

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.preprocessing import StandardScaler
import sklearn.metrics as metrics
import statsmodels.api as sm
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
```

```
In [2]: file =pd.read_csv("baseball.csv")

file
```

	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
17	97	689	5491	1341	272	30	171	567	1518	95	608	546	3.36	6	21	48	111
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
25	92	667	5385	1346	263	26	187	563	1258	59	595	553	3.44	6	21	47	75
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 [4]: file.shape

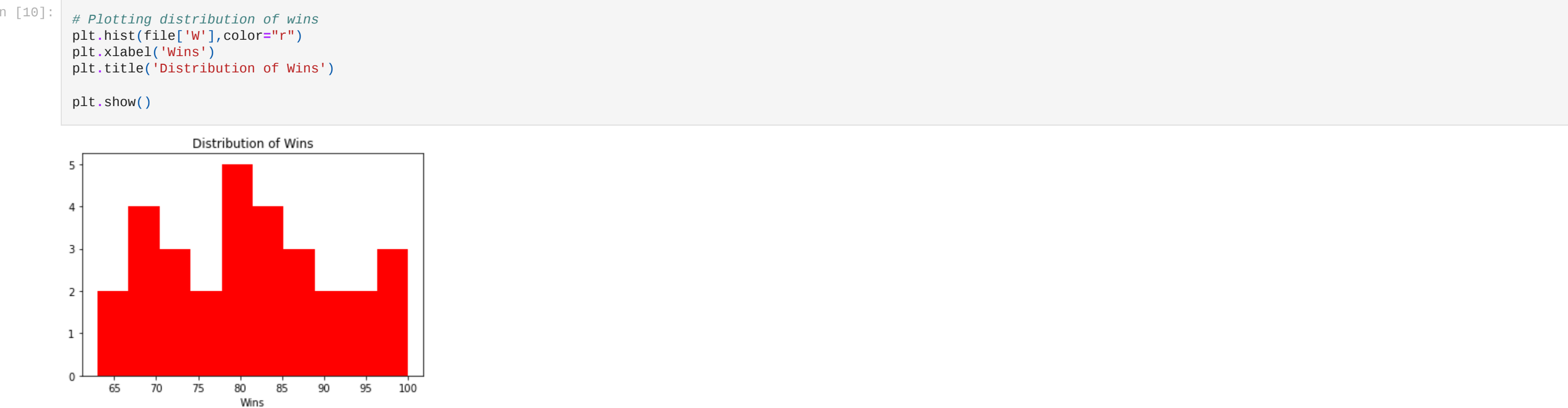
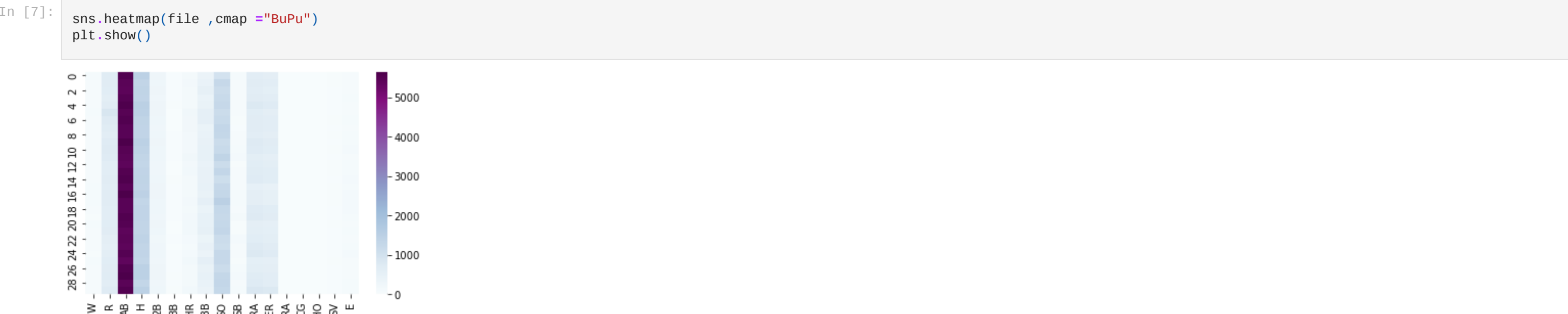
(30, 17)
```

```
In [5]: file.isnull().sum()
```

Out[5]: 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 [6]: file.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 17 columns):
#   Column      Non-Null Count  Dtype
---  -
0    W           30 non-null      int64
1    R           30 non-null      int64
2    AB          30 non-null      int64
3    H           30 non-null      int64
4    2B          30 non-null      int64
5    3B          30 non-null      int64
6    HR          30 non-null      int64
7    BB          30 non-null      int64
8    SO          30 non-null      int64
9    SB          30 non-null      int64
10   RA          30 non-null      int64
11   ER          30 non-null      int64
12   ERA        30 non-null      float64
13   CG          30 non-null      int64
14   SHO        30 non-null      int64
15   SV          30 non-null      int64
16   E           30 non-null      int64
dtypes: float64(1), int64(16)
memory usage: 4.1 KB
```



```
In [20]: # Label encoding
label_encoder = preprocessing.LabelEncoder()
file['W']= label_encoder.fit_transform(file['W'])
file

Out[20]:
```

	W	R	AB	H	2B	3B	HR	BB	SO	SB	RA	ER	ERA	CG	SHO	SV	E
0	20	724	5575	1497	300	42	139	383	973	104	641	601	3.73	2	8	56	88
1	11	696	5467	1349	277	44	156	439	1264	70	700	653	4.07	2	12	45	86
2	10	669	5439	1395	303	29	141	533	1157	86	640	584	3.67	11	10	38	79
3	6	622	5533	1381	260	27	136	404	1231	68	701	643	3.98	7	9	37	101
4	5	689	5605	1515	289	49	151	455	1259	83	803	746	4.64	7	12	35	86
5	19	891	5509	1480	308	17	232	570	1151	88	670	609	3.80	7	10	34	88
6	15	764	5567	1397	272	19	212	554	1227	63	698	652	4.03	3	4	48	93
7	10	713	5485	1370	246	20	217	418	1331	44	693	646	4.05	0	10	43	77
8	9	644	5485	1383	278	32	167	436	1310	87	642	604	3.74	1	12	60	95
9	7	748	5640	1495	294	33	161	478	1148	71	753	694	4.31	3	10	40	97
10	16	751	5511	1419	279	32	172	503	1233	101	733	680	4.24	5	9	45	119
11	14	729	5459	1363	278	26	230	486	1392	121	618	572	3.57	5	13	39	85
12	13	661	5417	1331	243	21	176	435	1150	52	675	630	3.94	2	12	46	93
13	6	656	5544	1379	262	22	198	478	1336	69	726	677	4.16	6	12	45	94
14	3	694	5600	1405	277	46	146	475	1119	78	729	664	4.14	5	15	28	126
15	23	647	5484	1386	288	39	137	506	1267	69	525	478	2.94	1	15	62	96
16	22	697	5631	1462	292	27	140	461	1322	98	596	532	3.21	0	13	54	122
17	21	689	5491	1341	272	30	171	567	1518	95	608	546	3.36	6	21	48	111
18	3	655	5480	1378	274	34	145	412	1299	84	737	682	4.28	1	7	40	116
19	1	640	5571	1382	257	27	167	496	1255	134	754	700	4.33	2	8	35	90
20	17	683	5527	1351	295	17	177	488	1290	51	613	557	3.43	1	14	50	88
21	11	703	5428	1363	265	13	177	539	1344	57	635	577	3.62	4	13	41	90
22	4	613	5463	1420	236	40	120	375	1150	112	678	638	4.02	0	12	35	77
23	2	573	5420	1361	251	18	100	471	1107	69	760	698	4.41	3	10	44	90
24	0	626	5529	1374	272	37	130	387	1274	88	809	749	4.69	1	7	35	117
25	18	667	5385	1346	263	26	187	563	1258	59	595	553	3.44	6	21	47	75
26	12	696	5565	1486	288	39	136	457	1159	93	627	597	3.72	7	18	41	78
27	8	720	5649	1494	289	48	154	490	1312	132	713	659	4.04	1	12	44	86
28	5	650	5457	1324	260	36	148	426	1327	82	731	655	4.09	1	6	41	92
29	3	737	5572	1479	274	49	186	388	1283	97	844	799	5.04	4	4	36	95

```
In [27]: # model fitting

from sklearn.model_selection import train_test_split

X=file.iloc[:,1:78]
y=file['W']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=50)
y_test = np.array(y_test,dtype = float)
```

```
In [28]: # standar scaler
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

```
In [29]: def model_accuracy(model,X_train=X_train,y_train=y_train):
accuracies = cross_val_score(estimator = model, X = X_train, y = y_train, cv =10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

```
In [30]: rand_regressor = RandomForestRegressor()
rand_regressor.fit(X_train, y_train)
y_pred_rf =rand_regressor.predict(X_test)
RandomForestRegressor(y_test,y_pred_rf)
model_accuracy(rand_regressor)

Accuracy: -880.03 %
Standard Deviation: 2235.95 %
```

```
In [31]: rand_regressor = RandomForestRegressor()
rand_regressor.fit(X_train, y_train)
y_pred_rf = rand_regressor.predict(X_test)
LinearRegression(y_test,y_pred_rf)
model_accuracy(rand_regressor)

Accuracy: -902.13 %
Standard Deviation: 2219.33 %
```

```
In [ ]:
```