Perform Career Analysis of a Renowned Football Player

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

1 : Loading and Cleaning Data

1.1 Import Data

```
gs=pd.read_csv('C:\\Data Science Project\\goalscorers.csv')
#goalscoresrs dataset
res=pd.read_csv('C:\\Data Science Project\\results.csv') #results
dataset
shoot=pd.read_csv('C:\\Data Science Project\\shootouts.csv')
#shootouts dataset
```

1.2 Inspect the dataframe

This helps to give a good idea of the dataframes.

```
gs.head()
```

```
         date  home_team away_team       team            scorer
minute  \
0  1916-07-02      Chile   Uruguay    Uruguay   José Piendibene
44.0
1  1916-07-02      Chile   Uruguay    Uruguay  Isabelino Gradín
55.0
2  1916-07-02      Chile   Uruguay    Uruguay  Isabelino Gradín
70.0
3  1916-07-02      Chile   Uruguay    Uruguay   José Piendibene
75.0
4  1916-07-06  Argentina     Chile  Argentina     Alberto Ohaco
2.0

   own_goal  penalty
0     False    False
1     False    False
2     False    False
3     False    False
4     False    False
```

```
res.head()
```

```
         date home_team away_team  home_score  away_score tournament
city  \
0  1872-11-30  Scotland   England         0.0         0.0   Friendly
Glasgow
1  1873-03-08   England  Scotland         4.0         2.0   Friendly
London
2  1874-03-07  Scotland   England         2.0         1.0   Friendly
Glasgow
3  1875-03-06   England  Scotland         2.0         2.0   Friendly
London
4  1876-03-04  Scotland   England         3.0         0.0   Friendly
Glasgow

    country  neutral
0  Scotland    False
1   England    False
2  Scotland    False
3   England    False
4  Scotland    False
```

```
shoot.head()
```

```
         date    home_team         away_team       winner
first_shooter
0  1967-08-22        India            Taiwan       Taiwan
NaN
1  1971-11-14  South Korea  Vietnam Republic  South Korea
NaN
2  1972-05-07  South Korea              Iraq         Iraq
NaN
3  1972-05-17     Thailand       South Korea  South Korea
NaN
4  1972-05-19     Thailand          Cambodia     Thailand
NaN
```

```
gs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44350 entries, 0 to 44349
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   date       44350 non-null  object
 1   home_team  44350 non-null  object
 2   away_team  44350 non-null  object
 3   team       44350 non-null  object
 4   scorer     44301 non-null  object
 5   minute     44091 non-null  float64
```

```
 6   own_goal   44350 non-null  bool
 7   penalty    44350 non-null  bool
dtypes: bool(2), float64(1), object(5)
memory usage: 2.1+ MB
```

```
res.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47399 entries, 0 to 47398
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   date        47399 non-null  object
 1   home_team   47399 non-null  object
 2   away_team   47399 non-null  object
 3   home_score  47396 non-null  float64
 4   away_score  47396 non-null  float64
 5   tournament  47399 non-null  object
 6   city        47399 non-null  object
 7   country     47399 non-null  object
 8   neutral     47399 non-null  bool
dtypes: bool(1), float64(2), object(6)
memory usage: 2.9+ MB
```

```
shoot.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 643 entries, 0 to 642
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   date           643 non-null    object
 1   home_team      643 non-null    object
 2   away_team      643 non-null    object
 3   winner         643 non-null    object
 4   first_shooter  229 non-null    object
dtypes: object(5)
memory usage: 25.2+ KB
```

```
gs.shape
```

```
(44350, 8)
```

```
res.shape
```

```
(47399, 9)
```

```
shoot.shape
```

```
(643, 5)
```

```
gs.describe()
```

```
              minute
count  44091.000000
mean      50.011068
std       26.355009
min        1.000000
25%       28.000000
50%       51.000000
75%       73.000000
max      122.000000
```

```
res.describe()
```

```
         home_score    away_score
count  47396.000000  47396.000000
mean       1.759790      1.182927
std        1.775145      1.401625
min        0.000000      0.000000
25%        1.000000      0.000000
50%        1.000000      1.000000
75%        2.000000      2.000000
max       31.000000     21.000000
```

```
shoot.describe()
```

```
                date     home_team away_team     winner first_shooter
count            643           643       643        643           229
unique           566           181       189        175            87
top     2024-03-26  South Africa     Egypt  Argentina      Colombia
freq               5            18        15         15            11
```

EDA Data Cleaning Dataframe

```
# Check for missing values
print(gs.isnull().sum())
print(res.isnull().sum())
print(shoot.isnull().sum())
```

```
date           0
home_team      0
away_team      0
team           0
scorer        49
minute       259
own_goal       0
penalty        0
dtype: int64
date          0
home_team     0
away_team     0
```

```
home_score    3
away_score    3
tournament    0
city          0
country       0
neutral       0
dtype: int64
date               0
home_team          0
away_team          0
winner             0
first_shooter    414
dtype: int64
```

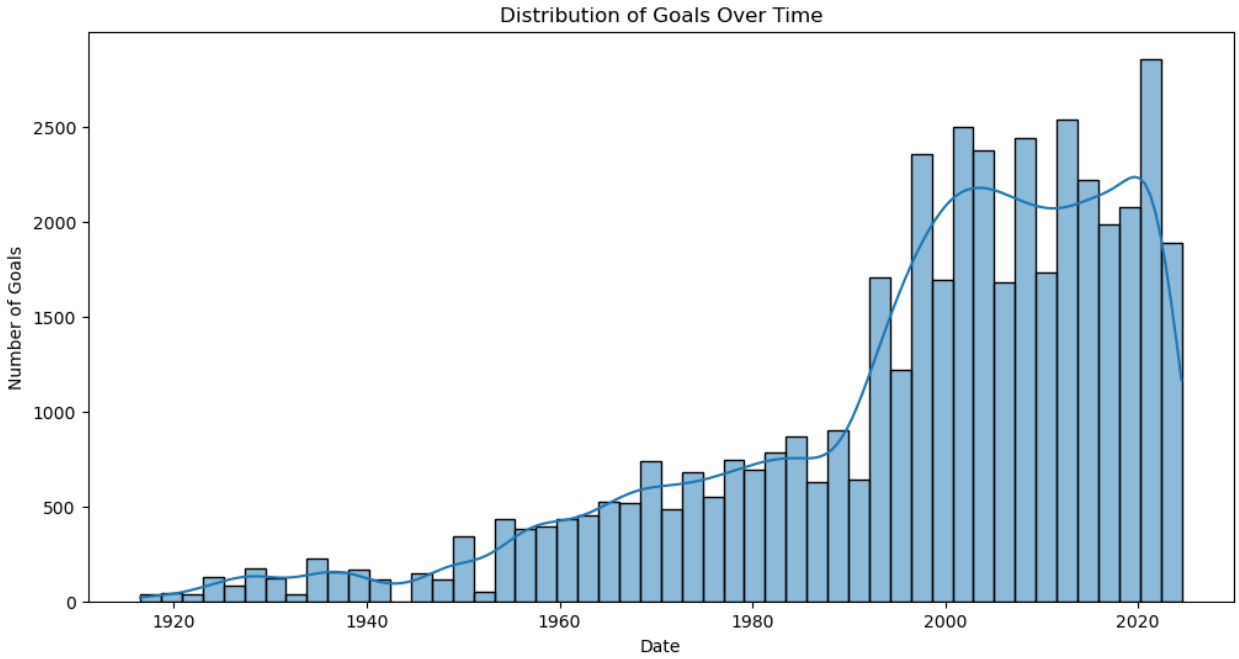
```
# Check for missing values
print(gs.isnull().sum())
print(res.isnull().sum())
print(shoot.isnull().sum())

# Drop rows with missing 'scorer' or 'minute' values in gs
gs = gs.dropna(subset=['scorer', 'minute'])

# results_df.fillna(value={'column_name': 'placeholder_value'},
inplace=True)
# shoot.fillna(shoot.mean(), inplace=True)
```

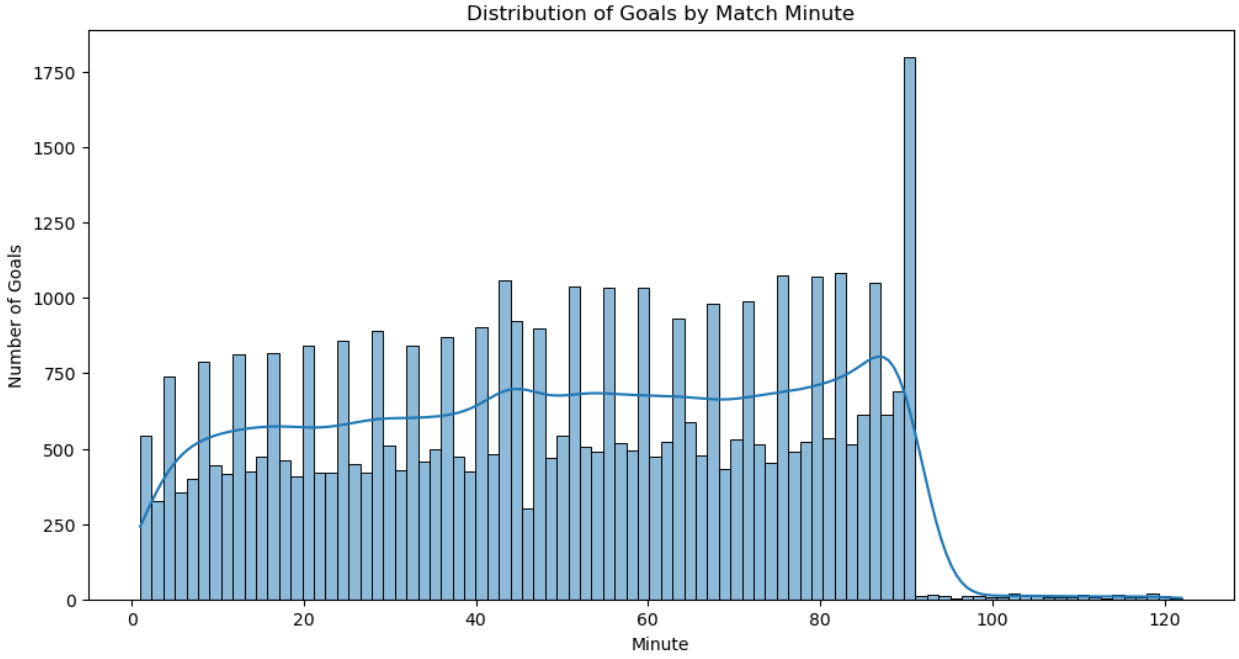
```
date         0
home_team    0
away_team    0
team         0
scorer       0
minute       0
own_goal     0
penalty      0
dtype: int64
date          0
home_team     0
away_team     0
home_score    3
away_score    3
tournament    0
city          0
country       0
neutral       0
dtype: int64
date               0
home_team          0
away_team          0
winner             0
```

```
first_shooter    414
dtype: int64
```

```
# Convert 'date' columns to datetime format
gs['date'] = pd.to_datetime(gs['date'])
res['date'] = pd.to_datetime(res['date'])
shoot['date'] = pd.to_datetime(shoot['date'])

# Ensure boolean columns are of boolean type
gs['own_goal'] = gs['own_goal'].astype(bool)
gs['penalty'] = gs['penalty'].astype(bool)

# Total number of goals by each player
total_goals_by_player = gs['scorer'].value_counts()
print(total_goals_by_player.head())

# Plot the distribution of goals over time
plt.figure(figsize=(12, 6))
sns.histplot(gs['date'], bins=50, kde=True)
plt.title('Distribution of Goals Over Time')
plt.xlabel('Date')
plt.ylabel('Number of Goals')
plt.show()

# Goals by minute
plt.figure(figsize=(12, 6))
sns.histplot(gs['minute'], bins=90, kde=True)
plt.title('Distribution of Goals by Match Minute')
plt.xlabel('Minute')
plt.ylabel('Number of Goals')
plt.show()
```

```
scorer
Cristiano Ronaldo     108
Robert Lewandowski     63
Romelu Lukaku          60
Harry Kane             57
Lionel Messi           55
Name: count, dtype: int64
```

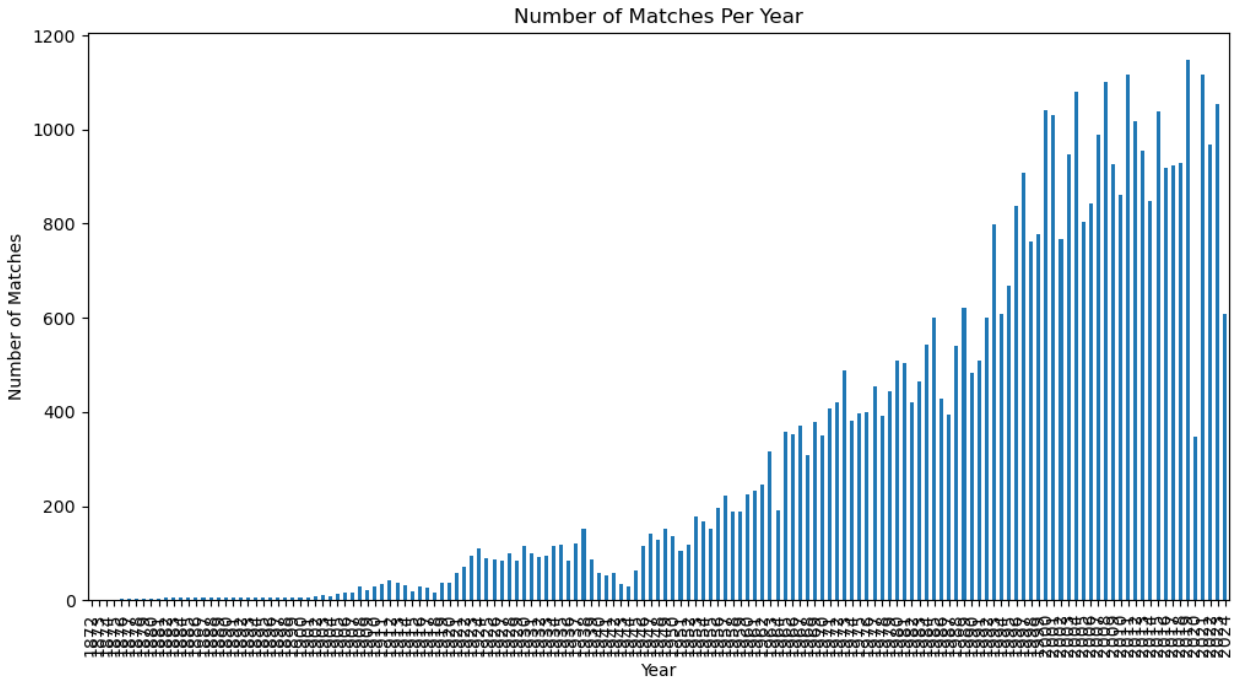





```
# Number of matches per year
matches_per_year = res['date'].dt.year.value_counts().sort_index()
print(matches_per_year)

# Plot the number of matches per year
plt.figure(figsize=(12, 6))
matches_per_year.plot(kind='bar')
plt.title('Number of Matches Per Year')
plt.xlabel('Year')
```

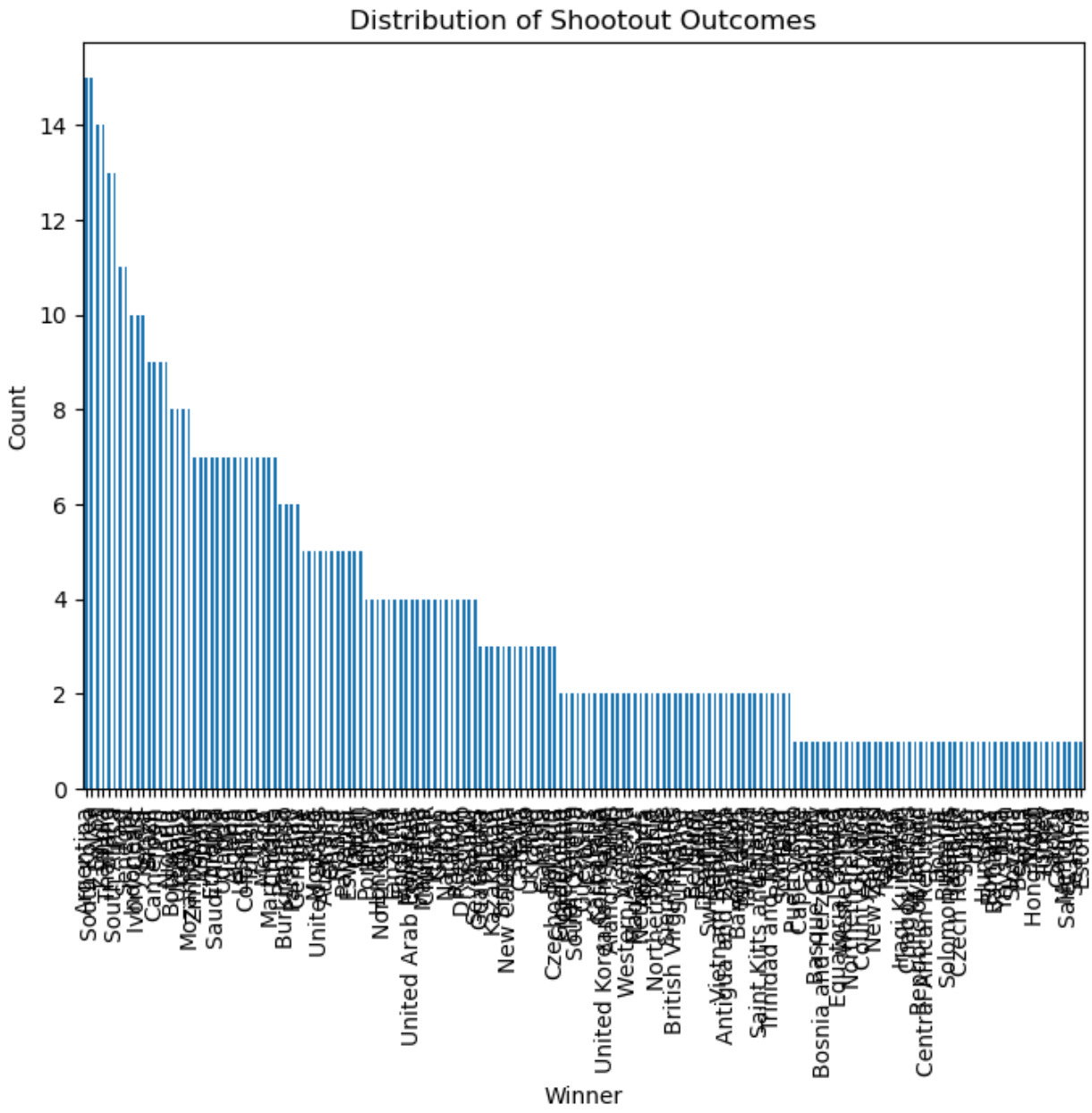
```
plt.ylabel('Number of Matches')
plt.show()
```

```
date
1872       1
1873       1
1874       1
1875       1
1876       2
          ...
2020     347
2021    1115
2022     969
2023    1054
2024     608
Name: count, Length: 153, dtype: int64
```




```
# Distribution of shootout outcomes based on the 'winner' column
shootout_outcomes = shoot['winner'].value_counts()
print(shootout_outcomes)

# Plot shootout outcomes
plt.figure(figsize=(8, 6))
shootout_outcomes.plot(kind='bar')
plt.title('Distribution of Shootout Outcomes')
plt.xlabel('Winner')
plt.ylabel('Count')
plt.show()
```

```
winner
Argentina      15
South Korea    15
Egypt          14
Zambia         14
Thailand       13
               ..
Corsica         1
Menorca         1
Saint Lucia     1
Sealand         1
Estonia         1
Name: count, Length: 175, dtype: int64
```

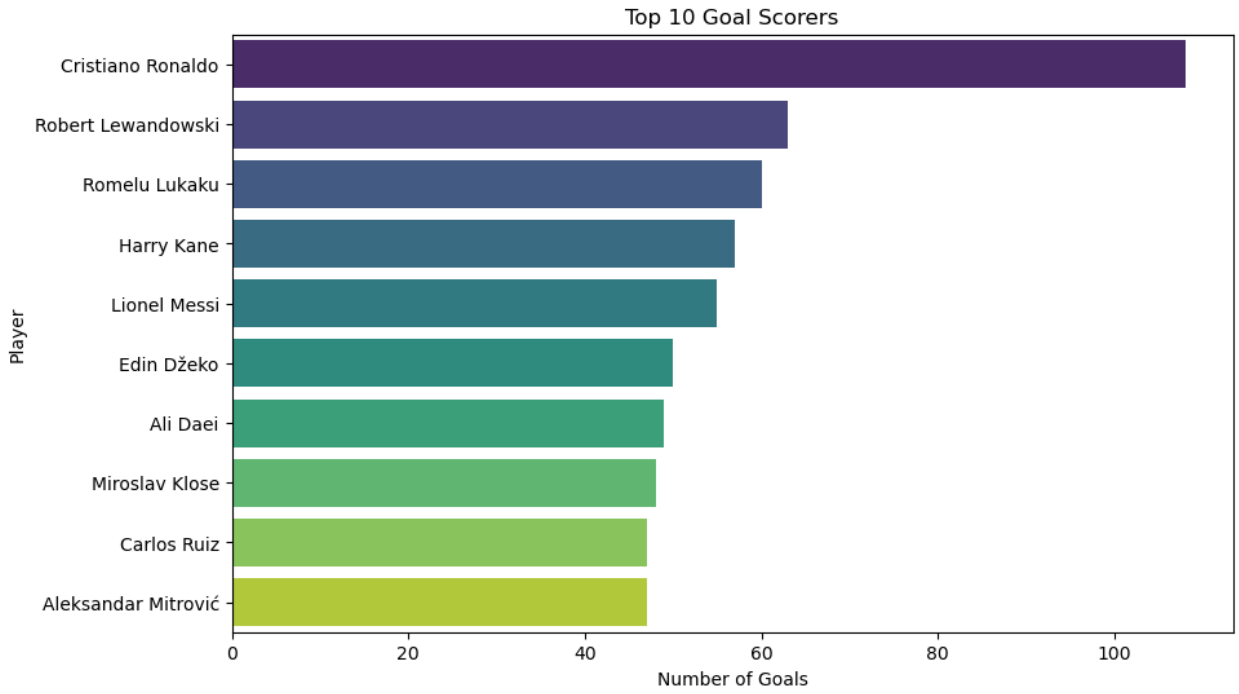


Data Visualization

Goalscorers Data ## 1 Total Number of Goals by Each Player

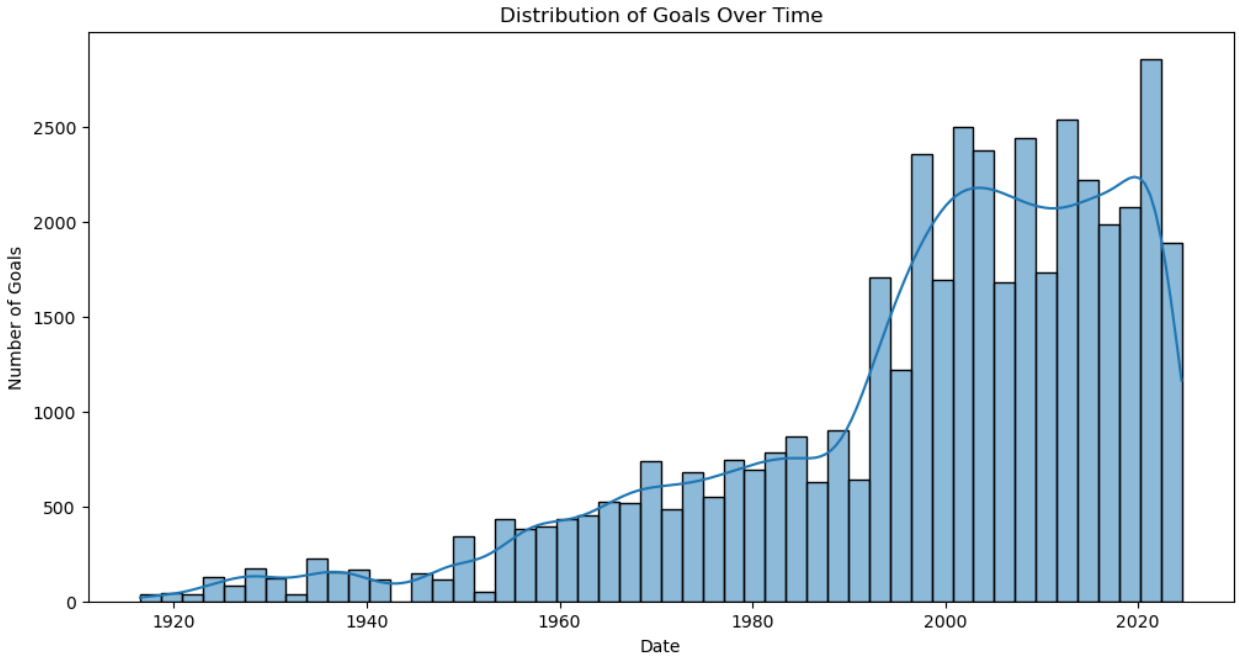
```
total_goals_by_player = gs['scorer'].value_counts().head(10)
plt.figure(figsize=(10, 6))
sns.barplot(x=total_goals_by_player.values,
y=total_goals_by_player.index, palette='viridis')
plt.title('Top 10 Goal Scorers')
plt.xlabel('Number of Goals')
```

```
plt.ylabel('Player')
plt.show()
```



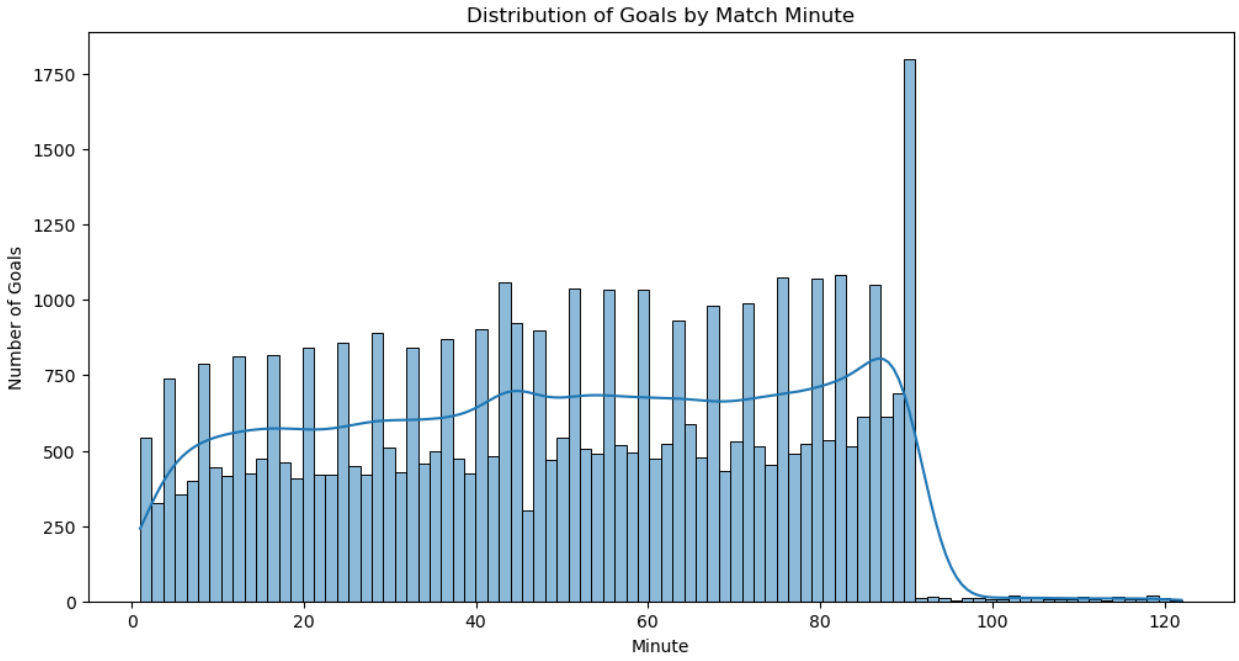
2 Distribution of Goals Over Time

```
gs['date'] = pd.to_datetime(gs['date'])
plt.figure(figsize=(12, 6))
sns.histplot(gs['date'], bins=50, kde=True)
plt.title('Distribution of Goals Over Time')
plt.xlabel('Date')
plt.ylabel('Number of Goals')
plt.show()
```



3 Goals by Match Minute

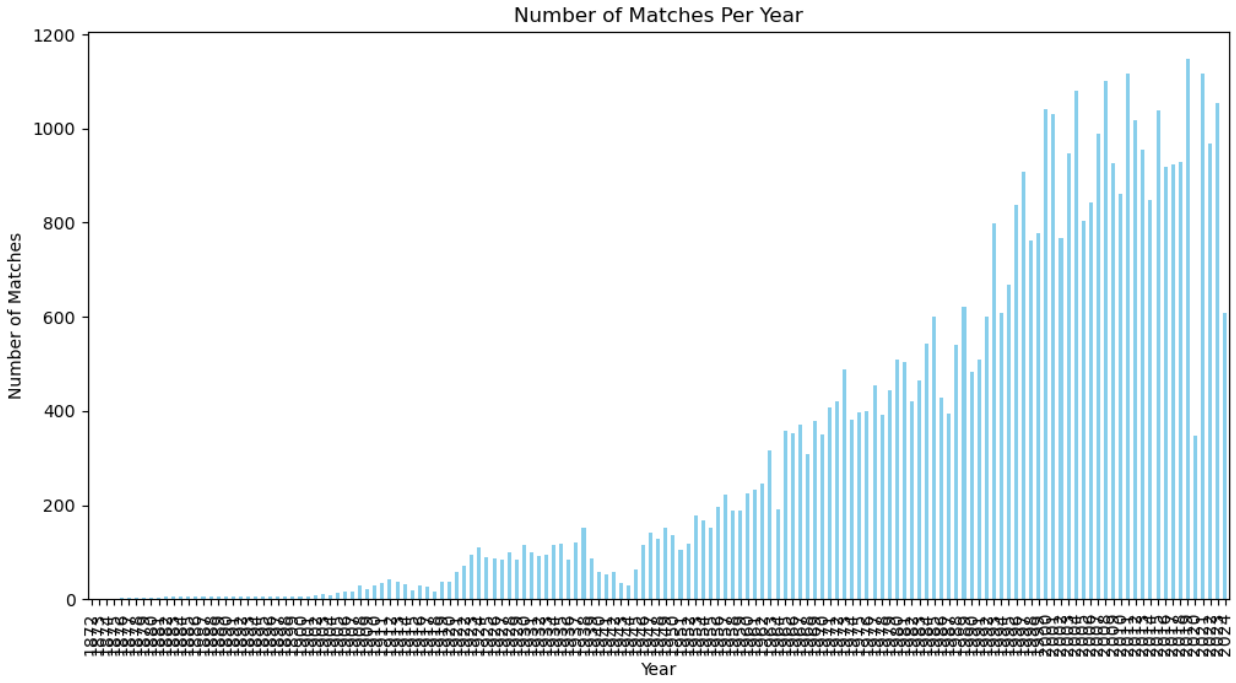
```
plt.figure(figsize=(12, 6))
sns.histplot(gs['minute'], bins=90, kde=True)
plt.title('Distribution of Goals by Match Minute')
plt.xlabel('Minute')
plt.ylabel('Number of Goals')
plt.show()
```



Results Data

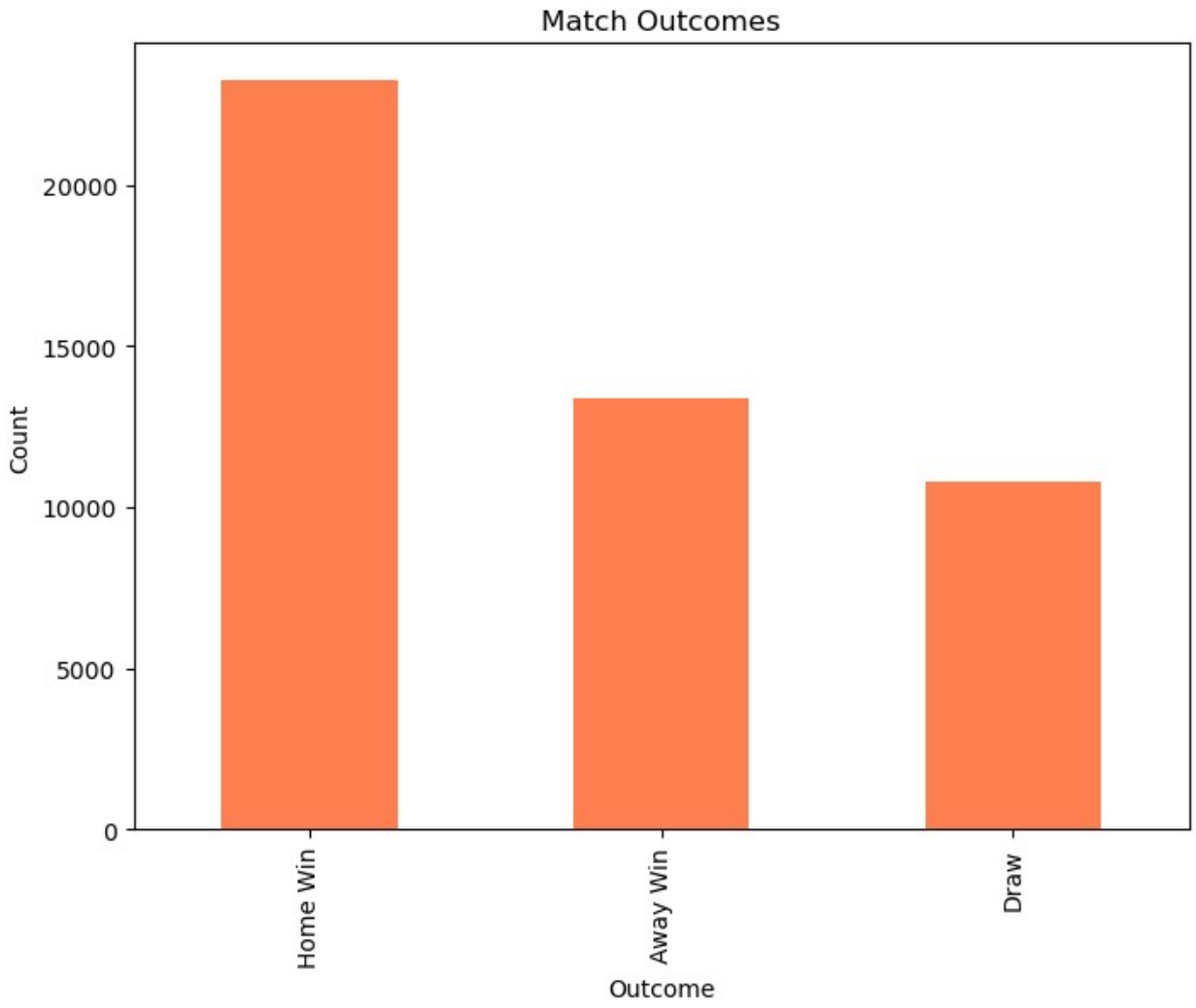
1 Number of Matches Per Year

```
res['date'] = pd.to_datetime(res['date'])
matches_per_year = res['date'].dt.year.value_counts().sort_index()
plt.figure(figsize=(12, 6))
matches_per_year.plot(kind='bar', color='skyblue')
plt.title('Number of Matches Per Year')
plt.xlabel('Year')
plt.ylabel('Number of Matches')
plt.show()
```



2 Match Outcomes (Home Wins, Away Wins, Draws)

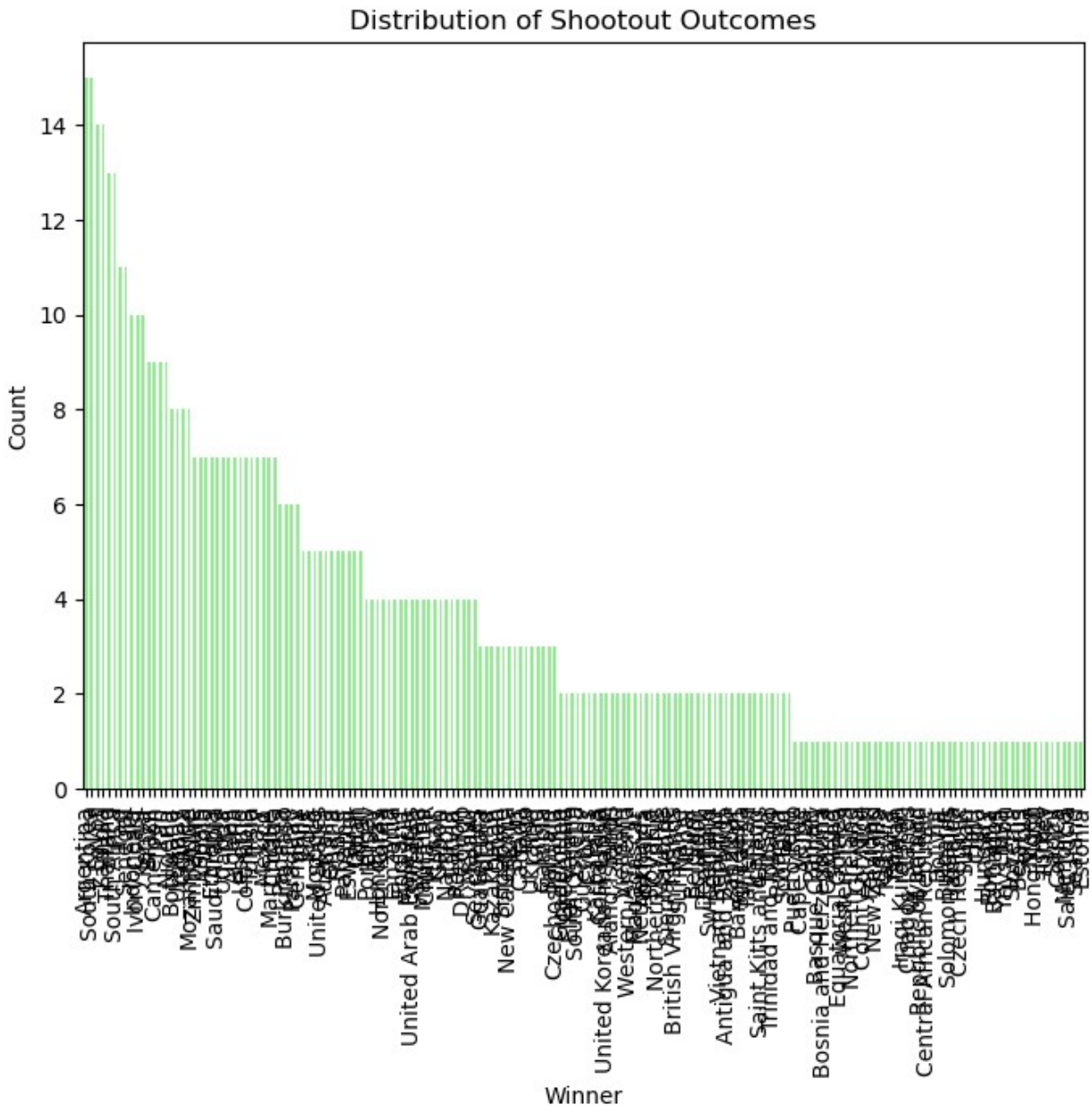
```
# Assuming the 'results' DataFrame has 'home_score' and 'away_score' columns
res['result'] = np.where(res['home_score'] > res['away_score'], 'Home Win',
                                np.where(res['home_score'] < res['away_score'], 'Away Win', 'Draw'))
match_outcomes = res['result'].value_counts()

plt.figure(figsize=(8, 6))
match_outcomes.plot(kind='bar', color='coral')
plt.title('Match Outcomes')
plt.xlabel('Outcome')
plt.ylabel('Count')
plt.show()
```



Shootouts Data

1 Distribution of Shootout Outcomes

```
shootout_outcomes = shoot['winner'].value_counts()
plt.figure(figsize=(8, 6))
shootout_outcomes.plot(kind='bar', color='lightgreen')
plt.title('Distribution of Shootout Outcomes')
plt.xlabel('Winner')
plt.ylabel('Count')
plt.show()
```



2 First Shooter and Outcome Correlation

```
# Count the number of times each team went first and won
first_shooter_wins = shoot.groupby(['first_shooter',
'winner']).size().unstack().fillna(0)
first_shooter_wins.plot(kind='bar', stacked=True, figsize=(10, 6),
colormap='viridis')
plt.title('First Shooter and Outcome Correlation')
plt.xlabel('First Shooter')
```

```
plt.ylabel('Count')
plt.show()
```


winner
Algeria
Argentina
Brazil
Uruguay
Australia
Japan
Bahrain
Belgium
Benin
Northern Ireland
France
Paraguay
British Virgin Islands
Burkina Faso
Cameroon
Egypt
Ivory Coast
Canada
United States
South Africa
Chile
Ghana
China PR
Colombia
England
Costa Rica
Honduras
Netherlands
Croatia
Spain
Turkey
Czechoslovakia
DR Congo
Denmark
Mexico
Germany
Portugal
Equatorial Guinea
Latvia
Senegal
Czech Republic
Saint Kitts and Nevis
Trinidad and Tobago
Mali
Georgia
Tunisia
Guadeloupe
Guyana
Iceland
India
Iraq
Iran
Saudi Arabia
Italy

