In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['figure.figsize'] = (10.0, 8.0)
import seaborn as sns
from scipy import stats
from scipy.stats import norm
```

In [3]:

```
df = pd.read_csv('nba 2.csv')
df.head(10)
#decrease bars too many
```

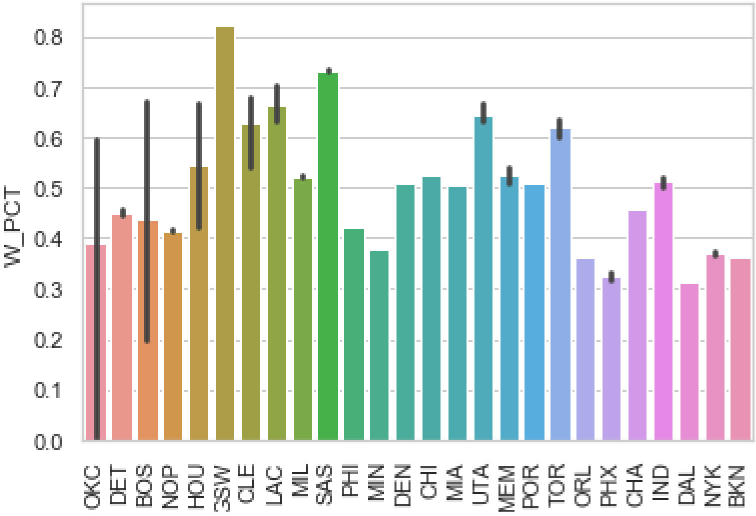
Out[3]:

	PLAYER_ID	PLAYER_NAME	TEAM_ID	TEAM_ABBREVIATION	AGE	GP	W	L	W_PCT	MI
0	201566	Russell Westbrook	1610612760	OKC	28	81	46	35	0.568	34.
1	1626246	Boban Marjanovic	1610612765	DET	28	35	16	19	0.457	8.
2	1627743	Demetrius Jackson	1610612738	BOS	22	5	1	4	0.200	3.
3	203076	Anthony Davis	1610612740	NOP	24	75	31	44	0.413	36.
4	201935	James Harden	1610612745	HOU	27	81	54	27	0.667	36.
5	201142	Kevin Durant	1610612744	GSW	28	62	51	11	0.823	33.
6	2544	LeBron James	1610612739	CLE	32	74	51	23	0.689	37.
7	101108	Chris Paul	1610612746	LAC	32	61	43	18	0.705	31.
8	202326	DeMarcus Cousins	1610612740	NOP	26	72	30	42	0.417	34.
9	203507	Giannis Antetokounmpo	1610612749	MIL	22	80	42	38	0.525	35.

10 rows × 63 columns

In [5]:
```
#plotting how many games these teams have the highest win percentage
sns.set(style="whitegrid", color_codes=True)
sns.barplot(x = 'TEAM_ABBREVIATION', y = 'W_PCT', data=df)
plt.xticks(rotation = 90)
```

Out[5]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25]),
 <a list of 26 Text xticklabel objects>)
```



In [52]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 63 columns):
PLAYER_ID                          100 non-null int64
PLAYER_NAME                        100 non-null object
TEAM_ID                            100 non-null int64
TEAM_ABBREVIATION                  100 non-null object
AGE                                100 non-null int64
GP                                 100 non-null int64
W                                  100 non-null int64
L                                  100 non-null int64
W_PCT                              100 non-null float64
MIN                                100 non-null float64
OFF_RATING                         100 non-null float64
DEF_RATING                         100 non-null float64
NET_RATING                         100 non-null float64
AST_PCT                            100 non-null float64
AST_TO                             100 non-null float64
AST_RATIO                          100 non-null float64
OREB_PCT                           100 non-null float64
DREB_PCT                           100 non-null float64
REB_PCT                            100 non-null float64
TM_TOV_PCT                         100 non-null float64
EFG_PCT                            100 non-null float64
TS_PCT                             100 non-null float64
USG_PCT                            100 non-null float64
PACE                               100 non-null float64
PIE                                100 non-null float64
FGM                                100 non-null int64
FGA                                100 non-null int64
FGM_PG                             100 non-null float64
FGA_PG                             100 non-null float64
FG_PCT                             100 non-null float64
GP_RANK                            100 non-null int64
W_RANK                             100 non-null int64
L_RANK                             100 non-null int64
W_PCT_RANK                         100 non-null int64
MIN_RANK                           100 non-null int64
OFF_RATING_RANK                    100 non-null int64
DEF_RATING_RANK                    100 non-null int64
NET_RATING_RANK                    100 non-null int64
AST_PCT_RANK                       100 non-null int64
AST_TO_RANK                        100 non-null int64
AST_RATIO_RANK                     100 non-null int64
OREB_PCT_RANK                      100 non-null int64
DREB_PCT_RANK                      100 non-null int64
REB_PCT_RANK                       100 non-null int64
TM_TOV_PCT_RANK                    100 non-null int64
EFG_PCT_RANK                       100 non-null int64
TS_PCT_RANK                        100 non-null int64
USG_PCT_RANK                       100 non-null int64
PACE_RANK                          100 non-null int64
PIE_RANK                           100 non-null int64
FGM_RANK                           100 non-null int64
FGA_RANK                           100 non-null int64
```

