shooting visualizations

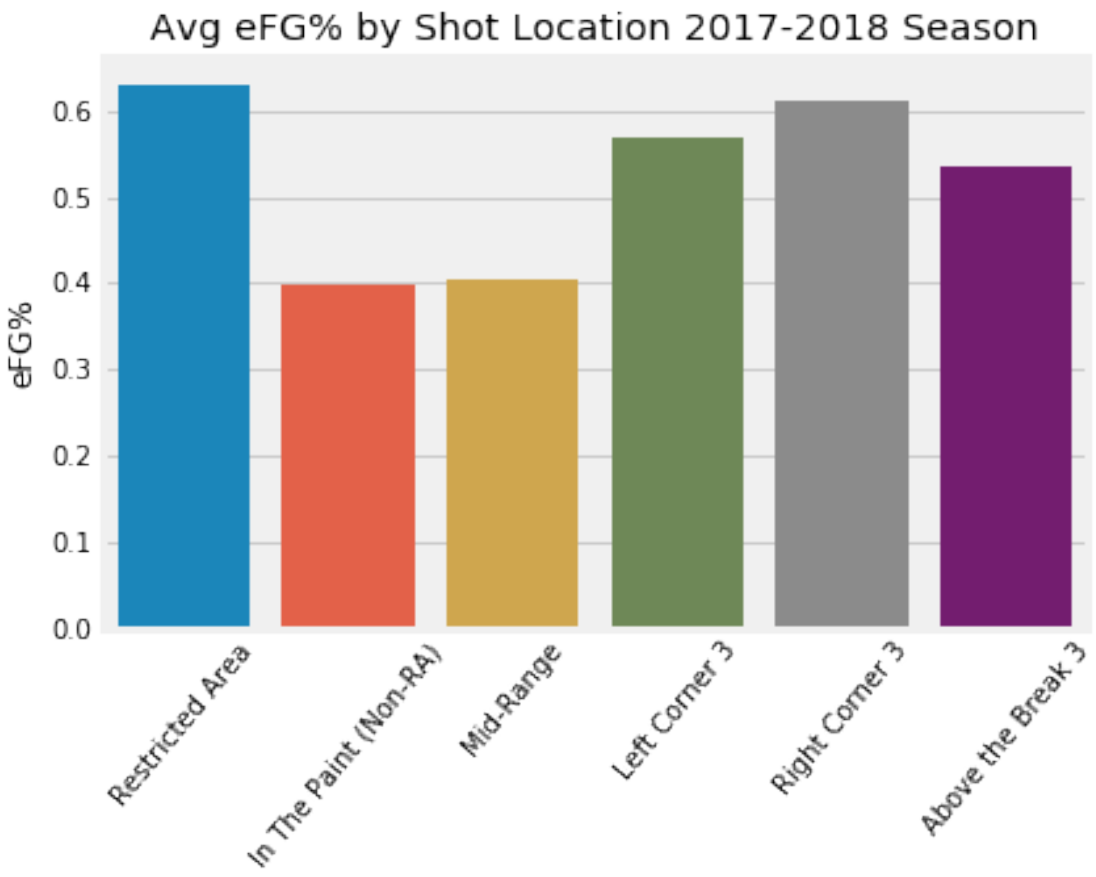
March 25, 2018

In [1]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import time
%matplotlib inline
```

In [42]:
```
all_players = pd.read_csv('all_players.csv')
real_players = pd.read_csv('real_players.csv').drop(['Unnamed: 0','Unnamed: 0.1'], ax
```

In [24]:
```
ax = sns.barplot(x = list(avgs.index),y = avgs.values)
plt.xticks(rotation=50)
ax.set_ylabel('eFG%')
ax.set_title('Avg eFG% by Shot Location 2017-2018 Season')
```

Out[24]:
```
<matplotlib.text.Text at 0x11501ff60>
```




