In [1]:

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

In [3]:

```
# Loading the dataset
data = pd.read_csv('player_data.csv')
```

In [4]:

```
data.head()
```

Out[4]:

	Name	Age	Team	Games	Rebounds	Assists	Steals	Blocks	Turnovers	Points
0	James Harden	25	HOU	81	459	565	154	60	321	2217
1	Chris Paul	29	LAC	82	376	838	156	15	190	1564
2	Stephen Curry	26	GSW	80	341	619	163	16	249	1900
3	Anthony Davis	21	NOP	68	696	149	100	200	95	1656
4	DeAndre Jordan	26	LAC	82	1226	61	81	183	109	946

Exploratory Data Analysis

In [6]:

```
# checking for duplicates

duplicates = data[data.duplicated()]

if duplicates.empty:
    print('No duplicates found.')
else:
    print('Duplicates found:')
    print(duplicates)
```

```
No duplicates found.
```

In [8]:

```
# check for null values
print(data.isnull().sum())
```

```
Name         0
Age          0
Team         0
Games        0
Rebounds     0
Assists      0
Steals       0
Blocks       0
Turnovers    0
Points       0
dtype: int64
```

In [9]:

```
data = data.dropna()
```

In [16]:

```
# Statistical Summary
data.describe()
```

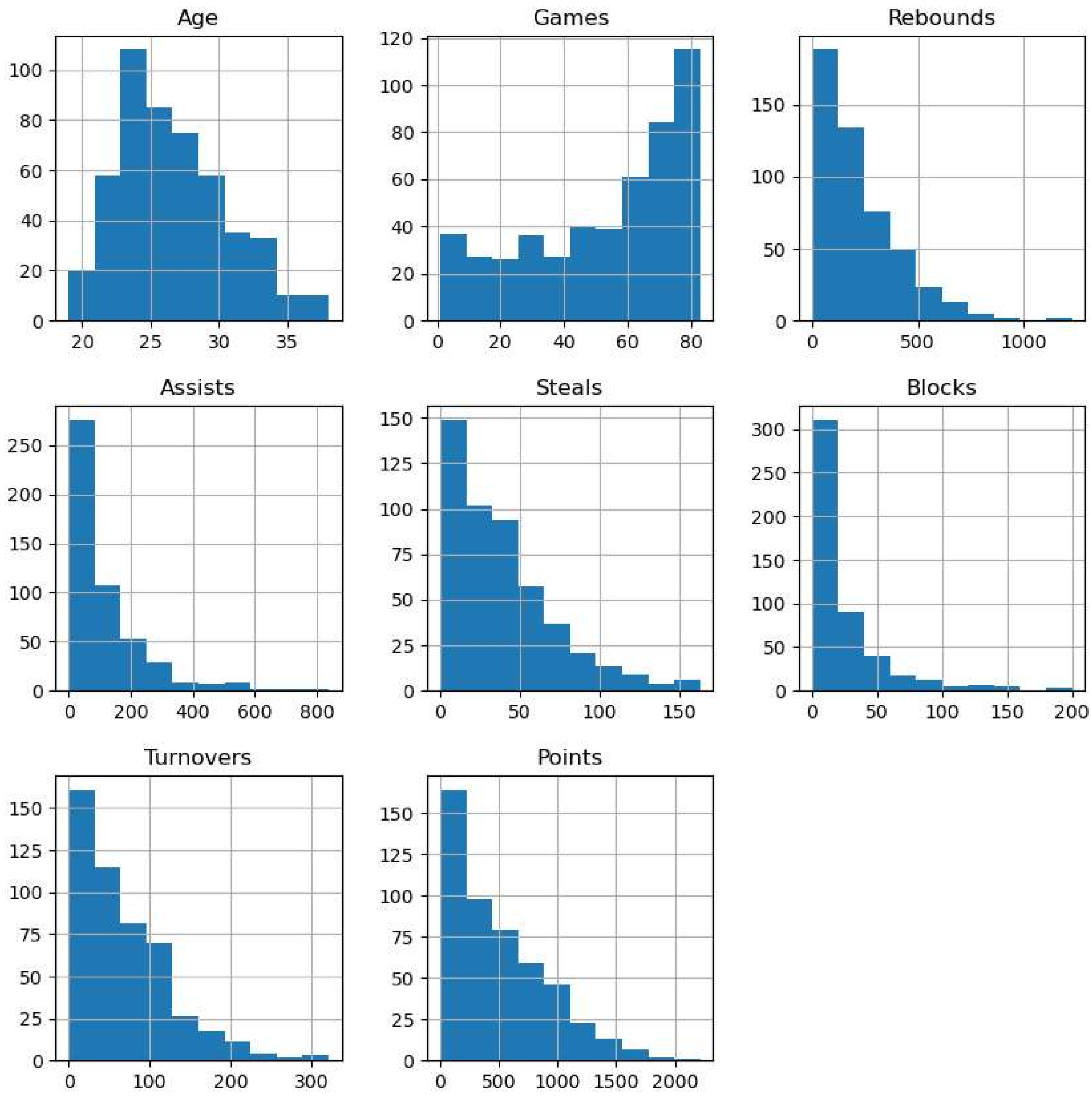
Out[16]:

	Age	Games	Rebounds	Assists	Steals	Blocks	Turnovers	
count	492.000000	492.000000	492.000000	492.000000	492.000000	492.000000	492.000000	4
mean	26.536585	52.806911	216.471545	110.166667	38.680894	23.977642	68.546748	5
std	4.193473	24.344049	192.089723	125.922489	33.413136	31.635486	58.525882	4
min	19.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	23.000000	33.000000	68.750000	23.750000	12.000000	4.750000	22.000000	1
50%	26.000000	61.000000	176.000000	69.500000	32.000000	14.000000	55.500000	4
75%	29.000000	74.000000	314.250000	150.250000	54.500000	29.250000	100.000000	7
max	38.000000	83.000000	1226.000000	838.000000	163.000000	200.000000	321.000000	22

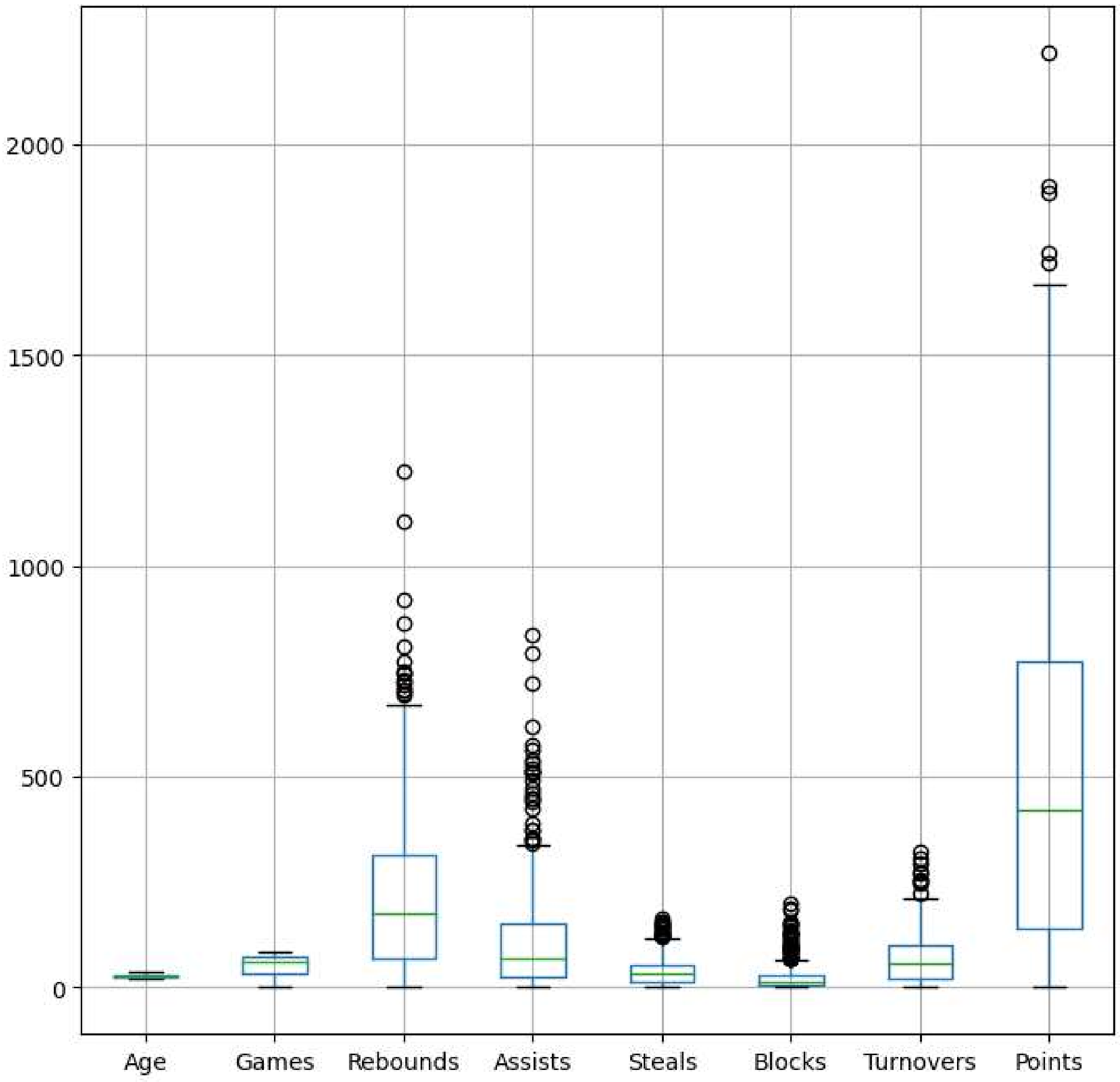
Distribution of Data

In [10]:

```
data.hist(figsize=(10, 10))
plt.show()
```



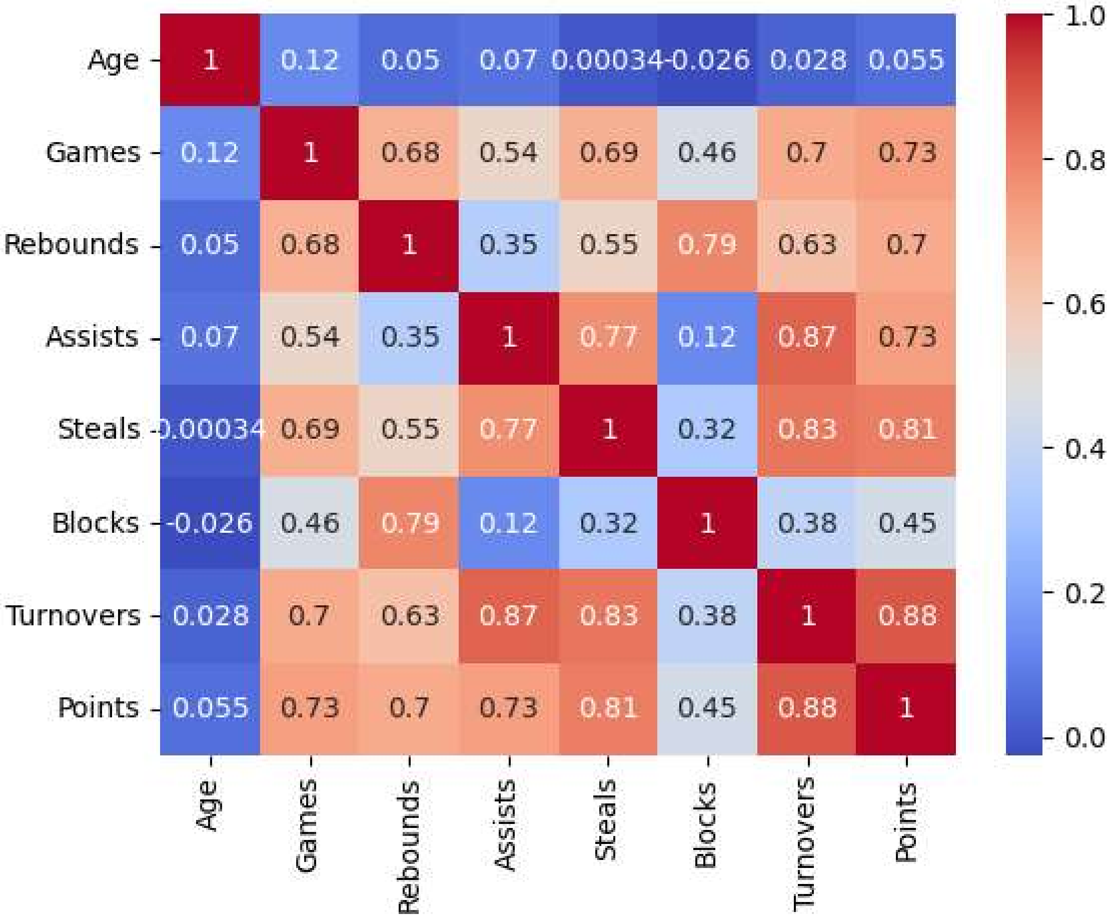
In [15]:
```
# Plotting a boxplot for each column in the dataset
data.boxplot(figsize=(8,8))
plt.show()
```



Correlation Analysis

In [11]:

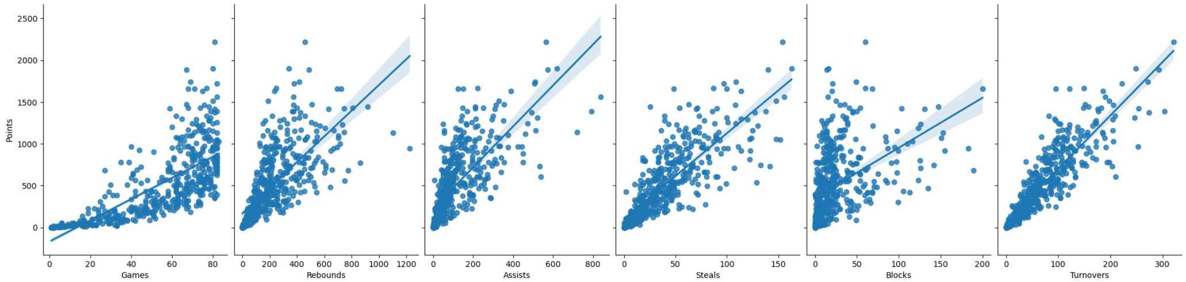
```
corr_matrix = data.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()
```



Performance Analysis

In [13]:

```
sns.pairplot(data, x_vars = ['Games', 'Rebounds', 'Assists', 'Steals', 'Blocks
             y_vars=['Points'], height = 5, aspect = 0.7, kind = 'reg')
plt.show()
```



In [ ]:

```
```