```
FGM_PG_RANK                      100 non-null int64
FGA_PG_RANK                      100 non-null int64
FG_PCT_RANK                      100 non-null int64
CFID                             100 non-null int64
CFPARAMS                         100 non-null object
WIKIPEDIA_HANDLE                 100 non-null object
TWITTER_HANDLE                   100 non-null object
SALARY_MILLIONS                  100 non-null float64
PTS                              100 non-null float64
ACTIVE_TWITTER_LAST_YEAR         100 non-null int64
TWITTER_FOLLOWER_COUNT_MILLIONS    100 non-null float64
dtypes: float64(23), int64(35), object(5)
memory usage: 49.3+ KB
```

In [ ]:

In [63]:

```
df.shape
```

Out[63]:

```
(100, 63)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [66]:

```
#plotting how many games these teams have the highest win percentage
```

In [50]:

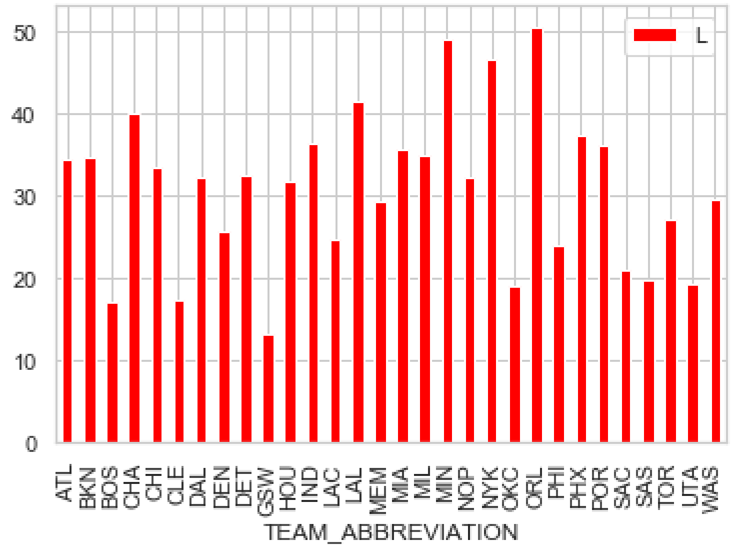
```
#Teams win percentage
sp_pivot = df.pivot_table(index='TEAM_ABBREVIATION', values='L',)
sp_pivot
```

Out[50]:

	L
TEAM_ABBREVIATION	
ATL	34.333333
BKN	34.500000
BOS	17.000000
CHA	40.000000
CHI	33.500000
CLE	17.250000
DAL	32.250000
DEN	25.500000
DET	32.500000
GSW	13.200000
HOU	31.666667
IND	36.333333
LAC	24.600000
LAL	41.500000
MEM	29.250000
MIA	35.666667
MIL	34.750000
MIN	49.000000
NOP	32.250000
NYK	46.666667
OKC	19.000000
ORL	50.500000
PHI	24.000000
PHX	37.333333
POR	36.000000
SAC	21.000000
SAS	19.800000
TOR	27.000000
UTA	19.250000
WAS	29.500000

```
In [58]: sp_pivot.plot(kind='bar',color="red")
```

```
Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x11d324078d0>
```



```
In [ ]:
```