```
In [41]: real_players.sort_values(by=['SEASON','PTS'],ascending=False)[['PLAYER','SEASON','POSI
```

Out[41]:

	PLAYER	SEASON	POSITION	PTS	FG3A	FG3M	\
2812	Russell Westbrook	2016	G	31.580247	7.197531	2.469136	
2667	James Harden	2016	G	29.086420	9.333333	3.234568	
2590	Isaiah Thomas	2016	G	28.934211	8.500000	3.223684	
2614	Anthony Davis	2016	F-C	27.986667	1.786667	0.533333	
2828	DeMar DeRozan	2016	G	27.297297	1.675676	0.445946	
2780	Damian Lillard	2016	G	26.986667	7.720000	2.853333	
2610	DeMarcus Cousins	2016	F-C	26.972222	5.041667	1.819444	
2606	LeBron James	2016	F	26.405405	4.621622	1.675676	
2802	Kawhi Leonard	2016	F	25.513514	5.229730	1.986486	
2657	Stephen Curry	2016	G	25.303797	9.987342	4.101266	
2600	Kyrie Irving	2016	G	25.222222	6.125000	2.458333	
2718	Karl-Anthony Towns	2016	C	25.134146	3.353659	1.231707	
2659	Kevin Durant	2016	F	25.080645	5.032258	1.887097	
2627	Jimmy Butler	2016	F	23.894737	3.263158	1.197368	
2753	Paul George	2016	F	23.666667	6.613333	2.600000	
2717	Andrew Wiggins	2016	F	23.573171	3.524390	1.256098	

2880	Kemba Walker	2016	G	23.164557	7.620253	3.037975
2856	John Wall	2016	G	23.141026	3.487179	1.141026
2857	Bradley Beal	2016	G	23.103896	7.168831	2.896104
2782	CJ McCollum	2016	G	22.962500	5.487500	2.312500
2708	Giannis Antetokounmpo	2016	F-G	22.900000	2.250000	0.612500
2733	Carmelo Anthony	2016	F	22.418919	5.689189	2.040541
2826	Kyle Lowry	2016	G	22.400000	7.800000	3.216667
2653	Klay Thompson	2016	G	22.333333	8.294872	3.435897
2770	Devin Booker	2016	G	22.128205	5.192308	1.884615
2842	Gordon Hayward	2016	F	21.931507	5.123288	2.041096
2679	Blake Griffin	2016	F	21.573770	1.852459	0.622951
2771	Eric Bledsoe	2016	G	21.060606	4.696970	1.575758
2724	Brook Lopez	2016	C	20.520000	5.160000	1.786667
2850	Mike Conley	2016	G	20.507246	6.072464	2.478261
...	...	...	...	...	...	...
204	Ime Udoka	2007	F	5.794521	2.260274	0.835616
264	Jared Dudley	2007	F	5.780822	0.561644	0.123288
194	Anthony Johnson	2007	G	5.608696	1.347826	0.608696
143	Sean Williams	2007	F-C	5.602740	0.000000	0.000000
116	Royal Ivey	2007	G	5.573333	1.426667	0.466667
88	Josh Powell	2007	C-F	5.515625	0.046875	0.000000
93	Trevor Ariza	2007	F	5.485714	0.514286	0.142857
202	Damon Stoudamire	2007	G	5.283333	2.483333	0.833333
85	Dan Dickau	2007	G	5.253731	1.611940	0.537313
176	Reggie Evans	2007	F	5.246914	0.012346	0.012346
7	Zaza Pachulia	2007	C	5.193548	0.048387	0.000000
266	Earl Boykins	2007	G	5.138889	1.222222	0.388889
35	Rasual Butler	2007	F-G	4.941176	2.607843	0.862745
245	Kwame Brown	2007	C	4.842105	0.000000	0.000000
205	Fabricio Oberto	2007	C	4.841463	0.012195	0.000000
19	Ben Wallace	2007	C-F	4.833333	0.055556	0.000000
189	Joel Przybilla	2007	C	4.818182	0.012987	0.000000
51	Malik Allen	2007	F	4.630137	0.027397	0.013699
89	Brevin Knight	2007	G	4.567568	0.081081	0.000000
2	Acie Law	2007	G	4.196429	0.607143	0.125000
207	Jacque Vaughn	2007	G	4.135135	0.135135	0.040541
87	Quinton Ross	2007	G-F	4.092105	0.276316	0.118421
127	Greg Buckner	2007	G-F	4.000000	1.612903	0.483871
48	Eddie Jones	2007	G-F	3.744681	2.468085	0.723404
147	Jared Jeffries	2007	F	3.684932	0.342466	0.054795
54	Devean George	2007	F-G	3.660377	1.283019	0.415094
82	Dikembe Mutombo	2007	C	3.025641	0.000000	0.000000
81	Chuck Hayes	2007	F	3.000000	0.037975	0.000000
139	DeSagana Diop	2007	C	2.860759	0.000000	0.000000
244	Jason Collins	2007	C-F	1.891892	0.013514	0.000000

	FG3_PCT	FGA	FGM	FG_PCT
2812	0.327531	23.962963	10.172840	0.425136

2667	0.339099	18.925926	8.320988	0.443272
2590	0.374763	19.381579	8.973684	0.463868
2614	0.194453	20.346667	10.266667	0.501253
2828	0.184351	20.878378	9.743243	0.465730
2780	0.358987	19.840000	8.813333	0.440600
2610	0.339222	19.888889	8.986111	0.451403
2606	0.327662	18.162162	9.945946	0.552878
2802	0.382622	17.729730	8.594595	0.486068
2657	0.401316	18.265823	8.544304	0.467443
2600	0.401472	19.722222	9.319444	0.476694
2718	0.355610	18.048780	9.780488	0.542793
2659	0.360855	16.548387	8.887097	0.533984
2627	0.341079	16.473684	7.500000	0.458526
2753	0.387200	17.973333	8.293333	0.456147
2717	0.341159	19.146341	8.646341	0.445110
2880	0.392519	18.341772	8.139241	0.447392
2856	0.296436	18.397436	8.294872	0.448462
2857	0.408740	17.168831	8.272727	0.478922
2782	0.412550	18.012500	8.650000	0.475700
2708	0.227912	15.737500	8.200000	0.510813
2733	0.343230	18.770270	8.135135	0.434230
2826	0.412333	15.300000	7.100000	0.468300
2653	0.409038	17.641026	8.256410	0.467295
2770	0.333679	18.346154	7.769231	0.418321
2842	0.395959	15.835616	7.465753	0.466753
2679	0.232574	15.918033	7.852459	0.490967
2771	0.309955	15.666667	6.803030	0.436000
2724	0.307053	15.626667	7.400000	0.470453
2850	0.379638	14.623188	6.724638	0.462507
...	...	...	...	...
204	0.312192	5.136986	2.178082	0.379151
264	0.084466	4.767123	2.232877	0.415712
194	0.266667	5.144928	2.246377	0.397116
143	0.000000	4.178082	2.246575	0.465342
116	0.213773	5.040000	1.986667	0.353320
88	0.000000	4.921875	2.265625	0.401969
93	0.085714	3.885714	1.971429	0.413886
202	0.249333	5.616667	2.000000	0.351200
85	0.250463	4.417910	1.850746	0.392313
176	0.012346	4.271605	1.876543	0.415901
7	0.000000	3.951613	1.725806	0.410677
266	0.245361	4.777778	1.694444	0.326611
35	0.237431	5.098039	1.784314	0.305745
245	0.000000	3.736842	1.894737	0.486763
205	0.000000	3.609756	2.195122	0.547744
19	0.000000	4.888889	1.916667	0.377847
189	0.000000	3.246753	1.870130	0.566571
51	0.013699	4.479452	2.150685	0.414192

```
89    0.000000   4.621622   1.864865  0.331459
2     0.083321   4.232143   1.696429  0.319393
207   0.033784   3.824324   1.635135  0.365892
87    0.073461   4.407895   1.723684  0.357987
127   0.176323   3.774194   1.451613  0.275903
48    0.238596   3.829787   1.404255  0.331745
147   0.034247   3.698630   1.479452  0.353192
54    0.206604   3.905660   1.396226  0.315472
82    0.000000   2.051282   1.102564  0.347462
81    0.000000   2.797468   1.430380  0.503861
139   0.000000   2.278481   1.189873  0.386924
244   0.000000   1.527027   0.716216  0.288770

