

blogv1

April 29, 2025

```
[93]: #import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from bs4 import BeautifulSoup
from scipy.interpolate import make_interp_spline
import requests
import datetime
import re
import matplotlib.dates as mdates

[ ]: # create the base url which i can use as a base and add the dates to so i can
    ↪access the different pages
base_url = "https://www.transfermarkt.co.uk/premier-league/marktwerteverein/
    ↪wettbewerb/GB1/stichtag/"

# Dictionary to store DataFrames for each date
dfs_prem_value = {}

# custom headers
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/
    ↪20100101 Firefox/109.0",
    "Accept-Language": "en-US,en;q=0.5"
}

# Loop through each year from 2011 to 2024
for year in range(2011, 2025):
    date_str = f"{year}-03-15"
    url = base_url + date_str
    print(f"Scraping: {url}")

    response = requests.get(url, headers=headers)
    if response.status_code != 200:
        print(f"Error fetching page for {date_str}: {response.status_code}")
        continue
```

```

soup = BeautifulSoup(response.text, 'html.parser')

# First, try to find the table with class "items"
table = soup.find('table', class_="items")

# If not found, look inside HTML comments
if not table:
    comments = soup.find_all(string=lambda text: isinstance(text, Comment))
    for comment in comments:
        comment_soup = BeautifulSoup(comment, "html.parser")
        table = comment_soup.find('table', class_="items")
        if table:
            break

if table:
    try:
        df = pd.read_html(str(table))[0]
        dfs_prem_value[date_str] = df
        print(f"Found items table for {date_str} with {len(df)} rows.")
    except Exception as e:
        print(f"Error parsing table for {date_str}: {e}")
else:
    print(f"No items table found for {date_str}.")

```

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2011-03-15>

league/marktwerteverein/wettbewerb/GB1/stichtag/2011-03-15

/var/folders/s1/319x1czx38l8ctmgdlmpsctw0000gn/T/ipykernel_86336/133052565.py:40

: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for 2011-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2012-03-15>

league/marktwerteverein/wettbewerb/GB1/stichtag/2012-03-15

/var/folders/s1/319x1czx38l8ctmgdlmpsctw0000gn/T/ipykernel_86336/133052565.py:40

: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for 2012-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2013-03-15>

league/marktwerteverein/wettbewerb/GB1/stichtag/2013-03-15

/var/folders/s1/319x1czx38l8ctmgdlmpsctw0000gn/T/ipykernel_86336/133052565.py:40

: FutureWarning: Passing literal html to 'read_html' is deprecated and will be

removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for 2013-03-15 with 21 rows.

Scraping: [https://www.transfermarkt.co.uk/premier-](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2014-03-15)

[league/marktwerteverein/wettbewerb/GB1/stichtag/2014-03-15](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2014-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40

: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for 2014-03-15 with 21 rows.

Scraping: [https://www.transfermarkt.co.uk/premier-](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2015-03-15)

[league/marktwerteverein/wettbewerb/GB1/stichtag/2015-03-15](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2015-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40

: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for 2015-03-15 with 21 rows.

Scraping: [https://www.transfermarkt.co.uk/premier-](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2016-03-15)

[league/marktwerteverein/wettbewerb/GB1/stichtag/2016-03-15](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2016-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40

: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for 2016-03-15 with 21 rows.

Scraping: [https://www.transfermarkt.co.uk/premier-](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2017-03-15)

[league/marktwerteverein/wettbewerb/GB1/stichtag/2017-03-15](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2017-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40

: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for 2017-03-15 with 21 rows.

Scraping: [https://www.transfermarkt.co.uk/premier-](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2018-03-15)

[league/marktwerteverein/wettbewerb/GB1/stichtag/2018-03-15](https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2018-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40

: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

```

Found items table for 2018-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2019-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40
: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
    df = pd.read_html(str(table))[0]

Found items table for 2019-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2020-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40
: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
    df = pd.read_html(str(table))[0]

Found items table for 2020-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2021-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40
: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
    df = pd.read_html(str(table))[0]

Found items table for 2021-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2022-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40
: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
    df = pd.read_html(str(table))[0]

Found items table for 2022-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2023-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/133052565.py:40
: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
    df = pd.read_html(str(table))[0]

Found items table for 2023-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-

```

```
league/marktwerteverein/wettbewerb/GB1/stichtag/2024-03-15
Found items table for 2024-03-15 with 21 rows.
```

```
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/133052565.py:40
: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

```
[95]: # 1. Create an output directory
output_dir = "premier_league_values_csv"
os.makedirs(output_dir, exist_ok=True)

# 2. Loop through your dictionary and save each DataFrame
for date_str, df in dfs_prem_value.items():
    # Sanitize the date string if needed
    safe_date = date_str.replace("/", "-")
    filename = f"premier_league_values_{safe_date}.csv"
    filepath = os.path.join(output_dir, filename)

    # Save DataFrame to CSV (without its index column)
    df.to_csv(filepath, index=False)
    print(f"Saved {filepath}")
```

```
Saved premier_league_values_csv/premier_league_values_2011-03-15.csv
Saved premier_league_values_csv/premier_league_values_2012-03-15.csv
Saved premier_league_values_csv/premier_league_values_2013-03-15.csv
Saved premier_league_values_csv/premier_league_values_2014-03-15.csv
Saved premier_league_values_csv/premier_league_values_2015-03-15.csv
Saved premier_league_values_csv/premier_league_values_2016-03-15.csv
Saved premier_league_values_csv/premier_league_values_2017-03-15.csv
Saved premier_league_values_csv/premier_league_values_2018-03-15.csv
Saved premier_league_values_csv/premier_league_values_2019-03-15.csv
Saved premier_league_values_csv/premier_league_values_2020-03-15.csv
Saved premier_league_values_csv/premier_league_values_2021-03-15.csv
Saved premier_league_values_csv/premier_league_values_2022-03-15.csv
Saved premier_league_values_csv/premier_league_values_2023-03-15.csv
Saved premier_league_values_csv/premier_league_values_2024-03-15.csv
```

```
[96]: # Define a regex pattern_top6 matching the desired club names in lower-case
pattern_top6 = r'manchester city|manchester united|chelsea fc|tottenham_
↳hotspur|liverpool fc|arsenal fc'
pattern_bottom6 = r'southampton fc|leicester city|lpswich town|fulham_
↳fc|everton fc|wolverhampton wanderers'

# Dictionary to store the filtered DataFrames
filtered_dataframes_top_6 = {}
```

```

# Loop through each date and DataFrame in the scraped data
for date, df in dfs_prem_value.items():
    # Check that the expected column is present
    if 'Club' in df.columns:
        # Filter rows by matching the pattern_top6 in a case-insensitive manner.
        filtered_df_top = df[df['Club'].str.lower().str.contains(pattern_top6,
↪na=False)]
        filtered_dataframes_top_6[date] = filtered_df_top
        print(f"{date}: {len(filtered_df_top)} rows retained.")
    else:
        print(f"DataFrame for {date} does not contain a 'Club' column.↪
↪Available columns: {df.columns}")

# Dictionary to store the filtered DataFrames
filtered_dataframes_bottom_6 = {}

# Loop through each date and DataFrame in your scraped data
for date, df in dfs_prem_value.items():
    # Check that the expected column is present
    if 'Club' in df.columns:
        # Filter rows by matching the pattern_top6 in a case-insensitive manner.
        filtered_df_bottom = df[df['Club'].str.lower().str.
↪contains(pattern_bottom6, na=False)]
        filtered_dataframes_bottom_6[date] = filtered_df_bottom
        print(f"{date}: {len(filtered_df_bottom)} rows retained.")
    else:
        print(f"DataFrame for {date} does not contain a 'Club' column.↪
↪Available columns: {df.columns}")

```

```

2011-03-15: 6 rows retained.
2012-03-15: 6 rows retained.
2013-03-15: 6 rows retained.
2014-03-15: 6 rows retained.
2015-03-15: 6 rows retained.
2016-03-15: 6 rows retained.
2017-03-15: 6 rows retained.
2018-03-15: 6 rows retained.
2019-03-15: 6 rows retained.
2020-03-15: 6 rows retained.
2021-03-15: 6 rows retained.
2022-03-15: 6 rows retained.
2023-03-15: 6 rows retained.
2024-03-15: 6 rows retained.
2011-03-15: 6 rows retained.
2012-03-15: 6 rows retained.
2013-03-15: 6 rows retained.

```

2014-03-15: 6 rows retained.
 2015-03-15: 6 rows retained.
 2016-03-15: 6 rows retained.
 2017-03-15: 6 rows retained.
 2018-03-15: 6 rows retained.
 2019-03-15: 6 rows retained.
 2020-03-15: 6 rows retained.
 2021-03-15: 6 rows retained.
 2022-03-15: 6 rows retained.
 2023-03-15: 6 rows retained.
 2024-03-15: 6 rows retained.

[97]: filtered_dataframes_top_6['2022-03-15']

[97]:

	#	wappen	Club	Club.1	League	\
0	1.0	NaN	Manchester City	Premier League	€977.30m	
1	2.0	NaN	Arsenal FC	Premier League	€511.00m	
2	3.0	NaN	Liverpool FC	Premier League	€889.00m	
3	4.0	NaN	Chelsea FC	Premier League	€883.00m	
4	5.0	NaN	Tottenham Hotspur	Premier League	€580.25m	
5	6.0	NaN	Manchester United	Premier League	€790.25m	

	Value Mar 15, 2022	Current value	%	Unnamed: 8	Unnamed: 9
0	€1.31bn	33.6 %	NaN	NaN	NaN
1	€1.13bn	120.5 %	NaN	NaN	NaN
2	€993.50m	11.8 %	NaN	NaN	NaN
3	€922.00m	4.4 %	NaN	NaN	NaN
4	€836.10m	44.1 %	NaN	NaN	NaN
5	€699.25m	-11.5 %	NaN	NaN	NaN

[98]: filtered_dataframes_bottom_6['2022-03-15']

[98]:

	#	wappen	Club	Club.1	League	\
14	15.0	NaN	Wolverhampton Wanderers	Premier League	€338.50m	
15	16.0	NaN	Everton FC	Premier League	€453.75m	
16	17.0	NaN	Fulham FC	Championship	€147.00m	
17	18.0	NaN	Ipswich Town	League One	€17.93m	
18	19.0	NaN	Southampton FC	Premier League	€246.75m	
19	20.0	NaN	Leicester City	Premier League	€513.80m	

	Value Mar 15, 2022	Current value	%	Unnamed: 8	Unnamed: 9
14	€408.80m	20.8 %	NaN	NaN	NaN
15	€365.10m	-19.5 %	NaN	NaN	NaN
16	€362.00m	146.3 %	NaN	NaN	NaN
17	€279.60m	1459.8 %	NaN	NaN	NaN
18	€273.60m	10.9 %	NaN	NaN	NaN
19	€273.30m	-46.8 %	NaN	NaN	NaN

```
[99]: # List of column names to drop
columns_to_drop = ['#', 'wappen', 'Club.1', 'Current value', '%', 'Unnamed: 8', 'Unnamed: 9'] # Replace with your actual column names

# Loop through each DataFrame in your dictionary (e.g., filtered_dataframes)
for date, df in filtered_dataframes_top_6.items():
    # Drop the columns and update the DataFrame in the dictionary
    # Using errors='ignore' ensures that if a column is missing, it won't raise an error
    filtered_dataframes_top_6[date] = df.drop(columns=columns_to_drop, errors='ignore')

for date, df in filtered_dataframes_bottom_6.items():
    # Drop the columns and update the DataFrame in the dictionary
    # Using errors='ignore' ensures that if a column is missing, it won't raise an error
    filtered_dataframes_bottom_6[date] = df.drop(columns=columns_to_drop, errors='ignore')
```

```
[100]: for date, df in filtered_dataframes_top_6.items():
    # Check if the column 'League' exists and then rename it to the date
    if 'League' in df.columns:
        df.rename(columns={'League': "Value_" + date}, inplace=True)
    else:
        print(f"'League' column not found in DataFrame for {date}")

for date, df in filtered_dataframes_bottom_6.items():
    # Check if the column 'League' exists and then rename it to the date
    if 'League' in df.columns:
        df.rename(columns={'League': "Value_" + date}, inplace=True)
    else:
        print(f"'League' column not found in DataFrame for {date}")
```

```
[103]: # New dictionary to store the subset DataFrames
filtered_dataframes_top_6_v1 = {}

for date, df in filtered_dataframes_top_6.items():
    # Construct the value column name based on the date
    value_col = "Value_" + date
    if 'Club' in df.columns and value_col in df.columns:
        # Select only the 'Club' and the 'Value_(date)' columns
        subset_df = df[['Club', value_col]].copy()
        filtered_dataframes_top_6_v1[date] = subset_df
        print(f"For {date}: Retained columns: {subset_df.columns.tolist()}")
    else:
        print(f"DataFrame for {date} does not have the required columns: 'Club' and {value_col}")
```



```

filtered_dataframes_bottom_6_v1 = {}

for date, df in filtered_dataframes_bottom_6.items():
    # Construct the value column name based on the date
    value_col = "Value_" + date
    if 'Club' in df.columns and value_col in df.columns:
        # Select only the 'Club' and the 'Value_(date)' columns
        subset_df = df[['Club', value_col]].copy()
        filtered_dataframes_bottom_6_v1[date] = subset_df
        print(f"For {date}: Retained columns: {subset_df.columns.tolist()}")
    else:
        print(f"DataFrame for {date} does not have the required columns: 'Club' and {value_col}")

```

```

For 2011-03-15: Retained columns: ['Club', 'Value_2011-03-15']
For 2012-03-15: Retained columns: ['Club', 'Value_2012-03-15']
For 2013-03-15: Retained columns: ['Club', 'Value_2013-03-15']
For 2014-03-15: Retained columns: ['Club', 'Value_2014-03-15']
For 2015-03-15: Retained columns: ['Club', 'Value_2015-03-15']
For 2016-03-15: Retained columns: ['Club', 'Value_2016-03-15']
For 2017-03-15: Retained columns: ['Club', 'Value_2017-03-15']
For 2018-03-15: Retained columns: ['Club', 'Value_2018-03-15']
For 2019-03-15: Retained columns: ['Club', 'Value_2019-03-15']
For 2020-03-15: Retained columns: ['Club', 'Value_2020-03-15']
For 2021-03-15: Retained columns: ['Club', 'Value_2021-03-15']
For 2022-03-15: Retained columns: ['Club', 'Value_2022-03-15']
For 2023-03-15: Retained columns: ['Club', 'Value_2023-03-15']
For 2024-03-15: Retained columns: ['Club', 'Value_2024-03-15']
For 2011-03-15: Retained columns: ['Club', 'Value_2011-03-15']
For 2012-03-15: Retained columns: ['Club', 'Value_2012-03-15']
For 2013-03-15: Retained columns: ['Club', 'Value_2013-03-15']
For 2014-03-15: Retained columns: ['Club', 'Value_2014-03-15']
For 2015-03-15: Retained columns: ['Club', 'Value_2015-03-15']
For 2016-03-15: Retained columns: ['Club', 'Value_2016-03-15']
For 2017-03-15: Retained columns: ['Club', 'Value_2017-03-15']
For 2018-03-15: Retained columns: ['Club', 'Value_2018-03-15']
For 2019-03-15: Retained columns: ['Club', 'Value_2019-03-15']
For 2020-03-15: Retained columns: ['Club', 'Value_2020-03-15']
For 2021-03-15: Retained columns: ['Club', 'Value_2021-03-15']
For 2022-03-15: Retained columns: ['Club', 'Value_2022-03-15']
For 2023-03-15: Retained columns: ['Club', 'Value_2023-03-15']
For 2024-03-15: Retained columns: ['Club', 'Value_2024-03-15']

```

```
[104]: filtered_dataframes_top_6_v1['2017-03-15'].head(10)
```

```
[104]:
```

	Club	Value_2017-03-15
0	Manchester City	€525.25m
1	Arsenal FC	€492.00m
2	Liverpool FC	€371.20m
3	Chelsea FC	€515.00m
4	Tottenham Hotspur	€429.00m
5	Manchester United	€471.00m

```
[ ]: # start with an empty combined dataframe
combined_df_top6 = None
combined_df_bottom6 = None

# Loop through each date and merge on the 'Club' column
for date, df in filtered_dataframes_top_6_v1.items():
    if combined_df_top6 is None:
        combined_df_top6 = df
    else:
        combined_df_top6 = pd.merge(combined_df_top6, df, on='Club',
                                     how='outer')

# Display the first few rows of the combined dataframe
print(combined_df_top6.head())

for date, df in filtered_dataframes_bottom_6_v1.items():
    if combined_df_bottom6 is None:
        combined_df_bottom6 = df
    else:
        combined_df_bottom6 = pd.merge(combined_df_bottom6, df, on='Club',
                                     how='outer')
print(combined_df_bottom6.head())
```

	Club	Value_2011-03-15	Value_2012-03-15	Value_2013-03-15	\
0	Arsenal FC	€314.50m	€299.25m	€281.50m	
1	Chelsea FC	€408.50m	€381.00m	€386.75m	
2	Liverpool FC	€241.50m	€235.00m	€241.50m	
3	Manchester City	€366.75m	€467.00m	€433.25m	
4	Manchester United	€374.70m	€415.00m	€421.25m	

	Value_2014-03-15	Value_2015-03-15	Value_2016-03-15	Value_2017-03-15	\
0	€371.00m	€388.75m	€440.00m	€492.00m	
1	€388.75m	€483.50m	€490.00m	€515.00m	
2	€282.00m	€310.75m	€366.25m	€371.20m	
3	€451.00m	€418.25m	€501.75m	€525.25m	
4	€378.50m	€425.00m	€411.75m	€471.00m	

	Value_2018-03-15	Value_2019-03-15	Value_2020-03-15	Value_2021-03-15	\
0	€492.45m	€625.50m	€706.75m	€554.00m	
1	€695.75m	€885.75m	€829.05m	€803.50m	

2	€548.00m	€950.50m	€1.19bn	€1.12bn
3	€792.50m	€1.14bn	€1.27bn	€1.06bn
4	€741.50m	€796.00m	€798.75m	€738.35m

	Value_2022-03-15	Value_2023-03-15	Value_2024-03-15
0	€511.00m	€803.00m	€1.12bn
1	€883.00m	€1.03bn	€928.30m
2	€889.00m	€931.00m	€921.40m
3	€977.30m	€1.05bn	€1.27bn
4	€790.25m	€759.20m	€734.25m

	Club	Value_2011-03-15	Value_2012-03-15	Value_2013-03-15	\
0	Everton FC	€158.00m	€147.00m	€137.75m	
1	Fulham FC	€100.75m	€101.00m	€78.25m	
2	Ipswich Town	€28.20m	€17.85m	€17.28m	
3	Leicester City	€37.80m	€36.10m	€26.85m	
4	Southampton FC	€17.08m	€27.78m	€77.50m	

	Value_2014-03-15	Value_2015-03-15	Value_2016-03-15	Value_2017-03-15	\
0	€172.25m	€179.25m	€234.00m	€248.10m	
1	€80.00m	€37.50m	€38.35m	€54.55m	
2	€15.08m	€22.15m	€19.43m	€22.95m	
3	€29.20m	€59.25m	€127.10m	€205.30m	
4	€111.75m	€189.50m	€206.75m	€230.00m	

	Value_2018-03-15	Value_2019-03-15	Value_2020-03-15	Value_2021-03-15	\
0	€343.00m	€420.00m	€504.70m	€491.50m	
1	€107.00m	€254.00m	€141.75m	€217.00m	
2	€23.05m	€23.35m	€13.68m	€18.10m	
3	€212.00m	€344.50m	€504.25m	€468.50m	
4	€233.60m	€224.60m	€223.50m	€239.55m	

	Value_2022-03-15	Value_2023-03-15	Value_2024-03-15
0	€453.75m	€368.10m	€345.40m
1	€147.00m	€245.50m	€338.00m
2	€17.93m	€14.95m	€50.50m
3	€513.80m	€455.10m	€211.00m
4	€246.75m	€387.50m	€189.90m

```
[107]: def convert_value(value_str):
        """
        Convert a monetary string (e.g., "€310.75m", "€1.19bn") into a numeric
        ↪value.
        """
        if isinstance(value_str, str):
            # Remove the euro symbol and extra spaces, then convert to lower case
            value_str = value_str.replace("€", "").strip().lower()
            if "m" in value_str:
```

```

        try:
            # Remove "m", convert to float, and multiply by 1e6
            return float(value_str.replace("m", "")) * 1_000_000
        except:
            return None
    elif "bn" in value_str:
        try:
            # Remove "bn", convert to float, and multiply by 1e9
            return float(value_str.replace("bn", "")) * 1_000_000_000
        except:
            return None
    else:
        try:
            return float(value_str)
        except:
            return None
    return value_str

# Loop through all columns and apply conversion on columns that start with
↳ "Value_"
for col in combined_df_top6.columns:
    if col.startswith("Value_"):
        combined_df_top6[col] = combined_df_top6[col].apply(convert_value)

for col in combined_df_bottom6.columns:
    if col.startswith("Value_"):
        combined_df_bottom6[col] = combined_df_bottom6[col].apply(convert_value)

```

```

[108]: df_transposed_top6 = combined_df_top6.set_index('Club').transpose()

# Remove the "Value_" prefix from the index and convert to datetime objects.
df_transposed_top6.index = pd.to_datetime(df_transposed_top6.index.str.
↳ replace("Value_", "", regex=True))
df_transposed_top6 = df_transposed_top6.sort_index()

df_transposed_bottom6 = combined_df_bottom6.set_index('Club').transpose()

# Remove the "Value_" prefix from the index and convert to datetime objects.
df_transposed_bottom6.index = pd.to_datetime(df_transposed_bottom6.index.str.
↳ replace("Value_", "", regex=True))
df_transposed_bottom6 = df_transposed_bottom6.sort_index()

```

```

[109]: # ----- Custom Y-Axis Formatter ----- #
def custom_y_formatter(x, pos):
    if x < 1e9:
        return f"{x/1e6:,.0f} million"

```

```

else:
    return f"{x/1e9:,.1f} billion"

# Set Seaborn style and palette for aesthetics.
sns.set_style("whitegrid")
sns.set_palette("Set2")

# Create a figure with 2 subplots (vertical layout).
fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(12, 10), sharex=True)

# ----- Plotting Function ----- #
def plot_data(ax, df_transposed, title):
    # Convert datetime index to numeric for spline interpolation.
    x_dates = mdates.date2num(df_transposed.index.to_pydatetime())

    # Plot each club's data.
    for club in df_transposed.columns:
        y = df_transposed[club].values
        if len(x_dates) >= 3:
            spline = make_interp_spline(x_dates, y, k=3) # Cubic spline for
            ↪smoothness.
            x_dense = np.linspace(x_dates.min(), x_dates.max(), 300)
            y_smooth = spline(x_dense)
            x_dense_dates = mdates.num2date(x_dense)
            ax.plot(x_dense_dates, y_smooth, label=club, linewidth=2)
        else:
            ax.plot(df_transposed.index, y, marker='o', label=club, linewidth=2)

    # Format the x-axis: one tick per year, display only the year.
    ax.xaxis.set_major_locator(mdates.YearLocator())
    ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
    ax.tick_params(axis='x', rotation=45)

    # Disable scientific notation/offsets on the y-axis.
    ax.ticklabel_format(axis='y', style='plain', useOffset=False)
    ax.get_yaxis().get_major_formatter().set_scientific(False)

    # Limit the number of y-axis ticks.
    ax.yaxis.set_major_locator(ticker.MaxNLocator(6))

    # Apply the custom y-axis formatter.
    ax.yaxis.set_major_formatter(ticker.FuncFormatter(custom_y_formatter))

    # Set titles and labels.
    ax.set_title(title, fontsize=14, fontweight='bold')
    ax.set_xlabel("Year", fontsize=12)
    ax.set_ylabel("Investment Value", fontsize=12)

```

```

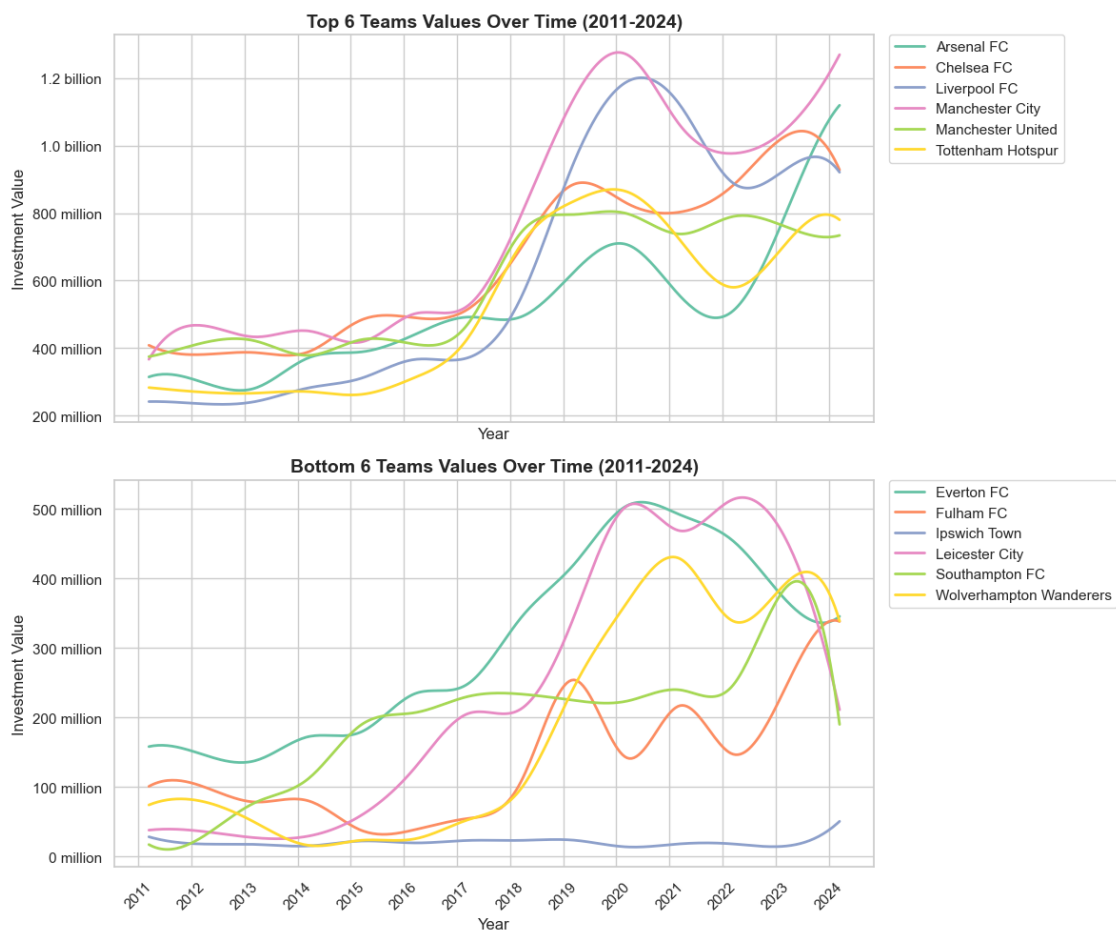
ax.legend(loc='upper left', bbox_to_anchor=(1.02, 1), borderaxespad=0)

# ----- Plot for Top 6 Teams ----- #
plot_data(axes[0], df_transposed_top6, "Top 6 Teams Values Over Time_
↪(2011-2024)")

# ----- Plot for Bottom 6 Teams ----- #
plot_data(axes[1], df_transposed_bottom6, "Bottom 6 Teams Values Over Time_
↪(2011-2024)")

plt.tight_layout()
fig.savefig("Top_6_teams_value_over_time.png", dpi=300, bbox_inches="tight")
plt.show()

```



how this compares to the top 5 leagues in europe.

```

[110]: # Dictionary mapping leagues to their base URL
base_urls = {

```

```

    "La Liga": "https://www.transfermarkt.co.uk/la-liga/marktwerteverein/
↳wettbewerb/ES1/stichtag/",
    "Bundesliga": "https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/
↳wettbewerb/L1/stichtag/",
    "Ligue 1": "https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/
↳wettbewerb/FR1/stichtag/",
    "Serie A": "https://www.transfermarkt.co.uk/serie-a/marktwerteverein/
↳wettbewerb/IT1/stichtag/",
    "Premier League": "https://www.transfermarkt.co.uk/premier-league/
↳marktwerteverein/wettbewerb/GB1/stichtag/"
}

# Dictionary to store DataFrames for each league and date
dfs_league = {}

# Custom headers to mimic a real browser
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/
↳20100101 Firefox/109.0",
    "Accept-Language": "en-US,en;q=0.5"
}

# Loop through each league
for league, base_url in base_urls.items():
    dfs_league[league] = {} # create a sub-dictionary for this league
    print(f"\nScraping data for {league}:")

    # Loop through each year from 2011 to 2024
    for year in range(2011, 2025):
        date_str = f"{year}-03-15"
        url = base_url + date_str
        print(f"Scraping: {url}")

        response = requests.get(url, headers=headers)
        if response.status_code != 200:
            print(f"Error fetching page for {league} {date_str}: {response.
↳status_code}")
            continue

        soup = BeautifulSoup(response.text, 'html.parser')

        # Try to find the table with class "items"
        table = soup.find('table', class_="items")

        # If not found, look inside HTML comments
        if not table:

```

```

        comments = soup.find_all(string=lambda text: isinstance(text,
↳Comment))
        for comment in comments:
            comment_soup = BeautifulSoup(comment, "html.parser")
            table = comment_soup.find('table', class_="items")
            if table:
                break

        if table:
            try:
                df = pd.read_html(str(table))[0]
                dfs_league[league][date_str] = df
                print(f"Found items table for {league} {date_str} with
↳{len(df)} rows.")
            except Exception as e:
                print(f"Error parsing table for {league} {date_str}: {e}")
        else:
            print(f"No items table found for {league} {date_str}.")

```

Scraping data for La Liga:

Scraping: [https://www.transfermarkt.co.uk/la-](https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2011-03-15)

[liga/marktwerteverein/wettbewerb/ES1/stichtag/2011-03-15](https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2011-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2011-03-15 with 21 rows.

Scraping: [https://www.transfermarkt.co.uk/la-](https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2012-03-15)

[liga/marktwerteverein/wettbewerb/ES1/stichtag/2012-03-15](https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2012-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2012-03-15 with 21 rows.

Scraping: [https://www.transfermarkt.co.uk/la-](https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2013-03-15)

[liga/marktwerteverein/wettbewerb/ES1/stichtag/2013-03-15](https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2013-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```


Found items table for La Liga 2013-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2014-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2014-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2015-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2015-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2016-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2016-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2017-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2017-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2018-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2018-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2019-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2019-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2020-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2020-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2021-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2021-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2022-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2022-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2023-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2023-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/la-liga/marktwerteverein/wettbewerb/ES1/stichtag/2024-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
```

'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga 2024-03-15 with 21 rows.

Scraping data for Bundesliga:

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2011-03-15>

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga 2011-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2012-03-15>

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga 2012-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2013-03-15>

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga 2013-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2014-03-15>

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga 2014-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2015-03-15>

/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a

```

'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Bundesliga 2015-03-15 with 19 rows.
Scraping: https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2016-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Bundesliga 2016-03-15 with 19 rows.
Scraping: https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2017-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Bundesliga 2017-03-15 with 19 rows.
Scraping: https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2018-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Bundesliga 2018-03-15 with 19 rows.
Scraping: https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2019-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Bundesliga 2019-03-15 with 19 rows.
Scraping: https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2020-03-15

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

```

Found items table for Bundesliga 2020-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2021-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga 2021-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2022-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga 2022-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2023-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga 2023-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/bundesliga/marktwerteverein/wettbewerb/L1/stichtag/2024-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga 2024-03-15 with 19 rows.

Scraping data for Ligue 1:

Scraping: <https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2011-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2011-03-15 with 19 rows.

Scraping: [https://www.transfermarkt.co.uk/ligue-](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2012-03-15)

[1/marktwerteverein/wettbewerb/FR1/stichtag/2012-03-15](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2012-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2012-03-15 with 19 rows.

Scraping: [https://www.transfermarkt.co.uk/ligue-](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2013-03-15)

[1/marktwerteverein/wettbewerb/FR1/stichtag/2013-03-15](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2013-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2013-03-15 with 19 rows.

Scraping: [https://www.transfermarkt.co.uk/ligue-](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2014-03-15)

[1/marktwerteverein/wettbewerb/FR1/stichtag/2014-03-15](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2014-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2014-03-15 with 19 rows.

Scraping: [https://www.transfermarkt.co.uk/ligue-](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2015-03-15)

[1/marktwerteverein/wettbewerb/FR1/stichtag/2015-03-15](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2015-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2015-03-15 with 19 rows.

Scraping: [https://www.transfermarkt.co.uk/ligue-](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2016-03-15)

[1/marktwerteverein/wettbewerb/FR1/stichtag/2016-03-15](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2016-03-15)

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5

1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2016-03-15 with 19 rows.

Scraping: [https://www.transfermarkt.co.uk/ligue-](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2017-03-15)

[1/marktwerteverein/wettbewerb/FR1/stichtag/2017-03-15](https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2017-03-15)

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2017-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2018-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2018-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2019-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2019-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2020-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2020-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2021-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Ligue 1 2021-03-15 with 19 rows.

Scraping: <https://www.transfermarkt.co.uk/ligue-1/marktwerteverein/wettbewerb/FR1/stichtag/2022-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
```

```

'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Ligue 1 2022-03-15 with 19 rows.
Scraping: https://www.transfermarkt.co.uk/ligue-
1/marktwerteverein/wettbewerb/FR1/stichtag/2023-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Ligue 1 2023-03-15 with 19 rows.
Scraping: https://www.transfermarkt.co.uk/ligue-
1/marktwerteverein/wettbewerb/FR1/stichtag/2024-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Ligue 1 2024-03-15 with 19 rows.

Scraping data for Serie A:
Scraping: https://www.transfermarkt.co.uk/serie-
a/marktwerteverein/wettbewerb/IT1/stichtag/2011-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Serie A 2011-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/serie-
a/marktwerteverein/wettbewerb/IT1/stichtag/2012-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Serie A 2012-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/serie-
a/marktwerteverein/wettbewerb/IT1/stichtag/2013-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a

```



```

'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Serie A 2013-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/serie-
a/marktwerteverein/wettbewerb/IT1/stichtag/2014-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Serie A 2014-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/serie-
a/marktwerteverein/wettbewerb/IT1/stichtag/2015-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Serie A 2015-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/serie-
a/marktwerteverein/wettbewerb/IT1/stichtag/2016-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Serie A 2016-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/serie-
a/marktwerteverein/wettbewerb/IT1/stichtag/2017-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Serie A 2017-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/serie-
a/marktwerteverein/wettbewerb/IT1/stichtag/2018-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

```

Found items table for Serie A 2018-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/serie-a/marktwerteverein/wettbewerb/IT1/stichtag/2019-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Serie A 2019-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/serie-a/marktwerteverein/wettbewerb/IT1/stichtag/2020-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Serie A 2020-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/serie-a/marktwerteverein/wettbewerb/IT1/stichtag/2021-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Serie A 2021-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/serie-a/marktwerteverein/wettbewerb/IT1/stichtag/2022-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Serie A 2022-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/serie-a/marktwerteverein/wettbewerb/IT1/stichtag/2023-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Serie A 2023-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/serie-a/marktwerteverein/wettbewerb/IT1/stichtag/2024-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Serie A 2024-03-15 with 21 rows.

Scraping data for Premier League:

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2011-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2011-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2012-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2012-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2013-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2013-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2014-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2014-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2015-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2015-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2016-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2016-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2017-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2017-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2018-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2018-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2019-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League 2019-03-15 with 21 rows.

Scraping: <https://www.transfermarkt.co.uk/premier-league/marktwerteverein/wettbewerb/GB1/stichtag/2020-03-15>

```
/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
```

```

'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Premier League 2020-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2021-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Premier League 2021-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2022-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Premier League 2022-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2023-03-15
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

Found items table for Premier League 2023-03-15 with 21 rows.
Scraping: https://www.transfermarkt.co.uk/premier-
league/marktwerteverein/wettbewerb/GB1/stichtag/2024-03-15
Found items table for Premier League 2024-03-15 with 21 rows.
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/3923648800.py:5
1: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
df = pd.read_html(str(table))[0]

```

```

[111]: # 1. Create an output directory
output_dir = "league_values_csv"
os.makedirs(output_dir, exist_ok=True)

# 2. Loop through dfs_league and save each DataFrame
for league, league_data in dfs_league.items():
    for date_str, df in league_data.items():

```

```

# sanitize the date string for a filename
safe_date = date_str.replace("/", "-")
filename = f"{league}_values_{safe_date}.csv"
filepath = os.path.join(output_dir, filename)

# save without the index
df.to_csv(filepath, index=False)
print(f"Saved {filepath}")

```

```

Saved league_values_csv/La Liga_values_2011-03-15.csv
Saved league_values_csv/La Liga_values_2012-03-15.csv
Saved league_values_csv/La Liga_values_2013-03-15.csv
Saved league_values_csv/La Liga_values_2014-03-15.csv
Saved league_values_csv/La Liga_values_2015-03-15.csv
Saved league_values_csv/La Liga_values_2016-03-15.csv
Saved league_values_csv/La Liga_values_2017-03-15.csv
Saved league_values_csv/La Liga_values_2018-03-15.csv
Saved league_values_csv/La Liga_values_2019-03-15.csv
Saved league_values_csv/La Liga_values_2020-03-15.csv
Saved league_values_csv/La Liga_values_2021-03-15.csv
Saved league_values_csv/La Liga_values_2022-03-15.csv
Saved league_values_csv/La Liga_values_2023-03-15.csv
Saved league_values_csv/La Liga_values_2024-03-15.csv
Saved league_values_csv/Bundesliga_values_2011-03-15.csv
Saved league_values_csv/Bundesliga_values_2012-03-15.csv
Saved league_values_csv/Bundesliga_values_2013-03-15.csv
Saved league_values_csv/Bundesliga_values_2014-03-15.csv
Saved league_values_csv/Bundesliga_values_2015-03-15.csv
Saved league_values_csv/Bundesliga_values_2016-03-15.csv
Saved league_values_csv/Bundesliga_values_2017-03-15.csv
Saved league_values_csv/Bundesliga_values_2018-03-15.csv
Saved league_values_csv/Bundesliga_values_2019-03-15.csv
Saved league_values_csv/Bundesliga_values_2020-03-15.csv
Saved league_values_csv/Bundesliga_values_2021-03-15.csv
Saved league_values_csv/Bundesliga_values_2022-03-15.csv
Saved league_values_csv/Bundesliga_values_2023-03-15.csv
Saved league_values_csv/Bundesliga_values_2024-03-15.csv
Saved league_values_csv/Ligue 1_values_2011-03-15.csv
Saved league_values_csv/Ligue 1_values_2012-03-15.csv
Saved league_values_csv/Ligue 1_values_2013-03-15.csv
Saved league_values_csv/Ligue 1_values_2014-03-15.csv
Saved league_values_csv/Ligue 1_values_2015-03-15.csv
Saved league_values_csv/Ligue 1_values_2016-03-15.csv
Saved league_values_csv/Ligue 1_values_2017-03-15.csv
Saved league_values_csv/Ligue 1_values_2018-03-15.csv
Saved league_values_csv/Ligue 1_values_2019-03-15.csv
Saved league_values_csv/Ligue 1_values_2020-03-15.csv
Saved league_values_csv/Ligue 1_values_2021-03-15.csv

```

```

Saved league_values_csv/Ligue 1_values_2022-03-15.csv
Saved league_values_csv/Ligue 1_values_2023-03-15.csv
Saved league_values_csv/Ligue 1_values_2024-03-15.csv
Saved league_values_csv/Serie A_values_2011-03-15.csv
Saved league_values_csv/Serie A_values_2012-03-15.csv
Saved league_values_csv/Serie A_values_2013-03-15.csv
Saved league_values_csv/Serie A_values_2014-03-15.csv
Saved league_values_csv/Serie A_values_2015-03-15.csv
Saved league_values_csv/Serie A_values_2016-03-15.csv
Saved league_values_csv/Serie A_values_2017-03-15.csv
Saved league_values_csv/Serie A_values_2018-03-15.csv
Saved league_values_csv/Serie A_values_2019-03-15.csv
Saved league_values_csv/Serie A_values_2020-03-15.csv
Saved league_values_csv/Serie A_values_2021-03-15.csv
Saved league_values_csv/Serie A_values_2022-03-15.csv
Saved league_values_csv/Serie A_values_2023-03-15.csv
Saved league_values_csv/Serie A_values_2024-03-15.csv
Saved league_values_csv/Premier League_values_2011-03-15.csv
Saved league_values_csv/Premier League_values_2012-03-15.csv
Saved league_values_csv/Premier League_values_2013-03-15.csv
Saved league_values_csv/Premier League_values_2014-03-15.csv
Saved league_values_csv/Premier League_values_2015-03-15.csv
Saved league_values_csv/Premier League_values_2016-03-15.csv
Saved league_values_csv/Premier League_values_2017-03-15.csv
Saved league_values_csv/Premier League_values_2018-03-15.csv
Saved league_values_csv/Premier League_values_2019-03-15.csv
Saved league_values_csv/Premier League_values_2020-03-15.csv
Saved league_values_csv/Premier League_values_2021-03-15.csv
Saved league_values_csv/Premier League_values_2022-03-15.csv
Saved league_values_csv/Premier League_values_2023-03-15.csv
Saved league_values_csv/Premier League_values_2024-03-15.csv

```

```

[112]: # List of column names to drop
columns_to_drop = ['#', 'wappen', 'Club.1', 'Current value', '%', 'Unnamed: 8', 'Unnamed: 9' ] # Replace with your actual column names

# Loop through each league and its corresponding DataFrames in the nested dictionary
for league, league_data in dfs_league.items():
    for date, df in league_data.items():
        # Drop the columns and update the DataFrame in the nested dictionary
        # Using errors='ignore' ensures that if a column is missing, it won't raise an error
        dfs_league[league][date] = df.drop(columns=columns_to_drop, errors='ignore')

```

```

# Loop through each league and its corresponding DataFrames in the nested
↳ dictionary
for league, league_data in dfs_league.items():
    for date, df in league_data.items():
        # Check if the column 'League' exists and then rename it to the date
        if 'League' in df.columns:
            df.rename(columns={'League': "Value_" + date}, inplace=True)
        else:
            print(f"'League' column not found in DataFrame for {league} on
↳ {date}")

# New dictionary to store the subset DataFrames
dfs_league_cleaned = {}

for league, league_data in dfs_league.items():
    for date, df in league_data.items():
        # Construct the value column name based on the date
        value_col = "Value_" + date
        if 'Club' in df.columns and value_col in df.columns:
            # Select only the 'Club' and the 'Value_(date)' columns
            subset_df = df[['Club', value_col]].copy()
            if league not in dfs_league_cleaned:
                dfs_league_cleaned[league] = {}
            dfs_league_cleaned[league][date] = subset_df

# Start with an empty combined dataframe

combined_la_liga = None
combined_ligue_1 = None
combined_bundesliga = None
combined_serie_a = None
combined_premier_league = None

# Loop through each league and its corresponding DataFrames in the nested
↳ dictionary
for league, league_data in dfs_league_cleaned.items():
    for date, df in league_data.items():
        if league == 'La Liga':
            if combined_la_liga is None:
                combined_la_liga = df
            else:
                combined_la_liga = pd.merge(combined_la_liga, df, on='Club',
↳ how='outer')
        elif league == 'Ligue 1':
            if combined_ligue_1 is None:
                combined_ligue_1 = df

```



```

        else:
            combined_ligue_1 = pd.merge(combined_ligue_1, df, on='Club',
↳how='outer')
        elif league == 'Bundesliga':
            if combined_bundesliga is None:
                combined_bundesliga = df
            else:
                combined_bundesliga = pd.merge(combined_bundesliga, df,
↳on='Club', how='outer')
        elif league == 'Serie A':
            if combined_serie_a is None:
                combined_serie_a = df
            else:
                combined_serie_a = pd.merge(combined_serie_a, df, on='Club',
↳how='outer')
        elif league == 'Premier League':
            if combined_premier_league is None:
                combined_premier_league = df
            else:
                combined_premier_league = pd.merge(combined_premier_league, df,
↳on='Club', how='outer')

# realised that there was a unknown row in the combined dataframe that was
↳causing issues with the plotting
combined_ligue_1_clean = combined_ligue_1.drop(18)
combined_la_liga_clean = combined_la_liga.drop(17)
combined_bundesliga_clean = combined_bundesliga.drop(15)
combined_serie_a_clean = combined_serie_a.drop(17)
combined_premier_league_clean = combined_premier_league.drop(17)

def convert_value(value_str):
    """
    Convert a monetary string (e.g., "€310.75m", "€1.19bn") into a numeric
↳value.
    """
    if isinstance(value_str, str):
        # Remove the euro symbol and extra spaces, then convert to lower case
        value_str = value_str.replace("€", "").strip().lower()
        if "m" in value_str:
            try:
                # Remove "m", convert to float, and multiply by 1e6
                return float(value_str.replace("m", "")) * 1_000_000

```

```

        except:
            return None
    elif "bn" in value_str:
        try:
            # Remove "bn", convert to float, and multiply by 1e9
            return float(value_str.replace("bn", "")) * 1_000_000_000
        except:
            return None
    else:
        try:
            return float(value_str)
        except:
            return None
    return value_str

# List of league DataFrames
league_dfs1 = [
    combined_ligue_1_clean,
    combined_la_liga_clean,
    combined_bundesliga_clean,
    combined_serie_a_clean,
    combined_premier_league_clean
]

# Loop through each league DataFrame and convert the columns starting with
↳ "Value_"
for df in league_dfs1:
    for col in df.columns:
        if col.startswith("Value_"):
            df[col] = df[col].apply(convert_value)

league_dfs1[0] = combined_ligue_1_clean
league_dfs1[1] = combined_la_liga_clean
league_dfs1[2] = combined_bundesliga_clean
league_dfs1[3] = combined_serie_a_clean
league_dfs1[4] = combined_premier_league_clean

# Transpose and prepare data for Ligue 1
df_transposed_ligue_1 = combined_ligue_1_clean.set_index('Club').transpose()
df_transposed_ligue_1.index = pd.to_datetime(df_transposed_ligue_1.index.str.
↳ replace("Value_", "", regex=True))
df_transposed_ligue_1 = df_transposed_ligue_1.sort_index()

# Transpose and prepare data for La Liga
df_transposed_la_liga = combined_la_liga_clean.set_index('Club').transpose()

```

```

df_transposed_la_liga.index = pd.to_datetime(df_transposed_la_liga.index.str.
    ↪replace("Value_", "", regex=True))
df_transposed_la_liga = df_transposed_la_liga.sort_index()

# Transpose and prepare data for Bundesliga
df_transposed_bundesliga = combined_bundesliga_clean.set_index('Club').
    ↪transpose()
df_transposed_bundesliga.index = pd.to_datetime(df_transposed_bundesliga.index.
    ↪str.replace("Value_", "", regex=True))
df_transposed_bundesliga = df_transposed_bundesliga.sort_index()

# Transpose and prepare data for Serie A
df_transposed_serie_a = combined_serie_a_clean.set_index('Club').transpose()
df_transposed_serie_a.index = pd.to_datetime(df_transposed_serie_a.index.str.
    ↪replace("Value_", "", regex=True))
df_transposed_serie_a = df_transposed_serie_a.sort_index()

# Transpose and prepare data for Premier League
df_transposed_premier_league = combined_premier_league_clean.set_index('Club').
    ↪transpose()
df_transposed_premier_league.index = pd.
    ↪to_datetime(df_transposed_premier_league.index.str.replace("Value_", "",
    ↪regex=True))
df_transposed_premier_league = df_transposed_premier_league.sort_index()

```

```

[113]: # ----- Custom Y-Axis Formatter ----- #
def custom_y_formatter(x, pos):
    if x < 1e9:
        return f"{x/1e6:,.0f} million"
    else:
        return f"{x/1e9:,.1f} billion"

# ----- Set Seaborn Style ----- #
sns.set_style("whitegrid")
sns.set_palette("Set2")

# ----- Function to Plot League Average ----- ↪
    ↪#
def plot_league_avg(ax, df, label):
    # Compute the average value across all clubs for each date (index)
    avg_series = df.mean(axis=1)
    # Convert the datetime index to numeric values for spline interpolation.
    x_dates = mdates.date2num(avg_series.index.to_pydatetime())
    y = avg_series.values

```

```

    # If there are at least 3 points, use cubic spline interpolation for a
    ↪ smooth curve.
    if len(x_dates) >= 3:
        spline = make_interp_spline(x_dates, y, k=3)
        x_dense = np.linspace(x_dates.min(), x_dates.max(), 300)
        y_smooth = spline(x_dense)
        x_dense_dates = mdates.num2date(x_dense)
        ax.plot(x_dense_dates, y_smooth, label=label, linewidth=2)
    else:
        ax.plot(avg_series.index, y, marker='o', label=label, linewidth=2)

# ----- Create Figure and Axis ----- #
fig, ax = plt.subplots(figsize=(12, 6))

# ----- Plot the Average for Each League ↪
    ↪ ----- #
plot_league_avg(ax, df_transposed_ligue_1, "Ligue 1")
plot_league_avg(ax, df_transposed_la_liga, "La Liga")
plot_league_avg(ax, df_transposed_bundesliga, "Bundesliga")
plot_league_avg(ax, df_transposed_serie_a, "Serie A")
plot_league_avg(ax, df_transposed_premier_league, "Premier League")

# ----- Highlight Significant Financial Events ↪
    ↪ ----- #
# Define events with their year and label
events = [
    {"year": 2011, "label": "UEFA FFP introduced"},
    {"year": 2020, "label": "COVID-19 Pandemic"}
]

for event in events:
    # Define the start and end of the event year.
    start_date = datetime.datetime(event['year'], 1, 1)
    end_date = datetime.datetime(event['year'], 12, 31)
    # Shade the entire year with a semi-transparent gray.
    ax.axvspan(start_date, end_date, color='gray', alpha=0.2)
    # Add a vertical annotation for the event.
    ax.text(start_date, ax.get_ylim()[1]*0.95, event['label'], rotation=90,
            verticalalignment='top', fontsize=10, color='black')

# ----- Format X-Axis ----- #
ax.xaxis.set_major_locator(mdates.YearLocator())
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
ax.tick_params(axis='x', rotation=45)

# ----- Format Y-Axis ----- #
ax.ticklabel_format(axis='y', style='plain', useOffset=False)

```

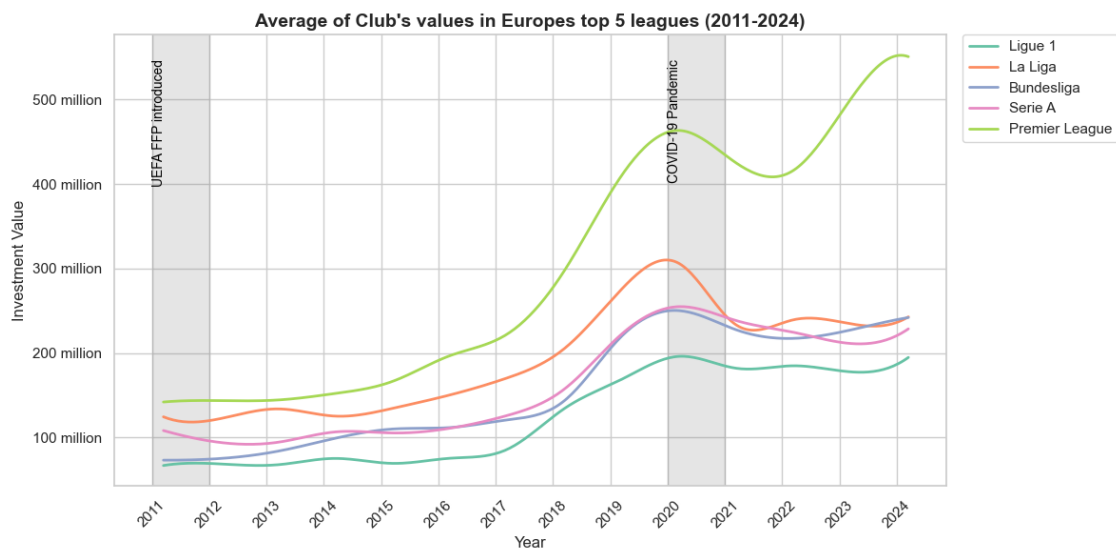
```

ax.get_yaxis().get_major_formatter().set_scientific(False)
ax.yaxis.set_major_locator(ticker.MaxNLocator(6))
ax.yaxis.set_major_formatter(ticker.FuncFormatter(custom_y_formatter))

# ----- Set Title, Labels, and Legend ----- #
ax.set_title("Average of Club's values in Europe's top 5 leagues (2011-2024)",
    ↳ fontsize=14, fontweight='bold')
ax.set_xlabel("Year", fontsize=12)
ax.set_ylabel("Investment Value", fontsize=12)
ax.legend(loc='upper left', bbox_to_anchor=(1.02, 1), borderaxespad=0)

plt.tight_layout()
fig.savefig("Average_of_clubs_values_europe_top_5.png", dpi=300,
    ↳ bbox_inches="tight")
plt.show()

```



```

[114]: import requests
import pandas as pd
from bs4 import BeautifulSoup, Comment

# Dictionary mapping leagues to their revenue/expenditure URL
base_urls_2 = {
    "Premier League": "https://www.transfermarkt.co.uk/premier-league/
    ↳ einnahmenausgaben/wettbewerb/GB1/plus/0?
    ↳ ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0",
    "La Liga": "https://www.transfermarkt.co.uk/laliga/einnahmenausgaben/
    ↳ wettbewerb/ES1/plus/0?
    ↳ ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0",

```

```

    "Serie A": "https://www.transfermarkt.co.uk/serie-a/einnahmenausgaben/
↳wettbewerb/IT1/plus/0?
↳ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0",
    "Bundesliga": "https://www.transfermarkt.co.uk/bundesliga/einnahmenausgaben/
↳wettbewerb/L1/plus/0?
↳ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0",
    "Ligue 1": "https://www.transfermarkt.co.uk/ligue-1/einnahmenausgaben/
↳wettbewerb/FR1/plus/0?
↳ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0"
}

# Dictionary to store DataFrames for each league
dfs_income_vs_expenditure = {}

# Custom headers to mimic a real browser
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/
↳20100101 Firefox/109.0",
    "Accept-Language": "en-US,en;q=0.5"
}

# Loop through each league URL and scrape the table
for league, url in base_urls_2.items():
    print(f"\nScraping data for {league}:")
    print(f"Scraping: {url}")

    response = requests.get(url, headers=headers)
    if response.status_code != 200:
        print(f"Error fetching page for {league}: {response.status_code}")
        continue

    soup = BeautifulSoup(response.text, 'html.parser')

    # Try to find the table with class "items"
    table = soup.find('table', class_="items")

    # If not found, look inside HTML comments
    if not table:
        comments = soup.find_all(string=lambda text: isinstance(text, Comment))
        for comment in comments:
            comment_soup = BeautifulSoup(comment, "html.parser")
            table = comment_soup.find('table', class_="items")
            if table:
                break

    if table:
        try:

```

```

df = pd.read_html(str(table))[0]
dfs_income_vs_expenditure[league] = df
print(f"Found items table for {league} with {len(df)} rows.")
except Exception as e:
    print(f"Error parsing table for {league}: {e}")
else:
    print(f"No items table found for {league}.")

```

Scraping data for Premier League:

Scraping: https://www.transfermarkt.co.uk/premier-league/einnahmenausgaben/wettbewerb/GB1/plus/0?ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0

/var/folders/s1/319x1czx38l8ctmgdlmpsctw0000gn/T/ipykernel_86336/1404027738.py:49: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League with 25 rows.

Scraping data for La Liga:

Scraping: https://www.transfermarkt.co.uk/laliga/einnahmenausgaben/wettbewerb/ES1/plus/0?ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0

/var/folders/s1/319x1czx38l8ctmgdlmpsctw0000gn/T/ipykernel_86336/1404027738.py:49: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga with 25 rows.

Scraping data for Serie A:

Scraping: https://www.transfermarkt.co.uk/serie-a/einnahmenausgaben/wettbewerb/IT1/plus/0?ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0

/var/folders/s1/319x1czx38l8ctmgdlmpsctw0000gn/T/ipykernel_86336/1404027738.py:49: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Serie A with 25 rows.

Scraping data for Bundesliga:

Scraping: <https://www.transfermarkt.co.uk/bundesliga/einnahmenausgaben/wettbewerb>

```
b/L1/plus/0?ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=
&leihe=&intern=0
```

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/1404027738.py:4
9: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

Found items table for Bundesliga with 25 rows.

Scraping data for Ligue 1:

Scraping: [https://www.transfermarkt.co.uk/ligue-](https://www.transfermarkt.co.uk/ligue-1/einnahmenausgaben/wettbewerb/FR1/plus/0?ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0)

[1/einnahmenausgaben/wettbewerb/FR1/plus/0?ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0](https://www.transfermarkt.co.uk/ligue-1/einnahmenausgaben/wettbewerb/FR1/plus/0?ids=a&sa=&saison_id=&saison_id_bis=2024&nat=&pos=&altersklasse=&w_s=&leihe=&intern=0)

Found items table for Ligue 1 with 25 rows.

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/1404027738.py:4
9: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

```
[115]: # 1. Create an output directory
output_dir = "income_vs_expenditure_csv"
os.makedirs(output_dir, exist_ok=True)

# 2. Loop through dfs_income_vs_expenditure and save each DataFrame
for date_str, df in dfs_income_vs_expenditure.items():
    # sanitize the date string for a filename
    safe_date = date_str.replace("/", "-")
    filename = f"income_vs_expenditure_{safe_date}.csv"
    filepath = os.path.join(output_dir, filename)

    # save without the index
    df.to_csv(filepath, index=False)
    print(f"Saved {filepath}")
```

Saved income_vs_expenditure_csv/income_vs_expenditure_Premier League.csv

Saved income_vs_expenditure_csv/income_vs_expenditure_La Liga.csv

Saved income_vs_expenditure_csv/income_vs_expenditure_Serie A.csv

Saved income_vs_expenditure_csv/income_vs_expenditure_Bundesliga.csv

Saved income_vs_expenditure_csv/income_vs_expenditure_Ligue 1.csv

```
[116]: dfs_income_vs_expenditure['Premier League'].head(30)
```

```
[116]:      #  Club              Club.1  Club.2  Expenditure  Arrivals  \
0    1   NaN              Chelsea FC  €3.93bn          1111  €2.02bn
1    2   NaN      Manchester City  €3.09bn          1090  €1.26bn
```


2	3	NaN	Manchester United	€2.86bn	962	€997.86m
3	4	NaN	Liverpool FC	€2.20bn	1068	€1.29bn
4	5	NaN	Tottenham Hotspur	€2.18bn	989	€1.18bn
5	6	NaN	Arsenal FC	€2.05bn	895	€986.63m
6	7	NaN	Newcastle United	€1.53bn	1298	€807.82m
7	8	NaN	Aston Villa	€1.49bn	955	€896.05m
8	9	NaN	West Ham United	€1.44bn	982	€832.53m
9	10	NaN	Everton FC	€1.29bn	938	€992.79m
10	11	NaN	Southampton FC	€1.01bn	957	€910.39m
11	12	NaN	Wolverhampton Wanderers	€994.87m	1124	€633.67m
12	13	NaN	Leicester City	€902.53m	1027	€693.21m
13	14	NaN	Brighton & Hove Albion	€834.24m	988	€535.47m
14	15	NaN	Fulham FC	€818.90m	1293	€407.18m
15	16	NaN	Leeds United	€734.28m	1065	€630.58m
16	17	NaN	Crystal Palace	€685.16m	959	€375.49m
17	18	NaN	AFC Bournemouth	€644.04m	820	€285.67m
18	19	NaN	Nottingham Forest	€640.04m	1124	€396.79m
19	20	NaN	Sunderland AFC	€575.96m	1129	€351.15m
20	21	NaN	Middlesbrough FC	€545.77m	977	€419.28m
21	22	NaN	Burnley FC	€481.44m	853	€373.61m
22	23	NaN	West Bromwich Albion	€443.56m	858	€276.33m
23	24	NaN	Watford FC	€438.35m	1011	€487.08m
24	25	NaN	Stoke City	€422.22m	1007	€189.76m

		Income	Departures	Balance
0	1073	€-1,912.66m	NaN	
1	1080	€-1,831.05m	NaN	
2	942	€-1,863.43m	NaN	
3	1041	€-904.43m	NaN	
4	970	€-1,005.51m	NaN	
5	892	€-1,058.56m	NaN	
6	1284	€-724.13m	NaN	
7	943	€-594.20m	NaN	
8	961	€-611.34m	NaN	
9	924	€-300.73m	NaN	
10	934	€-103.44m	NaN	
11	1108	€-361.20m	NaN	
12	1011	€-209.32m	NaN	
13	955	€-298.78m	NaN	
14	1270	€-411.72m	NaN	
15	1051	€-103.70m	NaN	
16	939	€-309.67m	NaN	
17	797	€-358.37m	NaN	
18	1117	€-243.25m	NaN	
19	1122	€-224.81m	NaN	
20	962	€-126.49m	NaN	
21	838	€-107.83m	NaN	

22	844	€-167.23m	NaN
23	995	€48.73m	NaN
24	989	€-232.46m	NaN

```
[117]: # List of column names to drop
columns_to_drop = ['#', 'Club', 'Arrivals', 'Income', 'Departures', 'Balance' ]

# Loop through each league and its corresponding DataFrames in the nested
↳ dictionary
for league, df in dfs_income_vs_expenditure.items():
    # Drop the columns and update the DataFrame in the nested dictionary
    # Using errors='ignore' ensures that if a column is missing, it won't
    ↳ raise an error
    dfs_income_vs_expenditure[league] = df.drop(columns=columns_to_drop,
    ↳ errors='ignore')
```

```
[118]: dfs_income_vs_expenditure['Ligue 1'].head(30)
```

```
[118]:
```

	Club.1	Club.2	Expenditure
0	Paris Saint-Germain	€2.61bn	756
1	AS Monaco	€1.40bn	917
2	Olympique Lyon	€1.10bn	713
3	Olympique Marseille	€1.10bn	1077
4	Stade Rennais FC	€852.52m	863
5	LOSC Lille	€503.66m	737
6	OGC Nice	€432.88m	818
7	FC Girondins Bordeaux	€327.50m	801
8	RC Lens	€304.23m	795
9	AS Saint-Étienne	€242.22m	768
10	RC Strasbourg Alsace	€219.12m	790
11	FC Lorient	€184.83m	586
12	FC Nantes	€177.52m	672
13	FC Toulouse	€177.08m	633
14	Stade Reims	€161.89m	675
15	Montpellier HSC	€103.96m	605
16	FC Metz	€86.43m	900
17	AJ Auxerre	€82.38m	525
18	ESTAC Troyes	€71.96m	584
19	FC Sochaux-Montbéliard	€60.25m	734
20	Stade Brestois 29	€59.50m	508
21	Angers SCO	€53.62m	740
22	Dijon FCO	€51.49m	383
23	SM Caen	€50.49m	566
24	Amiens SC	€46.26m	615

```
[119]: # Loop through each league and its corresponding DataFrames in the nested
↳ dictionary
```

```

for league, df in dfs_income_vs_expenditure.items():
    # Check if the column 'Club.1' exists and then rename it to the date
    if 'Club.1' in df.columns:
        df.rename(columns={'Club.1': "Club"}, inplace=True)
    if 'Club.2' in df.columns:
        df.rename(columns={'Club.2': "Club's Expenditure"}, inplace=True)
    if 'Expenditure' in df.columns:
        df.rename(columns={'Expenditure': "Arrivals"}, inplace=True)

```

```

[120]: # only keep the first 15 rows in each league df
# Iterate through each DataFrame in the dictionary and slice it
for league, df in dfs_income_vs_expenditure.items():
    dfs_income_vs_expenditure[league] = df.iloc[:15]

```

```

[121]: # Convert the "Club's Expenditure" column to numeric values
for league, df in dfs_income_vs_expenditure.items():
    dfs_income_vs_expenditure[league]["Club's Expenditure"] =
    ↪dfs_income_vs_expenditure[league]["Club's Expenditure"].apply(convert_value)

```

```

[122]: dfs_income_vs_expenditure['La Liga'].head(30)

```

```

[122]:

```

	Club	Club's Expenditure	Arrivals
0	FC Barcelona	2.780000e+09	1069
1	Real Madrid	2.730000e+09	910
2	Atlético de Madrid	1.920000e+09	996
3	Valencia CF	1.080000e+09	841
4	Sevilla FC	9.960100e+08	866
5	Villarreal CF	7.594100e+08	580
6	Real Betis Balompié	6.045800e+08	796
7	Real Sociedad	3.730900e+08	648
8	Deportivo de La Coruña	3.648200e+08	841
9	Celta de Vigo	3.284500e+08	758
10	RCD Espanyol Barcelona	2.788800e+08	1054
11	Athletic Bilbao	2.151600e+08	634
12	Getafe CF	2.135800e+08	538
13	UD Almería	2.023700e+08	518
14	Real Zaragoza	1.991300e+08	837

```

[123]: # ----- Compute Average Expenditure ----- #
league_avg_expenditure = {}
for league, df in dfs_income_vs_expenditure.items():
    if "Club's Expenditure" in df.columns:
        # Compute the mean expenditure for each league (ignoring missing values)
        league_avg_expenditure[league] = df["Club's Expenditure"].mean()

df_avg = pd.DataFrame(
    list(league_avg_expenditure.items()),

```

```

        columns=["League", "AvgExpenditure"]
    )
    df_avg.sort_values("AvgExpenditure", inplace=True)

    # ----- Compute Average Arrivals ----- #
    league_avg_arrivals = {}
    for league, df in dfs_income_vs_expenditure.items():
        if "Arrivals" in df.columns:
            # Compute the mean arrivals for each league (ignoring missing values)
            league_avg_arrivals[league] = df["Arrivals"].mean()

    df_arrivals = pd.DataFrame(
        list(league_avg_arrivals.items()),
        columns=["League", "AvgArrivals"]
    )
    # Ensure the order of arrivals matches expenditure (using League as key)
    df_arrivals = df_arrivals.set_index("League").loc[df_avg["League"]].
        ↪reset_index()

    # ----- Plotting the Bar Chart ----- #
    sns.set_theme(style="whitegrid")
    plt.figure(figsize=(10, 6))
    ax = sns.barplot(x="League", y="AvgExpenditure", data=df_avg, palette="Set2")

    ax.set_title("Average Club Expenditure by League with Average Arrivals",
        ↪fontsize=16)
    ax.set_xlabel("League", fontsize=14)
    ax.set_ylabel("Average Expenditure (€)", fontsize=14)

    # ----- Custom Y-Axis Formatter for Expenditure ----- #
    ↪----- #
    def custom_y_formatter(x, pos):
        if x < 1e9:
            return f"{x/1e6:,.0f}M"
        else:
            return f"{x/1e9:,.1f}B"

    ax.yaxis.set_major_formatter(plt.FuncFormatter(custom_y_formatter))
    plt.xticks(rotation=45)
    plt.tight_layout()

    # Annotate each bar with its expenditure value
    for p in ax.patches:
        height = p.get_height()
        label = custom_y_formatter(height, 0)
        ax.text(
            p.get_x() + p.get_width() / 2,

```

```

        height - (height * 0.1),
        label,
        ha='center',
        va='top',
        fontsize=12,
        color='black'
    )

# ----- Overlay Smooth Line Plot for Average Arrivals
# ----- #
# Create a secondary y-axis for arrivals
ax2 = ax.twinx()
ax2.set_ylabel("Average Arrivals (Count)", fontsize=14, color='black')
ax2.tick_params(axis='y', labelcolor='black')
ax2.grid(False) # Disable gridlines on the secondary axis

# Convert categorical league names to numeric positions
leagues = df_avg["League"].tolist()
x_numeric = np.arange(len(leagues))
y_arrivals = df_arrivals["AvgArrivals"].values

# Create a smooth (soft) line using spline interpolation if possible
if len(x_numeric) >= 3:
    x_dense = np.linspace(x_numeric.min(), x_numeric.max(), 300)
    spline = make_interp_spline(x_numeric, y_arrivals, k=2) # quadratic spline
    # for smoothing
    y_dense = spline(x_dense)
    ax2.plot(x_dense, y_dense, color='blue', linewidth=1.5, label="Avg
    # Arrivals")
else:
    ax2.plot(x_numeric, y_arrivals, color='blue', marker='o', linewidth=1.5,
    # label="Avg Arrivals")

# Overlay the original data points
ax2.scatter(x_numeric, y_arrivals, color='black', marker='o', s=40)

# Set x-ticks on the secondary axis to match the league names
ax2.set_xticks(x_numeric)
ax2.set_xticklabels(leagues)

# Compute a vertical offset (3% of the y-axis range) to shift the text upward
offset = (ax2.get_ylim()[1] - ax2.get_ylim()[0]) * 0.03

# Annotate each point on the arrivals line with its value (shifted upward)
# Also, shift the annotation horizontally for Premier League
for x_val, y_val, league in zip(x_numeric, y_arrivals, leagues):
    x_offset = 0.0

```

```

if league == "Premier League":
    x_offset = 0.15 # Adjust this value as needed
    ax2.text(x_val + x_offset, y_val + offset, f"{y_val:.1f}", ha='center',
    ↪va='bottom', fontsize=10, color='black')

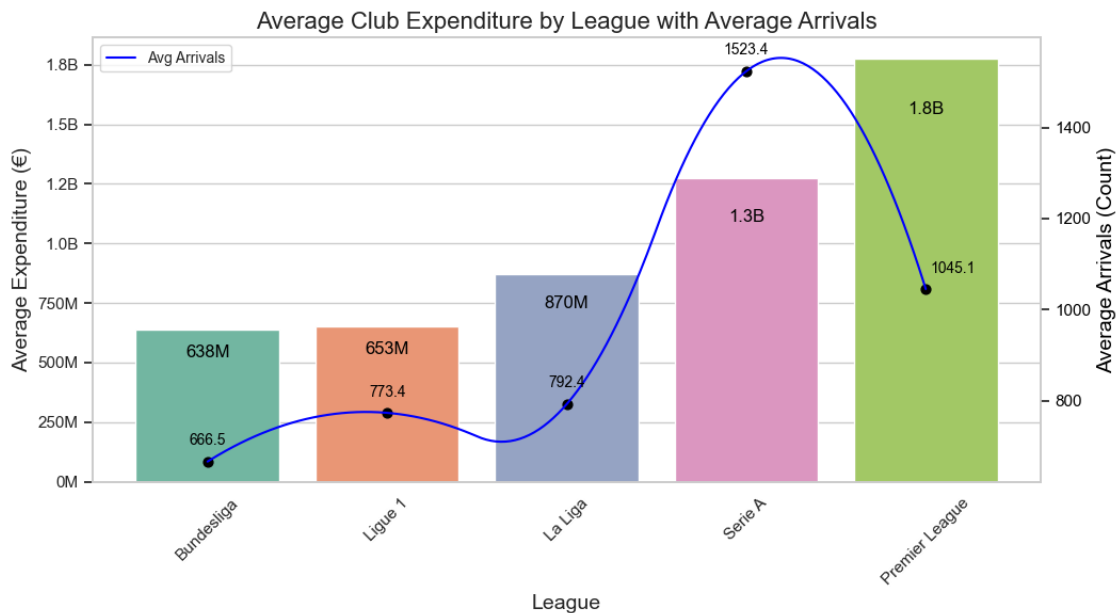
# Add a legend for the arrivals line (optional)
ax2.legend(loc='upper left', fontsize=10)
plt.savefig("Average_club_expenditure_with_league_arrivals.png", dpi=300,
    ↪bbox_inches="tight")
plt.show()

```

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/1570610478.py:3
1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
ax = sns.barplot(x="League", y="AvgExpenditure", data=df_avg, palette="Set2")
```



```

[124]: # create the base url's which i can use as a base and add the dates and leagues
    ↪to so i can access the different pages
base_urls_3 = {
    "Premier League": "https://www.transfermarkt.co.uk/premier-league/
    ↪besucherzahlen/wettbewerb/GB1/plus/1?saizon_id=2023",
    "La Liga": "https://www.transfermarkt.co.uk/laliga/besucherzahlen/
    ↪wettbewerb/ES1/plus/1?saizon_id=2023",

```

```

    "Serie A": "https://www.transfermarkt.co.uk/serie-a/besucherzahlen/
↳wettbewerb/IT1/plus/1?saison_id=2023",
    "Bundesliga": "https://www.transfermarkt.co.uk/bundesliga/besucherzahlen/
↳wettbewerb/L1/plus/1?saison_id=2023",
    "Ligue 1": "https://www.transfermarkt.co.uk/ligue-1/besucherzahlen/
↳wettbewerb/FR1/plus/1?saison_id=2023"
}

# Dictionary to store DataFrames for each league
dfs_stadium_capacity = {}

# Custom headers to mimic a real browser
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/
↳20100101 Firefox/109.0",
    "Accept-Language": "en-US,en;q=0.5"
}

# Loop through each league URL and scrape the table
for league, url in base_urls_3.items():
    print(f"\nScraping data for {league}:")
    print(f"Scraping: {url}")

    response = requests.get(url, headers=headers)
    if response.status_code != 200:
        print(f"Error fetching page for {league}: {response.status_code}")
        continue

    soup = BeautifulSoup(response.text, 'html.parser')

    # First, try to find the table with class "items"
    table = soup.find('table', class_="items")

    # If not found, look inside HTML comments
    if not table:
        comments = soup.find_all(string=lambda text: isinstance(text, Comment))
        for comment in comments:
            comment_soup = BeautifulSoup(comment, "html.parser")
            table = comment_soup.find('table', class_="items")
            if table:
                break

    if table:
        try:
            df = pd.read_html(str(table))[0]
            dfs_stadium_capacity[league] = df
            print(f"Found items table for {league} with {len(df)} rows.")

```

```

except Exception as e:
    print(f"Error parsing table for {league}: {e}")
else:
    print(f"No items table found for {league}.")

```

Scraping data for Premier League:

Scraping: https://www.transfermarkt.co.uk/premier-league/besucherzahlen/wettbewerb/GB1/plus/1?saison_id=2023

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/1974918671.py:45: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Premier League with 61 rows.

Scraping data for La Liga:

Scraping: https://www.transfermarkt.co.uk/laliga/besucherzahlen/wettbewerb/ES1/plus/1?saison_id=2023

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/1974918671.py:45: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for La Liga with 61 rows.

Scraping data for Serie A:

Scraping: https://www.transfermarkt.co.uk/serie-a/besucherzahlen/wettbewerb/IT1/plus/1?saison_id=2023

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/1974918671.py:45: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```

Found items table for Serie A with 61 rows.

Scraping data for Bundesliga:

Scraping: https://www.transfermarkt.co.uk/bundesliga/besucherzahlen/wettbewerb/L1/plus/1?saison_id=2023

/var/folders/s1/319x1czx38l8ctmgdlnpsctw0000gn/T/ipykernel_86336/1974918671.py:45: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.

```
df = pd.read_html(str(table))[0]
```


Found items table for Bundesliga with 55 rows.

Scraping data for Ligue 1:

Scraping: https://www.transfermarkt.co.uk/ligue-1/besucherzahlen/wettbewerb/FR1/plus/1?saaison_id=2023

Found items table for Ligue 1 with 55 rows.

```
/var/folders/s1/319x1czx38l8ctmgdlnpstcw0000gn/T/ipykernel_86336/1974918671.py:4
5: FutureWarning: Passing literal html to 'read_html' is deprecated and will be
removed in a future version. To read from a literal string, wrap it in a
'StringIO' object.
```

```
df = pd.read_html(str(table))[0]
```

```
[125]: # 1. Create an output directory
output_dir = "stadium_capacity_csv"
os.makedirs(output_dir, exist_ok=True)

# 2. Loop through dfs_stadium_capacity and save each DataFrame
for key, df in dfs_stadium_capacity.items():
    # sanitize the key for a filename
    safe_key = key.replace("/", "-")
    filename = f"stadium_capacity_{safe_key}.csv"
    filepath = os.path.join(output_dir, filename)

    # save without the index
    df.to_csv(filepath, index=False)
    print(f"Saved {filepath}")
```

Saved stadium_capacity_csv/stadium_capacity_Premier League.csv

Saved stadium_capacity_csv/stadium_capacity_La Liga.csv

Saved stadium_capacity_csv/stadium_capacity_Serie A.csv

Saved stadium_capacity_csv/stadium_capacity_Bundesliga.csv

Saved stadium_capacity_csv/stadium_capacity_Ligue 1.csv

```
[126]: dfs_stadium_capacity['Premier League'].head(30)
```

```
[126]:
```

	#	Stadium	Capacity	Spectators	\
0	1.0	Old TraffordManchester United	74.879	1.397.148	
1	NaN	Old Trafford	NaN	NaN	
2	NaN	Manchester United	NaN	NaN	
3	2.0	London StadiumWest Ham United	62.500	1.186.367	
4	NaN	London Stadium	NaN	NaN	
5	NaN	West Ham United	NaN	NaN	
6	3.0	Tottenham Hotspur StadiumTottenham Hotspur	62.850	1.167.725	
7	NaN	Tottenham Hotspur Stadium	NaN	NaN	
8	NaN	Tottenham Hotspur	NaN	NaN	
9	4.0	Emirates StadiumArsenal FC	60.704	1.144.488	
10	NaN	Emirates Stadium	NaN	NaN	

11	NaN	Arsenal FC	NaN	NaN
12	5.0	AnfieldLiverpool FC	61.276	1.060.362
13	NaN	Anfield	NaN	NaN
14	NaN	Liverpool FC	NaN	NaN
15	6.0	Etihad StadiumManchester City	55.097	1.012.469
16	NaN	Etihad Stadium	NaN	NaN
17	NaN	Manchester City	NaN	NaN
18	7.0	St James' ParkNewcastle United	52.258	990.904
19	NaN	St James' Park	NaN	NaN
20	NaN	Newcastle United	NaN	NaN
21	8.0	Villa ParkAston Villa	42.918	796.501
22	NaN	Villa Park	NaN	NaN
23	NaN	Aston Villa	NaN	NaN
24	9.0	Stamford BridgeChelsea FC	41.631	751.952
25	NaN	Stamford Bridge	NaN	NaN
26	NaN	Chelsea FC	NaN	NaN
27	10.0	Goodison ParkEverton FC	39.414	741.793
28	NaN	Goodison Park	NaN	NaN
29	NaN	Everton FC	NaN	NaN

	Average	Matches	sold out	Capacity.1
0	73.534	19.0	-	98.2 %
1	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN
3	62.440	19.0	-	99.9 %
4	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN
6	61.459	19.0	-	97.8 %
7	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN
9	60.236	19.0	-	99.2 %
10	NaN	NaN	NaN	NaN
11	NaN	NaN	NaN	NaN
12	55.809	19.0	-	91.1 %
13	NaN	NaN	NaN	NaN
14	NaN	NaN	NaN	NaN
15	53.288	19.0	1	96.7 %
16	NaN	NaN	NaN	NaN
17	NaN	NaN	NaN	NaN
18	52.153	19.0	-	99.8 %
19	NaN	NaN	NaN	NaN
20	NaN	NaN	NaN	NaN
21	41.921	19.0	-	97.7 %
22	NaN	NaN	NaN	NaN
23	NaN	NaN	NaN	NaN
24	39.576	19.0	-	95.1 %
25	NaN	NaN	NaN	NaN

26	NaN	NaN	NaN	NaN
27	39.042	19.0	-	99.1 %
28	NaN	NaN	NaN	NaN
29	NaN	NaN	NaN	NaN

```
[127]: for league, df in dfs_stadium_capacity.items():
        # Drop the columns and update the DataFrame in the nested dictionary
        # Using errors='ignore' ensures that if a column is missing, it won't raise
        ↪an error
        dfs_stadium_capacity[league] = df.dropna()
```

```
[129]: for league, df in dfs_stadium_capacity.items():
        # Drop the columns and update the DataFrame in the nested dictionary
        # Using errors='ignore' ensures that if a column is missing, it won't raise
        ↪an error
        dfs_stadium_capacity[league]["Capacity.1"] = pd.
        ↪to_numeric(dfs_stadium_capacity[league]["Capacity.1"].str.rstrip('%'),
        ↪errors='coerce')
```

```
[132]: # 1. Create (or ensure) an output directory
        output_dir = "league_values_csv"
        os.makedirs(output_dir, exist_ok=True)

        # 2. Loop through dfs_league and save each DataFrame
        for league, league_data in dfs_league.items():
            for date_str, df in league_data.items():
                # sanitize the date string for a filename
                safe_date = date_str.replace("/", "-")
                filename = f"{league}_values_{safe_date}.csv"
                filepath = os.path.join(output_dir, filename)

                # save without the index
                df.to_csv(filepath, index=False)
                print(f"Saved {filepath}")
```

```
Saved league_values_csv/La Liga_values_2011-03-15.csv
Saved league_values_csv/La Liga_values_2012-03-15.csv
Saved league_values_csv/La Liga_values_2013-03-15.csv
Saved league_values_csv/La Liga_values_2014-03-15.csv
Saved league_values_csv/La Liga_values_2015-03-15.csv
Saved league_values_csv/La Liga_values_2016-03-15.csv
Saved league_values_csv/La Liga_values_2017-03-15.csv
Saved league_values_csv/La Liga_values_2018-03-15.csv
Saved league_values_csv/La Liga_values_2019-03-15.csv
Saved league_values_csv/La Liga_values_2020-03-15.csv
Saved league_values_csv/La Liga_values_2021-03-15.csv
Saved league_values_csv/La Liga_values_2022-03-15.csv
Saved league_values_csv/La Liga_values_2023-03-15.csv
```

[illegible]

```
Saved league_values_csv/Premier League_values_2016-03-15.csv
Saved league_values_csv/Premier League_values_2017-03-15.csv
Saved league_values_csv/Premier League_values_2018-03-15.csv
Saved league_values_csv/Premier League_values_2019-03-15.csv
Saved league_values_csv/Premier League_values_2020-03-15.csv
Saved league_values_csv/Premier League_values_2021-03-15.csv
Saved league_values_csv/Premier League_values_2022-03-15.csv
Saved league_values_csv/Premier League_values_2023-03-15.csv
Saved league_values_csv/Premier League_values_2024-03-15.csv
```

```
[130]: import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.ticker as ticker
import numpy as np

# Define colors and markers for each league
league_colors = {
    "Premier League": "blue",
    "La Liga": "green",
    "Bundesliga": "green",
    "Ligue 1": "green",
    "Serie A": "green"
}

league_markers = {
    "Premier League": "o",
    "La Liga": "o",
    "Bundesliga": "o",
    "Ligue 1": "o",
    "Serie A": "o"
}

plt.figure(figsize=(10, 6))

# Holders for all x/y to fit the line
all_x = []
all_y = []

# Flag to ensure we only label "Other European Leagues" once
other_labeled = False

# Loop through each league and plot the scatter for "Capacity" vs "Capacity.1"
for league, df in dfs_stadium_capacity.items():
    df_clean = df.dropna(subset=["Capacity", "Capacity.1"]).copy()

    x = df_clean["Capacity"].values
    y = df_clean["Capacity.1"].values
```

```

all_x.append(x)
all_y.append(y)

# Decide legend label
if league == "Premier League":
    label = "Premier League"
else:
    if not other_labeled:
        label = "Other European Leagues"
        other_labeled = True
    else:
        label = "_nolegend_" # skip further legend entries

plt.scatter(
    x, y,
    color=league_colors.get(league, "black"),
    marker=league_markers.get(league, "o"),
    s=80,
    alpha=0.7,
    label=label
)

# Combine all leagues into single arrays
all_x = np.concatenate(all_x)
all_y = np.concatenate(all_y)

# Compute line of best fit
m, b = np.polyfit(all_x, all_y, 1)
x_fit = np.linspace(all_x.min(), all_x.max(), 100)
y_fit = m * x_fit + b
plt.plot(x_fit, y_fit, color='red', linewidth=2,
         label=f'Line of Best Fit')

# Labels, title, legend, grid
plt.xlabel("Stadium Capacity")
plt.ylabel("Percentage of Capacity Full")
plt.title("Average % throughout the year of Stadium Attendance vs the stadium_
↳ capacity")
plt.legend(title="League")
plt.grid(True)

ax = plt.gca()
ax.yaxis.set_major_locator(ticker.MaxNLocator(6))
ymin, ymax = ax.get_ylim()
if ymin > ymax:
    ax.set_ylim(ymax, ymin)

```

```
plt.savefig("Stadium_attendance_vs_stadium_capacity.png", dpi=300,
            ↳bbox_inches="tight")
plt.show()
```

