

```
In [1]: 1 import numpy as np
        2 import pandas as pd
        3 from matplotlib import pyplot as plt
        4 import seaborn as sns
        5 import time
        6 plt.style.use('ggplot')
        7 %matplotlib inline
```

```
In [2]: 1 from nba_py.team import TeamYearOverYearSplits, TeamList
        2 team_list = TeamList().info().head(30)
```

```
In [3]: 1 regular_stats = pd.read_csv('all_team_playoffs.csv')
        2 advs_stats = pd.read_csv('all_team_playoffs_adv.csv')
```

```
In [4]: 1 regular_features = regular_stats[['GROUP_VALUE', 'FGM', 'FGA', 'FG_PCT',
        2                                'FT_PCT', 'OREB', 'DREB', 'REB', 'AST', 'TOV', 'STL', 'BLK', 'BLK_PCT',
        3                                'PF', 'PFD', 'PTS', 'PLUS_MINUS', 'PLAYOFFS']]
```

```
In [5]: 1 playoff_reg = regular_features[regular_features['PLAYOFFS'] == 1].groupby(['YEAR'])
        2 playoff_reg['YEAR'] = playoff_reg.index
```

```
In [6]: 1 reg_means = regular_features.groupby(['GROUP_VALUE']).mean()
        2 reg_means['YEAR'] = reg_means.index
```

```
In [7]: 1 advs_features = advs_stats[['GROUP_VALUE', 'NET_RATING', 'AST_PCT', 'AST_RATIO',
        2                            'AST_RATIO', 'OREB_PCT', 'DREB_PCT', 'REB_PCT', 'TM_TOV_PCT', 'EFG_PCT',
        3                            'TS_PCT', 'PACE', 'PIE', 'PLAYOFFS']]
        4 adv_means = advs_features.groupby(['GROUP_VALUE']).mean()
        5 adv_means['YEAR'] = adv_means.index
        6 playoff_adv = advs_features[advs_features['PLAYOFFS'] == 1].groupby(['GROUP_VALUE'])
        7 playoff_adv['YEAR'] = playoff_adv.index
```

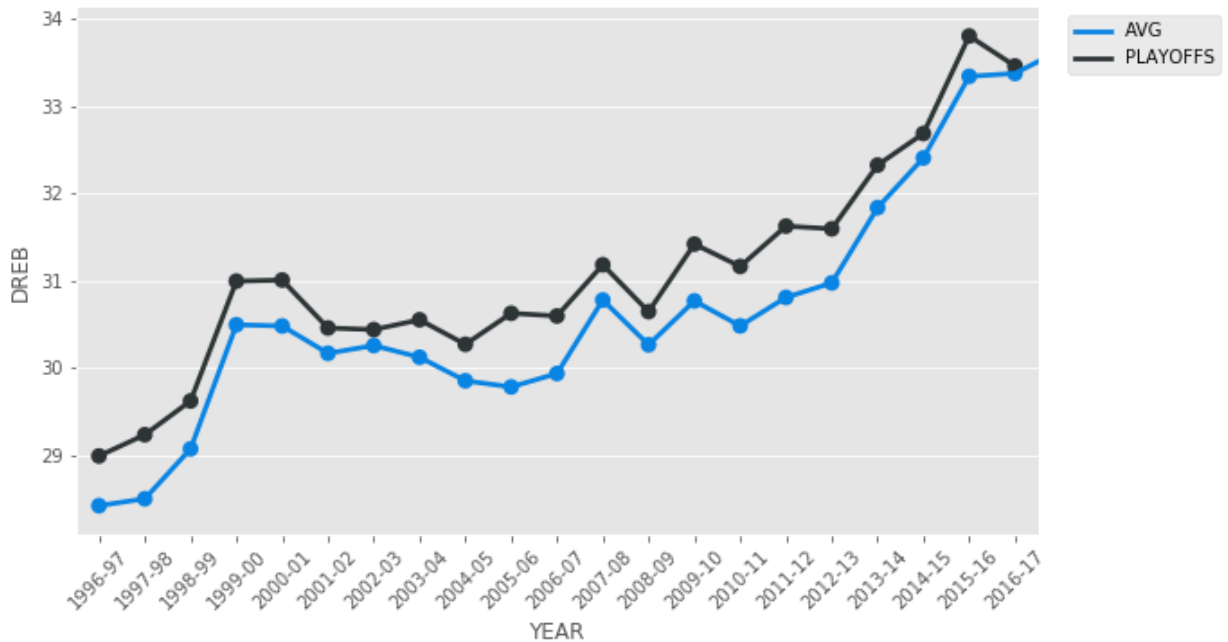
In [33]:

```

1 fig, ax = plt.subplots(figsize=(11,6))
2 sns.pointplot(x='YEAR',y='DREB', data=reg_means,color='#0984e3')
3 sns.pointplot(x='YEAR',y='DREB', data=playoff_reg, color='#2d3436')
4 leg = plt.legend(labels=['AVG','PLAYOFFS'], bbox_to_anchor=[1.2,1.0])
5 leg.legendHandles[0].set_color('#0984e3')
6 leg.legendHandles[1].set_color('#2d3436')
7 fig.suptitle('Avg DREB by Year for Playoff Teams vs League AVG', fontsize=14)
8 ticks = plt.xticks(rotation=45)
9 plt.subplots_adjust(bottom=0.2,right=0.8)
10 fig.savefig('dreb_by_year')

```

Avg DREB by Year for Playoff Teams vs League AVG

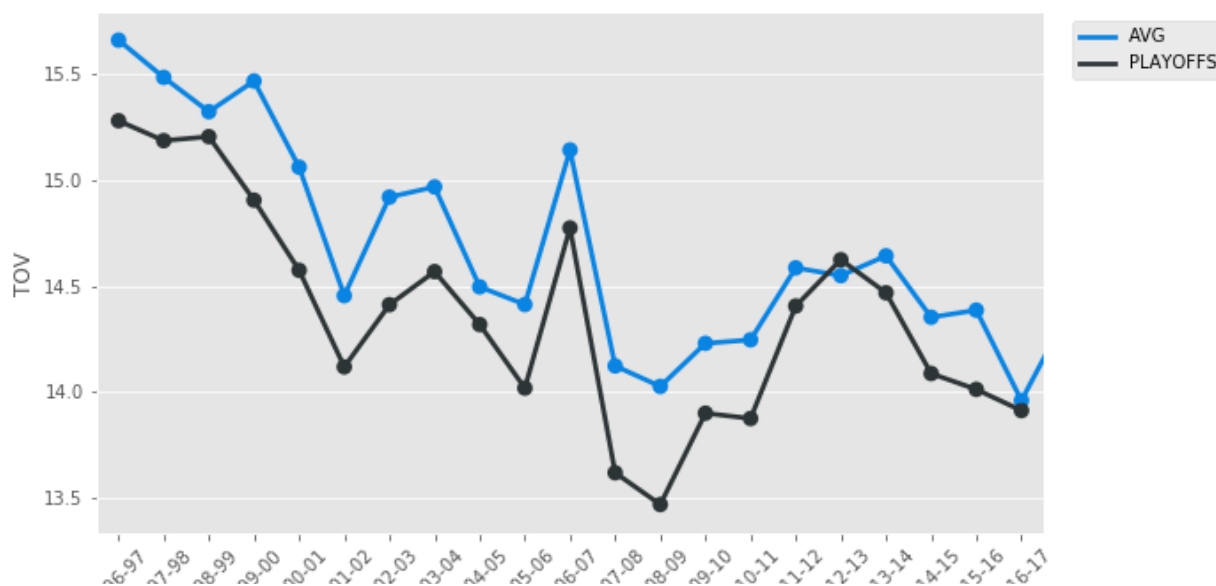


```

In [34]: 1 stat_to_analyze = 'TOV'
2 fig, ax = plt.subplots(figsize=(11,6))
3 sns.pointplot(x='YEAR',y=stat_to_analyze, data=reg_means,color='#0984e3')
4 sns.pointplot(x='YEAR',y=stat_to_analyze, data=playoff_reg, color='#2d3436')
5 leg = plt.legend(labels=['AVG','PLAYOFFS'], bbox_to_anchor=[1.2,1.0])
6 leg.legendHandles[0].set_color('#0984e3')
7 leg.legendHandles[1].set_color('#2d3436')
8 fig.suptitle('Avg %s by Year for Playoff Teams vs League avg' % (stat_to_analyze))
9 ticks = plt.xticks(rotation=45)
10 plt.subplots_adjust(bottom=0.2,right=0.8)
11 fig.savefig('%s_by_year' % stat_to_analyze)