[2884 rows x 10 columns]
```

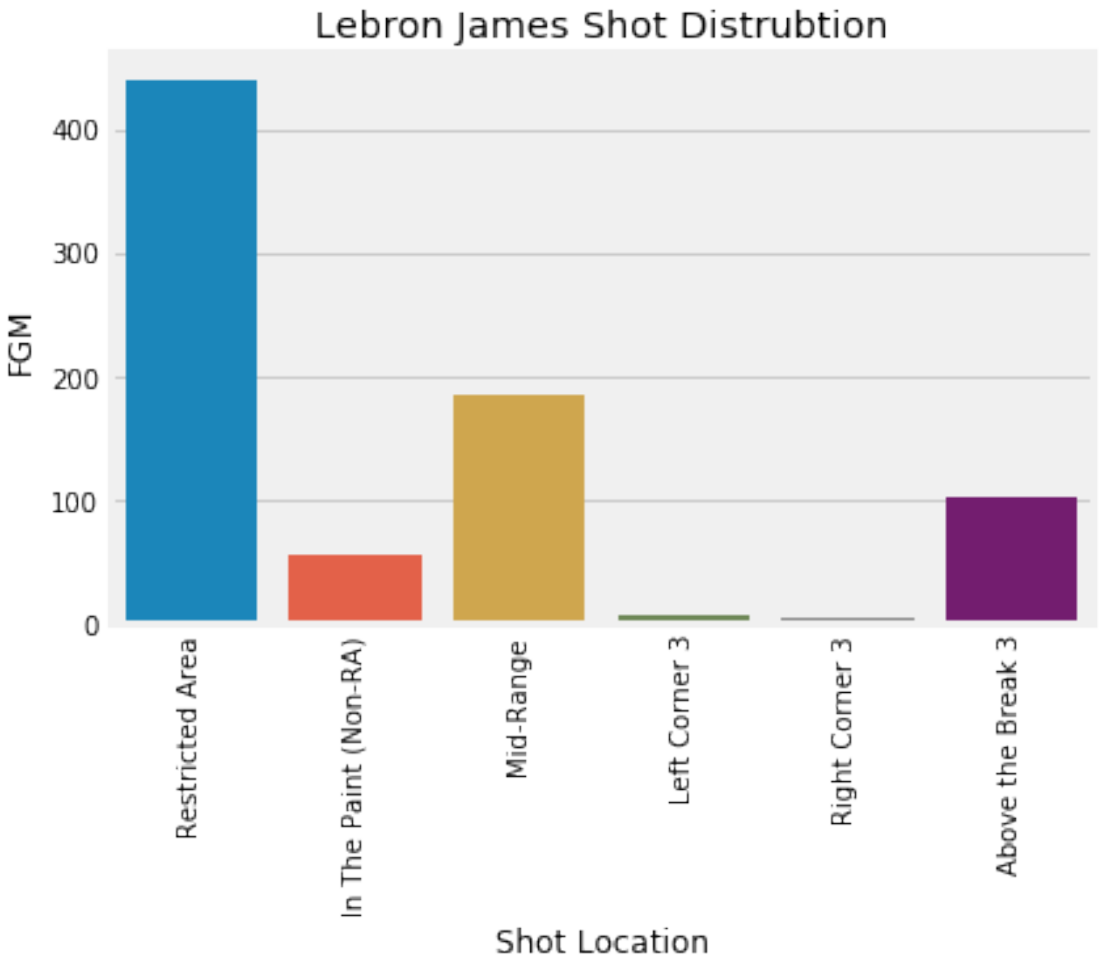
In [125]:
```
top_scorers = []
for year in range(2007,2017):
    top_scorers.append(real_players.iloc[real_players[real_players['SEASON'] == year]
```

In [69]:
```
from nba_py.player import PlayerShootingSplits
lebron_shots = PlayerShootingSplits(2544, season='2007-08')
```

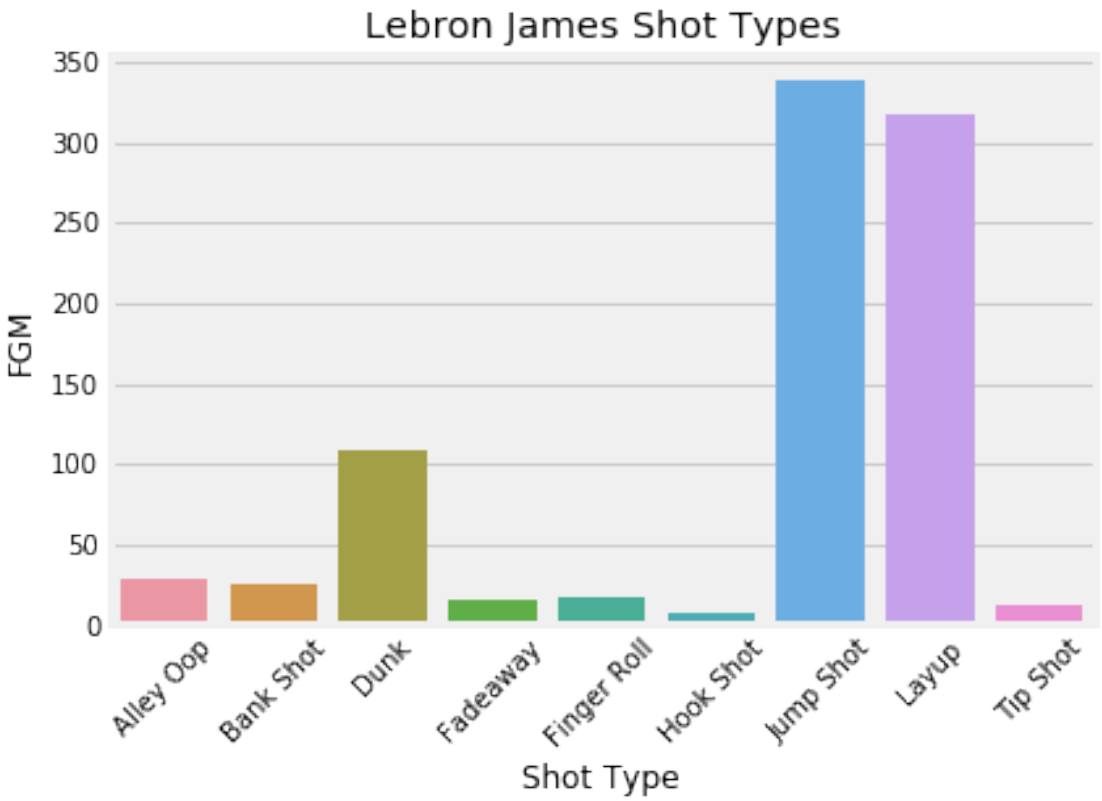
In [71]:
```
shot_areas = lebron_shots.shot_areas().drop(lebron_shots.shot_areas().index[6])
shot_types = lebron_shots.shot_types_summary()
```

In [90]:
```
ax = sns.barplot(x='GROUP_VALUE', y='FGM', data=shot_areas)
plt.xticks(rotation=90)
ax.set_xlabel('Shot Location')
ax.set_title('Lebron James Shot Distrubtion')
```

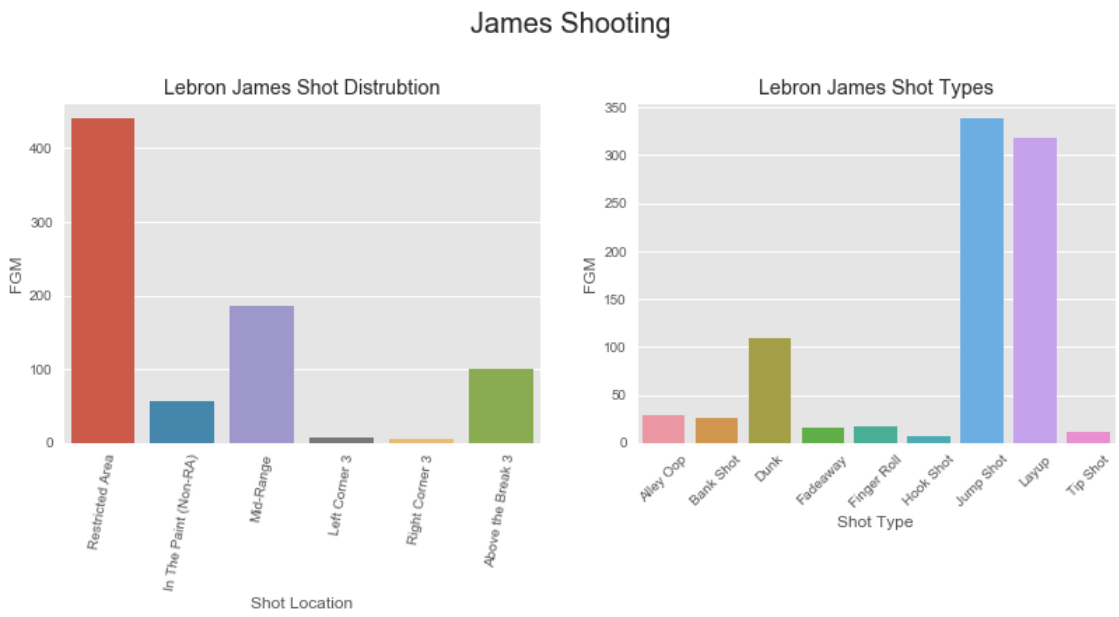
Out[90]:
```
<matplotlib.text.Text at 0x119c02e80>
```




