In [2]:
```
##- Input features: Runs, At Bats, Hits, Doubles, Triples, Homeruns, Walks, Strikeouts, Stolen Bases, Runs Allowe

## Output: Number of predicted wins (W)
```

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
# EDA
```

In [4]:
```
df=pd.read_csv('https://raw.githubusercontent.com/dsrscientist/Data-Science-ML-Capstone-Projects/master/baseball.
```

In [5]:
```
df.head(10)
```

Out[5]:

	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	95	724	5575	1497	300	42	139	383	973	104	641	601	3.73	2	8	56	88
1	83	696	5467	1349	277	44	156	439	1264	70	700	653	4.07	2	12	45	86
2	81	669	5439	1395	303	29	141	533	1157	86	640	584	3.67	11	10	38	79
3	76	622	5533	1381	260	27	136	404	1231	68	701	643	3.98	7	9	37	101
4	74	689	5605	1515	289	49	151	455	1259	83	803	746	4.64	7	12	35	86
5	93	891	5509	1480	308	17	232	570	1151	88	670	609	3.80	7	10	34	88
6	87	764	5567	1397	272	19	212	554	1227	63	698	652	4.03	3	4	48	93
7	81	713	5485	1370	246	20	217	418	1331	44	693	646	4.05	0	10	43	77
8	80	644	5485	1383	278	32	167	436	1310	87	642	604	3.74	1	12	60	95
9	78	748	5640	1495	294	33	161	478	1148	71	753	694	4.31	3	10	40	97

In [7]:
```
## checking for null values
```

In [14]:
```
df.shape
```

Out[14]:
```
(30, 17)
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
W      0
R      0
AB     0
H      0
2B     0
3B     0
HR     0
BB     0
SO     0
SB     0
RA     0
ER     0
ERA    0
CG     0
SHO    0
SV     0
E      0
dtype: int64
```

In [9]:
```
## there are no missing values in the dataset..so we can move ahead
```

In [10]:
```
df.dtypes
```

Out[10]:
```
W        int64
```

```
R        int64
AB       int64
H        int64
2B       int64
3B       int64
HR       int64
BB       int64
SO       int64
SB       int64
RA       int64
ER       int64
ERA    float64
CG       int64
SHO      int64
SV       int64
E        int64
dtype: object
```

In [11]:
```
## there are no categorical variables ..
```

In [12]:
```
df.describe()
```

Out[12]:

	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA
count	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	30.00000	30.000000	30.000000
mean	80.966667	688.233333	5516.266667	1403.533333	274.733333	31.300000	163.633333	469.100000	1248.20000	83.500000	688.233333
std	10.453455	58.761754	70.467372	57.140923	18.095405	10.452355	31.823309	57.053725	103.75947	22.815225	72.108005
min	63.000000	573.000000	5385.000000	1324.000000	236.000000	13.000000	100.000000	375.000000	973.00000	44.000000	525.000000
25%	74.000000	651.250000	5464.000000	1363.000000	262.250000	23.000000	140.250000	428.250000	1157.50000	69.000000	636.250000
50%	81.000000	689.000000	5510.000000	1382.500000	275.500000	31.000000	158.500000	473.000000	1261.50000	83.500000	695.500000
75%	87.750000	718.250000	5570.000000	1451.500000	288.750000	39.000000	177.000000	501.250000	1311.50000	96.500000	732.500000
max	100.000000	891.000000	5649.000000	1515.000000	308.000000	49.000000	232.000000	570.000000	1518.00000	134.000000	844.000000

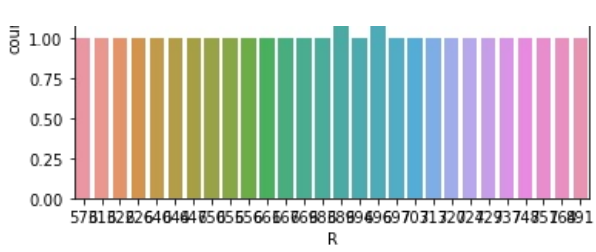
In [13]:
```
df.skew() ## checking the skewness of the data
```

Out[13]:
```
W      0.047089
R      1.200786
AB     0.183437
H      0.670254
2B    -0.230650
3B     0.129502
HR     0.516441
BB     0.158498
SO    -0.156065
SB     0.479893
RA     0.045734
ER     0.058710
ERA    0.053331
CG     0.736845
SHO    0.565790
SV     0.657524
E      0.890132
dtype: float64
```

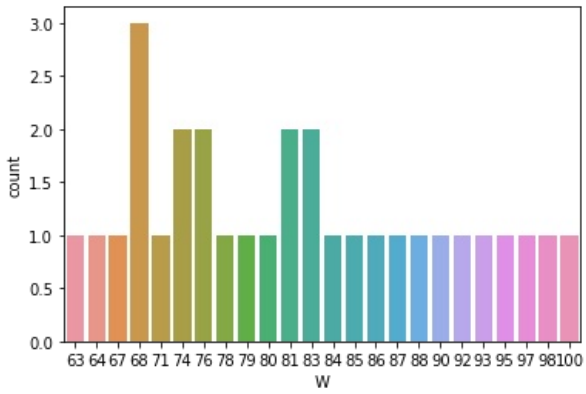
In [15]:
```
## only some independent variables contains skewness but it can be neglected for a while
```

In [16]:
```
## plotting the independent variables
```

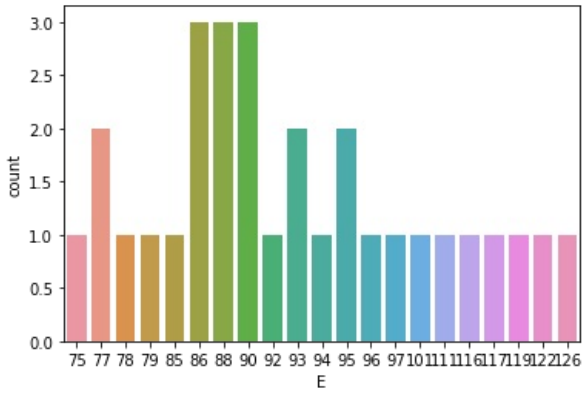
In [17]:
```
sns.countplot(x='R',data=df)
plt.show()
```



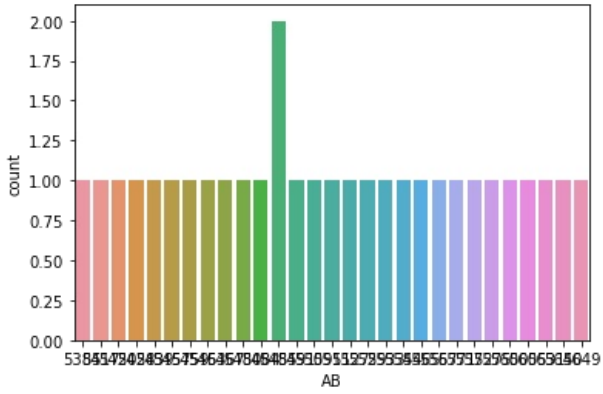
In [18]:
```
sns.countplot(x='W',data=df)
plt.show()
```



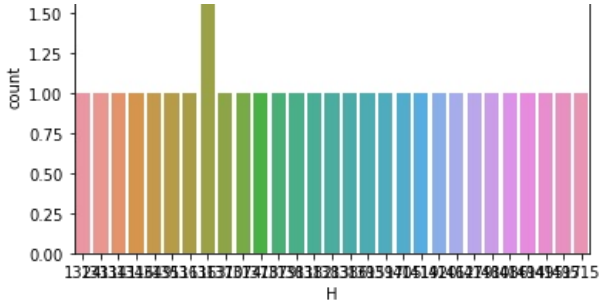
In [19]:
```
sns.countplot(x='E',data=df)
plt.show()
```



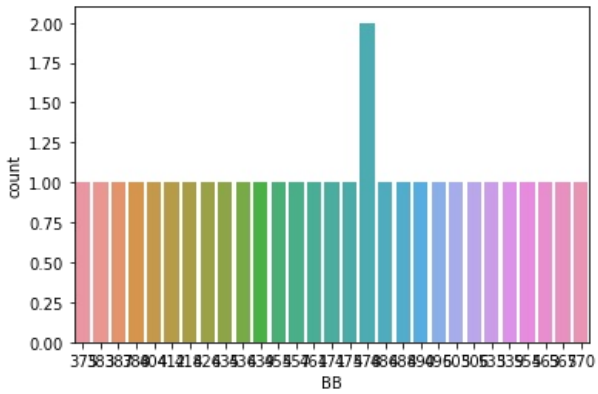
In [20]:
```
sns.countplot(x='AB',data=df)
plt.show()
```



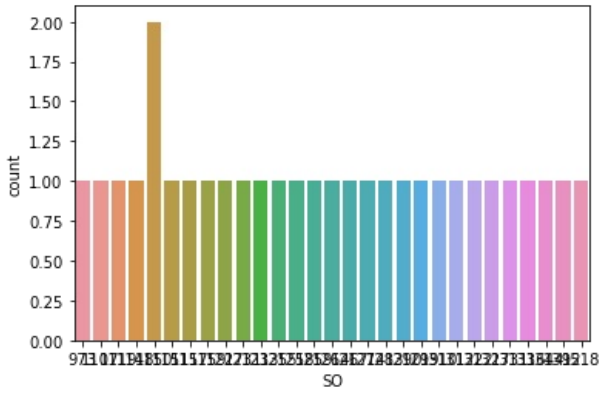
In [21]:
```
sns.countplot(x='H',data=df)
plt.show()
```



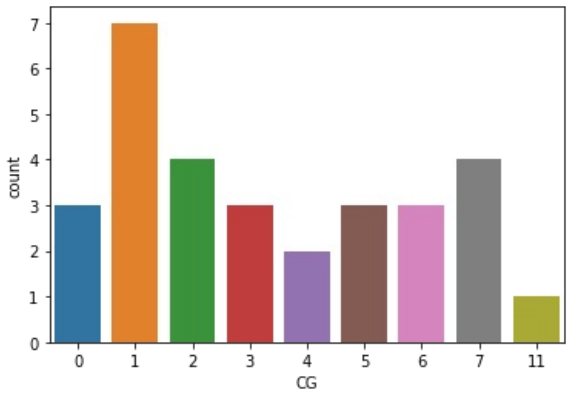
In [22]:
```
sns.countplot(x='BB',data=df)
plt.show()
```



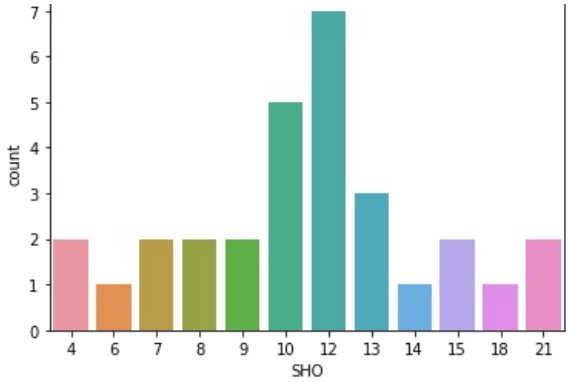
In [23]:
```
sns.countplot(x='SO',data=df)
plt.show()
```



In [24]:
```
sns.countplot(x='CG',data=df)
plt.show()
```

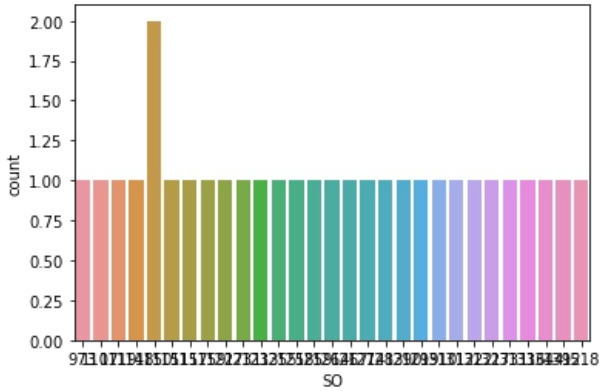


In [25]:
```
sns.countplot(x='SHO',data=df)
plt.show()
```



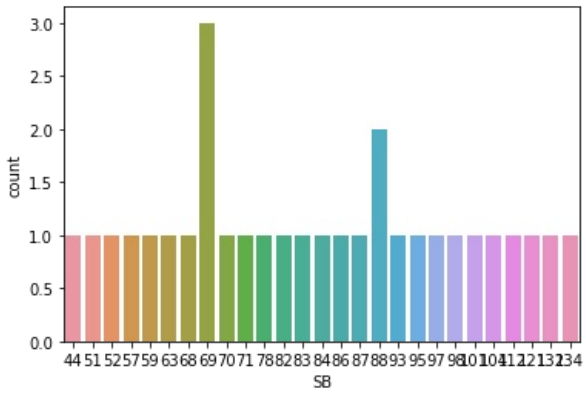
In [26]:

```
sns.countplot(x='SO',data=df)
plt.show()
```



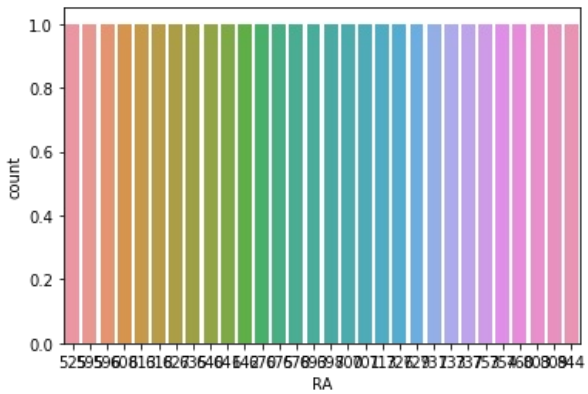
In [27]:

```
sns.countplot(x='SB',data=df)
plt.show()
```



In [28]:

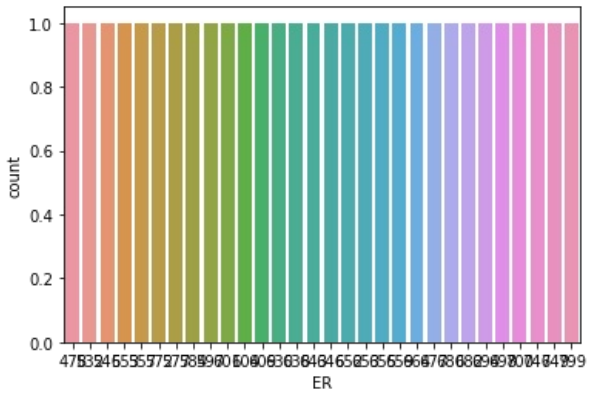
```
sns.countplot(x='RA',data=df)
plt.show()
```



In [29]:

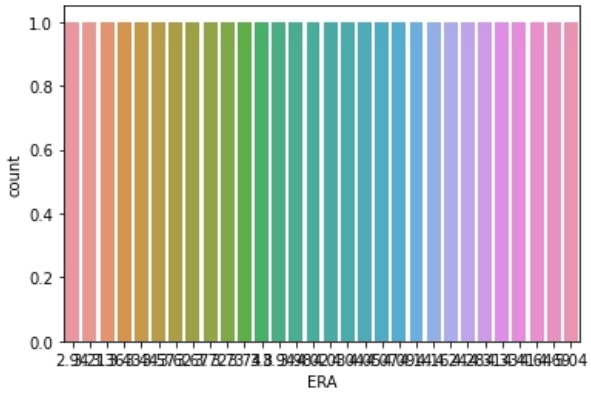
```
sns.countplot(x='ER',data=df)
```

```
plt.show()
```



In [30]:

```
sns.countplot(x='ERA',data=df)
plt.show()
```



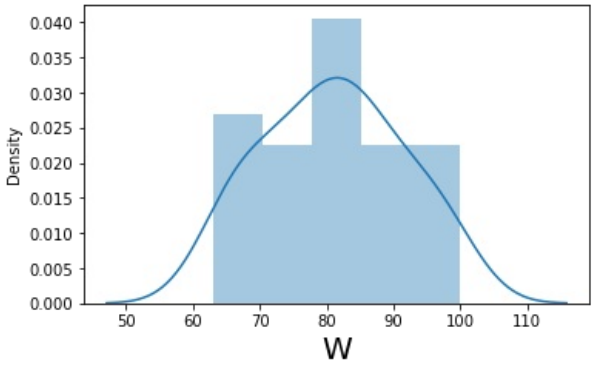
In [31]:

```
## Now we plot the distribution
```

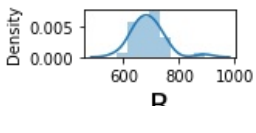
In [32]:

```
plt.figure(figsize=(20,25))
plotnumber=1
for column in df:
    if plotnumber<=17:
        ax=plt.subplot(6,3,plotnumber)
        sns.distplot(df[column])
        plt.xlabel(column,fontsize=20)
        plotnumber+=1
        plt.show()
```

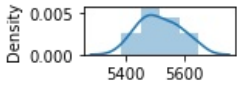
```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

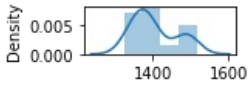


```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



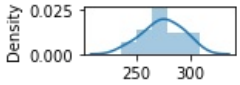
AB

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



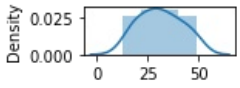
H

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



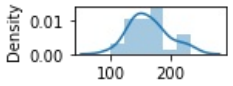
2B

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



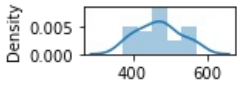
3B

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



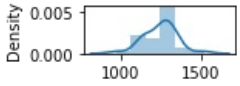
HR

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



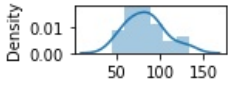
BB

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



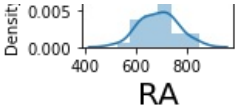
SO

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

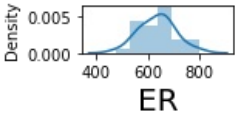


SB

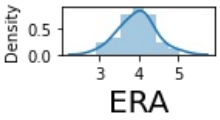
```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



