In [1]:

```
'''
Load data
'''
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%pylab inline

pd.set_option('display.max_columns',100)
pd.set_option('display.max_rows',20)
players_df = pd.read_csv('data/nba-enhanced-stats/2017-18_playerBoxScore.csv')
team_df = pd.read_csv('data/nba-enhanced-stats/2017-18_teamBoxScore.csv')

print('\n\nThe dimension of players df is:',players_df.shape)
players_df.head(1)
```

```
Populating the interactive namespace from numpy and matplotlib


The dimension of players df is: (26109, 51)
```

Out[1]:

	gmDate	gmTime	seasTyp	playLNm	playFNm	teamAbbr	teamConf	teamDiv	teamLoc
0	2017-10-17	08:00	Regular	Brown	Jaylen	BOS	East	Atlantic	Away

In [2]:

```
print('The dimension of team df is:',team_df.shape)
team_df.head(1)
```

```
The dimension of team df is: (2460, 123)
```

Out[2]:

	gmDate	gmTime	seasTyp	offLNm1	offFNm1	offLNm2	offFNm2	offLNm3	offFNm3	t
0	2017-10-17	08:00	Regular	Forte	Brian	Smith	Michael	McCutchen	Monty	

1 rows × 123 columns

In [3]:

```
'''
Filter & Clean Data:
    Filter data to only include teams for playoffs matches
    Deal with NA
    Output data
'''

# Filter data to only include teams for playoffs matches
playoffs_team = ['TOR','BOS','PHI','CLE','IND','MIA','MIL','WAS','HOU','GS','PO
R','OKC','UTA',
                 'NO','SAC','MIN']

players_playoffs_df = players_df[players_df['teamAbbr'].isin(playoffs_team)]
team_playoffs_df = team_df[team_df['teamAbbr'].isin(playoffs_team)]

# Compute numbers of value with na
na_play = players_playoffs_df.isna().sum()
print ('na information for players:\n',na_play[na_play>0])

# Remove columns with na, here are offLNm3 and offFNm3 (not important features)
na_play = na_play[na_play>0].index.tolist()
players_playoffs_df.drop(columns = na_play,inplace= True)
print('The new dimension of players df is:',players_playoffs_df.shape)

# Compute numbers of value with na
na_team = team_playoffs_df.isna().sum()
print('\n na information for teams:\n',na_team[na_team>0])

# Remove columns with na, here are offLNm3 and offFNm3 (not important features)
na_team = na_team[na_team>0].index.tolist()
team_playoffs_df.drop(columns = na_team,inplace= True)
print('The new dimension of teams df is:',team_playoffs_df.shape)

# Ouput these two df
players_playoffs_df.to_csv('cleaned_players_stat.csv')
team_playoffs_df.to_csv('cleaned_teams_stat.csv')
```

```
na information for players:
 offLNm3    20
offFNm3    20
dtype: int64
The new dimension of players df is: (13844, 49)

 na information for teams:
 offLNm3    2
offFNm3    2
dtype: int64
The new dimension of teams df is: (1312, 121)

/Users/stevechen/Documents/Tools/anaconda3/lib/python3.7/site-packag
es/pandas/core/frame.py:4102: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y
  errors=errors,
```

In [4]:

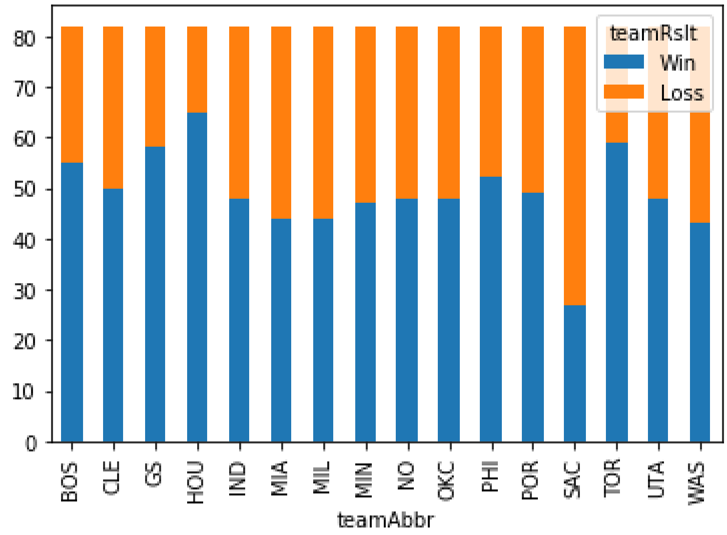
```
'''
Simply data visualizations:
    Stacked plot to show numbers of win and loss for teams

'''

# Stacked plot to show numbers of win and loss for teams
wl_df = team_playoffs_df.groupby(['teamAbbr','teamRslt'])['teamAbbr'].count().un
stack('teamRslt')
wl_df[['Win','Loss']].plot(kind='bar', stacked=True)
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11ba96150>
```



In [5]:

```
wl_df
team_playoffs_df.groupby(['teamAbbr','teamRslt'])['teamAbbr'].count().unstack('t
eamRslt')
```

Out[5]:

teamRslt	Loss	Win
teamAbbr		
BOS	27	55
CLE	32	50
GS	24	58
HOU	17	65
IND	34	48
MIA	38	44
MIL	38	44
MIN	35	47
NO	34	48
OKC	34	48
PHI	30	52
POR	33	49
SAC	55	27
TOR	23	59
UTA	34	48
WAS	39	43

In [6]:

```
team_playoffs_df1 = team_playoffs_df[['teamAbbr','teamFG%']]
team_playoffs_df1
```

Out[6]:

	teamFG%
0	0.4091
1	0.4578
2	0.4845
3	0.5375
7	0.5196
...	...
2453	0.4952
2456	0.3708
2457	0.4607
2458	0.3780
2459	0.4750

1312 rows × 1 columns

In [ ]:

```
playoffs_team = ['TOR','BOS','PHI','CLE','IND','MIA','MIL','WAS','HOU','GS','POR','OKC','UTA',
                'NO','SAC','MIN']
```

In [ ]:

```
```