```

Avg TOV by Year for Playoff Teams vs League avg



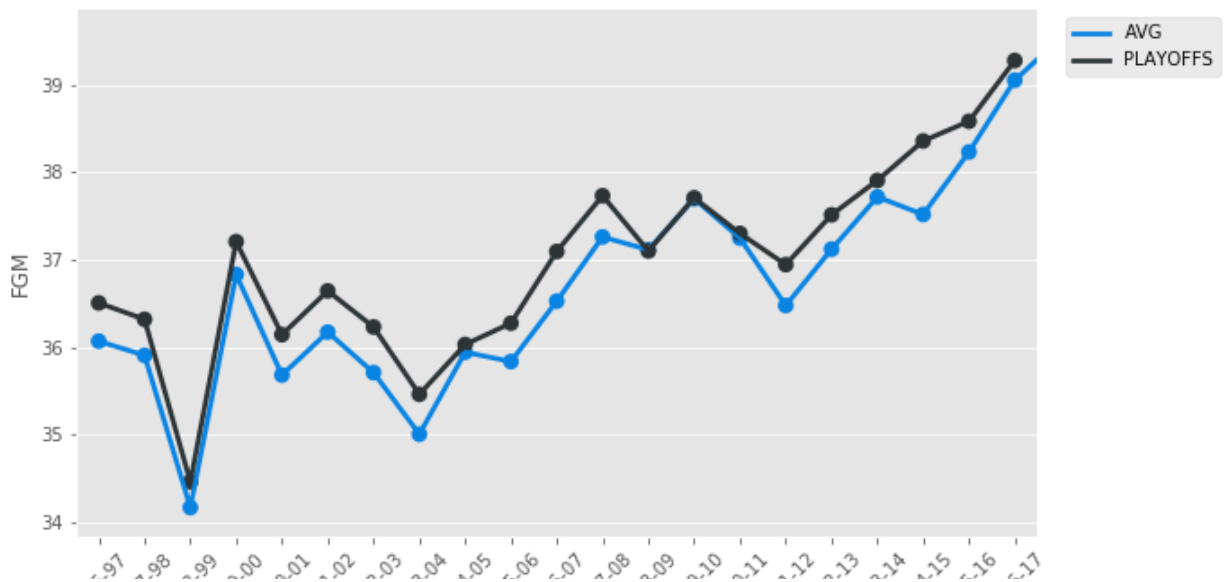
In [36]:

```

1 features_analyze = ['FGM', 'FGA', 'FG_PCT', 'FG3M', 'FG3A', 'FG3_PCT',
2                     'FT_PCT', 'OREB', 'DREB', 'REB', 'AST', 'TOV', 'STL', 'BLK', 'BLK_PCT',
3                     'PF', 'PFD', 'PTS']
4 for feature in features_analyze:
5     fig, ax = plt.subplots(figsize=(11,6))
6     sns.pointplot(x='YEAR',y=feature, data=reg_means,color='#0984e3')
7     sns.pointplot(x='YEAR',y=feature, data=playoff_reg, color='#2d3436')
8     leg = plt.legend(labels=['AVG','PLAYOFFS'], bbox_to_anchor=[1.2,1.0])
9     leg.legendHandles[0].set_color('#0984e3')
10    leg.legendHandles[1].set_color('#2d3436')
11    fig.suptitle('Avg %s by Year for Playoff Teams vs League avg' % (feature))
12    ticks = plt.xticks(rotation=45)
13    plt.subplots_adjust(bottom=0.2,right=0.8)
14    fig.savefig('%s_by_year' % feature)

```

Avg FGM by Year for Playoff Teams vs League avg

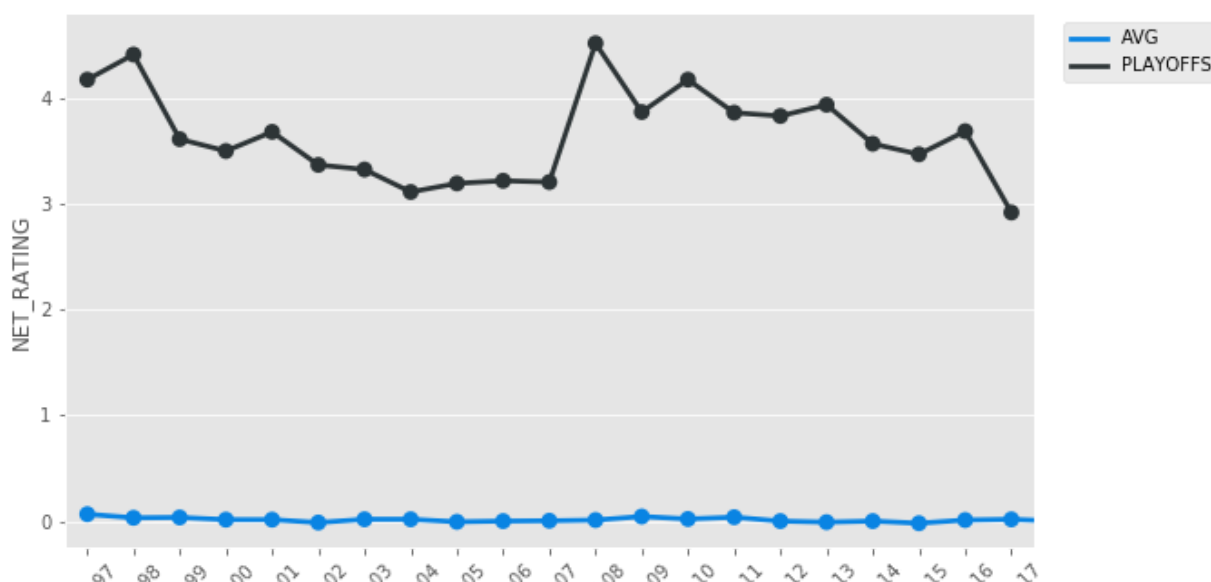


```

In [37]: 1 features_analyze = ['NET_RATING', 'AST_PCT', 'AST_TO',
2         'AST_RATIO', 'OREB_PCT', 'DREB_PCT', 'REB_PCT', 'TM_TOV_PCT', 'E
3         'TS_PCT', 'PACE', 'PIE']
4 for feature in features_analyze:
5     fig, ax = plt.subplots(figsize=(11,6))
6     sns.pointplot(x='YEAR',y=feature, data=adv_means,color='#0984e3')
7     sns.pointplot(x='YEAR',y=feature, data=playoff_adv, color='#2d3436')
8     leg = plt.legend(labels=['AVG','PLAYOFFS'], bbox_to_anchor=[1.2,1.0]
9     leg.legendHandles[0].set_color('#0984e3')
10    leg.legendHandles[1].set_color('#2d3436')
11    fig.suptitle('Avg %s by Year for Playoff Teams vs League avg' % (fea
12    ticks = plt.xticks(rotation=45)
13    plt.subplots_adjust(bottom=0.2,right=0.8)
14    fig.savefig('%s_by_year' % feature)

```

Avg NET\_RATING by Year for Playoff Teams vs League avg



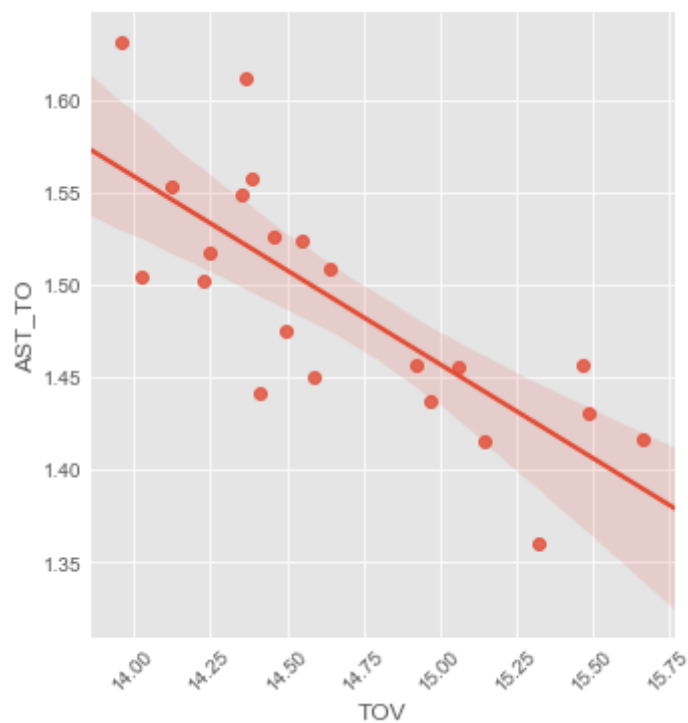
```

In [31]: 1 all_stats = pd.concat([reg_means,adv_means], axis=1)
2        playoff_all_stats = pd.concat([playoff_reg,playoff_adv], axis=1)

```

```
In [70]: 1 sns.lmplot(x='TOV',y='AST_TO', data=all_stats)
        2 plt.xticks(rotation=45)
```

```
Out[70]: (array([ 13.75,  14.   ,  14.25,  14.5 ,  14.75,  15.   ,  15.25,  15.5 ,
                  15.75,  16.   ]), <a list of 10 Text xticklabel objects>)
```



```
In [ ]: 1
```