Import Libraries

```
import numpy as np
import pandas as pd
import scipy.stats as stats

import plotly.express as px
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('ggplot')
plt.rcParams["figure.figsize"] = (15,5)

import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns',None)
```

Data Loading and Preliminary Check

```
advStats = pd.read_csv('NBA Advanced Cleaned Stats.csv')
stats = pd.read_csv('nba_stats_cleaned.csv')
merge = pd.merge(stats, advStats, on='Player-additional',
suffixes=('_trad', '_adv'))

stats.head()
```

```
   Rk            Player Pos  Age   Tm   G  GS    MP   FG   FGA    FG%  
3P  \
0   1  Precious Achiuwa   C   23  TOR  55  12  1140  196   404  0.485   
29   
1   2      Steven Adams   C   29  MEM  42  42  1133  157   263  0.597   
0   
2   3       Bam Adebayo   C   25  MIA  75  75  2598  602  1114  0.540   
1   
3   4      Ochai Agbaji  SG   22  UTA  59  22  1209  165   386  0.427   
81   
4   5      Santi Aldama  PF   22  MEM  77  20  1682  247   525  0.470   
94   

   3PA    3P%   2P   2PA    2P%   eFG%   FT  FTA    FT%  ORB  DRB  TRB  
AST  \
0  108  0.269  167   296  0.564  0.521   87  124  0.702  100  228  328   
50   
1    1  0.000  157   262  0.599  0.597   47  129  0.364  214  271  485   
97   
2   12  0.083  601  1102  0.545  0.541  324  402  0.806  184  504  688  
240   
3  228  0.355   84   158  0.532  0.532   56   69  0.812   43   78  121   
```

```
67
4  266  0.353  153   259  0.591  0.560  108  144  0.750   85  286  371
97

   STL  BLK  TOV   PF   PTS Player-additional
0   31   30   59  102   508         achiupr01
1   36   46   79   98   361         adamsst01
2   88   61  187  208  1529         adebaba01
3   16   15   41   99   467         agbajoc01
4   45   48   60  143   696         aldamsa01
```

```
advStats.head()
```

```
   Rk             Player Pos  Age   Tm   G    MP   PER    TS%   3PAr
FTr  \
0   1  Precious Achiuwa   C   23  TOR  55  1140  15.2  0.554  0.267
0.307
1   2      Steven Adams   C   29  MEM  42  1133  17.5  0.564  0.004
0.490
2   3       Bam Adebayo   C   25  MIA  75  2598  20.1  0.592  0.011
0.361
3   4      Ochai Agbaji  SG   22  UTA  59  1209   9.5  0.561  0.591
0.179
4   5      Santi Aldama  PF   22  MEM  77  1682  13.9  0.591  0.507
0.274

   ORB%  DRB%  TRB%  AST%  STL%  BLK%  TOV%  USG%  Unnamed: 19  OWS
DWS   WS  \
0   9.3  24.4  16.3   6.3   1.3   2.6  11.4  19.4          NaN  0.8
1.4  2.2
1  20.1  25.3  22.7  11.2   1.5   3.7  19.8  14.6          NaN  1.3
2.1  3.4
2   8.0  23.6  15.5  15.9   1.7   2.4  12.7  25.2          NaN  3.6
3.8  7.4
3   3.9   6.9   5.4   7.5   0.6   1.0   9.0  15.8          NaN  0.9
0.4  1.3
4   5.4  18.0  11.7   7.6   1.3   2.6   9.3  16.0          NaN  2.1
2.4  4.6

   WS/48  Unnamed: 24  OBPM  DBPM   BPM  VORP Player-additional
0  0.093          NaN  -1.4  -0.8 -2.3  -0.1         achiupr01
1  0.144          NaN  -0.3   0.9  0.6   0.7         adamsst01
2  0.137          NaN   0.8   0.8  1.5   2.3         adebaba01
3  0.053          NaN  -1.7  -1.4 -3.0  -0.3         agbajoc01
4  0.130          NaN  -0.3   0.8  0.5   1.1         aldamsa01
```

Advanced/Traditional Statistic Dataset Merge + Data Loading and Preliminary Check

```
merge = merge.drop(columns=['Unnamed: 19', 'Unnamed: 24'])

merge.head()
```

```
   Rk_trad       Player_trad Pos_trad  Age_trad Tm_trad  G_trad  GS
MP_trad  \
0        1  Precious Achiuwa        C        23     TOR      55  12
1140
1        2      Steven Adams        C        29     MEM      42  42
1133
2        3       Bam Adebayo        C        25     MIA      75  75
2598
3        4      Ochai Agbaji       SG        22     UTA      59  22
1209
4        5      Santi Aldama       PF        22     MEM      77  20
1682

    FG   FGA    FG%  3P  3PA    3P%   2P   2PA    2P%   eFG%   FT  FTA
FT%  \
0  196   404  0.485  29  108  0.269  167   296  0.564  0.521   87  124
0.702
1  157   263  0.597   0    1  0.000  157   262  0.599  0.597   47  129
0.364
2  602  1114  0.540   1   12  0.083  601  1102  0.545  0.541  324  402
0.806
3  165   386  0.427  81  228  0.355   84   158  0.532  0.532   56   69
0.812
4  247   525  0.470  94  266  0.353  153   259  0.591  0.560  108  144
0.750

   ORB  DRB  TRB  AST  STL  BLK  TOV   PF   PTS Player-additional
Rk_adv  \
0  100  228  328   50   31   30   59  102   508         achiupr01
1
1  214  271  485   97   36   46   79   98   361         adamsst01
2
2  184  504  688  240   88   61  187  208  1529         adebaba01
3
3   43   78  121   67   16   15   41   99   467         agbajoc01
4
4   85  286  371   97   45   48   60  143   696         aldamsa01
5

         Player_adv Pos_adv  Age_adv Tm_adv  G_adv  MP_adv   PER    TS
%  \
0  Precious Achiuwa       C       23    TOR     55    1140  15.2
0.554
```

```
1      Steven Adams       C       29    MEM     42    1133  17.5
0.564
2       Bam Adebayo       C       25    MIA     75    2598  20.1
0.592
3      Ochai Agbaji      SG       22    UTA     59    1209   9.5
0.561
4      Santi Aldama      PF       22    MEM     77    1682  13.9
0.591

    3PAr    FTr  ORB%  DRB%  TRB%  AST%  STL%  BLK%  TOV%  USG%  OWS
DWS  \
0  0.267  0.307   9.3  24.4  16.3   6.3   1.3   2.6  11.4  19.4  0.8
1.4
1  0.004  0.490  20.1  25.3  22.7  11.2   1.5   3.7  19.8  14.6  1.3
2.1
2  0.011  0.361   8.0  23.6  15.5  15.9   1.7   2.4  12.7  25.2  3.6
3.8
3  0.591  0.179   3.9   6.9   5.4   7.5   0.6   1.0   9.0  15.8  0.9
0.4
4  0.507  0.274   5.4  18.0  11.7   7.6   1.3   2.6   9.3  16.0  2.1
2.4

    WS  WS/48  OBPM  DBPM  BPM  VORP
0  2.2  0.093  -1.4  -0.8 -2.3  -0.1
1  3.4  0.144  -0.3   0.9  0.6   0.7
2  7.4  0.137   0.8   0.8  1.5   2.3
3  1.3  0.053  -1.7  -1.4 -3.0  -0.3
4  4.6  0.130  -0.3   0.8  0.5   1.1
```

```
sg = merge[merge['Pos_adv'] == 'SG']
sg.head()
```

```
    Rk_trad               Player_trad Pos_trad  Age_trad Tm_trad
G_trad  GS  \
3         4              Ochai Agbaji       SG        22     UTA
59  22
5         6  Nickeil Alexander-Walker       SG        24     MIN
59   3
6         7             Grayson Allen       SG        27     MIL
