In [2]:
```
import pandas as pd
df=pd.read_csv("batting_card - Copy2.csv")
df
```

Out[2]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	count
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
3	2023.0	1359475	GT v CSK	NaN	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
...	...	...	...	...	...	...	...	
1349	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1350	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1351	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1352	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1353	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc

1354 rows × 25 columns

In [3]:
```
# to check the number of rows and columns
df.shape
```

Out[3]:
```
(1354, 25)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1354 entries, 0 to 1353
Data columns (total 25 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Unnamed: 0       1310 non-null   float64
 1   match_id         1354 non-null   int64  
 2   match_name       1342 non-null   object 
 3   home_team        1333 non-null   object 
 4   away_team        1341 non-null   object 
 5   venue            1354 non-null   object 
 6   city             1354 non-null   object 
 7   country          1354 non-null   object 
 8   current_innings  1354 non-null   object 
 9   innings_id       1354 non-null   int64  
 10  name             1354 non-null   object 
 11  fullName         1354 non-null   object 
 12  runs             1331 non-null   float64
 13  ballsFaced       1332 non-null   float64
 14  minutes          1273 non-null   object 
 15  fours            1328 non-null   float64
 16  sixes            1328 non-null   float64
 17  strikeRate       1354 non-null   object 
 18  captain          1354 non-null   bool   
 19  isNotOut         1354 non-null   bool   
 20  runningScore     1354 non-null   object 
 21  runningOver      1046 non-null   float64
 22  shortText        1354 non-null   object 
 23  commentary       1046 non-null   object 
 24  link             0 non-null      float64
dtypes: bool(2), float64(7), int64(2), object(14)
memory usage: 246.1+ KB
```

In [5]:
```
#checking for null values
df.isnull()
```

Out[5]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	country	current
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	True	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
1349	False	False	False	False	False	False	False	False	
1350	False	False	False	False	False	False	False	False	
1351	False	False	False	False	False	False	False	False	
1352	False	False	False	False	False	False	False	False	
1353	False	False	False	False	False	False	False	False	

1354 rows × 25 columns

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Unnamed: 0          44
match_id             0
match_name          12
home_team           21
away_team           13
venue                0
city                 0
country              0
current_innings      0
innings_id           0
name                 0
fullName             0
runs                23
ballsFaced          22
minutes             81
fours               26
sixes               26
strikeRate           0
captain              0
isNotOut             0
runningScore         0
runningOver        308
shortText            0
commentary         308
link              1354
dtype: int64
```

In [7]:
```
(df.isnull().sum()/len(df))*100
```

Out[7]:
```
Unnamed: 0           3.249631
match_id             0.000000
match_name           0.886263
home_team            1.550960
away_team            0.960118
venue                0.000000
city                 0.000000
country              0.000000
current_innings      0.000000
innings_id           0.000000
name                 0.000000
fullName             0.000000
runs                 1.698671
ballsFaced           1.624815
minutes              5.982275
fours                1.920236
sixes                1.920236
strikeRate           0.000000
captain              0.000000
isNotOut             0.000000
runningScore         0.000000
runningOver         22.747415
shortText            0.000000
commentary          22.747415
link               100.000000
dtype: float64
```

In [8]:
```
df.median(numeric_only=True)
```

Out[8]:
```
Unnamed: 0        2023.0
match_id       1359515.0
innings_id           2.0
runs                12.0
ballsFaced          10.0
fours                1.0
sixes                0.0
captain              0.0
isNotOut             0.0
runningOver         12.3
link                 NaN
dtype: float64
```

In [9]:
```
df1=df.fillna(df.median())
df1
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_496\3484211276.py:1: FutureWarning:
The default value of numeric_only in DataFrame.median is deprecated. In a fut
ure version, it will default to False. In addition, specifying 'numeric_only=
None' is deprecated. Select only valid columns or specify the value of numeri
c_only to silence this warning.
  df1=df.fillna(df.median())
```

Out[9]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	count
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
3	2023.0	1359475	GT v CSK	NaN	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
...	...	...	...	...	...	...	...	
1349	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1350	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1351	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1352	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1353	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc

1354 rows × 25 columns

In [10]:
```
(df1.isnull().sum()/len(df1))*100
```

Out[10]:
```
Unnamed: 0           0.000000
match_id             0.000000
match_name           0.886263
home_team            1.550960
away_team            0.960118
venue                0.000000
city                 0.000000
country              0.000000
current_innings      0.000000
innings_id           0.000000
name                 0.000000
fullName             0.000000
runs                 0.000000
ballsFaced           0.000000
minutes              5.982275
fours                0.000000
sixes                0.000000
strikeRate           0.000000
captain              0.000000
isNotOut             0.000000
runningScore         0.000000
runningOver          0.000000
shortText            0.000000
commentary          22.747415
link               100.000000
dtype: float64
```

In [11]:
```
df1["minutes"].mode()
df1["away_team"].mode()
df1["home_team"].mode()
df1["match_name"].mode()
df1
```

Out[11]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	count
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
3	2023.0	1359475	GT v CSK	NaN	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
...	...	...	...	...	...	...	...	
1349	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1350	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1351	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1352	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc
1353	2023.0	1359527	KKR v PBKS	KKR	PBKS	Eden Gardens, Kolkata	Kolkata	Inc

1354 rows × 25 columns

In [12]:
```
df1["minutes"]=df1["minutes"].fillna(df1["minutes"].mode()[0])
df1["away_team"]=df1["away_team"].fillna(df1["away_team"].mode()[0])
df1["home_team"]=df1["home_team"].fillna(df1["home_team"].mode()[0])
df1["match_name"]=df1["match_name"].fillna(df1["match_name"].mode()[0])
```

In [13]:
```
(df1.isnull().sum()/len(df1))*100
```

Out[13]:
```
Unnamed: 0           0.000000
match_id             0.000000
match_name           0.000000
home_team            0.000000
away_team            0.000000
venue                0.000000
city                 0.000000
country              0.000000
current_innings      0.000000
innings_id           0.000000
name                 0.000000
fullName             0.000000
runs                 0.000000
ballsFaced           0.000000
minutes              0.000000
fours                0.000000
sixes                0.000000
strikeRate           0.000000
captain              0.000000
isNotOut             0.000000
runningScore         0.000000
runningOver          0.000000
shortText            0.000000
commentary          22.747415
link               100.000000
dtype: float64
```

In [14]:
```
df1.duplicated()
```

Out[14]:
```
0       False
1       False
2       False
3       False
4       False
        ...
1349     True
1350     True
1351     True
1352     True
1353     True
Length: 1354, dtype: bool
```

In [15]:
```
df1.duplicated().sum()
```

Out[15]:
```
173
```

In [16]:
```
(df1.duplicated().sum()/len(df1))*100
```

Out[16]:
```
12.776957163958642
```

In [17]:
```
#to delete duplicates
df=df1.drop_duplicates()
```

In [18]:
```
df.duplicated().sum()
```

Out[18]:
```
0
```

In [19]:
```
(df.duplicated().sum()/len(df))*100
```

Out[19]:
```
0.0
```

In [20]:
```
df.drop(['commentary','link'],axis=1,inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_496\1727197674.py:1: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df.drop(['commentary','link'],axis=1,inplace=True)
```

In [21]:
```
(df.isnull().sum()/len(df))*100
```

Out[21]:
```
Unnamed: 0         0.0
match_id           0.0
match_name         0.0
home_team          0.0
away_team          0.0
venue              0.0
city               0.0
country            0.0
current_innings    0.0
innings_id         0.0
name               0.0
fullName           0.0
runs               0.0
ballsFaced         0.0
minutes            0.0
fours              0.0
sixes              0.0
strikeRate         0.0
captain            0.0
isNotOut           0.0
runningScore       0.0
runningOver        0.0
shortText          0.0
dtype: float64
```

In [ ]:
```
df.to_csv("batting_card_ready.csv", index=False)
```

In [30]:
```
import pandas as pd
test1=pd.read_csv("batting_card_ready.csv")
test1
```

Out[30]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	count
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
3	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
...	...	...	...	...	...	...	...	
1176	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1177	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1178	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1179	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1180	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc

1181 rows × 23 columns

In [31]:
```
test1.shape
```

Out[31]:
```
(1181, 23)
```

In [32]:
```
test1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1181 entries, 0 to 1180
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Unnamed: 0       1181 non-null   float64
 1   match_id         1181 non-null   int64  
 2   match_name       1181 non-null   object 
 3   home_team        1181 non-null   object 
 4   away_team        1181 non-null   object 
 5   venue            1181 non-null   object 
 6   city             1181 non-null   object 
 7   country          1181 non-null   object 
 8   current_innings  1181 non-null   object 
 9   innings_id       1181 non-null   int64  
 10  name             1181 non-null   object 
 11  fullName         1181 non-null   object 
 12  runs             1181 non-null   float64
 13  ballsFaced       1181 non-null   float64
 14  minutes          1181 non-null   object 
 15  fours            1181 non-null   float64
 16  sixes            1181 non-null   float64
 17  strikeRate       1181 non-null   object 
 18  captain          1181 non-null   bool   
 19  isNotOut         1181 non-null   bool   
 20  runningScore     1181 non-null   object 
 21  runningOver      1181 non-null   float64
 22  shortText        1181 non-null   object 
dtypes: bool(2), float64(6), int64(2), object(13)
memory usage: 196.2+ KB
```

In [33]:
```
#mean
test1["match_id"].mean()
```

Out[33]:
```
1360123.733276884
```

In [34]:
```
test1["innings_id"].mean()
```

Out[34]:
```
1.5088907705334462
```

In [35]:
```
test1["runs"].mean()
```

Out[35]:
```
20.18035563082134
```

In [36]:
```
test1["ballsFaced"].mean()
```

Out[36]:
```
14.504657070279425
```

In [37]:
```
test1["fours"].mean()
```

Out[37]:
```
1.8137171888230312
```

In [38]:
```
test1["sixes"].mean()
```

Out[38]:
```
0.9297205757832345
```

In [39]:
```
#median
test1["match_id"].median()
```

Out[39]:
```
1359510.0
```

In [40]:
```
test1["innings_id"].median()
```

Out[40]:
```
2.0
```

In [41]:
```
test1["runs"].median()
```

Out[41]:
```
12.0
```

In [42]:
```
test1["ballsFaced"].median()
```

Out[42]:
```
10.0
```

In [43]:
```
test1["fours"].median()
```

Out[43]:
```
1.0
```

In [44]:
```
test1["sixes"].median()
```

Out[44]:
```
0.0
```

In [45]:
```
#mode
test1["match_name"].mode()
```

Out[45]:
```
0    GT v MI
Name: match_name, dtype: object
```

In [46]:
```
test1["home_team"].mode()
```

Out[46]:
```
0    GT
Name: home_team, dtype: object
```

In [47]:
```
test1["away_team"].mode()
```

Out[47]:
```
0    MI
Name: away_team, dtype: object
```

In [48]:
```
test1["venue"].mode()
```

Out[48]:
```
0    MA Chidambaram Stadium, Chepauk, Chennai
Name: venue, dtype: object
```

In [49]:
```
test1["city"].mode()
```

Out[49]:
```
0    Chennai
Name: city, dtype: object
```

In [50]:
```
test1["current_innings"].mode()
```

Out[50]:
```
0    DC
Name: current_innings, dtype: object
```

In [51]:
```
test1["minutes"].mode()
```

Out[51]:
```
0    6
Name: minutes, dtype: object
```

In [52]:
```
test1["fullName"].mode()
```

Out[52]:
```
0       Shubman Gill
1    Wriddhiman Saha
Name: fullName, dtype: object
```

In [53]:
```
#standard diviation
test1["match_id"].std()
```

Out[53]:
```
2509.3701102039395
```

In [54]:
```
test1["innings_id"].std()
```

Out[54]:
```
0.5001327339998186
```

In [55]:
```
test1["runs"].std()
```

Out[55]:
```
22.294496247376074
```

In [56]:
```
test1["ballsFaced"].std()
```

Out[56]:
```
13.37716319861832
```

In [57]:
```
test1["fours"].std()
```

Out[57]:
```
2.434744580196755
```

In [58]:
```
test1["sixes"].std()
```

Out[58]:
```
1.5566061539342817
```

In [59]:
```
#varience
test1["match_id"].var()
```

Out[59]:
```
6296938.349984931
```

In [60]:
```
test1["innings_id"].var()
```

Out[60]: 0.2501327516181333

In [61]:
```
test1["runs"].var()
```

Out[61]: 497.0445629242658

In [62]:
```
test1["ballsFaced"].var()
```

Out[62]: 178.9484952424683

In [63]:
```
test1["fours"].var()
```

Out[63]: 5.927981170797472

In [64]:
```
test1["sixes"].var()
```

Out[64]: 2.423022718466077

In [65]:
```
#range
test1["fours"].max()
test1["fours"].min()
test1["fours"].max()-test1["fours"].min()
```

Out[65]: 16.0

In [66]:
```
test1["sixes"].max()
test1["sixes"].min()
test1["sixes"].max()-test1["sixes"].min()
```

Out[66]: 10.0

In [67]:
```
test1["ballsFaced"].max()
test1["ballsFaced"].min()
test1["ballsFaced"].max()-test1["ballsFaced"].min()
```

Out[67]: 66.0

In [68]:
```
test1["runs"].max()
test1["runs"].min()
test1["runs"].max()-test1["runs"].min()
```

Out[68]: 129.0

In [69]:
```
test1["innings_id"].max()
test1["innings_id"].min()
test1["innings_id"].max()-test1["innings_id"].min()
```

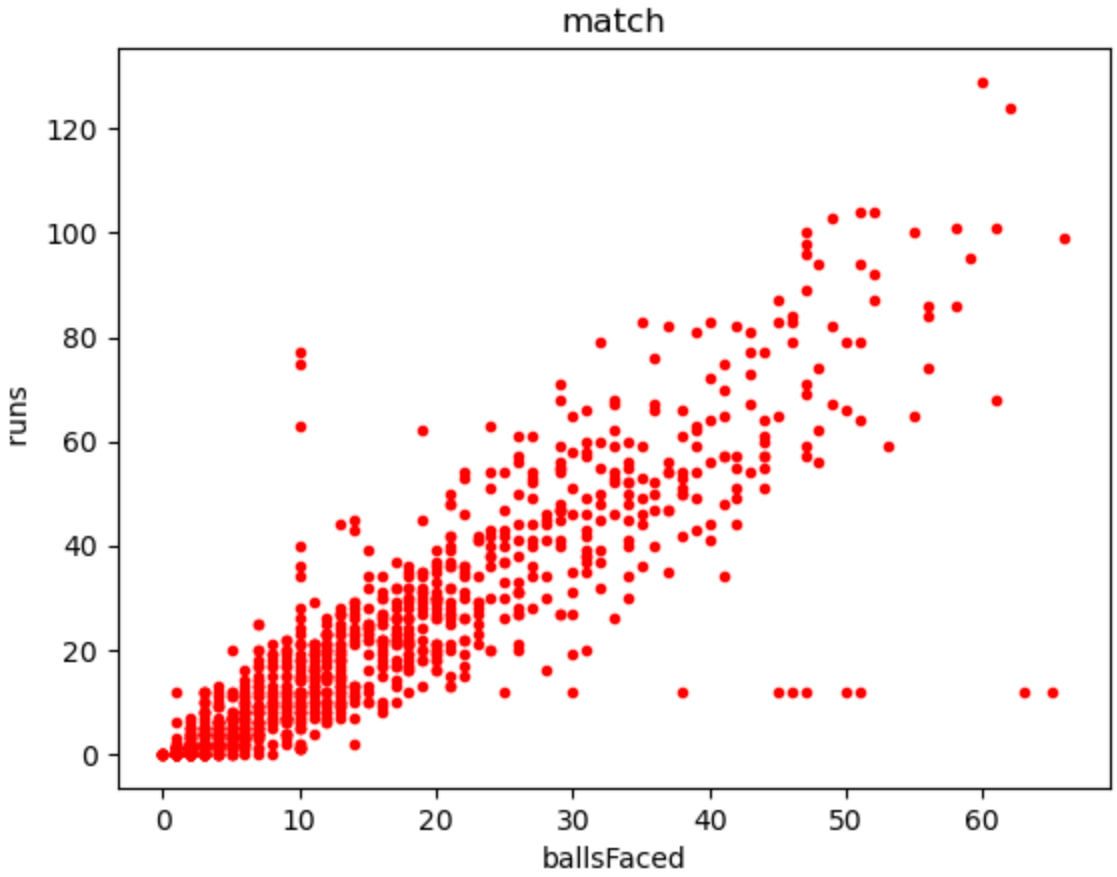
Out[69]: 1

In [70]:
```
test1["match_id"].max()
test1["match_id"].min()
test1["match_id"].max()-test1["match_id"].min()
```

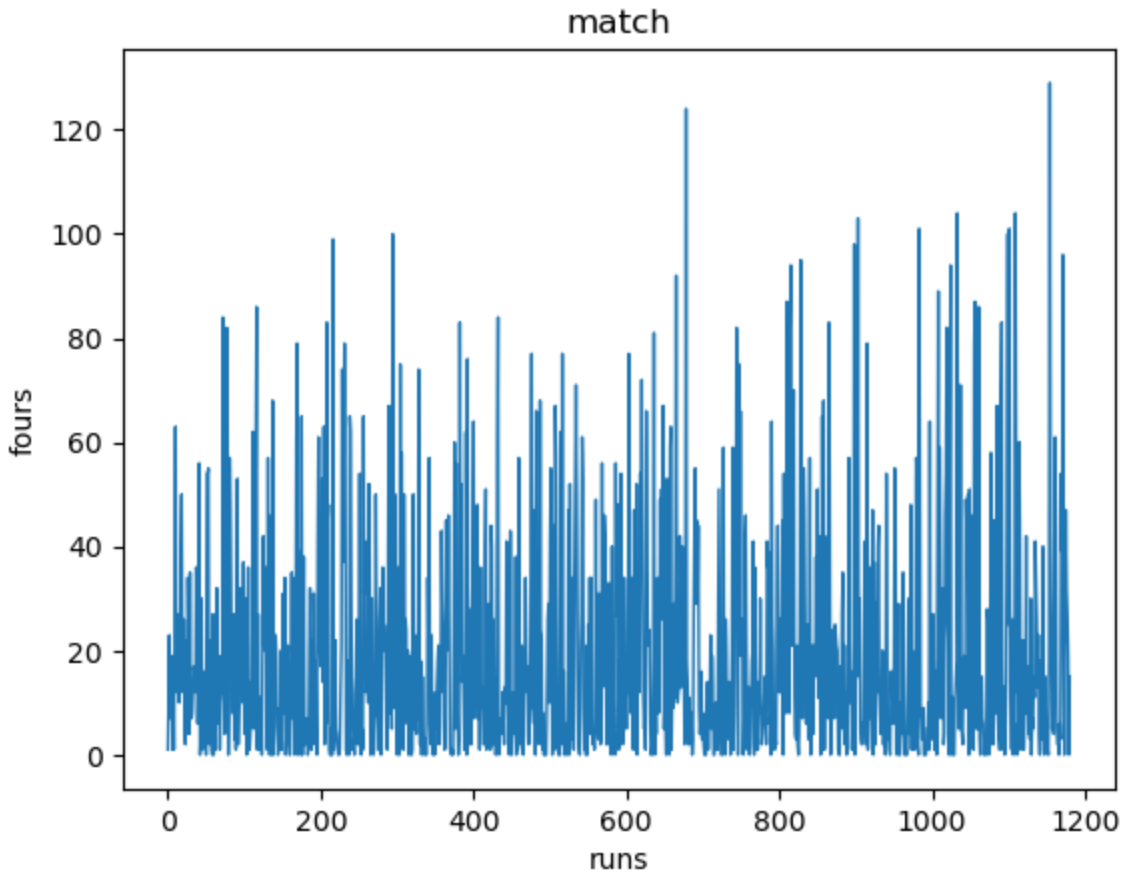
Out[70]: 10878

In [71]:
```
#plots
```

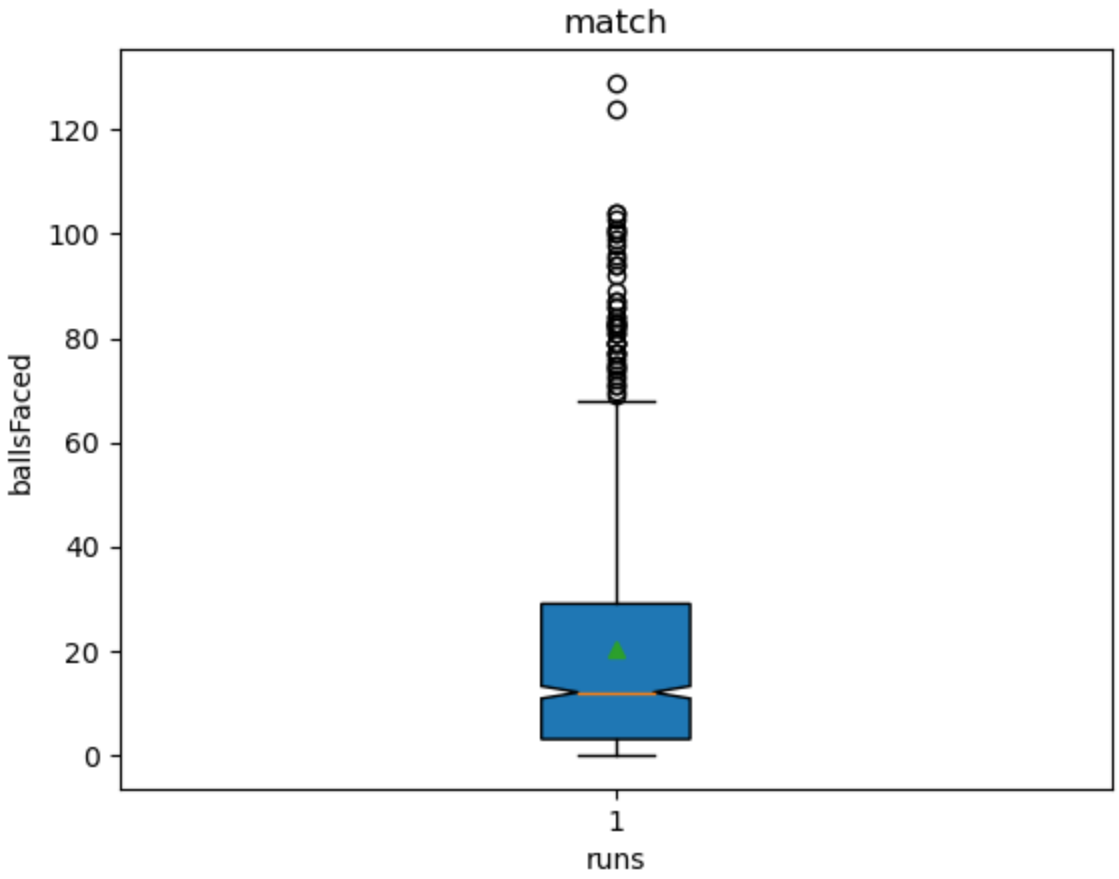
In [72]:
```
#scatter plot
import matplotlib.pyplot as plt
plt.scatter(test1['ballsFaced'],test1['runs'], color="r", marker=".")
plt.xlabel('ballsFaced')
plt.ylabel(' runs')
plt.title("match")
plt.savefig("scatterplot")
plt.show()
```



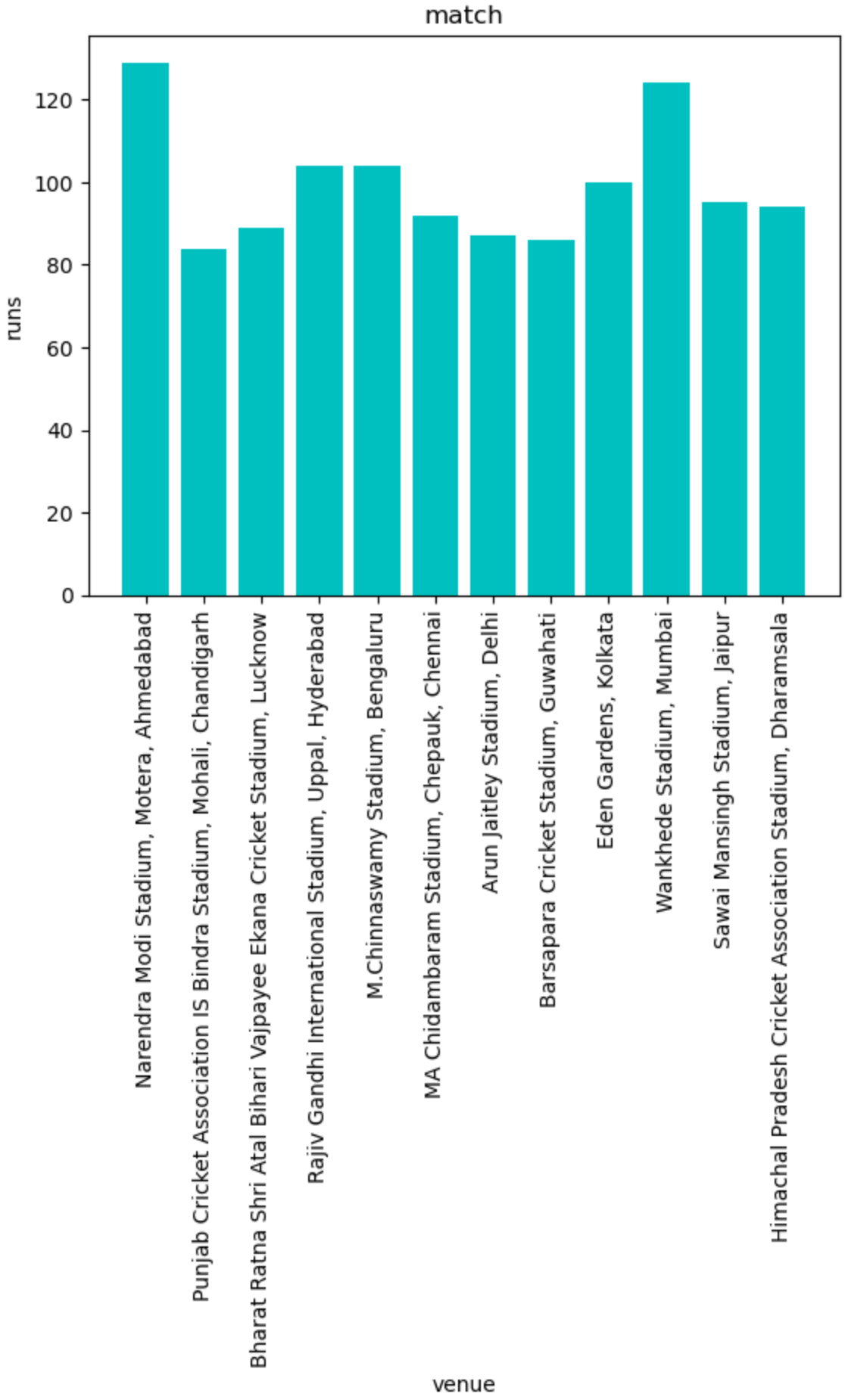
In [73]:
```
#line plot
plt.plot(test1['runs'],label = ['fours'], linewidth='1')
plt.xlabel("runs")
plt.ylabel("fours")
plt.title("match")
plt.savefig("lineplot")
plt.show()
```



In [74]:
```
#box plot
plt.boxplot(test1['runs'], vert=True, showmeans=True,notch=True, patch_artist=T
plt.xlabel('runs')
plt.ylabel('ballsFaced')
plt.title("match")
plt.savefig("boxplot")
plt.show()
```



In [75]:
```
#bar plot
plt.bar(test1['venue'],test1['runs'],color='c')
plt.xlabel('venue')
plt.ylabel('runs')
plt.title("match")
plt.xticks(rotation=90)
plt.savefig("barplot")
plt.show()
```


match
runs
0
20
40
60
80
100
120
Narendra Modi Stadium, Motera, Ahmedabad
Punjab Cricket Association IS Bindra Stadium, Mohali, Chandigarh
Bharat Ratna Shri Atal Bihari Vajpayee Ekana Cricket Stadium, Lucknow
Rajiv Gandhi International Stadium, Uppal, Hyderabad
M.Chinnaswamy Stadium, Bengaluru
MA Chidambaram Stadium, Chepauk, Chennai
Arun Jaitley Stadium, Delhi
Barsapara Cricket Stadium, Guwahati
Eden Gardens, Kolkata
Wankhede Stadium, Mumbai
Sawai Mansingh Stadium, Jaipur
Himachal Pradesh Cricket Association Stadium, Dharamsala
venue

In [76]:
```
#Indexing and Slicing
test=['home_team','away_team','runs','sixes','fours']
```

In [77]:
```
test[::4]
```

Out[77]: ['home_team', 'fours']

In [78]:
```
test[-1]
```

Out[78]: 'fours'

In [79]:
```
test[0]
```

Out[79]: 'home_team'

In [80]:
```
test[2::]
```

Out[80]: ['runs', 'sixes', 'fours']

In [81]:
```
print(test)
```

```
['home_team', 'away_team', 'runs', 'sixes', 'fours']
```

In [82]:
```
test=(['home_team','away_team','runs','sixes','fours'])
```

In [83]:
```
test[0]
```

Out[83]: 'home_team'

In [84]:
```
test[-2]
```

Out[84]: 'sixes'

In [85]:
```
test={'home_team','away_team','runs','sixes','fours'}
```

In [86]:
```
type(test)
```

Out[86]: set

In [87]:
```
test=('home_team','away_team','runs','sixes','fours')
```

In [88]:
```
type(test)
```

Out[88]: tuple

In [ ]:
```

```

In [92]:
```
#Grouping With Aggregation
test1
```

Out[92]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	count
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
3	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
...	...	...	...	...	...	...	...	
1176	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1177	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1178	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1179	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1180	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc

1181 rows × 23 columns

In [156]:
```
test1.groupby(['home_team','city']).runs.agg('mean','min')
```

Out[156]:

home_team	city	
CSK	Chennai	19.076923
DC	Delhi	18.008929
	Dharamsala	30.461538
GT	Ahmedabad	20.563758
	Bengaluru	50.666667
	Chandigarh	9.500000
	Chennai	12.000000
	Delhi	22.000000
	Guwahati	36.000000
	Hyderabad	15.000000
	Jaipur	10.000000
	Kolkata	2.000000
	Lucknow	0.000000
	Mumbai	12.000000
KKR	Kolkata	22.045455
	Mumbai	14.333333
LSG	Chennai	12.850000
	Lucknow	15.159292
MI	Mumbai	25.477273
PBKS	Chandigarh	22.746667
	Dharamsala	24.066667
RCB	Bengaluru	21.800000
RR	Guwahati	20.562500
	Jaipur	20.589041
SRH	Hyderabad	18.882883

Name: runs, dtype: float64

In [94]:
```
test1.groupby(['home_team','city']).runs.agg(['median','max'])
```

Out[94]:

home_team	city	median	max
CSK	Chennai	15.0	92.0
DC	Delhi	10.5	87.0
	Dharamsala	22.0	94.0
GT	Ahmedabad	12.0	129.0
	Bengaluru	47.0	82.0
	Chandigarh	9.5	12.0
	Chennai	12.0	12.0
	Delhi	22.0	43.0
	Guwahati	36.0	36.0
	Hyderabad	15.0	15.0
	Jaipur	10.0	10.0
	Kolkata	2.0	2.0
	Lucknow	0.0	0.0
	Mumbai	12.0	12.0
KKR	Kolkata	15.0	100.0
	Mumbai	12.0	43.0
LSG	Chennai	9.5	41.0
	Lucknow	8.0	89.0
MI	Mumbai	16.5	124.0
PBKS	Chandigarh	19.0	84.0
	Dharamsala	19.0	51.0
RCB	Bengaluru	12.0	104.0
RR	Guwahati	9.5	86.0
	Jaipur	15.0	95.0
SRH	Hyderabad	12.0	104.0

In [95]:
```
test1.groupby(['current_innings','city']).runs.agg(['std'])
```

Out[95]:

		std
current_innings	**city**	
CSK	**Ahmedabad**	13.048627
	Bengaluru	28.369690
	Chennai	18.964217
	Delhi	37.567273
	Jaipur	19.270011
...	**...**	...
SRH	**Hyderabad**	19.563773
	Jaipur	18.283482
	Kolkata	35.116947
	Lucknow	13.055863
	Mumbai	34.425904

82 rows × 1 columns

In [96]:
```
test1.groupby(['home_team','away_team',]).runs.agg(['var'])
```

Out[96]:

		var
home_team	**away_team**	
CSK	**DC**	114.260526
	GT	265.315789
	KKR	403.192308
	LSG	287.869281
	MI	314.691176
...	**...**	...
SRH	**LSG**	293.923077
	MI	339.202614
	PBKS	935.302198
	RCB	1118.622222
	RR	385.007353

75 rows × 1 columns

In [97]:
```
#Measuring Percantile
test1.describe()
```

Out[97]:

	Unnamed: 0	match_id	innings_id	runs	ballsFaced	fours	sixes
count	1181.0	1.181000e+03	1181.000000	1181.000000	1181.000000	1181.000000	1181.000000
mean	2023.0	1.360124e+06	1.508891	20.180356	14.504657	1.813717	0.929721
std	0.0	2.509370e+03	0.500133	22.294496	13.377163	2.434745	1.556606
min	2023.0	1.359475e+06	1.000000	0.000000	0.000000	0.000000	0.000000
25%	2023.0	1.359493e+06	1.000000	3.000000	4.000000	0.000000	0.000000
50%	2023.0	1.359510e+06	2.000000	12.000000	10.000000	1.000000	0.000000
75%	2023.0	1.359529e+06	2.000000	29.000000	21.000000	3.000000	1.000000
max	2023.0	1.370353e+06	2.000000	129.000000	66.000000	16.000000	10.000000

In [98]:
```
test1['runs'].describe()
```

Out[98]:
```
count    1181.000000
mean       20.180356
std        22.294496
min         0.000000
25%         3.000000
50%        12.000000
75%        29.000000
max       129.000000
Name: runs, dtype: float64
```

In [99]:
```
test1['runs'].quantile(0.50)
```

Out[99]:
```
12.0
```

In [100]:
```
test1['ballsFaced'].describe()
```

Out[100]:
```
count    1181.000000
mean       14.504657
std        13.377163
min         0.000000
25%         4.000000
50%        10.000000
75%        21.000000
max        66.000000
Name: ballsFaced, dtype: float64
```

In [101]:
```
test1['ballsFaced'].quantile(0.25)
```

Out[101]:
```
4.0
```

In [102]:
```
test1['fours'].describe()
```

Out[102]:
```
count    1181.000000
mean        1.813717
std         2.434745
min         0.000000
25%         0.000000
50%         1.000000
75%         3.000000
max        16.000000
Name: fours, dtype: float64
```

In [103]:
```
test1['fours'].quantile(1)
```

Out[103]:
```
16.0
```

In [104]:
```
#Shape Distribution(Skewness, Kurtosis, Frequency Table)
import seaborn as sns
```

In [105]:
```
test1
```

Out[105]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	count
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
3	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
...	...	...	...	...	...	...	...	
1176	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1177	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1178	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1179	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1180	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc

1181 rows × 23 columns

In [106]:
```
test1.skew()
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\556298337.py:1: FutureWarning:
The default value of numeric_only in DataFrame.skew is deprecated. In a futur
e version, it will default to False. In addition, specifying 'numeric_only=No
ne' is deprecated. Select only valid columns or specify the value of numeric_
only to silence this warning.
  test1.skew()
```

Out[106]:
```
Unnamed: 0     0.000000
match_id       3.836848
innings_id    -0.035614
runs           1.529380
ballsFaced     1.287755
fours          1.948378
sixes          2.327727
captain        2.350637
isNotOut       1.323008
runningOver   -0.473828
dtype: float64
```

In [107]:
```
#Kurtosis
test1.kurt()
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\2675755251.py:2: FutureWarnin
g: The default value of numeric_only in DataFrame.kurt is deprecated. In a fu
ture version, it will default to False. In addition, specifying 'numeric_only
=None' is deprecated. Select only valid columns or specify the value of numer
ic_only to silence this warning.
  test1.kurt()
```

Out[107]:
```
Unnamed: 0      0.000000
match_id       12.744110
innings_id     -2.002125
runs            2.212996
ballsFaced      1.137010
fours           4.524300
sixes           6.412719
captain         3.531473
isNotOut       -0.250076
runningOver    -0.580940
dtype: float64
```

In [108]:
```
#distplot
sns.distplot(test1['match_id'])
plt.savefig('Yashwanth_Distplot1.png')
```

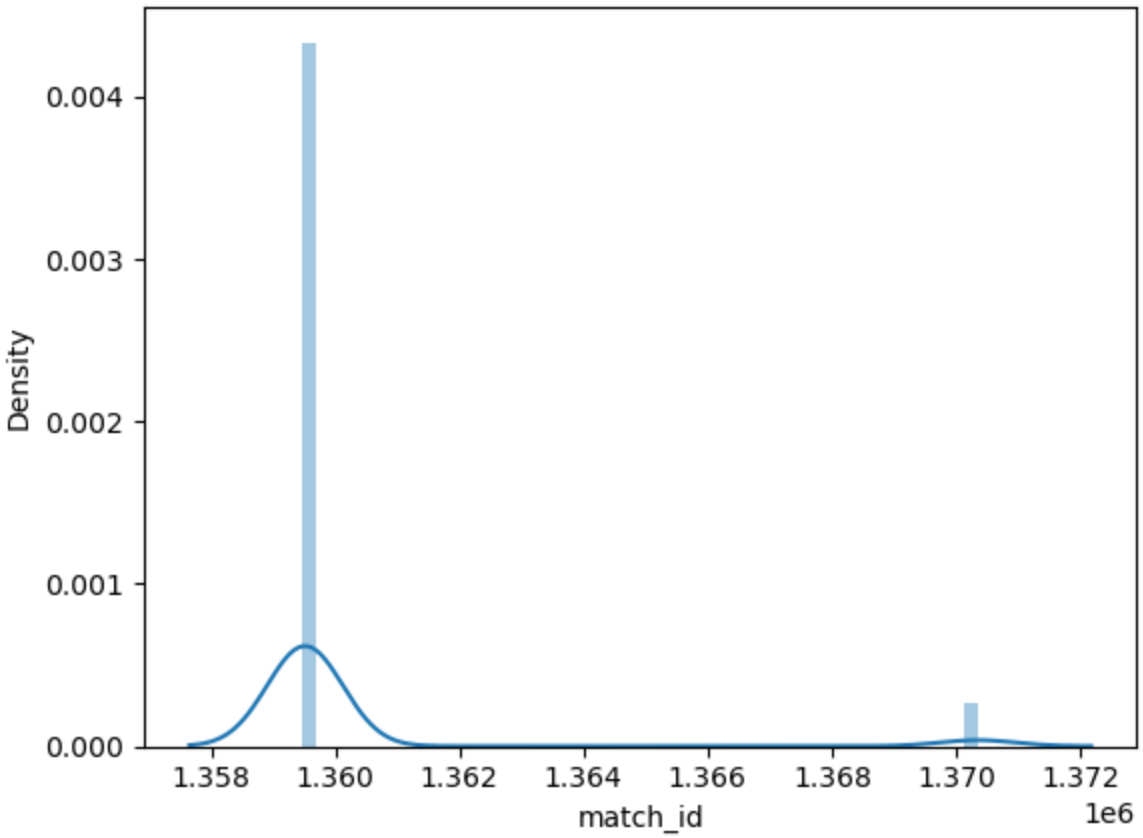
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\3138998622.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['match_id'])
```



In [109]:
```
sns.distplot(test1['innings_id'])
plt.savefig('Yashwanth_Distplot2.png')
```

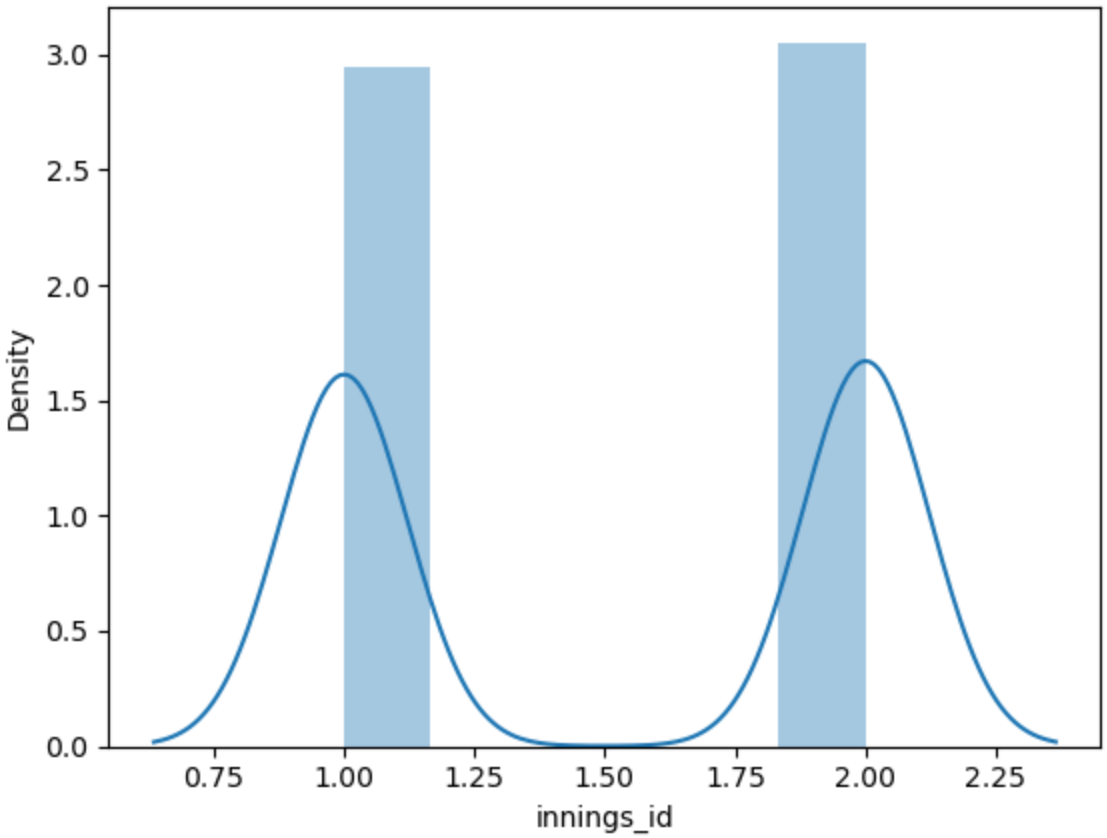
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\2952858844.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['innings_id'])
```



In [110]:
```
sns.distplot(test1['runs'])
plt.savefig('Yashwanth_Distplot1.png')
```

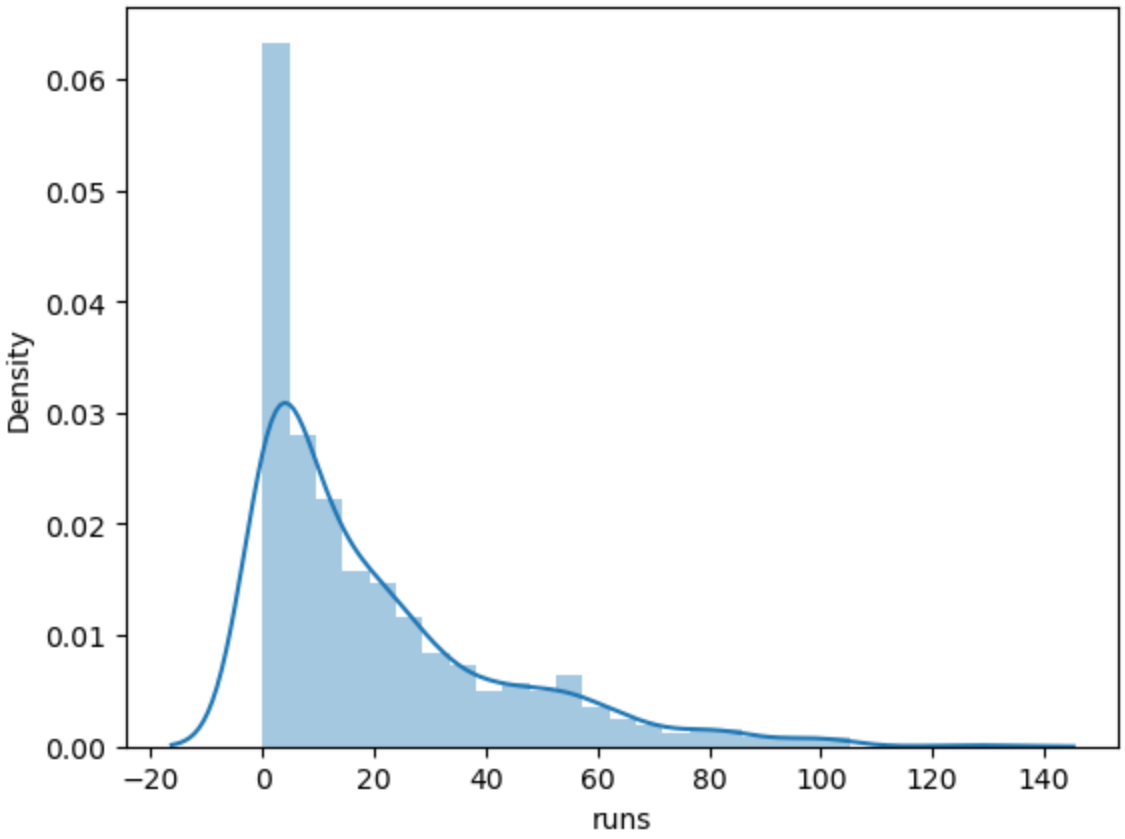
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\2014188942.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['runs'])
```



In [111]:
```
sns.distplot(test1['ballsFaced'])
plt.savefig('Yashwanth_Distplot1.png')
```

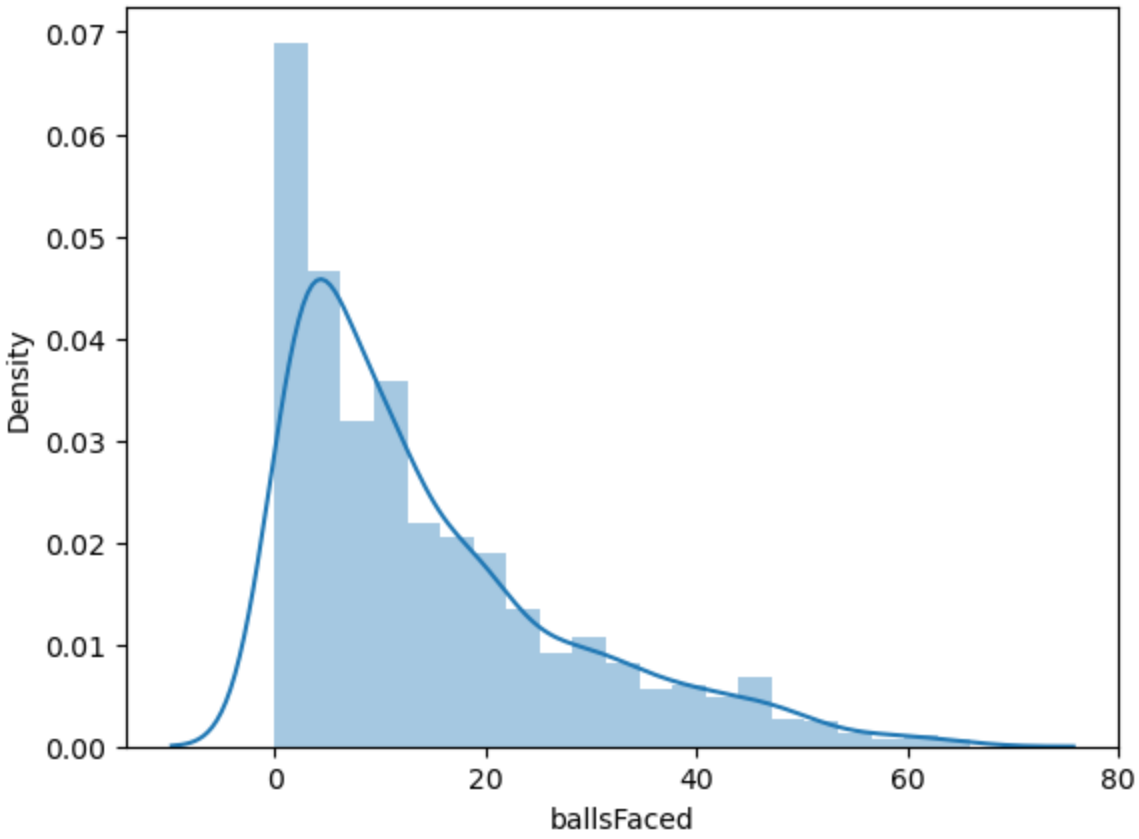
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\396210273.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['ballsFaced'])
```



In [112]:
```
sns.distplot(test1['fours'])
plt.savefig('Yashwanth_Distplot1.png')
```

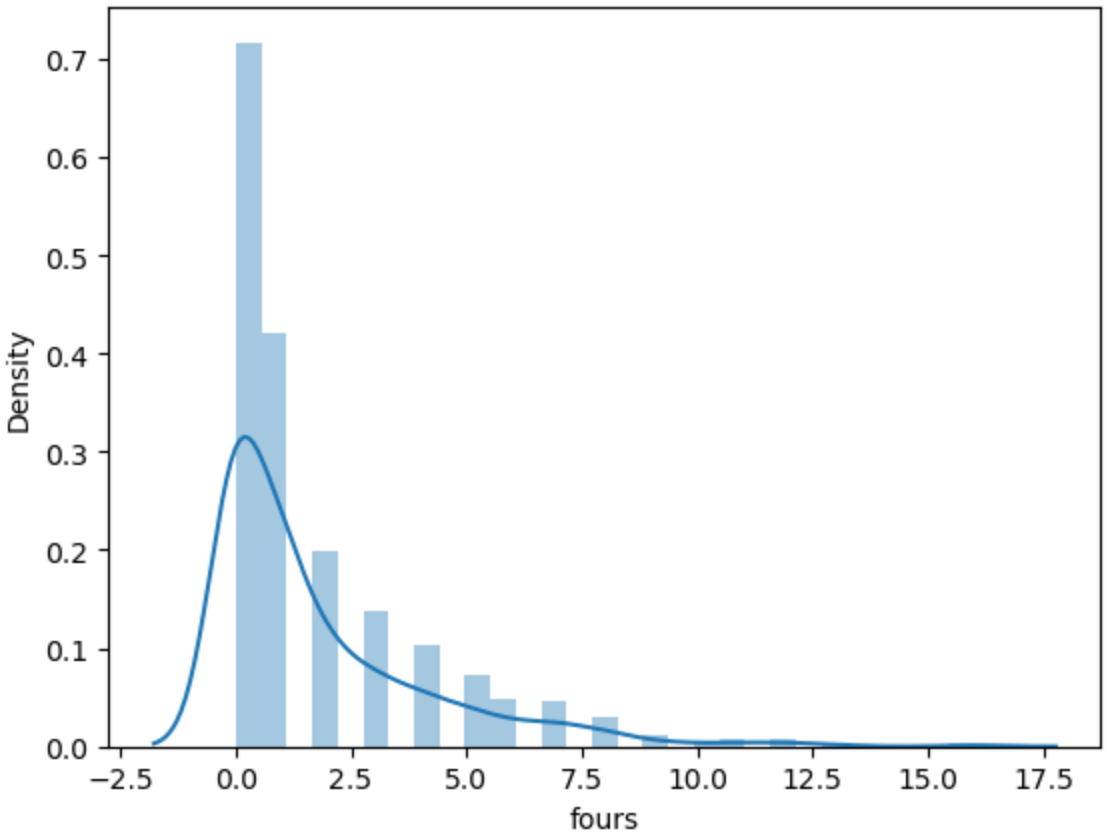
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\2453800341.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['fours'])
```



In [113]:
```
sns.distplot(test1['sixes'])
plt.savefig('Yashwanth_Distplot1.png')
```

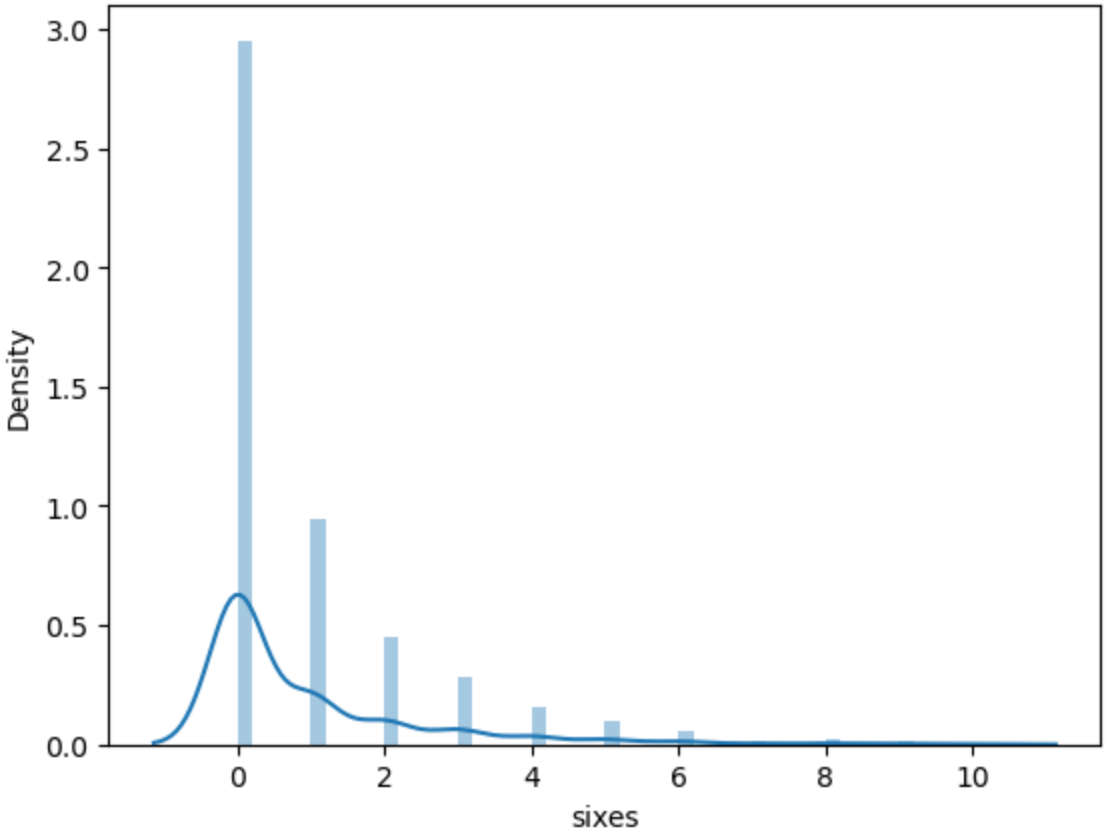
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\3919911415.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['sixes'])
```



In [114]:
```
sns.distplot(test1['captain'])
plt.savefig('Yashwanth_Distplot1.png')
```

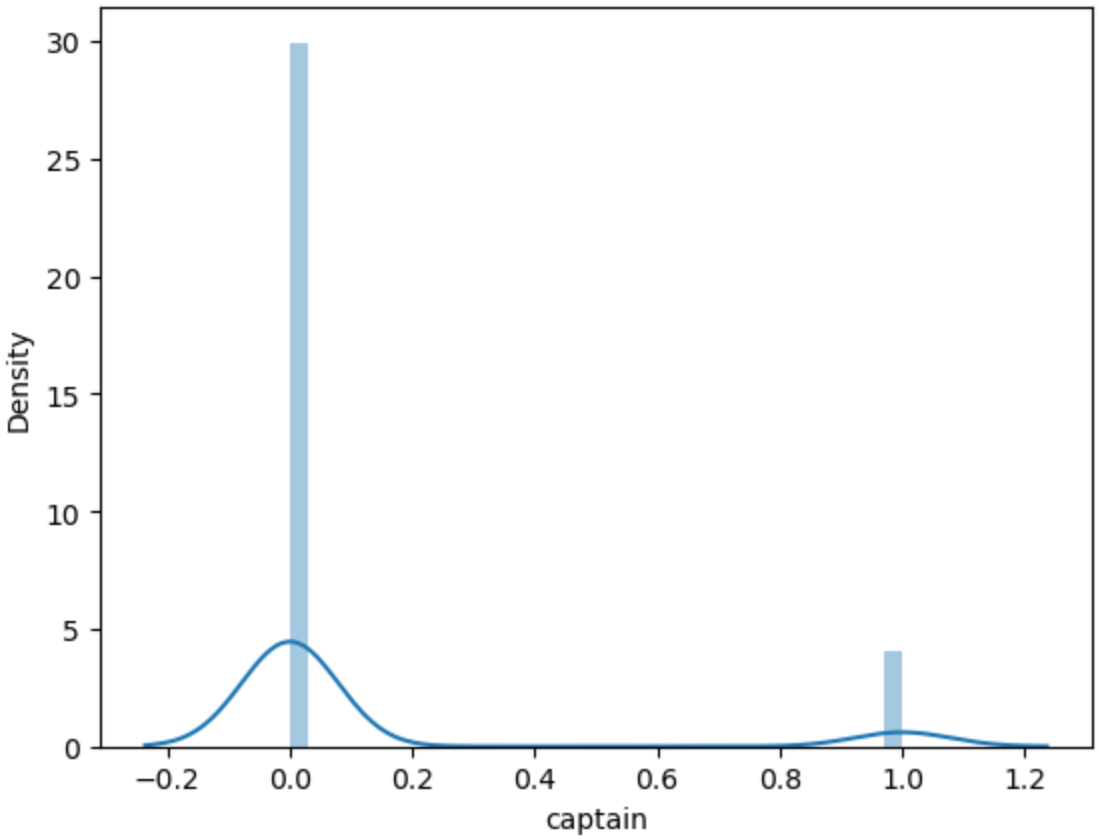
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\1206168601.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['captain'])
```



In [115]:
```
sns.distplot(test1['isNotOut'])
plt.savefig('Yashwanth_Distplot1.png')
```

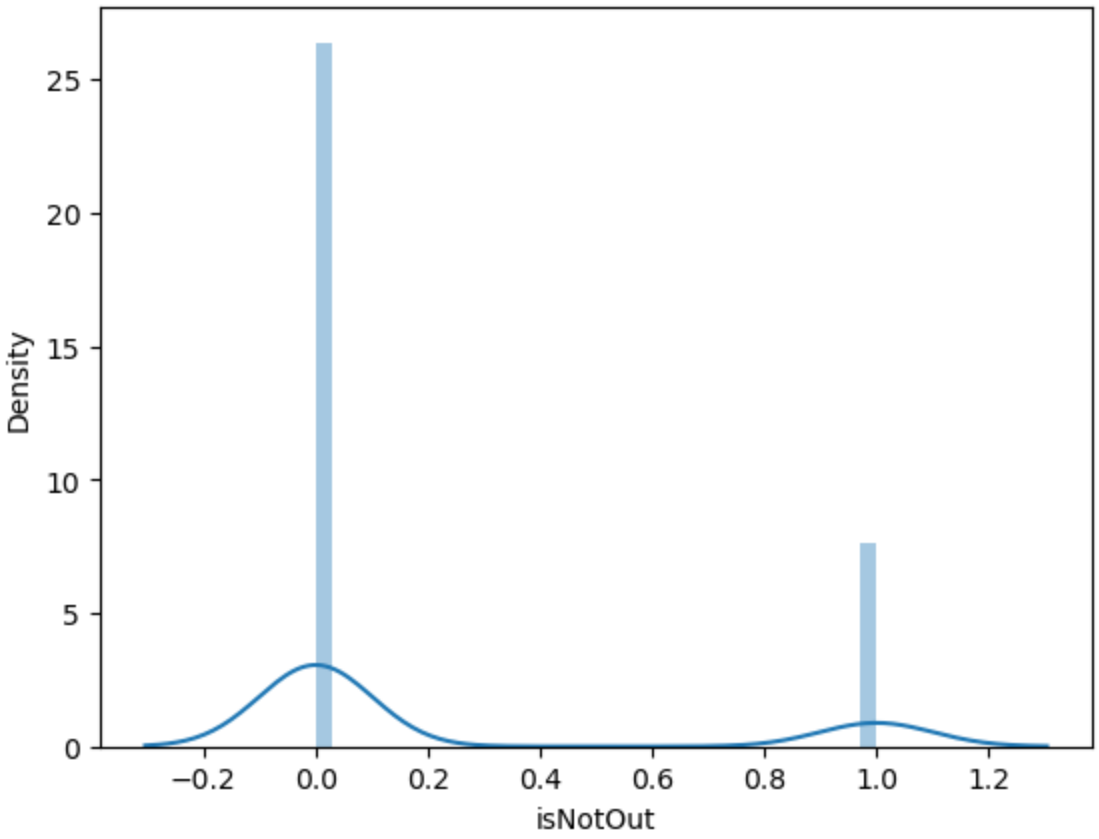
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\2029658573.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['isNotOut'])
```



In [116]:
```
sns.distplot(test1['runningOver'])
plt.savefig('Yashwanth_Distplot1.png')
```

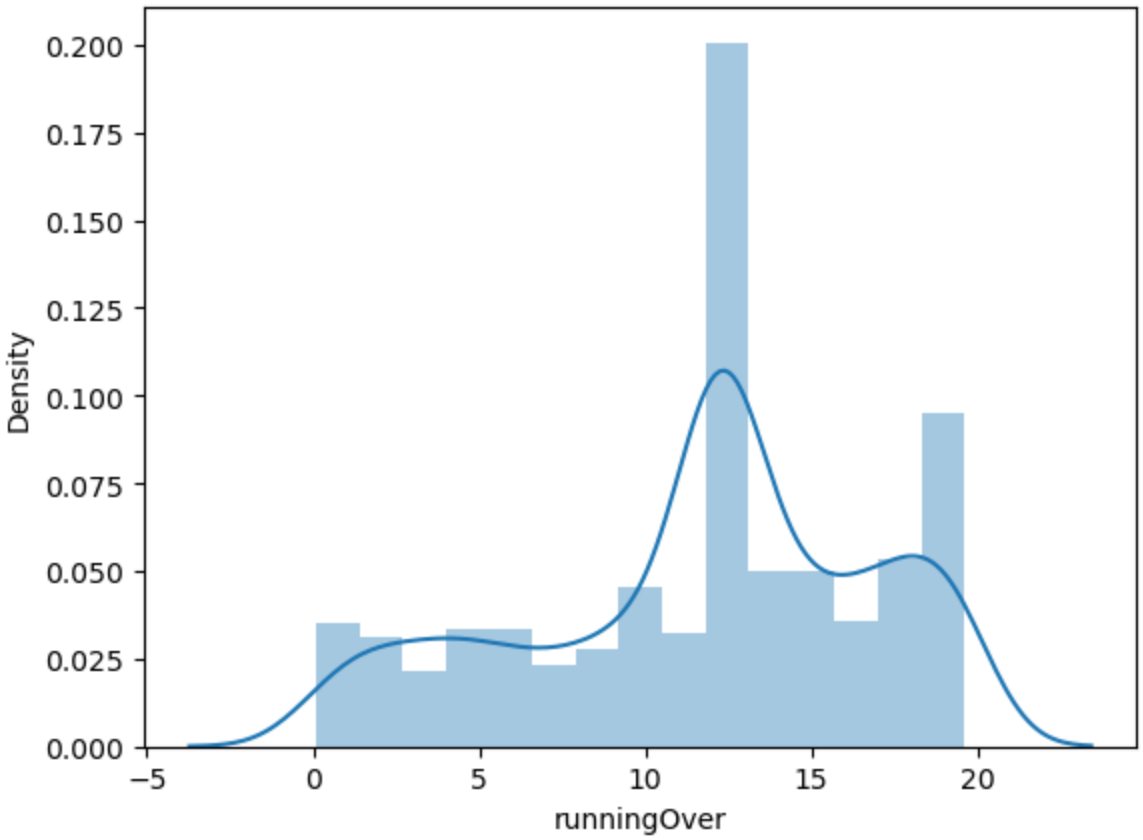
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\3079190271.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(test1['runningOver'])
```



In [117]:
```
#value count
test1['match_name'].value_counts()
```

Out[117]:
```
GT v MI      35
LSG v RCB    33
LSG v MI     32
GT v CSK     28
GT v SRH     22
             ..
PBKS v MI    11
DC v RCB     11
MI v SRH     11
SRH v RCB    10
LSG v CSK     8
Name: match_name, Length: 71, dtype: int64
```

In [118]:
```
test1['strikeRate'].value_counts()
```

Out[118]:
```
0         121
100        83
50         62
150        29
66.66      27
          ...
111.47      1
126.31      1
170.58      1
146.42      1
188         1
Name: strikeRate, Length: 380, dtype: int64
```

In [119]:
```
#Heatmap and Correlation.
test1.head(1)
```

Out[119]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	country
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	India

1 rows × 23 columns

In [120]:
```
test1.corr()
```

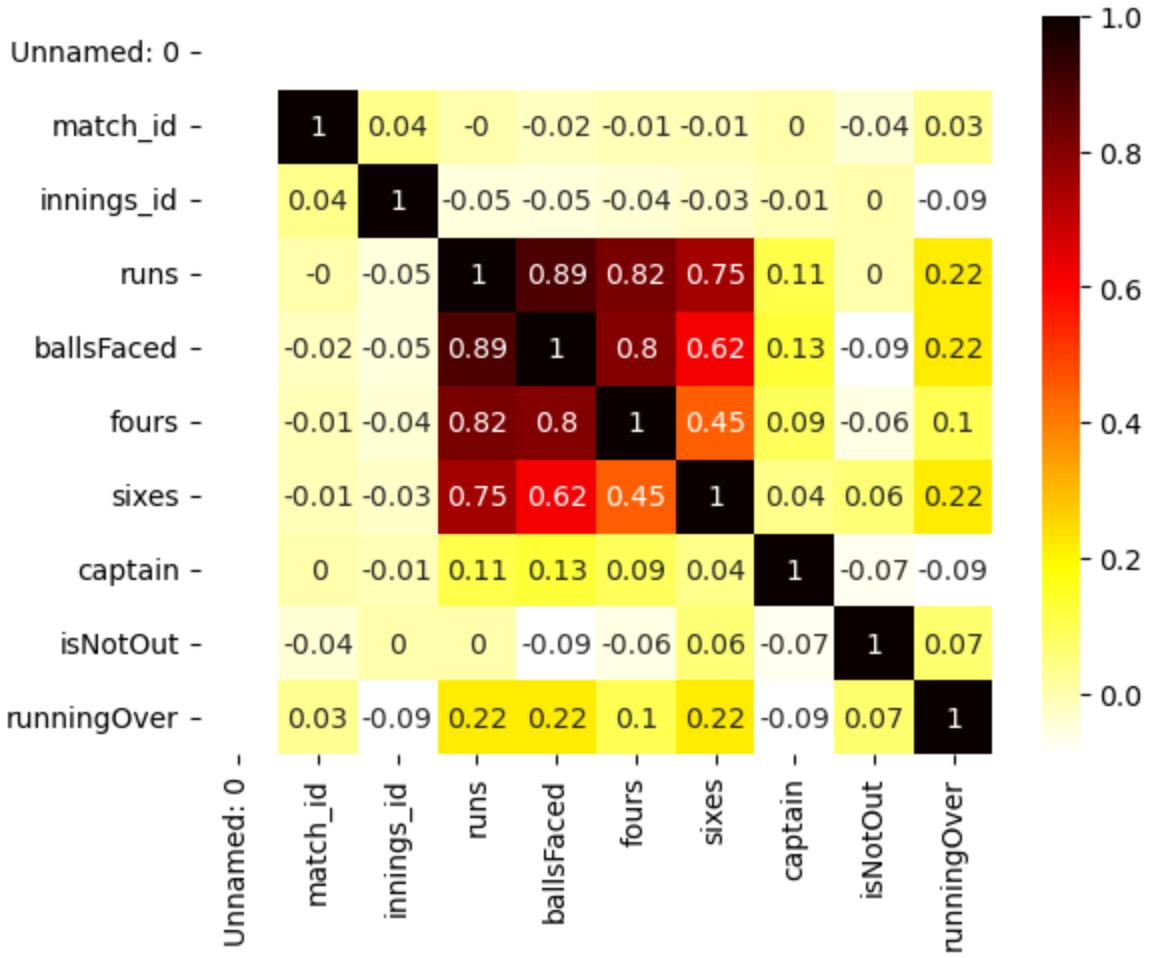
```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\1350350619.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.corr is deprecated. In a fu
ture version, it will default to False. Select only valid columns or specify
the value of numeric_only to silence this warning.
  test1.corr()
```

Out[120]:

	Unnamed: 0	match_id	innings_id	runs	ballsFaced	fours	sixes	capt
Unnamed: 0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
match_id	NaN	1.000000	0.043252	-0.003408	-0.017098	-0.006376	-0.009772	0.000
innings_id	NaN	0.043252	1.000000	-0.048140	-0.047919	-0.041092	-0.033486	-0.014
runs	NaN	-0.003408	-0.048140	1.000000	0.893231	0.822841	0.747952	0.109
ballsFaced	NaN	-0.017098	-0.047919	0.893231	1.000000	0.797814	0.619178	0.133
fours	NaN	-0.006376	-0.041092	0.822841	0.797814	1.000000	0.447558	0.089
sixes	NaN	-0.009772	-0.033486	0.747952	0.619178	0.447558	1.000000	0.040
captain	NaN	0.000118	-0.014384	0.109632	0.133943	0.089345	0.040128	1.000
isNotOut	NaN	-0.043981	0.000584	0.003753	-0.085722	-0.061411	0.062124	-0.066
runningOver	NaN	0.030243	-0.087820	0.221199	0.217636	0.097968	0.216130	-0.085

In [121]:
```
sns.heatmap(test1.corr().round(2), annot=True,cmap='hot_r')
plt.savefig('Yashwanth_Heatmap1.png')
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_17240\3736218771.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.corr is deprecated. In a fu
ture version, it will default to False. Select only valid columns or specify
the value of numeric_only to silence this warning.
  sns.heatmap(test1.corr().round(2), annot=True,cmap='hot_r')
```



In [122]:
```
#Slicing and Minmax Scaling.
test3=pd.read_csv('batting_card_ready.csv')
test3
```

Out[122]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	count
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
3	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
...	...	...	...	...	...	...	...	
1176	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1177	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1178	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1179	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1180	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc

1181 rows × 23 columns

In [123]:
```
test3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1181 entries, 0 to 1180
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Unnamed: 0       1181 non-null   float64
 1   match_id         1181 non-null   int64  
 2   match_name       1181 non-null   object 
 3   home_team        1181 non-null   object 
 4   away_team        1181 non-null   object 
 5   venue            1181 non-null   object 
 6   city             1181 non-null   object 
 7   country          1181 non-null   object 
 8   current_innings  1181 non-null   object 
 9   innings_id       1181 non-null   int64  
 10  name             1181 non-null   object 
 11  fullName         1181 non-null   object 
 12  runs             1181 non-null   float64
 13  ballsFaced       1181 non-null   float64
 14  minutes          1181 non-null   object 
 15  fours            1181 non-null   float64
 16  sixes            1181 non-null   float64
 17  strikeRate       1181 non-null   object 
 18  captain          1181 non-null   bool   
 19  isNotOut         1181 non-null   bool   
 20  runningScore     1181 non-null   object 
 21  runningOver      1181 non-null   float64
 22  shortText        1181 non-null   object 
dtypes: bool(2), float64(6), int64(2), object(13)
memory usage: 196.2+ KB
```

In [124]:
```
#Dropping non-numerical Columns

test3.drop(['match_name','home_team','away_team','venue','city','captain','isNo
```

In [125]:
```
test3.info()
```

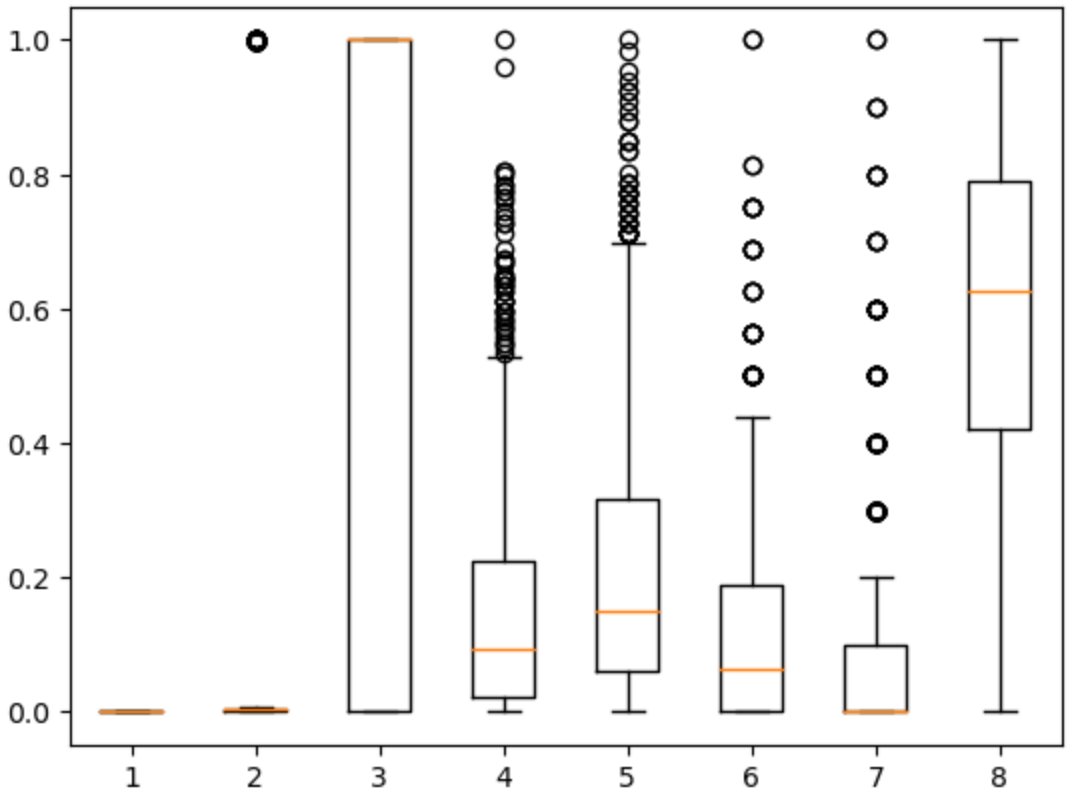
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1181 entries, 0 to 1180
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Unnamed: 0   1181 non-null   float64
 1   match_id     1181 non-null   int64  
 2   innings_id   1181 non-null   int64  
 3   runs         1181 non-null   float64
 4   ballsFaced   1181 non-null   float64
 5   fours        1181 non-null   float64
 6   sixes        1181 non-null   float64
 7   runningOver  1181 non-null   float64
dtypes: float64(6), int64(2)
memory usage: 73.9 KB
```

In [126]:
```
from sklearn.preprocessing import MinMaxScaler
minscaler=MinMaxScaler()
minscaler_data=minscaler.fit_transform(test3)
minscaled_data=pd.DataFrame(minscaler_data, columns=test3.columns)
minscaled_data.head()
```

Out[126]:

	Unnamed: 0	match_id	innings_id	runs	ballsFaced	fours	sixes	runningOver
0	0.0	0.0	0.0	0.007752	0.090909	0.0000	0.0	0.107692
1	0.0	0.0	0.0	0.093023	0.757576	0.2500	0.9	0.871795
2	0.0	0.0	0.0	0.178295	0.257576	0.2500	0.1	0.276923
3	0.0	0.0	0.0	0.054264	0.151515	0.0625	0.0	0.374359
4	0.0	0.0	0.0	0.093023	0.181818	0.0000	0.1	0.635897

In [127]:
```
plt.boxplot(minscaled_data)
plt.savefig('Yashwanth_Minscaler.png')
plt.show()
```



In [ ]:
```
```

In [134]:
```
#Feature Scaling
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import Normalizer
```

In [135]:
```
test3.head(5)
```

Out[135]:

	Unnamed: 0	match_id	innings_id	runs	ballsFaced	fours	sixes	runningOver
0	2023.0	1359475	1	1.0	6.0	0.0	0.0	2.2
1	2023.0	1359475	1	12.0	50.0	4.0	9.0	17.1
2	2023.0	1359475	1	23.0	17.0	4.0	1.0	5.5
3	2023.0	1359475	1	7.0	10.0	1.0	0.0	7.4
4	2023.0	1359475	1	12.0	12.0	0.0	1.0	12.5

In [136]:
```
test3_1=StandardScaler()
```

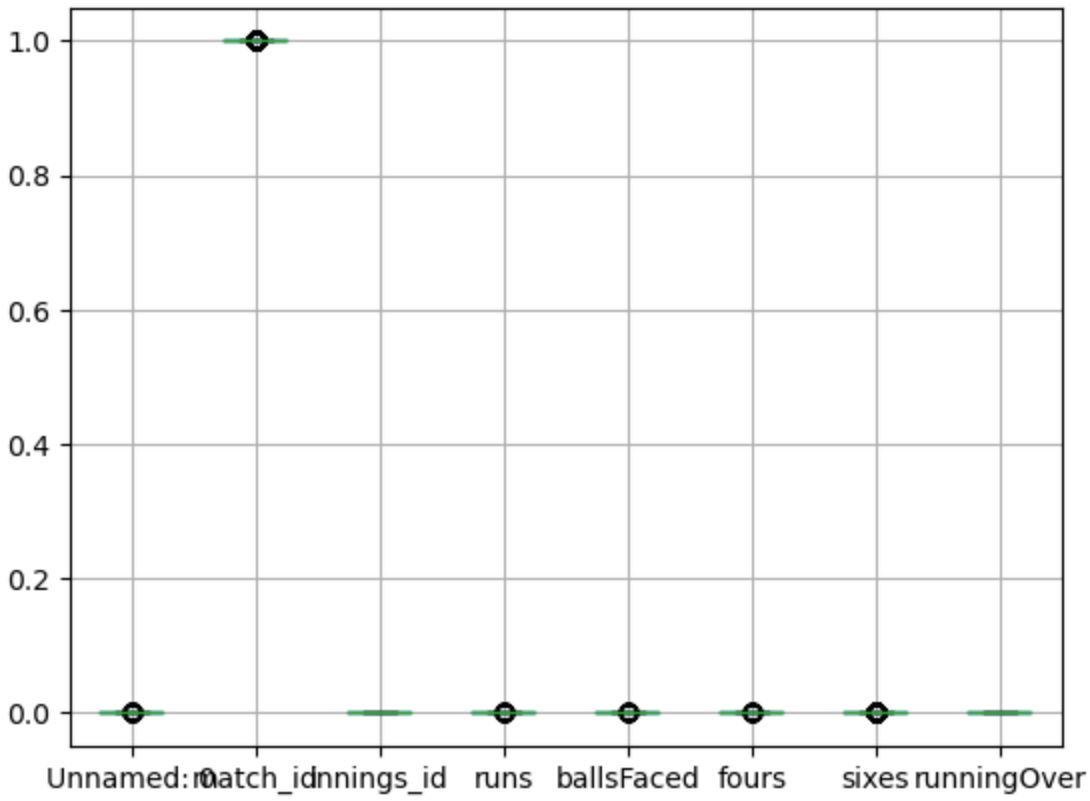
In [137]:
```
test3_data=test3_1.fit_transform(test3)
```

In [138]:
```
test3_data_1=pd.DataFrame(test3_data, columns = test3.columns)
test3_data_1.head(5)
```

Out[138]:

	Unnamed: 0	match_id	innings_id	runs	ballsFaced	fours	sixes	runningOver
0	0.0	-0.258634	-1.017942	-0.860682	-0.636029	-0.745247	-0.597527	-1.804506
1	0.0	-0.258634	-1.017942	-0.367078	2.654552	0.898332	5.186732	1.038287
2	0.0	-0.258634	-1.017942	0.126526	0.186617	0.898332	0.045168	-1.174894
3	0.0	-0.258634	-1.017942	-0.591444	-0.336885	-0.334352	-0.597527	-0.812390
4	0.0	-0.258634	-1.017942	-0.367078	-0.187313	-0.745247	0.045168	0.160646

In [145]:
```
test3_data_1.boxplot()
plt.savefig('Yashwanth_Standardizer_png')
plt.show()
```



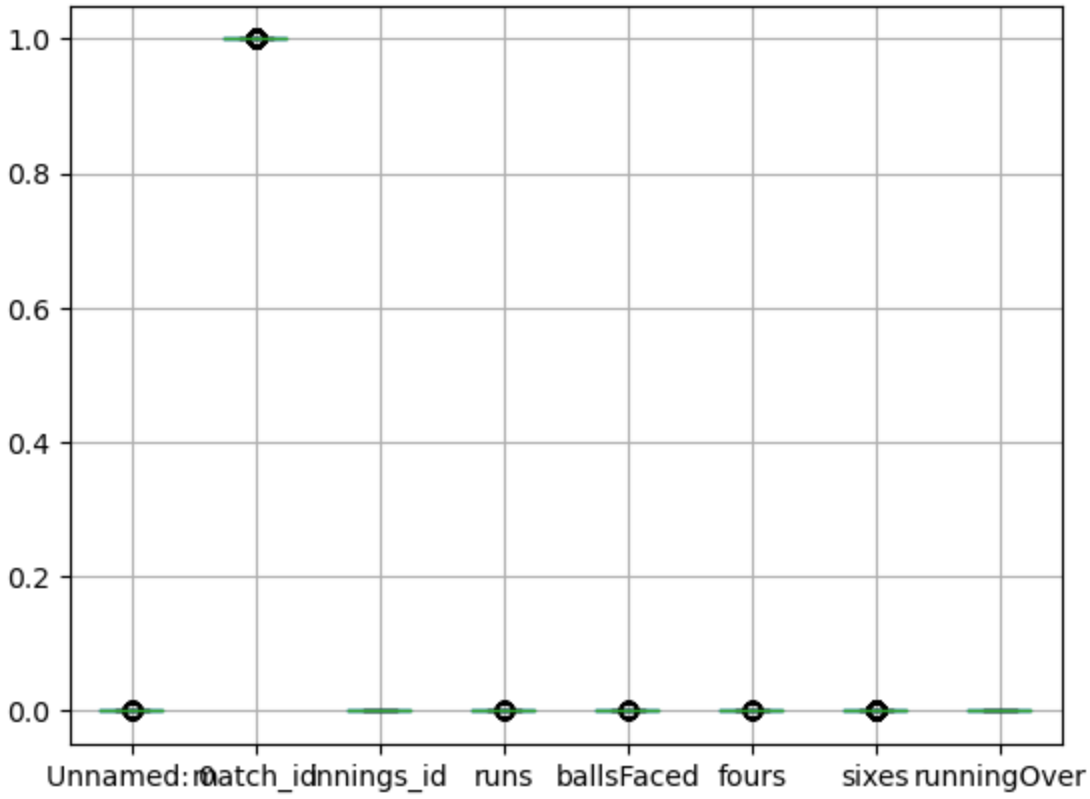
In [139]:
```
test3_2=Normalizer()
test3_2_1=test3_2.fit_transform(test3)
```

In [140]:
```
test3_data_1=pd.DataFrame(test3_2_1, columns = test3.columns)
test3_data_1.head(5)
```

Out[140]:

	Unnamed: 0	match_id	innings_id	runs	ballsFaced	fours	sixes	runningO
0	0.001488	0.999999	7.355773e-07	7.355773e-07	0.000004	0.000000e+00	0.000000e+00	0.000
1	0.001488	0.999999	7.355773e-07	8.826927e-06	0.000037	2.942309e-06	6.620195e-06	0.000
2	0.001488	0.999999	7.355773e-07	1.691828e-05	0.000013	2.942309e-06	7.355773e-07	0.000
3	0.001488	0.999999	7.355773e-07	5.149041e-06	0.000007	7.355773e-07	0.000000e+00	0.000
4	0.001488	0.999999	7.355773e-07	8.826927e-06	0.000009	0.000000e+00	7.355773e-07	0.000

In [146]:
```
test3_data_1.boxplot()
plt.savefig('Yashwanth_Normalizer.png')
plt.show()
```



In [143]:
```
#Categorigal Encoding
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
```

In [152]:
```
test1.head()
```

Out[152]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	country
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	India
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	India
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	India
3	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	India
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	India

5 rows × 23 columns

In [154]:
```
enc=OneHotEncoder()
enc_data=pd.DataFrame(enc.fit_transform(df1[["runs"]]).toarray())
enc_data
```

Out[154]:

	0	1	2	3	4	5	6	7	8	9	...	88	89	90	91	92	93	94	95	96
0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1349	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1350	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1351	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1352	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1353	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

1354 rows × 98 columns

In [ ]:
```
```

In [155]:
```
e1=test1.join(enc_e)
e1
```

Out[155]:

	Unnamed: 0	match_id	match_name	home_team	away_team	venue	city	count
0	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
2	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
3	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
4	2023.0	1359475	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
...	...	...	...	...	...	...	...	
1176	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1177	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1178	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1179	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc
1180	2023.0	1370353	GT v CSK	GT	CSK	Narendra Modi Stadium, Motera, Ahmedabad	Ahmedabad	Inc

1181 rows × 2668 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: