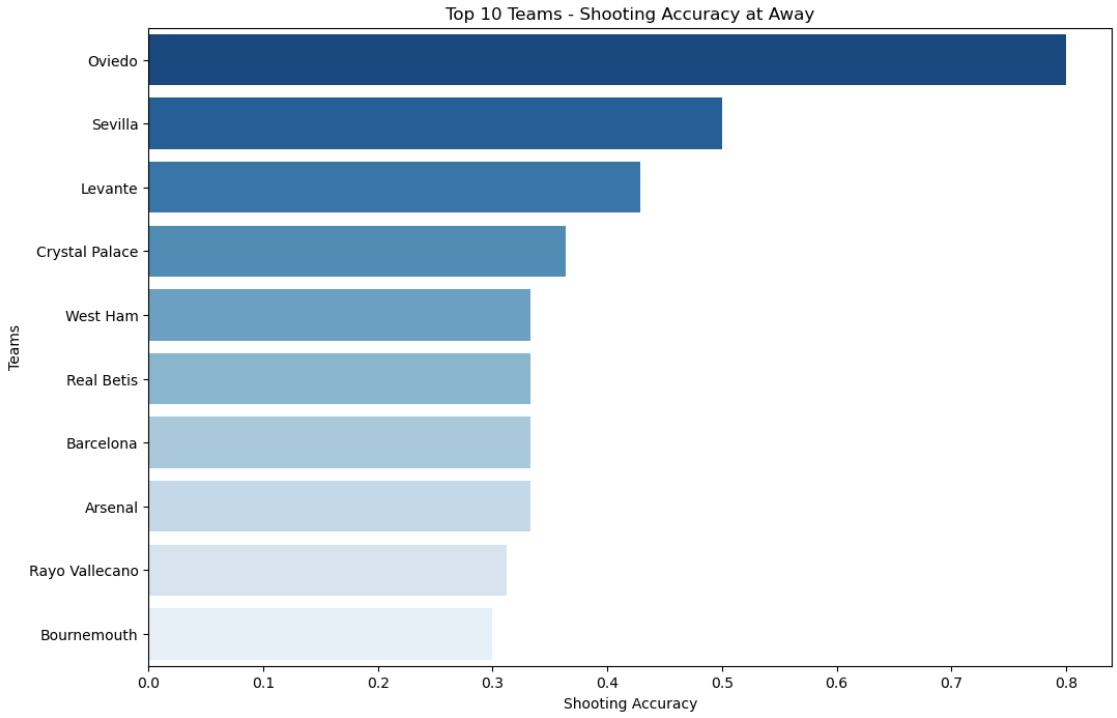
```
plt.show()
```

```
Top 5 Teams - Shooting Accuracy at Home:
 Home_Team
Aston Villa     1.000000
Espanyol        0.555556
Liverpool       0.526316
Sunderland      0.500000
Nottm Forest    0.454545
Name: Home_Shooting_Accuracy, dtype: float64
Top 5 Teams - Shooting Accuracy at Away:
 Away_Team
Oviedo            0.800000
Sevilla           0.500000
Levante           0.428571
Crystal Palace    0.363636
West Ham          0.333333
Name: Away_Shooting_Accuracy, dtype: float64
```