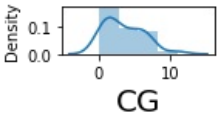

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



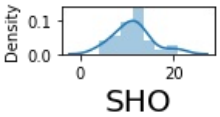

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



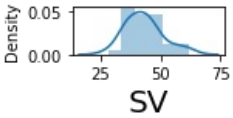

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



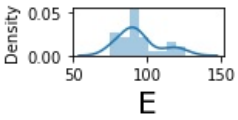

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



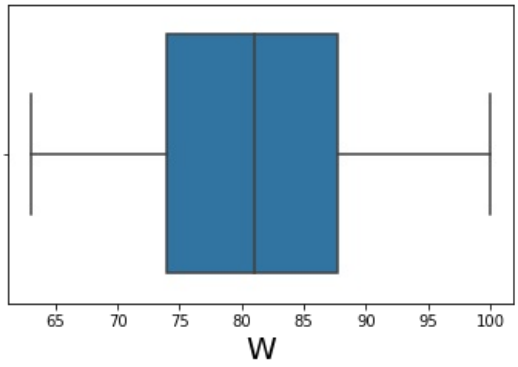

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



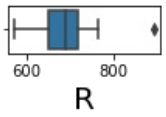
In [33]:

```
plt.figure(figsize=(20,25))
plotnumber=1
for column in df:
    if plotnumber<=17:
        ax=plt.subplot(6,3,plotnumber)
        sns.boxplot(df[column])
        plt.xlabel(column,fontsize=20)
        plotnumber+=1
        plt.show()
```

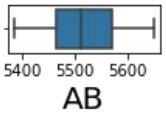
```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



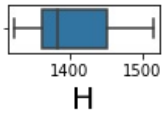

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



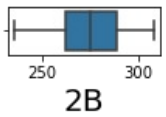

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```




```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

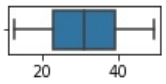


```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



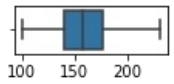
2B

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



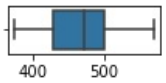
3B

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



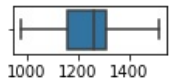
HR

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



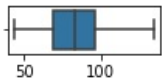
BB

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



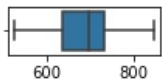
SO

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



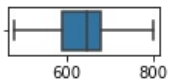
SB

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



RA

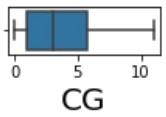
```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



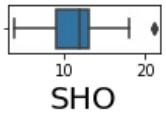
ER

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

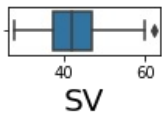

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



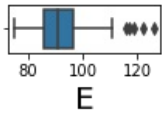

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```




```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```




```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [34]:
```
## SHO,SV,E column contains outliers
```

In [35]:
```
## Removing the outliers
```

In [36]:
```
## using IQR to identify and remove the outliers in the dataset
```

In [37]:
```
q1=df.quantile(0.25)
q3=df.quantile(0.75)
iqr=q3-q1
print(iqr)
```

```
W       13.7500
R       67.0000
AB     106.0000
H       88.5000
2B      26.5000
3B      16.0000
HR      36.7500
BB      73.0000
SO     154.0000
SB      27.5000
RA      96.2500
ER      92.0000
ERA      0.5375
CG       4.7500
SHO      4.0000
SV       9.5000
E       10.7500
dtype: float64
```

In [38]:
```
## validating one outlier
```

In [39]:
```
sho_=(q3.SHO+(1.5*iqr.SHO))
sho_
```

Out[39]:
```
19.0
```

In [40]:
```
## check the index which have higher value
```

In [41]:
```
index=np.where(df['SHO']>sho_)
print(index)
```

```
(array([17, 25], dtype=int64),)
```

In [42]:
```
## drop the index which we found in the above cell
```

In [43]:
```
df=df.drop(df.index[index])
df
```

Out[43]:

	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	95	724	5575	1497	300	42	139	383	973	104	641	601	3.73	2	8	56	88
1	83	696	5467	1349	277	44	156	439	1264	70	700	653	4.07	2	12	45	86
2	81	669	5439	1395	303	29	141	533	1157	86	640	584	3.67	11	10	38	79
3	76	622	5533	1381	260	27	136	404	1231	68	701	643	3.98	7	9	37	101
4	74	689	5605	1515	289	49	151	455	1259	83	803	746	4.64	7	12	35	86
5	93	891	5509	1480	308	17	232	570	1151	88	670	609	3.80	7	10	34	88
6	87	764	5567	1397	272	19	212	554	1227	63	698	652	4.03	3	4	48	93
7	81	713	5485	1370	246	20	217	418	1331	44	693	646	4.05	0	10	43	77
8	80	644	5485	1383	278	32	167	436	1310	87	642	604	3.74	1	12	60	95
9	78	748	5640	1495	294	33	161	478	1148	71	753	694	4.31	3	10	40	97
10	88	751	5511	1419	279	32	172	503	1233	101	733	680	4.24	5	9	45	119
11	86	729	5459	1363	278	26	230	486	1392	121	618	572	3.57	5	13	39	85
12	85	661	5417	1331	243	21	176	435	1150	52	675	630	3.94	2	12	46	93

13	76	656	5544	1379	262	22	198	478	1336	69	726	677	4.16	6	12	45	94
14	68	694	5600	1405	277	46	146	475	1119	78	729	664	4.14	5	15	28	126
15	100	647	5484	1386	288	39	137	506	1267	69	525	478	2.94	1	15	62	96
16	98	697	5631	1462	292	27	140	461	1322	98	596	532	3.21	0	13	54	122
18	68	655	5480	1378	274	34	145	412	1299	84	737	682	4.28	1	7	40	116
19	64	640	5571	1382	257	27	167	496	1255	134	754	700	4.33	2	8	35	90
20	90	683	5527	1351	295	17	177	488	1290	51	613	557	3.43	1	14	50	88
21	83	703	5428	1363	265	13	177	539	1344	57	635	577	3.62	4	13	41	90
22	71	613	5463	1420	236	40	120	375	1150	112	678	638	4.02	0	12	35	77
23	67	573	5420	1361	251	18	100	471	1107	69	760	698	4.41	3	10	44	90
24	63	626	5529	1374	272	37	130	387	1274	88	809	749	4.69	1	7	35	117
26	84	696	5565	1486	288	39	136	457	1159	93	627	597	3.72	7	18	41	78
27	79	720	5649	1494	289	48	154	490	1312	132	713	659	4.04	1	12	44	86
28	74	650	5457	1324	260	36	148	426	1327	82	731	655	4.09	1	6	41	92
29	68	737	5572	1479	274	49	186	388	1283	97	844	799	5.04	4	4	36	95

In [44]:
```
df.shape
```

Out[44]: (28, 17)

In [45]:
```
df.reset_index()
```

Out[45]:

	index	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	0	95	724	5575	1497	300	42	139	383	973	104	641	601	3.73	2	8	56	88
1	1	83	696	5467	1349	277	44	156	439	1264	70	700	653	4.07	2	12	45	86
2	2	81	669	5439	1395	303	29	141	533	1157	86	640	584	3.67	11	10	38	79
3	3	76	622	5533	1381	260	27	136	404	1231	68	701	643	3.98	7	9	37	101
4	4	74	689	5605	1515	289	49	151	455	1259	83	803	746	4.64	7	12	35	86
5	5	93	891	5509	1480	308	17	232	570	1151	88	670	609	3.80	7	10	34	88
6	6	87	764	5567	1397	272	19	212	554	1227	63	698	652	4.03	3	4	48	93
7	7	81	713	5485	1370	246	20	217	418	1331	44	693	646	4.05	0	10	43	77
8	8	80	644	5485	1383	278	32	167	436	1310	87	642	604	3.74	1	12	60	95
9	9	78	748	5640	1495	294	33	161	478	1148	71	753	694	4.31	3	10	40	97
10	10	88	751	5511	1419	279	32	172	503	1233	101	733	680	4.24	5	9	45	119
11	11	86	729	5459	1363	278	26	230	486	1392	121	618	572	3.57	5	13	39	85
12	12	85	661	5417	1331	243	21	176	435	1150	52	675	630	3.94	2	12	46	93
13	13	76	656	5544	1379	262	22	198	478	1336	69	726	677	4.16	6	12	45	94
14	14	68	694	5600	1405	277	46	146	475	1119	78	729	664	4.14	5	15	28	126
15	15	100	647	5484	1386	288	39	137	506	1267	69	525	478	2.94	1	15	62	96
16	16	98	697	5631	1462	292	27	140	461	1322	98	596	532	3.21	0	13	54	122
17	18	68	655	5480	1378	274	34	145	412	1299	84	737	682	4.28	1	7	40	116
18	19	64	640	5571	1382	257	27	167	496	1255	134	754	700	4.33	2	8	35	90
19	20	90	683	5527	1351	295	17	177	488	1290	51	613	557	3.43	1	14	50	88
20	21	83	703	5428	1363	265	13	177	539	1344	57	635	577	3.62	4	13	41	90
21	22	71	613	5463	1420	236	40	120	375	1150	112	678	638	4.02	0	12	35	77
22	23	67	573	5420	1361	251	18	100	471	1107	69	760	698	4.41	3	10	44	90
23	24	63	626	5529	1374	272	37	130	387	1274	88	809	749	4.69	1	7	35	117
24	26	84	696	5565	1486	288	39	136	457	1159	93	627	597	3.72	7	18	41	78
25	27	79	720	5649	1494	289	48	154	490	1312	132	713	659	4.04	1	12	44	86
26	28	74	650	5457	1324	260	36	148	426	1327	82	731	655	4.09	1	6	41	92
27	29	68	737	5572	1479	274	49	186	388	1283	97	844	799	5.04	4	4	36	95