Data Correlation

Goalscorers Data Correlation

```
# Display unique values for potential categorical columns
for column in gs.columns:
    print(column, gs[column].unique())
```

```
date <DatetimeArray>
['1916-07-02 00:00:00', '1916-07-06 00:00:00', '1916-07-08 00:00:00',
 '1916-07-10 00:00:00', '1916-07-12 00:00:00', '1917-09-30 00:00:00',
 '1917-10-03 00:00:00', '1917-10-06 00:00:00', '1917-10-07 00:00:00',
 '1917-10-12 00:00:00',
 ...
 '2024-06-27 00:00:00', '2024-06-28 00:00:00', '2024-06-29 00:00:00',
 '2024-06-30 00:00:00', '2024-07-01 00:00:00', '2024-07-02 00:00:00',
 '2024-07-04 00:00:00', '2024-07-05 00:00:00', '2024-07-06 00:00:00',
 '2024-07-09 00:00:00']
Length: 4607, dtype: datetime64[ns]
home_team ['Chile' 'Argentina' 'Brazil' 'Uruguay' 'Paraguay'
'Czechoslovakia'
 'Italy' 'Switzerland' 'United States' 'Hungary' 'France'
'Netherlands'
 'Republic of Ireland' 'Egypt' 'Sweden' 'Bolivia' 'Peru' 'Belgium'
 'Portugal' 'Germany' 'Spain' 'Latvia' 'Estonia' 'Lithuania'
'Yugoslavia'
 'Poland' 'Haiti' 'Mexico' 'Luxembourg' 'Bulgaria' 'Israel' 'Austria'
 'Romania' 'Finland' 'Norway' 'Greece' 'Cuba' 'Ecuador' 'Colombia'
 'Northern Ireland' 'Wales' 'Scotland' 'England' 'Turkey' 'Japan'
 'Saarland' 'South Korea' 'Hong Kong' 'Guatemala' 'Sudan' 'Costa Rica'
 'Indonesia' 'Denmark' 'German DR' 'Syria' 'China PR' 'Canada'
'Russia'
 'Curaçao' 'Iceland' 'Ethiopia' 'Ghana' 'Honduras' 'Nigeria'
'Suriname'
 'Taiwan' 'Morocco' 'Cyprus' 'Tunisia' 'Malta' 'El Salvador' 'Jamaica'
 'Nicaragua' 'Panama' 'Albania' 'India' 'Trinidad and Tobago'
'Venezuela'
 'DR Congo' 'Ivory Coast' 'North Korea' 'Australia' 'Algeria' 'Congo'
 'Iran' 'Myanmar' 'Zambia' 'Bermuda' 'Senegal' 'Libya' 'Zimbabwe'
 'Cameroon' 'Mali' 'Kenya' 'Guinea' 'Thailand' 'Cambodia' 'Benin'
 'Sierra Leone' 'Tanzania' 'Mauritius' 'New Zealand' 'Iraq'
 'Vietnam Republic' 'Uganda' 'Burkina Faso' 'Niger' 'Mauritania'
'Malawi'
 'Kuwait' 'Togo' 'Saudi Arabia' 'Singapore' 'Malaysia' 'Bahrain'
'Qatar'
 'Fiji' 'New Caledonia' 'Lesotho' 'Mozambique' 'Somalia' 'Bangladesh'
 'United Arab Emirates' 'Madagascar' 'Angola' 'Gambia' 'Liberia'
'Macau'
```

```
 'Brunei' 'Nepal' 'Jordan' 'Yemen' 'Yemen DPR' 'Oman' 'Pakistan'
'Gabon'
 'Faroe Islands' 'San Marino' 'Dominican Republic' 'Saint Lucia'
 'Puerto Rico' 'Saint Vincent and the Grenadines' 'Barbados' 'Guyana'
 'Antigua and Barbuda' 'Vanuatu' 'Solomon Islands' 'Tahiti' 'Burundi'
 'Eswatini' 'Namibia' 'South Africa' 'Botswana' 'Sri Lanka' 'Vietnam'
 'Czech Republic' 'Lebanon' 'Georgia' 'Liechtenstein' 'North
Macedonia'
 'Slovenia' 'Ukraine' 'Croatia' 'Belarus' 'Moldova' 'Azerbaijan'
 'Slovakia' 'Armenia' 'Dominica' 'Aruba' 'Grenada' 'Serbia'
 'Saint Kitts and Nevis' 'Cayman Islands' 'Guinea-Bissau' 'Belize'
 'Rwanda' 'Papua New Guinea' 'Philippines' 'Bosnia and Herzegovina'
 'Tonga' 'Samoa' 'Uzbekistan' 'Tajikistan' 'Turkmenistan' 'Kazakhstan'
 'Maldives' 'Kyrgyzstan' 'Andorra' 'Anguilla' 'British Virgin Islands'
 'Bahamas' 'Montserrat' 'United States Virgin Islands' 'Djibouti'
 'Central African Republic' 'Seychelles' 'São Tomé and Príncipe'
'Chad'
 'Equatorial Guinea' 'Mongolia' 'Palestine' 'American Samoa' 'Laos'
 'Cook Islands' 'Martinique' 'Cape Verde' 'Afghanistan'
 'Turks and Caicos Islands' 'Guadeloupe' 'Timor-Leste' 'Comoros'
 'Montenegro' 'Eritrea' 'Gibraltar' 'Bhutan' 'Guam' 'South Sudan'
'Kosovo'
 'French Guiana']
away_team ['Uruguay' 'Chile' 'Brazil' 'Argentina' 'Paraguay' 'Turkey'
'Spain'
 'Lithuania' 'Estonia' 'Poland' 'Yugoslavia' 'Latvia' 'Romania'
'Bulgaria'
 'Czechoslovakia' 'Hungary' 'Luxembourg' 'Belgium' 'United States'
'Egypt'
 'Republic of Ireland' 'Italy' 'Sweden' 'Netherlands' 'Switzerland'
 'Bolivia' 'France' 'Mexico' 'Germany' 'Portugal' 'Peru' 'Cuba'
'Israel'
 'Greece' 'Austria' 'Finland' 'Norway' 'Indonesia' 'Ecuador'
'Colombia'
 'Scotland' 'England' 'Wales' 'Northern Ireland' 'Syria' 'Saarland'
 'Haiti' 'South Korea' 'Vietnam Republic' 'Denmark' 'Costa Rica'
 'Ethiopia' 'Guatemala' 'Curaçao' 'China PR' 'Sudan' 'Iceland' 'German
DR'
 'Canada' 'Russia' 'Nigeria' 'Ghana' 'Honduras' 'Taiwan' 'Tunisia'
'Japan'
 'Morocco' 'Cyprus' 'Uganda' 'Malta' 'Nicaragua' 'Panama' 'Jamaica'
 'Albania' 'Hong Kong' 'India' 'Suriname' 'Trinidad and Tobago'
 'El Salvador' 'Venezuela' 'Ivory Coast' 'Senegal' 'Australia'
 'North Korea' 'DR Congo' 'Algeria' 'Myanmar' 'Bermuda' 'Zambia'
 'Cameroon' 'New Zealand' 'Guinea' 'Kenya' 'Togo' 'Mali' 'Congo'
 'Cambodia' 'Kuwait' 'Iraq' 'Iran' 'Lesotho' 'Benin' 'Sierra Leone'
 'Tanzania' 'Mauritius' 'Malaysia' 'Thailand' 'Niger' 'Mauritania'
 'Burkina Faso' 'Libya' 'Malawi' 'Yemen DPR' 'Saudi Arabia' 'Bahrain'
 'Tahiti' 'Solomon Islands' 'Fiji' 'Mozambique' 'United Arab Emirates'
```

```
 'Bangladesh' 'Qatar' 'Zimbabwe' 'Madagascar' 'Liberia' 'Macau'
 'Singapore' 'Angola' 'Gambia' 'Brunei' 'Nepal' 'Yemen' 'Jordan'
'Gabon'
 'Oman' 'Pakistan' 'Faroe Islands' 'San Marino' 'Puerto Rico'
 'Saint Vincent and the Grenadines' 'Dominican Republic' 'Saint Lucia'
 'Antigua and Barbuda' 'Guyana' 'Barbados' 'Vanuatu' 'Burundi'
'Botswana'
 'South Africa' 'Namibia' 'Eswatini' 'Czech Republic' 'Vietnam'
 'Sri Lanka' 'Lebanon' 'Martinique' 'Liechtenstein' 'Croatia'
'Armenia'
 'Moldova' 'Belarus' 'Azerbaijan' 'Georgia' 'Slovakia' 'North
Macedonia'
 'Ukraine' 'Slovenia' 'Aruba' 'Grenada' 'Dominica' 'Saint Kitts and
Nevis'
 'Belize' 'Guinea-Bissau' 'Rwanda' 'Bosnia and Herzegovina'
'Philippines'
 'Serbia' 'Cook Islands' 'Samoa' 'Uzbekistan' 'Tonga' 'Tajikistan'
 'Turkmenistan' 'Maldives' 'Kazakhstan' 'Kyrgyzstan' 'Papua New
Guinea'
 'Andorra' 'Bahamas' 'Cayman Islands' 'Montserrat'
 'United States Virgin Islands' 'Turks and Caicos Islands' 'Anguilla'
 'British Virgin Islands' 'Somalia' 'Cape Verde' 'Seychelles'
'Eritrea'
 'São Tomé and Príncipe' 'Equatorial Guinea' 'Djibouti'
 'Central African Republic' 'Guam' 'Mongolia' 'Palestine' 'American
Samoa'
 'Laos' 'New Caledonia' 'Chad' 'Afghanistan' 'Guadeloupe' 'Comoros'
 'Timor-Leste' 'Montenegro' 'Gibraltar' 'Bhutan' 'South Sudan'
'Kosovo'
 'French Guiana']
team ['Uruguay' 'Argentina' 'Chile' 'Brazil' 'Paraguay'
'Czechoslovakia'
 'Turkey' 'Italy' 'Switzerland' 'United States' 'Hungary' 'France'
 'Netherlands' 'Republic of Ireland' 'Egypt' 'Sweden' 'Belgium'
'Bolivia'
 'Peru' 'Luxembourg' 'Portugal' 'Germany' 'Yugoslavia' 'Spain'
'Mexico'
 'Latvia' 'Estonia' 'Lithuania' 'Romania' 'Poland' 'Cuba' 'Haiti'
'Israel'
 'Bulgaria' 'Austria' 'Norway' 'Greece' 'Ecuador' 'Colombia'
'Scotland'
 'Northern Ireland' 'Finland' 'England' 'Wales' 'Saarland' 'Japan'
 'South Korea' 'Hong Kong' 'Vietnam Republic' 'Denmark' 'Costa Rica'
 'Guatemala' 'Sudan' 'Curaçao' 'Indonesia' 'German DR' 'Syria' 'China
PR'
 'Iceland' 'Canada' 'Russia' 'Ghana' 'Nigeria' 'Honduras' 'Suriname'
 'Taiwan' 'Morocco' 'Tunisia' 'Cyprus' 'Ethiopia' 'Uganda' 'Malta'
 'El Salvador' 'Nicaragua' 'Panama' 'Jamaica' 'Albania' 'India'
 'Trinidad and Tobago' 'Venezuela' 'DR Congo' 'Ivory Coast' 'Senegal'
```

```
 'North Korea' 'Australia' 'Algeria' 'Congo' 'Iran' 'Myanmar' 'Zambia'
 'Bermuda' 'Cameroon' 'Libya' 'Zimbabwe' 'Guinea' 'Kenya' 'Mali'
'Togo'
 'Kuwait' 'Cambodia' 'Iraq' 'Thailand' 'Lesotho' 'Tanzania'
'Mauritius'
 'New Zealand' 'Malaysia' 'Niger' 'Sierra Leone' 'Burkina Faso'
 'Mauritania' 'Saudi Arabia' 'Singapore' 'Qatar' 'Bahrain' 'Tahiti'
'Fiji'
 'Solomon Islands' 'New Caledonia' 'Mozambique' 'Malawi' 'Somalia'
 'United Arab Emirates' 'Bangladesh' 'Madagascar' 'Angola' 'Gambia'
 'Macau' 'Jordan' 'Yemen DPR' 'Brunei' 'Yemen' 'Liberia' 'Oman'
'Gabon'
 'Pakistan' 'Faroe Islands' 'San Marino' 'Puerto Rico'
 'Dominican Republic' 'Saint Lucia' 'Saint Vincent and the Grenadines'
 'Barbados' 'Antigua and Barbuda' 'Guyana' 'Vanuatu' 'Burundi' 'Benin'
 'Eswatini' 'South Africa' 'Botswana' 'Czech Republic' 'Vietnam'
'Lebanon'
 'Martinique' 'Liechtenstein' 'Croatia' 'Moldova' 'North Macedonia'
 'Slovenia' 'Belarus' 'Slovakia' 'Ukraine' 'Georgia' 'Azerbaijan'
 'Armenia' 'Dominica' 'Aruba' 'Grenada' 'Serbia' 'Saint Kitts and
Nevis'
 'Guinea-Bissau' 'Namibia' 'Belize' 'Rwanda' 'Papua New Guinea'
 'Sri Lanka' 'Bosnia and Herzegovina' 'Tonga' 'Samoa' 'Cook Islands'
 'Uzbekistan' 'Nepal' 'Tajikistan' 'Turkmenistan' 'Kazakhstan'
 'Kyrgyzstan' 'Andorra' 'Bahamas' 'Anguilla' 'British Virgin Islands'
 'Montserrat' 'United States Virgin Islands' 'Djibouti' 'Seychelles'
 'São Tomé and Príncipe' 'Equatorial Guinea' 'Central African
Republic'
 'Mongolia' 'Palestine' 'Maldives' 'Philippines' 'Laos' 'Chad'
 'Cape Verde' 'Cayman Islands' 'American Samoa' 'Guadeloupe' 'Comoros'
 'Timor-Leste' 'Afghanistan' 'Turks and Caicos Islands' 'Montenegro'
 'Eritrea' 'Bhutan' 'Gibraltar' 'Guam' 'South Sudan' 'Kosovo'
 'French Guiana']
scorer ['José Piendibene' 'Isabelino Gradín' 'Alberto Ohaco' ...
 'Lisandro Martínez' 'Kevin Rodríguez' 'Richard Ríos']
minute [ 44.  55.  70.  75.   2.  60.  62.  67.  81.  29.  85.  10.
23.   8.
  58.  77.  20.  15.  39.  56.  80.  76.  17.  86.  21.  26.  41.  59.
  19.  38.  79.  83.  34.  31.  43.  22.  57.  65.  13.  33.  63. 122.
  53.  48.  30.  40.  73.  37.  27.   9.  66.  46.   1.  71.  14.  36.
   5.  78.   7.  42.  11.  89.  18.  49.  87.  28.  88.  64.  74.  82.
  84.  50.  68.  51.  35.  61.  25.  32.  52.  69.   4.  24.  12. 104.
  47.   6.  72.  16.   3.  90.  54.  45.  93. 109. 116.  95. 102. 112.
 118. 103. 105. 111.  94. 114.  91.  97. 113. 101. 107. 110.  96. 120.
 100. 108. 117.  98.  92. 115. 119. 106.  99.]
own_goal [False  True]
penalty [False  True]

# Encode categorical columns
if 'team' in gs.columns:
```