72  70
23       24              Desmond Bane       SG        24     MEM
58  58
28       29                RJ Barrett       SG        22     NYK
73  73

    MP_trad   FG   FGA    FG%   3P  3PA    3P%   2P  2PA    2P%   eFG%
FT  \
3      1209  165   386  0.427   81  228  0.355   84  158  0.532  0.532
56
5       884  131   295  0.444   61  159  0.384   70  136  0.515  0.547
```

```
40
6      1972  245   557  0.440  146  366  0.399   99  191  0.518  0.571
114
23     1841  450   939  0.479  166  407  0.408  284  532  0.534  0.568
181
28     2475  510  1176  0.434  121  390  0.310  389  786  0.495  0.485
290

    FTA    FT%  ORB  DRB  TRB  AST  STL  BLK  TOV   PF   PTS  \
3    69  0.812   43   78  121   67   16   15   41   99   467
5    60  0.667   15   86  101  108   32   21   55   88   363
6   126  0.905   61  176  237  163   62   14   72  117   750
23  205  0.883   43  248  291  254   56   22  126  150  1247
28  392  0.740   60  308  368  201   31   15  164  179  1431

   Player-additional  Rk_adv                Player_adv Pos_adv
Age_adv  \
3          agbajoc01       4              Ochai Agbaji      SG
22
5          alexani01       6  Nickeil Alexander-Walker      SG
24
6          allengr01       7             Grayson Allen      SG
27
23          banede01      24              Desmond Bane      SG
24
28         barrerj01      29                RJ Barrett      SG
22

   Tm_adv  G_adv  MP_adv   PER    TS%   3PAr    FTr  ORB%  DRB%  TRB%
AST%  \
3     UTA     59    1209   9.5  0.561  0.591  0.179   3.9   6.9   5.4
7.5
5     MIN     59     884  11.6  0.565  0.539  0.203   1.9  10.5   6.3
16.7
6     MIL     72    1972  12.3  0.612  0.657  0.226   3.4   9.0   6.3
10.9
23    MEM     58    1841  19.1  0.606  0.433  0.218   2.5  14.3   8.4
20.8
28    NYK     73    2475  12.9  0.531  0.332  0.333   2.7  13.7   8.2
12.4

    STL%  BLK%  TOV%  USG%  OWS  DWS   WS  WS/48  OBPM  DBPM  BPM
VORP
3    0.6   1.0   9.0  15.8  0.9  0.4  1.3  0.053  -1.7  -1.4 -3.0  -
0.3
5    1.7   2.0  14.6  17.9  0.3  0.8  1.1  0.062  -1.4   0.4 -0.9
0.2
6    1.5   0.6  10.5  14.6  2.8  2.2  5.1  0.123  -0.6   0.7  0.1
1.0
23   1.4   1.1  10.9  26.1  3.5  2.3  5.8  0.151   3.3   0.1  3.5
```

```
2.5
28   0.6   0.6  10.8  26.2  0.5  1.8  2.2  0.043  -1.2  -1.9 -3.1  -
0.7
```

```
sg.describe()
```

```
          Rk_trad    Age_trad      G_trad          GS      MP_trad  \
count  129.000000  129.000000  129.000000  129.000000   129.000000
mean   289.992248   24.604651   47.178295   20.263566  1088.480620
std    155.878583    3.841424   25.470402   26.081758   846.249687
min      4.000000   19.000000    1.000000    0.000000     2.000000
25%    152.000000   22.000000   25.000000    0.000000   268.000000
50%    314.000000   24.000000   53.000000    6.000000   993.000000
75%    423.000000   27.000000   70.000000   32.000000  1841.000000
max    534.000000   36.000000   82.000000   79.000000  2842.000000

               FG          FGA         FG%          3P         3PA  \
count  129.000000   129.000000  128.000000  129.000000  129.000000
mean   188.666667   420.496124    0.429914   71.294574  193.573643
std    177.675764   385.188390    0.076292   64.570262  170.953017
min      0.000000     0.000000    0.000000    0.000000    0.000000
25%     31.000000    78.000000    0.400000   14.000000   37.000000
50%    144.000000   333.000000    0.437000   59.000000  162.000000
75%    281.000000   625.000000    0.463250  114.000000  310.000000
max    707.000000  1541.000000    0.714000  245.000000  658.000000

              3P%          2P         2PA         2P%        eFG%
FT  \
count  127.000000  129.000000  129.000000  126.000000  128.000000
129.000000
mean     0.344843  117.372093  226.922481    0.503635    0.514844
77.813953
std      0.110662  123.025345  233.533711    0.112263    0.088688
90.540458
min      0.000000    0.000000    0.000000    0.000000    0.000000
0.000000
25%      0.324000   16.000000   37.000000    0.469500    0.481500
11.000000
50%      0.365000   83.000000  162.000000    0.506000    0.524500
45.000000
75%      0.390500  192.000000  355.000000    0.549500    0.559250
103.000000
max      1.000000  494.000000  963.000000    1.000000    0.857000
376.000000

              FTA         FT%         ORB         DRB         TRB
AST  \
count  129.000000  124.000000  129.000000  129.000000  129.000000
129.000000
mean    95.961240    0.792145   25.674419  109.201550  134.875969
```

```
114.581395
std    110.218136    0.127268   23.525971   90.611066  109.506208
112.734180
min      0.000000    0.250000    0.000000    0.000000    0.000000
0.000000
25%     15.000000    0.723750    7.000000   31.000000   37.000000
22.000000
50%     60.000000    0.810500   18.000000   87.000000  109.000000
78.000000
75%    126.000000    0.864750   42.000000  171.000000  210.000000
184.000000
max    463.000000    1.000000  109.000000  411.000000  458.000000
515.000000

              STL         BLK         TOV          PF          PTS  \
count  129.000000  129.000000  129.000000  129.000000   129.000000
mean    34.147287   12.031008   59.457364   80.666667   526.441860
std     29.792276   12.338923   56.328557   59.974517   496.238427
min      0.000000    0.000000    0.000000    0.000000     0.000000
25%      9.000000    3.000000   11.000000   23.000000    87.000000
50%     30.000000    8.000000   46.000000   78.000000   397.000000
75%     50.000000   17.000000   94.000000  120.000000   780.000000
max    126.000000   76.000000  259.000000  242.000000  1946.000000

           Rk_adv     Age_adv       G_adv       MP_adv         PER  \
count  129.000000  129.000000  129.000000   129.000000  129.000000
mean   289.992248   24.604651   47.178295  1088.480620   12.158915
std    155.878583    3.841424   25.470402   846.249687    6.744357
min      4.000000   19.000000    1.000000     2.000000  -20.900000
25%    152.000000   22.000000   25.000000   268.000000    9.500000
50%    314.000000   24.000000   53.000000   993.000000   11.800000
75%    423.000000   27.000000   70.000000  1841.000000   14.500000
max    534.000000   36.000000   82.000000  2842.000000   65.600000

              TS%        3PAr         FTr        ORB%        DRB%
TRB%  \
count  128.000000  128.000000  128.000000  129.000000  129.000000
129.000000
mean     0.548438    0.486852    0.225352    3.006977   11.961240
7.482171
std      0.072654    0.183291    0.210625    2.238763    5.343034
2.713694
min      0.337000    0.000000    0.000000    0.000000    0.000000
0.000000
25%      0.522000    0.375000    0.126750    1.700000    9.800000
6.100000
50%      0.562000    0.467500    0.196500    2.500000   11.300000
7.100000
75%      0.593250    0.586000    0.265000    3.900000   13.400000
8.400000
```

```
max      0.825000    1.000000    2.000000   12.700000   55.400000
27.400000

             AST%        STL%        BLK%        TOV%        USG%
OWS  \
count  129.000000  129.000000  129.000000  129.000000  129.000000
129.000000
mean    13.721705    1.737209    1.406977   12.149612   19.614729
0.842636
std      7.300717    2.163297    3.917632    9.394290    6.253150
1.399987
min      0.000000    0.000000    0.000000    0.000000   10.300000   -
1.900000
25%      8.700000    1.100000    0.500000    9.000000   15.200000
0.000000
50%     12.900000    1.400000    0.900000   11.000000   18.400000
0.300000
75%     18.000000    2.000000    1.300000   14.000000   22.800000
1.400000
max     35.700000   24.200000   44.100000  100.000000   52.500000
5.400000

              DWS          WS       WS/48        OBPM        DBPM
BPM  \
count  129.000000  129.000000  129.000000  129.000000  129.000000
129.000000
mean     0.909302    1.754264    0.058085   -1.690698   -0.305426   -
1.995349
std      0.868048    2.074737    0.096561    3.512910    3.318987
5.946688
min      0.000000   -1.600000   -0.517000  -21.300000   -6.100000  -
26.500000
25%      0.200000    0.100000    0.028000   -3.000000   -1.400000   -
3.500000
50%      0.600000    1.100000    0.067000   -1.400000   -0.500000   -
1.900000
75%      1.400000    2.800000    0.101000   -0.300000    0.200000
0.100000
max      3.600000    8.900000    0.626000   15.900000   32.700000
48.600000

             VORP
count  129.000000
mean     0.320930
std      0.925985
min     -1.300000
25%     -0.200000
50%      0.000000
75%      0.600000
max      5.000000
```

```
merge['Pos_trad'].value_counts()
```

```
SG       126
C        108
PF       102
SF       102
PG        91
SG-PG      2
SF-SG      2
PG-SG      2
SF-PF      1
PF-C       1
PF-SF      1
Name: Pos_trad, dtype: int64
```

```
merge['Tm_trad'].value_counts()
```

```
POR    22
DAL    21
WAS    21
SAS    21
BRK    20
SAC    20
TOR    19
UTA    19
DET    19
MEM    18
IND    18
ATL    18
LAC    18
PHI    18
MIL    18
MIN    18
CHO    17
GSW    17
PHO    17
ORL    17
CLE    17
CHI    17
MIA    17
BOS    17
LAL    17
NOP    16
DEN    16
OKC    16
HOU    15
NYK    14
Name: Tm_trad, dtype: int64
```

```
top =
['MIL','BOS','PHI','DEN','MEM','CLE','SAC','NYK','PHO','BRK','MIA','LA
C','GSW','LAL','MIN']
bot =
['NOP','ATL','TOR','CHI','OKC','DAL','UTA','IND','WAS','ORL','POR','CH
O','HOU','SAS','DET']

mask = sg['Tm_trad'].isin(top)
top_sg=sg[mask]
top_sg['Tm_trad'].value_counts()
```

```
BRK    7
MIN    6
PHO    6
PHI    6
NYK    5
LAL    5
GSW    5
SAC    4
LAC    4
CLE    4
MEM    3
DEN    3
MIL    2
BOS    2
MIA    2
Name: Tm_trad, dtype: int64
```

```
mask = sg['Tm_trad'].isin(bot)
bot_sg=sg[mask]
bot_sg['Tm_trad'].value_counts()
```

```
ATL    6
DET    6
DAL    6
POR    6
CHI    5
SAS    5
IND    5
UTA    4
CHO    4
ORL    4
OKC    4
WAS    3
NOP    3
TOR    2
HOU    2
Name: Tm_trad, dtype: int64
```

```
top_sg.describe()
```

```
          Rk_trad   Age_trad     G_trad         GS      MP_trad
FG  \
count   64.000000  64.000000  64.000000  64.000000    64.000000
64.000000
mean   277.187500  25.109375  49.468750  21.203125  1149.187500
197.359375
std    151.602328   3.625821  25.711444  27.076716   868.877544
177.367556
min      6.000000  19.000000   1.000000   0.000000     5.000000
0.000000
25%    141.500000  22.750000  26.500000   1.000000   323.500000
34.500000
50%    310.000000  24.500000  58.500000   6.500000  1143.500000
161.000000
75%    403.250000  27.250000  72.000000  36.000000  1911.750000
291.250000
max    525.000000  35.000000  82.000000  79.000000  2842.000000
707.000000

              FGA        FG%         3P         3PA        3P%
2P  \
count    64.00000  63.000000   64.000000   64.000000  63.000000
64.000000
mean    436.03125   0.437810   79.609375  210.046875   0.365016
117.750000
std     383.35536   0.054864   69.043951  179.907598   0.076544
118.259199
min       0.00000   0.250000    0.000000    0.000000   0.083000
0.000000
25%      81.75000   0.420000   15.500000   43.750000   0.352500
17.500000
50%     392.50000   0.442000   66.500000  197.500000   0.377000
90.000000
75%     651.50000   0.472000  133.500000  358.500000   0.398500
187.000000
max    1541.00000   0.594000  245.000000  658.000000   0.500000
494.000000

              2PA        2P%       eFG%          FT         FTA
FT%  \
count   64.000000  62.000000  63.000000   64.000000   64.000000
61.000000
mean   225.984375   0.506371   0.531778   83.265625  101.796875
0.809344
std    223.587240   0.119979   0.063447   89.977818  109.807216
0.116279
min      0.000000   0.000000   0.350000    0.000000    0.000000
0.500000
25%     41.750000   0.474250   0.502000   11.750000   15.750000
0.740000
```

```
50%    186.500000   0.513000   0.538000   47.500000   62.000000
0.817000
75%    333.250000   0.545000   0.568000  132.250000  173.250000
0.882000
max    963.000000   1.000000   0.688000  319.000000  422.000000
1.000000

              ORB         DRB         TRB         AST         STL
BLK  \
count   64.000000   64.000000   64.000000   64.000000   64.000000
64.000000
mean    25.343750  118.250000  143.593750  120.921875   35.906250
12.859375
std     24.150467   97.652282  117.011154  113.192953   31.310386
13.903142
min      0.000000    1.000000    1.000000    0.000000    0.000000
0.000000
25%      5.750000   32.750000   36.750000   21.750000    8.750000
3.750000
50%     16.000000  103.500000  118.500000   94.000000   31.000000
8.500000
75%     39.750000  181.500000  237.750000  198.000000   51.000000
18.250000
max    109.000000  411.000000  458.000000  515.000000  126.000000
76.000000

              TOV          PF          PTS      Rk_adv     Age_adv
G_adv  \
count   64.000000   64.000000    64.000000   64.000000   64.000000
64.000000
mean    60.921875   84.453125   557.593750  277.187500   25.109375
49.468750
std     54.574343   60.673974   500.137179  151.602328    3.625821
25.711444
min      0.000000    0.000000     0.000000    6.000000   19.000000
1.000000
25%     12.750000   23.000000    89.000000  141.500000   22.750000
26.500000
50%     50.500000   85.500000   485.500000  310.000000   24.500000
58.500000
75%     95.500000  126.750000   838.750000  403.250000   27.250000
72.000000
max    259.000000  191.000000  1946.000000  525.000000   35.000000
82.000000

            MP_adv         PER        TS%        3PAr         FTr
ORB%  \
count    64.000000   64.000000  63.000000   63.000000   63.000000
64.000000
mean   1149.187500   11.912500   0.561810    0.509460    0.221079
```

```
2.728125
std     868.877544   5.648669   0.066541   0.168967   0.168625
2.039099
min       5.000000 -20.900000   0.367000   0.143000   0.000000
0.000000
25%     323.500000  10.275000   0.534000   0.386500   0.124500
1.400000
50%    1143.500000  12.350000   0.567000   0.500000   0.203000
2.150000
75%    1911.750000  14.750000   0.604500   0.607000   0.263500
3.400000
max    2842.000000  22.900000   0.701000   1.000000   1.200000
8.800000

            DRB%       TRB%       AST%       STL%       BLK%
TOV%  \
count  64.000000  64.000000  64.000000  64.000000  64.000000
64.000000
mean   12.015625   7.407813  14.226562   1.585938   1.054688
12.995312
std     3.787960   2.138261   7.157109   0.839665   0.901309
12.125605
min     4.500000   3.100000   0.000000   0.000000   0.000000
0.000000
25%     9.875000   5.975000   9.375000   1.100000   0.500000
9.850000
50%    11.300000   7.100000  13.000000   1.500000   0.900000
10.900000
75%    13.725000   8.625000  18.325000   2.100000   1.200000
13.525000
max    27.600000  14.400000  35.700000   3.900000   5.200000
100.000000

            USG%        OWS        DWS         WS      WS/48
OBPM  \
count  64.000000  64.000000  64.000000  64.000000  64.000000
64.000000
mean   19.506250   1.120312   1.115625   2.229687   0.065562  -
1.626562
std     5.385338   1.530554   0.955721   2.259841   0.094470
3.593656
min    10.300000  -0.900000   0.000000  -0.500000  -0.517000 -
21.300000
25%    15.675000   0.000000   0.300000   0.200000   0.048250  -
2.750000
50%    18.450000   0.500000   0.900000   1.350000   0.077500  -
1.200000
75%    22.100000   1.925000   1.625000   3.900000   0.111000  -
0.250000
max    32.500000   5.400000   3.600000   8.900000   0.176000
```

```
5.600000

            DBPM        BPM       VORP
count  64.000000  64.000000  64.000000
mean   -0.292188  -1.915625   0.523438
std     1.500403   4.429956   0.991990
min    -5.600000 -26.500000  -0.700000
25%    -0.925000  -2.725000   0.000000
50%    -0.150000  -1.400000   0.100000
75%     0.425000   0.500000   1.000000
max     2.900000   6.300000   5.000000
```

```
bot_sg.describe()
```

```
         Rk_trad   Age_trad     G_trad         GS      MP_trad
FG  \
count   65.00000  65.000000  65.000000  65.000000    65.000000
65.000000
mean   302.60000  24.107692  44.923077  19.338462  1028.707692
180.107692
std    160.14591   4.008285  25.225426  25.240268   825.718470
178.940469
min      4.00000  19.000000   1.000000   0.000000     2.000000
0.000000
25%    188.00000  21.000000  25.000000   0.000000   232.000000
31.000000
50%    314.00000  23.000000  48.000000   5.000000   844.000000
138.000000
75%    445.00000  27.000000  66.000000  32.000000  1673.000000
269.000000
max    534.00000  36.000000  80.000000  77.000000  2768.000000
673.000000

               FGA        FG%          3P         3PA        3P%
2P  \
count    65.000000  65.000000   65.000000   65.000000  64.000000
65.000000
mean    405.200000   0.422262   63.107692  177.353846   0.324984
117.000000
std     389.353519   0.092267   59.235526  161.401037   0.133860
128.465827
min       1.000000   0.000000    0.000000    0.000000   0.000000
0.000000
25%      72.000000   0.393000   11.000000   27.000000   0.296500
12.000000
50%     307.000000   0.430000   58.000000  156.000000   0.350000
63.000000
75%     611.000000   0.447000   99.000000  263.000000   0.375000
192.000000
max    1388.000000   0.714000  212.000000  562.000000   1.000000
```

```
479.000000

              2PA        2P%       eFG%          FT         FTA
FT%  \
count   65.000000  64.000000  65.000000   65.000000   65.000000
63.000000
mean   227.846154   0.500984   0.498431   72.446154   90.215385
0.775492
std    244.671414   0.105135   0.105605   91.469433  111.172233
0.135925
min      0.000000   0.000000   0.000000    0.000000    0.000000
0.250000
25%     26.000000   0.467250   0.460000   11.000000   13.000000
0.723000
50%    122.000000   0.500000   0.514000   45.000000   56.000000
0.800000
75%    356.000000   0.552250   0.548000   78.000000  100.000000
0.840500
max    932.000000   0.750000   0.857000  376.000000  463.000000
1.000000

             ORB         DRB         TRB         AST         STL
BLK  \
count  65.000000   65.000000   65.000000   65.000000   65.000000
65.000000
mean   26.000000  100.292308  126.292308  108.338462   32.415385
11.215385
std    23.077993   82.893140  101.758096  112.808504   28.354834
10.623518
min     0.000000    0.000000    0.000000    0.000000    0.000000
0.000000
25%     8.000000   31.000000   37.000000   22.000000    9.000000
2.000000
50%    18.000000   77.000000   98.000000   65.000000   27.000000
8.000000
75%    42.000000  153.000000  196.000000  155.000000   48.000000
16.000000
max    87.000000  336.000000  389.000000  448.000000  112.000000
45.000000

              TOV         PF         PTS      Rk_adv    Age_adv
G_adv  \
count   65.000000  65.000000   65.000000   65.00000  65.000000
65.000000
mean    58.015385  76.938462  495.769231  302.60000  24.107692
44.923077
std     58.392929  59.511521  494.318115  160.14591   4.008285
25.225426
min      0.000000   0.000000    2.000000    4.00000  19.000000
1.000000
```

```
25%     10.000000   23.000000    85.000000  188.00000  21.000000
25.000000
50%     43.000000   65.000000   372.000000  314.00000  23.000000
48.000000
75%     83.000000  117.000000   703.000000  445.00000  27.000000
66.000000
max    237.000000  242.000000  1913.000000  534.00000  36.000000
80.000000

            MP_adv        PER        TS%       3PAr        FTr
ORB%  \
count    65.000000  65.000000  65.000000  65.000000  65.000000
65.000000
mean   1028.707692  12.401538   0.535477   0.464938   0.229492
3.281538
std     825.718470   7.709947   0.076413   0.194981   0.245868
2.403506
min       2.000000   3.500000   0.337000   0.000000   0.000000
0.000000
25%     232.000000   9.200000   0.512000   0.336000   0.129000
1.800000
50%     844.000000  11.500000   0.540000   0.447000   0.182000
2.700000
75%    1673.000000  14.500000   0.572000   0.568000   0.268000
4.000000
max    2768.000000  65.600000   0.825000   1.000000   2.000000
12.700000

            DRB%       TRB%       AST%       STL%       BLK%       TOV
%  \
count  65.000000  65.000000  65.000000  65.000000  65.000000
65.000000
mean   11.907692   7.555385  13.224615   1.886154   1.753846
11.316923
std     6.554824   3.196337   7.461318   2.936030   5.445155
5.509355
min     0.000000   0.000000   0.000000   0.000000   0.000000
0.000000
25%     9.700000   6.200000   7.600000   1.100000   0.600000
8.300000
50%    11.500000   7.100000  12.800000   1.400000   1.000000
11.100000
75%    13.000000   7.900000  18.000000   1.800000   1.400000
14.000000
max    55.400000  27.400000  34.900000  24.200000  44.100000
27.600000

            USG%        OWS        DWS         WS      WS/48
OBPM  \
count  65.000000  65.000000  65.000000  65.000000  65.000000
```

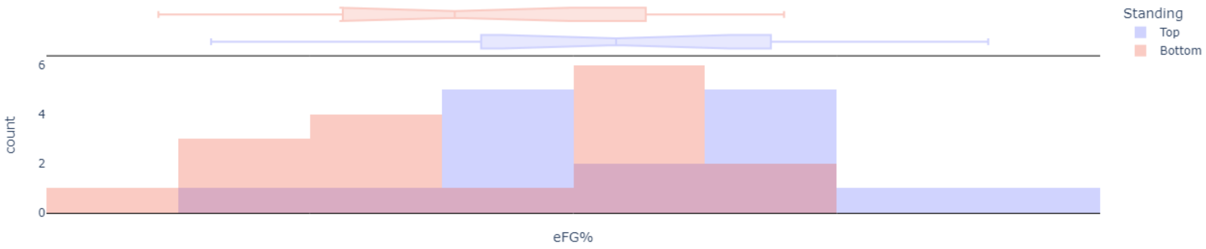
```
65.000000
mean    19.721538   0.569231   0.706154   1.286154   0.050723  -
1.753846
std      7.044978   1.208683   0.723463   1.770141   0.098752
3.458372
min     10.400000  -1.900000   0.000000  -1.600000  -0.225000  -
9.100000
25%     15.000000  -0.200000   0.200000   0.000000   0.008000  -
3.000000
50%     18.400000   0.200000   0.400000   0.700000   0.053000  -
1.800000
75%     23.700000   0.900000   1.100000   2.100000   0.088000  -
0.400000
max     52.500000   5.400000   2.900000   7.400000   0.626000
15.900000

            DBPM        BPM       VORP
count  65.000000  65.000000  65.000000
mean   -0.318462  -2.073846   0.121538
std     4.451400   7.169080   0.815378
min    -6.100000 -15.200000  -1.300000
25%    -1.700000  -4.300000  -0.200000
50%    -0.700000  -2.400000   0.000000
75%     0.000000  -0.700000   0.300000
max    32.700000  48.600000   3.500000
```

```
sg = sg.reset_index(drop=True)
sg.head()
```

```
   Rk_trad               Player_trad Pos_trad  Age_trad Tm_trad
G_trad  GS  \
0        4              Ochai Agbaji       SG        22     UTA
59  22
1        6  Nickeil Alexander-Walker       SG        24     MIN
59   3
2        7             Grayson Allen       SG        27     MIL
72  70
3       24              Desmond Bane       SG        24     MEM
58  58
4       29                RJ Barrett       SG        22     NYK
73  73

   MP_trad   FG   FGA    FG%   3P  3PA    3P%   2P  2PA    2P%   eFG%
FT  \
0     1209  165   386  0.427   81  228  0.355   84  158  0.532  0.532
56
1      884  131   295  0.444   61  159  0.384   70  136  0.515  0.547
40
2     1972  245   557  0.440  146  366  0.399   99  191  0.518  0.571
114
```

```
3     1841  450   939  0.479  166  407  0.408  284  532  0.534  0.568
181
4     2475  510  1176  0.434  121  390  0.310  389  786  0.495  0.485
290

   FTA    FT%  ORB  DRB  TRB  AST  STL  BLK  TOV   PF   PTS Player-
additional  \
0   69  0.812   43   78  121   67   16   15   41   99   467
agbajoc01
1   60  0.667   15   86  101  108   32   21   55   88   363
alexani01
2  126  0.905   61  176  237  163   62   14   72  117   750
allengr01
3  205  0.883   43  248  291  254   56   22  126  150  1247
banede01
4  392  0.740   60  308  368  201   31   15  164  179  1431
barrerj01

   Rk_adv                Player_adv Pos_adv  Age_adv Tm_adv  G_adv
MP_adv  \
0      4              Ochai Agbaji      SG       22    UTA     59
1209
1      6  Nickeil Alexander-Walker      SG       24    MIN     59
884
2      7             Grayson Allen      SG       27    MIL     72
1972
3     24              Desmond Bane      SG       24    MEM     58
1841
4     29                RJ Barrett      SG       22    NYK     73
2475

    PER    TS%   3PAr    FTr  ORB%  DRB%  TRB%  AST%  STL%  BLK%  TOV%
USG%  \
0   9.5  0.561  0.591  0.179   3.9   6.9   5.4   7.5   0.6   1.0   9.0
15.8
1  11.6  0.565  0.539  0.203   1.9  10.5   6.3  16.7   1.7   2.0  14.6
17.9
2  12.3  0.612  0.657  0.226   3.4   9.0   6.3  10.9   1.5   0.6  10.5
14.6
3  19.1  0.606  0.433  0.218   2.5  14.3   8.4  20.8   1.4   1.1  10.9
26.1
4  12.9  0.531  0.332  0.333   2.7  13.7   8.2  12.4   0.6   0.6  10.8
26.2

   OWS  DWS   WS  WS/48  OBPM  DBPM  BPM  VORP
0  0.9  0.4  1.3  0.053  -1.7  -1.4 -3.0  -0.3
1  0.3  0.8  1.1  0.062  -1.4   0.4 -0.9   0.2
2  2.8  2.2  5.1  0.123  -0.6   0.7  0.1   1.0
3  3.5  2.3  5.8  0.151   3.3   0.1  3.5   2.5
4  0.5  1.8  2.2  0.043  -1.2  -1.9 -3.1  -0.7
```

```python
sg['Standing']=0
for i in range(len(sg)):
    if sg.loc[i,'Tm_trad'] in top:
        sg.loc[i,'Standing']='Top'
    else:
        sg.loc[i,'Standing']='Bottom'

sg2 = sg[((sg['MP_trad']>=1673.000000)&(sg['Standing']=='Bottom'))|
               ((sg['MP_trad']>=1911.750000)&(sg['Standing']=='Top'))]

fig = px.histogram(sg2, x='eFG%', color='Standing',
                   marginal="box") # or violin, rug
fig.update_traces(opacity=0.3)
fig.update_layout(barmode='overlay',plot_bgcolor='rgba(0, 0, 0, 0)')
fig.update_xaxes(dtick=50,ticks='outside',showline=True,linecolor='bla
ck',mirror=False)
fig.show()
```



T-Tests (If error re-run import libraries)

```python
bfg=sg2[sg2['Standing']=='Bottom']['FG']
tfg=sg2[sg2['Standing']=='Top']['FG']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.32723259916851116,
pvalue=0.7456914916053919)
```

```python
bfg=sg2[sg2['Standing']=='Bottom']['FGA']
tfg=sg2[sg2['Standing']=='Top']['FGA']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.260794572745127,
pvalue=0.7959756224482695)
```

```python
bfg=sg2[sg2['Standing']=='Bottom']['FG%']
tfg=sg2[sg2['Standing']=='Top']['FG%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.4262799049309876,
pvalue=0.6728494444758419)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['3P']
tfg=sg2[sg2['Standing']=='Top']['3P']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-2.51763371590298,
pvalue=0.017195226061534275)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['3PA']
tfg=sg2[sg2['Standing']=='Top']['3PA']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-1.797539926986465,
pvalue=0.08199765288946745)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['3P%']
tfg=sg2[sg2['Standing']=='Top']['3P%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-3.71881195385597,
pvalue=0.0007927949257008825)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['2P']
tfg=sg2[sg2['Standing']=='Top']['2P']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=1.3726933973663757,
pvalue=0.17969763584889678)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['2PA']
tfg=sg2[sg2['Standing']=='Top']['2PA']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=1.3252722612072785,
pvalue=0.19476603714618115)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['2P%']
tfg=sg2[sg2['Standing']=='Top']['2P%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.12685251301160952,
pvalue=0.8998763744301205)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['eFG%']
tfg=sg2[sg2['Standing']=='Top']['eFG%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-2.0056002349755193,
pvalue=0.053694629397961637)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['FT']
tfg=sg2[sg2['Standing']=='Top']['FT']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.44253808566157277,
pvalue=0.661172355584941)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['FTA']
tfg=sg2[sg2['Standing']=='Top']['FTA']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.37519789682890764,
pvalue=0.7100682335646804)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['FT%']
tfg=sg2[sg2['Standing']=='Top']['FT%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.1323945952165199,
pvalue=0.8955278145699475)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['ORB']
tfg=sg2[sg2['Standing']=='Top']['ORB']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.8470822459520901,
pvalue=0.40344174297495095)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['DRB']
tfg=sg2[sg2['Standing']=='Top']['DRB']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-1.17830523736527, pvalue=0.247641020872919)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['TRB']
tfg=sg2[sg2['Standing']=='Top']['TRB']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.8319906222934168,
pvalue=0.4117776980112412)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['AST']
tfg=sg2[sg2['Standing']=='Top']['AST']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.31339831154786435,
pvalue=0.7560770885858216)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['STL']
tfg=sg2[sg2['Standing']=='Top']['STL']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.8560111618115336,
pvalue=0.3985600681969622)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['BLK']
tfg=sg2[sg2['Standing']=='Top']['BLK']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-1.1149727501842366,
pvalue=0.27343153553290117)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['TOV']
tfg=sg2[sg2['Standing']=='Top']['TOV']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.8247250018597946,
pvalue=0.41582886754439574)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['PF']
tfg=sg2[sg2['Standing']=='Top']['PF']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.2916752878858614,
pvalue=0.7724778718248438)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['PTS']
tfg=sg2[sg2['Standing']=='Top']['PTS']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.06024034039970528,
pvalue=0.9523509061333495)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['PER']
tfg=sg2[sg2['Standing']=='Top']['PER']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.024079782330597294,
pvalue=0.9809433074104141)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['TS%']
tfg=sg2[sg2['Standing']=='Top']['TS%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-1.839768948163441,
pvalue=0.07539559986829392)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['3PAr']
tfg=sg2[sg2['Standing']=='Top']['3PAr']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-2.705107385550637,
pvalue=0.010993508802570147)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['FTr']
tfg=sg2[sg2['Standing']=='Top']['FTr']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.20659593681616964,
pvalue=0.837676405072489)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['ORB%']
tfg=sg2[sg2['Standing']=='Top']['ORB%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=1.02022779646352, pvalue=0.3155168321771063)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['DRB%']
tfg=sg2[sg2['Standing']=='Top']['DRB%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.8754418936352838,
pvalue=0.38806641285533483)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['TRB%']
tfg=sg2[sg2['Standing']=='Top']['TRB%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.4662870607080138,
pvalue=0.6442701568258136)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['AST%']
tfg=sg2[sg2['Standing']=='Top']['AST%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=1.037182882424567,
pvalue=0.30767364885699955)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['STL%']
tfg=sg2[sg2['Standing']=='Top']['STL%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.6463973495632801,
pvalue=0.5227792195645539)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['BLK%']
tfg=sg2[sg2['Standing']=='Top']['BLK%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.8191052663264906,
pvalue=0.4189791949477325)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['TOV%']
tfg=sg2[sg2['Standing']=='Top']['TOV%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=1.315581675491667,
pvalue=0.19796176184284361)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['USG%']
tfg=sg2[sg2['Standing']=='Top']['USG%']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=0.9355060775928583,
pvalue=0.3567583967997665)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['OWS']
tfg=sg2[sg2['Standing']=='Top']['OWS']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-1.4496023325489182,
pvalue=0.15721162321624801)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['DWS']
tfg=sg2[sg2['Standing']=='Top']['DWS']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-2.704644147127477,
pvalue=0.011005864957710589)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['WS']
tfg=sg2[sg2['Standing']=='Top']['WS']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-2.242816483014249,
pvalue=0.032198962725666164)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['WS/48']
tfg=sg2[sg2['Standing']=='Top']['WS/48']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-2.0965128747823116,
pvalue=0.04429232324502591)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['OBPM']
tfg=sg2[sg2['Standing']=='Top']['OBPM']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-0.6539517633081748,
pvalue=0.5179650249311627)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['DBPM']
tfg=sg2[sg2['Standing']=='Top']['DBPM']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-1.9711540989341156,
pvalue=0.05768986642964039)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['BPM']
tfg=sg2[sg2['Standing']=='Top']['BPM']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-1.6778840485222701,
pvalue=0.10342530910977865)
```

```
bfg=sg2[sg2['Standing']=='Bottom']['VORP']
tfg=sg2[sg2['Standing']=='Top']['VORP']
stats.ttest_ind(a=bfg, b=tfg,equal_var=True)
```

```
Ttest_indResult(statistic=-1.664735765018737,
pvalue=0.1060423766218895)
```

```
column_name = sg.columns.tolist()
print(column_name)
```

```
['Rk_trad', 'Player_trad', 'Pos_trad', 'Age_trad', 'Tm_trad',
'G_trad', 'GS', 'MP_trad', 'FG', 'FGA', 'FG%', '3P', '3PA', '3P%',
'2P', '2PA', '2P%', 'eFG%', 'FT', 'FTA', 'FT%', 'ORB', 'DRB', 'TRB',
'AST', 'STL', 'BLK', 'TOV', 'PF', 'PTS', 'Player-additional',
'Rk_adv', 'Player_adv', 'Pos_adv', 'Age_adv', 'Tm_adv', 'G_adv',
'MP_adv', 'PER', 'TS%', '3PAr', 'FTr', 'ORB%', 'DRB%', 'TRB%', 'AST%',
'STL%', 'BLK%', 'TOV%', 'USG%', 'OWS', 'DWS', 'WS', 'WS/48', 'OBPM',
'DBPM', 'BPM', 'VORP', 'Standing']
```

```
stats_list = ['FG', 'FGA', 'FG%', '3P', '3PA', '3P%', '2P', '2PA', '2P
%', 'eFG%', 'FT', 'FTA', 'FT%', 'ORB', 'DRB', 'TRB', 'AST', 'STL',
'BLK', 'TOV', 'PF', 'PTS',
              'PER', 'TS%', '3PAr', 'FTr', 'ORB%', 'DRB%', 'TRB%',
'AST%', 'STL%', 'BLK%', 'TOV%', 'USG%', 'OWS', 'DWS', 'WS', 'WS/48',
'OBPM', 'DBPM', 'BPM', 'VORP']

test_results = []

for stat in stats_list:
    bfg = sg2[sg2['Standing'] == 'Bottom'][stat].dropna()  # Drop NaN
values for the test
    tfg = sg2[sg2['Standing'] == 'Top'][stat].dropna()     # Drop NaN
values for the test
    t_stat, p_val = stats.ttest_ind(bfg, tfg, equal_var=True)

    test_results.append({'Stat': stat, 'T-Statistic': t_stat, 'P-
Value': p_val, 'Reject Null': p_val < 0.05})

results_df = pd.DataFrame(test_results)

results_df
```

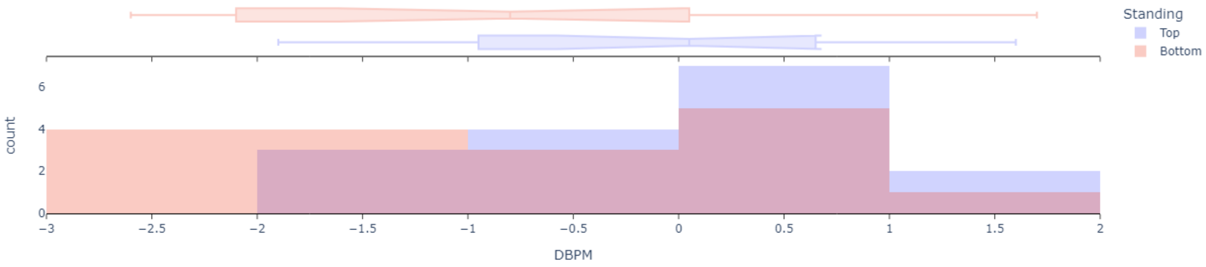
```
     Stat  T-Statistic   P-Value  Reject Null
0      FG     0.327233  0.745691        False
1     FGA     0.260795  0.795976        False
2     FG%     0.426280  0.672849        False
3      3P    -2.517634  0.017195         True
4     3PA    -1.797540  0.081998        False
```

```
5     3P%  -3.718812  0.000793         True
6      2P   1.372693  0.179698        False
7     2PA   1.325272  0.194766        False
8     2P%   0.126853  0.899876        False
9    eFG%  -2.005600  0.053695        False
10     FT   0.442538  0.661172        False
11    FTA   0.375198  0.710068        False
12    FT%   0.132395  0.895528        False
13    ORB   0.847082  0.403442        False
14    DRB  -1.178305  0.247641        False
15    TRB  -0.831991  0.411778        False
16    AST   0.313398  0.756077        False
17    STL  -0.856011  0.398560        False
18    BLK  -1.114973  0.273432        False
19    TOV   0.824725  0.415829        False
20     PF  -0.291675  0.772478        False
21    PTS   0.060240  0.952351        False
22    PER  -0.024080  0.980943        False
23    TS%  -1.839769  0.075396        False
24   3PAr  -2.705107  0.010994         True
25    FTr   0.206596  0.837676        False
26   ORB%   1.020228  0.315517        False
27   DRB%  -0.875442  0.388066        False
28   TRB%  -0.466287  0.644270        False
29   AST%   1.037183  0.307674        False
30   STL%  -0.646397  0.522779        False
31   BLK%  -0.819105  0.418979        False
32   TOV%   1.315582  0.197962        False
33   USG%   0.935506  0.356758        False
34    OWS  -1.449602  0.157212        False
35    DWS  -2.704644  0.011006         True
36     WS  -2.242816  0.032199         True
37  WS/48  -2.096513  0.044292         True
38   OBPM  -0.653952  0.517965        False
39   DBPM  -1.971154  0.057690        False
40    BPM  -1.677884  0.103425        False
41   VORP  -1.664736  0.106042        False
```

```python
rejected_null_df = results_df[results_df['Reject Null']]

rejected_null_df
```

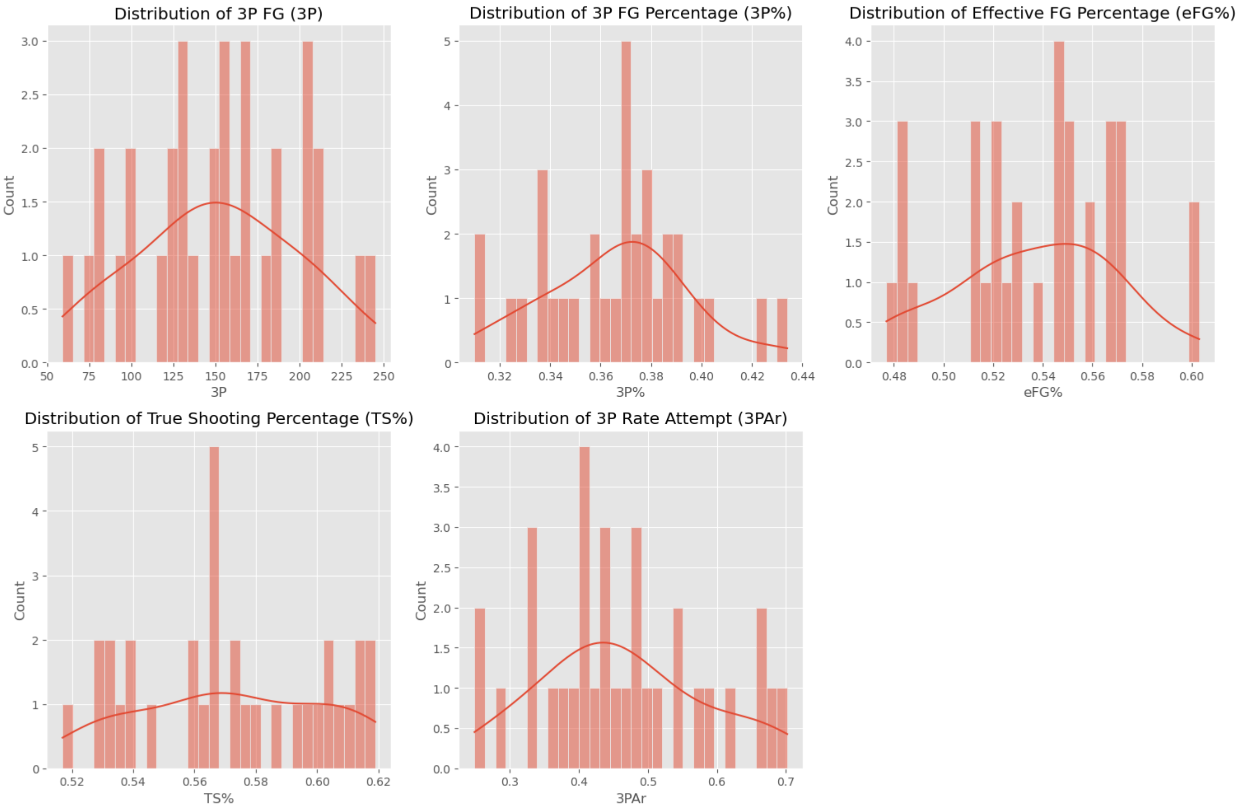
```
     Stat  T-Statistic   P-Value  Reject Null
3      3P    -2.517634  0.017195         True
5     3P%    -3.718812  0.000793         True
24   3PAr    -2.705107  0.010994         True
35    DWS    -2.704644  0.011006         True
36     WS    -2.242816  0.032199         True
37  WS/48    -2.096513  0.044292         True
```

```
fig = px.histogram(sg2, x='DBPM', color='Standing',
                   marginal="box") # or violin, rug
fig.update_traces(opacity=0.3)
fig.update_layout(barmode='overlay',plot_bgcolor='rgba(0, 0, 0, 0)')
fig.update_xaxes(dtick=0.5,ticks='outside',showline=True,linecolor='black',mirror=False)
fig.show()
```

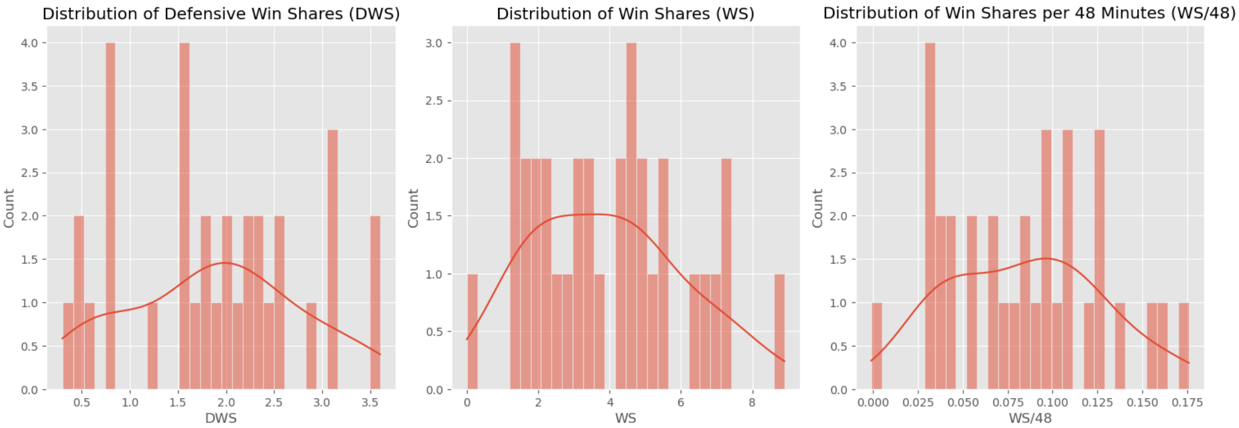


Visualization of Key Stats

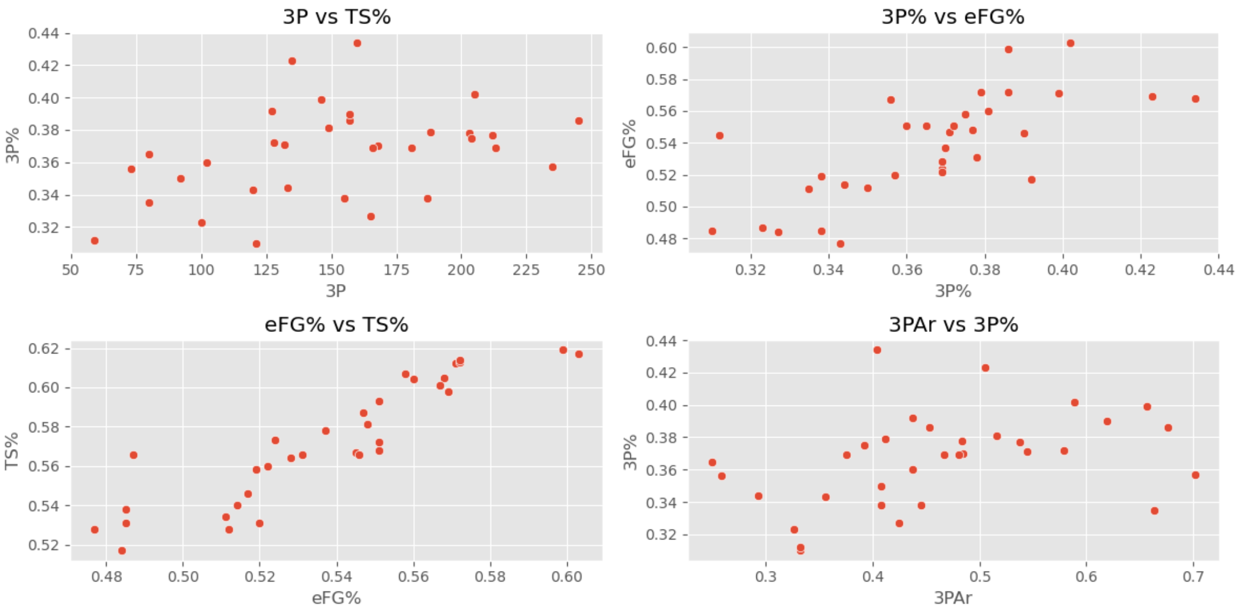
```
#['3P', '3P%', 'eFG%', '3PAr', 'DWS', 'WS', 'WS/48']
plt.figure(figsize=(15, 10))

plt.subplot(2, 3, 1)
sns.histplot(sg2['3P'], kde=True, bins=30)
plt.title('Distribution of 3P FG (3P)')

plt.subplot(2, 3, 2)
sns.histplot(sg2['3P%'], kde=True, bins=30)
plt.title('Distribution of 3P FG Percentage (3P%)')

plt.subplot(2, 3, 3)
sns.histplot(sg2['eFG%'], kde=True, bins=30)
plt.title('Distribution of Effective FG Percentage (eFG%)')

plt.subplot(2, 3, 4)
sns.histplot(sg2['TS%'], kde=True, bins=30)
plt.title('Distribution of True Shooting Percentage (TS%)')

plt.subplot(2, 3, 5)
sns.histplot(sg2['3PAr'], kde=True, bins=30)
plt.title('Distribution of 3P Rate Attempt (3PAr)')

plt.tight_layout()
plt.show()
```



```
#['3P', '3P%', 'eFG%', 'TS%'- .055, '3PAr', 'DWS', 'WS', 'WS/48']
plt.figure(figsize=(15, 10))

plt.subplot(2, 3, 1)
sns.histplot(sg2['DWS'], kde=True, bins=30)
plt.title('Distribution of Defensive Win Shares (DWS)')

plt.subplot(2, 3, 2)
sns.histplot(sg2['WS'], kde=True, bins=30)
plt.title('Distribution of Win Shares (WS)')

plt.subplot(2, 3, 3)
sns.histplot(sg2['WS/48'], kde=True, bins=30)
plt.title('Distribution of Win Shares per 48 Minutes (WS/48)')

plt.tight_layout()
plt.show()
```



```
# Correlation Analysis
correlation_matrix = sg2[['3P', '3P%', 'eFG%', 'TS%', '3PAr', 'DWS',
'WS', 'WS/48']].corr()
correlation_matrix

             3P       3P%      eFG%       TS%      3PAr       DWS
WS  \
3P     1.000000  0.389293  0.184016  0.209303  0.357400  0.220140
0.352966
3P%    0.389293  1.000000  0.716392  0.672925  0.387707  0.322050
0.541328
eFG%   0.184016  0.716392  1.000000  0.905679  0.329762  0.296495
0.617699
TS%    0.209303  0.672925  0.905679  1.000000  0.146836  0.268091
0.712465
3PAr   0.357400  0.387707  0.329762  0.146836  1.000000  0.036277
0.044122
DWS    0.220140  0.322050  0.296495  0.268091  0.036277  1.000000
0.688008
WS     0.352966  0.541328  0.617699  0.712465  0.044122  0.688008
1.000000
WS/48  0.250620  0.591228  0.695021  0.762919  0.112892  0.635237
0.968806

          WS/48
3P     0.250620
3P%    0.591228
eFG%   0.695021
TS%    0.762919
3PAr   0.112892
DWS    0.635237
WS     0.968806
WS/48  1.000000

#['3P', '3P%', 'eFG%', 'TS%'- .055, '3PAr', 'DWS', 'WS', 'WS/48']
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
sns.scatterplot(x='3P', y='3P%', data=sg2)
plt.title('3P vs TS%')

plt.subplot(2, 2, 2)
sns.scatterplot(x='3P%', y='eFG%', data=sg2)
plt.title('3P% vs eFG%')

plt.subplot(2, 2, 3)
sns.scatterplot(x='eFG%', y='TS%', data=sg2)
plt.title('eFG% vs TS%')

plt.subplot(2, 2, 4)
sns.scatterplot(x='3PAr', y='3P%', data=sg2)
plt.title('3PAr vs 3P%')

plt.tight_layout()
plt.show()
```



Weighting & Ranking

```
#[3P, 3P%, 3PAr, DWS, WS, WS/48]
weights = {'3P': 0.166, '3P%': 0.167, '3PAr': 0.167, 'DWS': 0.167,
'WS': 0.167, 'WS/48': 0.166}

# Normalize the stats and calculate weighted score
for stat in weights.keys():
    max_value = sg2[stat].max()
    sg2.loc[:, stat + '_norm'] = sg2[stat] / max_value
```

```
sg2.loc[:, 'Weighted_Score'] = sum([sg2[stat + '_norm'] * weight for
stat, weight in weights.items()])

# Rank players
ranked_sg2 = sg2.sort_values('Weighted_Score', ascending=False)

# Display the top ranked players
ranked_sg2[['Player_adv', 'Tm_adv', 'Pos_adv', 'Weighted_Score']]
```

```
                   Player_adv Tm_adv Pos_adv  Weighted_Score
72            Donovan Mitchell    CLE      SG        0.917656
121              Derrick White    BOS      SG        0.798223
57                 Zach LaVine    CHI      SG        0.759535
47                Kyrie Irving    DAL      SG        0.752825
89           Immanuel Quickley    NYK      SG        0.733209
2                Grayson Allen    MIL      SG        0.722513
69           De'Anthony Melton    PHI      SG        0.715916
46               Kevin Huerter    SAC      SG        0.705428
24           Spencer Dinwiddie    BRK      SG        0.682127
38             Quentin Grimes    NYK      SG        0.681874
28             Anthony Edwards    MIN      SG        0.674183
65                Tyrese Maxey    PHI      SG        0.667792
16    Kentavious Caldwell-Pope    DEN      SG        0.660612
60                Caris LeVert    CLE      SG        0.656816
44                 Tyler Herro    MIA      SG        0.643051
112             Gary Trent Jr.    TOR      SG        0.642308
120                 Coby White    CHI      SG        0.595973
7               Malik Beasley    LAL      SG        0.594269
125             Jalen Williams    OKC      SG        0.585752
8            Patrick Beverley    CHI      SG        0.569290
77             Dejounte Murray    ATL      SG        0.566298
103            Anfernee Simons    POR      SG        0.528452
33                 Eric Gordon    LAC      SG        0.481016
97                Terry Rozier    CHO      SG        0.479746
6                 Bradley Beal    WAS      SG        0.466728
26                 Ayo Dosunmu    CHI      SG        0.458675
19             Jordan Clarkson    UTA      SG        0.449566
4                   RJ Barrett    NYK      SG        0.445587
36                 Jalen Green    HOU      SG        0.442860
101             Shaedon Sharpe    POR      SG        0.392126
78             Andrew Nembhard    IND      SG        0.381052
64          Bennedict Mathurin    IND      SG        0.376323
48                  Jaden Ivey    DET      SG        0.334147
```