In [46]:

In [ ]:
```
## same process for other two columns
```

In [47]:
```
sv_=(q3.SV+(1.5*iqr.SV))
sv_
```

Out[47]:
```
61.0
```

In [48]:
```
index=np.where(df['SV']>sv_)
print(index)
```

```
(array([15], dtype=int64),)
```

In [49]:
```
df=df.drop(df.index[index])
df
```

Out[49]:

	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	95	724	5575	1497	300	42	139	383	973	104	641	601	3.73	2	8	56	88
1	83	696	5467	1349	277	44	156	439	1264	70	700	653	4.07	2	12	45	86
2	81	669	5439	1395	303	29	141	533	1157	86	640	584	3.67	11	10	38	79
3	76	622	5533	1381	260	27	136	404	1231	68	701	643	3.98	7	9	37	101
4	74	689	5605	1515	289	49	151	455	1259	83	803	746	4.64	7	12	35	86
5	93	891	5509	1480	308	17	232	570	1151	88	670	609	3.80	7	10	34	88
6	87	764	5567	1397	272	19	212	554	1227	63	698	652	4.03	3	4	48	93
7	81	713	5485	1370	246	20	217	418	1331	44	693	646	4.05	0	10	43	77
8	80	644	5485	1383	278	32	167	436	1310	87	642	604	3.74	1	12	60	95
9	78	748	5640	1495	294	33	161	478	1148	71	753	694	4.31	3	10	40	97
10	88	751	5511	1419	279	32	172	503	1233	101	733	680	4.24	5	9	45	119
11	86	729	5459	1363	278	26	230	486	1392	121	618	572	3.57	5	13	39	85
12	85	661	5417	1331	243	21	176	435	1150	52	675	630	3.94	2	12	46	93
13	76	656	5544	1379	262	22	198	478	1336	69	726	677	4.16	6	12	45	94
14	68	694	5600	1405	277	46	146	475	1119	78	729	664	4.14	5	15	28	126
16	98	697	5631	1462	292	27	140	461	1322	98	596	532	3.21	0	13	54	122
18	68	655	5480	1378	274	34	145	412	1299	84	737	682	4.28	1	7	40	116
19	64	640	5571	1382	257	27	167	496	1255	134	754	700	4.33	2	8	35	90
20	90	683	5527	1351	295	17	177	488	1290	51	613	557	3.43	1	14	50	88
21	83	703	5428	1363	265	13	177	539	1344	57	635	577	3.62	4	13	41	90
22	71	613	5463	1420	236	40	120	375	1150	112	678	638	4.02	0	12	35	77
23	67	573	5420	1361	251	18	100	471	1107	69	760	698	4.41	3	10	44	90
24	63	626	5529	1374	272	37	130	387	1274	88	809	749	4.69	1	7	35	117
26	84	696	5565	1486	288	39	136	457	1159	93	627	597	3.72	7	18	41	78
27	79	720	5649	1494	289	48	154	490	1312	132	713	659	4.04	1	12	44	86
28	74	650	5457	1324	260	36	148	426	1327	82	731	655	4.09	1	6	41	92
29	68	737	5572	1479	274	49	186	388	1283	97	844	799	5.04	4	4	36	95

In [50]:
```
df.reset_index()
```

Out[50]:

	index	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	0	95	724	5575	1497	300	42	139	383	973	104	641	601	3.73	2	8	56	88
1	1	83	696	5467	1349	277	44	156	439	1264	70	700	653	4.07	2	12	45	86
2	2	81	669	5439	1395	303	29	141	533	1157	86	640	584	3.67	11	10	38	79
3	3	76	622	5533	1381	260	27	136	404	1231	68	701	643	3.98	7	9	37	101
4	4	74	689	5605	1515	289	49	151	455	1259	83	803	746	4.64	7	12	35	86
5	5	93	891	5509	1480	308	17	232	570	1151	88	670	609	3.80	7	10	34	88
6	6	87	764	5567	1397	272	19	212	554	1227	63	698	652	4.03	3	4	48	93

7	7	81	713	5485	1370	246	20	217	418	1331	44	693	646	4.05	0	10	43	77
8	8	80	644	5485	1383	278	32	167	436	1310	87	642	604	3.74	1	12	60	95
9	9	78	748	5640	1495	294	33	161	478	1148	71	753	694	4.31	3	10	40	97
10	10	88	751	5511	1419	279	32	172	503	1233	101	733	680	4.24	5	9	45	119
11	11	86	729	5459	1363	278	26	230	486	1392	121	618	572	3.57	5	13	39	85
12	12	85	661	5417	1331	243	21	176	435	1150	52	675	630	3.94	2	12	46	93
13	13	76	656	5544	1379	262	22	198	478	1336	69	726	677	4.16	6	12	45	94
14	14	68	694	5600	1405	277	46	146	475	1119	78	729	664	4.14	5	15	28	126
15	16	98	697	5631	1462	292	27	140	461	1322	98	596	532	3.21	0	13	54	122
16	18	68	655	5480	1378	274	34	145	412	1299	84	737	682	4.28	1	7	40	116
17	19	64	640	5571	1382	257	27	167	496	1255	134	754	700	4.33	2	8	35	90
18	20	90	683	5527	1351	295	17	177	488	1290	51	613	557	3.43	1	14	50	88
19	21	83	703	5428	1363	265	13	177	539	1344	57	635	577	3.62	4	13	41	90
20	22	71	613	5463	1420	236	40	120	375	1150	112	678	638	4.02	0	12	35	77
21	23	67	573	5420	1361	251	18	100	471	1107	69	760	698	4.41	3	10	44	90
22	24	63	626	5529	1374	272	37	130	387	1274	88	809	749	4.69	1	7	35	117
23	26	84	696	5565	1486	288	39	136	457	1159	93	627	597	3.72	7	18	41	78
24	27	79	720	5649	1494	289	48	154	490	1312	132	713	659	4.04	1	12	44	86
25	28	74	650	5457	1324	260	36	148	426	1327	82	731	655	4.09	1	6	41	92
26	29	68	737	5572	1479	274	49	186	388	1283	97	844	799	5.04	4	4	36	95

In [51]:
```
df.shape
```

Out[51]: (27, 17)

In [52]:
```
# for E column
```

In [53]:
```
e_=(q3.E+(1.5*iqr.E))
e_
```

Out[53]: 112.875

In [54]:
```
index=np.where(df['E']>e_)
print(index)
```

```
(array([10, 14, 15, 16, 22], dtype=int64),)
```

In [55]:
```
df=df.drop(df.index[index])
df
```

Out[55]:

	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	95	724	5575	1497	300	42	139	383	973	104	641	601	3.73	2	8	56	88
1	83	696	5467	1349	277	44	156	439	1264	70	700	653	4.07	2	12	45	86
2	81	669	5439	1395	303	29	141	533	1157	86	640	584	3.67	11	10	38	79
3	76	622	5533	1381	260	27	136	404	1231	68	701	643	3.98	7	9	37	101
4	74	689	5605	1515	289	49	151	455	1259	83	803	746	4.64	7	12	35	86
5	93	891	5509	1480	308	17	232	570	1151	88	670	609	3.80	7	10	34	88
6	87	764	5567	1397	272	19	212	554	1227	63	698	652	4.03	3	4	48	93
7	81	713	5485	1370	246	20	217	418	1331	44	693	646	4.05	0	10	43	77
8	80	644	5485	1383	278	32	167	436	1310	87	642	604	3.74	1	12	60	95
9	78	748	5640	1495	294	33	161	478	1148	71	753	694	4.31	3	10	40	97
11	86	729	5459	1363	278	26	230	486	1392	121	618	572	3.57	5	13	39	85
12	85	661	5417	1331	243	21	176	435	1150	52	675	630	3.94	2	12	46	93

13	76	656	5544	1379	262	22	198	478	1336	69	726	677	4.16	6	12	45	94
19	64	640	5571	1382	257	27	167	496	1255	134	754	700	4.33	2	8	35	90
20	90	683	5527	1351	295	17	177	488	1290	51	613	557	3.43	1	14	50	88
21	83	703	5428	1363	265	13	177	539	1344	57	635	577	3.62	4	13	41	90
22	71	613	5463	1420	236	40	120	375	1150	112	678	638	4.02	0	12	35	77
23	67	573	5420	1361	251	18	100	471	1107	69	760	698	4.41	3	10	44	90
26	84	696	5565	1486	288	39	136	457	1159	93	627	597	3.72	7	18	41	78
27	79	720	5649	1494	289	48	154	490	1312	132	713	659	4.04	1	12	44	86
28	74	650	5457	1324	260	36	148	426	1327	82	731	655	4.09	1	6	41	92
29	68	737	5572	1479	274	49	186	388	1283	97	844	799	5.04	4	4	36	95

