

```

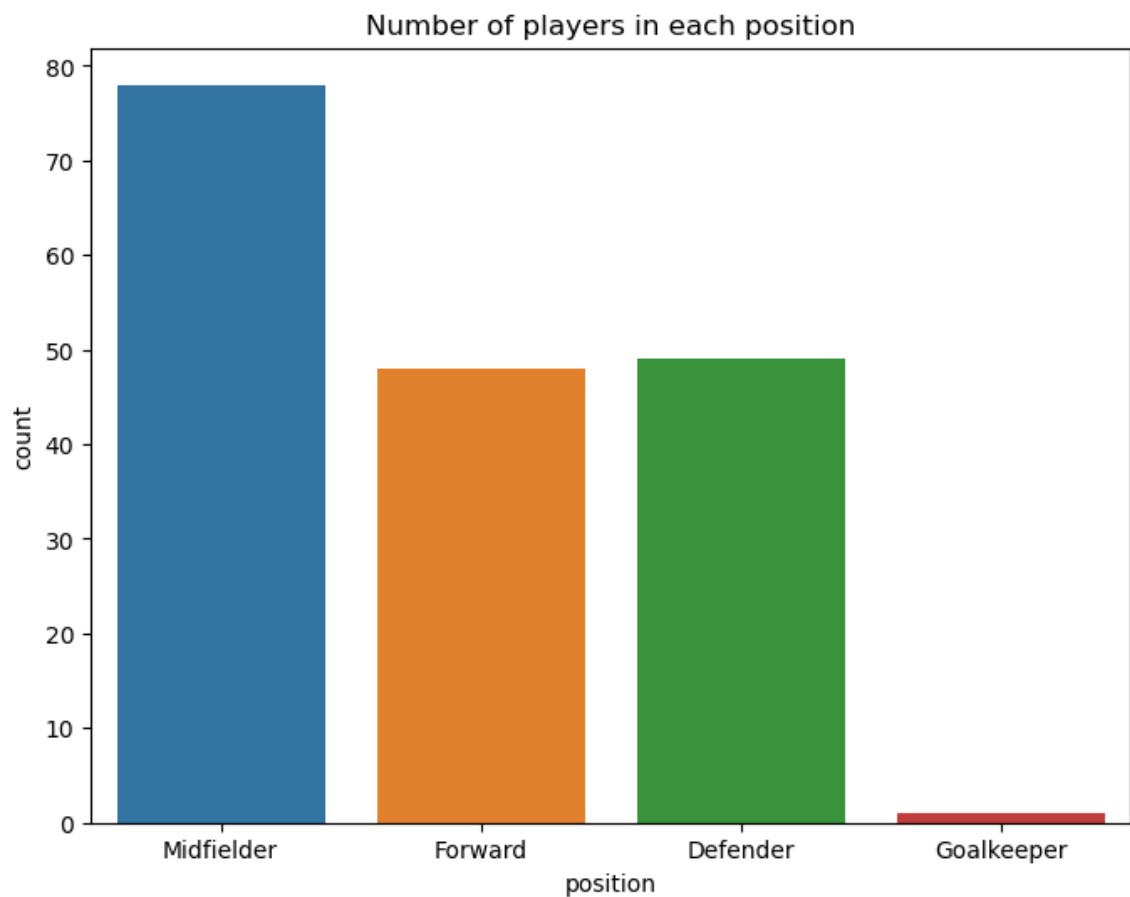
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

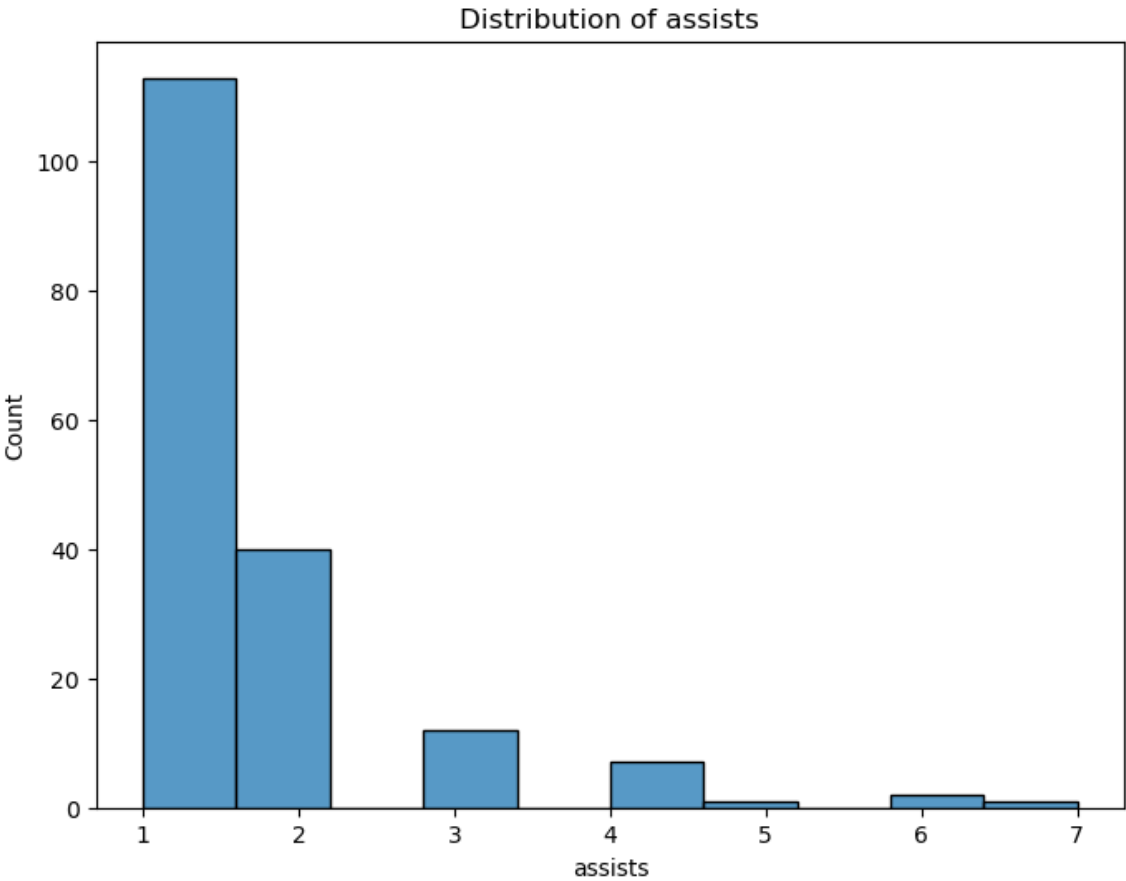
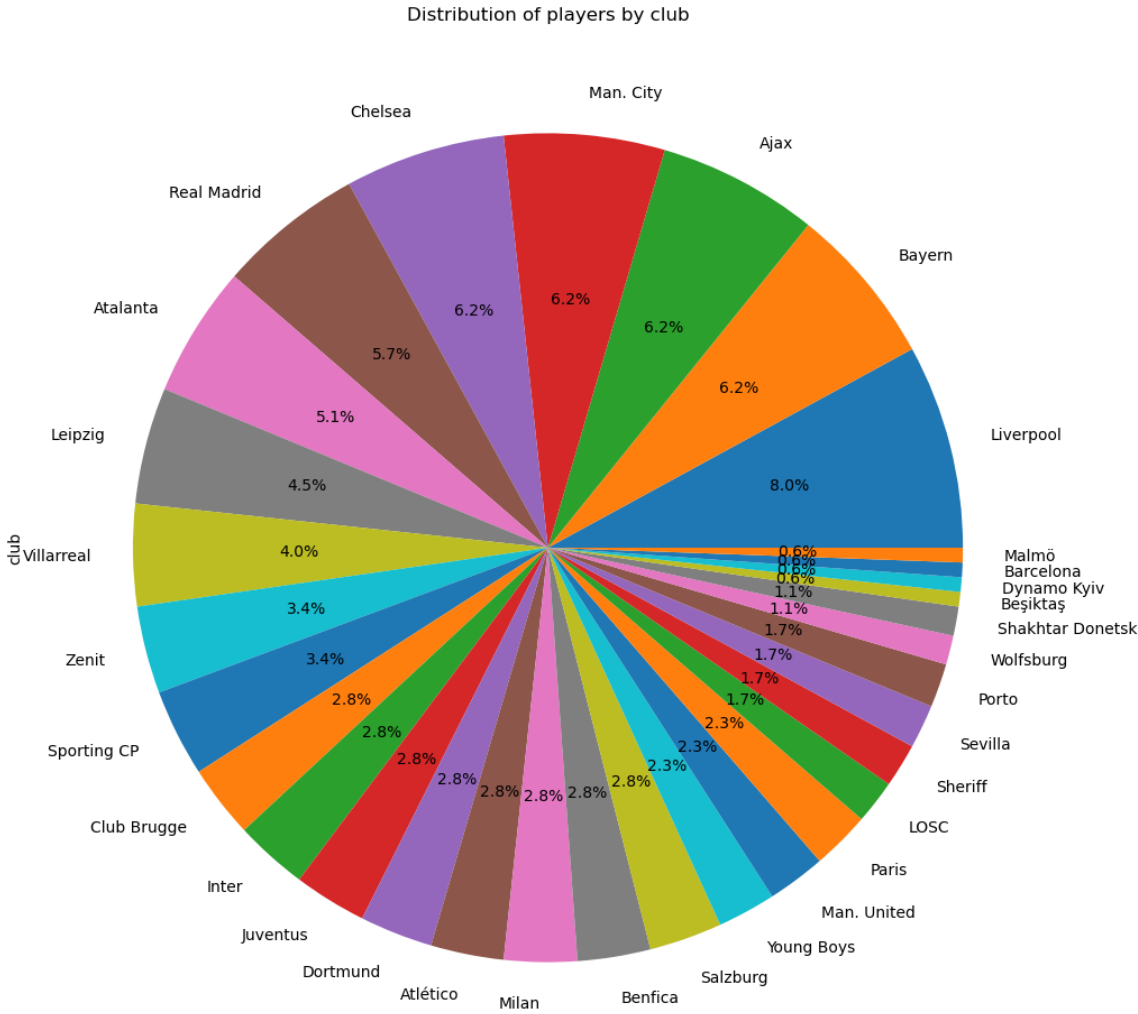
# read the dataset
df = pd.read_csv(r"C:\Users\hp\Desktop\python\attacking.csv")

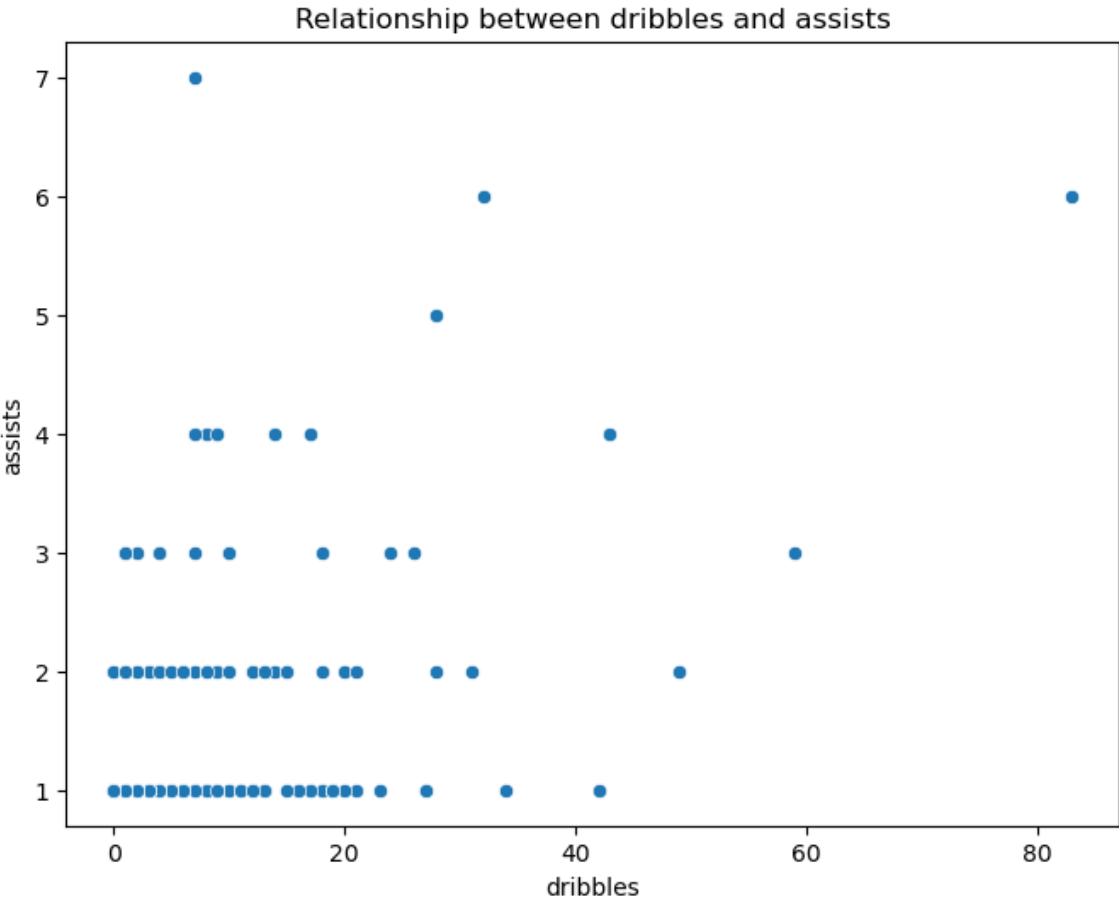
# display the first 5 rows of the dataset
df.head()
# check for missing values
df.isnull().sum()
# bar plot for number of players in each position
plt.figure(figsize=(8,6))
sns.countplot(data=df, x='position')
plt.title('Number of players in each position')
plt.show()
# pie chart for distribution of players by club
plt.figure(figsize=(12,12))
df['club'].value_counts().plot(kind='pie', autopct='%1.1f%%')
plt.title('Distribution of players by club')
plt.show()
# histogram for distribution of assists
plt.figure(figsize=(8,6))
sns.histplot(data=df, x='assists', bins=10)
plt.title('Distribution of assists')
plt.show()
# scatterplot for relationship between dribbles and assists
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='dribbles', y='assists')
plt.title('Relationship between dribbles and assists')
plt.show()
# pairplot for relationships between all numerical variables
sns.pairplot(df[['assists', 'corner_taken', 'offsides', 'dribbles', 'match_played']])
plt.show()
# line graph for trend in number of assists over match_played
plt.figure(figsize=(8,6))
sns.lineplot(data=df, x='match_played', y='assists')
plt.title('Trend in number of assists over match_played')
plt.show()
# facetgrid with lineplot for trend in numerical variables over match_played
grid = sns.FacetGrid(df, col='position', hue='position', col_wrap=3, height=4)
grid.map(sns.lineplot, 'match_played', 'assists')
grid.map(sns.lineplot, 'match_played', 'corner_taken')
grid.map(sns.lineplot, 'match_played', 'offsides')
grid.map(sns.lineplot, 'match_played', 'dribbles')
grid.add_legend()
plt.show()
# box and whisker plot for distribution of dribbles by position
plt.figure(figsize=(8,6))
sns.boxplot(data=df, x='position', y='dribbles')
plt.title('Distribution of dribbles by position')
plt.show()
# violin plot for distribution of corner_taken by position
plt.figure(figsize=(8,6))
sns.violinplot(data=df, x='position', y='corner_taken')
plt.title('Distribution of corner_taken by position')
plt.show()
# heatmap to visualize correlations between numerical variables

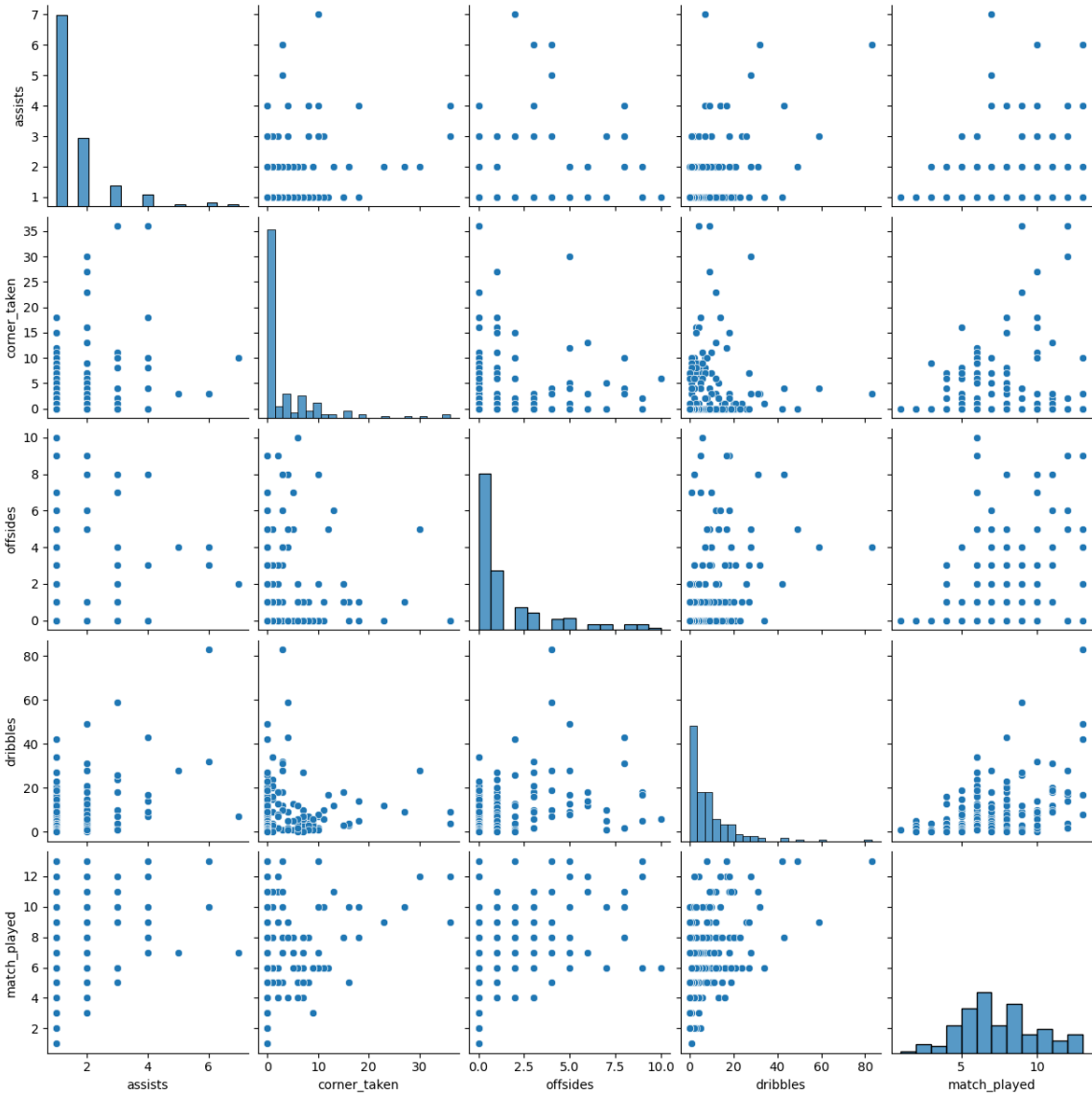
```

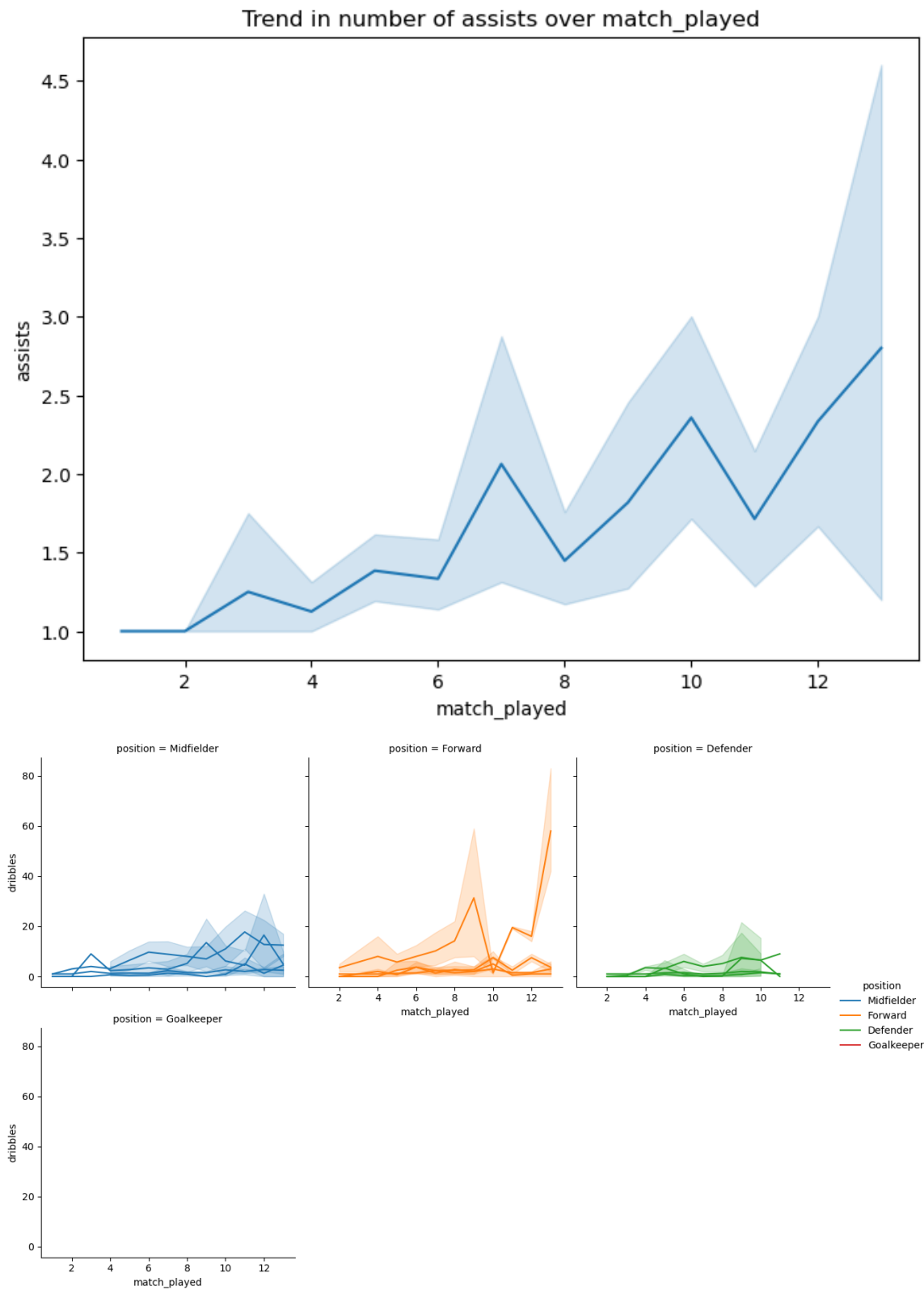
```
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(), cmap='coolwarm', annot=True)
plt.title('Correlation between numerical variables')
plt.show()
# scatterplot to visualize the relationship between assists and match_played
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='match_played', y='assists')
plt.title('Relationship between assists and match_played')
plt.show()
```

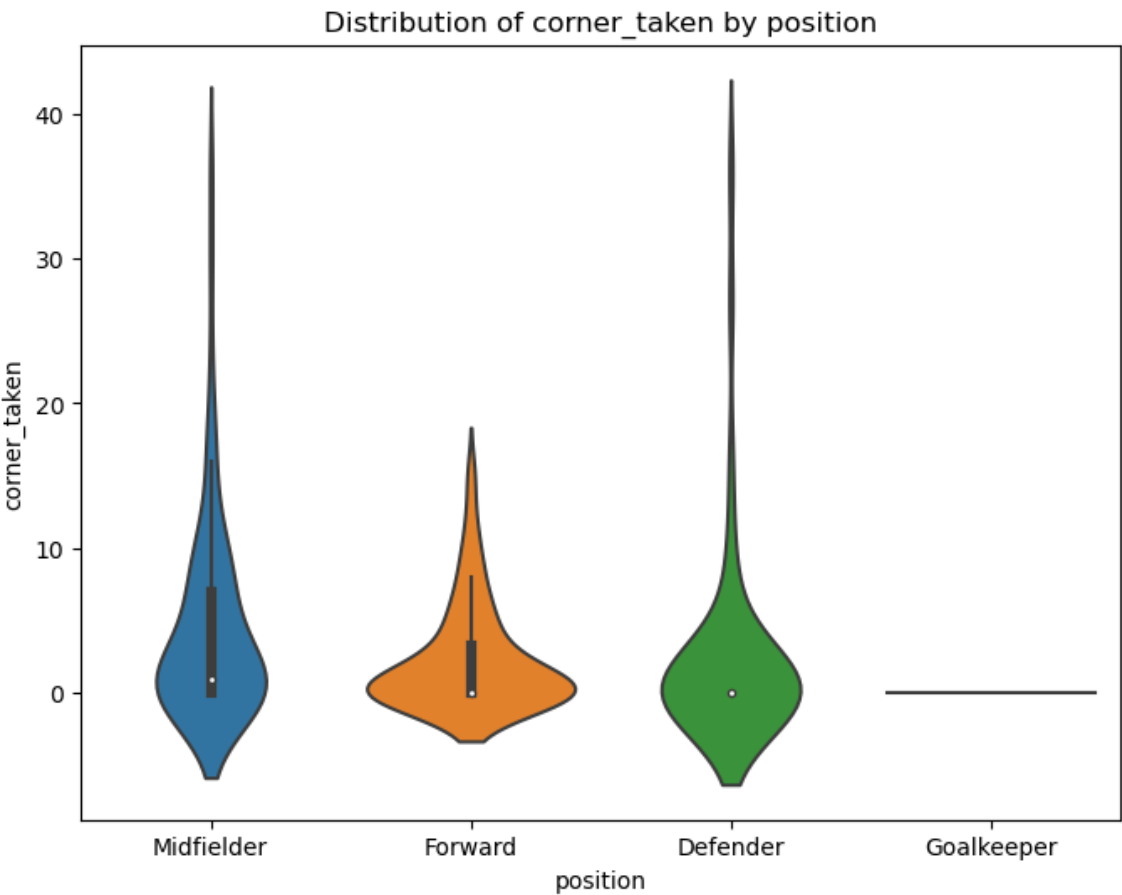
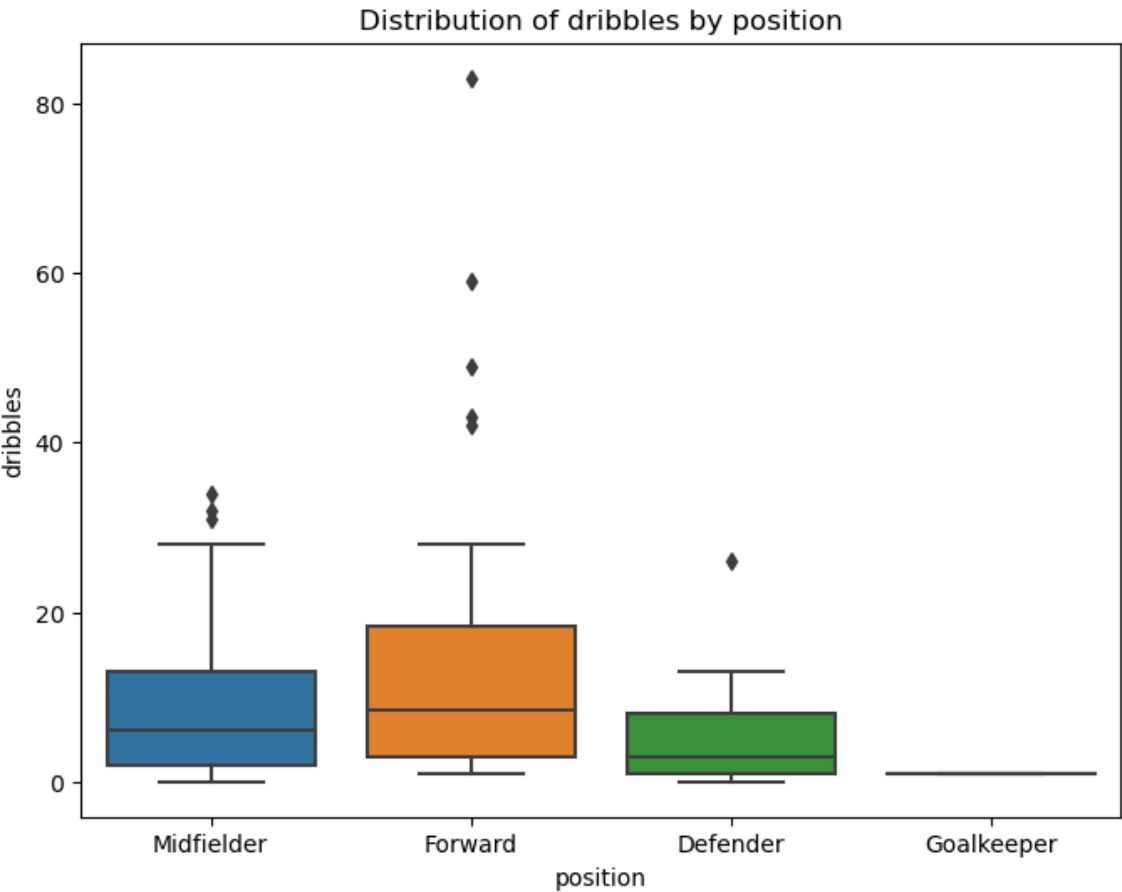




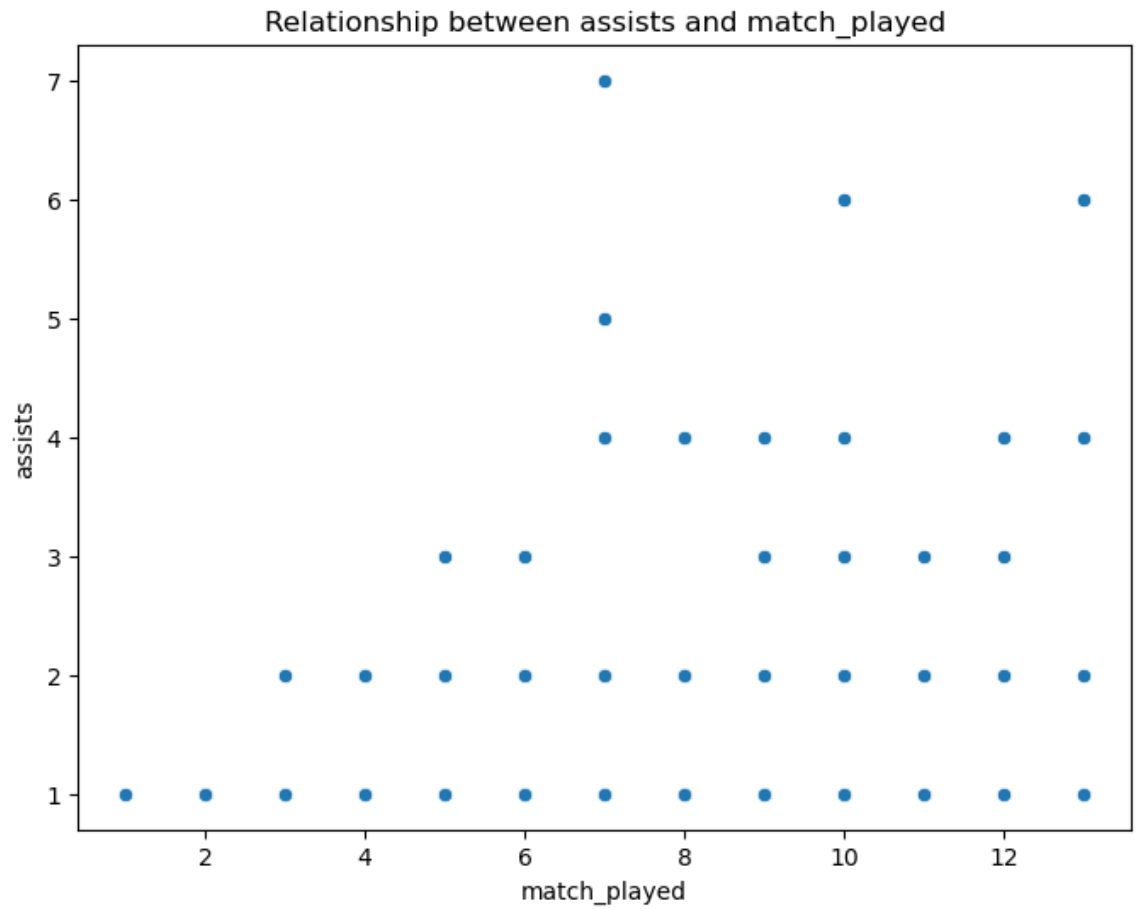
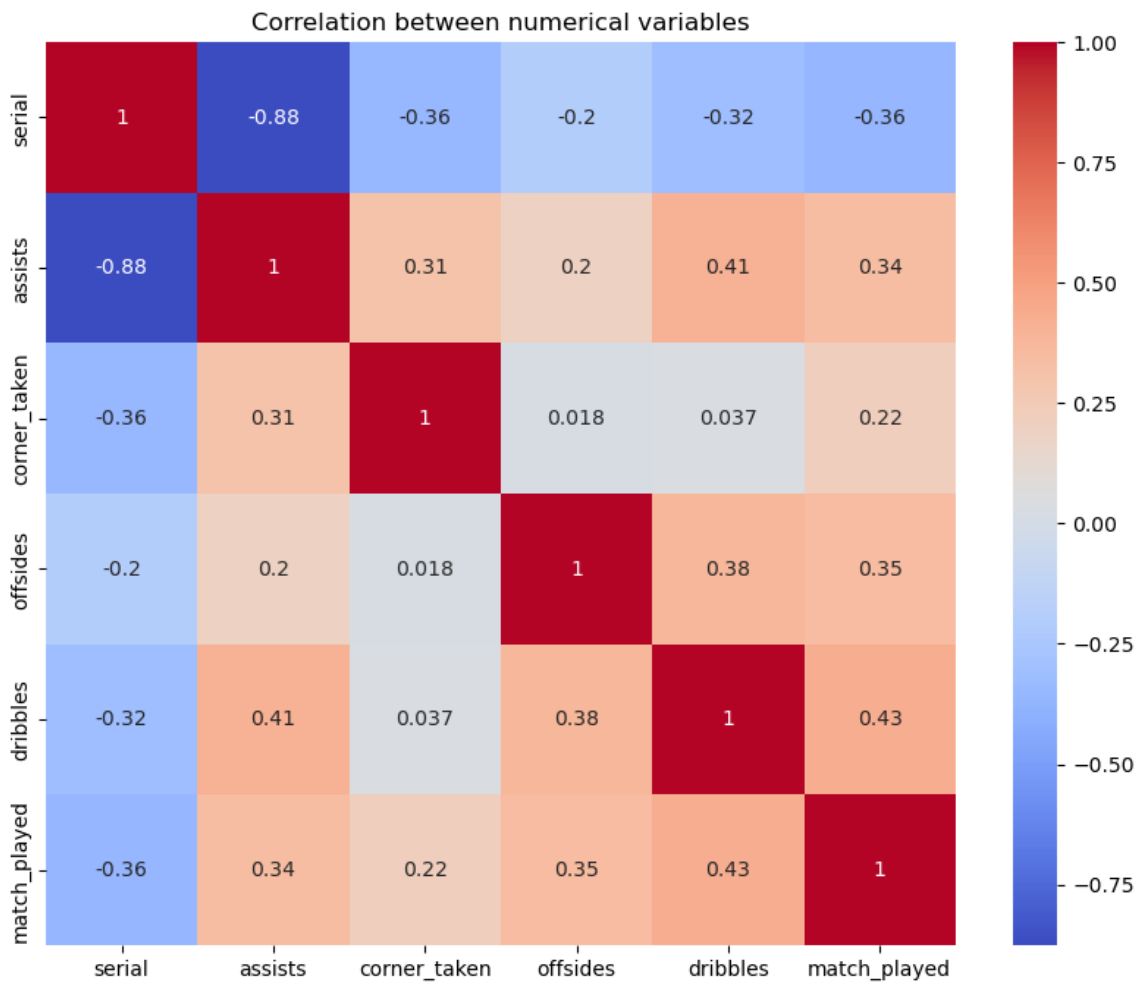








```
C:\Users\hp\AppData\Local\Temp\ipykernel_16584\969170931.py:60: FutureWarning:  
The default value of numeric_only in DataFrame.corr is deprecated. In a future  
version, it will default to False. Select only valid columns or specify the val  
ue of numeric_only to silence this warning.  
sns.heatmap(df.corr(), cmap='coolwarm', annot=True)
```



```
In [ ]:
```