📌 بخش ۱: وارد کردن کتابخانه‌ها

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

- pandas: برای خواندن، بررسی و پردازش داده‌های جدولی.(DataFrame)
- matplotlib.pyplot: برای رسم نمودارهای ساده مانند Bar ، Scatter، Pieو...
- seaborn: برای رسم نمودارهای آماری پیشرفته‌تر مانند Violin Plot و Heatmap.

📌 بخش ۲: خواندن و پیش‌پردازش داده‌ها

```
df = pd.read_csv('footballds.csv')
```

- فایل CSV با نام footballds.csvخوانده می‌شود و در قالب یک دیتافریم (df) ذخیره می‌شود.

```
df['goals'] = pd.to_numeric(df['goals'], errors='coerce')
df['assists'] = pd.to_numeric(df['assists'], errors='coerce')
df['rating'] = pd.to_numeric(df['rating'], errors='coerce')
df['age'] = pd.to_numeric(df['age'], errors='coerce')
df['position'] = df['position'].astype(str)
```

- ستون‌های عددی به نوع عددی (numeric) تبدیل می‌شوند. اگر مقداری غیرقابل تبدیل باشد، تبدیل به NaN می‌شود.
- ستون position هم به رشته تبدیل می‌شود.

```
df_filtered = df.dropna(subset=['goals', 'position'])
```

- تمام ردیف‌هایی که مقدار goals یا position آن‌ها خالی (NaN) است، حذف می‌شوند.

📌 بخش ۳: رسم نمودارها

1.نمودار میله‌ای - مجموع گل‌ها بر اساس پوزیشن

```
plt.figure(figsize=(10,6))
goals_by_position = df_filtered.groupby('position')['goals'].sum()
goals_by_position.plot(kind='bar', color='skyblue')
plt.title('Total Goals by Position', fontsize=14)
plt.xlabel('Position', fontsize=12)
plt.ylabel('Total Goals', fontsize=12)
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```

- این نمودار مجموع گل‌ها برای هر پوزیشن (مهاجم، هافبک و...) را نمایش می‌دهد.

2.نمودار پراکندگی - (Scatter Plot) گل‌ها نسبت به سن

```
plt.figure(figsize=(8, 6))
plt.scatter(df['age'], df['goals'], alpha=0.6, color='green')
plt.title('Scatter Plot: Goals vs Age')
plt.xlabel('Age')
plt.ylabel('Goals')
plt.grid(True)
plt.tight_layout()
plt.show()
```

- بررسی می‌شود که بازیکنان در چه سنی بیشتر گل می‌زنند. هر نقطه نمایانگر یک بازیکن است.

3.نمودار جدید - هیستوگرام تعداد بازیکنان بر اساس تعداد گل

```
plt.figure(figsize=(10,6))
plt.hist(df['goals'].dropna(), bins=15, color='orange', edgecolor='black')
plt.title('Histogram of Goals Scored')
```

```
plt.xlabel('Goals')
plt.ylabel('Number of Players')
plt.tight_layout()
plt.show()
```

- این نمودار نشان می‌دهد چه تعداد بازیکن چند گل زده‌اند (توزیع تعداد گل‌ها).

4.نمودار - BoxPlot ریتینگ بر اساس پوزیشن

```
plt.figure(figsize=(10,6))
sns.boxplot(data=df, x='position', y='rating', palette='pastel')
plt.title('Boxplot of Rating by Position')
plt.xlabel('Position')
plt.ylabel('Rating')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

- برای هر پوزیشن، محدوده پراکندگی امتیاز (Rating) بازیکنان نمایش داده می‌شود.

5.نمودار دایره‌ای - (Pie) درصد بازیکنان در هر پوزیشن

```
plt.figure(figsize=(8, 8))
position_counts = df['position'].value_counts()
plt.pie(position_counts, labels=position_counts.index, autopct='%1.1f%%', startangle=140, colors=sns.color_palette('Set2'))
plt.title('Player Distribution by Position')
plt.axis('equal')
plt.tight_layout()
plt.show()
```

- درصد بازیکنان در هر پوزیشن را نمایش می‌دهد.

6.نمودار - Violin توزیع ریتینگ بازیکنان بر اساس پوزیشن

```
plt.figure(figsize=(12, 6))
sns.violinplot(data=df, x='position', y='rating', palette='Set3')
plt.title('Distribution of Ratings by Position')
plt.xlabel('Position')
plt.ylabel('Rating')
plt.xticks(rotation=45)
plt.tight_layout()
```

```
plt.show()
```

🔹 ترکیبی از BoxPlot و KDE که شکل دقیق‌تری از توزیع امتیاز بازیکنان در پوزیشن‌های مختلف نشان می‌دهد.

7.نمودار - Heatmap همبستگی بین ویژگی‌ها

```
plt.figure(figsize=(8, 6))
correlation = df[['goals', 'assists', 'rating', 'age']].corr()
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f", square=True)
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```

🔹 این نمودار همبستگی (رابطه) بین متغیرهای عددی را نمایش می‌دهد.

📌 بخش ٤: بررسی و پاکسازی مقادیر گمشده(Nulls)

درصد مقادیر گمشده

```
total_null = df.isnull().sum().sum()
total_cells = df.size
null_percentage = (total_null / total_cells) * 100
```

```
print(f"all null: {null_percentage:.2f}%")
```

🔹 درصد کل سلول‌های گمشده را محاسبه می‌کند.

```
null_percentage_per_column = (df.isnull().sum() / len(df)) * 100
print("Null percentage per column:\n",
null_percentage_per_column.round(2))
```

🔹 درصد مقادیر گمشده برای هر ستون را نشان می‌دهد.

📌 بخش ۵: پر کردن مقادیر گمشده

```
df['position'].fillna('Unknown', inplace=True)
```

🔹 مقدارهای خالی در پوزیشن به "Unknown" تبدیل می‌شود.

```
df['position_encoded'] = df['position'].astype('category').cat.codes
```

🔹 پوزیشن‌ها به صورت عددی کدگذاری می‌شوند.

پر کردن مقدار assists بر اساس میانگین هر تعداد گل

```
goals_assist_avg = df.groupby('goals')['assists'].mean()
def fill_assists(row):
if pd.isnull(row['assists']):
return goals_assist_avg.get(row['goals'], df['assists'].mean())
else:
```

```
return row['assists']

df['assists'] = df.apply(fill_assists, axis=1)
```

- اگر مقدار assist خالی باشد، میانگین assist برای همان تعداد گل را جایگزین می‌کند.

پر کردن مقدار rating بر اساس میانگین ترکیبی گل وassist

```
grouped_avg = df.groupby(['goals', 'assists'])['rating'].mean()

def fill_rating(row):

if pd.isnull(row['rating']):

return grouped_avg.get((row['goals'], row['assists']), df['rating'].mean())

else:

return row['rating']

df['rating'] = df.apply(fill_rating, axis=1)
```

- اگر امتیاز بازیکن خالی باشد، از میانگین بازیکنانی با تعداد گل و پاس مشابه استفاده می‌شود.

پر کردن مقدار club با حالت رایج‌تر در همان پوزیشن

```
position_club_mode = df.groupby('position')['club'].agg(lambda x: x.mode().iloc[0] if not x.mode().empty else None)

def fill_club(row):
```

```
if pd.isnull(row['club']):
return position_club_mode.get(row['position'], df['club'].mode()[0])
else:
return row['club']
df['club'] = df.apply(fill_club, axis=1)
```

- اگر نام باشگاه بازیکن خالی باشد، رایج‌ترین باشگاه برای آن پوزیشن جایگزین می‌شود.

📌 بخش آخر: ذخیره فایل تمیزشده

```
df_cleaned = df.dropna()
df_cleaned.to_excel("cleaned_dataset.xlsx", index=False)
print("cleaned_dataset.xlsx successfully saved")
```

- تمام ردیف‌های باقی‌مانده با مقدارهای کامل در فایل Excel ذخیره می‌شود.