In [56]:
```
df.reset_index()
```

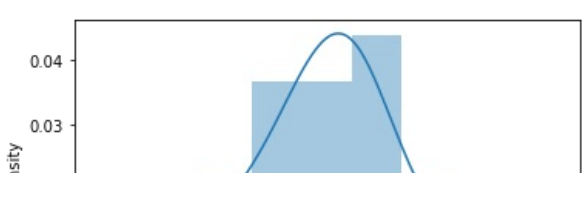
Out[56]:

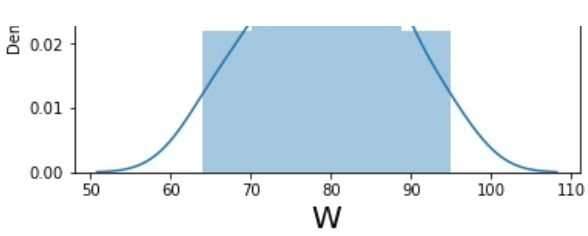
	index	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	0	95	724	5575	1497	300	42	139	383	973	104	641	601	3.73	2	8	56	88
1	1	83	696	5467	1349	277	44	156	439	1264	70	700	653	4.07	2	12	45	86
2	2	81	669	5439	1395	303	29	141	533	1157	86	640	584	3.67	11	10	38	79
3	3	76	622	5533	1381	260	27	136	404	1231	68	701	643	3.98	7	9	37	101
4	4	74	689	5605	1515	289	49	151	455	1259	83	803	746	4.64	7	12	35	86
5	5	93	891	5509	1480	308	17	232	570	1151	88	670	609	3.80	7	10	34	88
6	6	87	764	5567	1397	272	19	212	554	1227	63	698	652	4.03	3	4	48	93
7	7	81	713	5485	1370	246	20	217	418	1331	44	693	646	4.05	0	10	43	77
8	8	80	644	5485	1383	278	32	167	436	1310	87	642	604	3.74	1	12	60	95
9	9	78	748	5640	1495	294	33	161	478	1148	71	753	694	4.31	3	10	40	97
10	11	86	729	5459	1363	278	26	230	486	1392	121	618	572	3.57	5	13	39	85
11	12	85	661	5417	1331	243	21	176	435	1150	52	675	630	3.94	2	12	46	93
12	13	76	656	5544	1379	262	22	198	478	1336	69	726	677	4.16	6	12	45	94
13	19	64	640	5571	1382	257	27	167	496	1255	134	754	700	4.33	2	8	35	90
14	20	90	683	5527	1351	295	17	177	488	1290	51	613	557	3.43	1	14	50	88
15	21	83	703	5428	1363	265	13	177	539	1344	57	635	577	3.62	4	13	41	90
16	22	71	613	5463	1420	236	40	120	375	1150	112	678	638	4.02	0	12	35	77
17	23	67	573	5420	1361	251	18	100	471	1107	69	760	698	4.41	3	10	44	90
18	26	84	696	5565	1486	288	39	136	457	1159	93	627	597	3.72	7	18	41	78
19	27	79	720	5649	1494	289	48	154	490	1312	132	713	659	4.04	1	12	44	86
20	28	74	650	5457	1324	260	36	148	426	1327	82	731	655	4.09	1	6	41	92
21	29	68	737	5572	1479	274	49	186	388	1283	97	844	799	5.04	4	4	36	95

In [57]:
```
## once again plotting the distribution
```

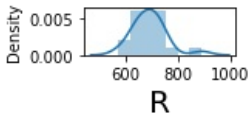
In [71]:
```
plt.figure(figsize=(20,25))
plotnumber=1
for column in df:
    if plotnumber<=17:
        ax=plt.subplot(6,3,plotnumber)
        sns.distplot(df[column])
        plt.xlabel(column,fontsize=20)
        plotnumber+=1
        plt.show()
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

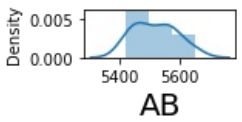




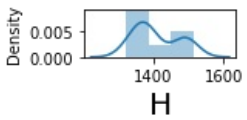

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



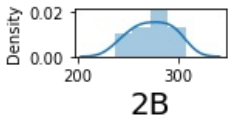

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



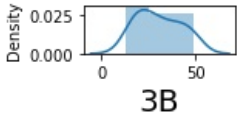

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



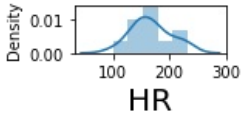

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



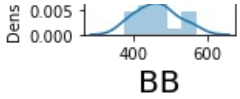

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



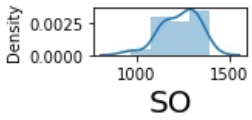

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



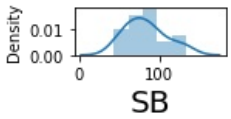

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



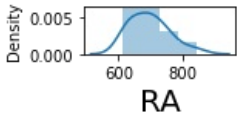

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



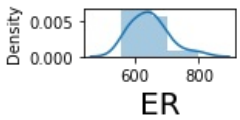

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



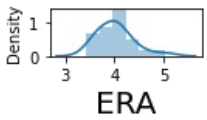

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



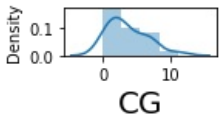

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



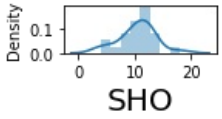

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

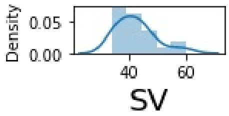



```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

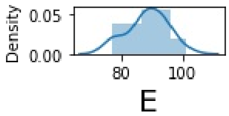



```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
```

```
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [72]:

```
## removing skewness
```

In [74]:

```
cont=['W','R','AB','H','2B','3B','HR','BB','SO','SB','RA','ER','ERA','CG','SHO','SV','E']
```

In [75]:

```
from sklearn.preprocessing import PowerTransformer
pt=PowerTransformer()
```

In [76]:

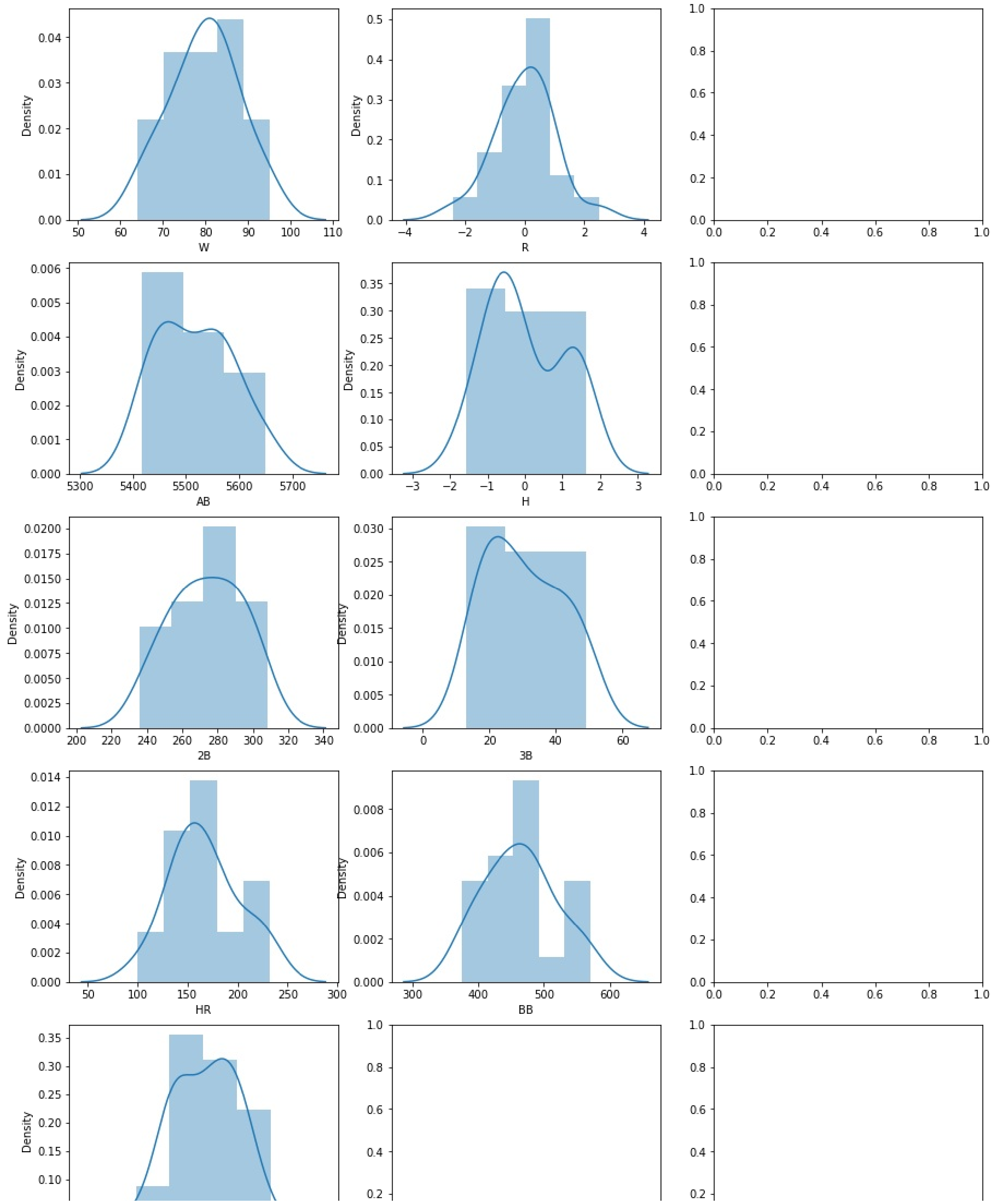
```
for i in cont:
    if np.abs(df[i].skew())>0.5:
        df[i]=pt.fit_transform(df[i].values.reshape(-1,1))
```

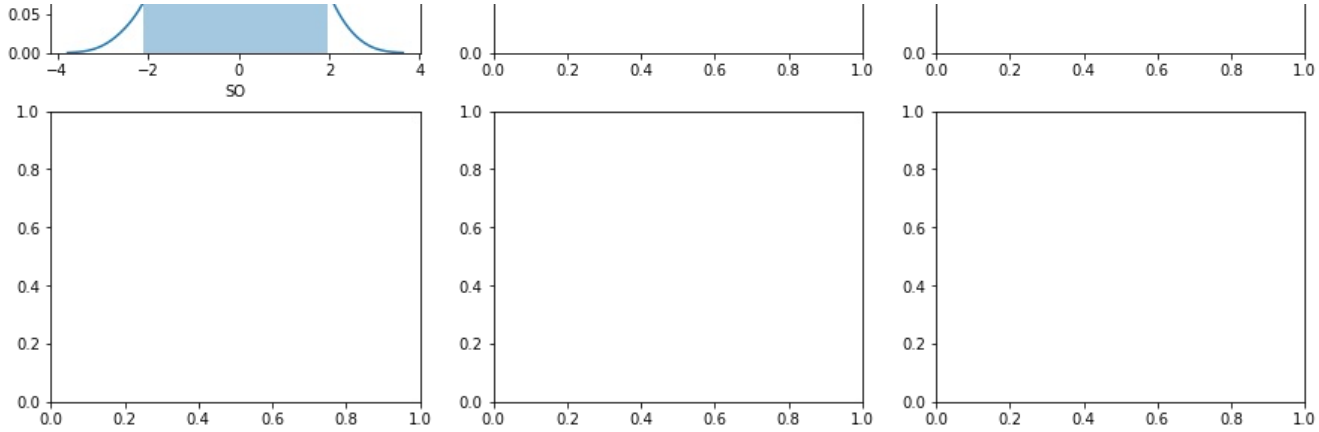
In [80]:

```
fig,ax=plt.subplots(6,3,figsize=(15,25))
r=0
c=0
for i,n in enumerate(cont):
    if r==4 and c==1:
        break
    if i%2==0 and i>0:
        r+=1
        c=0
    sns.distplot(df[n],ax=ax[r,c])
    c+=1
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
```

```
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```





In [81]:
```
df.skew()
```

Out[81]:
```
W      -0.095453
R      -0.039208
AB      0.261917
H       0.322280
2B     -0.120666
3B      0.282484
HR      0.296858
BB      0.231521
SO     -0.097476
SB     -0.004785
RA      0.070272
ER      0.039362
ERA     0.026821
CG     -0.040834
SHO    -0.245153
SV      0.081804
E      -0.327280
dtype: float64
```

In [90]:
```
df.head()
```

Out[90]:

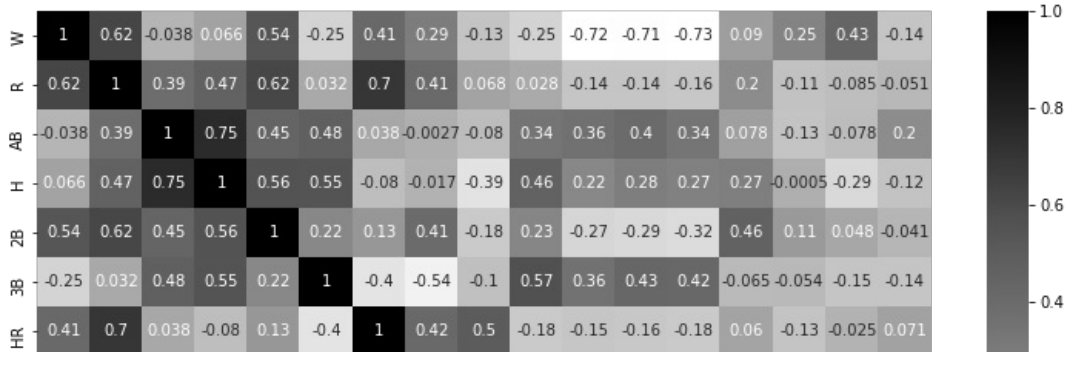
	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	95	0.618565	5575	1.406639	300	42	139	383	-2.115970	0.886834	-0.959449	-0.755001	-0.793546	-0.403509	8	1.709245	88
1	83	0.186956	5467	-1.026155	277	44	156	439	0.199573	-0.442694	0.220514	0.291679	0.303841	-0.403509	12	0.605666	86
2	81	-0.275080	5439	-0.126828	303	29	141	533	-0.888802	0.243811	-0.982936	-1.168576	-1.024572	1.939036	10	-0.621266	79
3	76	-1.208515	5533	-0.380484	260	27	136	404	-0.170878	-0.538520	0.237483	0.112566	0.044582	1.180075	9	-0.852921	101
4	74	0.071758	5605	1.625706	289	49	151	455	0.141317	0.124652	1.592659	1.588634	1.562916	1.180075	12	-1.374471	86

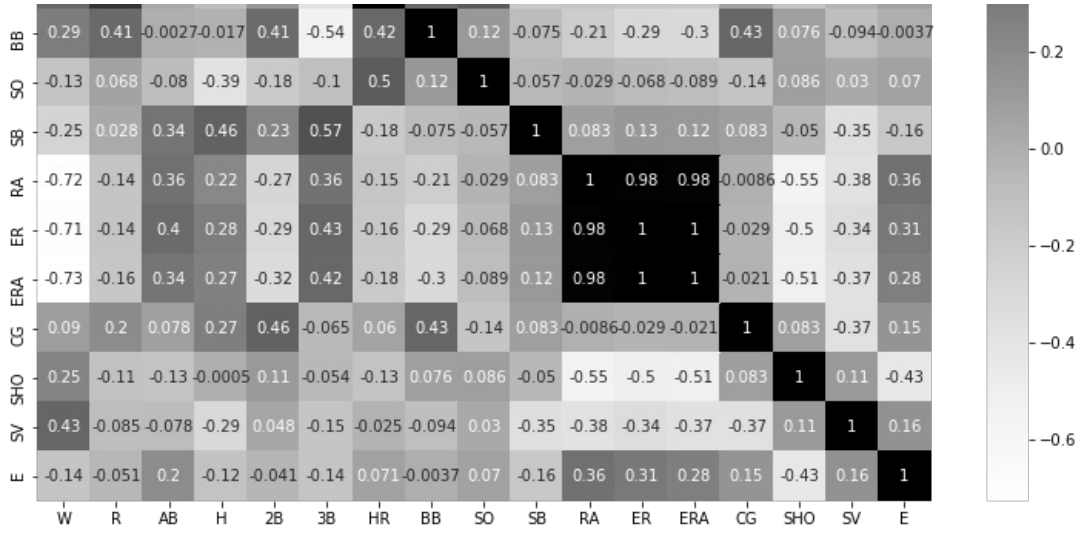
In [82]:
```
##skewness has been removed
```

In [83]:
```
## checking for correlation
```

In [84]:
```
plt.figure(figsize=(13,10))
sns.heatmap(df.corr(),annot=True,cmap='Greys')
```

Out[84]:
```
<AxesSubplot:>
```





In [85]:
```
## lets work on the feature selection
```

In [86]:
```
## from the above RA,ERA,ER are highly correlated..we remove ERA,ER
```

In [121…]:
```
df_=df.drop(columns=['ERA','ER','R','AB'])
df_
```

Out[121…]

	W	H	2B	3B	HR	BB	SO	SB	RA	CG	SHO	SV	E
0	95	1.406639	300	42	139	383	-2.115970	0.886834	-0.959449	-0.403509	8	1.709245	88
1	83	-1.026155	277	44	156	439	0.199573	-0.442694	0.220514	-0.403509	12	0.605666	86
2	81	-0.126828	303	29	141	533	-0.888802	0.243811	-0.982936	1.939036	10	-0.621266	79
3	76	-0.380484	260	27	136	404	-0.170878	-0.538520	0.237483	1.180075	9	-0.852921	101
4	74	1.625706	289	49	151	455	0.141317	0.124652	1.592659	1.180075	12	-1.374471	86
5	93	1.187572	308	17	232	570	-0.940664	0.321129	-0.332367	1.180075	10	-1.668827	88
6	87	-0.080709	272	19	212	554	-0.213578	-0.789983	0.186312	0.026797	4	0.980072	93
7	81	-0.588021	246	20	217	418	1.058170	-1.951877	0.099230	-1.821797	10	0.313154	77
8	80	-0.345895	278	32	167	436	0.772954	0.282676	-0.936095	-0.966423	12	1.972548	95
9	78	1.383579	294	33	161	478	-0.966257	-0.395726	1.011339	0.026797	10	-0.207021	97
11	86	-0.726379	278	26	230	486	1.976620	1.405376	-1.535247	0.680425	13	-0.406502	85
12	85	-1.418169	243	21	176	435	-0.949220	-1.415269	-0.233977	-0.403509	12	0.738331	93
13	76	-0.426604	262	22	198	478	1.128340	-0.490286	0.634646	0.944300	12	0.605666	94
19	64	-0.357425	257	27	167	496	0.095268	1.758009	1.024360	-0.403509	8	-1.374471	90
20	90	-0.980035	295	17	177	488	0.515203	-1.478017	-1.670947	-0.966423	14	1.194355	88
21	83	-0.726379	265	13	177	539	1.242459	-1.117283	-1.102394	0.379436	13	-0.021401	90
22	71	0.311305	236	40	120	375	-0.949220	1.139961	-0.176202	-1.821797	12	-1.374471	77
23	67	-0.772498	251	18	100	471	-1.294232	-0.490286	1.101186	0.026797	10	0.464206	90
26	84	1.268281	288	39	136	457	-0.871314	0.507515	-1.300786	1.180075	18	-0.021401	78
27	79	1.360520	289	48	154	490	0.799466	1.705888	0.434432	-0.966423	12	0.464206	86
28	74	-1.579586	260	36	148	426	1.002666	0.084044	0.708238	-0.966423	6	-0.021401	92
29	68	1.176042	274	49	186	388	0.428099	0.650046	1.980002	0.379436	4	-1.103295	95

In [122…]:
```
x=df_.drop(['W'],axis=1)
x
```

Out[122…]

	H	2B	3B	HR	BB	SO	SB	RA	CG	SHO	SV	E
0	1.406639	300	42	139	383	-2.115970	0.886834	-0.959449	-0.403509	8	1.709245	88
1	-1.026155	277	44	156	439	0.199573	-0.442694	0.220514	-0.403509	12	0.605666	86
2	-0.126828	303	29	141	533	-0.888802	0.243811	-0.982936	1.939036	10	-0.621266	79
3	-0.380484	260	27	136	404	-0.170878	-0.538520	0.237483	1.180075	9	-0.852921	101

4	1.625706	289	49	151	455	0.141317	0.124652	1.592659	1.180075	12	-1.374471	86
5	1.187572	308	17	232	570	-0.940664	0.321129	-0.332367	1.180075	10	-1.668827	88
6	-0.080709	272	19	212	554	-0.213578	-0.789983	0.186312	0.026797	4	0.980072	93
7	-0.588021	246	20	217	418	1.058170	-1.951877	0.099230	-1.821797	10	0.313154	77
8	-0.345895	278	32	167	436	0.772954	0.282676	-0.936095	-0.966423	12	1.972548	95
9	1.383579	294	33	161	478	-0.966257	-0.395726	1.011339	0.026797	10	-0.207021	97
11	-0.726379	278	26	230	486	1.976620	1.405376	-1.535247	0.680425	13	-0.406502	85
12	-1.418169	243	21	176	435	-0.949220	-1.415269	-0.233977	-0.403509	12	0.738331	93
13	-0.426604	262	22	198	478	1.128340	-0.490286	0.634646	0.944300	12	0.605666	94
19	-0.357425	257	27	167	496	0.095268	1.758009	1.024360	-0.403509	8	-1.374471	90
20	-0.980035	295	17	177	488	0.515203	-1.478017	-1.670947	-0.966423	14	1.194355	88
21	-0.726379	265	13	177	539	1.242459	-1.117283	-1.102394	0.379436	13	-0.021401	90
22	0.311305	236	40	120	375	-0.949220	1.139961	-0.176202	-1.821797	12	-1.374471	77
23	-0.772498	251	18	100	471	-1.294232	-0.490286	1.101186	0.026797	10	0.464206	90
26	1.268281	288	39	136	457	-0.871314	0.507515	-1.300786	1.180075	18	-0.021401	78
27	1.360520	289	48	154	490	0.799466	1.705888	0.434432	-0.966423	12	0.464206	86
28	-1.579586	260	36	148	426	1.002666	0.084044	0.708238	-0.966423	6	-0.021401	92
29	1.176042	274	49	186	388	0.428099	0.650046	1.980002	0.379436	4	-1.103295	95

In [123…
```
y=df_.W
y
```

Out[123…
```
0     95
1     83
2     81
3     76
4     74
5     93
6     87
7     81
8     80
9     78
11    86
12    85
13    76
19    64
20    90
21    83
22    71
23    67
26    84
27    79
28    74
29    68
Name: W, dtype: int64
```

# #

In [124…
```
x.shape
```

Out[124… (22, 12)

In [125…
```
y.shape
```

Out[125… (22,)

In [142…
```
from sklearn.preprocessing import StandardScaler
```

In [143…
```
st=StandardScaler()
```

```
st=StandardScaler()
st.fit_transform(x)
```

Out[143…

```
array([[ 1.39830288,  1.30470044,  1.03421049, -0.83435263, -1.5093976 ,
        -2.11596983,  0.88683374, -0.95944913, -0.40350895, -0.79509994,
         1.70924532, -0.08530614],
       [-1.03457642,  0.15656405,  1.21196541, -0.3334732 , -0.46056633,
         0.19957267, -0.44269353,  0.22051426, -0.40350895,  0.47705996,
         0.60566562, -0.39809532],
       [-0.13521819,  1.45445736, -0.12119654, -0.77542564,  1.29997188,
         -0.8888024 ,  0.24381121, -0.98293611,  1.93903572, -0.15901999,
         -0.62126567, -1.49285746],
       [-0.38888333, -0.69205849, -0.29895147, -0.92274312, -1.11608587,
         -0.17087814, -0.5385199 ,  0.23748296,  1.18007456, -0.47705996,
         -0.85292105,  1.94782355],
       [ 1.61737732,  0.75559173,  1.65635273, -0.48079068, -0.16090025,
         0.14131708,  0.12465163,  1.59265942,  1.18007456,  0.47705996,
         -1.37447073, -0.39809532],
       [ 1.17922844,  1.70405222, -1.18772611,  1.90575249,  1.99294968,
         -0.94066383,  0.32112851, -0.33236744,  1.18007456, -0.15901999,
         -1.66882725, -0.08530614],
       [-0.08909726, -0.09303081, -1.00997118,  1.31648257,  1.6932836 ,
         -0.21357786, -0.78998287,  0.18631153,  0.02679659, -2.06725983,
         0.98007242,  0.69666682],
       [-0.59642754, -1.39092412, -0.92109371,  1.46380005, -0.85387806,
         1.05816979, -1.951877  ,  0.09922951, -1.82179714, -0.15901999,
         0.31315382, -1.80564665],
       [-0.35429263,  0.20648303,  0.14543585, -0.00937475, -0.51675372,
         0.77295404,  0.28267584, -0.93609481, -0.96642268,  0.47705996,
         1.97254764,  1.009456  ],
       [ 1.37524241,  1.0051866 ,  0.23431331, -0.18615572,  0.26986973,
         -0.96625696, -0.39572578,  1.0113393 ,  0.02679659, -0.15901999,
         -0.20702116,  1.32224518],
       [-0.73479034,  0.20648303, -0.38782893,  1.8468255 ,  0.41970277,
         1.97661995,  1.40537554, -1.53524678,  0.68042466,  0.79509994,
         -0.40650209, -0.55448992],
       [-1.42660436, -1.54068104, -0.83221625,  0.25579671, -0.53548285,
         -0.94921976, -1.41526875, -0.23397746, -0.40350895,  0.47705996,
         0.73833053,  0.69666682],
       [-0.43500427, -0.59222055, -0.74333879,  0.90399363,  0.26986973,
         1.12834026, -0.49028577,  0.63464584,  0.94429987,  0.47705996,
         0.60566562,  0.85306141],
       [-0.36582287, -0.84181541, -0.29895147, -0.00937475,  0.60699407,
         0.09526771,  1.75800925,  1.02436034, -0.40350895, -0.79509994,
         -1.37447073,  0.22748304],
       [-0.98845548,  1.05510557, -1.18772611,  0.28526021,  0.45716103,
         0.5152029 , -1.47801707, -1.67094733, -0.96642268,  1.11313991,
         1.19435543, -0.08530614],
       [-0.73479034, -0.44246363, -1.54323596,  0.28526021,  1.41234666,
         1.24245916, -1.11728278, -1.10239444,  0.37943628,  0.79509994,
         -0.0214013 ,  0.22748304],
       [ 0.30293068, -1.89011385,  0.85645556, -1.39415906, -1.65923064,
         -0.94921976,  1.13996069, -0.17620224, -1.82179714,  0.47705996,
         -1.37447073, -1.80564665],
       [-0.78091128, -1.14132925, -1.09884864, -1.98342898,  0.13876583,
         -1.29423159, -0.49028577,  1.10118632,  0.02679659, -0.15901999,
         0.46420618,  0.22748304],
       [ 1.25994007,  0.70567276,  0.7675781 , -0.92274312, -0.12344199,
         -0.8713137 ,  0.50751476, -1.30078591,  1.18007456,  2.38529981,
         -0.0214013 , -1.64925206],
       [ 1.35218194,  0.75559173,  1.56747527, -0.3924002 ,  0.49461929,
         0.79946562,  1.70588832,  0.4344321 , -0.96642268,  0.47705996,
         0.46420618, -0.39809532],
       [-1.58802763, -0.69205849,  0.5009457 , -0.56918117, -0.70404502,
         1.00266611,  0.08404362,  0.70823846, -0.96642268, -1.43117988,
         -0.0214013 ,  0.54027223],
       [ 1.16769821,  0.00680713,  1.65635273,  0.55043167, -1.41575195,
         0.42809854,  0.65004611,  1.98000161,  0.37943628, -2.06725983,
         -1.10329545,  1.009456  ]])
```