```
In [92]: ax = sns.barplot(x='GROUP_VALUE', y='FGM', data=shot_types)
         plt.xticks(rotation=45)
         ax.set_xlabel('Shot Type')
         ax.set_title('Lebron James Shot Types')

Out[92]: <matplotlib.text.Text at 0x119cc4128>
```

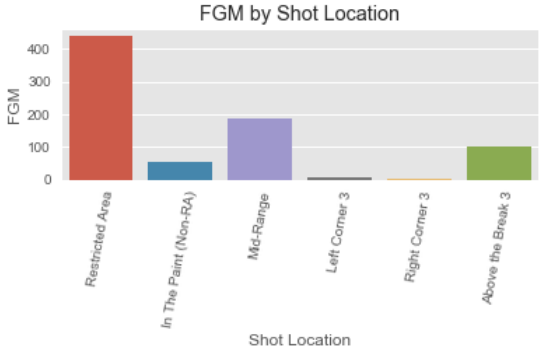


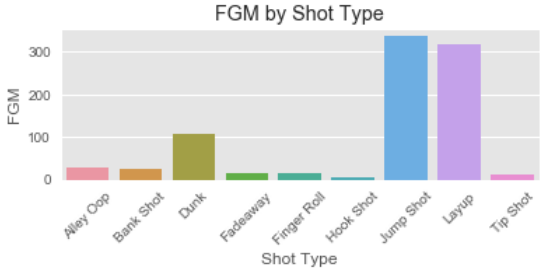

```
In [119]: plt.style.use('ggplot')
          fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12,5))
          sns.barplot(x='GROUP_VALUE', y='FGM', data=shot_areas, ax=ax1)
          ax1.set_xlabel('Shot Location')
          ax1.set_title('Lebron James Shot Distrubtion')
          for tick in ax1.get_xticklabels():
              tick.set_rotation(80)
          sns.barplot(x='GROUP_VALUE', y='FGM', data=shot_types, ax=ax2)
          plt.xticks(rotation=45)
          ax2.set_xlabel('Shot Type')
          ax2.set_title('Lebron James Shot Types')
          fig.suptitle('James Shooting', fontsize=20)
          plt.subplots_adjust(wspace=0.2, top=0.8)
          fig.savefig("bronbron.png")
          plt.show()
```



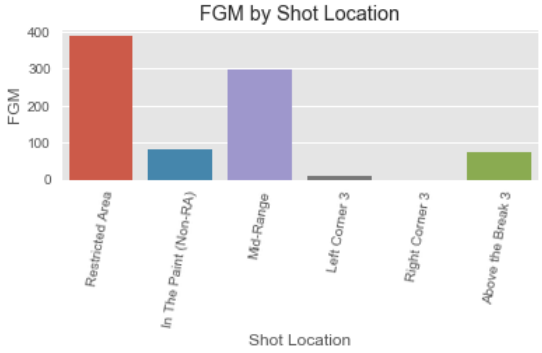