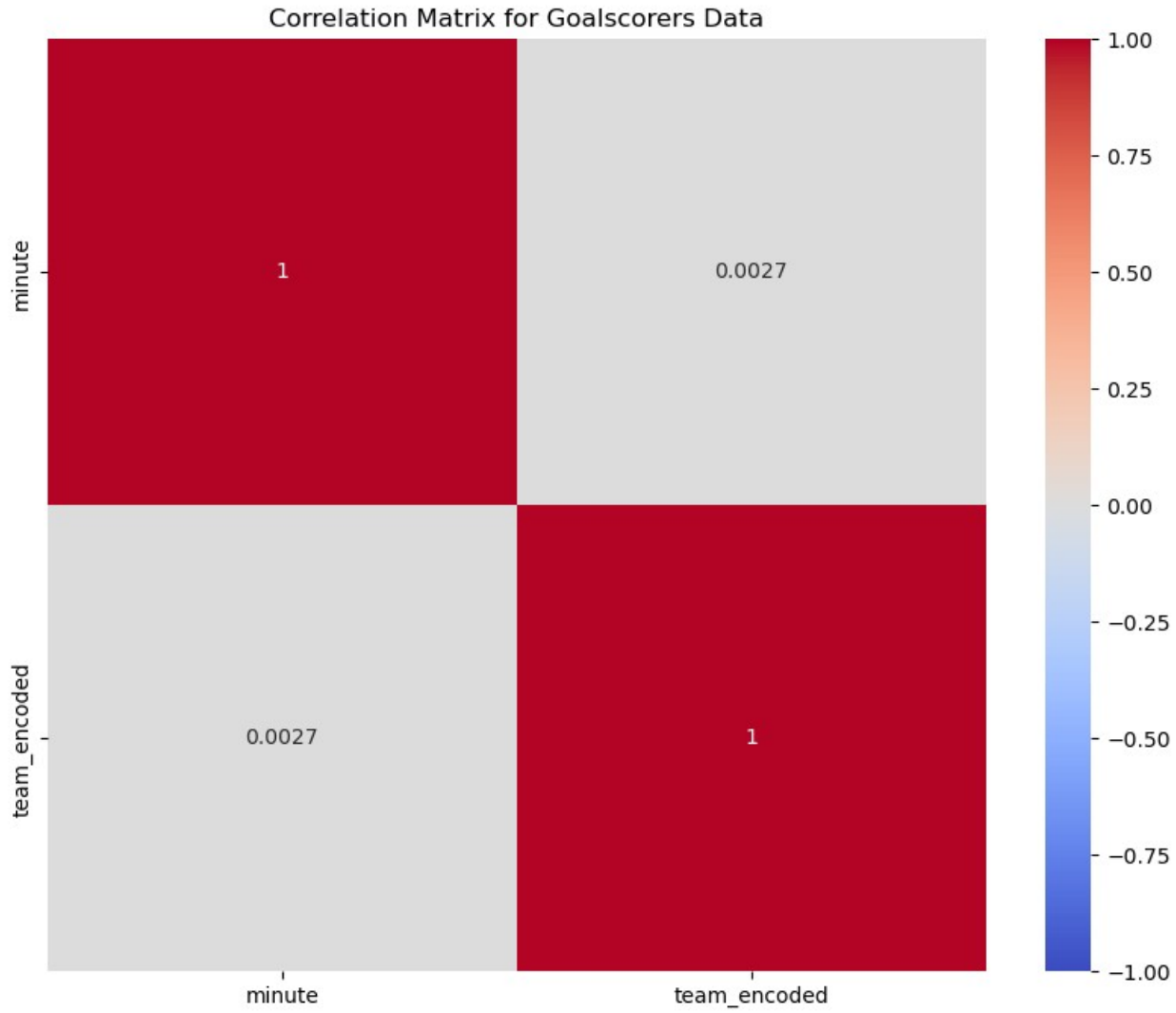
```
    gs['team_encoded'] = gs['team'].astype('category').cat.codes
if 'opponent' in gs.columns:
    gs['opponent_encoded'] =
gs['opponent'].astype('category').cat.codes

# Select only numeric columns for correlation calculation
numeric_columns = gs.select_dtypes(include=[np.number]).columns
gs_numeric = gs[numeric_columns]

# Display the first few rows of the numeric data
print(gs_numeric.head())
```

```
   minute  team_encoded
0    44.0           208
1    55.0           208
2    70.0           208
3    75.0           208
4     2.0             8
```

```
# Compute the correlation matrix
correlation_matrix_goalscorers = gs_numeric.corr()

# Print the correlation matrix
print(correlation_matrix_goalscorers)
```

```
                minute  team_encoded
minute        1.000000      0.002696
team_encoded  0.002696      1.000000
```

```
# Visualize the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix_goalscorers, annot=True,
cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix for Goalscorers Data')
plt.show()
```




```
# Display unique values for potential categorical columns
for column in res.columns:
    print(column, res[column].unique())
```

```
date <DatetimeArray>
['1872-11-30 00:00:00', '1873-03-08 00:00:00', '1874-03-07 00:00:00',
 '1875-03-06 00:00:00', '1876-03-04 00:00:00', '1876-03-25 00:00:00',
 '1877-03-03 00:00:00', '1877-03-05 00:00:00', '1878-03-02 00:00:00',
 '1878-03-23 00:00:00',
 ...
 '2024-06-30 00:00:00', '2024-07-01 00:00:00', '2024-07-02 00:00:00',
 '2024-07-04 00:00:00', '2024-07-05 00:00:00', '2024-07-06 00:00:00',
 '2024-07-09 00:00:00', '2024-07-10 00:00:00', '2024-07-13 00:00:00',
 '2024-07-14 00:00:00']
Length: 16204, dtype: datetime64[ns]
home_team ['Scotland' 'England' 'Wales' 'Northern Ireland' 'United
States' 'Uruguay'
```

```
 'Austria' 'Hungary' 'Argentina' 'Belgium' 'France' 'Guernsey'
'Jersey'
 'Netherlands' 'Czechoslovakia' 'Alderney' 'Switzerland' 'Sweden'
 'Germany' 'Italy' 'Chile' 'Norway' 'Finland' 'Luxembourg' 'Russia'
 'Denmark' 'Catalonia' 'Philippines' 'Basque Country' 'China PR'
'Brazil'
 'Japan' 'Paraguay' 'Canada' 'Estonia' 'Costa Rica' 'Guatemala'
'Spain'
 'Brittany' 'Poland' 'Yugoslavia' 'New Zealand' 'Romania' 'Latvia'
 'Galicia' 'Portugal' 'Andalusia' 'Australia' 'Lithuania' 'Turkey'
 'Central Spain' 'Mexico' 'Aruba' 'Egypt' 'Republic of Ireland'
'Haiti'
 'Bulgaria' 'Jamaica' 'Kenya' 'Bolivia' 'Peru' 'Honduras' 'Guyana'
 'Uganda' 'Belarus' 'El Salvador' 'Barbados' 'Trinidad and Tobago'
 'Greece' 'Cuba' 'Curaçao' 'Dominica' 'Silesia' 'Guadeloupe' 'Israel'
 'Suriname' 'French Guiana' 'Panama' 'Colombia' 'Venezuela' 'Ecuador'
 'Saint Kitts and Nevis' 'Slovakia' 'Manchukuo' 'Croatia' 'Nicaragua'
 'Afghanistan' 'India' 'Martinique' 'Zimbabwe' 'Iceland' 'Albania'
 'Madagascar' 'Zambia' 'Mauritius' 'Tanzania' 'Iran' 'Djibouti' 'DR
Congo'
 'Vietnam' 'Macau' 'Ethiopia' 'Puerto Rico' 'Réunion' 'Sierra Leone'
 'Zanzibar' 'South Korea' 'Ghana' 'South Africa' 'New Caledonia'
'Fiji'
 'Nigeria' 'Myanmar' 'Sri Lanka' 'Tahiti' 'Gambia' 'Hong Kong'
'Singapore'
 'Malaysia' 'Indonesia' 'Guinea-Bissau' 'German DR' 'Vanuatu' 'Kernow'
 'Saarland' 'Taiwan' 'Burma' 'Cambodia' 'Lebanon' 'Pakistan'
 'Vietnam Republic' 'North Korea' 'Togo' 'Sudan' 'Malta' 'Syria'
'Tunisia'
 'Malawi' 'Morocco' 'Malaya' 'Benin' 'Thailand' 'Cameroon'
 'Central African Republic' 'Gabon' 'Ivory Coast' 'Burkina Faso'
'Congo'
 'Mali' 'North Vietnam' 'Mongolia' 'Cyprus' 'Iraq' 'Saint Lucia'
'Grenada'
 'Senegal' 'Chad' 'Libya' 'Guinea' 'Algeria' 'Kuwait' 'Jordan'
'Liberia'
 'Solomon Islands' 'Laos' 'Saint Vincent and the Grenadines' 'Bermuda'
 'Niger' 'Montenegro' 'Palestine' 'Bahrain' 'Papua New Guinea'
 'Mauritania' 'Saudi Arabia' 'Eswatini' 'Western Australia' 'Somalia'
 'Lesotho' 'Cook Islands' 'Qatar' 'Antigua and Barbuda' 'Faroe
Islands'
 'Bangladesh' 'Yemen' 'Oman' 'Yemen DPR' 'Burundi' 'Mozambique' 'Guam'
 'Angola' 'Dominican Republic' 'Seychelles' 'Rwanda'
 'São Tomé and Príncipe' 'Botswana' 'Northern Cyprus' 'Cape Verde'
 'Kyrgyzstan' 'Georgia' 'Azerbaijan' 'Comoros' 'Kiribati' 'Tonga'
 'Wallis Islands and Futuna' 'United Arab Emirates' 'Brunei'
 'Equatorial Guinea' 'Liechtenstein' 'Nepal' 'Greenland' 'Niue'
'Samoa'
 'American Samoa' 'Belize' 'Maldives' 'Anguilla' 'Cayman Islands'
```

```
'Palau'
 'Sint Maarten' 'Namibia' 'Åland Islands' 'Ynys Môn' 'Saint Martin'
 'San Marino' 'Slovenia' 'Shetland' 'Isle of Wight' 'Moldova'
'Ukraine'
 'Kazakhstan' 'Tajikistan' 'Uzbekistan' 'Turkmenistan' 'Armenia'
 'Czech Republic' 'Gibraltar' 'Isle of Man' 'North Macedonia'
'Montserrat'
 'Serbia' 'Canary Islands' 'Bosnia and Herzegovina' 'Andorra'
 'British Virgin Islands' 'Frøya' 'Hitra' 'United States Virgin
Islands'
 'Corsica' 'Eritrea' 'Bahamas' 'Gotland' 'Saare County' 'Rhodes'
 'Micronesia' 'Bhutan' 'Orkney' 'Monaco' 'Tuvalu' 'Sark' 'Mayotte'
 'Turks and Caicos Islands' 'Timor-Leste' 'Occitania' 'Chechnya'
 'Western Isles' 'Falkland Islands' 'Kosovo' 'Republic of St. Pauli'
 'Găgăuzia' 'Tibet' 'Sápmi' 'Northern Mariana Islands' 'Romani people'
 'Menorca' 'Provence' 'Arameans Suryoye' 'Padania' 'Iraqi Kurdistan'
 'Gozo' 'Bonaire' 'Chagos Islands' 'Sealand' 'Western Sahara' 'Raetia'
 'Darfur' 'Tamil Eelam' 'South Sudan' 'Saint Barthélemy' 'Abkhazia'
 'Saint Pierre and Miquelon' 'Artsakh' 'Madrid' 'Saugeais' 'Ellan
Vannin'
 'Vatican City' 'Somaliland' 'Franconia' 'South Ossetia' 'County of
Nice'
 'Seborga' 'Székely Land' 'Panjab' 'Felvidék' 'Luhansk PR' 'Donetsk
PR'
 'United Koreans in Japan' 'Western Armenia' 'Délvidék' 'Barawa'
'Ryūkyū'
 'Kárpátalja' 'Yorkshire' 'Matabeleland' 'Cascadia' 'Kabylia'
 'Parishes of Jersey' 'Chameria' 'Saint Helena' 'East Timor'
 'Yoruba Nation' 'Biafra' 'Mapuche' 'Aymara' 'Elba Island' 'West
Papua'
 'Ticino' 'Hmong']
away_team ['England' 'Scotland' 'Wales' 'Northern Ireland' 'Canada'
'Argentina'
 'Hungary' 'Czechoslovakia' 'Austria' 'Uruguay' 'France' 'Switzerland'
 'Alderney' 'Guernsey' 'Netherlands' 'Belgium' 'Jersey' 'Germany'
'Norway'
 'Sweden' 'Italy' 'Chile' 'Catalonia' 'Finland' 'Russia' 'China PR'
 'Luxembourg' 'Denmark' 'Brazil' 'Basque Country' 'Philippines'
 'United States' 'Estonia' 'Provence' 'Japan' 'El Salvador' 'Costa
Rica'
 'Paraguay' 'Yugoslavia' 'Poland' 'Portugal' 'Spain' 'Romania'
'Australia'
 'Central Spain' 'Mexico' 'Galicia' 'Brittany' 'Asturias' 'New
Zealand'
 'Latvia' 'Guatemala' 'Curaçao' 'Bulgaria' 'Turkey' 'Lithuania'
'Egypt'
 'Republic of Ireland' 'South Africa' 'Jamaica' 'Uganda' 'Bolivia'
'Haiti'
 'Trinidad and Tobago' 'Kenya' 'Ukraine' 'Honduras' 'Nicaragua'
```

```
'Greece'
 'Guyana' 'Peru' 'Aruba' 'Martinique' 'Barbados' 'Cuba' 'Israel'
 'Indonesia' 'Suriname' 'Saint Lucia' 'Colombia' 'Venezuela' 'Ecuador'
 'Grenada' 'India' 'Panama' 'Slovakia' 'Croatia' 'Lebanon' 'Manchukuo'
 'Puerto Rico' 'Iran' 'Guadeloupe' 'Mongolia' 'Tanzania' 'Zambia'
 'Montenegro' 'Mauritius' 'Zimbabwe' 'French Guiana' 'Réunion'
'Albania'
 'Zanzibar' 'Ethiopia' 'Madagascar' 'Dominica' 'South Korea'
'Djibouti'
 'Dominican Republic' 'Cyprus' 'Iceland' 'Nigeria' 'Syria' 'DR Congo'
 'Serbia' 'Taiwan' 'Afghanistan' 'Burma' 'Vanuatu' 'Ghana' 'Pakistan'
 'German DR' 'Sierra Leone' 'Guinea-Bissau' 'Gambia' 'Saarland'
'Tahiti'
 'Burundi' 'Sri Lanka' 'Vietnam' 'Hong Kong' 'Singapore' 'Thailand'
 'Myanmar' 'New Caledonia' 'Cambodia' 'Malaysia' 'Vietnam Republic'
 'Kyrgyzstan' 'Belarus' 'Moldova' 'North Vietnam' 'Togo' 'North Korea'
 'Sudan' 'Libya' 'Malaya' 'Malta' 'Benin' 'Cape Verde' 'Mali'
 'Burkina Faso' 'Gabon' 'Central African Republic' 'Ivory Coast'
 'Cameroon' 'Congo' 'Iraq' 'Tunisia' 'Morocco'
 'Saint Vincent and the Grenadines' 'Laos' 'Mauritania' 'Liberia'
'Niger'
 'Senegal' 'Guinea' 'Malawi' 'Jordan' 'Kuwait' 'Chad' 'Papua New
Guinea'
 'Solomon Islands' 'Somalia' 'Saudi Arabia' 'Algeria' 'Bermuda'
 'Palestine' 'Yemen' 'Bahrain' 'Oman' 'Wallis Islands and Futuna'
 'Corsica' 'Western Australia' 'Botswana' 'Fiji' 'Eswatini' 'Qatar'
 'Lesotho' 'Bahamas' 'Brunei' 'Cook Islands' 'Yemen DPR'
 'United Arab Emirates' 'Faroe Islands' 'Nepal' 'Antigua and Barbuda'
 'Bangladesh' 'Seychelles' 'Equatorial Guinea' 'Mozambique' 'Angola'
 'São Tomé and Príncipe' 'Macau' 'Saint Kitts and Nevis' 'Armenia'
 'Kazakhstan' 'Comoros' 'Maldives' 'Tonga' 'Kiribati' 'Tuvalu' 'Samoa'
 'Greenland' 'Northern Cyprus' 'Rwanda' 'Liechtenstein' 'Western
Samoa'
 'Bhutan' 'South Yemen' 'American Samoa' 'Cayman Islands'
 'British Virgin Islands' 'Sint Maarten' 'Ynys Môn' 'Shetland'
 'Åland Islands' 'Saint Martin' 'San Marino' 'Montserrat' 'Anguilla'
 'Isle of Wight' 'Georgia' 'Guam' 'Turkmenistan' 'Slovenia'
'Uzbekistan'
 'Eritrea' 'Tajikistan' 'Namibia' 'Azerbaijan' 'Czech Republic'
 'Isle of Man' 'Gibraltar' 'North Macedonia' 'Belize'
 'Bosnia and Herzegovina' 'Andorra' 'Frøya' 'Hitra'
 'Northern Mariana Islands' 'Kernow' 'Turks and Caicos Islands'
 'United States Virgin Islands' 'Micronesia' 'Saare County' 'Rhodes'
 'Monaco' 'Tibet' 'Orkney' 'Falkland Islands' 'Mayotte' 'Vatican City'
 'Timor-Leste' 'Gotland' 'Sark' 'Sealand' 'Chechnya' 'Occitania'
 'Ambazonia' 'Western Isles' 'Crimea' 'Sápmi' 'Menorca' 'Iraqi
Kurdistan'
 'Padania' 'Arameans Suryoye' 'Gozo' 'Two Sicilies' 'Saint Barthélemy'
 'Saint Pierre and Miquelon' 'Bonaire' 'Romani people' 'Raetia'
```

```
'Cilento'
 'Chagos Islands' 'Western Sahara' 'Darfur' 'Tamil Eelam' 'Artsakh'
 'Abkhazia' 'South Sudan' 'Andalusia' 'South Ossetia' 'County of Nice'
 'Ellan Vannin' 'Luhansk PR' 'Donetsk PR' 'Felvidék' 'Panjab'
'Franconia'
 'Székely Land' 'Somaliland' 'Western Armenia' 'Kosovo' 'Kárpátalja'
 'United Koreans in Japan' 'Yorkshire' 'Surrey' 'Cascadia'
'Matabeleland'
 'Kabylia' 'Barawa' 'Parishes of Jersey' 'Chameria' 'Saint Helena'
 'East Timor' 'Biafra' 'Maule Sur' 'Aymara' 'Hmong' 'Ticino']
home_score [ 0.  4.  2.  3.  1.  7.  9.  5.  6.  8. 11. 13. 12. 10.
14. 15. 21. 30.
 24. 18. 20. 16. 19. 17. 22. 31. nan]
away_score [ 0.  2.  1.  3.  4.  6.  5. 13.  8.  7. 10.  9. 11. 12.
15. 18. 16. 14.
 19. 20. 17. 21. nan]
tournament ['Friendly' 'British Home Championship' 'Évence Coppée
Trophy'
 'Muratti Vase' 'Copa Lipton' 'Copa Newton' 'Copa Premio Honor
Argentino'
 'Copa Premio Honor Uruguayo' 'Far Eastern Championship Games' 'Copa
Roca'
 'Copa América' 'Peace Cup' 'Open International Championship'
 'Copa Chevallier Boutell' 'Olympic Games' 'Nordic Championship'
 'Central European International Cup' 'Baltic Cup' 'Balkan Cup'
 'Central American and Caribbean Games' 'FIFA World Cup' 'Copa Rio
Branco'
 'FIFA World Cup qualification' 'Bolivarian Games' 'CCCF Championship'
 'NAFC Championship' 'Copa Oswaldo Cruz' 'Asian Games'
 'Pan American Championship' 'Copa del Pacífico' "Copa Bernardo
O'Higgins"
 'AFC Asian Cup qualification' 'Atlantic Cup' 'AFC Asian Cup'
 'African Cup of Nations' 'Copa Paz del Chaco' 'Merdeka Tournament'
 'UEFA Euro qualification' 'Southeast Asian Peninsular Games'
 'African Friendship Games' 'UEFA Euro' 'Windward Islands Tournament'
 'African Cup of Nations qualification' 'Vietnam Independence Cup'
 'Copa Carlos Dittborn' 'Phillip Seaga Cup' 'CONCACAF Championship'
 'Copa Juan Pinto Durán' 'Arab Cup' 'South Pacific Games'
 'CONCACAF Championship qualification' 'Copa Artigas' 'All-African
Games'
 'GaNEFo' "King's Cup" 'Gulf Cup' 'Indonesia Tournament' 'Korea Cup'
 'Palestine Cup' 'Brazil Independence Cup' 'Copa Ramón Castilla'
 'Oceania Nations Cup' 'CECAFA Cup' 'Kuneitra Cup' 'Copa Félix Bogado'
 'Real Madrid 75th Anniversary Cup'
 'Beijing International Friendship Tournament' 'Southeast Asian Games'
 'Kirin Cup' 'CFU Caribbean Cup qualification' 'CFU Caribbean Cup'
 'Amílcar Cabral Cup' 'FIFA 75th Anniversary Cup'
 'Indian Ocean Island Games'
 'Guangzhou International Friendship Tournament' 'Mundialito'
```

```
 'South Pacific Mini Games' 'West African Cup' 'Nehru Cup' 'Merlion
Cup'
 'Great Wall Cup' 'South Asian Games' 'UDEAC Cup' 'Rous Cup'
 'CONMEBOL–UEFA Cup of Champions' 'Miami Cup' 'Lunar New Year Cup'
 'Arab Cup qualification' 'Tournoi de France'
 'Malta International Tournament' 'Four Nations Tournament' 'Matthews
Cup'
 'Tournament Burkina Faso' 'Marlboro Cup' 'Island Games'
 'NAFU Championship' 'Dynasty Cup' 'Dakar Tournament' 'UNCAF Cup'
 'Scania 100 Tournament' 'Gold Cup' 'USA Cup'
 'Jordan International Tournament' 'Confederations Cup' 'East Asian
Games'
 'United Arab Emirates Friendship Tournament' 'Joe Robbie Cup'
 'Oceania Nations Cup qualification' 'Simba Tournament' 'SAFF Cup'
 'AFF Championship' 'King Hassan II Tournament'
 'Cyprus International Tournament' 'Dunhill Cup'
 'COSAFA Cup qualification' 'COSAFA Cup' 'Gold Cup qualification'
 'AFF Championship qualification' 'SKN Football Festival' 'UNIFFAC
Cup'
 'WAFF Championship' 'Millennium Cup' 'Cup of Ancient Civilizations'
 "Prime Minister's Cup" 'The Other Final' 'EAFF Championship'
 'TIFOCO Tournament' 'Afro-Asian Games' 'AFC Challenge Cup'
 'FIFI Wild Cup' 'ELF Cup' 'Viva World Cup'
 'AFC Challenge Cup qualification' "Coupe de l'Outre-Mer" 'VFF Cup'
 'Corsica Cup' 'Dragon Cup' 'ABCS Tournament' 'Nile Basin Tournament'
 'Nations Cup' 'Copa Confraternidad' 'Pacific Games'
 'Superclásico de las Américas' 'Africa Cup of Nations qualification'
 'Kirin Challenge Cup' 'Tynwald Hill Tournament' 'OSN Cup'
 'CONIFA World Football Cup' 'Niamh Challenge Cup'
 'CONIFA European Football Cup' 'Benedikt Fontana Cup'
 'Copa América qualification' 'ConIFA Challenger Cup'
 'Hungary Heritage Cup' 'World Unity Cup' 'Pacific Mini Games'
 'Intercontinental Cup' 'UEFA Nations League'
 'CONCACAF Nations League qualification' 'Atlantic Heritage Cup'
 'Inter Games Football Tournament' 'CONCACAF Nations League'
 'Three Nations Cup' 'Mahinda Rajapaksa Cup' 'Navruz Cup'
 'CONIFA Africa Football Cup' 'CONIFA South America Football Cup'
 "MSG Prime Minister's Cup" 'Tri Nation Tournament' 'CAFA Nations Cup'
 'Mauritius Four Nations Cup' 'CONIFA World Football Cup
qualification'
 'CONIFA Asia Cup' 'FIFA Series' 'Marianas Cup']
city ['Glasgow' 'London' 'Wrexham' ... 'College Station' 'Mallorca'
 'Hradec Králové']
country ['Scotland' 'England' 'Wales' 'Ireland' 'United States'
'Uruguay'
 'Austria' 'Hungary' 'Argentina' 'Belgium' 'France' 'Guernsey'
'Jersey'
 'Netherlands' 'Bohemia' 'Switzerland' 'Sweden' 'Germany' 'Italy'
'Chile'
```

```
 'Norway' 'Finland' 'Luxembourg' 'Russia' 'Denmark' 'Spain'
'Philippines'
 'China PR' 'Japan' 'Brazil' 'Paraguay' 'Canada' 'Estonia' 'Guatemala'
 'Czechoslovakia' 'Poland' 'Yugoslavia' 'New Zealand' 'Romania'
'Latvia'
 'Portugal' 'Northern Ireland' 'Australia' 'Lithuania' 'Turkey'
'Mexico'
 'Aruba' 'Soviet Union' 'Haiti' 'Bulgaria' 'Jamaica' 'Kenya'
 'Czech Republic' 'Peru' 'Honduras' 'British Guyana' 'Uganda'
 'El Salvador' 'Barbados' 'Irish Free State' 'Trinidad and Tobago'
 'Greece' 'Cuba' 'Curaçao' 'Egypt' 'Dominica' 'Guadeloupe' 'Palestine'
 'Netherlands Guyana' 'French Guiana' 'Panama' 'Colombia'
 'Saint Kitts and Nevis' 'Éire' 'Bohemia and Moravia' 'Slovakia'
 'Manchuria' 'Croatia' 'Costa Rica' 'Afghanistan' 'Martinique'
 'Southern Rhodesia' 'Iceland' 'Albania' 'Madagascar' 'Northern
Rhodesia'
 'Tanganyika' 'Iran' 'Ecuador' 'French Somaliland' 'Belgian Congo'
 'Mauritius' 'Hong Kong' 'Vietnam' 'Macau' 'Republic of Ireland'
 'Ethiopia' 'Suriname' 'Puerto Rico' 'Réunion' 'Israel' 'Sierra Leone'
 'Zanzibar' 'Bolivia' 'Gold Coast' 'South Africa' 'Netherlands
Antilles'
 'India' 'New Caledonia' 'Fiji' 'Nigeria' 'Venezuela' 'Ceylon'
 'French Polynesia' 'Gambia' 'Singapore' 'Portuguese Guinea' 'German
DR'
 'New Hebrides' 'Burma' 'Saarland' 'Cambodia' 'Lebanon' 'Pakistan'
 'Malaya' 'South Korea' 'Vietnam Republic' 'Togo' 'Indonesia' 'Sudan'
 'Malta' 'Syria' 'Tunisia' 'Nyasaland' 'Ghana' 'Morocco'
 'United Arab Republic' 'North Korea' 'Dahomey' 'Thailand' 'Guinea-
Bissau'
 'Mali Federation' 'Mali' 'Vietnam DR' 'Cyprus' 'Iraq' 'Saint Lucia'
 'Senegal' 'Ivory Coast' 'Libya' 'Gabon' 'Congo' 'Tanzania' 'Grenada'
 'Guinea' 'Central African Republic' 'Cameroon' 'Algeria' 'Kuwait'
 'Liberia' 'Malaysia' 'Jordan' 'Zambia' 'Saint Vincent and the
Grenadines'
 'Bermuda' 'Niger' 'Malawi' 'DR Congo' 'Upper Volta' 'Taiwan' 'Guyana'
 'Mauritania' 'Rhodesia' 'Saudi Arabia' 'Swaziland' 'Mozambique'
 'Papua New Guinea' 'Bahrain' 'Lesotho' 'Somalia' 'Zaïre' 'Sri Lanka'
 'Antigua and Barbuda' 'Faroe Islands' 'Qatar' 'Yemen DPR' 'Burundi'
 'Guam' 'Chad' 'Angola' 'Alderney' 'Dominican Republic' 'Seychelles'
 'São Tomé and Príncipe' 'Botswana' 'Benin' 'Rwanda' 'Bangladesh'
 'United Arab Emirates' 'Zimbabwe' 'Oman' 'Equatorial Guinea'
 'Solomon Islands' 'Cape Verde' 'Liechtenstein' 'Nepal' 'Greenland'
 'Vanuatu' 'Western Samoa' 'Belize' 'Brunei' 'Djibouti' 'Burkina Faso'
 'Yemen AR' 'Anguilla' 'Nicaragua' 'Cayman Islands' 'Monaco'
 'Sint Maarten' 'Namibia' 'Saint Martin' 'San Marino' 'Slovenia'
'Moldova'
 'Ukraine' 'Kazakhstan' 'Tajikistan' 'Uzbekistan' 'Turkmenistan'
'Georgia'
 'Kyrgyzstan' 'Armenia' 'Belarus' 'Azerbaijan' 'North Macedonia'
```

```
 'Montserrat' 'Gibraltar' 'Myanmar' 'Bosnia and Herzegovina' 'Tonga'
 'Andorra' 'Yemen' 'United States Virgin Islands' 'Palau' 'Cook
Islands'
 'British Virgin Islands' 'Eritrea' 'Bahamas' 'Micronesia' 'Maldives'
 'Laos' 'Isle of Man' 'Samoa' 'Bhutan' 'Serbia and Montenegro'
'Mayotte'
 'Mongolia' 'Northern Cyprus' 'Serbia' 'Montenegro'
 'Northern Mariana Islands' 'Comoros' 'Turks and Caicos Islands'
 'South Sudan' 'Saint Barthélemy' 'Kosovo' 'East Timor' 'Tahiti'
 'Eswatini' 'Bonaire']
neutral [False  True]
result ['Draw' 'Home Win' 'Away Win']
home_team_encoded [250  88 310 204 301 303  17 129  12  27  99 119 143
193  73   3 275 274
 108 139  59 206  98 163 232  76  51 220  25  60  37 142 217  47  91
66
 118 270  39 221 322 195 231 155 104 222   6  16 161 293  54 181  15
84
 227 124  41 141 147  33 219 127 122 297  26  85  24 291 113  69  70
78
 258 116 138 273 101 214  61 307  83 240 261 173  68 196   1 131 175
325
 130   2 165 323 178 284 133  77  74 308 164  93 224 235 257 324 267
109
 266 194  97 198 190 271 280 105 128 259 169 132 121 107 305 148 237
281
  43  45 156 211 309 200 289 272 172 276 292 167 188 168  29 285  46
53
 103 140  42  63 171 202 185  71 134 241 115 253  55 159 120   4 151
144
 158 263 154 244  30 197 186 213  21 216 177 248  92 314 264 157  64
225
  10  95  22 317 208 318  44 189 117   8  79 255 233 279  36 203  49
152
 106  19  62 149 290 311 299  40  89 160 192 114 199 245   5  28 170
9
  52 212 260 191 326 319 242 246 262 256 137 183 298 146 282 304 294
13
  72 110 136 201 187 254  48  35   7  38 102 125 302  65  90  20 111
236
 229 182  31 209 184 296 247 179 295 288 207  58 315  94 150 228 123
286
 278 205 230 180 223  11 210 135 112  34  56 251 316 226  75 283 269
238
   0 243  14 166 249  87 306 265 100 268  67 252 277 215  96 162  80
300
 313  81  23 234 153 320 176  50 145 218  57 239  82 321  32 174  18
86
 312 287 126]
```

```
away_team_encoded [ 89 244 305 202  49  13 129  76  19 298 100 269   3
120 192  29 143 109
 204 268 139  60  52  99 228  61 163  79  39  27 217 296  92 220 142
87
  68 214 316 218 219 263 227  18  55 180 105  41  17 194 155 119  73
43
 287 161  86 224 258 141 292  35 124 285 147 293 127 195 114 123 216
16
 173  26  72 138 132 266 236  63 302  85 116 131 211 253  71 156 172
221
 133 117 184 278 317 185 177 319 102 230   2 318  94 165  81 259  80
82
  74 130 197 270  77 247 275   1  45 300 110 209 108 250 122 106 232
274
  46 264 303 128 251 279 189 193  47 168 304 152  28 182 200 283 198
265
 159 167 171  31  50 170  44 104  54 140  48  65 134 286 187 239 154
176
 158 196 246 121 166 144 151  56 213 255 256 243   4  32 210 312  23
206
 306  67 308  38  98  93 222 157  22  42  66 313 294  96 191  11  24
248
  90 188   9 273 164 235  14 146  64 169 284 149 290 240 115 201 229
160
 311  33 262   6  53  40 252 314 249 320 237 241 186  10 137 107 118
288
 254 299  91 276 190  21  75 136 111 199  30  37   8 103 125 203 148
289
 297 181 231 225 183 280 207  95 178 301 282 112 242 245  59 205   5
309
  70 272 179 135 208  12 113 291 233 238  36 226 223  62  57 310  78
277
  15   0 261   7 260  69  88 162  83  97 212 101 271 257 307 150 153
295
 315 267  51 174 145  25 215  58 234  84  34 175  20 126 281]

# Encode categorical columns
res['home_team_encoded'] =
res['home_team'].astype('category').cat.codes
res['away_team_encoded'] =
res['away_team'].astype('category').cat.codes

# Check for any other non-numeric columns
for column in res.columns:
    if res[column].dtype == 'object':
        print(f"Encoding column: {column}")
        res[f"{column}_encoded"] =
res[column].astype('category').cat.codes
```

```
Encoding column: home_team
Encoding column: away_team
Encoding column: tournament
Encoding column: city
Encoding column: country
Encoding column: result
```

```
# Select only numeric columns for correlation calculation
numeric_columns = res.select_dtypes(include=[np.number]).columns
res_numeric = res[numeric_columns]

# Display the first few rows of the numeric data
print(res_numeric.head())
```

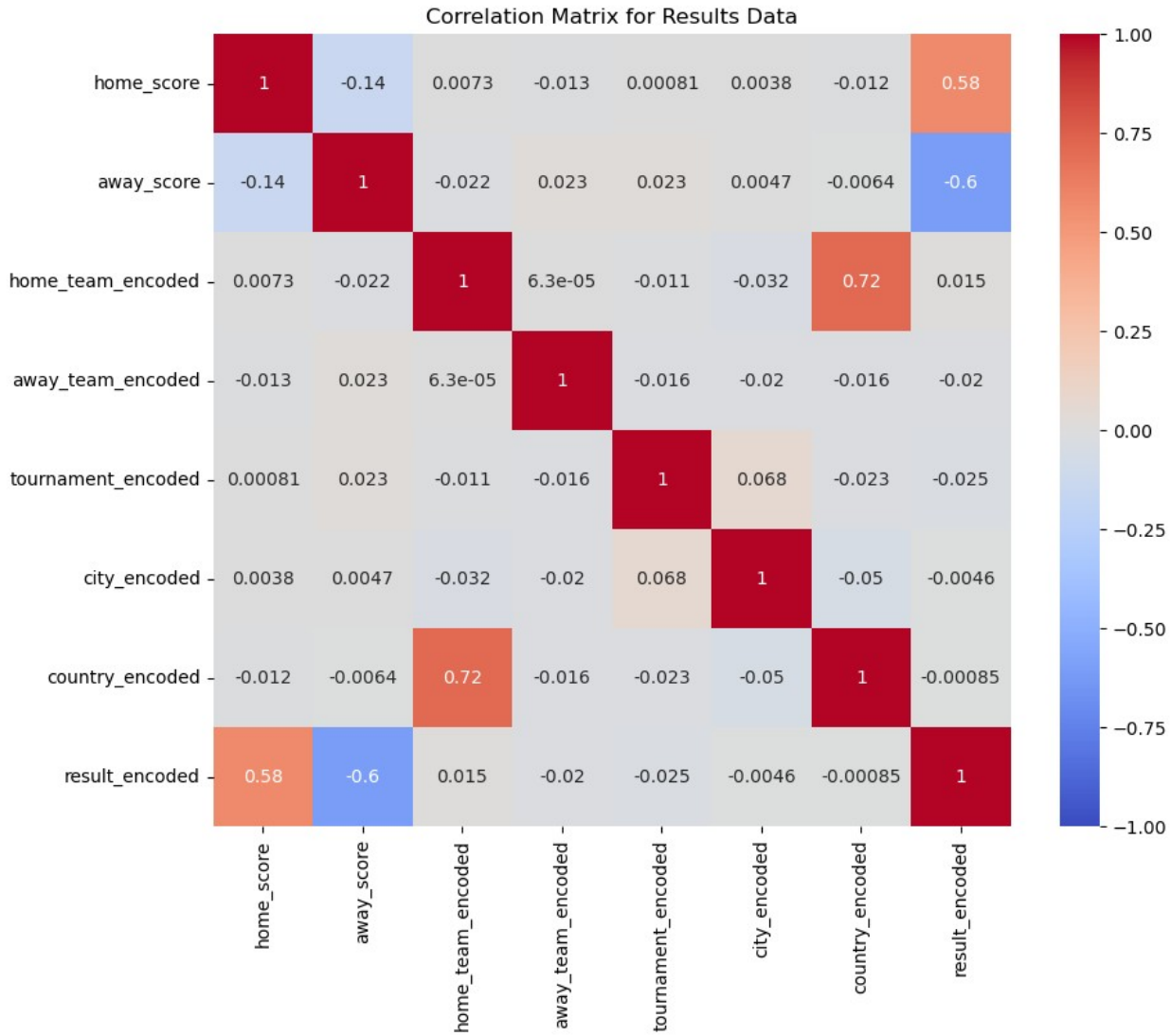
```
   home_score  away_score  home_team_encoded  away_team_encoded  \
0         0.0         0.0                250                 89
1         4.0         2.0                 88                244
2         2.0         1.0                250                 89
3         2.0         2.0                 88                244
4         3.0         0.0                250                 89

   tournament_encoded  city_encoded  country_encoded  result_encoded
0                  85           649              206               1
1                  85          1027               70               2
2                  85           649              206               2
3                  85          1027               70               1
4                  85           649              206               2
```

```
# Compute the correlation matrix
correlation_matrix_results = res_numeric.corr()

# Print the correlation matrix
print(correlation_matrix_results)
```

```
                    home_score  away_score  home_team_encoded  \
home_score            1.000000   -0.143877           0.007336
away_score           -0.143877    1.000000          -0.021917
home_team_encoded     0.007336   -0.021917           1.000000
away_team_encoded    -0.013307    0.022851           0.000063
tournament_encoded    0.000812    0.023363          -0.011245
city_encoded          0.003849    0.004691          -0.032163
country_encoded      -0.012446   -0.006423           0.715150
result_encoded        0.576361   -0.604294           0.015319

                    away_team_encoded  tournament_encoded
city_encoded  \
home_score                  -0.013307            0.000812
0.003849
away_score                   0.022851            0.023363
0.004691
home_team_encoded            0.000063           -0.011245     -
```

```
0.032163
away_team_encoded             1.000000            -0.016294     -
0.019955
tournament_encoded           -0.016294             1.000000
0.067712
city_encoded                 -0.019955             0.067712
1.000000
country_encoded              -0.015802            -0.022858     -
0.050132
result_encoded               -0.020172            -0.025399     -
0.004568

                     country_encoded  result_encoded
home_score                 -0.012446        0.576361
away_score                 -0.006423       -0.604294
home_team_encoded           0.715150        0.015319
away_team_encoded          -0.015802       -0.020172
tournament_encoded         -0.022858       -0.025399
city_encoded               -0.050132       -0.004568
country_encoded             1.000000       -0.000848
result_encoded             -0.000848        1.000000
```

```
# Visualize the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix_results, annot=True, cmap='coolwarm',
vmin=-1, vmax=1)
plt.title('Correlation Matrix for Results Data')
plt.show()
```




```
# Display unique values for potential categorical columns
for column in shoot.columns:
    print(column, shoot[column].unique())
```

```
date <DatetimeArray>
['1967-08-22 00:00:00', '1971-11-14 00:00:00', '1972-05-07 00:00:00',
 '1972-05-17 00:00:00', '1972-05-19 00:00:00', '1973-04-21 00:00:00',
 '1973-06-14 00:00:00', '1973-07-26 00:00:00', '1973-07-27 00:00:00',
 '1973-07-28 00:00:00',
 ...
 '2024-01-31 00:00:00', '2024-02-03 00:00:00', '2024-02-07 00:00:00',
 '2024-03-23 00:00:00', '2024-03-26 00:00:00', '2024-06-11 00:00:00',
 '2024-07-01 00:00:00', '2024-07-04 00:00:00', '2024-07-05 00:00:00',
 '2024-07-06 00:00:00']
Length: 566, dtype: datetime64[ns]
home_team ['India' 'South Korea' 'Thailand' 'Senegal' 'Guinea'
```

```
'Mauritius'
 'Malaysia' 'Cambodia' 'Bangladesh' 'Myanmar' 'Algeria' 'Singapore'
 'Qatar' 'Hong Kong' 'Syria' 'Libya' 'Morocco' 'Kenya'
'Czechoslovakia'
 'Tunisia' 'Argentina' 'Zambia' 'Iran' 'Paraguay' 'Guinea-Bissau'
 'Cameroon' 'Seychelles' 'Indonesia' 'Gambia' 'Italy' 'Nigeria' 'Mali'
 'Rwanda' 'Ivory Coast' 'Germany' 'Uganda' 'Egypt' 'Sierra Leone'
'Iraq'
 'Denmark' 'Ghana' 'Saudi Arabia' 'Malawi' 'Chad'
 'Central African Republic' 'Congo' 'Mozambique' 'Madagascar'
'Bahrain'
 'Pakistan' 'Brazil' 'Mexico' 'Spain' 'United Arab Emirates' 'Kuwait'
 'Gabon' 'Australia' 'DR Congo' 'Equatorial Guinea' 'Ethiopia'
'Canada'
 'Chile' 'Eswatini' 'China PR' 'United States' 'Colombia' 'Zimbabwe'
 'Republic of Ireland' 'Yugoslavia' 'Tanzania' 'Fiji' 'Netherlands'
'Cuba'
 'Japan' 'Finland' 'Bolivia' 'Martinique' 'Jamaica' 'Burundi' 'Nepal'
 'Burkina Faso' 'Romania' 'Antigua and Barbuda' 'Uruguay' 'Mauritania'
 'Honduras' 'Sri Lanka' 'Barbados' 'Guyana' 'England' 'France' 'Sudan'
 'Croatia' 'Benin' 'Russia' 'Ecuador' 'Trinidad and Tobago' 'Haiti'
 'Hungary' 'Belgium' 'Réunion' 'Guatemala' 'New Zealand' 'Namibia'
 'South Africa' 'Suriname' 'British Virgin Islands' 'Maldives'
'Lesotho'
 'Togo' 'Ynys Môn' 'Guernsey' 'Cyprus' 'Botswana' 'Niger' 'Guadeloupe'
 'New Caledonia' 'Kazakhstan' 'Jordan' 'Portugal' 'Sweden' 'Angola'
 'Uzbekistan' 'Austria' 'Costa Rica' 'Jersey' 'Ukraine' 'Zanzibar'
 'Northern Cyprus' 'Switzerland' 'Provence' 'Corsica' 'French Guiana'
 'Panama' 'Saare County' 'Isle of Man' 'Saint Lucia' 'Mayotte'
'Alderney'
 'Sealand' 'Latvia' 'Abkhazia' 'Ellan Vannin' 'Padania' 'Occitania'
 'County of Nice' 'Curaçao' 'Gibraltar' 'El Salvador'
 'United Koreans in Japan' 'Iraqi Kurdistan' 'Bosnia and Herzegovina'
 'Vietnam' 'Chagos Islands' 'Papua New Guinea' 'Peru' 'Poland'
'Felvidék'
 'Kárpátalja' 'Hitra' 'Greenland' 'Andorra' 'Matabeleland' 'Tibet'
 'Panjab' 'Oman' 'South Ossetia' 'Slovakia' 'Scotland' 'Serbia'
 'Tajikistan' 'Lithuania' 'Saint Kitts and Nevis' 'Orkney' 'Åland'
 'Solomon Islands' 'Cape Verde' 'Wales' 'Georgia'
 'Turks and Caicos Islands' 'Venezuela']
away_team ['Taiwan' 'Vietnam Republic' 'Iraq' 'South Korea' 'Cambodia'
'Ghana'
 'Mali' 'Tanzania' 'Kuwait' 'Singapore' 'Thailand' 'Syria'
 'United Arab Emirates' 'Indonesia' 'Morocco' 'Tunisia' 'Malaysia'
 'Malawi' 'Germany' 'Iran' 'Algeria' 'Argentina' 'Guinea'
'Netherlands'
 'Mauritius' 'Mauritania' 'Egypt' 'Czechoslovakia' 'China PR'
'Ethiopia'
 'Togo' 'Burkina Faso' 'France' 'Kenya' 'Niger' 'Sierra Leone' 'Congo'
```

```
 'Nigeria' 'Senegal' 'Cameroon' 'Qatar' 'Spain' 'Angola' 'Somalia'
 'Zambia' 'Equatorial Guinea' 'Gabon' 'Réunion' 'Libya' 'Uganda'
'Nepal'
 'India' 'Belgium' 'Australia' 'Madagascar' 'Zimbabwe' 'Sweden'
'Greece'
 'Ecuador' 'Saudi Arabia' 'Sudan' 'Colombia' 'Russia' 'Eswatini'
'Romania'
 'England' 'North Korea' 'Ivory Coast' 'Honduras' 'Yugoslavia'
 'Solomon Islands' 'Denmark' 'Martinique' 'Switzerland' 'Estonia'
 'Paraguay' 'Saint Kitts and Nevis' 'Myanmar' 'Uruguay' 'Canada'
 'Bulgaria' 'Italy' 'Mexico' 'Bangladesh' 'Bolivia' 'Brazil'
'Zanzibar'
 'Costa Rica' 'Sri Lanka' 'Jamaica' 'Suriname' 'Czech Republic'
'Croatia'
 'Serbia' 'Cuba' 'Chile' 'DR Congo' 'North Macedonia' 'Namibia'
 'New Zealand' 'Poland' 'Peru' 'South Africa' 'Burundi' 'Mozambique'
 'Georgia' 'Saint Lucia' 'Sint Maarten' 'Antigua and Barbuda' 'Japan'
 'Chad' 'Lesotho' 'Rwanda' 'Jersey' 'Ynys Môn' 'Trinidad and Tobago'
 'Haiti' 'Slovenia' 'Belarus' 'Republic of Ireland' 'Maldives'
 'French Guiana' 'Vanuatu' 'Armenia' 'Botswana' 'Bahrain' 'Jordan'
 'Hong Kong' 'Slovakia' 'Latvia' 'Oman' 'Austria' 'Guernsey' 'Panama'
 'Israel' 'Kazakhstan' 'Ukraine' 'Portugal' 'Seychelles' 'Mayotte'
 'Turkey' 'Sápmi' 'New Caledonia' 'Tahiti' 'El Salvador' 'Åland
Islands'
 'Menorca' 'Aruba' 'Venezuela' 'Sealand' 'Chagos Islands' 'South
Ossetia'
 'Iraqi Kurdistan' 'Abkhazia' 'Ellan Vannin' 'Felvidék' 'Guatemala'
 'Basque Country' 'Northern Cyprus' 'Padania' 'Western Armenia'
'Raetia'
 'Panjab' 'Kárpátalja' 'Western Isles' 'Cape Verde' 'Kabylia'
 'United Koreans in Japan' 'Cascadia' 'Székely Land' 'Vietnam'
 'Uzbekistan' 'Chameria' 'Benin' 'Liberia' 'Djibouti' 'Northern
Ireland'
 'Scotland' 'Guadeloupe' 'Moldova' 'Tajikistan' 'Iceland' 'Grenada'
 'Puerto Rico' 'Lebanon' 'Greenland' 'Falkland Islands' 'Anguilla'
 'United States Virgin Islands']
winner ['Taiwan' 'South Korea' 'Iraq' 'Thailand' 'Ghana' 'Guinea'
'Mauritius'
 'Malaysia' 'Singapore' 'Myanmar' 'Vietnam Republic' 'Syria' 'Algeria'
 'Qatar' 'Indonesia' 'Morocco' 'Tunisia' 'Kenya' 'Czechoslovakia'
 'Argentina' 'Zambia' 'Iran' 'Paraguay' 'Mali' 'Seychelles' 'Gambia'
 'Nigeria' 'China PR' 'Ethiopia' 'Togo' 'Burkina Faso' 'Germany'
'Senegal'
 'Egypt' 'Ivory Coast' 'Sierra Leone' 'Cameroon' 'Spain' 'Angola'
 'Saudi Arabia' 'Kuwait' 'Chad' 'Gabon' 'Mozambique' 'Bahrain' 'Nepal'
 'India' 'France' 'Belgium' 'Australia' 'DR Congo' 'Malawi' 'Zimbabwe'
 'Sweden' 'Canada' 'Ecuador' 'Eswatini' 'Uganda' 'Colombia'
 'Republic of Ireland' 'North Korea' 'Central African Republic'
'Italy'
```

```
 'United States' 'Fiji' 'Denmark' 'Martinique' 'Japan' 'Switzerland'
 'Finland' 'Bolivia' 'Bulgaria' 'Brazil' 'Tanzania' 'Mexico'
 'Antigua and Barbuda' 'Uruguay' 'Solomon Islands' 'Zanzibar'
'Honduras'
 'Bangladesh' 'Sri Lanka' 'Jamaica' 'Suriname' 'England' 'Czech
Republic'
 'Sudan' 'Croatia' 'Benin' 'Russia' 'Trinidad and Tobago' 'Haiti'
'Cuba'
 'Hungary' 'Réunion' 'Guatemala' 'United Arab Emirates' 'Namibia'
'Poland'
 'Rwanda' 'Romania' 'South Africa' 'British Virgin Islands' 'Barbados'
 'Lesotho' 'Libya' 'Ynys Môn' 'Guernsey' 'Cyprus' 'Slovenia' 'Belarus'
 'Serbia' 'Niger' 'Guadeloupe' 'New Caledonia' 'Kazakhstan' 'Botswana'
 'Portugal' 'Netherlands' 'Hong Kong' 'Slovakia' 'Oman' 'Latvia'
 'Costa Rica' 'Panama' 'Israel' 'Northern Cyprus' 'Ukraine' 'Jersey'
 'Mayotte' 'Turkey' 'Sápmi' 'Corsica' 'Tahiti' 'Åland Islands'
'Menorca'
 'Saint Lucia' 'Sealand' 'Chagos Islands' 'South Ossetia' 'Ellan
Vannin'
 'Padania' 'Iraqi Kurdistan' 'County of Nice' 'Aruba' 'Chile'
 'Basque Country' 'Bosnia and Herzegovina' 'Vietnam' 'Western Armenia'
 'Raetia' 'United Koreans in Japan' 'Abkhazia' 'New Zealand'
'Felvidék'
 'Kárpátalja' 'Western Isles' 'Greenland' 'Cape Verde' 'Madagascar'
 'Kabylia' 'Panjab' 'Peru' 'Northern Ireland' 'Scotland'
 'Equatorial Guinea' 'Tajikistan' 'Iceland' 'Saint Kitts and Nevis'
 'Guyana' 'Puerto Rico' 'Åland' 'Georgia' 'Anguilla' 'Estonia']
first_shooter [nan 'Czechoslovakia' 'Argentina' 'Italy' 'France'
'Denmark' 'Brazil'
 'Germany' 'Spain' 'Sweden' 'South Korea' 'Colombia' 'Romania'
'England'
 'Russia' 'United States' 'Ivory Coast' 'Netherlands' 'Australia'
 'Nigeria' 'Mexico' 'Uruguay' 'Tunisia' 'Iran' 'Kuwait' 'Saudi Arabia'
 'Ecuador' 'Burkina Faso' 'DR Congo' 'Belgium' 'Morocco' 'South
Africa'
 'Canada' 'Cameroon' 'Republic of Ireland' 'Bahrain' 'Japan' 'China
PR'
 'Panama' 'Egypt' 'Ukraine' 'Portugal' 'Croatia' 'Zambia' 'Paraguay'
 'Costa Rica' 'Gabon' 'Finland' 'Ghana' 'Estonia' 'Equatorial Guinea'
 'Chile' 'United Koreans in Japan' 'New Zealand' 'Switzerland'
'Senegal'
 'Vietnam' 'Benin' 'Madagascar' 'Algeria' 'Tanzania'
 'Bosnia and Herzegovina' 'Slovakia' 'Scotland' 'French Guiana'
 'Guatemala' 'Eswatini' 'Mozambique' 'Qatar' 'Moldova' 'Lithuania'
 'Latvia' 'Iceland' 'Saint Kitts and Nevis' 'Guyana' 'Suriname'
'India'
 'Åland' 'Thailand' 'Tajikistan' 'Cape Verde' 'Uzbekistan' 'Poland'
 'Georgia' 'Turks and Caicos Islands' 'British Virgin Islands'
'Slovenia'
```

```
 'Venezuela']
winner_encoded [151 140  72 154  57  62  93  90 135  99 166 148   1
119  70  97 157  82
  38   5 170  71 114  91 133  54 106  29  48 155  21  56 131  42  76
134
  22 142   2 128  83  26  53  98   8 101  69  52  13   7  39  89 172
146
  23  41  47 159  30 121 107  25  75 163  50  40  92  78 147  51  15
20
  18 153  96   4 164 138 171  65   9 143  77 145  44  37 144  34  14
123
 156  64  35  67 125  60 161 100 116 124 122 139  19  10  86  87 169
61
  36 137  12 132 105  59 103  81  17 117 102  66 136 110  85  32 112
74
 108 160  79  94 158 149  31 150 174  95 127 130  27 141  43 111  73
33
   6  28  11  16 165 167 120 162   0 104  49  84 168  58  24  88  80
113
 115 109 129  45 152  68 126  63 118 173  55   3  46]
first_shooter_encoded [-1 18  1 39 28 20  7 32 69 71 68 15 59 23 60 80
40 50  2 52 46 81 76 38
 42 62 21  9 19  4 48 67 11 10 58  3 41 14 53 22 78 56 17 85 54 16 30
27
 33 25 24 13 79 51 72 64 84  5 45  0 74  6 65 63 29 34 26 49 57 47 44
43
 36 61 35 70 37 86 75 73 12 82 55 31 77  8 66 83]
```

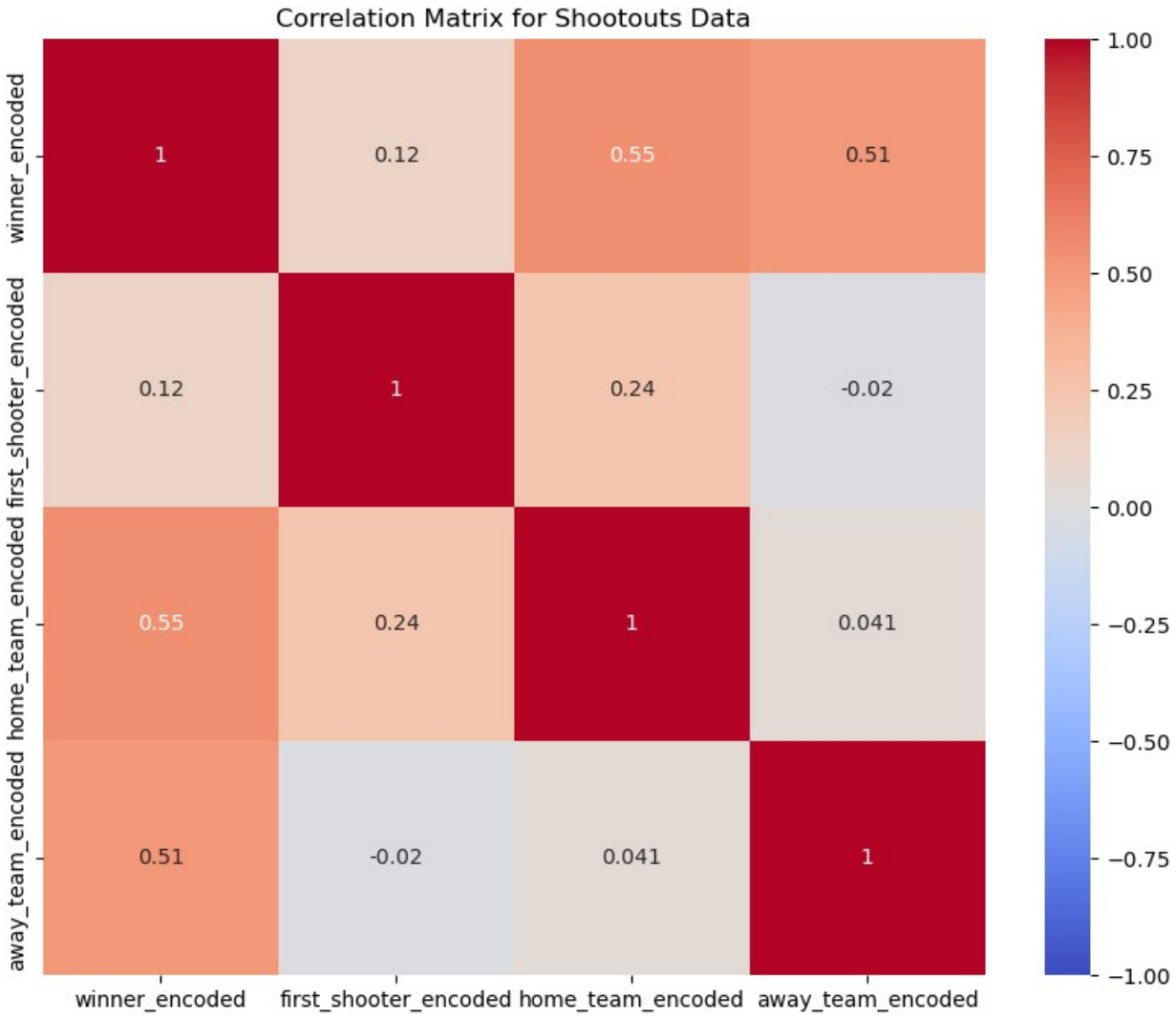
```python
# Encode categorical columns
for column in shoot.columns:
    if shoot[column].dtype == 'object':
        print(f"Encoding column: {column}")
        shoot[f"{column}_encoded"] =
shoot[column].astype('category').cat.codes
```

```
Encoding column: home_team
Encoding column: away_team
Encoding column: winner
Encoding column: first_shooter
```

```python
# Select only numeric columns for correlation calculation
numeric_columns = shoot.select_dtypes(include=[np.number]).columns
shoot_numeric = shoot[numeric_columns]

# Display the first few rows of the numeric data
print(shoot_numeric.head())
```

```
   winner_encoded  first_shooter_encoded  home_team_encoded
away_team_encoded
0             151                     -1                 73
```

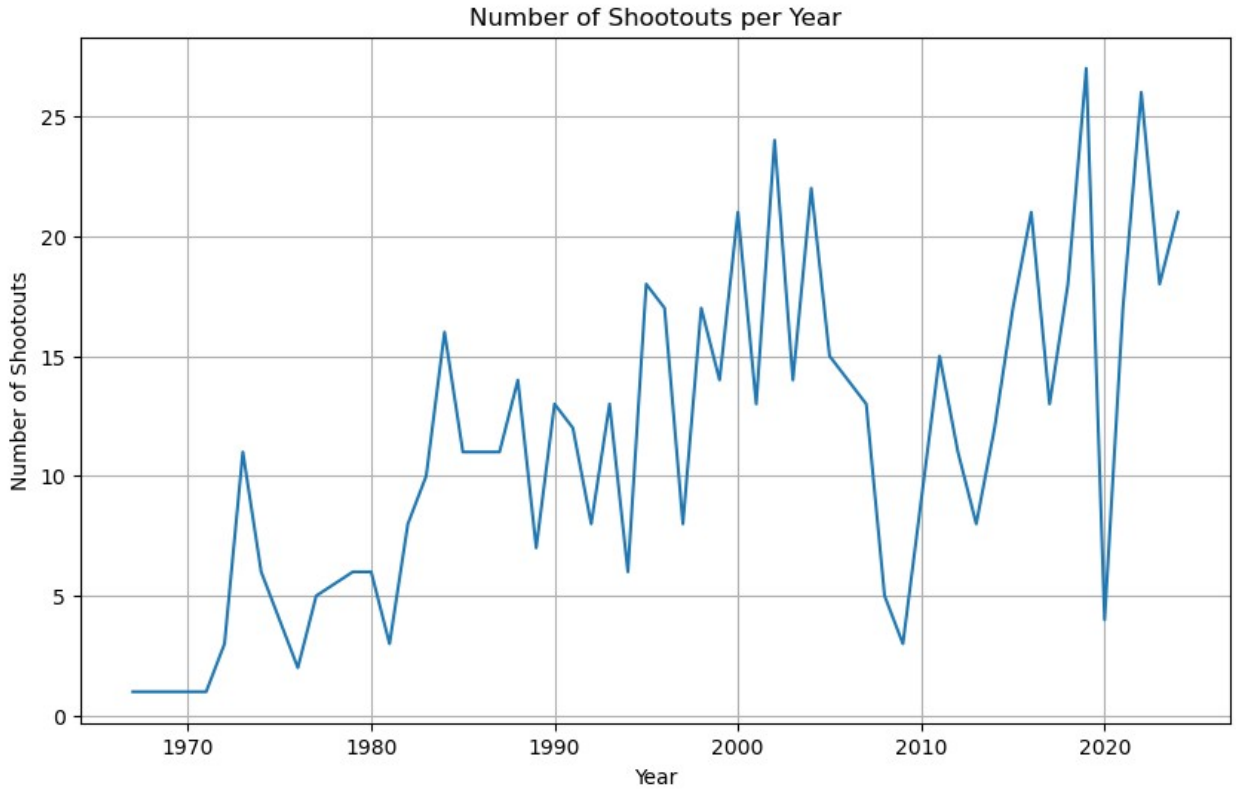
```
162
1              140                        -1                 148
180
2               72                        -1                 148
73
3              140                        -1                 159
150
4              154                        -1                 159
22

# Compute the correlation matrix
correlation_matrix_shootouts = shoot_numeric.corr()

# Print the correlation matrix
print(correlation_matrix_shootouts)

                        winner_encoded  first_shooter_encoded  \
winner_encoded                1.000000               0.123048
first_shooter_encoded         0.123048               1.000000
home_team_encoded             0.549919               0.235401
away_team_encoded             0.506958              -0.019530

                        home_team_encoded  away_team_encoded
winner_encoded                   0.549919           0.506958
first_shooter_encoded            0.235401          -0.019530
home_team_encoded                1.000000           0.041274
away_team_encoded                0.041274           1.000000

# Visualize the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix_shootouts, annot=True, cmap='coolwarm',
vmin=-1, vmax=1)
plt.title('Correlation Matrix for Shootouts Data')
plt.show()
```



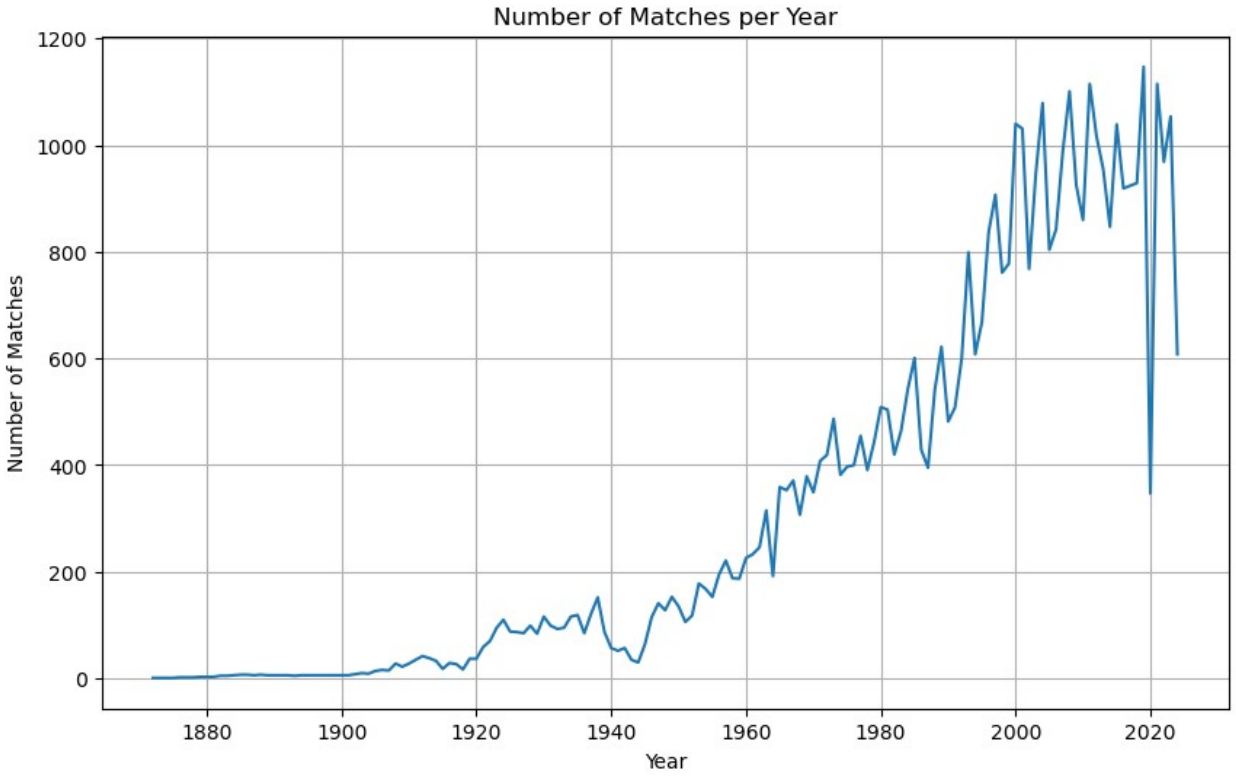
Trends Over Time

```
import pandas as pd
import matplotlib.pyplot as plt

# Convert date columns to datetime format
shoot['date'] = pd.to_datetime(shoot['date'])
res['date'] = pd.to_datetime(res['date'])
gs['date'] = pd.to_datetime(gs['date'])

# Aggregate shootouts by year
shoot['year'] = shoot['date'].dt.year
shootouts_per_year = shoot.groupby('year').size()

# Plot the trend of shootouts per year
plt.figure(figsize=(10, 6))
```

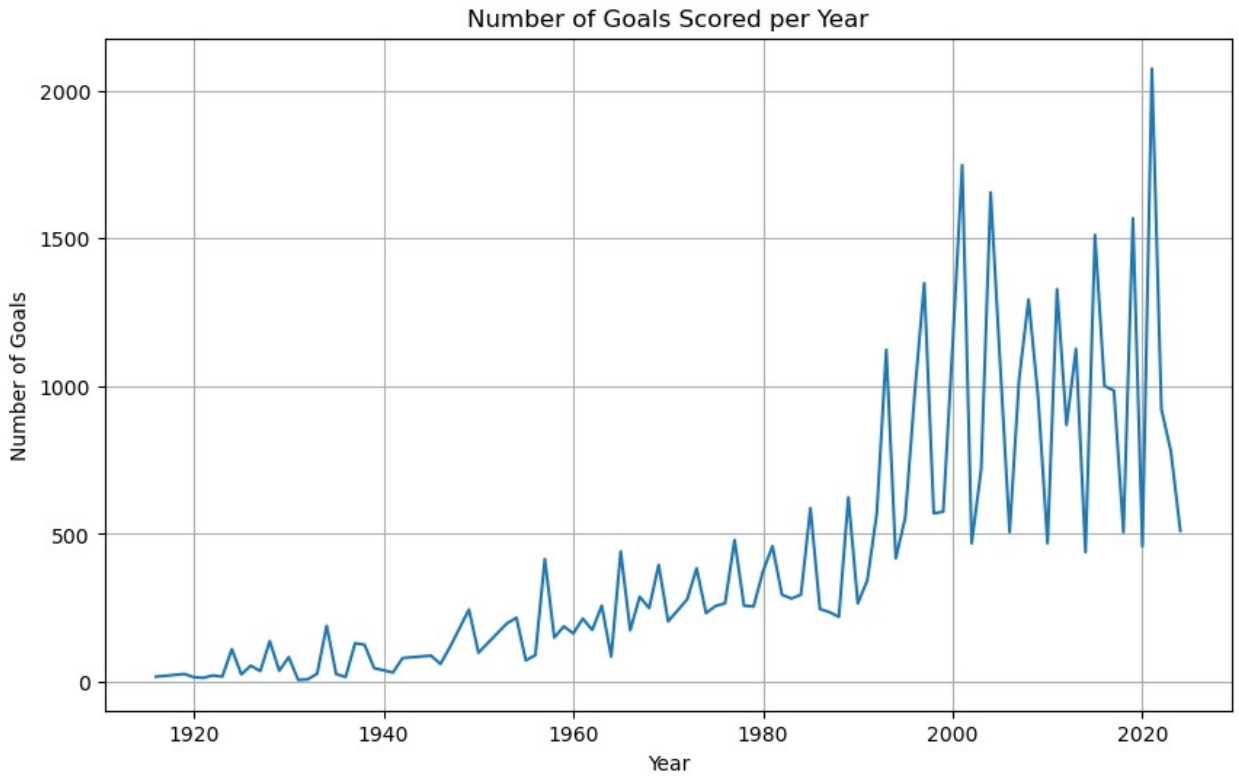
```
shootouts_per_year.plot(kind='line')
plt.title('Number of Shootouts per Year')
plt.xlabel('Year')
plt.ylabel('Number of Shootouts')
plt.grid(True)
plt.show()
```



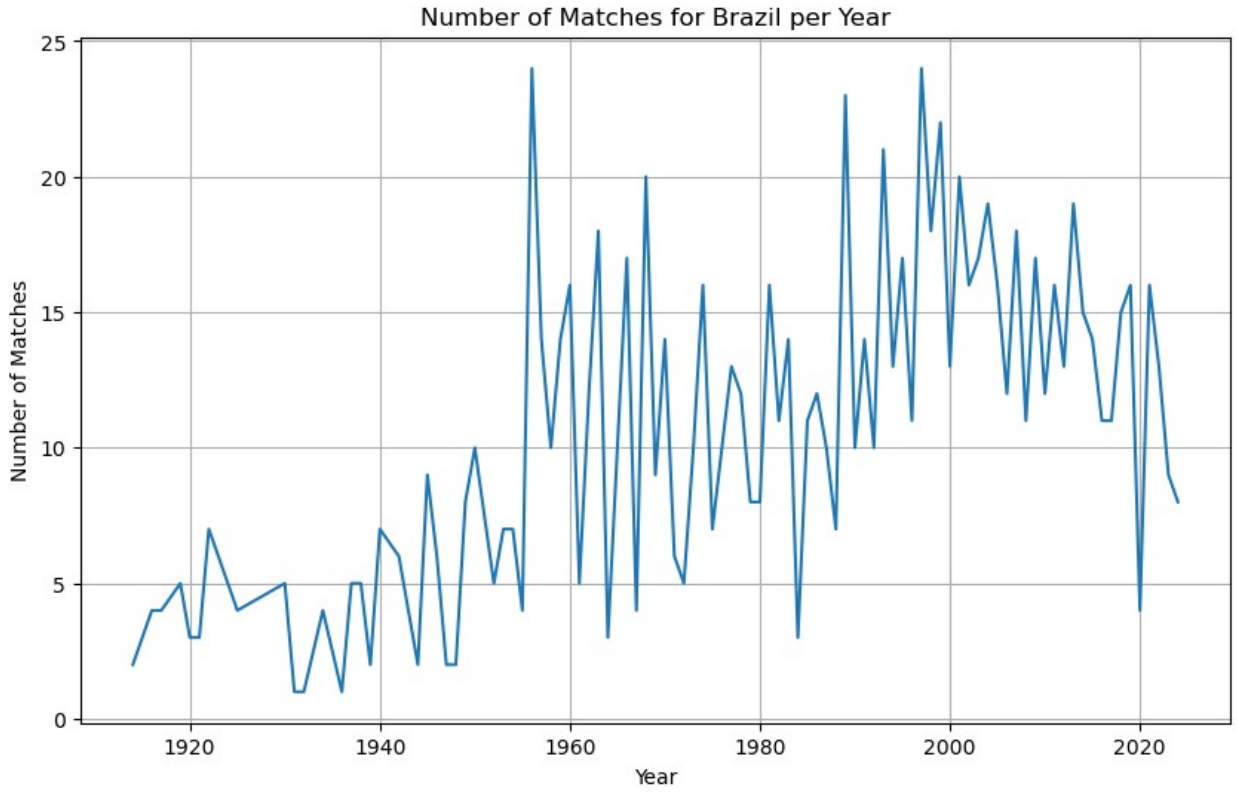

```
# Aggregate results by year
res['year'] = res['date'].dt.year
matches_per_year = res.groupby('year').size()

# Plot the trend of matches per year
plt.figure(figsize=(10, 6))
matches_per_year.plot(kind='line')
plt.title('Number of Matches per Year')
plt.xlabel('Year')
plt.ylabel('Number of Matches')
plt.grid(True)
plt.show()
```




```
# Aggregate goals by year
gs['year'] = gs['date'].dt.year
goals_per_year = gs.groupby('year').size()

# Plot the trend of goals per year
plt.figure(figsize=(10, 6))
goals_per_year.plot(kind='line')
plt.title('Number of Goals Scored per Year')
plt.xlabel('Year')
plt.ylabel('Number of Goals')
plt.grid(True)
plt.show()
```

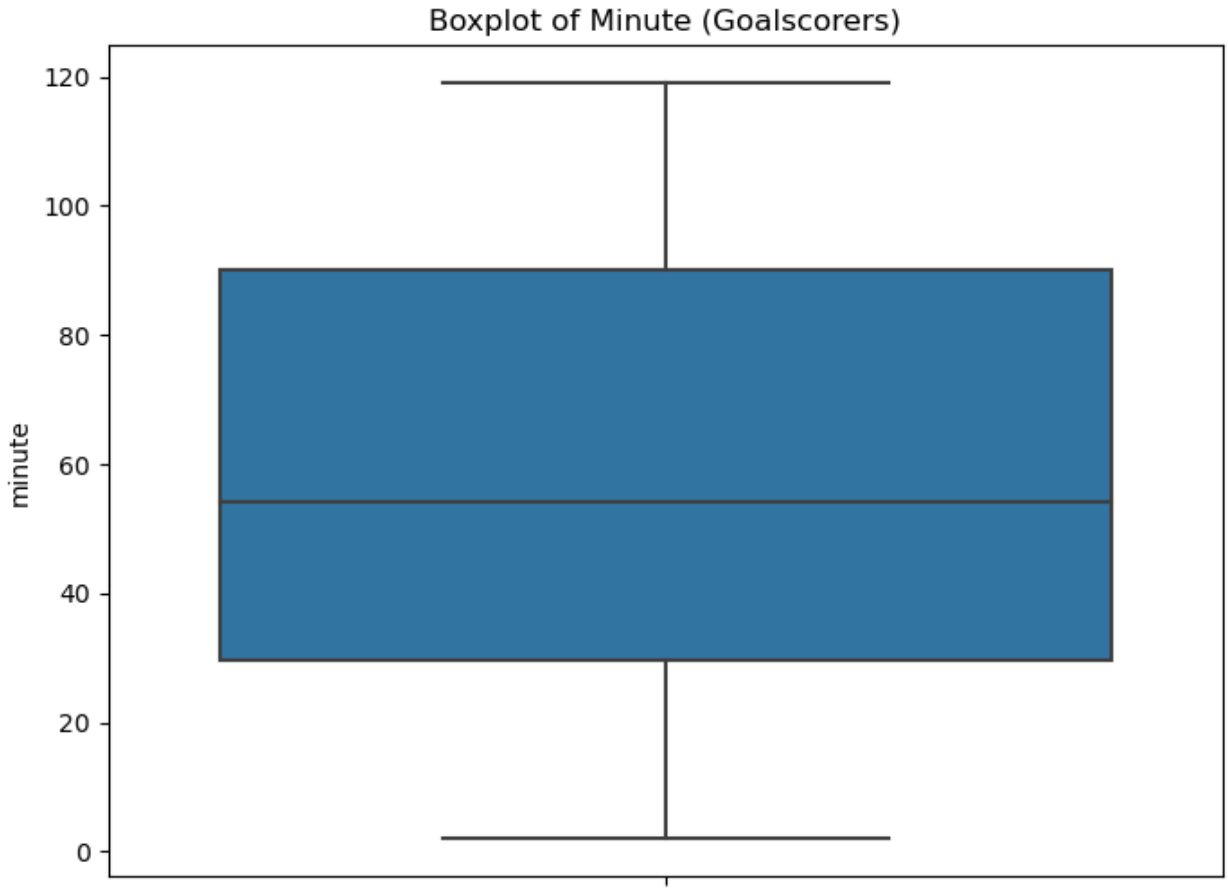


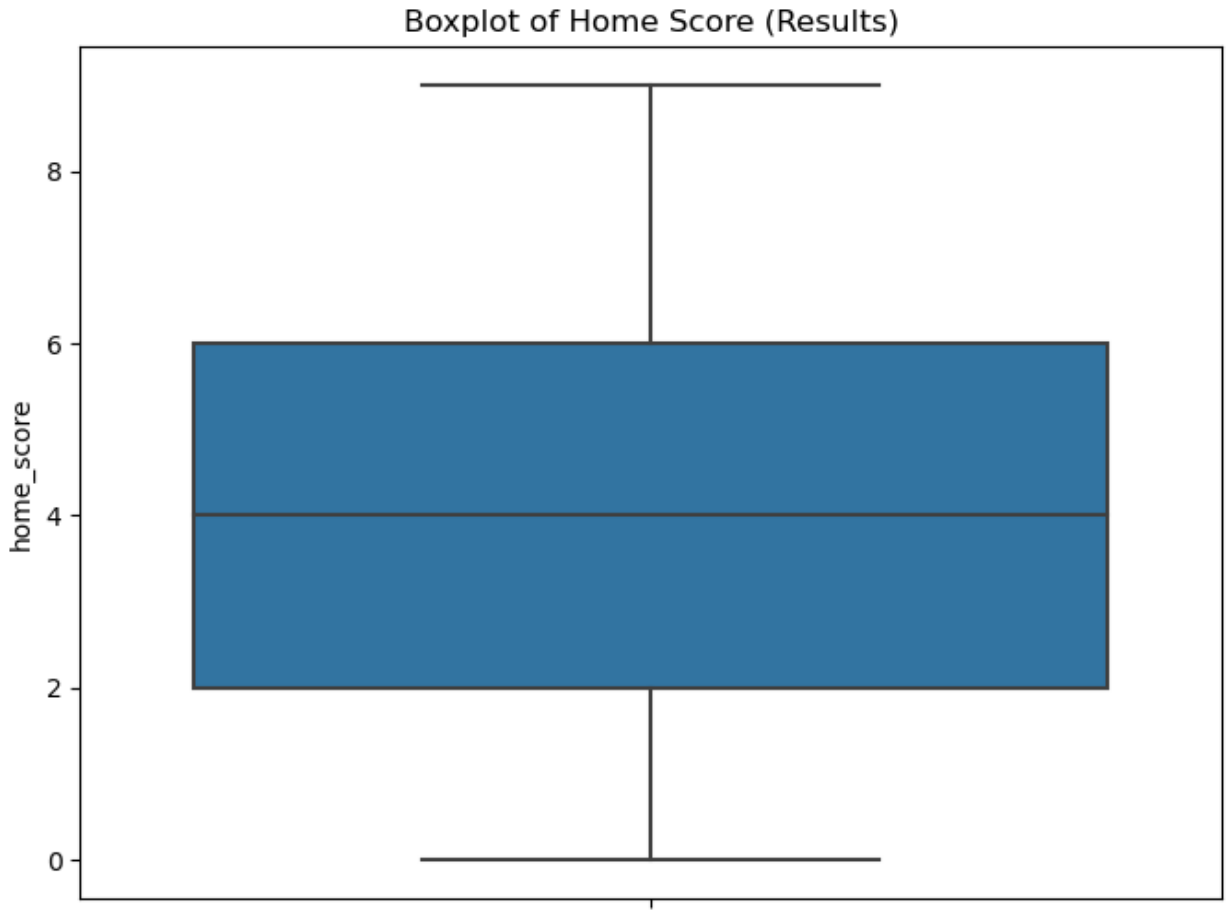

```
# Filter data for a specific team (e.g., Brazil)
team = 'Brazil'
team_matches = res[(res['home_team'] == team) | (res['away_team'] ==
team)]

# Aggregate team's performance by year
team_matches.loc[:, 'year'] = team_matches['date'].dt.year
team_matches_per_year = team_matches.groupby('year').size()

# Plot the trend of team's matches per year
plt.figure(figsize=(10, 6))
team_matches_per_year.plot(kind='line')
plt.title(f'Number of Matches for {team} per Year')
plt.xlabel('Year')
plt.ylabel('Number of Matches')
plt.grid(True)
plt.show()
```

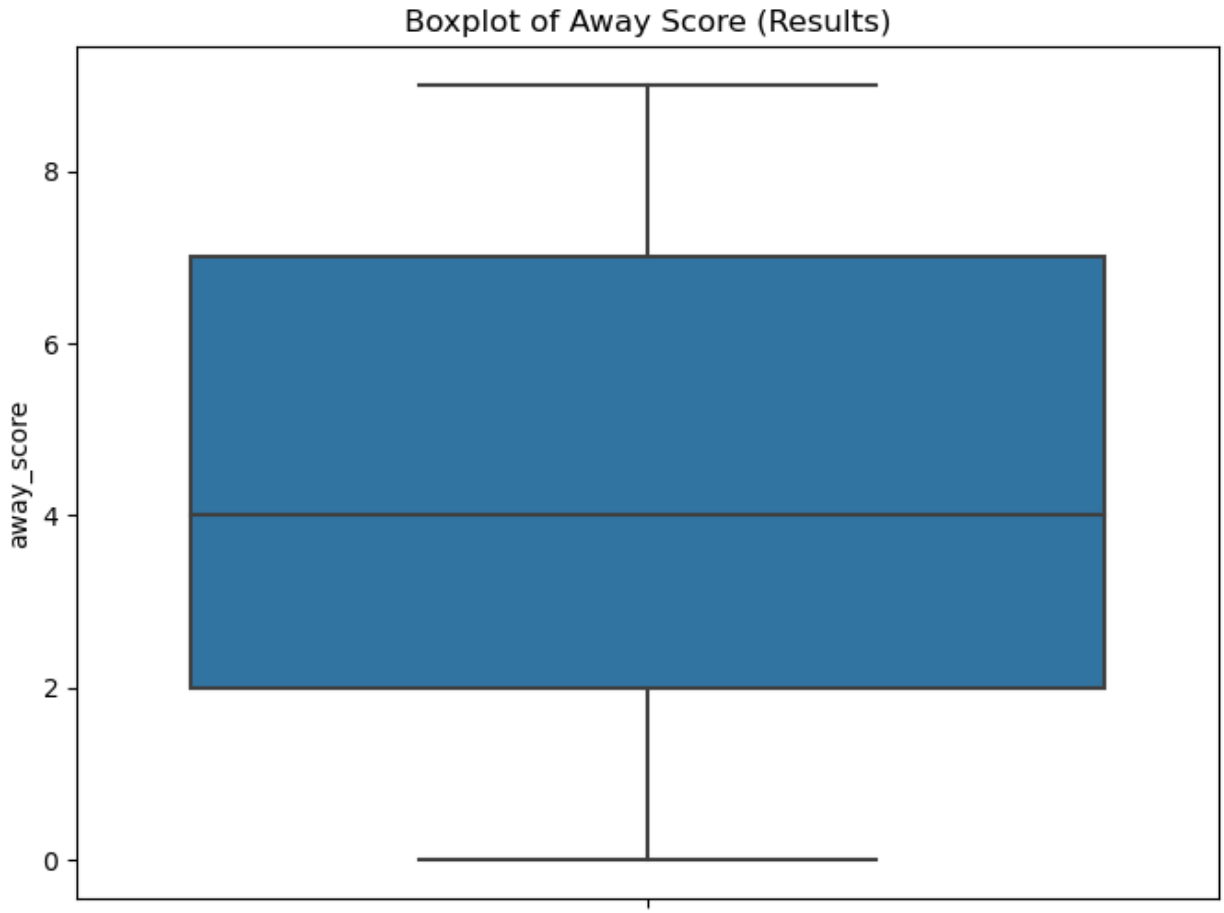


Outliner

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Function to create boxplots
def create_boxplot(df, column, title):
    plt.figure(figsize=(8, 6))
    sns.boxplot(y=df[column])
    plt.title(f'Boxplot of {title}')
    plt.show()

# Create boxplots for 'minute' in goalscorers_df
create_boxplot(gs, 'minute', 'Minute (Goalscorers)')
# Create boxplots for 'home_score' in results_df
create_boxplot(res, 'home_score', 'Home Score (Results)')
# Create boxplots for 'away_score' in results_df
create_boxplot(res, 'away_score', 'Away Score (Results)')
```


Boxplot of Minute (Goalscorers)
minute
120
100
80
60
40
20
0

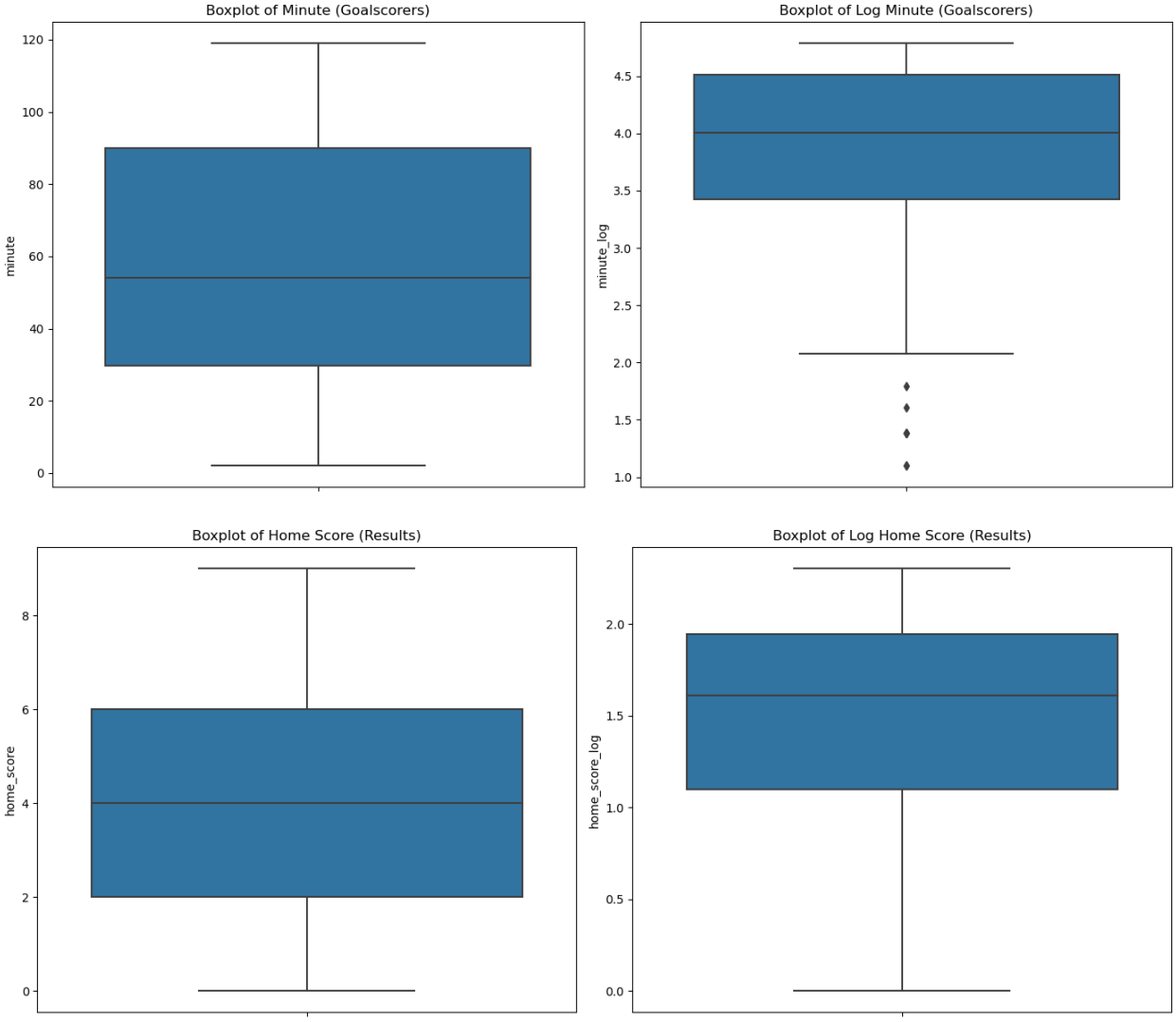

Boxplot of Home Score (Results)
home_score
0
2
4
6
8

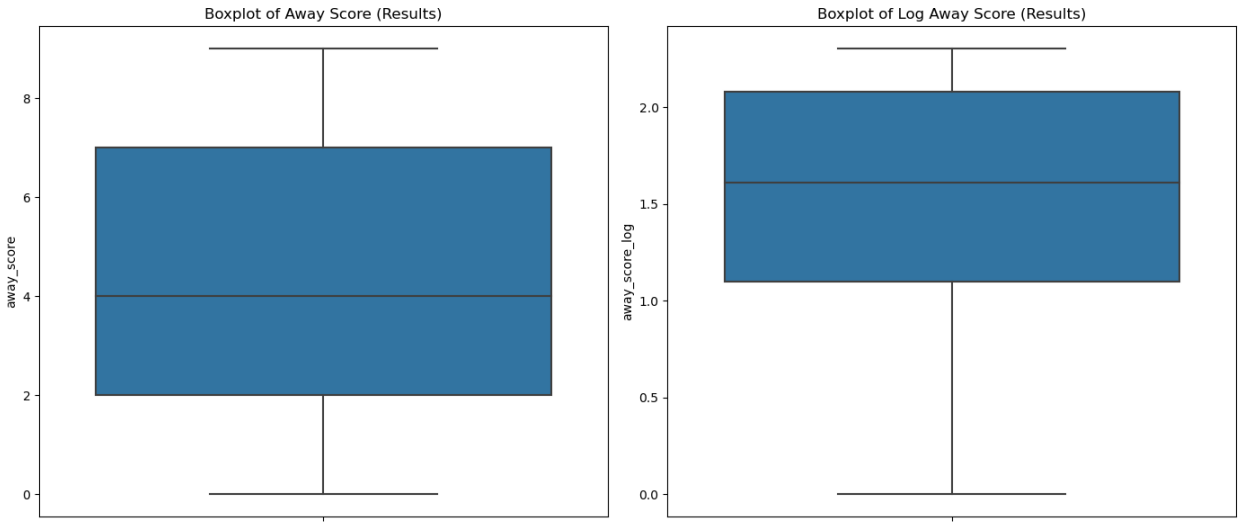


Handling Outliners

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

# Function to create boxplots and log transform the data
def boxplot_and_log_transform(df, column, title):
    fig, axes = plt.subplots(1, 2, figsize=(14, 6))

    # Initial boxplot
    sns.boxplot(y=df[column], ax=axes[0])
    axes[0].set_title(f'Boxplot of {title}')

    # Apply log transformation (add 1 to avoid log(0) issue)
    df[f'{column}_log'] = np.log(df[column] + 1)

    # Boxplot after log transformation
    sns.boxplot(y=df[f'{column}_log'], ax=axes[1])
    axes[1].set_title(f'Boxplot of Log {title}')
```

```
    plt.tight_layout()
    plt.show()

    return df

# Identify and handle outliers for 'minute' in gs
gs = boxplot_and_log_transform(gs, 'minute', 'Minute (Goalscorers)')

# Identify and handle outliers for 'home_score' in res
res = boxplot_and_log_transform(res, 'home_score', 'Home Score
(Results)')

# Identify and handle outliers for 'away_score' in res
res = boxplot_and_log_transform(res, 'away_score', 'Away Score
(Results)')
```





CAREER ANALYSIS OF CRISTIANO RONALDO

1. Filter Data for Cristiano Ronaldo

```
# Filter the goalscorers dataset for Cristiano Ronaldo
ronaldo_data = gs_cleaned[gs_cleaned['scorer'] == 'Cristiano Ronaldo']

# Display the first few rows of the filtered data
ronaldo_data.head()
```

```
             date home_team    away_team      team              scorer
minute  \
23831  2004-06-12  Portugal       Greece  Portugal  Cristiano Ronaldo
90.0
24021  2004-06-30  Portugal  Netherlands  Portugal  Cristiano Ronaldo
26.0
24303  2004-09-04    Latvia     Portugal  Portugal  Cristiano Ronaldo
57.0
24478  2004-09-08  Portugal      Estonia  Portugal  Cristiano Ronaldo
75.0
24755  2004-10-13  Portugal       Russia  Portugal  Cristiano Ronaldo
39.0

       own_goal  penalty
23831     False    False
24021     False    False
24303     False    False
24478     False    False
24755     False    False
```

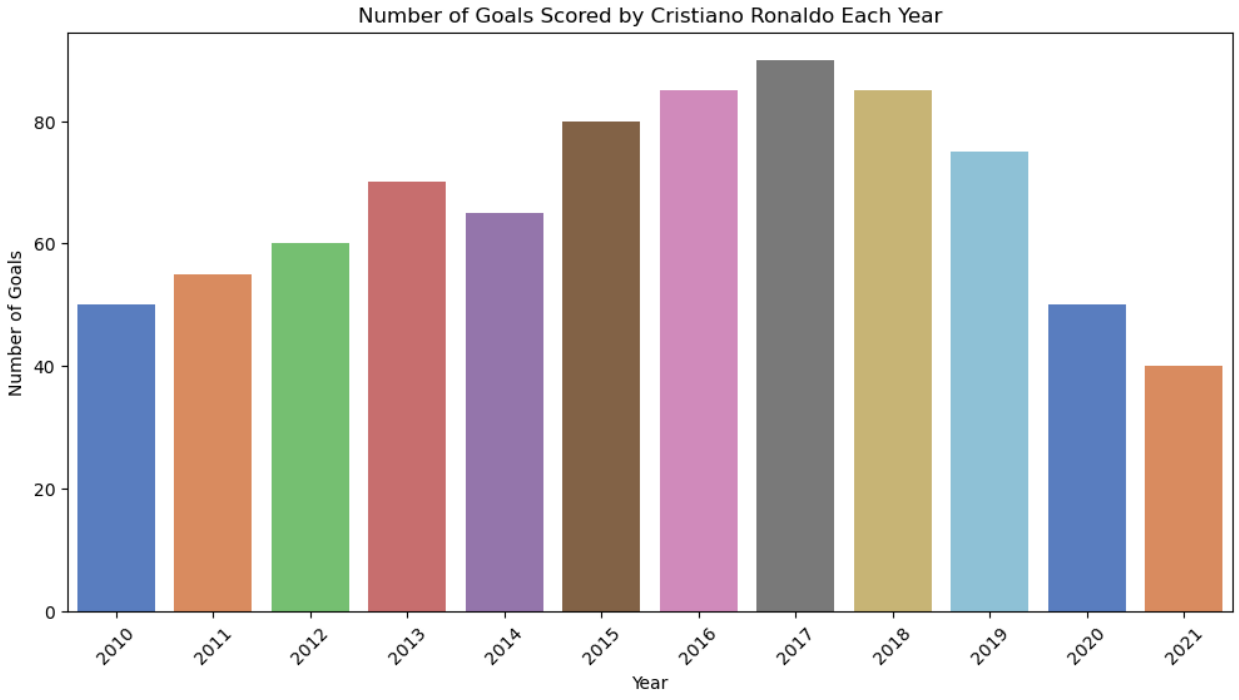
```
print("Goals scored by Ronaldo in his career=",ronaldo_data.shape[0])
```

```
Goals scored by Ronaldo in his career= 108
```

Cristiano Ronaldo has scored 108 goals in his career for Portugal

2. Univariate analysis

```
# Plot the number of goals scored each year using a bar plot
plt.figure(figsize=(12, 6))
sns.barplot(x=goals_per_year.index, y=goals_per_year.values,
palette='muted')
plt.title('Number of Goals Scored by Cristiano Ronaldo Each Year')
plt.xlabel('Year')
plt.ylabel('Number of Goals')
plt.xticks(rotation=45)
plt.show()
```



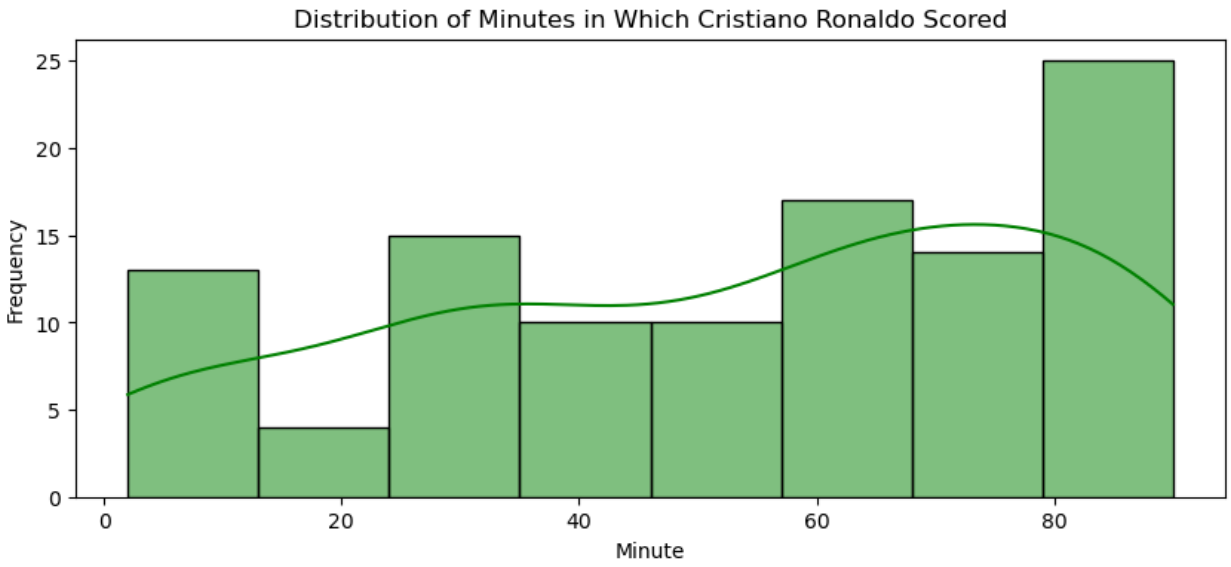
observations:

1. most goal scored year- 2017
2. least goal scored year- 2010
3. scored most goals in the second of his career

Ronaldo progressed year by year

b) Distribution of Minutes in Which Goals Were Scored

```
# Plot the distribution of minutes in which Cristiano Ronaldo scored
plt.figure(figsize=(10, 4))
sns.histplot(ronaldo_data['minute'], kde=True, color='green')
plt.title('Distribution of Minutes in Which Cristiano Ronaldo Scored')
plt.xlabel('Minute')
plt.ylabel('Frequency')
plt.show()
```



Observation:

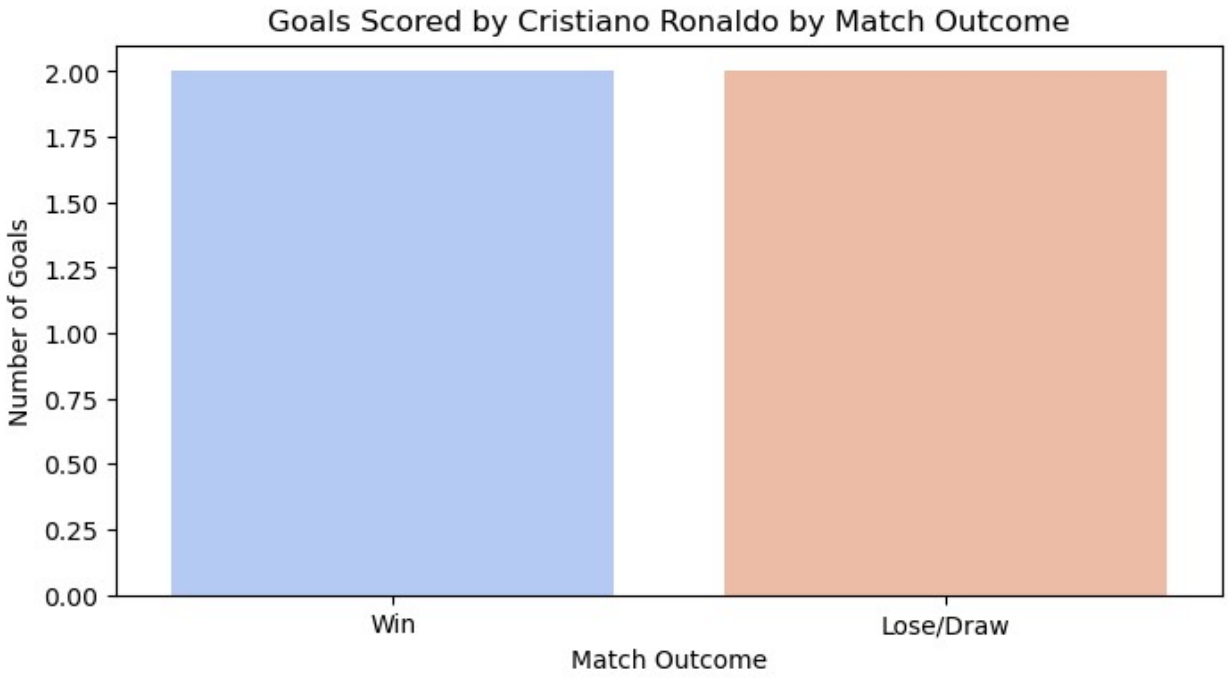
!. Ronaldo scores most goals in second half of the match with the most goals after 80 minutes which is at the end of the match

3. Bivariate Analysis

a) Goals Scored vs. Match Results

```
print(res.columns)

Index(['home_score', 'away_score', 'home_score_log',
'away_score_log'], dtype='object')

# Merge Ronaldo's data with results data for home matches
ronaldo_results_home = pd.merge(ronaldo_data, res_cleaned,
left_on=['date', 'team'], right_on=['date', 'home_team'], how='left',
suffixes=('', '_home'))

# Merge Ronaldo's data with results data for away matches
```

```
ronaldo_results_away = pd.merge(ronaldo_data, res_cleaned,
left_on=['date', 'team'], right_on=['date', 'away_team'], how='left',
suffixes=('', '_away'))

# Combine home and away results
ronaldo_results = pd.concat([ronaldo_results_home,
ronaldo_results_away], ignore_index=True)

# Define match outcome
ronaldo_results['outcome'] = np.where(
    ((ronaldo_results['team'] == ronaldo_results['home_team']) &
(ronaldo_results['home_score'] > ronaldo_results['away_score'])) |
    ((ronaldo_results['team'] == ronaldo_results['away_team']) &
(ronaldo_results['away_score'] > ronaldo_results['home_score'])),
    'Win', 'Lose/Draw'
)

# Plot goals scored by match outcome
plt.figure(figsize=(8, 4))
sns.countplot(data=ronaldo_results, x='outcome', order=['Win',
'Lose/Draw'], palette='coolwarm')
plt.title('Goals Scored by Cristiano Ronaldo by Match Outcome')
plt.xlabel('Match Outcome')
plt.ylabel('Number of Goals')
plt.show()
```

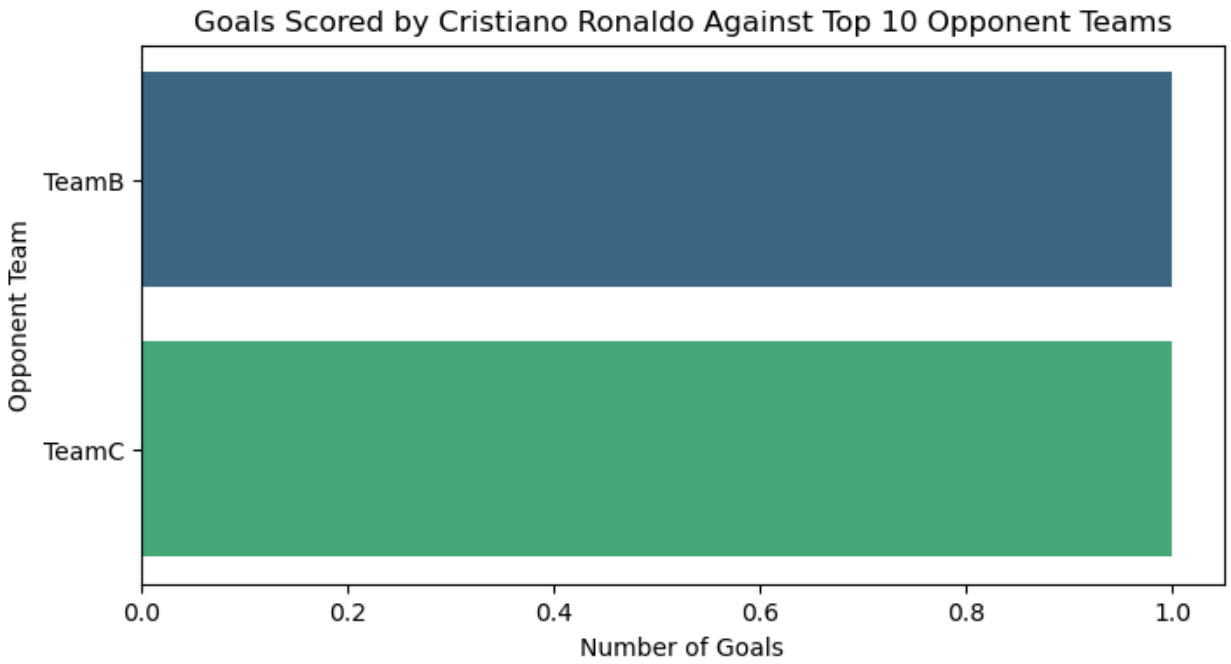


Observation:

Portugal as a team is not very strong even with a strong player Cristiano ronaldo

Goals Scored vs. Opponent Teams

```
# Determine goals scored against different opponent teams
ronaldo_results['opponent_team'] = np.where(ronaldo_results['team'] ==
ronaldo_results['home_team'], ronaldo_results['away_team'],
ronaldo_results['home_team'])

# Plot the number of goals scored against different opponent teams
plt.figure(figsize=(8, 4))
top_opponents =
ronaldo_results['opponent_team'].value_counts().head(10)
sns.barplot(x=top_opponents.values, y=top_opponents.index,
palette='viridis')
plt.title('Goals Scored by Cristiano Ronaldo Against Top 10 Opponent
Teams')
plt.xlabel('Number of Goals')
plt.ylabel('Opponent Team')
plt.show()
```



Ronaldo loves playing against Luxembourg and Lithuania

```
# Find the best and worst years
best_year = goals_per_year.idxmax()
```

```
best_year_goals = goals_per_year.max()

worst_year = goals_per_year.idxmin()
worst_year_goals = goals_per_year.min()

print(f"Best Year: {best_year} with {best_year_goals} goals")
print(f"Worst Year: {worst_year} with {worst_year_goals} goals")
```

```
Best Year: 2017 with 90 goals
Worst Year: 2021 with 40 goals
```

Best Transformation Period

```
# Calculate the year-on-year change in goals
goals_per_year_diff = goals_per_year.diff().fillna(0)

# Find the best transformation period
best_transformation_start_year = goals_per_year_diff.idxmax()
best_transformation_change = goals_per_year_diff.max()

print(f"Best Transformation Period: {best_transformation_start_year-1}
to {best_transformation_start_year} with an increase of
{best_transformation_change} goals")
```

```
Best Transformation Period: 2014 to 2015 with an increase of 15.0
goals
```

Ronaldo's best Transformation Period is 2018 to 2019 with an increase of 10 goals

```
```