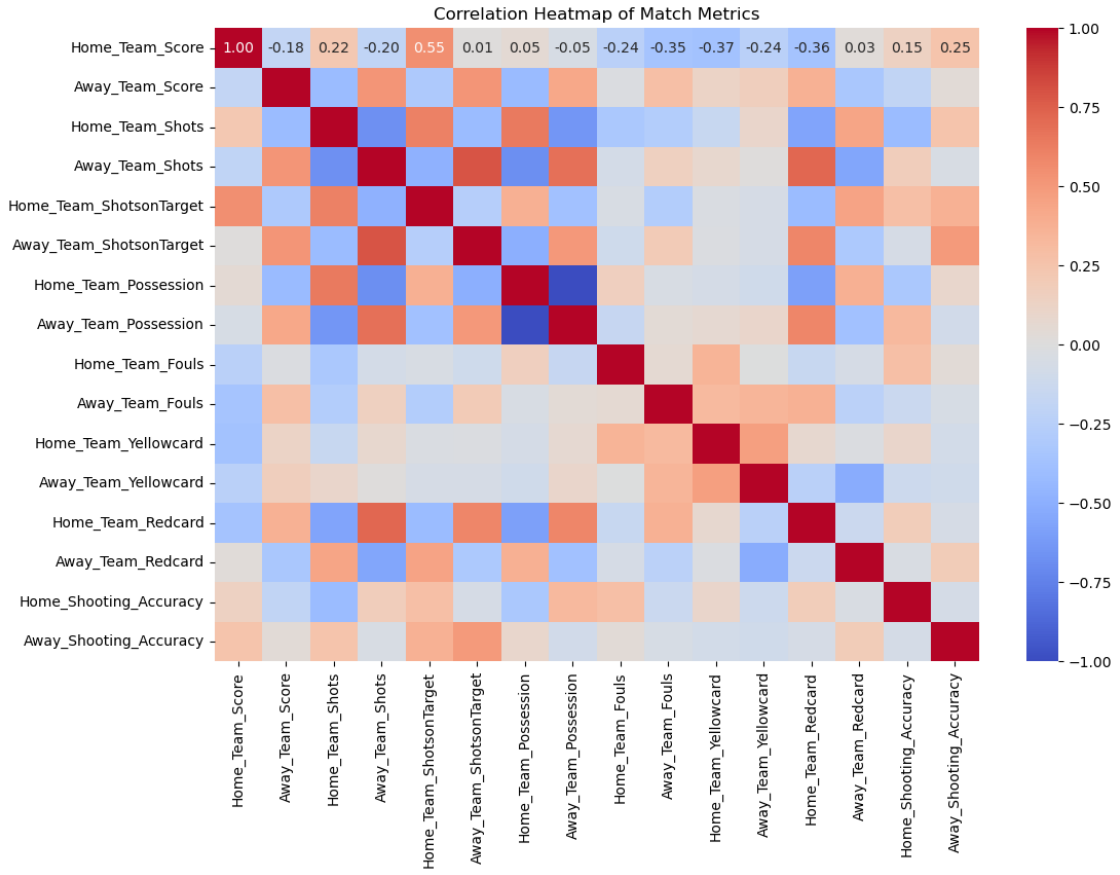


[66]:
```
plt.figure(figsize=(12,8))
sns.barplot(x=away_accuracy.head(10).values, y=away_accuracy.head(10).index,␣
 ↪palette="Blues_r")
plt.title("Top 10 Teams - Shooting Accuracy at Away")
plt.xlabel("Shooting Accuracy")
plt.ylabel("Teams")
```

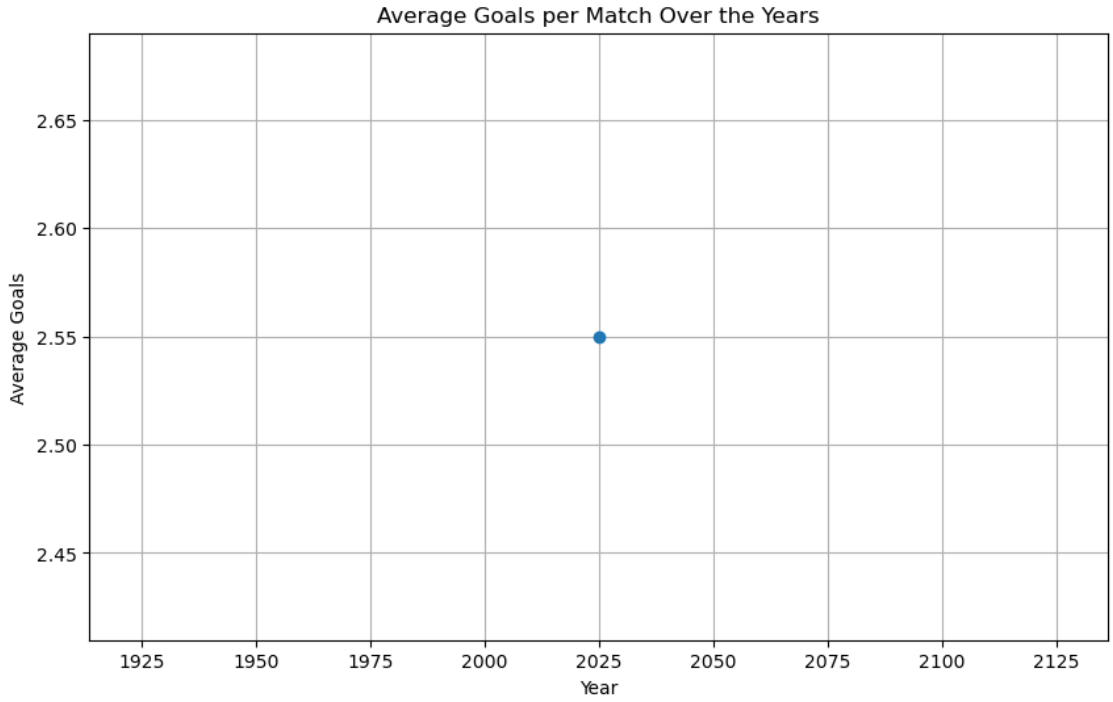
```
plt.show()
```



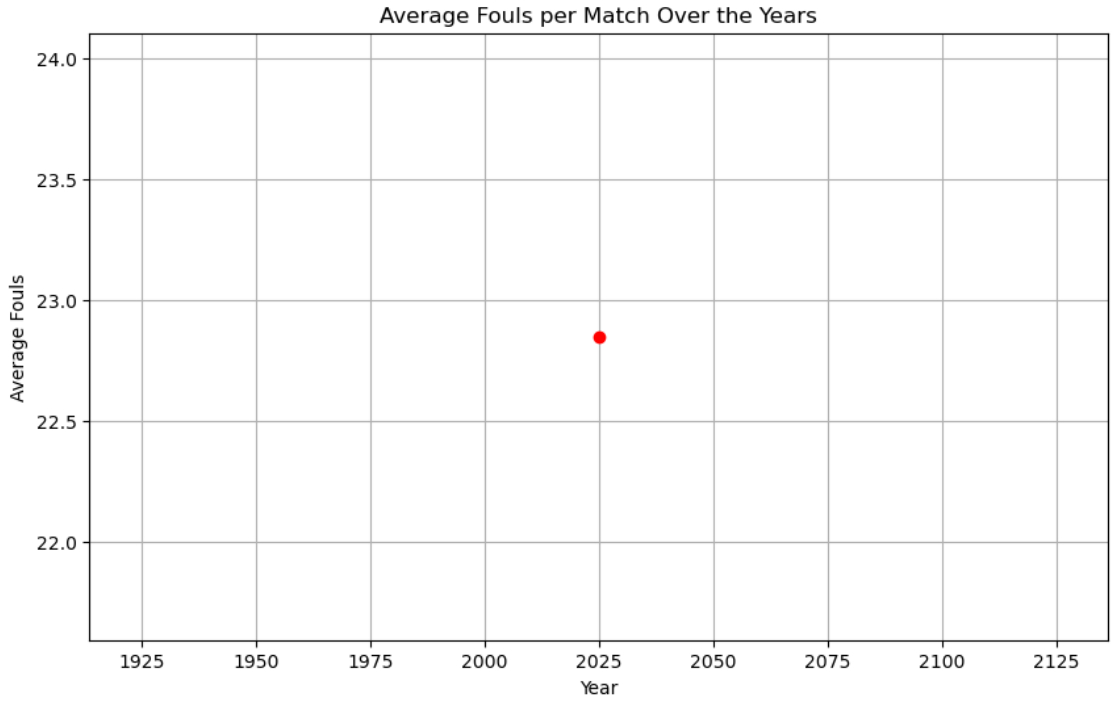
[67]:
```
metrics = [
    "Home_Team_Score", "Away_Team_Score",
    "Home_Team_Shots", "Away_Team_Shots",
    "Home_Team_ShotsonTarget", "Away_Team_ShotsonTarget",
    "Home_Team_Possession", "Away_Team_Possession",
    "Home_Team_Fouls", "Away_Team_Fouls",
    "Home_Team_Yellowcard", "Away_Team_Yellowcard",
    "Home_Team_Redcard", "Away_Team_Redcard",
    "Home_Shooting_Accuracy", "Away_Shooting_Accuracy"
]

corr = df[metrics].corr()

plt.figure(figsize=(12,8))
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f", cbar=True)
plt.title("Correlation Heatmap of Match Metrics")
plt.show()
```



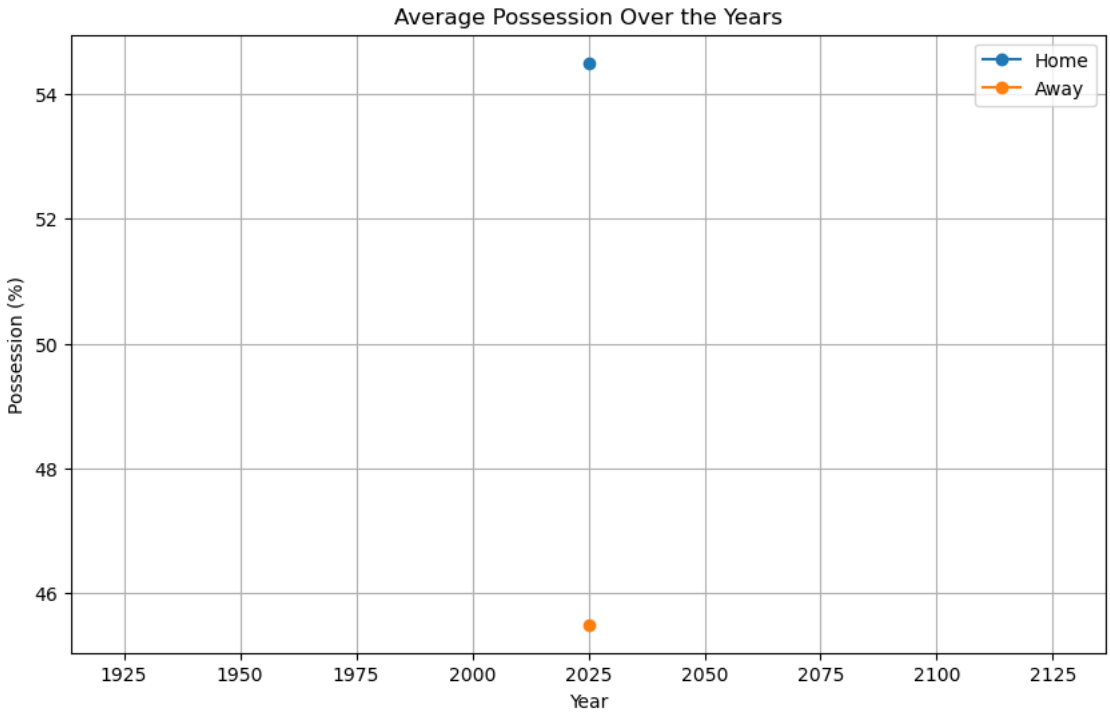
[68]:
```
df['Year'] = df['Date'].dt.year
goals_per_year = df.groupby('Year')['Total_Goals'].mean()

plt.figure(figsize=(10,6))
goals_per_year.plot(marker='o')
plt.title("Average Goals per Match Over the Years")
plt.xlabel("Year")
plt.ylabel("Average Goals")
plt.grid(True)
plt.show()
```



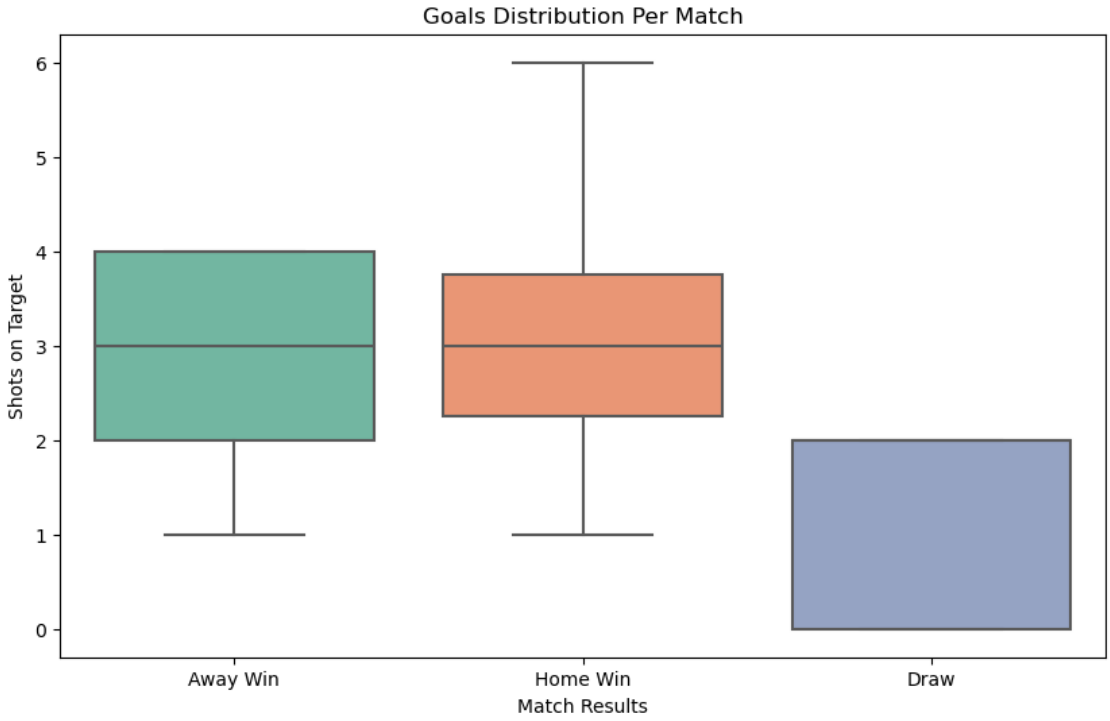
[69]:
```
fouls_per_year = df.groupby('Year')['Total_Fouls'].mean()
plt.figure(figsize=(10,6))
fouls_per_year.plot(marker='o', color="red")
plt.title("Average Fouls per Match Over the Years")
plt.xlabel("Year")
plt.ylabel("Average Fouls")
plt.grid(True)
plt.show()
```



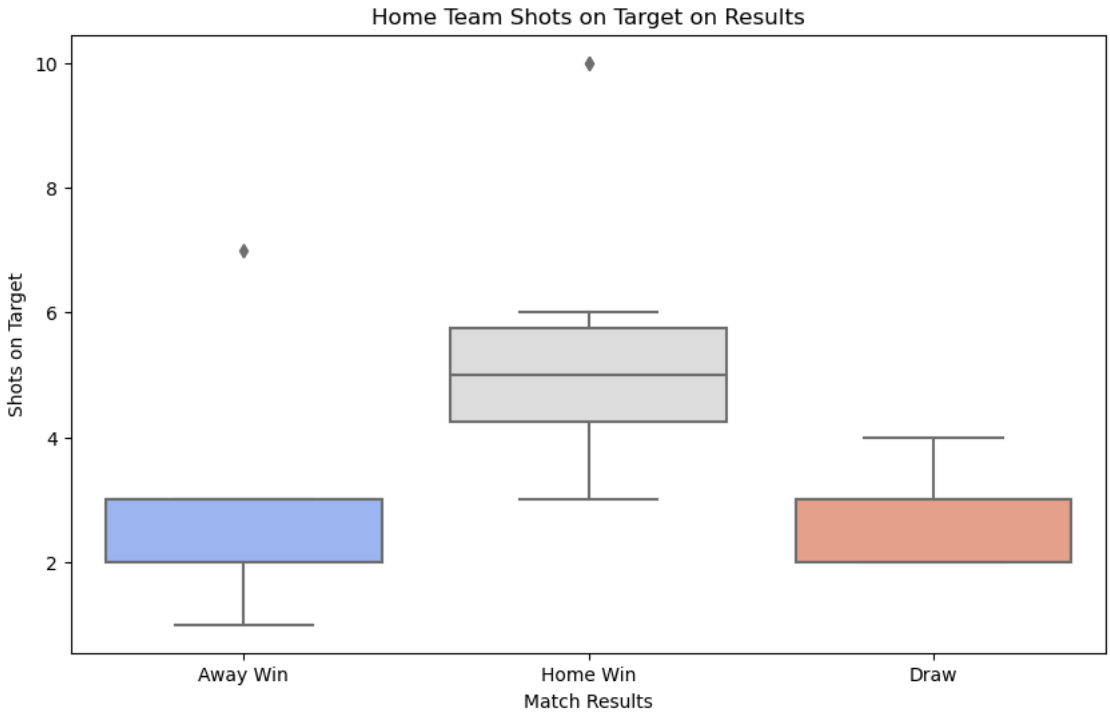
[70]:
```
possession_per_year = df.
  ↪groupby('Year')[['Home_Team_Possession','Away_Team_Possession']].mean()

possession_per_year.plot(figsize=(10,6), marker='o')
plt.title("Average Possession Over the Years")
plt.xlabel("Year")
plt.ylabel("Possession (%)")
plt.legend(["Home", "Away"])
plt.grid(True)
plt.show()
```



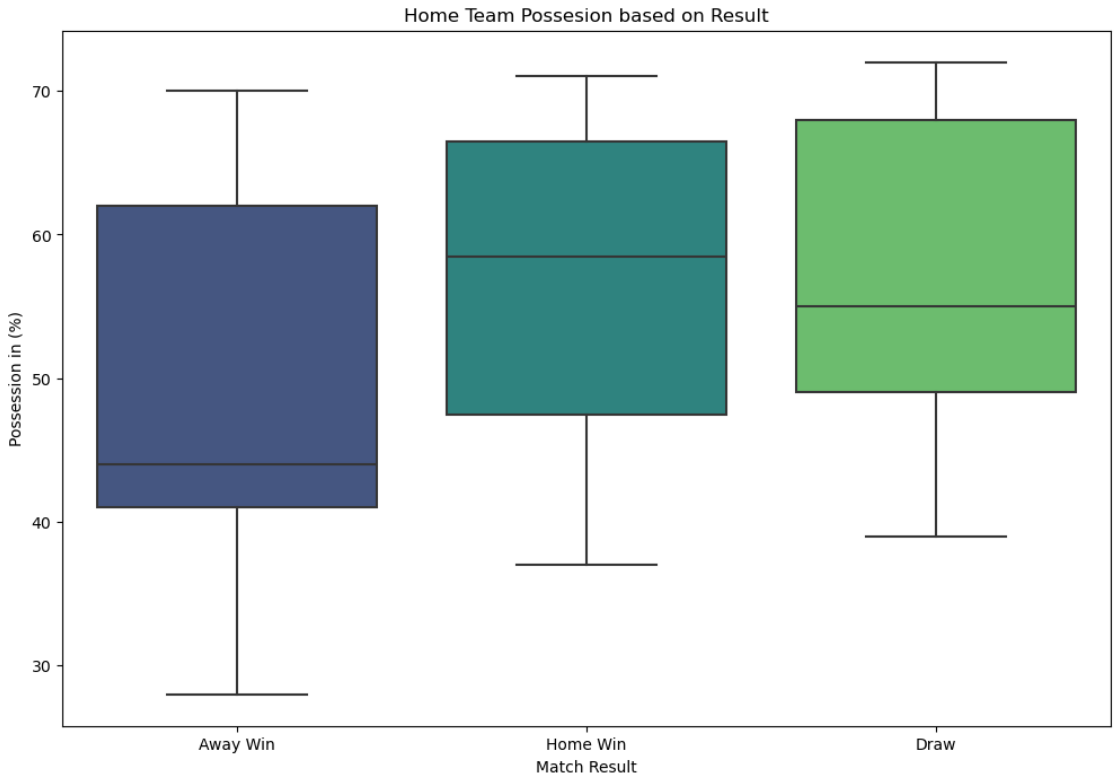
[71]:
```
plt.figure(figsize = (10,6))
sns.boxplot(x = "Result", y = "Total_Goals", data = df, palette = "Set2")
plt.title("Goals Distribution Per Match")
plt.xlabel("Match Results")
plt.ylabel("Shots on Target")
plt.show()
```



[72]:
```
plt.figure(figsize=(10,6))
sns.boxplot(x="Result", y="Home_Team_ShotsonTarget", data=df,␣
 ↪palette="coolwarm")
plt.title("Home Team Shots on Target on Results")
plt.xlabel("Match Results")
plt.ylabel("Shots on Target")
plt.show()
```



[73]:
```
plt.figure(figsize=(12,8))
sns.boxplot(x="Result", y="Home_Team_Possession", data = df, palette="viridis")
plt.title("Home Team Possesion based on Result")
plt.xlabel("Match Result")
plt.ylabel("Possession in (%)")
plt.show()
```



[74]:
```
best_attack = df.groupby("Home_Team") ["Home_Team_Score"].mean().
  ↪sort_values(ascending = False).head(3)
print("Top 3 Best Attacking Teams (Home):\n", best_attack)
```

```
Top 3 Best Attacking Teams (Home):
 Home_Team
Liverpool        4.0
Athletic Club    3.0
Tottenham        3.0
Name: Home_Team_Score, dtype: float64
```

[75]:
```
best_defence = df.groupby("Home_Team") ["Away_Team_Score"].mean().
  ↪sort_values(ascending = False).head(3)
print("Top 3 Best Defensive Teams (Home):\n", best_defence)
```

```
Top 3 Best Defensive Teams (Home):
 Home_Team
Wolves      4.0
Mallorca    3.0
Girona      3.0
Name: Away_Team_Score, dtype: float64
```

[76]:
```
aggressive = df.groupby("Home_Team") ["Home_Team_Fouls"].mean().
  ↪sort_values(ascending = False).head(3)
print("Top 3 Most Aggressive Teams (Home):\n", aggressive)
```

```
Top 3 Most Aggressive Teams (Home):
 Home_Team
Brighton      16.0
Alaves        15.0
Celta Vigo    15.0
Name: Home_Team_Fouls, dtype: float64
```

[77]:
```
shooting_accuracy = df.groupby("Home_Team") ["Home_Shooting_Accuracy"].mean().
  ↪sort_values(ascending = False).head(3)
print("Top 3 by Shooting Accuracy (Home):\n", shooting_accuracy)
```

```
Top 3 by Shooting Accuracy (Home):
 Home_Team
Aston Villa    1.000000
Espanyol       0.555556
Liverpool      0.526316
Name: Home_Shooting_Accuracy, dtype: float64
```

Conclusion

Match outcomes showed that home advantage was strong, with 50

Liverpool dominated at home with an average of four goals, while Manchester City led away with the same average.

In terms of discipline, Mallorca had the highest number of red cards, while some teams maintained better control.

Arsenal and Atletico Madrid recorded the most fouls and yellow cards when playing away.

Barcelona had the highest away possession at 72

Aston Villa and Oviedo achieved very high shooting accuracy, showing efficiency with fewer chances.

Overall, possession and passing were linked to stronger performance, while fouls and cards negatively affected results.

In summary, home teams performed better, possession-based teams like Barcelona and Real Madrid controlled matches, and efficient finishers such as Liverpool and Manchester City converted their chances effectively.