In [144…

```
## MODELLING PHASE
```

In [145…

```
from sklearn.model_selection import train_test_split,cross_val_score
```

In [146…

```
#importing models
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import LinearRegression,Lasso,Ridge,ElasticNet
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
```

```
from sklearn.ensemble import RandomForestRegressor,AdaBoostRegressor,GradientBoostingRegressor
```

In [147…

```
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
```

In [148…

```
#Choosing the best random state using Logistic regression
def randomstate(a,b):
    maxx=1000
    for state in range(1,201):
        xtrain,xtest,ytrain,ytest=train_test_split(a,b,test_size=0.25,random_state=state)
        model=LinearRegression()
        model.fit(xtrain,ytrain)
        p=model.predict(xtest)
        mse=mean_squared_error(p,ytest)
        if maxx>mse:
            maxx=mse
            j=state
    return j
```

In [149…

```
#Creating list of models and another list mapped to their names
models=[KNeighborsRegressor(),SVR(),LinearRegression(),Lasso(),Ridge(),ElasticNet(),DecisionTreeRegressor(),
       RandomForestRegressor(),AdaBoostRegressor(),GradientBoostingRegressor()]

names=['KNeighborsRegressor','SVR','LinearRegression','Lasso','Ridge','ElasticNet','DecisionTreeRegressor',
       'RandomForestRegressor','AdaBoostRegressor','GradientBoostingRegressor']
```

In [150…

```
def createmodels(model_list,independent,dependent,n):
    xtrain,xtest,ytrain,ytest=train_test_split(independent,dependent,test_size=0.25,random_state=randomstate(inde
    name=[]
    meanabs=[]
    meansqd=[]
    rootmeansqd=[]
    r2=[]
    mcv=[]

    #Creating models
    for i,model in enumerate(model_list):
        model.fit(xtrain,ytrain)
        p=model.predict(xtest)
        score=cross_val_score(model,independent,dependent,cv=10)

        #Calculating scores of the model and appending them to a list
        name.append(n[i])
        meanabs.append(np.round(mean_absolute_error(p,ytest),4))
        meansqd.append(np.round(mean_squared_error(p,ytest),4))
        rootmeansqd.append(np.round(np.sqrt(mean_squared_error(p,ytest)),4))
        r2.append(np.round(r2_score(p,ytest),2))
        mcv.append(np.round(np.mean(score),4))

    #Creating Dataframe
    data=pd.DataFrame()
    data['Model']=name
    data['Mean Absolute Error']=meanabs
    data['Mean Squared Error']=meansqd
    data['Root Mean Squared Error']=rootmeansqd
    data['R2 Score']=r2
    data['Mean of Cross validaton Score']=mcv
    data.set_index('Model',inplace = True)
    return data
```

In [152…

```
createmodels(models,x,y,names)
```

Out[152…

Model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R2 Score	Mean of Cross validaton Score
KNeighborsRegressor	9.9333	108.0400	10.3942	-27.46	-22.4991
SVR	9.7912	118.7351	10.8966	-4919.73	-20.1299
LinearRegression	2.3628	10.5109	3.2421	0.86	-7.6458
Lasso	6.6537	51.6713	7.1883	-0.14	-25.3361
Ridge	4.0923	19.0181	4.3610	0.70	-6.9103
ElasticNet	6.1681	45.0510	6.7120	0.16	-24.7603
DecisionTreeRegressor	8.3333	94.0000	9.6954	-3.21	-44.3926
RandomForestRegressor	7.7467	80.7988	8.9888	-10.57	-12.9599
AdaBoostRegressor	8.1345	85.8991	9.2682	-14.02	-13.5566
GradientBoostingRegressor	7.6845	82.5595	9.0862	-5.64	-17.8425

In [153…
```
## looks like linear regression and ridge regression works better
```

In [154…
```
## hyper parameter tuning for ridge regression
```

In [155…
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.25,random_state=randomstate(x,y))
```

In [156…
```
from sklearn.model_selection import GridSearchCV
```

In [157…
```
params={'alpha':[200,230,250,270,275,290,300,400,500]}
```

In [159…
```
g=GridSearchCV(Ridge(),params,cv=10)
```

In [160…
```
g.fit(xtrain,ytrain)
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\metrics\_regression.py:682: UndefinedMetricWarning: R^2
score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\sklearn\model_selection\_search.py:918: UserWarning: One or mor
e of the test scores are non-finite: [nan nan nan nan nan nan nan nan nan]
  warnings.warn(
```

Out[160…

```
GridSearchCV(cv=10, estimator=Ridge(),
             param_grid={'alpha': [200, 230, 250, 270, 275, 290, 300, 400,
                                   500]})
```

In [163…

```
print(g.best_estimator_)
print(g.best_score_)
print(g.best_params_)
```

```
Ridge(alpha=200)
nan
{'alpha': 200}
```

In [165…

```
m=Ridge(alpha=200)
m.fit(xtrain,ytrain)
p=m.predict(xtest)
score=cross_val_score(m,x,y,cv=10)
```

In [166…

```
print('Mean Absolute Error is',np.round(mean_absolute_error(p,ytest),4))
print('Mean Squared Error is',np.round(mean_squared_error(p,ytest),4))
print('Root Mean Squared Error is',np.round(np.sqrt(mean_squared_error(p,ytest)),4))
print('R2 Score is',np.round(r2_score(p,ytest),4)*100)
print('Mean of cross validaton Score is',np.round(np.mean(score)*100,4))
```

```
Mean Absolute Error is 5.4003
```

```
Mean Squared Error is 47.9846
Root Mean Squared Error is 6.9271
R2 Score is -4.45
Mean of cross validaton Score is -2855.3343
```

In [167…
```
## finale model is linear regression
```

In [168…
```
model=LinearRegression()
model.fit(xtrain,ytrain)
p=model.predict(xtest)
score=cross_val_score(m,x,y,cv=10)
```

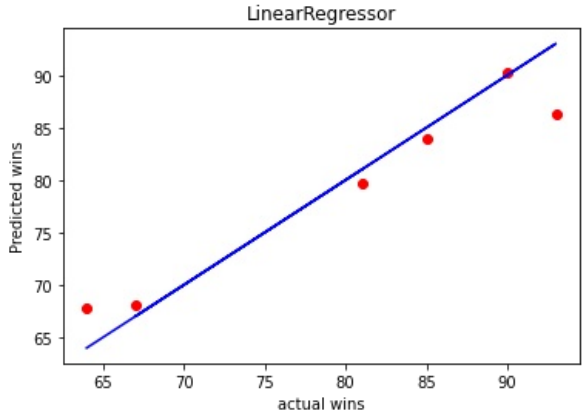
In [169…
```
## evoluation metrics
```

In [170…
```
print('Mean Absolute Error is',np.round(mean_absolute_error(p,ytest),4))
print('Mean Squared Error is',np.round(mean_squared_error(p,ytest),4))
print('Root Mean Squared Error is',np.round(np.sqrt(mean_squared_error(p,ytest)),4))
print('R2 Score is',np.round(r2_score(p,ytest),4)*100)
print('Mean of cross validaton Score is',np.round(np.mean(score)*100,4))
```

```
Mean Absolute Error is 2.3628
Mean Squared Error is 10.5109
Root Mean Squared Error is 3.2421
R2 Score is 85.98
Mean of cross validaton Score is -2855.3343
```

In [171…
```
plt.scatter(x=ytest,y=p,color='r')
plt.plot(ytest,ytest,color='b')
plt.xlabel('actual wins')
plt.ylabel('Predicted wins')
plt.title('LinearRegressor')
```

Out[171… `Text(0.5, 1.0, 'LinearRegressor')`



In [174…
```
import numpy as np
a=np.array(ytest)
pred_=np.array(model.predict(xtest))
df_con=pd.DataFrame({'true':a,'predicted':pred_},index=range(len(a)))
df_con.head(10)
```

Out[174…

	true	predicted
0	64	67.779681
1	90	90.241556
2	93	86.320091
3	85	83.961656
4	67	68.057768
5	81	79.620487

In [175…

```
#model saving
```

In [176…

```
import pickle
filename='baseball.pkl'
pickle.dump(model,open(filename,'wb'))
```

In [ ]: