Import library

```
import numpy as np # for mathematical caluclations
import pandas as pd
from datetime import datetime  # to access datetime
import scipy.stats as stats

# for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px # for interactive plotting
import plotly.graph_objects as go # for interactive plotting

# set the plot style in matplotlib to ggplot and the firgure size to
15x5## Augmented Dickey Fuller Test for Assessing Stationarity
plt.style.use('ggplot')
plt.rcParams["figure.figsize"] = (15,5)

# for ingnoring warnings
import warnings # to ignore warning
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
```

Dataset

```
df_regular_stats = pd.read_excel('nba_stats_cleaned.xlsx')
df_advanced_stats = pd.read_csv('NBA Advanced Cleaned Stats.csv')
# df = pd.read_excel('NBA Per Game Cleaned Stats.xlsx')

df_advanced_stats.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541 entries, 0 to 540
Data columns (total 30 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Rk                 541 non-null    int64
 1   Player             541 non-null    object
 2   Pos                541 non-null    object
 3   Age                541 non-null    int64
 4   Tm                 541 non-null    object
 5   G                  541 non-null    int64
 6   MP                 541 non-null    int64
 7   PER                541 non-null    float64
 8   TS%                539 non-null    float64
 9   3PAr               539 non-null    float64
 10  FTr                539 non-null    float64
 11  ORB%               541 non-null    float64
 12  DRB%               541 non-null    float64
```

```
 13  TRB%                 541 non-null    float64
 14  AST%                 541 non-null    float64
 15  STL%                 541 non-null    float64
 16  BLK%                 541 non-null    float64
 17  TOV%                 540 non-null    float64
 18  USG%                 541 non-null    float64
 19  Unnamed: 19          0 non-null      float64
 20  OWS                  541 non-null    float64
 21  DWS                  541 non-null    float64
 22  WS                   541 non-null    float64
 23  WS/48                541 non-null    float64
 24  Unnamed: 24          0 non-null      float64
 25  OBPM                 541 non-null    float64
 26  DBPM                 541 non-null    float64
 27  BPM                  541 non-null    float64
 28  VORP                 541 non-null    float64
 29  Player-additional    541 non-null    object
dtypes: float64(22), int64(4), object(4)
memory usage: 126.9+ KB
```

```
# number of rows should be 538
df = pd.merge(df_regular_stats, df_advanced_stats[['Player', 'MP',
'PER', 'TS%', '3PAr', 'FTr', 'ORB%', 'DRB%', 'TRB%','AST%', 'STL%',
'BLK%', 'TOV%', 'USG%', 'OWS', 'DWS', 'WS', 'WS/48', 'OBPM', 'DBPM',
'BPM', 'VORP']],  how='left', left_on=['Player','MP'], right_on =
['Player','MP'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 538 entries, 0 to 537
Data columns (total 51 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Rk                  538 non-null    int64
 1   Player              538 non-null    object
 2   Pos                 538 non-null    object
 3   Age                 538 non-null    int64
 4   Tm                  538 non-null    object
 5   G                   538 non-null    int64
 6   GS                  538 non-null    int64
 7   MP                  538 non-null    int64
 8   FG                  538 non-null    int64
 9   FGA                 538 non-null    int64
 10  FG%                 536 non-null    float64
 11  3P                  538 non-null    int64
 12  3PA                 538 non-null    int64
 13  3P%                 522 non-null    float64
 14  2P                  538 non-null    int64
 15  2PA                 538 non-null    int64
 16  2P%                 533 non-null    float64
```

```
 17  eFG%                536 non-null    float64
 18  FT                  538 non-null    int64
 19  FTA                 538 non-null    int64
 20  FT%                 515 non-null    float64
 21  ORB                 538 non-null    int64
 22  DRB                 538 non-null    int64
 23  TRB                 538 non-null    int64
 24  AST                 538 non-null    int64
 25  STL                 538 non-null    int64
 26  BLK                 538 non-null    int64
 27  TOV                 538 non-null    int64
 28  PF                  538 non-null    int64
 29  PTS                 538 non-null    int64
 30  Player-additional   538 non-null    object
 31  PER                 523 non-null    float64
 32  TS%                 521 non-null    float64
 33  3PAr                521 non-null    float64
 34  FTr                 521 non-null    float64
 35  ORB%                523 non-null    float64
 36  DRB%                523 non-null    float64
 37  TRB%                523 non-null    float64
 38  AST%                523 non-null    float64
 39  STL%                523 non-null    float64
 40  BLK%                523 non-null    float64
 41  TOV%                522 non-null    float64
 42  USG%                523 non-null    float64
 43  OWS                 523 non-null    float64
 44  DWS                 523 non-null    float64
 45  WS                  523 non-null    float64
 46  WS/48               523 non-null    float64
 47  OBPM                523 non-null    float64
 48  DBPM                523 non-null    float64
 49  BPM                 523 non-null    float64
 50  VORP                523 non-null    float64
dtypes: float64(25), int64(22), object(4)
memory usage: 214.5+ KB
```

```
df_point_guards = df[(df['Pos'] == 'PG') | (df['Pos'] == 'SG-PG')]

df_point_guards.describe()
```

```
                Rk         Age           G          GS            MP
FG  \
count   93.000000  93.000000  93.000000  93.000000    93.000000
93.000000
mean   263.580645  26.333333  46.795699  23.645161  1148.516129
205.989247
std    151.142628   4.518881  24.096527  27.439517   835.574244
192.958460
min      9.000000  19.000000   1.000000   0.000000     5.000000
```

```
0.000000
25%    130.000000  23.000000  27.000000   1.000000   335.000000
49.000000
50%    256.000000  25.000000  56.000000   7.000000  1106.000000
144.000000
75%    388.000000  30.000000  67.000000  56.000000  1940.000000
307.000000
max    536.000000  37.000000  82.000000  79.000000  2725.000000
719.000000

                FGA         FG%          3P         3PA         3P%
2P  \
count    93.000000  93.000000   93.000000   93.00000  93.000000
93.000000
mean    460.473118   0.418806   67.505376  188.83871   0.331065
138.483871
std     409.891147   0.081337   66.001091  175.18309   0.084066
142.907663
min       3.000000   0.000000    0.000000    1.00000   0.000000
0.000000
25%     110.000000   0.402000   11.000000   28.00000   0.303000
37.000000
50%     351.000000   0.423000   47.000000  134.00000   0.335000
97.000000
75%     710.000000   0.462000  117.000000  315.00000   0.371000
210.000000
max    1449.000000   0.566000  273.000000  658.00000   0.500000
646.000000

                2PA         2P%        eFG%          FT          FTA
FT%  \
count    93.000000  92.000000  93.000000   93.000000   93.000000
88.000000
mean    271.634409   0.470511   0.491323  102.784946  124.870968
0.769148
std     268.594198   0.126454   0.083903  136.440770  160.986512
0.146716
min       0.000000   0.000000   0.000000    0.000000    0.000000
0.000000
25%      68.000000   0.454250   0.464000   13.000000   18.000000
0.732750
50%     180.000000   0.492500   0.503000   49.000000   64.000000
0.811000
75%     425.000000   0.535250   0.537000  148.000000  170.000000
0.859250
max    1213.000000   0.662000   0.614000  669.000000  739.000000
1.000000

              ORB         DRB         TRB         AST         STL
BLK  \
```

```
count  93.000000   93.000000   93.000000   93.000000   93.000000
93.000000
mean   25.139785  118.978495  144.118280  213.989247   42.322581
12.666667
std    20.835584   99.894942  117.735504  185.798073   32.921237
12.187247
min     0.000000    0.000000    0.000000    1.000000    0.000000
0.000000
25%     8.000000   32.000000   44.000000   50.000000   10.000000
3.000000
50%    25.000000  109.000000  123.000000  172.000000   39.000000
10.000000
75%    40.000000  189.000000  225.000000  369.000000   67.000000
20.000000
max    89.000000  515.000000  569.000000  741.000000  123.000000
65.000000

               TOV          PF          PTS        PER         TS%
3PAr  \
count   93.000000   93.000000    93.000000  90.000000  90.000000
90.000000
mean    80.290323   82.505376   582.268817  13.242222   0.525989
0.431578
std     71.885716   60.844532   567.586579   5.872196   0.092120
0.171447
min      0.000000    0.000000     0.000000  -6.800000   0.000000
0.009000
25%     18.000000   24.000000   114.000000   9.925000   0.498500
0.331500
50%     62.000000   79.000000   407.000000  12.600000   0.540000
0.422000
75%    129.000000  126.000000   830.000000  16.275000   0.578750
0.536250
max    300.000000  219.000000  2138.000000  28.700000   0.656000
1.000000

              FTr        ORB%        DRB%        TRB%        AST%        STL
%  \
count  90.000000  90.000000  90.000000  90.000000  90.000000
90.000000
mean    0.228978   2.760000  11.134444   6.945556  24.363333
1.822222
std     0.139477   2.664869   4.080790   2.799846   8.924974
0.840248
min     0.000000   0.000000   0.000000   0.000000   7.800000
0.000000
25%     0.141500   1.700000   8.175000   5.100000  16.600000
1.325000
50%     0.201000   2.300000  10.650000   6.600000  23.400000
```

```
1.700000
75%     0.299750   3.200000  13.325000   8.175000  29.950000
2.275000
max     0.750000  21.900000  25.400000  21.600000  47.600000
5.700000

             BLK%        TOV%       USG%        OWS         DWS
WS  \
count  90.000000  90.000000  90.000000  90.000000  90.000000
90.000000
mean    0.985556  14.406667  20.245556   1.406667   1.090000
2.501111
std     0.755588   5.882173   6.384218   2.248285   0.909383
2.888975
min     0.000000   0.000000   5.100000  -1.900000   0.000000  -
1.200000
25%     0.500000  11.475000  15.900000   0.000000   0.225000
0.225000
50%     0.900000  13.350000  19.600000   0.300000   0.950000
1.400000
75%     1.300000  15.900000  24.275000   2.100000   1.800000
3.700000
max     4.500000  41.400000  37.600000   8.400000   3.000000
11.400000

           WS/48       OBPM       DBPM        BPM       VORP
count  90.000000  90.000000  90.000000  90.000000  90.000000
mean    0.070411  -1.005556  -0.177778  -1.178889   0.855556
std     0.085150   3.783060   1.747600   4.492778   1.532869
min    -0.378000 -12.500000  -9.800000 -22.200000  -1.000000
25%     0.035750  -3.250000  -0.900000  -3.325000   0.000000
50%     0.077000  -1.350000  -0.250000  -1.100000   0.150000
75%     0.119000   0.950000   0.800000   1.050000   1.300000
max     0.226000   8.300000   3.600000   8.900000   6.600000

df_point_guards.info()

<class 'pandas.core.frame.DataFrame'>
Index: 93 entries, 8 to 534
Data columns (total 51 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Rk                 93 non-null     int64
 1   Player             93 non-null     object
 2   Pos                93 non-null     object
 3   Age                93 non-null     int64
 4   Tm                 93 non-null     object
 5   G                  93 non-null     int64
 6   GS                 93 non-null     int64
 7   MP                 93 non-null     int64
```

```
 8   FG                  93 non-null     int64
 9   FGA                 93 non-null     int64
 10  FG%                 93 non-null     float64
 11  3P                  93 non-null     int64
 12  3PA                 93 non-null     int64
 13  3P%                 93 non-null     float64
 14  2P                  93 non-null     int64
 15  2PA                 93 non-null     int64
 16  2P%                 92 non-null     float64
 17  eFG%                93 non-null     float64
 18  FT                  93 non-null     int64
 19  FTA                 93 non-null     int64
 20  FT%                 88 non-null     float64
 21  ORB                 93 non-null     int64
 22  DRB                 93 non-null     int64
 23  TRB                 93 non-null     int64
 24  AST                 93 non-null     int64
 25  STL                 93 non-null     int64
 26  BLK                 93 non-null     int64
 27  TOV                 93 non-null     int64
 28  PF                  93 non-null     int64
 29  PTS                 93 non-null     int64
 30  Player-additional   93 non-null     object
 31  PER                 90 non-null     float64
 32  TS%                 90 non-null     float64
 33  3PAr                90 non-null     float64
 34  FTr                 90 non-null     float64
 35  ORB%                90 non-null     float64
 36  DRB%                90 non-null     float64
 37  TRB%                90 non-null     float64
 38  AST%                90 non-null     float64
 39  STL%                90 non-null     float64
 40  BLK%                90 non-null     float64
 41  TOV%                90 non-null     float64
 42  USG%                90 non-null     float64
 43  OWS                 90 non-null     float64
 44  DWS                 90 non-null     float64
 45  WS                  90 non-null     float64
 46  WS/48               90 non-null     float64
 47  OBPM                90 non-null     float64
 48  DBPM                90 non-null     float64
 49  BPM                 90 non-null     float64
 50  VORP                90 non-null     float64
dtypes: float64(25), int64(22), object(4)
memory usage: 37.8+ KB
```

Categorizing Top and Bottom Team based on Leaderboard

```
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']

df_point_guards['Team Standing'] = ['Top' if x in top else 'Bottom'
for x in df_point_guards['Tm']]
df_point_guards
```

```
      Rk                    Player    Pos  Age   Tm   G  GS    MP   FG
FGA  \
8      9            Jose Alvarado     PG   24  NOP  61  10  1310  201
489
12    13             Cole Anthony     PG   22  ORL  60   4  1552  277
610
14    15         Ryan Arcidiacono     PG   28  POR  20   4   172    9
37
20    21              LaMelo Ball     PG   21  CHO  36  36  1268  296
721
24    25            Dalano Banton     PG   23  TOR  31   2   279   55
130
38    39         Patrick Beverley  SG-PG   34  CHI  67  67  1816  144
360
52    53           Jamaree Bouyea     PG   23  WAS   5   0    71    6
14
59    60          Malcolm Brogdon     PG   30  BOS  67   0  1744  354
732
68    69            Jalen Brunson     PG   26  NYK  68  68  2379  587
1195
73    74             Jared Butler     PG   22  OKC   6   1    77   15
32
78    79         Facundo Campazzo     PG   31  DAL   8   0    52    3
13
82    83             Jevon Carter     PG   27  MIL  81  39  1810  239
565
84    85  Michael Carter-Williams     PG   31  ORL   4   0    44    6
14
85    86              Alex Caruso     PG   28  CHI  67  36  1575  130
286
87    89         Kennedy Chandler     PG   20  MEM  36   0   281   35
83
97    99              Mike Conley     PG   35  MIN  67  66  2029  261
610
104  106          Cade Cunningham     PG   21  DET  12  12   400   93
224
106  108            Stephen Curry     PG   34  GSW  56  56  1941  559
```

```
1133
107  109             Dyson Daniels     PG   19  NOP  59  11  1042   87
208
114  116      Matthew Dellavedova     PG   32  SAC  32   0   213   17
50
121  123        Spencer Dinwiddie  SG-PG   29  BRK  79  79  2725  460
1050
123  125              Luka Dončić     PG   23  DAL  66  66  2391  719
1449
127  129             Devon Dotson     PG   23  WAS   6   0    53    1
10
128  130              Jeff Dowtin     PG   25  TOR  25   0   259   25
57
130  132            Goran DragiÄ‡     PG   36  MIL  58   0   870  141
335
134  136               Kris Dunn     PG   28  UTA  22   3   568  116
216
145  147            Malachi Flynn     PG   24  TOR  53   2   691   87
242
151  153             De'Aaron Fox     PG   25  SAC  73  73  2435  682
1331
152  154           Markelle Fultz     PG   24  ORL  60  60  1778  349
679
155  157           Darius Garland     PG   23  CLE  69  69  2447  522
1129
162  164  Shai Gilgeous-Alexander     PG   24  OKC  68  68  2416  704
1381
164  166            Jacob Gilyard     PG   24  MEM   1   0    41    1
3
166  168           Jordan Goodwin     PG   24  WAS  62   7  1106  158
353
169  171          Devonte' Graham     PG   27  SAS  73   8  1338  161
431
184  186        Tyrese Haliburton     PG   22  IND  56  56  1883  412
841
188  190             James Harden     PG   33  PHI  58  58  2135  371
842
202  204            Killian Hayes     PG   21  DET  76  56  2154  307
815
211  213            Aaron Holiday     PG   26  ATL  63   6   845   92
220
212  214             Jrue Holiday     PG   32  MIL  67  65  2183  490
1023
219  221           Trevor Hudgins     PG   23  HOU   5   0    28    2
9
223  225             Bones Hyland     PG   22  LAC  56   1  1085  228
571
231  233            Frank Jackson     PG   24  UTA   1   0     5    0
3
```

235	237	Quenton Jackson	PG	24	WAS	9	0	135	19
42									
236	238	Reggie Jackson	PG	32	DEN	68	40	1657	263
640									
248	250	Carlik Jones	PG	25	CHI	7	0	56	6
15									
253	255	Tre Jones	PG	23	SAS	68	65	1984	341
743									
254	256	Tyus Jones	PG	26	MEM	80	22	1940	311
710									
256	258	Cory Joseph	PG	31	DET	62	2	1227	150
351									
274	276	Vit Krejci	PG	22	ATL	29	0	165	15
37									
284	286	Saben Lee	PG	23	PHO	25	1	373	49
121									
289	291	Kira Lewis Jr.	PG	21	NOP	25	0	235	40
88									
290	292	Damian Lillard	PG	32	POR	58	58	2107	556
1202									
298	300	Kyle Lowry	PG	36	MIA	55	44	1718	196
485									
300	302	ThÃ©o Maledon	PG	21	CHO	44	7	854	102
254									
303	305	Tre Mann	PG	21	OKC	67	5	1183	197
501									
315	317	Skylar Mays	PG	25	POR	6	6	189	34
68									
316	318	Miles McBride	PG	22	NYK	64	2	760	77
215									
319	321	T.J. McConnell	PG	30	IND	75	6	1526	283
521									
326	328	Jordan McLaughlin	PG	26	MIN	43	0	678	61
145									
331	333	Patty Mills	PG	34	BRK	40	2	567	86
209									
335	337	Davion Mitchell	PG	24	SAC	80	9	1447	181
399									
344	346	Ja Morant	PG	23	MEM	61	59	1948	566
1214									
347	349	Monte Morris	PG	27	WAS	62	61	1695	247
515									
350	352	Jamal Murray	PG	25	DEN	65	65	2133	473
1041									
357	359	Raul Neto	PG	30	CLE	48	1	505	57
110									
359	361	Daishen Nix	PG	20	HOU	57	7	914	81
237									
378	380	Chris Paul	PG	37	PHO	59	59	1889	294

668
379 381   Cameron Payne   PG   28 PHO 48 15   968 188
453
382 384   Scotty Pippen Jr.   PG   22 LAL   6   0   32   4
12
386 388   Jordan Poole   PG   23 GSW 82 43 2458 550
1278
387 389   Kevin Porter Jr.   PG   22 HOU 59 59 2024 391
884
395 397   Jason Preston   PG   23 LAC 14   0   124   18
41
398 400   Payton Pritchard   PG   25 BOS 48   3   643 101
245
419 421   Ryan Rollins   PG   20 GSW 12   0   62   7
20
420 422   Derrick Rose   PG   34 NYK 27   0   338   61
159
423 425   Ricky Rubio   PG   32 CLE 33   2   566   61
178
424 426   D'Angelo Russell   PG   26 LAL 71 71 2304 445
948
432 434   Dennis SchrÃ¶der   PG   29 LAL 66 50 1986 270
650
436 438   Collin Sexton   PG   24 UTA 48 15 1145 237
468
442 444   Ben Simmons   PG   26 BRK 42 33 1105 133
235
447 449   Marcus Smart   PG   28 BOS 61 61 1957 250
602
448 450   Dennis Smith Jr.   PG   25 CHO 54 15 1390 186
452
450 452   Ish Smith   PG   34 DEN 43   0   398   52
131
483 485   Fred VanVleet   PG   28 TOR 69 69 2535 437
1112
485 487   Gabe Vincent   PG   26 MIA 68 34 1759 228
567
493 495   Kemba Walker   PG   32 DAL   9   1   144   24
57
495 497   John Wall   PG   32 LAC 34   3   755 138
338
499 501   TyTy Washington Jr.   PG   21 HOU 31   2   433   58
160
505 507   Russell Westbrook   PG   34 LAC 73 24 2126 432
991
512 514   Lindell Wigginton   PG   24 MIL   7   1   87   18
37
530 532   Delon Wright   PG   30 WAS 50 14 1221 138
291

```
531  533       McKinley Wright IV     PG   24  DAL  27   1   335   46
98
534  536                Trae Young     PG   24  ATL  73  73  2541  597
1390

       FG%   3P  3PA    3P%   2P   2PA    2P%   eFG%   FT  FTA    FT%
ORB  \
8    0.411   83  247  0.336  118   242  0.488  0.496   65   80  0.813
28
12   0.454   75  206  0.364  202   404  0.500  0.516  152  170  0.894
47
14   0.243    8   23  0.348    1    14  0.071  0.351    0    0    NaN
0
20   0.411  144  383  0.376  152   338  0.450  0.510  102  122  0.836
42
24   0.423   15   51  0.294   40    79  0.506  0.481   17   24  0.708
11
38   0.400   80  239  0.335   64   121  0.529  0.511   47   65  0.723
44
52   0.429    2    6  0.333    4     8  0.500  0.500    1    2  0.500
1
59   0.484  132  297  0.444  222   435  0.510  0.574  160  184  0.870
42
68   0.491  134  322  0.416  453   873  0.519  0.547  325  392  0.829
40
73   0.469    7   14  0.500    8    18  0.444  0.578    0    0    NaN
1
78   0.231    3   11  0.273    0     2  0.000  0.346    1    2  0.500
0
82   0.423  142  337  0.421   97   228  0.425  0.549   31   38  0.816
35
84   0.429    1    3  0.333    5    11  0.455  0.464    4    7  0.571
1
85   0.455   55  151  0.364   75   135  0.556  0.551   59   73  0.808
42
87   0.422    2   15  0.133   33    68  0.485  0.434    6   13  0.462
9
97   0.428  135  351  0.385  126   259  0.486  0.539  141  169  0.834
33
104  0.415   17   61  0.279   76   163  0.466  0.453   36   43  0.837
12
106  0.493  273  639  0.427  286   494  0.579  0.614  257  281  0.915
39
107  0.418   27   86  0.314   60   122  0.492  0.483   26   40  0.650
36
114  0.340    9   27  0.333    8    23  0.348  0.430    4    7  0.571
1
121  0.438  181  490  0.369  279   560  0.498  0.524  268  330  0.812
28
```

123	0.496	185	541	0.342	534	908	0.588	0.560	515	694	0.742
54											
127	0.100	1	4	0.250	0	6	0.000	0.150	0	0	NaN
6											
128	0.439	5	16	0.313	20	41	0.488	0.482	6	9	0.667
6											
130	0.421	52	145	0.359	89	190	0.468	0.499	31	45	0.689
16											
134	0.537	17	36	0.472	99	180	0.550	0.576	41	53	0.774
9											
145	0.360	47	133	0.353	40	109	0.367	0.457	25	33	0.758
15											
151	0.512	119	367	0.324	563	964	0.584	0.557	343	440	0.780
40											
152	0.514	27	87	0.310	322	592	0.544	0.534	112	143	0.783
69											
155	0.462	169	412	0.410	353	717	0.492	0.537	277	321	0.863
28											
162	0.510	58	168	0.345	646	1213	0.533	0.531	669	739	0.905
59											
164	0.333	1	3	0.333	0	0	NaN	0.500	0	0	NaN
0											
166	0.448	38	118	0.322	120	235	0.511	0.501	53	69	0.768
57											
169	0.374	119	338	0.352	42	93	0.452	0.512	98	131	0.748
19											
184	0.490	161	402	0.400	251	439	0.572	0.586	175	201	0.871
33											
188	0.441	161	418	0.385	210	424	0.495	0.536	313	361	0.867
40											
202	0.377	80	286	0.280	227	529	0.429	0.426	92	112	0.821
28											
211	0.418	36	88	0.409	56	132	0.424	0.500	27	32	0.844
25											
212	0.479	158	411	0.384	332	612	0.542	0.556	152	177	0.859
79											
219	0.222	2	8	0.250	0	1	0.000	0.333	3	3	1.000
0											
223	0.399	117	315	0.371	111	256	0.434	0.502	86	102	0.843
20											
231	0.000	0	1	0.000	0	2	0.000	0.000	0	0	NaN
1											
235	0.452	1	12	0.083	18	30	0.600	0.464	17	22	0.773
2											
236	0.411	96	288	0.333	167	352	0.474	0.486	71	78	0.910
25											
248	0.400	3	6	0.500	3	9	0.333	0.500	5	8	0.625
1											
253	0.459	45	158	0.285	296	585	0.506	0.489	148	172	0.860

56
254 0.438 121 326 0.371 190 384 0.495 0.523 80 100 0.800
28
256 0.427 70 180 0.389 80 171 0.468 0.527 57 72 0.792
21
274 0.405 5 21 0.238 10 16 0.625 0.473 1 2 0.500
6
284 0.405 11 30 0.367 38 91 0.418 0.450 42 57 0.737
13
289 0.455 15 34 0.441 25 54 0.463 0.540 19 22 0.864
5
290 0.463 244 658 0.371 312 544 0.574 0.564 510 558 0.914
44
298 0.404 107 310 0.345 89 175 0.509 0.514 116 135 0.859
43
300 0.402 28 95 0.295 74 159 0.465 0.457 63 74 0.851
14
303 0.393 81 257 0.315 116 244 0.475 0.474 42 55 0.764
27
315 0.500 12 26 0.462 22 42 0.524 0.588 12 13 0.923
2
316 0.358 40 134 0.299 37 81 0.457 0.451 28 42 0.667
13
319 0.543 26 59 0.441 257 462 0.556 0.568 58 68 0.853
45
326 0.421 24 78 0.308 37 67 0.552 0.503 15 18 0.833
15
331 0.411 49 134 0.366 37 75 0.493 0.529 25 30 0.833
8
335 0.454 63 197 0.320 118 202 0.584 0.533 25 31 0.806
16
344 0.466 92 300 0.307 474 914 0.519 0.504 372 497 0.748
61
347 0.480 78 204 0.382 169 311 0.543 0.555 64 77 0.831
26
350 0.454 172 432 0.398 301 609 0.494 0.537 180 216 0.833
48
357 0.518 12 42 0.286 45 68 0.662 0.573 31 34 0.912
10
359 0.342 40 140 0.286 41 97 0.423 0.426 24 36 0.667
18
378 0.440 98 261 0.375 196 407 0.482 0.513 133 160 0.831
27
379 0.415 68 185 0.368 120 268 0.448 0.490 49 64 0.766
12
382 0.333 1 3 0.333 3 9 0.333 0.375 5 9 0.556
2
386 0.430 214 637 0.336 336 641 0.524 0.514 361 415 0.870
32

```
387  0.442  141  385  0.366  250   499  0.501  0.522  207  264  0.784
76
395  0.439    5   18  0.278   13    23  0.565  0.500    0    2  0.000
2
398  0.412   56  154  0.364   45    91  0.495  0.527   12   16  0.750
25
419  0.350    3    9  0.333    4    11  0.364  0.425    6    6  1.000
3
420  0.384   19   63  0.302   42    96  0.438  0.443   11   12  0.917
8
423  0.343   21   82  0.256   40    96  0.417  0.402   28   35  0.800
9
424  0.469  194  490  0.396  251   458  0.548  0.572  179  216  0.829
35
432  0.415   74  225  0.329  196   425  0.461  0.472  216  252  0.857
21
436  0.506   48  122  0.393  189   346  0.546  0.558  163  199  0.819
37
442  0.566    0    2  0.000  133   233  0.571  0.566   25   57  0.439
40
447  0.415  115  342  0.336  135   260  0.519  0.511   88  118  0.746
46
448  0.412   24  111  0.216  162   341  0.475  0.438   78  106  0.736
27
450  0.397    2   12  0.167   50   119  0.420  0.405    2    4  0.500
5
483  0.393  207  606  0.342  230   506  0.455  0.486  254  283  0.898
30
485  0.402  117  350  0.334  111   217  0.512  0.505   68   78  0.872
27
493  0.421    7   28  0.250   17    29  0.586  0.482   17   21  0.810
3
495  0.408   33  109  0.303  105   229  0.459  0.457   77  113  0.681
14
499  0.363   19   80  0.238   39    80  0.488  0.422   10   18  0.556
2
505  0.436   89  286  0.311  343   705  0.487  0.481  206  314  0.656
89
512  0.486    6   18  0.333   12    19  0.632  0.568    8    9  0.889
0
530  0.474   41  119  0.345   97   172  0.564  0.545   52   60  0.867
58
531  0.469    9   28  0.321   37    70  0.529  0.515   13   19  0.684
9
534  0.429  154  460  0.335  443   930  0.476  0.485  566  639  0.886
56

     DRB  TRB  AST  STL  BLK  TOV   PF   PTS Player-additional   PER
TS%  \
```

```
8    113  141  186   67   10   81  125   550          alvarjo01  11.8
0.525
12   241  288  235   37   31   91  158   781          anthoco01  16.0
0.570
14    15   15   23    5    0    7   17    26          arcidry01   2.7
0.351
20   189  231  304   46   11  129  118   838           balla01  17.9
0.541
24    34   45   36   13   13   18   34   142          bantoda01  14.9
0.505
38   203  247  194   63   41   60  187   415          beverpa01   8.9
0.534
52     5    6    4    4    2    4    6    15          bouyeja01   6.5
0.504
59   238  280  248   45   18   98  109  1000          brogdma01  18.2
0.615
68   201  241  421   61   15  142  152  1633          brunsja01  21.2
0.597
73     3    4    8    5    0    5    5    37          butleja02  13.0
0.578
78     2    2    9    6    0    3    3    10          campafa01   7.8
0.360
82   167  202  197   66   29   78  158   651          carteje01  10.9
0.560
84     4    5    7    1    1    4    5    17          cartemi01   9.0
0.498
85   154  196  193   98   46   77  159   374          carusal01  11.5
0.588
87    29   38   58   12    5   18   15    78          chandke01  10.6
0.440
97   149  182  450   73   14  103  139   798          conlemi01  14.7
0.583
104   62   74   72   10    7   39   34   239          cunnica01  14.0
0.492
106  302  341  352   52   20  179  117  1648          curryst01  24.1
0.656
107  152  188  134   43   11   57   99   227          daniedy01   8.8
0.503
114   12   13   41    7    0   10   20    47          dellama01   6.7
0.443
121  242  270  515   67   24  145  187  1369          dinwisp01  16.0
0.573
123  515  569  529   90   33  236  166  2138          doncilu01  28.7
0.609
127    4   10    8    5    0    3    9     3          dotsode01   4.8
0.150
128   17   23   31    9    3    5   13    61          dowtije01   9.9
0.500
130   65   81  151   13    4   63   65   365          dragigo01   NaN
```

```
NaN
134   91  100  124   25   10   35   58   290          dunnkr01  19.1
0.606
145   61   76   70   21    4   25   65   246         flynnma01   8.7
0.479
151  266  306  447   83   23  181  172  1826           foxde01  21.8
0.599
152  165  234  341   87   26  139  131   837         fultzma01  16.6
0.564
155  157  185  538   85    9  199  148  1490         garlada01  18.8
0.587
162  270  329  371  112   65  192  192  2135         gilgesh01  27.2
0.626
164    4    4    7    3    0    2    3     3         gilyaja01   7.3
0.500
166  148  205  168   58   26   57   95   407         goodwjo01  14.8
0.531
169  103  122  196   45   16   53   79   539         grahade01  12.5
0.552
184  172  205  585   91   25  141   69  1160         halibty01  23.6
0.624
188  314  354  618   71   31  195  112  1216         hardeja01  21.6
0.607
202  193  221  470  104   28  173  219   786         hayeski01  10.2
0.455
211   49   74   89   37   12   36   79   247         holidaa01   9.4
0.528
212  262  341  495   79   25  197  116  1290         holidjr01  19.2
0.586
219    0    0    3    0    0    1    1     9         hudgitr01   4.2
0.436
223  115  135  172   39   15   82   96   659         hylanbo01  14.9
0.535
231    1    2    1    0    0    0    0     0         jacksfr01  -6.8
0.000
235    6    8   15    4    1    4   10    56         jacksqu01  12.2
0.542
236  119  144  233   47    6  115  114   693         jacksre01  10.0
0.514
248    4    5    6    2    0    2    2    20         jonesca03  11.1
0.540
253  189  245  448   89    9  110   98   875         jonestr01  16.0
0.534
254  172  200  417   83    6   74   32   823         jonesty01  16.0
0.546
256   85  106  217   34    9   56   88   427         josepco01  12.1
0.558
274   20   26   17    5    1    5   17    36         krejcvi01   7.7
0.475
```

```
284   32   45   67   19    1   24   32   151          leesa01  13.1
0.517
289   28   33   23   10    2   10   24   114        lewiski01  14.5
0.584
290  233  277  425   50   18  191  109  1866        lillada01  26.7
0.645
298  182  225  281   57   21  103  143   615        lowryky01  12.6
0.565
300  109  123  152   37   12   59   61   295        maledth01   NaN
NaN
303  128  155  120   40   11   62   99   517         manntr01  10.0
0.492
315   17   19   50    6    1   10   11    92         mayssk01  19.4
0.624
316   37   50   72   37    8   25   60   222        mcbrimi01   8.4
0.475
319  189  234  397   81   10  143  105   650        mccontj01  16.8
0.590
326   47   62  148   32    4   33   28   161        mclaujo01  11.8
0.526
331   36   44   56   15    3   30   32   246        millspa02  10.6
0.554
335   89  105  185   44   15   62  112   450        mitchda01   9.4
0.545
344  296  357  493   66   16  206  100  1596        moranja01  23.3
0.557
347  184  210  326   43   13   60   75   636        morrimo01  15.0
0.579
350  209  257  400   66   16  145  103  1298        murraja01  18.0
0.571
357   36   46   79   17    4   24   49   157         netora01  12.6
0.628
359   80   98  132   31    7   86   59   226          nixda01   5.3
0.447
378  224  251  524   91   22  114  126   819         paulch01  17.7
0.555
379   94  106  214   33    8   81   88   493        payneca01  13.2
0.512
382    2    4    2    2    1    2    3    14        pippesc02   9.6
0.439
386  193  225  369   63   21  252  214  1675        poolejo01  14.6
0.573
387  238  314  338   82   17  188  156  1130        porteke02  16.2
0.565
395   20   22   27    2    0   10    7    41        prestja01  10.3
0.489
398   63   88   64   14    1   40   37   270        pritcpa01  10.3
0.536
419    9   12    6    1    1   16   10    23        rolliry01  -1.4
0.508
```

```
420   32   40   46    7    5   22   18   152          rosede01  10.1
0.463
423   61   70  115   26    6   31   51   171         rubiori01  10.4
0.442
424  180  215  437   70   29  186  140  1263         russeda01  16.3
0.605
432  144  165  298   50   10  112  145   830         schrode01   NaN
NaN
436   68  105  138   27    6   85  112   685         sextoco01  15.9
0.616
442  223  263  256   54   24   97  139   291         simmobe01  13.4
0.559
447  145  191  382   93   23  143  172   703         smartma01  12.1
0.538
448  140  167  261   75   25   82  122   474         smithde03  11.6
0.475
450   49   54  100    8    7   44   33   108         smithis01   6.5
0.407
483  250  280  495  123   38  140  193  1335         vanvlfr01  17.0
0.540
485  118  145  167   62    5   92  154   641         vincega01   9.0
0.533
493   13   16   19    2    2    4   11    72         walkeke02  15.0
0.543
495   78   92  178   27   12   80   59   386          walljo01  13.6
0.498
499   43   45   47   15    2   14   29   145         washity02   7.5
0.432
505  334  423  551   76   33  255  162  1159         westbru01  16.1
0.513
512    7    7   14    0    2   12    7    50         wiggili01  11.8
0.610
530  122  180  194   92   17   44   59   369         wrighde01  16.3
0.581
531   38   47   58    8    5   19   24   114         wrighmc01  12.4
0.536
534  161  217  741   80    9  300  104  1914         youngtr01  22.0
0.573

      3PAr    FTr  ORB%  DRB%  TRB%  AST%  STL%  BLK%  TOV%  USG%  OWS
DWS  \
8    0.505  0.164   2.4   9.8   6.1  19.9   2.5   0.7  13.4  19.8  0.1
1.7
12   0.338  0.279   3.4  17.6  10.5  22.9   1.2   1.9  11.7  21.5  2.1
1.7
14   0.622  0.000   0.0  10.0   5.0  16.9   1.4   0.0  15.9  11.1 -0.2
0.1
20   0.531  0.169   3.5  16.2   9.7  38.7   1.7   0.8  14.3  30.0  0.6
1.2
```

24	0.392	0.185	4.2	14.9	9.1	19.3	2.3	4.5	11.4	24.2	0.0
0.4											
38	0.664	0.181	2.7	12.0	7.5	13.3	1.7	2.0	13.4	10.6	0.8
2.1											
52	0.429	0.143	1.6	8.4	4.9	7.8	2.8	2.9	21.2	11.8	-0.1
0.1											
59	0.406	0.251	2.7	14.7	8.8	21.5	1.3	0.9	10.8	22.8	3.6
2.2											
68	0.269	0.328	1.8	9.3	5.6	28.7	1.3	0.6	9.4	27.2	6.9
1.8											
73	0.438	0.000	1.4	4.3	2.8	15.0	3.1	0.0	13.5	20.1	0.0
0.1											
78	0.846	0.154	0.0	4.5	2.2	22.6	5.7	0.0	17.8	14.7	-0.1
0.1											
82	0.596	0.067	2.1	9.3	5.8	14.5	1.7	1.3	11.8	15.3	1.2
2.3											
84	0.214	0.500	2.6	10.3	6.4	22.6	1.1	2.2	19.0	20.6	0.0
0.0											
85	0.528	0.255	3.1	10.9	7.1	15.4	3.0	2.8	19.5	11.1	1.0
2.6											
87	0.181	0.157	3.4	10.9	7.2	26.4	2.0	1.6	16.9	15.8	-0.1
0.4											
97	0.575	0.277	1.8	7.9	4.9	29.4	1.7	0.6	13.1	16.3	4.1
1.4											
104	0.272	0.192	3.2	17.7	10.3	30.7	1.2	1.5	13.8	30.0	-0.4
0.2											
106	0.564	0.248	2.3	16.8	9.7	30.0	1.3	0.9	12.5	31.0	5.8
2.0											
107	0.413	0.192	3.9	16.5	10.2	16.4	2.0	1.0	20.2	11.6	-0.1
1.4											
114	0.540	0.140	0.5	6.4	3.5	23.4	1.6	0.0	15.9	12.7	0.0
0.1											
121	0.467	0.314	1.2	10.2	5.7	28.1	1.2	0.8	10.8	22.1	4.5
1.8											
123	0.373	0.479	2.6	25.4	13.8	42.3	1.9	1.2	11.9	37.6	7.3
2.9											
127	0.400	0.000	13.1	8.1	10.5	17.6	4.6	0.0	23.1	10.7	-0.1
0.1											
128	0.281	0.158	2.4	8.0	5.0	15.5	1.7	1.1	7.6	10.9	0.2
0.2											
130	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN											
134	0.167	0.245	1.7	17.1	9.5	32.3	2.1	1.5	12.8	20.2	1.3
0.6											
145	0.550	0.136	2.3	10.8	6.2	13.7	1.5	0.6	8.9	17.4	0.0
0.5											
151	0.276	0.331	1.9	12.4	7.2	29.6	1.6	0.8	10.6	30.1	5.5
1.8											
152	0.128	0.211	4.4	10.5	7.4	29.8	2.4	1.4	15.8	21.3	1.7

```
2.0
155  0.365  0.284   1.3   7.6   4.5  34.1   1.7   0.3  13.5  26.9  4.5
3.0
162  0.122  0.535   2.6  12.2   7.3  25.7   2.2   2.5  10.1  32.8  8.4
3.0
164  1.000  0.000   0.0  10.3   5.2  19.4   3.5   0.0  40.0   5.1  0.0
0.1
166  0.334  0.195   6.0  14.3  10.3  20.8   2.6   2.0  12.9  17.3  0.9
1.3
169  0.784  0.304   1.6   8.7   5.1  19.4   1.6   1.1   9.8  17.1  1.5
1.0
184  0.478  0.239   1.9  10.1   6.0  47.6   2.3   1.1  13.2  23.8  6.4
1.2
188  0.496  0.429   2.3  17.1   9.8  43.3   1.6   1.4  16.3  25.0  5.8
2.6
202  0.351  0.137   1.4  10.2   5.7  32.3   2.3   1.1  16.7  20.5 -1.9
1.4
211  0.400  0.145   3.2   6.4   4.8  13.0   2.1   1.2  13.3  13.4  0.3
0.6
212  0.402  0.173   3.9  12.0   8.1  34.4   1.7   0.9  15.2  25.0  3.9
2.8
219  0.889  0.333   0.0   0.0   0.0  13.9   0.0   0.0   8.8  16.8  0.0
0.0
223  0.552  0.179   2.1  11.8   7.1  23.4   1.8   1.2  11.7  28.0  0.3
1.1
231  0.333  0.000  21.9  21.3  21.6  22.7   0.0   0.0   0.0  25.1  0.0
0.0
235  0.286  0.524   1.7   4.8   3.3  15.1   1.4   0.6   7.2  17.9  0.1
0.1
236  0.450  0.122   1.7   8.0   4.9  20.0   1.4   0.3  14.6  20.8 -0.6
1.2
248  0.400  0.533   2.1   8.0   5.1  13.9   1.7   0.0   9.7  16.2  0.0
0.1
253  0.213  0.231   3.0  10.8   6.7  31.5   2.1   0.4  11.8  19.3  2.6
0.8
254  0.459  0.141   1.5   9.4   5.5  28.8   2.0   0.3   8.9  17.7  3.4
2.3
256  0.513  0.205   1.8   7.9   4.8  25.4   1.3   0.6  12.8  15.2  1.5
0.3
274  0.568  0.054   3.9  13.4   8.6  12.4   1.4   0.5  11.7  10.9  0.0
0.1
284  0.248  0.471   3.8   9.7   6.7  24.3   2.5   0.2  14.1  19.5  0.2
0.4
289  0.386  0.250   2.4  13.5   8.0  14.0   2.1   0.8   9.3  19.7  0.3
0.3
290  0.547  0.464   2.4  12.8   7.6  35.0   1.2   0.8  11.7  33.8  8.2
0.8
298  0.639  0.278   2.8  12.9   7.7  23.4   1.7   1.2  15.9  16.7  1.5
1.9
```

300	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
303	0.513	0.110	2.4	11.8	7.0	14.0	1.6	0.9	10.6	20.7	-0.6	1.1
315	0.382	0.191	1.2	10.4	5.8	40.0	1.5	0.5	11.9	19.3	0.6	0.1
316	0.623	0.195	1.9	5.4	3.6	12.4	2.4	1.0	9.7	14.6	0.0	0.7
319	0.113	0.131	3.3	13.7	8.5	37.9	2.5	0.6	20.6	19.1	1.9	1.2
326	0.538	0.124	2.5	7.6	5.1	27.4	2.2	0.5	17.7	11.7	0.7	0.6
331	0.641	0.144	1.6	7.0	4.4	13.9	1.3	0.5	11.9	19.7	0.0	0.4
335	0.494	0.078	1.3	7.0	4.1	16.5	1.5	0.9	13.1	14.1	0.7	0.8
344	0.247	0.409	3.3	16.1	9.7	41.1	1.6	0.7	12.6	34.9	3.4	2.6
347	0.396	0.150	1.8	11.6	6.9	26.4	1.2	0.6	9.9	15.6	3.1	1.1
350	0.415	0.207	2.7	11.0	6.9	27.5	1.5	0.7	11.3	26.1	3.3	1.9
357	0.382	0.309	2.3	8.4	5.4	21.0	1.7	0.7	16.1	13.2	0.8	0.6
359	0.591	0.152	2.1	9.7	5.9	19.2	1.6	0.7	25.4	15.4	-1.5	0.3
378	0.391	0.240	1.6	13.4	7.4	38.7	2.4	1.0	13.4	19.2	3.7	2.5
379	0.408	0.141	1.3	11.0	6.1	32.6	1.7	0.7	14.4	24.8	0.0	1.0
382	0.250	0.750	6.9	6.5	6.7	8.2	3.0	2.6	11.1	23.7	-0.1	0.0
386	0.498	0.325	1.5	8.5	5.0	22.5	1.2	0.8	14.7	29.2	1.4	1.9
387	0.436	0.299	4.1	13.1	8.5	25.7	2.0	0.8	15.8	24.3	1.9	1.0
395	0.439	0.049	1.8	17.9	9.9	30.9	0.8	0.0	19.3	18.3	0.0	0.1
398	0.629	0.065	4.3	10.5	7.5	14.0	1.1	0.1	13.7	19.8	-0.1	0.7
419	0.450	0.300	5.4	15.7	10.6	12.4	0.8	1.4	41.4	26.1	-0.3	0.1
420	0.396	0.075	2.6	10.4	6.5	19.9	1.0	1.4	11.8	23.6	-0.3	0.3
423	0.461	0.197	1.9	12.7	7.3	27.1	2.3	1.0	13.8	17.7	-0.2	0.9
424	0.517	0.228	1.7	8.5	5.2	27.4	1.4	1.1	15.1	22.7	3.1	1.9
432	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

```
NaN
436  0.261  0.425   3.5   6.3   5.0  17.9   1.1   0.4  13.3  23.4  2.0
0.5
442  0.009  0.243   4.2  22.2  13.4  31.3   2.4   1.9  27.2  14.3  0.5
1.7
447  0.568  0.196   2.6   8.0   5.3  26.4   2.3   1.0  17.9  17.8  1.0
2.7
448  0.246  0.235   2.0  11.0   6.4  26.1   2.6   1.6  14.1  17.6 -0.4
1.5
450  0.092  0.031   1.5  13.8   7.8  32.5   1.0   1.6  24.9  19.3 -0.8
0.4
483  0.545  0.254   1.2  12.0   6.3  28.1   2.4   1.5  10.2  23.2  3.7
2.8
485  0.617  0.138   1.7   8.1   4.8  13.9   1.8   0.3  13.3  17.5 -0.3
1.7
493  0.491  0.368   2.4  10.6   6.5  20.1   0.7   1.2   5.7  22.1  0.2
0.1
495  0.322  0.334   2.1  11.4   6.8  35.3   1.8   1.4  17.1  27.0 -0.4
0.7
499  0.500  0.113   0.5  11.0   5.7  15.3   1.7   0.5   7.7  17.4 -0.3
0.1
505  0.289  0.317   4.7  16.5  10.8  38.6   1.7   1.3  18.4  27.7 -0.6
2.6
512  0.486  0.243   0.0   8.1   4.2  23.8   0.0   1.9  22.7  25.6  0.0
0.1
530  0.409  0.206   5.5  10.7   8.2  20.9   3.7   1.2  12.2  12.9  2.1
1.6
531  0.286  0.194   3.1  13.4   8.2  25.2   1.2   1.3  15.2  16.9  0.2
0.2
534  0.331  0.460   2.4   7.0   4.7  42.5   1.5   0.3  15.2  32.6  5.3
1.4
```

	WS	WS/48	OBPM	DBPM	BPM	VORP	Team	Standing
8	1.8	0.066	-2.0	0.7	-1.2	0.3		Bottom
12	3.7	0.116	0.7	0.5	1.3	1.3		Bottom
14	-0.2	-0.043	-7.3	-1.5	-8.8	-0.3		Bottom
20	1.8	0.068	3.2	-0.8	2.4	1.4		Bottom
24	0.4	0.064	-1.1	0.6	-0.5	0.1		Bottom
38	3.0	0.079	-2.9	1.7	-1.2	0.3		Bottom
52	0.0	0.013	-5.8	1.6	-4.2	0.0		Bottom
59	5.8	0.161	2.3	0.6	2.8	2.1		Top
68	8.7	0.175	4.4	-0.5	3.9	3.5		Top
73	0.1	0.061	-2.2	-0.4	-2.6	0.0		Bottom
78	0.0	-0.012	-4.8	3.6	-1.2	0.0		Bottom
82	3.5	0.092	-1.9	1.2	-0.6	0.6		Top
84	0.0	-0.003	-5.1	-1.2	-6.3	0.0		Bottom
85	3.6	0.109	-2.2	3.3	1.1	1.2		Bottom
87	0.3	0.044	-3.3	0.6	-2.7	0.0		Top
97	5.5	0.130	1.0	-0.1	0.9	1.5		Top

104	-0.1	-0.018	-0.1	-0.6	-0.7	0.1	Bottom
106	7.8	0.192	7.5	0.1	7.5	4.7	Top
107	1.3	0.061	-3.4	1.6	-1.8	0.0	Bottom
114	0.1	0.021	-4.4	-0.8	-5.2	-0.2	Top
121	6.3	0.111	1.6	-0.9	0.7	1.8	Top
123	10.2	0.204	7.6	1.4	8.9	6.6	Bottom
127	0.0	-0.025	-7.0	1.8	-5.3	0.0	Bottom
128	0.4	0.076	-2.8	0.9	-1.9	0.0	Bottom
130	NaN	NaN	NaN	NaN	NaN	NaN	Top
134	1.9	0.162	0.7	1.7	2.4	0.6	Bottom
145	0.5	0.034	-3.3	-0.8	-4.1	-0.4	Bottom
151	7.4	0.146	3.4	-0.9	2.5	2.7	Top
152	3.7	0.100	-0.1	0.7	0.5	1.2	Bottom
155	7.6	0.148	3.2	-0.7	2.4	2.7	Top
162	11.4	0.226	5.7	1.5	7.3	5.6	Bottom
164	0.1	0.079	-7.8	1.7	-6.1	0.0	Top
166	2.2	0.097	-0.7	1.4	0.7	0.8	Bottom
169	2.5	0.091	0.1	-0.2	-0.1	0.6	Bottom
184	7.6	0.195	7.0	0.2	7.2	4.4	Bottom
188	8.4	0.188	5.1	0.3	5.4	4.0	Top
202	-0.5	-0.012	-3.3	-0.5	-3.8	-1.0	Bottom
211	0.9	0.053	-2.9	0.9	-2.0	0.0	Bottom
212	6.7	0.148	3.0	0.1	3.1	2.8	Top
219	0.0	-0.035	-3.8	-4.7	-8.4	0.0	Bottom
223	1.4	0.062	0.4	-1.2	-0.8	0.3	Top
231	0.0	-0.378	-12.5	-9.8	-22.2	0.0	Bottom
235	0.2	0.078	-3.7	-1.5	-5.1	-0.1	Bottom
236	0.6	0.018	-2.9	-1.1	-3.9	-0.8	Top
248	0.1	0.078	-2.8	0.4	-2.5	0.0	Bottom
253	3.4	0.082	-0.2	-0.8	-1.0	0.5	Bottom
254	5.7	0.141	1.2	0.3	1.6	1.7	Top
256	1.8	0.070	-0.7	-0.8	-1.5	0.2	Bottom
274	0.1	0.042	-4.2	0.6	-3.6	-0.1	Bottom
284	0.7	0.088	-2.7	0.9	-1.8	0.0	Top
289	0.6	0.122	-0.8	0.5	-0.3	0.1	Bottom
290	9.0	0.205	8.3	-1.2	7.1	4.9	Bottom
298	3.4	0.096	-0.7	0.8	0.2	0.9	Top
300	NaN	NaN	NaN	NaN	NaN	NaN	Bottom
303	0.5	0.021	-3.0	-1.0	-3.9	-0.6	Bottom
315	0.7	0.171	2.1	-2.1	0.0	0.1	Bottom
316	0.7	0.047	-3.6	1.4	-2.3	-0.1	Top
319	3.1	0.098	0.4	0.1	0.5	1.0	Bottom
326	1.3	0.094	-1.3	0.6	-0.7	0.2	Top
331	0.4	0.035	-1.4	-1.7	-3.1	-0.2	Top
335	1.5	0.050	-2.4	-0.4	-2.8	-0.3	Top
344	6.0	0.148	5.2	0.5	5.7	3.8	Top
347	4.2	0.120	0.8	-0.4	0.5	1.1	Bottom
350	5.1	0.116	2.6	-1.3	1.3	1.8	Top
357	1.5	0.142	-1.7	1.5	-0.2	0.2	Top

```
359  -1.2 -0.062  -5.0  -1.5  -6.5  -1.0        Bottom
378   6.2  0.156   2.0   1.2   3.2   2.5           Top
379   1.1  0.053  -0.5  -0.7  -1.2   0.2           Top
382   0.0 -0.017  -5.1  -0.8  -5.9   0.0           Top
386   3.2  0.063  -0.1  -1.8  -1.9   0.1           Top
387   2.9  0.068   1.4  -0.8   0.6   1.3        Bottom
395   0.1  0.021  -2.1  -0.5  -2.6   0.0           Top
398   0.6  0.045  -2.1  -1.2  -3.4  -0.2           Top
419  -0.3 -0.218 -11.3  -3.7 -15.0  -0.2           Top
420   0.0 -0.001  -2.2  -1.5  -3.7  -0.1           Top
423   0.7  0.062  -3.8   1.1  -2.8  -0.1           Top
424   5.1  0.106   2.1  -0.7   1.5   2.0           Top
432   NaN    NaN   NaN   NaN   NaN   NaN           Top
436   2.5  0.105   0.1  -1.5  -1.3   0.2        Bottom
442   2.2  0.097  -2.2   2.6   0.4   0.7           Top
447   3.7  0.092  -1.8   0.8  -1.0   0.5           Top
448   1.1  0.039  -3.5   1.2  -2.2  -0.1        Bottom
450  -0.5 -0.056  -5.5  -1.1  -6.6  -0.5           Top
483   6.5  0.123   2.0   0.5   2.5   2.9        Bottom
485   1.4  0.038  -3.1  -0.3  -3.4  -0.6           Top
493   0.3  0.098   0.3  -0.6  -0.3   0.1        Bottom
495   0.3  0.020  -0.8  -0.4  -1.2   0.1           Top
499  -0.1 -0.016  -5.0  -1.3  -6.3  -0.5        Bottom
505   1.9  0.044   0.3  -0.1   0.2   1.2           Top
512   0.1  0.028  -3.1  -3.6  -6.7  -0.1           Top
530   3.7  0.146   0.4   3.1   3.4   1.7        Bottom
531   0.5  0.069  -1.9  -0.8  -2.7  -0.1        Bottom
534   6.7  0.126   5.3  -2.0   3.3   3.4        Bottom
```

```
df_point_guards['Team Standing'].unique()
```

```
array(['Bottom', 'Top'], dtype=object)
```

```
df_point_guards[df_point_guards['Team Standing'] == 'Top'].describe()
```

```
                Rk        Age          G         GS           MP
FG  \
count   44.000000  44.000000  44.000000  44.000000    44.000000
44.000000
mean   301.022727  28.000000  52.068182  27.681818  1309.113636
239.022727
std    138.999162   4.856068  21.983501  28.659811   847.741163
196.011923
min     60.000000  20.000000   1.000000   0.000000    32.000000
1.000000
25%    163.750000  24.000000  39.000000   0.750000   550.750000
61.000000
50%    335.000000  26.500000  58.000000  18.500000  1552.000000
212.000000
75%    421.250000  32.000000  68.000000  58.250000  1996.750000
```

```
386.250000
max    514.000000  37.000000  82.000000  79.000000  2725.000000
682.000000

                FGA         FG%          3P          3PA         3P%
2P  \
count    44.000000  44.000000   44.000000   44.000000  44.000000
44.000000
mean    533.954545   0.428318   83.795455  228.772727   0.332909
155.227273
std     412.828528   0.050405   69.454546  177.596206   0.078328
141.212501
min       3.000000   0.333000    0.000000    2.000000   0.000000
0.000000
25%     155.500000   0.404750   17.250000   57.750000   0.310250
39.500000
50%     525.000000   0.422500   81.500000  243.000000   0.335000
114.500000
75%     868.500000   0.456000  132.500000  344.000000   0.372000
229.250000
max    1331.000000   0.566000  273.000000  639.000000   0.444000
563.000000

               2PA         2P%        eFG%          FT          FTA
FT%  \
count   44.000000  43.000000  44.000000   44.000000   44.000000
43.000000
mean   305.181818   0.492953   0.505159  110.295455  135.863636
0.765279
std    264.679119   0.067934   0.052478  115.219221  140.514435
0.172912
min      0.000000   0.333000   0.375000    0.000000    0.000000
0.000000
25%     88.500000   0.458000   0.478750   22.500000   27.000000
0.741500
50%    231.000000   0.494000   0.512000   69.500000   78.000000
0.829000
75%    440.750000   0.521500   0.537500  179.250000  216.000000
0.861000
max    964.000000   0.662000   0.614000  372.000000  497.000000
1.000000

              ORB          DRB          TRB          AST         STL
BLK  \
count  44.000000   44.000000   44.000000   44.000000  44.000000
44.000000
mean   25.613636  130.772727  156.386364  247.522727  43.590909
12.318182
std    20.047868   97.905713  115.929653  185.133703  29.020697
9.780790
```

```
min     0.000000    2.000000    4.000000    2.000000   0.000000
0.000000
25%     9.750000   36.750000   49.000000   70.750000  14.750000
4.000000
50%    25.000000  118.500000  144.500000  205.500000  46.000000
9.500000
75%    39.250000  203.000000  243.500000  418.000000  66.250000
21.000000
max    89.000000  334.000000  423.000000  618.000000  93.000000
33.000000

               TOV          PF           PTS         PER         TS%
3PAr  \
count   44.000000   44.000000    44.000000  42.000000  42.000000
42.000000
mean    93.681818   90.431818   672.136364  13.461905   0.539500
0.443976
std     71.447093   59.863176   560.178526   5.121640   0.057527
0.172056
min      2.000000    3.000000     3.000000  -1.400000   0.407000
0.009000
25%     28.750000   32.000000   160.000000  10.300000   0.509000
0.369250
50%     81.500000  101.500000   628.000000  12.850000   0.545500
0.454500
75%    143.500000  140.750000  1173.250000  16.250000   0.580500
0.549000
max    255.000000  214.000000  1826.000000  24.100000   0.656000
1.000000

              FTr        ORB%        DRB%        TRB%        AST%        STL
%  \
count  42.000000  42.000000  42.000000  42.000000  42.000000
42.000000
mean    0.226857   2.345238  11.009524   6.723810  25.666667
1.678571
std     0.137736   1.354523   3.706146   2.192316   8.292479
0.614673
min     0.000000   0.000000   5.400000   3.500000   8.200000
0.000000
25%     0.140250   1.600000   8.100000   5.125000  20.250000
1.300000
50%     0.217500   2.000000  10.350000   6.600000  26.750000
1.700000
75%     0.306750   2.700000  12.850000   7.650000  30.675000
1.950000
max     0.750000   6.900000  22.200000  13.400000  43.300000
3.500000

             BLK%        TOV%        USG%         OWS         DWS
```

```
WS  \
count  42.000000  42.000000  42.000000  42.000000  42.000000
42.000000
mean    0.900000  15.823810  21.109524   1.571429   1.335714
2.911905
std     0.566116   6.826412   6.153754   2.179314   0.925472
2.885304
min     0.000000   8.900000   5.100000  -0.800000   0.000000  -
0.500000
25%     0.525000  11.825000  16.900000  -0.075000   0.450000
0.450000
50%     0.900000  13.950000  20.300000   0.600000   1.300000
1.500000
75%     1.275000  16.750000  25.975000   3.400000   1.975000
5.650000
max     2.600000  41.400000  34.900000   6.900000   3.000000
8.700000

           WS/48       OBPM       DBPM        BPM       VORP
count  42.000000  42.000000  42.000000  42.000000  42.000000
mean    0.076310  -0.773810  -0.230952  -1.007143   0.933333
std     0.074729   3.617104   1.274992   4.011167   1.414329
min    -0.218000 -11.300000  -3.700000 -15.000000  -0.800000
25%     0.039500  -2.850000  -0.900000  -3.025000  -0.100000
50%     0.083500  -1.350000  -0.350000  -0.900000   0.200000
75%     0.138250   1.900000   0.600000   1.450000   1.800000
max     0.192000   7.500000   2.600000   7.500000   4.700000
```

```
df_point_guards[df_point_guards['Team Standing'] ==
'Bottom'].describe()
```

```
               Rk        Age          G         GS            MP
FG  \
count   49.000000  49.000000  49.000000  49.000000    49.000000
49.000000
mean   229.959184  24.836735  42.061224  20.020408  1004.306122
176.326531
std    155.039640   3.630582  25.131660  26.056421   805.922096
187.206257
min      9.000000  19.000000   1.000000   0.000000     5.000000
0.000000
25%    109.000000  22.000000  20.000000   1.000000   189.000000
25.000000
50%    213.000000  24.000000  53.000000   6.000000  1042.000000
116.000000
75%    317.000000  27.000000  62.000000  36.000000  1575.000000
277.000000
max    536.000000  34.000000  76.000000  73.000000  2541.000000
719.000000
```

```
                FGA         FG%          3P         3PA         3P%
2P  \
count    49.000000   49.000000   49.000000   49.000000   49.000000
49.00000
mean    394.489796    0.410265   52.877551  152.979592    0.329408
123.44898
std     399.940002    0.101227   59.717262  166.707390    0.089684
144.19360
min       3.000000    0.000000    0.000000    1.000000    0.000000
0.00000
25%      57.000000    0.400000    8.000000   26.000000    0.286000
20.00000
50%     254.000000    0.427000   28.000000   95.000000    0.335000
74.00000
75%     521.000000    0.463000   78.000000  206.000000    0.371000
189.00000
max    1449.000000    0.543000  244.000000  658.000000    0.500000
646.00000

                2PA         2P%        eFG%          FT          FTA
FT%  \
count    49.000000   49.000000   49.000000   49.000000   49.000000
45.000000
mean    241.510204    0.450816    0.478898   96.040816  115.000000
0.772844
std     271.216021    0.159469    0.103404  153.915826  178.257538
0.118295
min       1.000000    0.000000    0.000000    0.000000    0.000000
0.500000
25%      41.000000    0.452000    0.464000   12.000000   18.000000
0.723000
50%     135.000000    0.488000    0.500000   42.000000   55.000000
0.792000
75%     346.000000    0.543000    0.534000   98.000000  122.000000
0.853000
max    1213.000000    0.625000    0.588000  669.000000  739.000000
1.000000

              ORB          DRB          TRB          AST          STL
BLK  \
count  49.000000   49.000000   49.000000   49.000000   49.000000
49.000000
mean   24.714286  108.387755  133.102041  183.877551   41.183673
12.979592
std    21.716929  101.475945  119.442846  183.042303   36.332190
14.098596
min     0.000000    0.000000    0.000000    1.000000    0.000000
0.000000
25%     5.000000   17.000000   23.000000   23.000000    6.000000
```

```
2.000000
50%    21.000000   91.000000  106.000000  134.000000   37.000000
10.000000
75%    42.000000  172.000000  217.000000  304.000000   75.000000
17.000000
max    76.000000  515.000000  569.000000  741.000000  123.000000
65.000000

              TOV          PF          PTS         PER         TS%
3PAr  \
count   49.000000   49.000000    49.000000  48.000000  48.000000
48.000000
mean    68.265306   75.387755   501.571429  13.050000   0.514167
0.420729
std     70.848658   61.449104   567.714240   6.506429   0.113475
0.171987
min      0.000000    0.000000     0.000000  -6.800000   0.000000
0.113000
25%      7.000000   17.000000    72.000000   8.975000   0.492000
0.319750
50%     56.000000   69.000000   295.000000  12.450000   0.538000
0.400000
75%     91.000000  112.000000   685.000000  16.225000   0.578250
0.516750
max    300.000000  219.000000  2138.000000  28.700000   0.645000
0.889000

              FTr        ORB%        DRB%        TRB%        AST%        STL
%  \
count  48.000000  48.000000  48.000000  48.000000  48.000000
48.000000
mean    0.230833   3.122917  11.243750   7.139583  23.222917
1.947917
std     0.142413   3.399827   4.418814   3.251038   9.381018
0.986287
min     0.000000   0.000000   0.000000   0.000000   7.800000
0.000000
25%     0.148750   1.775000   8.625000   5.100000  15.375000
1.400000
50%     0.194500   2.400000  10.800000   6.600000  20.850000
1.700000
75%     0.284000   3.325000  13.400000   8.500000  28.525000
2.325000
max     0.535000  21.900000  25.400000  21.600000  47.600000
5.700000

             BLK%        TOV%        USG%         OWS         DWS
WS  \
count  48.000000  48.000000  48.000000  48.000000  48.000000
48.000000
```

```
mean    1.060417  13.166667  19.489583   1.262500   0.875000
2.141667
std     0.888398   4.637146   6.549460   2.320182   0.847148
2.873935
min     0.000000   0.000000  10.600000  -1.900000   0.000000  -
1.200000
25%     0.500000  10.175000  15.350000   0.000000   0.100000
0.100000
50%     0.850000  12.850000  19.200000   0.250000   0.600000
1.000000
75%     1.425000  15.350000  22.375000   1.900000   1.400000
3.175000
max     4.500000  25.400000  37.600000   8.400000   3.000000
11.400000

           WS/48       OBPM       DBPM        BPM       VORP
count  48.000000  48.000000  48.000000  48.000000  48.000000
mean    0.065250  -1.208333  -0.131250  -1.329167   0.787500
std     0.093809   3.949351   2.088179   4.913029   1.641435
min    -0.378000 -12.500000  -9.800000 -22.200000  -1.000000
25%     0.030750  -3.425000  -0.850000  -3.650000   0.000000
50%     0.073000  -1.500000  -0.050000  -1.200000   0.100000
75%     0.110750   0.475000   0.975000   0.625000   1.125000
max     0.226000   8.300000   3.600000   8.900000   6.600000

df_point_guards.describe()

                Rk         Age           G          GS            MP
FG  \
count   93.000000  93.000000  93.000000  93.000000    93.000000
93.000000
mean   263.580645  26.333333  46.795699  23.645161  1148.516129
205.989247
std    151.142628   4.518881  24.096527  27.439517   835.574244
192.958460
min      9.000000  19.000000   1.000000   0.000000     5.000000
0.000000
25%    130.000000  23.000000  27.000000   1.000000   335.000000
49.000000
50%    256.000000  25.000000  56.000000   7.000000  1106.000000
144.000000
75%    388.000000  30.000000  67.000000  56.000000  1940.000000
307.000000
max    536.000000  37.000000  82.000000  79.000000  2725.000000
719.000000

                FGA         FG%          3P         3PA         3P%
2P  \
count    93.000000  93.000000   93.000000   93.00000  93.000000
93.000000
```

```
mean    460.473118   0.418806   67.505376  188.83871   0.331065
138.483871
std     409.891147   0.081337   66.001091  175.18309   0.084066
142.907663
min       3.000000   0.000000    0.000000    1.00000   0.000000
0.000000
25%     110.000000   0.402000   11.000000   28.00000   0.303000
37.000000
50%     351.000000   0.423000   47.000000  134.00000   0.335000
97.000000
75%     710.000000   0.462000  117.000000  315.00000   0.371000
210.000000
max    1449.000000   0.566000  273.000000  658.00000   0.500000
646.000000

                2PA        2P%       eFG%          FT         FTA
FT%  \
count    93.000000  92.000000  93.000000   93.000000   93.000000
88.000000
mean    271.634409   0.470511   0.491323  102.784946  124.870968
0.769148
std     268.594198   0.126454   0.083903  136.440770  160.986512
0.146716
min       0.000000   0.000000   0.000000    0.000000    0.000000
0.000000
25%      68.000000   0.454250   0.464000   13.000000   18.000000
0.732750
50%     180.000000   0.492500   0.503000   49.000000   64.000000
0.811000
75%     425.000000   0.535250   0.537000  148.000000  170.000000
0.859250
max    1213.000000   0.662000   0.614000  669.000000  739.000000
1.000000

              ORB         DRB         TRB         AST         STL
BLK  \
count  93.000000   93.000000   93.000000   93.000000   93.000000
93.000000
mean   25.139785  118.978495  144.118280  213.989247   42.322581
12.666667
std    20.835584   99.894942  117.735504  185.798073   32.921237
12.187247
min     0.000000    0.000000    0.000000    1.000000    0.000000
0.000000
25%     8.000000   32.000000   44.000000   50.000000   10.000000
3.000000
50%    25.000000  109.000000  123.000000  172.000000   39.000000
10.000000
75%    40.000000  189.000000  225.000000  369.000000   67.000000
```

```
20.000000
max    89.000000  515.000000  569.000000  741.000000  123.000000
65.000000

              TOV          PF          PTS        PER         TS%
3PAr  \
count   93.000000   93.000000    93.000000  90.000000  90.000000
90.000000
mean    80.290323   82.505376   582.268817  13.242222   0.525989
0.431578
std     71.885716   60.844532   567.586579   5.872196   0.092120
0.171447
min      0.000000    0.000000     0.000000  -6.800000   0.000000
0.009000
25%     18.000000   24.000000   114.000000   9.925000   0.498500
0.331500
50%     62.000000   79.000000   407.000000  12.600000   0.540000
0.422000
75%    129.000000  126.000000   830.000000  16.275000   0.578750
0.536250
max    300.000000  219.000000  2138.000000  28.700000   0.656000
1.000000

             FTr        ORB%        DRB%        TRB%        AST%        STL
%  \
count  90.000000  90.000000  90.000000  90.000000  90.000000
90.000000
mean    0.228978   2.760000  11.134444   6.945556  24.363333
1.822222
std     0.139477   2.664869   4.080790   2.799846   8.924974
0.840248
min     0.000000   0.000000   0.000000   0.000000   7.800000
0.000000
25%     0.141500   1.700000   8.175000   5.100000  16.600000
1.325000
50%     0.201000   2.300000  10.650000   6.600000  23.400000
1.700000
75%     0.299750   3.200000  13.325000   8.175000  29.950000
2.275000
max     0.750000  21.900000  25.400000  21.600000  47.600000
5.700000

             BLK%        TOV%        USG%         OWS         DWS
WS  \
count  90.000000  90.000000  90.000000  90.000000  90.000000
90.000000
mean    0.985556  14.406667  20.245556   1.406667   1.090000
2.501111
std     0.755588   5.882173   6.384218   2.248285   0.909383
2.888975
```

```
min     0.000000   0.000000   5.100000  -1.900000   0.000000  -
1.200000
25%     0.500000  11.475000  15.900000   0.000000   0.225000
0.225000
50%     0.900000  13.350000  19.600000   0.300000   0.950000
1.400000
75%     1.300000  15.900000  24.275000   2.100000   1.800000
3.700000
max     4.500000  41.400000  37.600000   8.400000   3.000000
11.400000

           WS/48       OBPM       DBPM        BPM       VORP
count  90.000000  90.000000  90.000000  90.000000  90.000000
mean    0.070411  -1.005556  -0.177778  -1.178889   0.855556
std     0.085150   3.783060   1.747600   4.492778   1.532869
min    -0.378000 -12.500000  -9.800000 -22.200000  -1.000000
25%     0.035750  -3.250000  -0.900000  -3.325000   0.000000
50%     0.077000  -1.350000  -0.250000  -1.100000   0.150000
75%     0.119000   0.950000   0.800000   1.050000   1.300000
max     0.226000   8.300000   3.600000   8.900000   6.600000
```

```
df_point_guards[df_point_guards['MP'] >= 1940]['Team
Standing'].value_counts()
```

```
Team Standing
Top       16
Bottom     8
Name: count, dtype: int64
```

Filtering players based on Minutes Played

```
df_point_guards['Team Standing'].value_counts()
```

```
Team Standing
Bottom    49
Top       44
Name: count, dtype: int64
```

```
df_point_guards[(df_point_guards['MP'] >= 1996.75)
                & (df_point_guards['Team Standing'] == 'Top')]['Team
Standing'].value_counts()

# df_point_guards[(df_point_guards['MPG'] >= 32)
#                 & (df_point_guards['Team Standing'] == 'Top')]['Team
Standing'].value_counts()
```

```
Team Standing
Top    11
Name: count, dtype: int64
```

```
df_point_guards[(df_point_guards['MP'] >= 1700)
                & (df_point_guards['Team Standing'] == 'Bottom')]
['Team Standing'].value_counts()

# df_point_guards[(df_point_guards['MPG'] >= 30)
#                 & (df_point_guards['Team Standing'] == 'Bottom')]
['Team Standing'].value_counts()
```

```
Team Standing
Bottom    11
Name: count, dtype: int64
```

```
df_point_guards_final = df_point_guards[((df_point_guards['MP'] >=
1996.75) & (df_point_guards['Team Standing'] == 'Top')) |
                                        ((df_point_guards['MP'] >=
1700) & (df_point_guards['Team Standing'] == 'Bottom'))]

# df_point_guards_final = df_point_guards[((df_point_guards['MPG'] >=
32) & (df_point_guards['Team Standing'] == 'Top')) |
#                                         ((df_point_guards['MPG'] >=
30) & (df_point_guards['Team Standing'] == 'Bottom'))]

df_point_guards_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 22 entries, 38 to 534
Data columns (total 52 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Rk                 22 non-null     int64
 1   Player             22 non-null     object
 2   Pos                22 non-null     object
 3   Age                22 non-null     int64
 4   Tm                 22 non-null     object
 5   G                  22 non-null     int64
 6   GS                 22 non-null     int64
 7   MP                 22 non-null     int64
 8   FG                 22 non-null     int64
 9   FGA                22 non-null     int64
 10  FG%                22 non-null     float64
 11  3P                 22 non-null     int64
 12  3PA                22 non-null     int64
 13  3P%                22 non-null     float64
 14  2P                 22 non-null     int64
 15  2PA                22 non-null     int64
 16  2P%                22 non-null     float64
 17  eFG%               22 non-null     float64
 18  FT                 22 non-null     int64
 19  FTA                22 non-null     int64
 20  FT%                22 non-null     float64
```

```
 21  ORB                 22 non-null     int64
 22  DRB                 22 non-null     int64
 23  TRB                 22 non-null     int64
 24  AST                 22 non-null     int64
 25  STL                 22 non-null     int64
 26  BLK                 22 non-null     int64
 27  TOV                 22 non-null     int64
 28  PF                  22 non-null     int64
 29  PTS                 22 non-null     int64
 30  Player-additional   22 non-null     object
 31  PER                 22 non-null     float64
 32  TS%                 22 non-null     float64
 33  3PAr                22 non-null     float64
 34  FTr                 22 non-null     float64
 35  ORB%                22 non-null     float64
 36  DRB%                22 non-null     float64
 37  TRB%                22 non-null     float64
 38  AST%                22 non-null     float64
 39  STL%                22 non-null     float64
 40  BLK%                22 non-null     float64
 41  TOV%                22 non-null     float64
 42  USG%                22 non-null     float64
 43  OWS                 22 non-null     float64
 44  DWS                 22 non-null     float64
 45  WS                  22 non-null     float64
 46  WS/48               22 non-null     float64
 47  OBPM                22 non-null     float64
 48  DBPM                22 non-null     float64
 49  BPM                 22 non-null     float64
 50  VORP                22 non-null     float64
 51  Team Standing       22 non-null     object
dtypes: float64(25), int64(22), object(5)
memory usage: 9.1+ KB
```

df_point_guards_final

	Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	\
38	39	Patrick Beverley	SG-PG	34	CHI	67	67	1816	144	360	
68	69	Jalen Brunson	PG	26	NYK	68	68	2379	587	1195	
97	99	Mike Conley	PG	35	MIN	67	66	2029	261	610	
121	123	Spencer Dinwiddie	SG-PG	29	BRK	79	79	2725	460	1050	
123	125	Luka Dončić	PG	23	DAL	66	66	2391	719	1449	
151	153	De'Aaron Fox	PG	25	SAC	73	73	2435	682	1331	

```
152  154           Markelle Fultz     PG   24  ORL  60  60  1778  349
679
155  157           Darius Garland     PG   23  CLE  69  69  2447  522
1129
162  164  Shai Gilgeous-Alexander     PG   24  OKC  68  68  2416  704
1381
184  186        Tyrese Haliburton     PG   22  IND  56  56  1883  412
841
188  190             James Harden     PG   33  PHI  58  58  2135  371
842
202  204            Killian Hayes     PG   21  DET  76  56  2154  307
815
212  214             Jrue Holiday     PG   32  MIL  67  65  2183  490
1023
253  255                Tre Jones     PG   23  SAS  68  65  1984  341
743
290  292           Damian Lillard     PG   32  POR  58  58  2107  556
1202
350  352             Jamal Murray     PG   25  DEN  65  65  2133  473
1041
386  388             Jordan Poole     PG   23  GSW  82  43  2458  550
1278
387  389         Kevin Porter Jr.     PG   22  HOU  59  59  2024  391
884
424  426         D'Angelo Russell     PG   26  LAL  71  71  2304  445
948
483  485            Fred VanVleet     PG   28  TOR  69  69  2535  437
1112
505  507        Russell Westbrook     PG   34  LAC  73  24  2126  432
991
534  536               Trae Young     PG   24  ATL  73  73  2541  597
1390

       FG%   3P  3PA    3P%   2P   2PA    2P%   eFG%   FT  FTA    FT%
ORB  \
38   0.400   80  239  0.335   64   121  0.529  0.511   47   65  0.723
44
68   0.491  134  322  0.416  453   873  0.519  0.547  325  392  0.829
40
97   0.428  135  351  0.385  126   259  0.486  0.539  141  169  0.834
33
121  0.438  181  490  0.369  279   560  0.498  0.524  268  330  0.812
28
123  0.496  185  541  0.342  534   908  0.588  0.560  515  694  0.742
54
151  0.512  119  367  0.324  563   964  0.584  0.557  343  440  0.780
40
152  0.514   27   87  0.310  322   592  0.544  0.534  112  143  0.783
69
```

```
155  0.462  169  412  0.410  353   717  0.492  0.537  277  321  0.863
28
162  0.510   58  168  0.345  646  1213  0.533  0.531  669  739  0.905
59
184  0.490  161  402  0.400  251   439  0.572  0.586  175  201  0.871
33
188  0.441  161  418  0.385  210   424  0.495  0.536  313  361  0.867
40
202  0.377   80  286  0.280  227   529  0.429  0.426   92  112  0.821
28
212  0.479  158  411  0.384  332   612  0.542  0.556  152  177  0.859
79
253  0.459   45  158  0.285  296   585  0.506  0.489  148  172  0.860
56
290  0.463  244  658  0.371  312   544  0.574  0.564  510  558  0.914
44
350  0.454  172  432  0.398  301   609  0.494  0.537  180  216  0.833
48
386  0.430  214  637  0.336  336   641  0.524  0.514  361  415  0.870
32
387  0.442  141  385  0.366  250   499  0.501  0.522  207  264  0.784
76
424  0.469  194  490  0.396  251   458  0.548  0.572  179  216  0.829
35
483  0.393  207  606  0.342  230   506  0.455  0.486  254  283  0.898
30
505  0.436   89  286  0.311  343   705  0.487  0.481  206  314  0.656
89
534  0.429  154  460  0.335  443   930  0.476  0.485  566  639  0.886
56

     DRB  TRB  AST  STL  BLK  TOV   PF   PTS Player-additional   PER
TS%  \
38   203  247  194   63   41   60  187   415         beverpa01   8.9
0.534
68   201  241  421   61   15  142  152  1633         brunsja01  21.2
0.597
97   149  182  450   73   14  103  139   798         conlemi01  14.7
0.583
121  242  270  515   67   24  145  187  1369         dinwisp01  16.0
0.573
123  515  569  529   90   33  236  166  2138         doncilu01  28.7
0.609
151  266  306  447   83   23  181  172  1826           foxde01  21.8
0.599
152  165  234  341   87   26  139  131   837         fultzma01  16.6
0.564
155  157  185  538   85    9  199  148  1490         garlada01  18.8
0.587
```

```
162  270  329  371  112   65  192  192  2135         gilgesh01  27.2
0.626
184  172  205  585   91   25  141   69  1160         halibty01  23.6
0.624
188  314  354  618   71   31  195  112  1216         hardeja01  21.6
0.607
202  193  221  470  104   28  173  219   786         hayeski01  10.2
0.455
212  262  341  495   79   25  197  116  1290         holidjr01  19.2
0.586
253  189  245  448   89    9  110   98   875         jonestr01  16.0
0.534
290  233  277  425   50   18  191  109  1866         lillada01  26.7
0.645
350  209  257  400   66   16  145  103  1298         murraja01  18.0
0.571
386  193  225  369   63   21  252  214  1675         poolejo01  14.6
0.573
387  238  314  338   82   17  188  156  1130         porteke02  16.2
0.565
424  180  215  437   70   29  186  140  1263         russeda01  16.3
0.605
483  250  280  495  123   38  140  193  1335         vanvlfr01  17.0
0.540
505  334  423  551   76   33  255  162  1159         westbru01  16.1
0.513
534  161  217  741   80    9  300  104  1914         youngtr01  22.0
0.573

      3PAr    FTr  ORB%  DRB%  TRB%  AST%  STL%  BLK%  TOV%  USG%  OWS
DWS  \
38   0.664  0.181   2.7  12.0   7.5  13.3   1.7   2.0  13.4  10.6  0.8
2.1
68   0.269  0.328   1.8   9.3   5.6  28.7   1.3   0.6   9.4  27.2  6.9
1.8
97   0.575  0.277   1.8   7.9   4.9  29.4   1.7   0.6  13.1  16.3  4.1
1.4
121  0.467  0.314   1.2  10.2   5.7  28.1   1.2   0.8  10.8  22.1  4.5
1.8
123  0.373  0.479   2.6  25.4  13.8  42.3   1.9   1.2  11.9  37.6  7.3
2.9
151  0.276  0.331   1.9  12.4   7.2  29.6   1.6   0.8  10.6  30.1  5.5
1.8
152  0.128  0.211   4.4  10.5   7.4  29.8   2.4   1.4  15.8  21.3  1.7
2.0
155  0.365  0.284   1.3   7.6   4.5  34.1   1.7   0.3  13.5  26.9  4.5
3.0
162  0.122  0.535   2.6  12.2   7.3  25.7   2.2   2.5  10.1  32.8  8.4
3.0
```

184	0.478	0.239	1.9	10.1	6.0	47.6	2.3	1.1	13.2	23.8	6.4
1.2											
188	0.496	0.429	2.3	17.1	9.8	43.3	1.6	1.4	16.3	25.0	5.8
2.6											
202	0.351	0.137	1.4	10.2	5.7	32.3	2.3	1.1	16.7	20.5	-1.9
1.4											
212	0.402	0.173	3.9	12.0	8.1	34.4	1.7	0.9	15.2	25.0	3.9
2.8											
253	0.213	0.231	3.0	10.8	6.7	31.5	2.1	0.4	11.8	19.3	2.6
0.8											
290	0.547	0.464	2.4	12.8	7.6	35.0	1.2	0.8	11.7	33.8	8.2
0.8											
350	0.415	0.207	2.7	11.0	6.9	27.5	1.5	0.7	11.3	26.1	3.3
1.9											
386	0.498	0.325	1.5	8.5	5.0	22.5	1.2	0.8	14.7	29.2	1.4
1.9											
387	0.436	0.299	4.1	13.1	8.5	25.7	2.0	0.8	15.8	24.3	1.9
1.0											
424	0.517	0.228	1.7	8.5	5.2	27.4	1.4	1.1	15.1	22.7	3.1
1.9											
483	0.545	0.254	1.2	12.0	6.3	28.1	2.4	1.5	10.2	23.2	3.7
2.8											
505	0.289	0.317	4.7	16.5	10.8	38.6	1.7	1.3	18.4	27.7	-0.6
2.6											
534	0.331	0.460	2.4	7.0	4.7	42.5	1.5	0.3	15.2	32.6	5.3
1.4											

	WS	WS/48	OBPM	DBPM	BPM	VORP	Team	Standing
38	3.0	0.079	-2.9	1.7	-1.2	0.3		Bottom
68	8.7	0.175	4.4	-0.5	3.9	3.5		Top
97	5.5	0.130	1.0	-0.1	0.9	1.5		Top
121	6.3	0.111	1.6	-0.9	0.7	1.8		Top
123	10.2	0.204	7.6	1.4	8.9	6.6		Bottom
151	7.4	0.146	3.4	-0.9	2.5	2.7		Top
152	3.7	0.100	-0.1	0.7	0.5	1.2		Bottom
155	7.6	0.148	3.2	-0.7	2.4	2.7		Top
162	11.4	0.226	5.7	1.5	7.3	5.6		Bottom
184	7.6	0.195	7.0	0.2	7.2	4.4		Bottom
188	8.4	0.188	5.1	0.3	5.4	4.0		Top
202	-0.5	-0.012	-3.3	-0.5	-3.8	-1.0		Bottom
212	6.7	0.148	3.0	0.1	3.1	2.8		Top
253	3.4	0.082	-0.2	-0.8	-1.0	0.5		Bottom
290	9.0	0.205	8.3	-1.2	7.1	4.9		Bottom
350	5.1	0.116	2.6	-1.3	1.3	1.8		Top
386	3.2	0.063	-0.1	-1.8	-1.9	0.1		Top
387	2.9	0.068	1.4	-0.8	0.6	1.3		Bottom
424	5.1	0.106	2.1	-0.7	1.5	2.0		Top
483	6.5	0.123	2.0	0.5	2.5	2.9		Bottom

```
505   1.9  0.044   0.3  -0.1  0.2   1.2           Top
534   6.7  0.126   5.3  -2.0  3.3   3.4        Bottom

df_point_guards_final['Pos'] = 'PG'
```

T-test for determining winning metrics

```
df_point_guards_final.columns
```

```
Index(['Rk', 'Player', 'Pos', 'Age', 'Tm', 'G', 'GS', 'MP', 'FG',
'FGA', 'FG%',
       '3P', '3PA', '3P%', '2P', '2PA', '2P%', 'eFG%', 'FT', 'FTA',
'FT%',
       'ORB', 'DRB', 'TRB', 'AST', 'STL', 'BLK', 'TOV', 'PF', 'PTS',
       'Player-additional', 'PER', 'TS%', '3PAr', 'FTr', 'ORB%', 'DRB
%',
       'TRB%', 'AST%', 'STL%', 'BLK%', 'TOV%', 'USG%', 'OWS', 'DWS',
'WS',
       'WS/48', 'OBPM', 'DBPM', 'BPM', 'VORP', 'Team Standing'],
      dtype='object')
```

```
df_point_guards_final.sort_values(by = 'MP')
```

```
      Rk                    Player Pos  Age   Tm   G  GS    MP   FG
FGA  \
152  154           Markelle Fultz  PG   24  ORL  60  60  1778  349
679
38    39         Patrick Beverley  PG   34  CHI  67  67  1816  144
360
184  186        Tyrese Haliburton  PG   22  IND  56  56  1883  412
841
253  255                Tre Jones  PG   23  SAS  68  65  1984  341
743
387  389         Kevin Porter Jr.  PG   22  HOU  59  59  2024  391
884
97    99              Mike Conley  PG   35  MIN  67  66  2029  261
610
290  292           Damian Lillard  PG   32  POR  58  58  2107  556
1202
505  507        Russell Westbrook  PG   34  LAC  73  24  2126  432
991
350  352             Jamal Murray  PG   25  DEN  65  65  2133  473
1041
188  190             James Harden  PG   33  PHI  58  58  2135  371
842
202  204            Killian Hayes  PG   21  DET  76  56  2154  307
815
212  214             Jrue Holiday  PG   32  MIL  67  65  2183  490
1023
424  426         D'Angelo Russell  PG   26  LAL  71  71  2304  445
```

```
948
68    69            Jalen Brunson  PG   26  NYK  68  68  2379  587
1195
123  125              Luka Dončić  PG   23  DAL  66  66  2391  719
1449
162  164  Shai Gilgeous-Alexander  PG   24  OKC  68  68  2416  704
1381
151  153             De'Aaron Fox  PG   25  SAC  73  73  2435  682
1331
155  157           Darius Garland  PG   23  CLE  69  69  2447  522
1129
386  388             Jordan Poole  PG   23  GSW  82  43  2458  550
1278
483  485            Fred VanVleet  PG   28  TOR  69  69  2535  437
1112
534  536               Trae Young  PG   24  ATL  73  73  2541  597
1390
121  123        Spencer Dinwiddie  PG   29  BRK  79  79  2725  460
1050

       FG%   3P  3PA    3P%   2P   2PA    2P%   eFG%   FT  FTA    FT%
ORB  \
152  0.514   27   87  0.310  322   592  0.544  0.534  112  143  0.783
69
38   0.400   80  239  0.335   64   121  0.529  0.511   47   65  0.723
44
184  0.490  161  402  0.400  251   439  0.572  0.586  175  201  0.871
33
253  0.459   45  158  0.285  296   585  0.506  0.489  148  172  0.860
56
387  0.442  141  385  0.366  250   499  0.501  0.522  207  264  0.784
76
97   0.428  135  351  0.385  126   259  0.486  0.539  141  169  0.834
33
290  0.463  244  658  0.371  312   544  0.574  0.564  510  558  0.914
44
505  0.436   89  286  0.311  343   705  0.487  0.481  206  314  0.656
89
350  0.454  172  432  0.398  301   609  0.494  0.537  180  216  0.833
48
188  0.441  161  418  0.385  210   424  0.495  0.536  313  361  0.867
40
202  0.377   80  286  0.280  227   529  0.429  0.426   92  112  0.821
28
212  0.479  158  411  0.384  332   612  0.542  0.556  152  177  0.859
79
424  0.469  194  490  0.396  251   458  0.548  0.572  179  216  0.829
35
68   0.491  134  322  0.416  453   873  0.519  0.547  325  392  0.829
```

```
40
123  0.496  185  541  0.342  534   908  0.588  0.560  515  694  0.742
54
162  0.510   58  168  0.345  646  1213  0.533  0.531  669  739  0.905
59
151  0.512  119  367  0.324  563   964  0.584  0.557  343  440  0.780
40
155  0.462  169  412  0.410  353   717  0.492  0.537  277  321  0.863
28
386  0.430  214  637  0.336  336   641  0.524  0.514  361  415  0.870
32
483  0.393  207  606  0.342  230   506  0.455  0.486  254  283  0.898
30
534  0.429  154  460  0.335  443   930  0.476  0.485  566  639  0.886
56
121  0.438  181  490  0.369  279   560  0.498  0.524  268  330  0.812
28

     DRB  TRB  AST  STL  BLK  TOV   PF   PTS Player-additional   PER
TS%  \
152  165  234  341   87   26  139  131   837         fultzma01  16.6
0.564
38   203  247  194   63   41   60  187   415         beverpa01   8.9
0.534
184  172  205  585   91   25  141   69  1160         halibty01  23.6
0.624
253  189  245  448   89    9  110   98   875         jonestr01  16.0
0.534
387  238  314  338   82   17  188  156  1130         porteke02  16.2
0.565
97   149  182  450   73   14  103  139   798         conlemi01  14.7
0.583
290  233  277  425   50   18  191  109  1866         lillada01  26.7
0.645
505  334  423  551   76   33  255  162  1159         westbru01  16.1
0.513
350  209  257  400   66   16  145  103  1298         murraja01  18.0
0.571
188  314  354  618   71   31  195  112  1216         hardeja01  21.6
0.607
202  193  221  470  104   28  173  219   786         hayeski01  10.2
0.455
212  262  341  495   79   25  197  116  1290         holidjr01  19.2
0.586
424  180  215  437   70   29  186  140  1263         russeda01  16.3
0.605
68   201  241  421   61   15  142  152  1633         brunsja01  21.2
0.597
123  515  569  529   90   33  236  166  2138         doncilu01  28.7
```

```
0.609
162  270  329  371  112   65  192  192  2135          gilgesh01  27.2
0.626
151  266  306  447   83   23  181  172  1826            foxde01  21.8
0.599
155  157  185  538   85    9  199  148  1490          garlada01  18.8
0.587
386  193  225  369   63   21  252  214  1675          poolejo01  14.6
0.573
483  250  280  495  123   38  140  193  1335          vanvlfr01  17.0
0.540
534  161  217  741   80    9  300  104  1914          youngtr01  22.0
0.573
121  242  270  515   67   24  145  187  1369          dinwisp01  16.0
0.573

      3PAr    FTr  ORB%  DRB%  TRB%  AST%  STL%  BLK%  TOV%  USG%  OWS
DWS  \
152  0.128  0.211   4.4  10.5   7.4  29.8   2.4   1.4  15.8  21.3  1.7
2.0
38   0.664  0.181   2.7  12.0   7.5  13.3   1.7   2.0  13.4  10.6  0.8
2.1
184  0.478  0.239   1.9  10.1   6.0  47.6   2.3   1.1  13.2  23.8  6.4
1.2
253  0.213  0.231   3.0  10.8   6.7  31.5   2.1   0.4  11.8  19.3  2.6
0.8
387  0.436  0.299   4.1  13.1   8.5  25.7   2.0   0.8  15.8  24.3  1.9
1.0
97   0.575  0.277   1.8   7.9   4.9  29.4   1.7   0.6  13.1  16.3  4.1
1.4
290  0.547  0.464   2.4  12.8   7.6  35.0   1.2   0.8  11.7  33.8  8.2
0.8
505  0.289  0.317   4.7  16.5  10.8  38.6   1.7   1.3  18.4  27.7 -0.6
2.6
350  0.415  0.207   2.7  11.0   6.9  27.5   1.5   0.7  11.3  26.1  3.3
1.9
188  0.496  0.429   2.3  17.1   9.8  43.3   1.6   1.4  16.3  25.0  5.8
2.6
202  0.351  0.137   1.4  10.2   5.7  32.3   2.3   1.1  16.7  20.5 -1.9
1.4
212  0.402  0.173   3.9  12.0   8.1  34.4   1.7   0.9  15.2  25.0  3.9
2.8
424  0.517  0.228   1.7   8.5   5.2  27.4   1.4   1.1  15.1  22.7  3.1
1.9
68   0.269  0.328   1.8   9.3   5.6  28.7   1.3   0.6   9.4  27.2  6.9
1.8
123  0.373  0.479   2.6  25.4  13.8  42.3   1.9   1.2  11.9  37.6  7.3
2.9
162  0.122  0.535   2.6  12.2   7.3  25.7   2.2   2.5  10.1  32.8  8.4
```

```
3.0
151  0.276  0.331   1.9  12.4   7.2  29.6   1.6   0.8  10.6  30.1  5.5
1.8
155  0.365  0.284   1.3   7.6   4.5  34.1   1.7   0.3  13.5  26.9  4.5
3.0
386  0.498  0.325   1.5   8.5   5.0  22.5   1.2   0.8  14.7  29.2  1.4
1.9
483  0.545  0.254   1.2  12.0   6.3  28.1   2.4   1.5  10.2  23.2  3.7
2.8
534  0.331  0.460   2.4   7.0   4.7  42.5   1.5   0.3  15.2  32.6  5.3
1.4
121  0.467  0.314   1.2  10.2   5.7  28.1   1.2   0.8  10.8  22.1  4.5
1.8
```

	WS	WS/48	OBPM	DBPM	BPM	VORP	Team	Standing
152	3.7	0.100	-0.1	0.7	0.5	1.2		Bottom
38	3.0	0.079	-2.9	1.7	-1.2	0.3		Bottom
184	7.6	0.195	7.0	0.2	7.2	4.4		Bottom
253	3.4	0.082	-0.2	-0.8	-1.0	0.5		Bottom
387	2.9	0.068	1.4	-0.8	0.6	1.3		Bottom
97	5.5	0.130	1.0	-0.1	0.9	1.5		Top
290	9.0	0.205	8.3	-1.2	7.1	4.9		Bottom
505	1.9	0.044	0.3	-0.1	0.2	1.2		Top
350	5.1	0.116	2.6	-1.3	1.3	1.8		Top
188	8.4	0.188	5.1	0.3	5.4	4.0		Top
202	-0.5	-0.012	-3.3	-0.5	-3.8	-1.0		Bottom
212	6.7	0.148	3.0	0.1	3.1	2.8		Top
424	5.1	0.106	2.1	-0.7	1.5	2.0		Top
68	8.7	0.175	4.4	-0.5	3.9	3.5		Top
123	10.2	0.204	7.6	1.4	8.9	6.6		Bottom
162	11.4	0.226	5.7	1.5	7.3	5.6		Bottom
151	7.4	0.146	3.4	-0.9	2.5	2.7		Top
155	7.6	0.148	3.2	-0.7	2.4	2.7		Top
386	3.2	0.063	-0.1	-1.8	-1.9	0.1		Top
483	6.5	0.123	2.0	0.5	2.5	2.9		Bottom
534	6.7	0.126	5.3	-2.0	3.3	3.4		Bottom
121	6.3	0.111	1.6	-0.9	0.7	1.8		Top

```
df_point_guards_final.info()

<class 'pandas.core.frame.DataFrame'>
Index: 22 entries, 38 to 534
Data columns (total 52 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Rk                 22 non-null     int64
 1   Player             22 non-null     object
 2   Pos                22 non-null     object
 3   Age                22 non-null     int64
 4   Tm                 22 non-null     object
```

```
 5   G                   22 non-null     int64
 6   GS                  22 non-null     int64
 7   MP                  22 non-null     int64
 8   FG                  22 non-null     int64
 9   FGA                 22 non-null     int64
 10  FG%                 22 non-null     float64
 11  3P                  22 non-null     int64
 12  3PA                 22 non-null     int64
 13  3P%                 22 non-null     float64
 14  2P                  22 non-null     int64
 15  2PA                 22 non-null     int64
 16  2P%                 22 non-null     float64
 17  eFG%                22 non-null     float64
 18  FT                  22 non-null     int64
 19  FTA                 22 non-null     int64
 20  FT%                 22 non-null     float64
 21  ORB                 22 non-null     int64
 22  DRB                 22 non-null     int64
 23  TRB                 22 non-null     int64
 24  AST                 22 non-null     int64
 25  STL                 22 non-null     int64
 26  BLK                 22 non-null     int64
 27  TOV                 22 non-null     int64
 28  PF                  22 non-null     int64
 29  PTS                 22 non-null     int64
 30  Player-additional   22 non-null     object
 31  PER                 22 non-null     float64
 32  TS%                 22 non-null     float64
 33  3PAr                22 non-null     float64
 34  FTr                 22 non-null     float64
 35  ORB%                22 non-null     float64
 36  DRB%                22 non-null     float64
 37  TRB%                22 non-null     float64
 38  AST%                22 non-null     float64
 39  STL%                22 non-null     float64
 40  BLK%                22 non-null     float64
 41  TOV%                22 non-null     float64
 42  USG%                22 non-null     float64
 43  OWS                 22 non-null     float64
 44  DWS                 22 non-null     float64
 45  WS                  22 non-null     float64
 46  WS/48               22 non-null     float64
 47  OBPM                22 non-null     float64
 48  DBPM                22 non-null     float64
 49  BPM                 22 non-null     float64
 50  VORP                22 non-null     float64
 51  Team Standing       22 non-null     object
dtypes: float64(25), int64(22), object(5)
memory usage: 9.1+ KB
```

```
## Advanced stats
# included=['PER', 'TS%', '3PAr', 'FTr', 'ORB%', 'DRB%', 'TRB%'
#           ,'AST%', 'STL%', 'BLK%', 'TOV%', 'USG%', 'OWS', 'DWS'
#           , 'WS', 'WS/48', 'OBPM', 'DBPM', 'BPM', 'VORP'
#          ]


## Regular stats
included=['FG', 'FGA', 'FG%', '3P', '3PA', '3P%', '2P', '2PA', '2P%',
       'eFG%', 'FT', 'FTA', 'FT%', 'ORB', 'DRB', 'TRB', 'AST', 'STL',
'BLK',
       'TOV', 'PF', 'PTS', 'TOV%']

result=[]
import scipy.stats as stats
for i in included:
    bfg=df_point_guards_final[df_point_guards_final['Team
Standing']=='Bottom'][i]
    tfg=df_point_guards_final[df_point_guards_final['Team
Standing']=='Top'][i]
    result.append(stats.ttest_ind(a=bfg, b=tfg,equal_var=True).pvalue)

alpha=0.05
decision=[]
for i in result:
    if i < alpha:
        decision.append('Reject')
    else:
        decision.append('Accept')

final=pd.DataFrame(data={'Metrics':included,'P-
Value':result,'Decision':decision})

final['Decision'].value_counts()
```

```
Decision
Accept    21
Reject     2
Name: count, dtype: int64
```

```
final[final['Decision'] == 'Reject']
```

```
   Metrics   P-Value Decision
5      3P%  0.025057   Reject
17     STL  0.025173   Reject
```

```
final
```

```
   Metrics   P-Value Decision
0       FG  0.652280   Accept
1      FGA  0.666614   Accept
```

```
2       FG%  0.719034   Accept
3        3P  0.209829   Accept
4       3PA  0.388794   Accept
5       3P%  0.025057   Reject
6        2P  0.966528   Accept
7       2PA  0.970489   Accept
8       2P%  0.850509   Accept
9      eFG%  0.240229   Accept
10       FT  0.489341   Accept
11      FTA  0.570402   Accept
12      FT%  0.614600   Accept
13      ORB  0.511704   Accept
14      DRB  0.834511   Accept
15      TRB  0.745401   Accept
16      AST  0.578361   Accept
17      STL  0.025173   Reject
18      BLK  0.258704   Accept
19      TOV  0.625229   Accept
20       PF  0.915260   Accept
21      PTS  0.848141   Accept
22     TOV%  0.830425   Accept
```

Weighting scores

```
# Assuming sg_key_stats is your DataFrame with players and their stats
#keyStats = ['Player_adv', 'PTS', 'TS%', '3P%', '3PA', '3P', '3PAr',
'AST', 'STL', 'DWS', 'PER', 'VORP', 'WS', 'USG%']
weights = {'3P%': 0.5, 'STL':0.5}

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

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

# Rank players
ranked_pg = df_point_guards_final.sort_values('Weighted_Score',
ascending=False)

# Display the top ranked players
# print(ranked_pg[['Player_adv', 'Weighted_Score']].head(25))

ranked_pg[['Player', 'Tm', 'Weighted_Score']]
```

```
                    Player   Tm  Weighted_Score
483           Fred VanVleet  TOR        0.911058
```

```
162  Shai Gilgeous-Alexander  OKC        0.869948
184        Tyrese Haliburton  IND        0.850688
155           Darius Garland  CLE        0.838317
212             Jrue Holiday  MIL        0.782677
123              Luka Dončić  DAL        0.776911
387         Kevin Porter Jr.  HOU        0.773237
424         D'Angelo Russell  LAL        0.760514
97               Mike Conley  MIN        0.759488
202            Killian Hayes  DET        0.759303
188             James Harden  PHI        0.751358
68             Jalen Brunson  NYK        0.747967
350             Jamal Murray  DEN        0.746658
534               Trae Young  ATL        0.727847
151             De'Aaron Fox  SAC        0.726821
152            Markelle Fultz  ORL        0.726255
121        Spencer Dinwiddie  BRK        0.715867
253                Tre Jones  SAS        0.704337
505        Russell Westbrook  LAC        0.682741
386             Jordan Poole  GSW        0.659944
38           Patrick Beverley  CHI        0.658742
290             Damian Lillard  POR        0.649165
```

```
ranked_pg.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 22 entries, 483 to 290
Data columns (total 55 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Rk                  22 non-null     int64  
 1   Player              22 non-null     object 
 2   Pos                 22 non-null     object 
 3   Age                 22 non-null     int64  
 4   Tm                  22 non-null     object 
 5   G                   22 non-null     int64  
 6   GS                  22 non-null     int64  
 7   MP                  22 non-null     int64  
 8   FG                  22 non-null     int64  
 9   FGA                 22 non-null     int64  
 10  FG%                 22 non-null     float64
 11  3P                  22 non-null     int64  
 12  3PA                 22 non-null     int64  
 13  3P%                 22 non-null     float64
 14  2P                  22 non-null     int64  
 15  2PA                 22 non-null     int64  
 16  2P%                 22 non-null     float64
 17  eFG%                22 non-null     float64
 18  FT                  22 non-null     int64  
 19  FTA                 22 non-null     int64  
 20  FT%                 22 non-null     float64
```

```
 21  ORB                 22 non-null     int64
 22  DRB                 22 non-null     int64
 23  TRB                 22 non-null     int64
 24  AST                 22 non-null     int64
 25  STL                 22 non-null     int64
 26  BLK                 22 non-null     int64
 27  TOV                 22 non-null     int64
 28  PF                  22 non-null     int64
 29  PTS                 22 non-null     int64
 30  Player-additional   22 non-null     object
 31  PER                 22 non-null     float64
 32  TS%                 22 non-null     float64
 33  3PAr                22 non-null     float64
 34  FTr                 22 non-null     float64
 35  ORB%                22 non-null     float64
 36  DRB%                22 non-null     float64
 37  TRB%                22 non-null     float64
 38  AST%                22 non-null     float64
 39  STL%                22 non-null     float64
 40  BLK%                22 non-null     float64
 41  TOV%                22 non-null     float64
 42  USG%                22 non-null     float64
 43  OWS                 22 non-null     float64
 44  DWS                 22 non-null     float64
 45  WS                  22 non-null     float64
 46  WS/48               22 non-null     float64
 47  OBPM                22 non-null     float64
 48  DBPM                22 non-null     float64
 49  BPM                 22 non-null     float64
 50  VORP                22 non-null     float64
 51  Team Standing       22 non-null     object
 52  3P%_norm            22 non-null     float64
 53  STL_norm            22 non-null     float64
 54  Weighted_Score      22 non-null     float64
dtypes: float64(28), int64(22), object(5)
memory usage: 9.6+ KB
```

Merging dataset with salary

```
df_salary = pd.read_excel('salary_players.xlsx')
df_salary.head()
```

```
               PLAYER    salary  salary_currentDollar
0      Stephen Curry  48070014              49497298
1          John Wall  47345760              48751539
2  Russell Westbrook  47080179              48478072
3       LeBron James  44474988              45795529
4       Kevin Durant  44119845              45429841
```

```
ranked_pg_final = pd.merge(ranked_pg, df_salary,  how='left',
left_on=['Player'], right_on = ['PLAYER'])
ranked_pg_final
```

```
     Rk                   Player Pos  Age   Tm   G  GS    MP   FG
FGA  \
0   485            Fred VanVleet  PG   28  TOR  69  69  2535  437
1112
1   164  Shai Gilgeous-Alexander  PG   24  OKC  68  68  2416  704
1381
2   186        Tyrese Haliburton  PG   22  IND  56  56  1883  412
841
3   157           Darius Garland  PG   23  CLE  69  69  2447  522
1129
4   214             Jrue Holiday  PG   32  MIL  67  65  2183  490
1023
5   125              Luka Dončić  PG   23  DAL  66  66  2391  719
1449
6   389         Kevin Porter Jr.  PG   22  HOU  59  59  2024  391
884
7   426         D'Angelo Russell  PG   26  LAL  71  71  2304  445
948
8    99              Mike Conley  PG   35  MIN  67  66  2029  261
610
9   204            Killian Hayes  PG   21  DET  76  56  2154  307
815
10  190             James Harden  PG   33  PHI  58  58  2135  371
842
11   69            Jalen Brunson  PG   26  NYK  68  68  2379  587
1195
12  352             Jamal Murray  PG   25  DEN  65  65  2133  473
1041
13  536               Trae Young  PG   24  ATL  73  73  2541  597
1390
14  153             De'Aaron Fox  PG   25  SAC  73  73  2435  682
1331
15  154           Markelle Fultz  PG   24  ORL  60  60  1778  349
679
16  123        Spencer Dinwiddie  PG   29  BRK  79  79  2725  460
1050
17  255                Tre Jones  PG   23  SAS  68  65  1984  341
743
18  507        Russell Westbrook  PG   34  LAC  73  24  2126  432
991
19  388             Jordan Poole  PG   23  GSW  82  43  2458  550
1278
20   39         Patrick Beverley  PG   34  CHI  67  67  1816  144
360
21  292           Damian Lillard  PG   32  POR  58  58  2107  556
1202
```

```
      FG%   3P  3PA    3P%   2P   2PA    2P%   eFG%   FT  FTA    FT%
ORB  \
0   0.393  207  606  0.342  230   506  0.455  0.486  254  283  0.898
30
1   0.510   58  168  0.345  646  1213  0.533  0.531  669  739  0.905
59
2   0.490  161  402  0.400  251   439  0.572  0.586  175  201  0.871
33
3   0.462  169  412  0.410  353   717  0.492  0.537  277  321  0.863
28
4   0.479  158  411  0.384  332   612  0.542  0.556  152  177  0.859
79
5   0.496  185  541  0.342  534   908  0.588  0.560  515  694  0.742
54
6   0.442  141  385  0.366  250   499  0.501  0.522  207  264  0.784
76
7   0.469  194  490  0.396  251   458  0.548  0.572  179  216  0.829
35
8   0.428  135  351  0.385  126   259  0.486  0.539  141  169  0.834
33
9   0.377   80  286  0.280  227   529  0.429  0.426   92  112  0.821
28
10  0.441  161  418  0.385  210   424  0.495  0.536  313  361  0.867
40
11  0.491  134  322  0.416  453   873  0.519  0.547  325  392  0.829
40
12  0.454  172  432  0.398  301   609  0.494  0.537  180  216  0.833
48
13  0.429  154  460  0.335  443   930  0.476  0.485  566  639  0.886
56
14  0.512  119  367  0.324  563   964  0.584  0.557  343  440  0.780
40
15  0.514   27   87  0.310  322   592  0.544  0.534  112  143  0.783
69
16  0.438  181  490  0.369  279   560  0.498  0.524  268  330  0.812
28
17  0.459   45  158  0.285  296   585  0.506  0.489  148  172  0.860
56
18  0.436   89  286  0.311  343   705  0.487  0.481  206  314  0.656
89
19  0.430  214  637  0.336  336   641  0.524  0.514  361  415  0.870
32
20  0.400   80  239  0.335   64   121  0.529  0.511   47   65  0.723
44
21  0.463  244  658  0.371  312   544  0.574  0.564  510  558  0.914
44

    DRB  TRB  AST  STL  BLK  TOV   PF   PTS Player-additional   PER
```

```
TS%  \
0   250  280  495  123   38  140  193  1335         vanvlfr01  17.0
0.540
1   270  329  371  112   65  192  192  2135         gilgesh01  27.2
0.626
2   172  205  585   91   25  141   69  1160         halibty01  23.6
0.624
3   157  185  538   85    9  199  148  1490         garlada01  18.8
0.587
4   262  341  495   79   25  197  116  1290         holidjr01  19.2
0.586
5   515  569  529   90   33  236  166  2138         doncilu01  28.7
0.609
6   238  314  338   82   17  188  156  1130         porteke02  16.2
0.565
7   180  215  437   70   29  186  140  1263         russeda01  16.3
0.605
8   149  182  450   73   14  103  139   798         conlemi01  14.7
0.583
9   193  221  470  104   28  173  219   786         hayeski01  10.2
0.455
10  314  354  618   71   31  195  112  1216         hardeja01  21.6
0.607
11  201  241  421   61   15  142  152  1633         brunsja01  21.2
0.597
12  209  257  400   66   16  145  103  1298         murraja01  18.0
0.571
13  161  217  741   80    9  300  104  1914         youngtr01  22.0
0.573
14  266  306  447   83   23  181  172  1826           foxde01  21.8
0.599
15  165  234  341   87   26  139  131   837         fultzma01  16.6
0.564
16  242  270  515   67   24  145  187  1369         dinwisp01  16.0
0.573
17  189  245  448   89    9  110   98   875         jonestr01  16.0
0.534
18  334  423  551   76   33  255  162  1159         westbru01  16.1
0.513
19  193  225  369   63   21  252  214  1675         poolejo01  14.6
0.573
20  203  247  194   63   41   60  187   415         beverpa01   8.9
0.534
21  233  277  425   50   18  191  109  1866         lillada01  26.7
0.645

     3PAr    FTr  ORB%  DRB%  TRB%  AST%  STL%  BLK%  TOV%  USG%  OWS
DWS  \
0   0.545  0.254   1.2  12.0   6.3  28.1   2.4   1.5  10.2  23.2  3.7
```

```
2.8
1   0.122  0.535   2.6  12.2   7.3  25.7   2.2   2.5  10.1  32.8  8.4
3.0
2   0.478  0.239   1.9  10.1   6.0  47.6   2.3   1.1  13.2  23.8  6.4
1.2
3   0.365  0.284   1.3   7.6   4.5  34.1   1.7   0.3  13.5  26.9  4.5
3.0
4   0.402  0.173   3.9  12.0   8.1  34.4   1.7   0.9  15.2  25.0  3.9
2.8
5   0.373  0.479   2.6  25.4  13.8  42.3   1.9   1.2  11.9  37.6  7.3
2.9
6   0.436  0.299   4.1  13.1   8.5  25.7   2.0   0.8  15.8  24.3  1.9
1.0
7   0.517  0.228   1.7   8.5   5.2  27.4   1.4   1.1  15.1  22.7  3.1
1.9
8   0.575  0.277   1.8   7.9   4.9  29.4   1.7   0.6  13.1  16.3  4.1
1.4
9   0.351  0.137   1.4  10.2   5.7  32.3   2.3   1.1  16.7  20.5 -1.9
1.4
10  0.496  0.429   2.3  17.1   9.8  43.3   1.6   1.4  16.3  25.0  5.8
2.6
11  0.269  0.328   1.8   9.3   5.6  28.7   1.3   0.6   9.4  27.2  6.9
1.8
12  0.415  0.207   2.7  11.0   6.9  27.5   1.5   0.7  11.3  26.1  3.3
1.9
13  0.331  0.460   2.4   7.0   4.7  42.5   1.5   0.3  15.2  32.6  5.3
1.4
14  0.276  0.331   1.9  12.4   7.2  29.6   1.6   0.8  10.6  30.1  5.5
1.8
15  0.128  0.211   4.4  10.5   7.4  29.8   2.4   1.4  15.8  21.3  1.7
2.0
16  0.467  0.314   1.2  10.2   5.7  28.1   1.2   0.8  10.8  22.1  4.5
1.8
17  0.213  0.231   3.0  10.8   6.7  31.5   2.1   0.4  11.8  19.3  2.6
0.8
18  0.289  0.317   4.7  16.5  10.8  38.6   1.7   1.3  18.4  27.7 -0.6
2.6
19  0.498  0.325   1.5   8.5   5.0  22.5   1.2   0.8  14.7  29.2  1.4
1.9
20  0.664  0.181   2.7  12.0   7.5  13.3   1.7   2.0  13.4  10.6  0.8
2.1
21  0.547  0.464   2.4  12.8   7.6  35.0   1.2   0.8  11.7  33.8  8.2
0.8

      WS  WS/48  OBPM  DBPM  BPM  VORP Team Standing  3P%_norm
STL_norm  \
0    6.5  0.123   2.0   0.5  2.5   2.9        Bottom  0.822115
1.000000
1   11.4  0.226   5.7   1.5  7.3   5.6        Bottom  0.829327
```

```
0.910569
2    7.6  0.195   7.0   0.2  7.2   4.4        Bottom  0.961538
0.739837
3    7.6  0.148   3.2  -0.7  2.4   2.7           Top  0.985577
0.691057
4    6.7  0.148   3.0   0.1  3.1   2.8           Top  0.923077
0.642276
5   10.2  0.204   7.6   1.4  8.9   6.6        Bottom  0.822115
0.731707
6    2.9  0.068   1.4  -0.8  0.6   1.3        Bottom  0.879808
0.666667
7    5.1  0.106   2.1  -0.7  1.5   2.0           Top  0.951923
0.569106
8    5.5  0.130   1.0  -0.1  0.9   1.5           Top  0.925481
0.593496
9   -0.5 -0.012  -3.3  -0.5 -3.8  -1.0        Bottom  0.673077
0.845528
10   8.4  0.188   5.1   0.3  5.4   4.0           Top  0.925481
0.577236
11   8.7  0.175   4.4  -0.5  3.9   3.5           Top  1.000000
0.495935
12   5.1  0.116   2.6  -1.3  1.3   1.8           Top  0.956731
0.536585
13   6.7  0.126   5.3  -2.0  3.3   3.4        Bottom  0.805288
0.650407
14   7.4  0.146   3.4  -0.9  2.5   2.7           Top  0.778846
0.674797
15   3.7  0.100  -0.1   0.7  0.5   1.2        Bottom  0.745192
0.707317
16   6.3  0.111   1.6  -0.9  0.7   1.8           Top  0.887019
0.544715
17   3.4  0.082  -0.2  -0.8 -1.0   0.5        Bottom  0.685096
0.723577
18   1.9  0.044   0.3  -0.1  0.2   1.2           Top  0.747596
0.617886
19   3.2  0.063  -0.1  -1.8 -1.9   0.1           Top  0.807692
0.512195
20   3.0  0.079  -2.9   1.7 -1.2   0.3        Bottom  0.805288
0.512195
21   9.0  0.205   8.3  -1.2  7.1   4.9        Bottom  0.891827
0.406504

    Weighted_Score                   PLAYER    salary
salary_currentDollar
0         0.911058             Fred VanVleet  21250000
21880950
1         0.869948  Shai Gilgeous-Alexander  30913750
31831634
2         0.850688         Tyrese Haliburton   4215120
```

```
4340274
3         0.838317           Darius Garland   8920794
9185668
4         0.782677             Jrue Holiday  34319520
35338527
5         0.776911              Luka Dončić  37096500
38197960
6         0.773237         Kevin Porter Jr.   3217631
3313168
7         0.760514         D'Angelo Russell  31377750
32309411
8         0.759488              Mike Conley  22680000
23353409
9         0.759303            Killian Hayes   5837760
6011093
10        0.751358             James Harden  33000000
33979828
11        0.747967            Jalen Brunson  27733332
28556783
12        0.746658             Jamal Murray  31650600
32590362
13        0.727847               Trae Young  37096500
38197960
14        0.726821             De'Aaron Fox  30351780
31252978
15        0.726255           Markelle Fultz  16500000
16989914
16        0.715867        Spencer Dinwiddie  19500000
20078989
17        0.704337                Tre Jones   1782621
1835550
18        0.682741        Russell Westbrook  47080179
48478072
19        0.659944             Jordan Poole   3901399
4017238
20        0.658742          Patrick Beverley  13801614
14211408
21        0.649165           Damian Lillard  42492492
43754169

ranked_pg_final[['Player', 'Tm', 'Pos', 'Weighted_Score',
'salary_currentDollar']].to_clipboard(index = False)

ranked_pg_final.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22 entries, 0 to 21
Data columns (total 58 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
```

```
0   Rk                      22 non-null     int64
1   Player                  22 non-null     object
2   Pos                     22 non-null     object
3   Age                     22 non-null     int64
4   Tm                      22 non-null     object
5   G                       22 non-null     int64
6   GS                      22 non-null     int64
7   MP                      22 non-null     int64
8   FG                      22 non-null     int64
9   FGA                     22 non-null     int64
10  FG%                     22 non-null     float64
11  3P                      22 non-null     int64
12  3PA                     22 non-null     int64
13  3P%                     22 non-null     float64
14  2P                      22 non-null     int64
15  2PA                     22 non-null     int64
16  2P%                     22 non-null     float64
17  eFG%                    22 non-null     float64
18  FT                      22 non-null     int64
19  FTA                     22 non-null     int64
20  FT%                     22 non-null     float64
21  ORB                     22 non-null     int64
22  DRB                     22 non-null     int64
23  TRB                     22 non-null     int64
24  AST                     22 non-null     int64
25  STL                     22 non-null     int64
26  BLK                     22 non-null     int64
27  TOV                     22 non-null     int64
28  PF                      22 non-null     int64
29  PTS                     22 non-null     int64
30  Player-additional       22 non-null     object
31  PER                     22 non-null     float64
32  TS%                     22 non-null     float64
33  3PAr                    22 non-null     float64
34  FTr                     22 non-null     float64
35  ORB%                    22 non-null     float64
36  DRB%                    22 non-null     float64
37  TRB%                    22 non-null     float64
38  AST%                    22 non-null     float64
39  STL%                    22 non-null     float64
40  BLK%                    22 non-null     float64
41  TOV%                    22 non-null     float64
42  USG%                    22 non-null     float64
43  OWS                     22 non-null     float64
44  DWS                     22 non-null     float64
45  WS                      22 non-null     float64
46  WS/48                   22 non-null     float64
47  OBPM                    22 non-null     float64
48  DBPM                    22 non-null     float64
```

```
 49  BPM                    22 non-null     float64
 50  VORP                   22 non-null     float64
 51  Team Standing          22 non-null     object
 52  3P%_norm               22 non-null     float64
 53  STL_norm               22 non-null     float64
 54  Weighted_Score         22 non-null     float64
 55  PLAYER                 22 non-null     object
 56  salary                 22 non-null     int64
 57  salary_currentDollar   22 non-null     int64
dtypes: float64(28), int64(24), object(6)
memory usage: 10.1+ KB
```

Player Hiring Optimization

PG

```
pg_final = ranked_pg_final[['Player', 'Tm', 'Pos', 'Weighted_Score',
'salary_currentDollar']].rename(columns={'salary_currentDollar':
'Salary'}).set_index('Player')
pg_final.head()
```

Player	Tm	Pos	Weighted_Score	Salary
Fred VanVleet	TOR	PG	0.911058	21880950
Shai Gilgeous-Alexander	OKC	PG	0.869948	31831634
Tyrese Haliburton	IND	PG	0.850688	4340274
Darius Garland	CLE	PG	0.838317	9185668
Jrue Holiday	MIL	PG	0.782677	35338527

Center

```
center_final = pd.read_csv('centers_final_list.csv', index_col=0,
usecols = ['Player', 'Tm','Pos', 'Weighted_Score', 'Salary'])
center_final.head()
```

Player	Tm	Pos	Weighted_Score	Salary
Nikola Jokić	DEN	C	0.972432	33047803
Joel Embiid	PHI	C	0.736519	33616770
Domantas Sabonis	SAC	C	0.571141	21100000
Anthony Davis	LAL	C	0.526742	37980720
Nic Claxton	BRK	C	0.420568	9350000

PF

```
pf_final = pd.read_csv('PF_final_list.csv', index_col=0, usecols =
['Player', 'Tm','Pos', 'Weighted_Score',
'salary_currentDollar']).rename(columns={'salary_currentDollar':
'Salary'})
```

SF

```
sf_final = pd.read_csv('sfs_final_list.csv',
index_col=0).rename(columns={'Team':'Tm', 'Position':'Pos'})
sf_final.head()
sf_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 22 entries, Cam Reddish to Deni Avdija
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Tm              22 non-null     object
 1   Pos             22 non-null     object
 2   Weighted_Score  22 non-null     float64
 3   Salary          22 non-null     int64
dtypes: float64(1), int64(1), object(2)
memory usage: 880.0+ bytes
```

SG

```
sg_final = pd.read_csv('sg_final_listv1.csv', index_col=0,
usecols=['Player', 'Tm', 'Pos', 'Weighted_Score', 'Salary'])
for i in sg_final.index:
    sg_final.loc[i, 'Salary'] = sg_final.loc[i, 'Salary'][2:-
4].replace(',', '')
sg_final['Salary'] = sg_final['Salary'].astype('int64')
sg_final.head()
```

Player	Tm	Pos	Weighted_Score	Salary
Donovan Mitchell	CLE	SG	0.917656	31831634
Derrick White	BOS	SG	0.798223	17651858
Zach LaVine	CHI	SG	0.759535	38197960
Kyrie Irving	DAL	SG	0.752825	40072573
Immanuel Quickley	NYK	SG	0.733209	2385013

All Combined

```
combined = [pg_final, center_final, pf_final, sf_final, sg_final]

optimization_player_final = pd.concat(combined)
optimization_player_final.sort_values(by = 'Salary').head()
```

	Tm	Pos	Weighted_Score	Salary
Justin Minaya	NOP	SF	0.389453	35096
Xavier Sneed	POR	SF	0.306511	102910
Tyler Dorsey	LAL	SF	0.364787	201802
Jack White	BOS	SF	0.401645	508891
Eugene Omoruyi	SAC	SF	0.450185	1013119

```
optimization_player_final =
optimization_player_final.join(df[['Player',
'Age']].set_index('Player'))
optimization_player_final.reset_index(inplace=True)
optimization_player_final.drop_duplicates(subset=['index'],
keep='first', inplace= True)
optimization_player_final.set_index(['index'], inplace=True)
optimization_player_final.head()
```

index	Tm	Pos	Weighted_Score	Salary	Age
Aaron Gordon	DEN	PF	0.487153	21305258	27
Al Horford	BOS	C	0.260273	26500000	36
Aleksej Pokusevski	OKC	PF	0.276719	3358319	21
Alperen Şengün	HOU	C	0.262866	3375360	20
Amir Coffey	TOR	SF	0.746137	3395062	25

```
optimization_player_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 163 entries, Aaron Gordon to Zeke Nnaji
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Tm              163 non-null    object
 1   Pos             163 non-null    object
 2   Weighted_Score  163 non-null    float64
 3   Salary          163 non-null    int64
 4   Age             163 non-null    int64
dtypes: float64(1), int64(2), object(2)
memory usage: 7.6+ KB
```

```
optimization_player_final.sort_values(by = 'Salary').head(10)
```

index	Tm	Pos	Weighted_Score	Salary	Age
Justin Minaya	NOP	SF	0.389453	35096	23
Xavier Sneed	POR	SF	0.306511	102910	25
Tyler Dorsey	LAL	SF	0.364787	201802	26
Jack White	BOS	SF	0.401645	508891	25
Eugene Omoruyi	SAC	SF	0.450185	1013119	25
Jabari Walker	ORL	SF	0.532698	1017781	20
Josh Minott	MIA	SF	0.448499	1017781	20
Isaiah Livers	DET	PF	0.182456	1609941	24
JT Thor	CHO	PF	0.090607	1609941	20
Trendon Watford	POR	PF	0.249674	1609941	22

```
optimization_player_final['Age_Category'] = ['>30yo' if x > 30 else
'<=30yo' for x in optimization_player_final['Age']]
optimization_player_final.head()
```

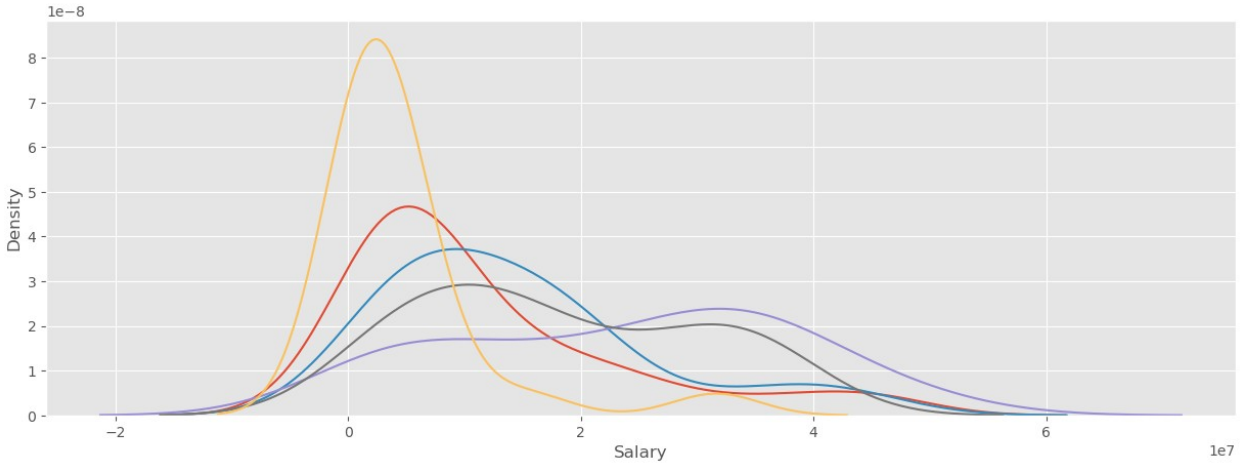
```
                      Tm Pos  Weighted_Score    Salary  Age
Age_Category
index

Aaron Gordon         DEN  PF        0.487153  21305258   27
<=30yo
Al Horford           BOS   C        0.260273  26500000   36
>30yo
Aleksej Pokusevski   OKC  PF        0.276719   3358319   21
<=30yo
Alperen Şengün       HOU   C        0.262866   3375360   20
<=30yo
Amir Coffey          TOR  SF        0.746137   3395062   25
<=30yo
```

```
optimization_player_final['Age_Category'].value_counts(normalize=True)
```

```
Age_Category
<=30yo    0.834356
>30yo     0.165644
Name: proportion, dtype: float64
```

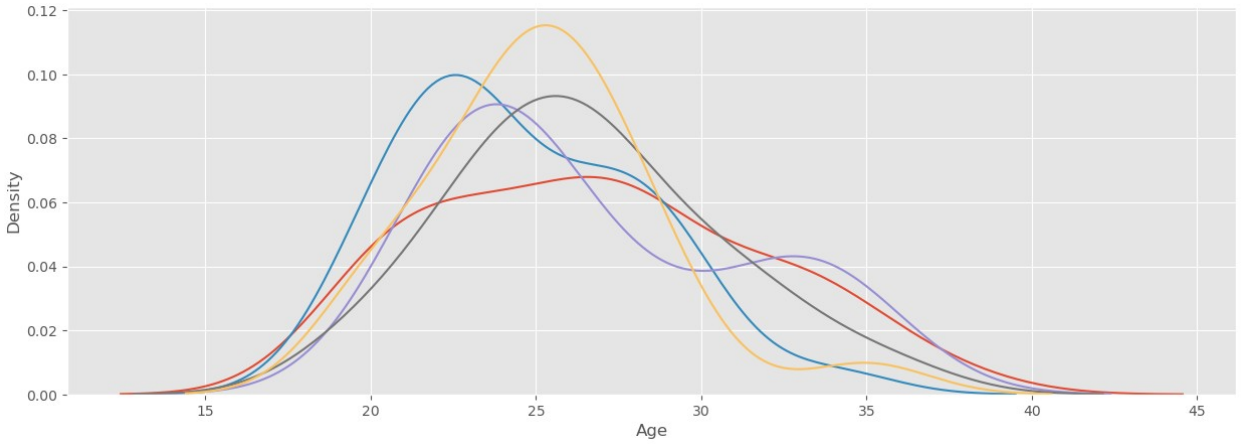
```
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'PF']
['Salary'])
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'SG']
['Salary'])
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'PG']
['Salary'])
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'C']
['Salary'])
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'SF']
['Salary'])
```

```
<Axes: xlabel='Salary', ylabel='Density'>
```



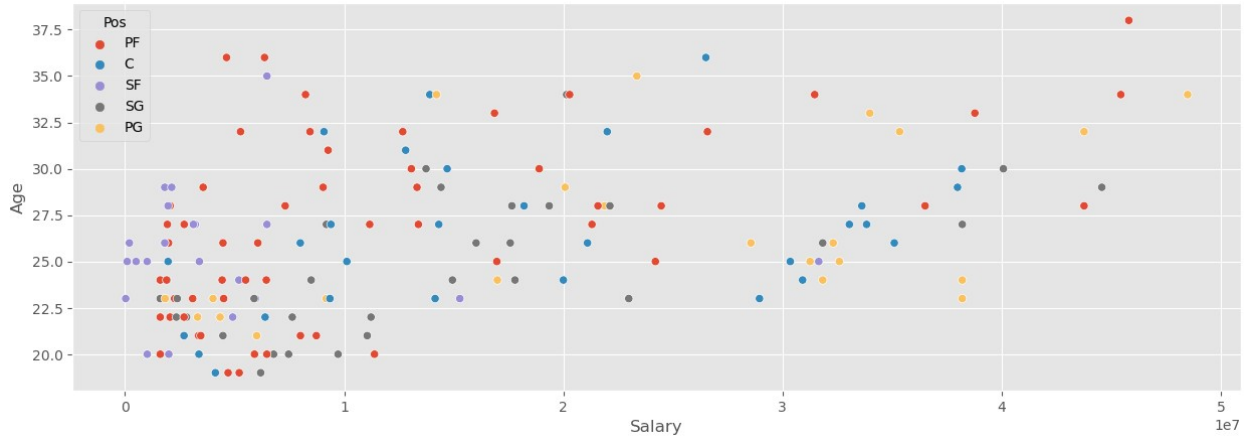

```
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'PF']
['Age'])
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'SG']
['Age'])
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'PG']
['Age'])
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'C']
['Age'])
sns.kdeplot(x =
optimization_player_final[optimization_player_final['Pos'] == 'SF']
['Age'])
```

```
<Axes: xlabel='Age', ylabel='Density'>
```




```
sns.scatterplot(data=optimization_player_final, x="Salary", y="Age",
hue="Pos")
```

```
<Axes: xlabel='Salary', ylabel='Age'>
```




```
optimization_player_final[optimization_player_final.Pos ==
'PF'].sort_values(by='Salary', ascending=False)
```

	Tm	Pos	Weighted_Score	Salary	Age	Age_Category
index						
LeBron James	LAL	PF	0.566235	45795529	38	>30yo
Kevin Durant	PHO	PF	0.672162	45429841	34	>30yo
Giannis Antetokounmpo	MIL	PF	0.821356	43754169	28	<=30yo
Jimmy Butler	MIA	PF	0.904961	38771293	33	>30yo
Pascal Siakam	TOR	PF	0.528294	36501206	28	<=30yo
Kevin Love	MIA	PF	0.330760	31464258	34	>30yo
Draymond Green	GSW	PF	0.513003	26572707	32	>30yo
Julius Randle	NYK	PF	0.559894	24465476	28	<=30yo
John Collins	ATL	PF	0.339439	24197756	25	<=30yo
Jerami Grant	POR	PF	0.282994	21577191	28	<=30yo
Aaron Gordon	DEN	PF	0.487153	21305258	27	<=30yo
Nicolas Batum	LAC	PF	0.407256	20285256	34	>30yo
Harrison Barnes	SAC	PF	0.334503	18897184	30	<=30yo

Lauri Markkanen	UTA	PF	0.535737	16964639	25	<=30yo
Marcus Morris	LAC	PF	0.222221	16858209	33	>30yo
Kyle Kuzma	WAS	PF	0.144403	13385993	27	<=30yo
Dorian Finney-Smith	BRK	PF	0.216150	13324055	29	<=30yo
Chris Boucher	TOR	PF	0.358045	13066788	30	<=30yo
Robert Covington	LAC	PF	0.411902	12673129	32	>30yo
Paolo Banchero	ORL	PF	0.219661	11383366	20	<=30yo
Bobby Portis	MIL	PF	0.392633	11165308	27	<=30yo
Maxi Kleber	DAL	PF	0.225553	9267225	31	>30yo
Kyle Anderson	MIN	PF	0.543034	9041196	29	<=30yo
Evan Mobley	CLE	PF	0.619509	8730468	21	<=30yo
JaMychal Green	GSW	PF	0.329246	8443471	32	>30yo
Thaddeus Young	TOR	PF	0.368933	8237534	34	>30yo
Patrick Williams	CHI	PF	0.329920	8006265	21	<=30yo
Taurean Prince	MIN	PF	0.177466	7310811	28	<=30yo
Josh Giddey	OKC	PF	0.359838	6474084	20	<=30yo
Rui Hachimura	LAL	PF	0.186427	6449153	24	<=30yo
Rudy Gay	UTA	PF	0.097376	6368128	36	>30yo
Cameron Johnson	BRK	PF	0.428816	6062721	26	<=30yo
Jonathan Kuminga	GSW	PF	0.270725	5910266	20	<=30yo
Obi Toppin	NYK	PF	0.246169	5507079	24	<=30yo
Torrey Craig	PHO	PF	0.371698	5274030	32	>30yo
Jeremy Sochan	SAS	PF	0.064714	5213864	19	<=30yo
Ousmane Dieng	OKC	PF	0.126533	4705526	19	<=30yo
Jeff Green	DEN	PF	0.164791	4633612	36	

```
>30yo
Jarred Vanderbilt        LAL  PF         0.494678   4503871   23
<=30yo
Brandon Clarke           MEM  PF         0.547184   4472897   26
<=30yo
Grant Williams           BOS  PF         0.384106   4434142   24
<=30yo
Georges Niang            PHI  PF         0.244103   3567882   29
<=30yo
Tari Eason               HOU  PF         0.234081   3458899   21
<=30yo
Aleksej Pokusevski       OKC  PF         0.276719   3358319   21
<=30yo
Kevin Knox               POR  PF         0.112141   3089075   23
<=30yo
Trey Lyles               SAC  PF         0.257361   2702940   27
<=30yo
Zeke Nnaji               DEN  PF         0.196948   2695527   22
<=30yo
David Roddy              MEM  PF         0.135936   2665501   21
<=30yo
Bol Bol                  ORL  PF         0.329463   2265321   23
<=30yo
Jeremiah Robinson-Earl   OKC  PF         0.223911   2059383   22
<=30yo
Kenrich Williams         OKC  PF         0.370002   2059383   28
<=30yo
Dean Wade                CLE  PF         0.356323   1988006   26
<=30yo
Keita Bates-Diop         SAS  PF         0.258812   1934502   27
<=30yo
Oshae Brissett           IND  PF         0.117490   1901570   24
<=30yo
Haywood Highsmith        MIA  PF         0.207958   1804676   26
<=30yo
JT Thor                  CHO  PF         0.090607   1609941   20
<=30yo
Isaiah Livers            DET  PF         0.182456   1609941   24
<=30yo
Trendon Watford          POR  PF         0.249674   1609941   22
<=30yo
```

Gurobi

- 2 players per position, sum of 10 players
- Sum of total salaries is at most 123.655 million
- Total salaries for each position should be at most (of total salaries):
- PG: 23.10%

- SG: 20.98%
- SF: 21.47%
- PF: 16.50%
- Center: 17.96%
- Maximum age is 30 y.o. for each player hired

```
total_salary = 123655000

salary_budget = pd.DataFrame(index =
list(optimization_player_final['Pos'].unique()), columns =
['salary_perc'])
salary_budget.loc['PF', 'salary_perc'] = 0.165
salary_budget.loc['C', 'salary_perc'] = 0.1796
salary_budget.loc['SF', 'salary_perc'] = 0.2147
salary_budget.loc['SG', 'salary_perc'] = 0.2097
salary_budget.loc['PG', 'salary_perc'] = 0.2310

salary_budget['budget'] = [x*total_salary for x in
salary_budget['salary_perc']]
salary_budget
```

```
   salary_perc      budget
PF       0.165  20403075.0
C       0.1796  22208438.0
SF      0.2147  26548728.5
SG      0.2097  25930453.5
PG       0.231  28564305.0
```

```
salary_budget.salary_perc.sum()
```

```
1.0
```

```
all_final_player = pd.DataFrame()
all_final_player
```

```
Empty DataFrame
Columns: []
Index: []
```

```
all_final_player = pd.DataFrame()

for pos in list(optimization_player_final['Pos'].unique()):
    import pandas as pd
    from gurobipy import Model, GRB
    I = optimization_player_final[optimization_player_final['Pos'] ==
pos].index
    mod = Model()
    x = mod.addVars(I, vtype = GRB.BINARY, name = 'x')
    mod.setObjective(sum(x[i]*optimization_player_final.loc[i,
'Weighted_Score'] for i in I), sense = GRB.MAXIMIZE)
```

```
    # total players per position
    mod.addConstr(sum(x[i] for i in I) == 2)

    # budget per position
    mod.addConstr(sum(x[i]*optimization_player_final.loc[i, 'Salary']
for i in I) <= salary_budget.loc[pos, 'budget'])

    # age
    for i in I:
        if optimization_player_final.loc[i, 'Age'] > 30:
            mod.addConstr(x[i] == 0)

    # mod.write('10-nba.lp')
    # %cat 10-nba.lp
    mod.setParam('OutputFlag', False)
    mod.optimize()
    mod.objval

    player = []

    for i in I:
        if x[i].x == 1:
            player.append(i)

    chosen = optimization_player_final.loc[player]

    all_final_player = pd.concat([all_final_player, chosen],
sort=False)

all_final_player.sort_values(by = ['Pos', 'Weighted_Score'])
```

```
Set parameter Username
Academic license - for non-commercial use only - expires 2024-10-04
```

index	Tm	Pos	Weighted_Score	Salary	Age	Age_Category
Walker Kessler	UTA	C	0.310922	2696400	21	<=30yo
Nic Claxton	BRK	C	0.420568	9350000	23	<=30yo
Brandon Clarke	MEM	PF	0.547184	4472897	26	<=30yo
Evan Mobley	CLE	PF	0.619509	8730468	21	<=30yo
Tyrese Haliburton	IND	PG	0.850688	4340274	22	<=30yo
Fred VanVleet	TOR	PG	0.911058	21880950	28	<=30yo
Amir Coffey	TOR	SF	0.746137	3395062	25	<=30yo
Cam Reddish	MIL	SF	0.776684	5954454	23	<=30yo
Immanuel Quickley	NYK	SG	0.733209	2385013	23	<=30yo
Derrick White	BOS	SG	0.798223	17651858	28	<=30yo

```
all_final_player['Salary'].max()
```

```
21880950
```

```
all_final_player['Weighted_Score'].sum()
```

```
6.714182462357157
```

```
list(optimization_player_final.index)
```

```
['Aaron Gordon',
 'Al Horford',
 'Aleksej Pokusevski',
 'Alperen Şengün',
 'Amir Coffey',
 'Andrew Nembhard',
 'Anfernee Simons',
 'Anthony Davis',
 'Anthony Edwards',
 'Ayo Dosunmu',
 'Bam Adebayo',
 'Bennedict Mathurin',
 'Bobby Portis',
 'Bol Bol',
 'Bradley Beal',
 'Brandon Clarke',
 'Brandon Ingram',
 'Brook Lopez',
 'Caleb Houstan',
 'Caleb Martin',
 'Cam Reddish',
 'Cameron Johnson',
 'Caris LeVert',
 'Chris Boucher',
 'Christian Wood',
 'Clint Capela',
 'Coby White',
 "D'Angelo Russell",
 'Damian Lillard',
 'Daniel Gafford',
 'Darius Garland',
 'David Roddy',
 "De'Aaron Fox",
 "De'Anthony Melton",
 'Dean Wade',
 'Deandre Ayton',
 'Dejounte Murray',
 'Deni Avdija',
 'Derrick White',
 'Domantas Sabonis',
 'Donovan Mitchell',
 'Dorian Finney-Smith',
 'Draymond Green',
 'Drew Eubanks',
```

```
'Eric Gordon',
'Eugene Omoruyi',
'Evan Mobley',
'Fred VanVleet',
'Gary Trent Jr.',
'Georges Niang',
'Giannis Antetokounmpo',
'Grant Williams',
'Grayson Allen',
'Hamidou Diallo',
'Harrison Barnes',
'Haywood Highsmith',
'Immanuel Quickley',
'Isaiah Livers',
'Ivica Zubac',
'JT Thor',
'JaMychal Green',
'Jabari Walker',
'Jack White',
'Jaden Ivey',
'Jakob Poeltl',
'Jalen Brunson',
'Jalen Duren',
'Jalen Green',
'Jalen Williams',
'Jamal Murray',
'James Harden',
'Jaren Jackson Jr.',
'Jarred Vanderbilt',
'Jarrett Allen',
'Javonte Green',
'Jeff Green',
'Jerami Grant',
'Jeremiah Robinson-Earl',
'Jeremy Sochan',
'Jimmy Butler',
'Joe Ingles',
'Joel Embiid',
'John Collins',
'Jonas Valančiūnas',
'Jonathan Kuminga',
'Jordan Clarkson',
'Jordan Poole',
'Josh Giddey',
'Josh Minott',
'Jrue Holiday',
'Juan Toscano-Anderson',
'Julius Randle',
'Justin Minaya',
```

```
 'Keita Bates-Diop',
 'Kelly Olynyk',
 'Kenrich Williams',
 'Kentavious Caldwell-Pope',
 'Kevin Durant',
 'Kevin Huerter',
 'Kevin Knox',
 'Kevin Love',
 'Kevin Porter Jr.',
 'Kevon Looney',
 'Killian Hayes',
 'Kristaps Porziņģis',
 'Kyle Anderson',
 'Kyle Kuzma',
 'Kyrie Irving',
 'Lauri Markkanen',
 'LeBron James',
 'Luguentz Dort',
 'Luka Dončić',
 'Malik Beasley',
 'Marcus Morris',
 'Markelle Fultz',
 'Mason Plumlee',
 'Max Strus',
 'Maxi Kleber',
 'Mike Conley',
 'Myles Turner',
 'Nic Claxton',
 'Nicolas Batum',
 'Nikola Jokić',
 'Nikola Vučević',
 'Obi Toppin',
 'Onyeka Okongwu',
 'Oshae Brissett',
 'Ousmane Dieng',
 'Paolo Banchero',
 'Pascal Siakam',
 'Patrick Beverley',
 'Patrick Williams',
 'Quentin Grimes',
 'RJ Barrett',
 'Robert Covington',
 'Rudy Gay',
 'Rudy Gobert',
 'Rui Hachimura',
 'Russell Westbrook',
 'Shaedon Sharpe',
 'Shai Gilgeous-Alexander',
 'Simone Fontecchio',
```

```
 'Spencer Dinwiddie',
 'Sterling Brown',
 'Tari Eason',
 'Taurean Prince',
 'Terry Rozier',
 'Thaddeus Young',
 'Torrey Craig',
 'Trae Young',
 'Tre Jones',
 'Trendon Watford',
 'Trey Lyles',
 'Tyler Dorsey',
 'Tyler Herro',
 'Tyrese Haliburton',
 'Tyrese Maxey',
 'Walker Kessler',
 'Wendell Carter Jr.',
 'Xavier Sneed',
 'Yuta Watanabe',
 'Zach LaVine',
 'Zeke Nnaji']
```

```
all_final_player.describe()
```

```
       Weighted_Score        Salary         Age
count       10.000000  1.000000e+01  10.000000
mean         0.671418  8.085738e+06  24.000000
std          0.193545  6.658343e+06   2.624669
min          0.310922  2.385013e+06  21.000000
25%          0.565266  3.631365e+06  22.250000
50%          0.739673  5.213676e+06  23.000000
75%          0.792838  9.195117e+06  25.750000
max          0.911058  2.188095e+07  28.000000
```

```
all_final_player['Salary'].sum()/total_salary
```

```
0.6538949173102584
```

```
all_final_player['Tm'].value_counts().reset_index()
```

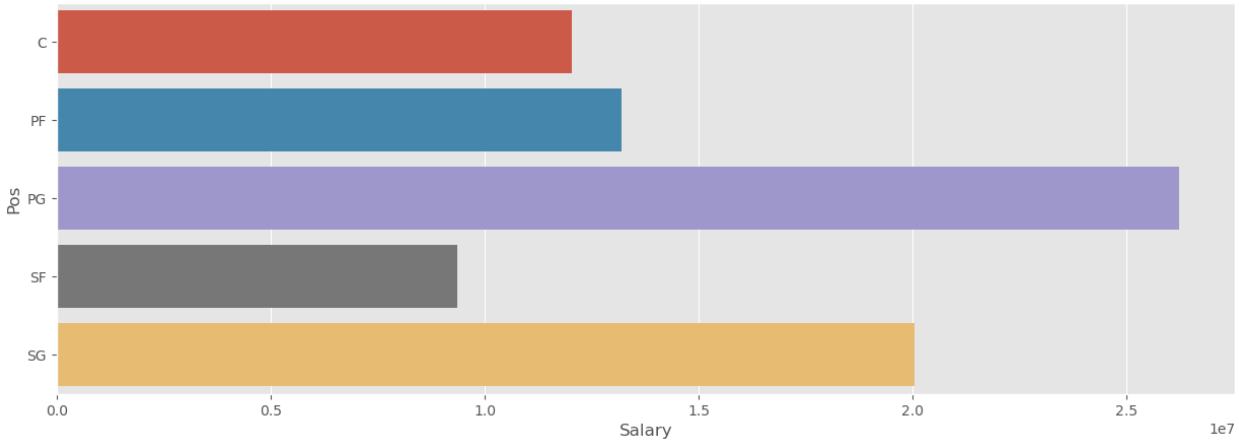
```
    Tm  count
0  TOR      2
1  MEM      1
2  CLE      1
3  BRK      1
4  UTA      1
5  MIL      1
6  BOS      1
7  NYK      1
8  IND      1
```

```
pos_grouped = all_final_player.groupby('Pos').sum()
['Salary'].reset_index().rename(columns={"Salary": "Budget Spent"})

pos_grouped['% Budget Spent'] = pos_grouped['Budget
Spent']/total_salary
pos_grouped
```

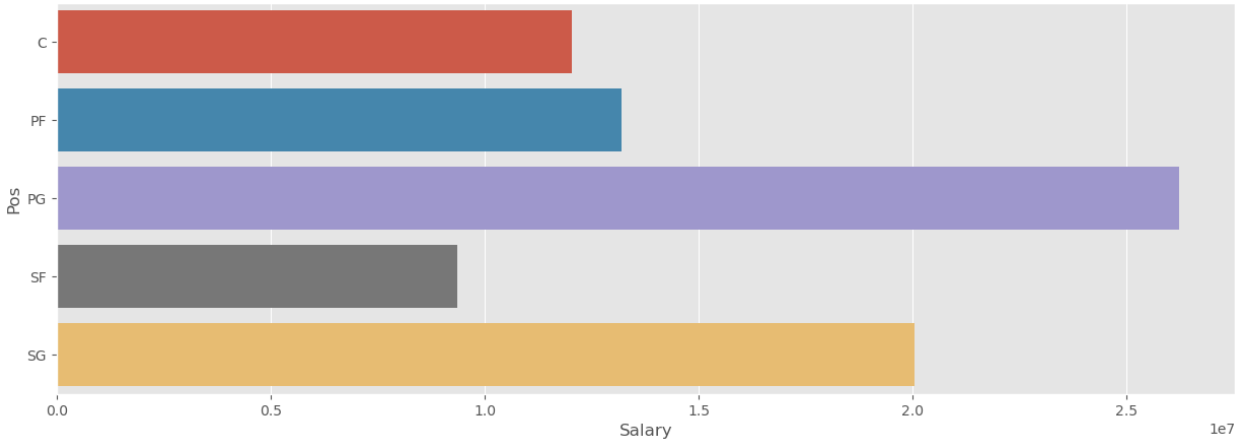
	Pos	Budget Spent	% Budget Spent
0	C	12046400	0.097419
1	PF	13203365	0.106776
2	PG	26221224	0.212051
3	SF	9349516	0.075610
4	SG	20036871	0.162039

```
pos_grouped.to_clipboard(index = False)

sns.barplot(all_final_player.groupby('Pos').sum()
['Salary'].reset_index(), x = 'Salary', y = 'Pos')
```

```
<Axes: xlabel='Salary', ylabel='Pos'>
```




```
groupedvalues = sns.barplot(all_final_player.groupby('Pos').sum()
['Salary'].reset_index(), x = 'Salary', y = 'Pos')

ax = sns.barplot(all_final_player.groupby('Pos').sum()
['Salary'].reset_index(), x = 'Salary', y = 'Pos')

# now use a for loop to iterate through
# each row of the grouped dataframe
# assign bar value  to each row
```


C
PF
PG
SF
SG
Pos
0.0
0.5
1.0
1.5
2.0
2.5
1e7
Salary