```
In [174]: for scorer in top_scorers:
              fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12,5))
              season = str(scorer['SEASON'])+"-"+str(scorer['SEASON']+1)[2:]
              player_shots = PlayerShootingSplits(scorer['PLAYER_ID'], season=season)
              shot_areas = player_shots.shot_areas().drop(player_shots.shot_areas().index[6])
              shot_types = player_shots.shot_types_summary()
              sns.barplot(x='GROUP_VALUE', y='FGM', data=shot_areas, ax=ax1)
              ax1.set_xlabel('Shot Location')
              ax1.set_title('FGM by Shot Location')
              for tick in ax1.get_xticklabels():
                  tick.set_rotation(80)
              sns.barplot(x='GROUP_VALUE', y='FGM', data=shot_types, ax=ax2)
              plt.xticks(rotation=45)
              ax2.set_xlabel('Shot Type')
              ax2.set_title('FGM by Shot Type')
              fig.suptitle('%s - %s Season (%.2f PPG) Shooting Breakdown' % (scorer['PLAYER'],s
              plt.subplots_adjust(wspace=0.2, top=0.8, bottom=0.5)
              fig.savefig("%s_%s_scoring.png" % (scorer['PLAYER'],season))
              plt.show()
              time.sleep(3)
```

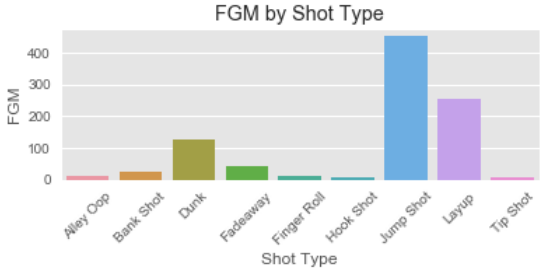
LeBron James - 2007-08 Season (30.00 PPG) Shooting Breakdown



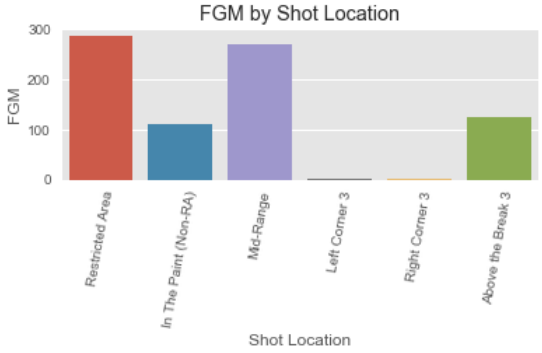


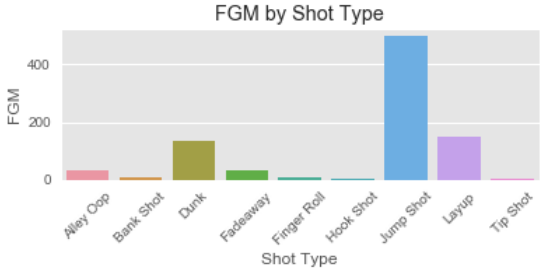
Dwyane Wade - 2008-09 Season (30.20 PPG) Shooting Breakdown



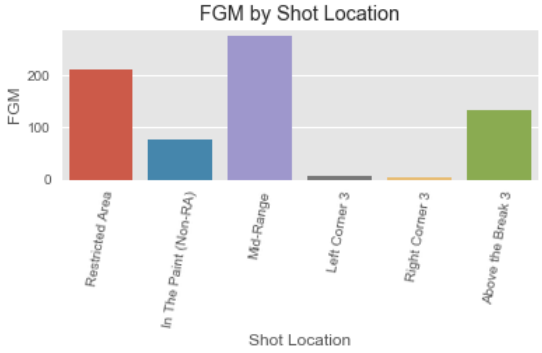


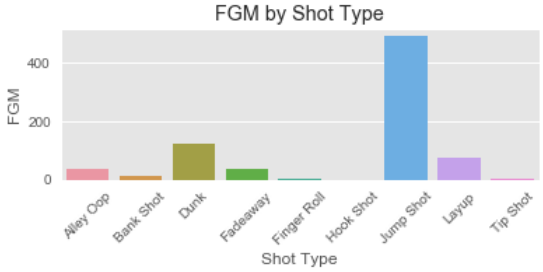
Kevin Durant - 2009-10 Season (30.15 PPG) Shooting Breakdown



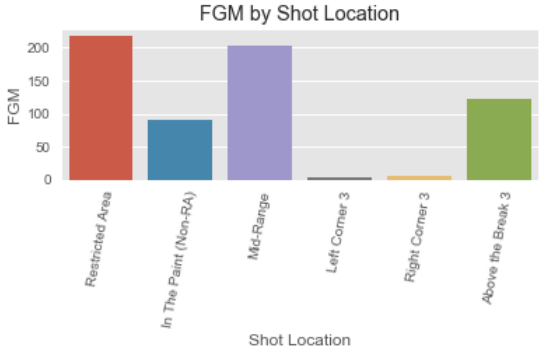


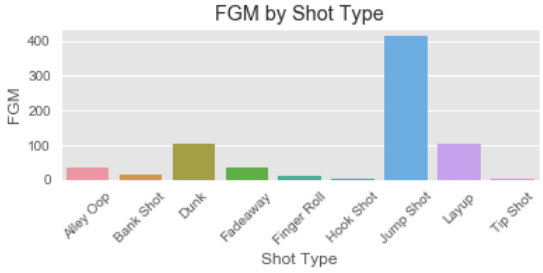
Kevin Durant - 2010-11 Season (27.71 PPG) Shooting Breakdown



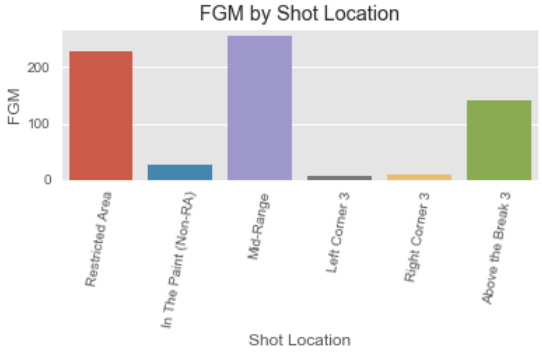


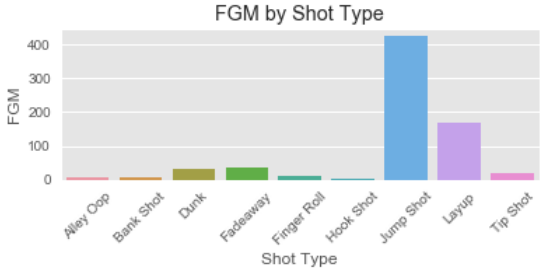
Kevin Durant - 2011-12 Season (28.03 PPG) Shooting Breakdown



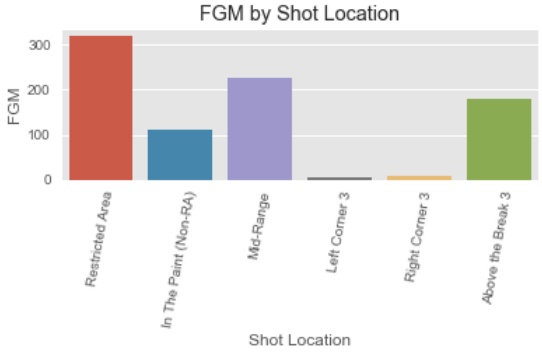


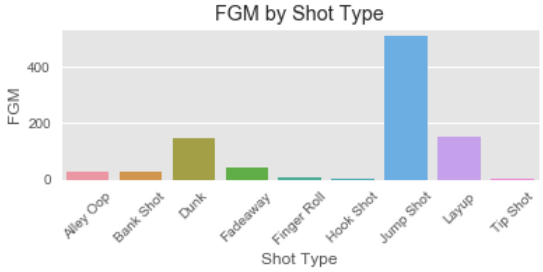
Carmelo Anthony - 2012-13 Season (28.66 PPG) Shooting Breakdown



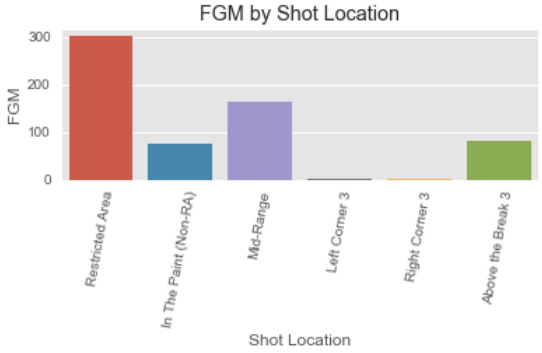


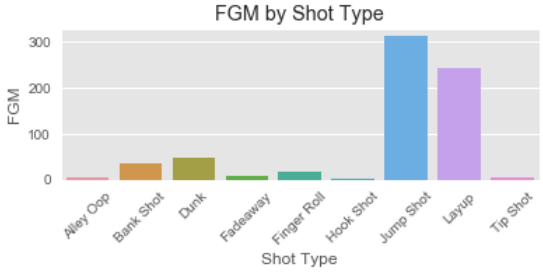
Kevin Durant - 2013-14 Season (32.01 PPG) Shooting Breakdown



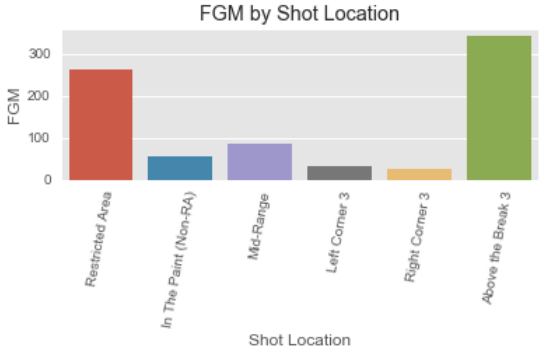


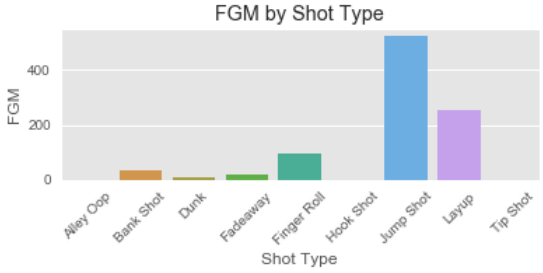
Russell Westbrook - 2014-15 Season (28.15 PPG) Shooting Breakdown

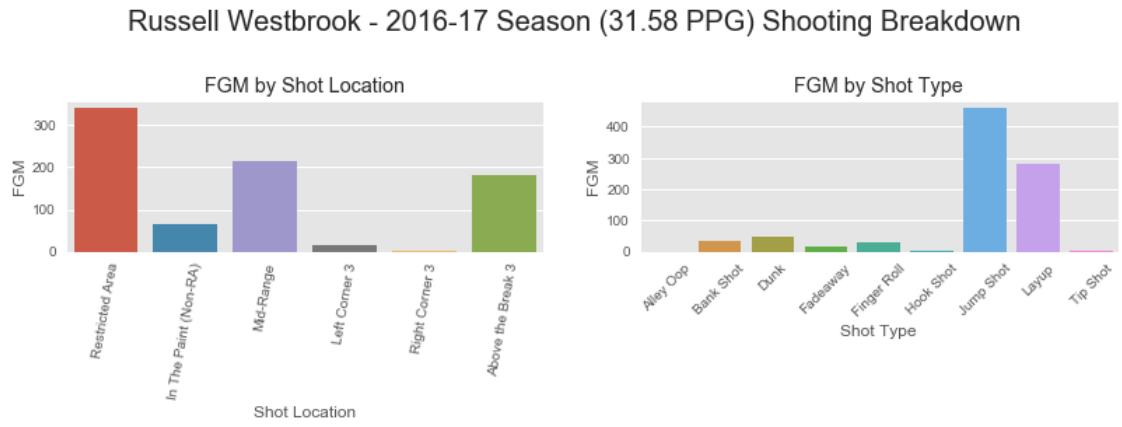




Stephen Curry - 2015-16 Season (30.06 PPG) Shooting Breakdown

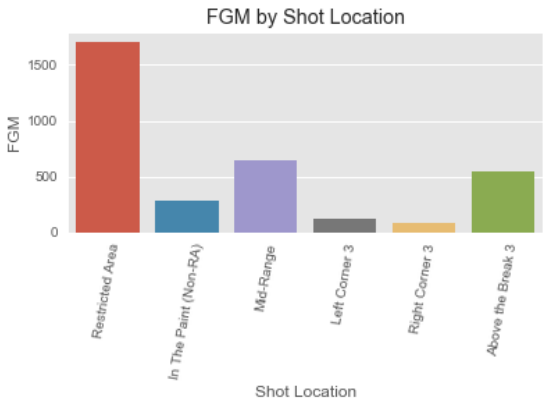


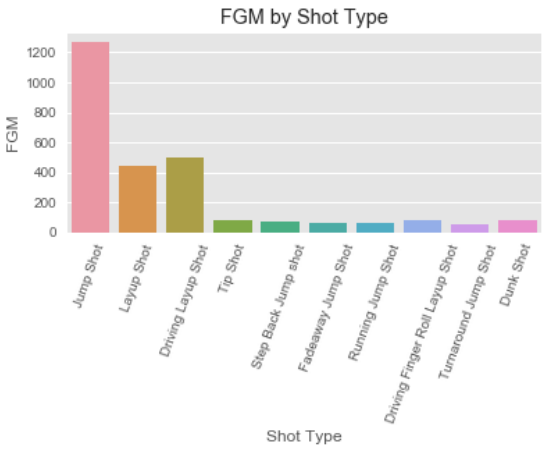




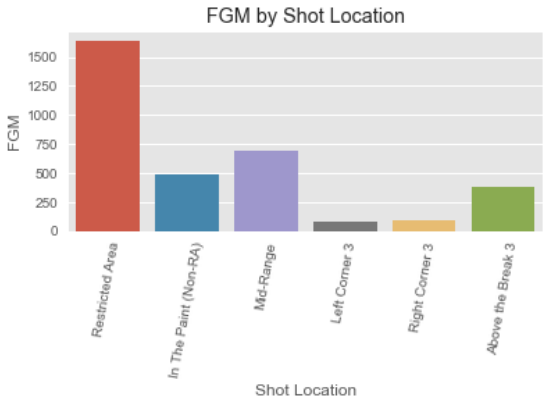

```
In [132]: from nba_py.team import TeamShootingSplits
          from nba_py.team import TeamList, TeamSummary
          from nba_py.team import TeamYearOverYearSplits
          all_teams = TeamList().info().head(30)

In [140]: top_scoring_teams = {}
          for i in range(2007,2017):
              season = str(i)+"-"+str(i+1)[2:]
              for index, row in all_teams.iterrows():
                  if TeamSummary(row['TEAM_ID'],season=season).season_ranks()['PTS_RANK'][0] ==
                      top_scoring_teams[season] = row['TEAM_ID']
                  time.sleep(2)

In [175]: for season in top_scoring_teams:
              fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12,5))
              team_abr = all_teams[all_teams['TEAM_ID'] == top_scoring_teams[season]].iloc[0][
              team_shots = TeamShootingSplits(top_scoring_teams[season], season=season)
              shot_areas = team_shots.shot_areas().drop(team_shots.shot_areas().index[6])
              shot_types = team_shots.shot_type_summary().sort_values(by=['FGA'], ascending=Fa
              sns.barplot(x='GROUP_VALUE', y='FGM', data=shot_areas, ax=ax1)
              ax1.set_xlabel('Shot Location')
              ax1.set_title('FGM by Shot Location')
              for tick in ax1.get_xticklabels():
                  tick.set_rotation(80)
              sns.barplot(x='GROUP_VALUE', y='FGM', data=shot_types, ax=ax2)
              plt.xticks(rotation=70)
              ax2.set_xlabel('Shot Type')
              ax2.set_title('FGM by Shot Type')
              fig.suptitle('%s - %s Season Shooting Breakdown' % (team_abr,season), fontsize=2
              plt.subplots_adjust(wspace=0.2, top=0.8, bottom=0.4)
              fig.savefig("%s_%s_scoring.png" % (team_abr,season))
              plt.show()
              time.sleep(3)
```

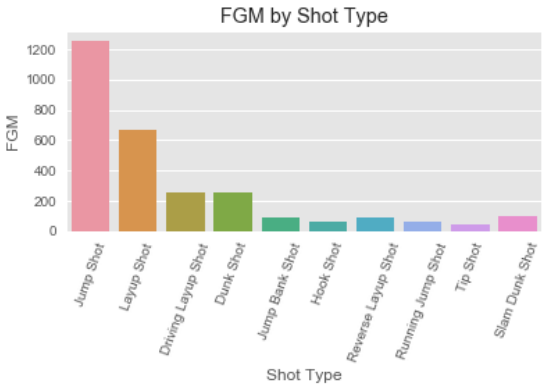
GSW - 2007-08 Season Shooting Breakdown



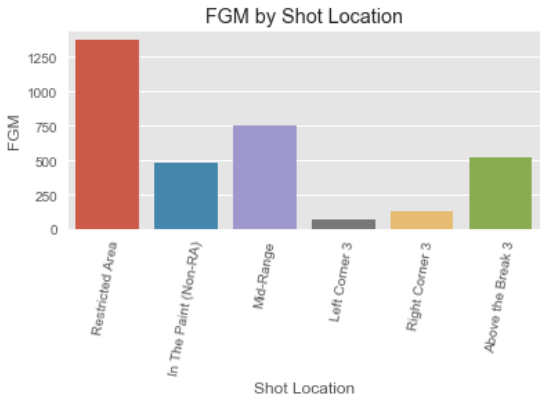


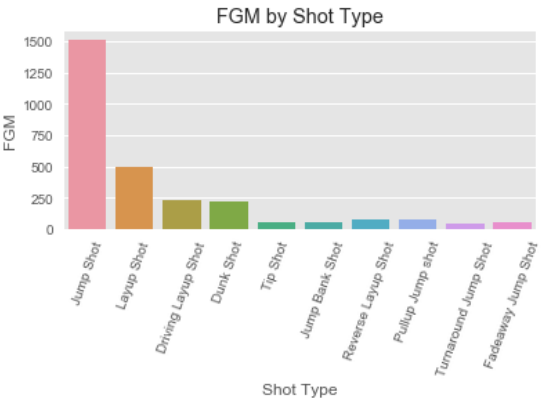
PHX - 2008-09 Season Shooting Breakdown



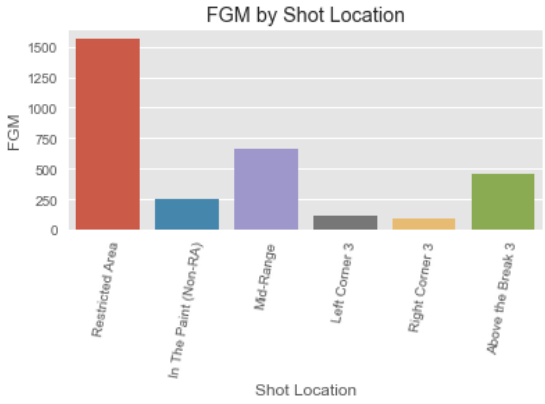


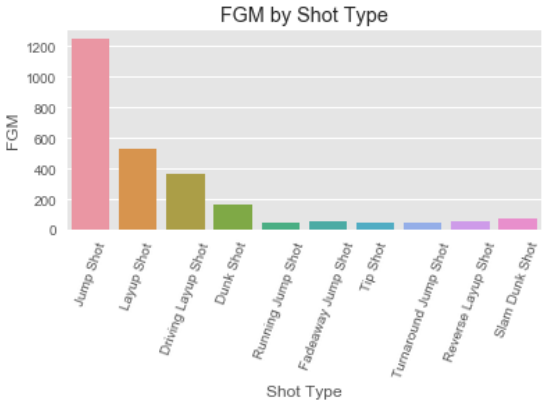
PHX - 2009-10 Season Shooting Breakdown



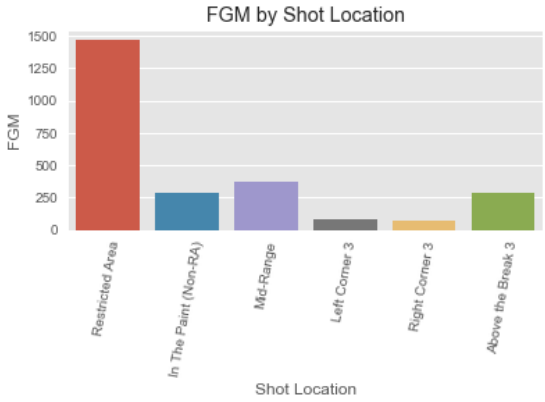


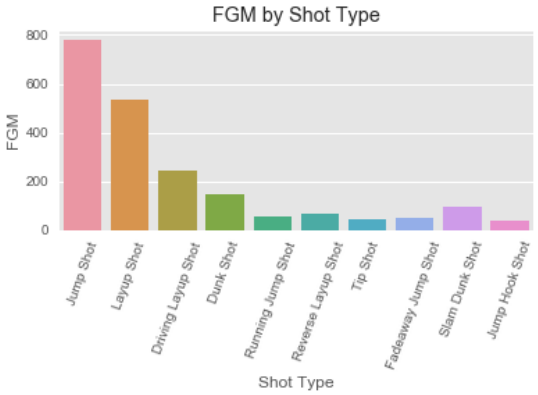
DEN - 2010-11 Season Shooting Breakdown



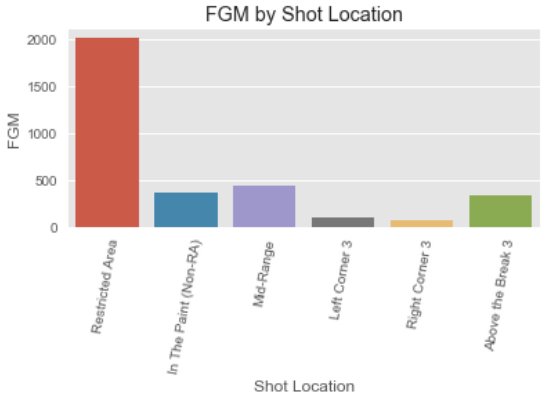


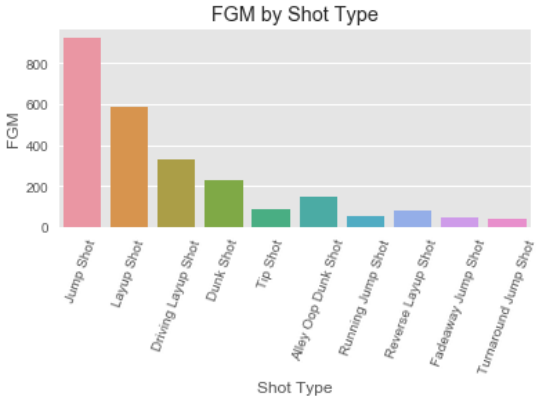
DEN - 2011-12 Season Shooting Breakdown



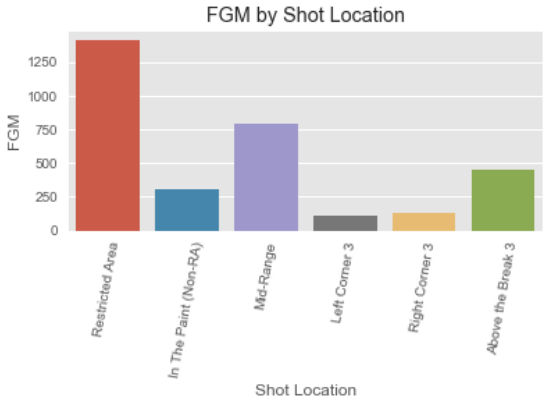


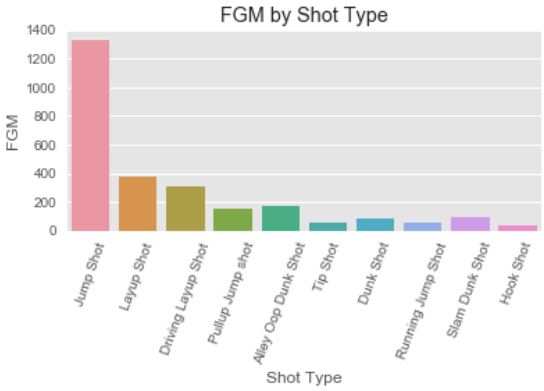
DEN - 2012-13 Season Shooting Breakdown



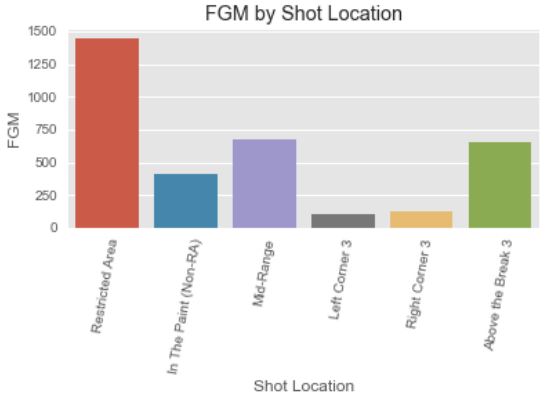


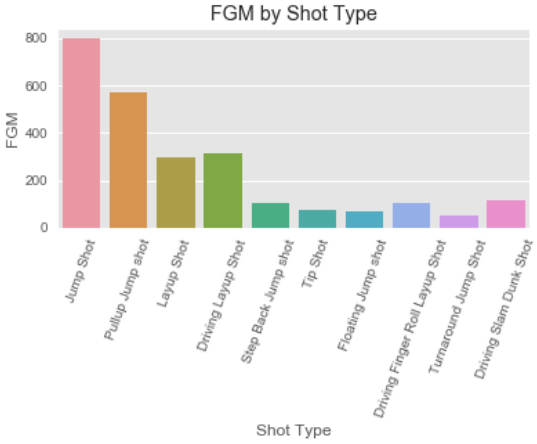
LAC - 2013-14 Season Shooting Breakdown

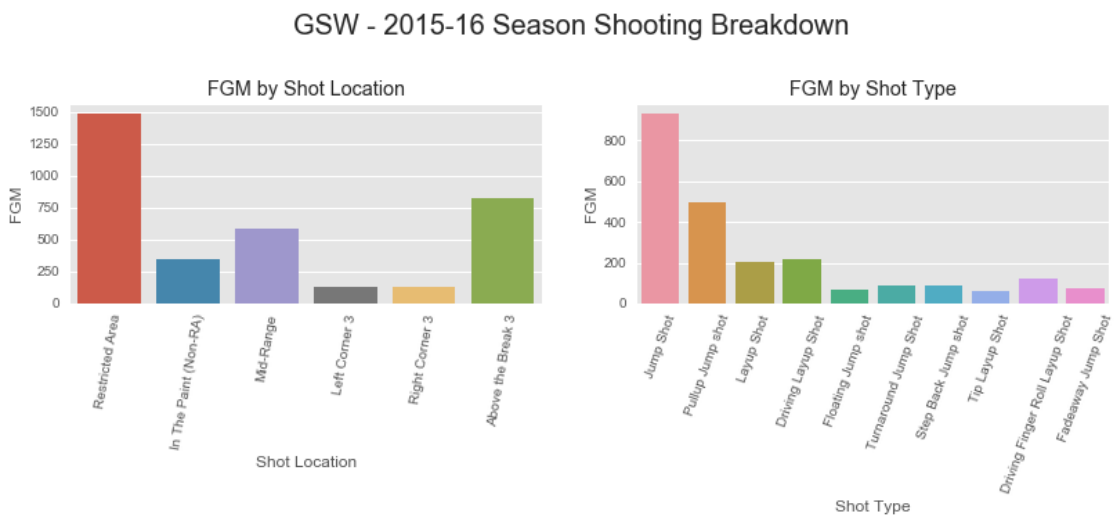


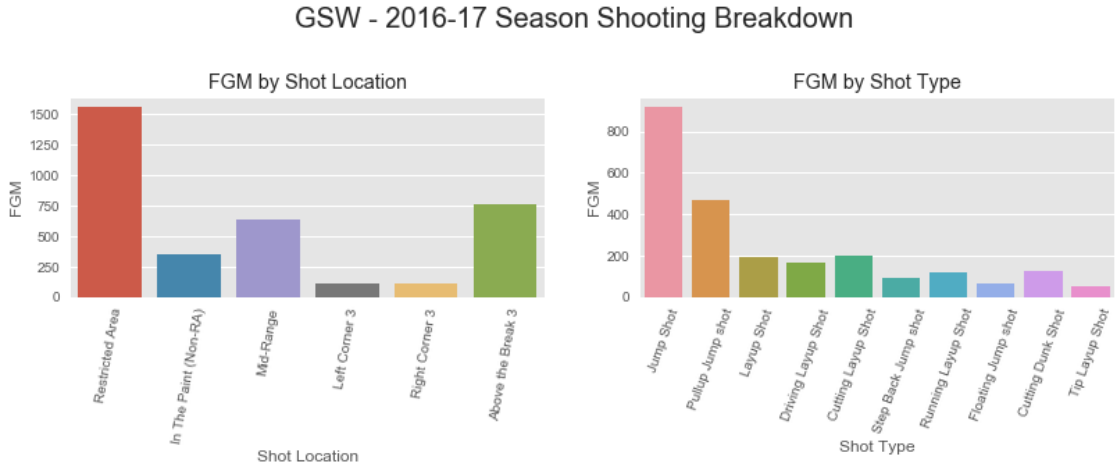


GSW - 2014-15 Season Shooting Breakdown










```
In [169]: shot_types.sort_values(by=['FGA'], ascending=False).head(10)
```

Out[169]:

	GROUP_SET	GROUP_VALUE	FGM	FGA	FG_PCT	FG3M	FG3A	FG3_PCT	\
24	Shot Type	Jump Shot	921	3038	0.303	594	1934	0.307	
27	Shot Type	Pullup Jump shot	465	763	0.609	269	433	0.621	
25	Shot Type	Layup Shot	193	491	0.393	0	0	0.000	
13	Shot Type	Driving Layup Shot	163	257	0.634	0	0	0.000	
4	Shot Type	Cutting Layup Shot	199	234	0.850	0	0	0.000	
41	Shot Type	Step Back Jump shot	92	165	0.558	45	79	0.570	
36	Shot Type	Running Layup Shot	116	161	0.720	0	0	0.000	
20	Shot Type	Floating Jump shot	63	134	0.470	0	1	0.000	
2	Shot Type	Cutting Dunk Shot	128	132	0.970	0	0	0.000	
43	Shot Type	Tip Layup Shot	49	128	0.383	0	0	0.000	

```
    EFG_PCT  BLKA         ...           EFG_PCT_RANK  BLKA_RANK  \
24    0.401    99         ...                     45         48
27    0.786     1         ...                     19         24
25    0.393   119         ...                     46         49
13    0.634    27         ...                     30         47
4     0.850     7         ...                     16         44
41    0.694     1         ...                     26         24
36    0.720     8         ...                     23         46
20    0.470     0         ...                     44          1
2     0.970     2         ...                      5         33
43    0.383     2         ...                     47         33

    PCT_AST_2PM_RANK  PCT_UAST_2PM_RANK  PCT_AST_3PM_RANK  PCT_UAST_3PM_RANK  \
24                13                 35                 4                  7
27                24                 24                 5                  6
25                14                 34                11                  8
13                36                 12                11                  8
4                  1                 40                11                  8
41                39                  9                 6                  5
36                23                 25                11                  8
20                29                 19                11                  8
2                  1                 40                11                  8
43                40                  1                11                  8

    PCT_AST_FGM_RANK  PCT_UAST_FGM_RANK  CFID            CFPARAMS
24                11                 37   174           Jump Shot
27                21                 27   174    Pullup Jump shot
25                14                 34   174          Layup Shot
13                37                 11   174  Driving Layup Shot
4                  1                 40   174  Cutting Layup Shot
41                33                 15   174 Step Back Jump shot
36                25                 23   174  Running Layup Shot
20                30                 18   174  Floating Jump shot
2                  1                 40   174   Cutting Dunk Shot
43                41                  1   174      Tip Layup Shot

[10 rows x 32 columns]
```

In [ ]: