

# Basketball Analyze

August 17, 2021

## 1 Basketball Analyze

```
[4]: #Dear Student,
#
#Welcome to the world of Basketball Data!
#I'm sure you will enjoy this section of the Python Programming course.
#
#Instructions for this dataset:
# Simply copy ALL the lines in this script by pressing
# CTRL+A on Windows or CMND+A on Mac and run the Jupyter cell
# Once you have executed the commands the following objects
# will be created:
# Matrices:
# - Salary
# - Games
# - MinutesPlayed
# - FieldGoals
# - FieldGoalAttempts
# - Points
# Lists:
# - Players
# - Seasons
# Dictionaries:
# - Sdict
# - Pdicit
#We will understand these inside the course.
#
#Sincerely,
#Kirill Eremenko
#www.superdatascience.com

#Copyright: These datasets were prepared using publicly available data.
#           However, theses scripts are subject to Copyright Laws.
#           If you wish to use these Python scripts outside of the Python
↪Programming Course
#           by Kirill Eremenko, you may do so by referencing www.
↪superdatascience.com in your work.
```

```

#Comments:
#Seasons are labeled based on the first year in the season
#E.g. the 2012-2013 season is preseneted as simply 2012

#Notes and Corrections to the data:
#Kevin Durant: 2006 - College Data Used
#Kevin Durant: 2005 - Proxied With 2006 Data
#Derrick Rose: 2012 - Did Not Play
#Derrick Rose: 2007 - College Data Used
#Derrick Rose: 2006 - Proxied With 2007 Data
#Derrick Rose: 2005 - Proxied With 2007 Data

#Import numpy
import numpy as np

#Seasons
Seasons =_
↳ ["2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014"]
Sdict = {"2005":0, "2006":1, "2007":2, "2008":3, "2009":4, "2010":5, "2011":6, "2012":
↳ 7, "2013":8, "2014":9}

#Players
Players =_
↳ ["KobeBryant", "JoeJohnson", "LeBronJames", "CarmeloAnthony", "DwightHoward", "ChrisBosh", "Chris
↳ "DerrickRose", "DwayneWade"]
Pdict = {"KobeBryant":0, "JoeJohnson":1, "LeBronJames":2, "CarmeloAnthony":
↳ 3, "DwightHoward":4, "ChrisBosh":5, "ChrisPaul":6,
↳ "KevinDurant":7, "DerrickRose":8, "DwayneWade":9}

#Salaries
KobeBryant_Salary =_
↳ [15946875, 17718750, 19490625, 21262500, 23034375, 24806250, 25244493, 27849149, 30453805, 23500000]
JoeJohnson_Salary =_
↳ [12000000, 12744189, 13488377, 14232567, 14976754, 16324500, 18038573, 19752645, 21466718, 23180790]
LeBronJames_Salary =_
↳ [4621800, 5828090, 13041250, 14410581, 15779912, 14500000, 16022500, 17545000, 19067500, 20644400]
CarmeloAnthony_Salary =_
↳ [3713640, 4694041, 13041250, 14410581, 15779912, 17149243, 18518574, 19450000, 22407474, 22458000]
DwightHoward_Salary =_
↳ [4493160, 4806720, 6061274, 13758000, 15202590, 16647180, 18091770, 19536360, 20513178, 21436271]
ChrisBosh_Salary =_
↳ [3348000, 4235220, 12455000, 14410581, 15779912, 14500000, 16022500, 17545000, 19067500, 20644400]
ChrisPaul_Salary =_
↳ [3144240, 3380160, 3615960, 4574189, 13520500, 14940153, 16359805, 17779458, 18668431, 20068563]

```

```

KevinDurant_Salary =
    ↳ [0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832627,18995624]
DerrickRose_Salary =
    ↳ [0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,18862875]
DwayneWade_Salary =
    ↳ [3031920,3841443,13041250,14410581,15779912,14200000,15691000,17182000,18673000,15000000]

#Matrix
Salary = np.array([KobeBryant_Salary, JoeJohnson_Salary, LeBronJames_Salary,
    ↳ CarmeloAnthony_Salary, DwightHoward_Salary,
    ChrisBosh_Salary, ChrisPaul_Salary, KevinDurant_Salary,
    ↳ DerrickRose_Salary, DwayneWade_Salary])

#Games
KobeBryant_G = [80,77,82,82,73,82,58,78,6,35]
JoeJohnson_G = [82,57,82,79,76,72,60,72,79,80]
LeBronJames_G = [79,78,75,81,76,79,62,76,77,69]
CarmeloAnthony_G = [80,65,77,66,69,77,55,67,77,40]
DwightHoward_G = [82,82,82,79,82,78,54,76,71,41]
ChrisBosh_G = [70,69,67,77,70,77,57,74,79,44]
ChrisPaul_G = [78,64,80,78,45,80,60,70,62,82]
KevinDurant_G = [35,35,80,74,82,78,66,81,81,27]
DerrickRose_G = [40,40,40,81,78,81,39,0,10,51]
DwayneWade_G = [75,51,51,79,77,76,49,69,54,62]

#Matrix
Games = np.array([KobeBryant_G, JoeJohnson_G, LeBronJames_G, CarmeloAnthony_G,
    ↳ DwightHoward_G, ChrisBosh_G, ChrisPaul_G,
    KevinDurant_G, DerrickRose_G, DwayneWade_G])

#Minutes Played
KobeBryant_MP = [3277,3140,3192,2960,2835,2779,2232,3013,177,1207]
JoeJohnson_MP = [3340,2359,3343,3124,2886,2554,2127,2642,2575,2791]
LeBronJames_MP = [3361,3190,3027,3054,2966,3063,2326,2877,2902,2493]
CarmeloAnthony_MP = [2941,2486,2806,2277,2634,2751,1876,2482,2982,1428]
DwightHoward_MP = [3021,3023,3088,2821,2843,2935,2070,2722,2396,1223]
ChrisBosh_MP = [2751,2658,2425,2928,2526,2795,2007,2454,2531,1556]
ChrisPaul_MP = [2808,2353,3006,3002,1712,2880,2181,2335,2171,2857]
KevinDurant_MP = [1255,1255,2768,2885,3239,3038,2546,3119,3122,913]
DerrickRose_MP = [1168,1168,1168,3000,2871,3026,1375,0,311,1530]
DwayneWade_MP = [2892,1931,1954,3048,2792,2823,1625,2391,1775,1971]

#Matrix
MinutesPlayed = np.array([KobeBryant_MP, JoeJohnson_MP, LeBronJames_MP,
    ↳ CarmeloAnthony_MP, DwightHoward_MP, ChrisBosh_MP,
    ChrisPaul_MP, KevinDurant_MP, DerrickRose_MP,
    ↳ DwayneWade_MP])

#Field Goals

```

```

KobeBryant_FG = [978,813,775,800,716,740,574,738,31,266]
JoeJohnson_FG = [632,536,647,620,635,514,423,445,462,446]
LeBronJames_FG = [875,772,794,789,768,758,621,765,767,624]
CarmeloAnthony_FG = [756,691,728,535,688,684,441,669,743,358]
DwightHoward_FG = [468,526,583,560,510,619,416,470,473,251]
ChrisBosh_FG = [549,543,507,615,600,524,393,485,492,343]
ChrisPaul_FG = [407,381,630,631,314,430,425,412,406,568]
KevinDurant_FG = [306,306,587,661,794,711,643,731,849,238]
DerrickRose_FG = [208,208,208,574,672,711,302,0,58,338]
DwayneWade_FG = [699,472,439,854,719,692,416,569,415,509]

#Matrix
FieldGoals = np.array([KobeBryant_FG, JoeJohnson_FG, LeBronJames_FG,
    ↪CarmeloAnthony_FG, DwightHoward_FG, ChrisBosh_FG,
    ChrisPaul_FG, KevinDurant_FG, DerrickRose_FG,
    ↪DwayneWade_FG])

#Field Goal Attempts
KobeBryant_FGA = [2173,1757,1690,1712,1569,1639,1336,1595,73,713]
JoeJohnson_FGA = [1395,1139,1497,1420,1386,1161,931,1052,1018,1025]
LeBronJames_FGA = [1823,1621,1642,1613,1528,1485,1169,1354,1353,1279]
CarmeloAnthony_FGA = [1572,1453,1481,1207,1502,1503,1025,1489,1643,806]
DwightHoward_FGA = [881,873,974,979,834,1044,726,813,800,423]
ChrisBosh_FGA = [1087,1094,1027,1263,1158,1056,807,907,953,745]
ChrisPaul_FGA = [947,871,1291,1255,637,928,890,856,870,1170]
KevinDurant_FGA = [647,647,1366,1390,1668,1538,1297,1433,1688,467]
DerrickRose_FGA = [436,436,436,1208,1373,1597,695,0,164,835]
DwayneWade_FGA = [1413,962,937,1739,1511,1384,837,1093,761,1084]

#Matrix
FieldGoalAttempts = np.array([KobeBryant_FGA, JoeJohnson_FGA, LeBronJames_FGA,
    ↪CarmeloAnthony_FGA, DwightHoward_FGA,
    ChrisBosh_FGA, ChrisPaul_FGA, KevinDurant_FGA,
    ↪DerrickRose_FGA, DwayneWade_FGA])

#Points
KobeBryant_PTS = [2832,2430,2323,2201,1970,2078,1616,2133,83,782]
JoeJohnson_PTS = [1653,1426,1779,1688,1619,1312,1129,1170,1245,1154]
LeBronJames_PTS = [2478,2132,2250,2304,2258,2111,1683,2036,2089,1743]
CarmeloAnthony_PTS = [2122,1881,1978,1504,1943,1970,1245,1920,2112,966]
DwightHoward_PTS = [1292,1443,1695,1624,1503,1784,1113,1296,1297,646]
ChrisBosh_PTS = [1572,1561,1496,1746,1678,1438,1025,1232,1281,928]
ChrisPaul_PTS = [1258,1104,1684,1781,841,1268,1189,1186,1185,1564]
KevinDurant_PTS = [903,903,1624,1871,2472,2161,1850,2280,2593,686]
DerrickRose_PTS = [597,597,597,1361,1619,2026,852,0,159,904]
DwayneWade_PTS = [2040,1397,1254,2386,2045,1941,1082,1463,1028,1331]

#Matrix
Points = np.array([KobeBryant_PTS, JoeJohnson_PTS, LeBronJames_PTS,
    ↪CarmeloAnthony_PTS, DwightHoward_PTS, ChrisBosh_PTS,

```

```
ChrisPaul_PTS, KevinDurant_PTS, DerrickRose_PTS, \
↳DwayneWade_PTS])
```

```
[5]: print(Salary)
```

```
[[15946875 17718750 19490625 21262500 23034375 24806250 25244493 27849149
 30453805 23500000]
 [12000000 12744189 13488377 14232567 14976754 16324500 18038573 19752645
 21466718 23180790]
 [ 4621800  5828090 13041250 14410581 15779912 14500000 16022500 17545000
 19067500 20644400]
 [ 3713640  4694041 13041250 14410581 15779912 17149243 18518574 19450000
 22407474 22458000]
 [ 4493160  4806720  6061274 13758000 15202590 16647180 18091770 19536360
 20513178 21436271]
 [ 3348000  4235220 12455000 14410581 15779912 14500000 16022500 17545000
 19067500 20644400]
 [ 3144240  3380160  3615960  4574189 13520500 14940153 16359805 17779458
 18668431 20068563]
 [      0      0 4171200  4484040  4796880  6053663 15506632 16669630
 17832627 18995624]
 [      0      0      0 4822800  5184480  5546160  6993708 16402500
 17632688 18862875]
 [ 3031920  3841443 13041250 14410581 15779912 14200000 15691000 17182000
 18673000 15000000]]
```

```
[6]: print(Points)
```

```
[[2832 2430 2323 2201 1970 2078 1616 2133   83  782]
 [1653 1426 1779 1688 1619 1312 1129 1170 1245 1154]
 [2478 2132 2250 2304 2258 2111 1683 2036 2089 1743]
 [2122 1881 1978 1504 1943 1970 1245 1920 2112  966]
 [1292 1443 1695 1624 1503 1784 1113 1296 1297  646]
 [1572 1561 1496 1746 1678 1438 1025 1232 1281  928]
 [1258 1104 1684 1781  841 1268 1189 1186 1185 1564]
 [ 903  903 1624 1871 2472 2161 1850 2280 2593  686]
 [ 597  597  597 1361 1619 2026  852    0  159  904]
 [2040 1397 1254 2386 2045 1941 1082 1463 1028 1331]]
```

```
[7]: import numpy as np
mydata=np.arange(0,20)
print(mydata)
type(mydata)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
```

```
[7]: numpy.ndarray
```

```
[8]: np.reshape(mydata,(5,4))
```

```
[8]: array([[ 0,  1,  2,  3],
           [ 4,  5,  6,  7],
           [ 8,  9, 10, 11],
           [12, 13, 14, 15],
           [16, 17, 18, 19]])
```

```
[9]: mart1=np.reshape(mydata,(5,4),order="C")
     print(mart1)
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]
 [16 17 18 19]]
```

```
[10]: mart1[2,2]
```

```
[10]: 10
```

```
[11]: mart2=np.reshape(mydata,(5,4), order="F")
     print(mart2)
```

```
[[ 0  5 10 15]
 [ 1  6 11 16]
 [ 2  7 12 17]
 [ 3  8 13 18]
 [ 4  9 14 19]]
```

```
[12]: mart2[0,2]
```

```
[12]: 10
```

```
[13]: mydata.reshape((5,4))
```

```
[13]: array([[ 0,  1,  2,  3],
           [ 4,  5,  6,  7],
           [ 8,  9, 10, 11],
           [12, 13, 14, 15],
           [16, 17, 18, 19]])
```

```
[14]: r1=["I","AM","HAPPY"]
     r2=["WHAT","A","DAY"]
     r3=[1,2,3]
     [r1,r2,r3]
```

```
[14]: [['I', 'AM', 'HAPPY'], ['WHAT', 'A', 'DAY'], [1, 2, 3]]
```

```
[15]: np.array([r1,r2,r3])
```

```
[15]: array([[ 'I', 'AM', 'HAPPY'],  
          ['WHAT', 'A', 'DAY'],  
          ['1', '2', '3']], dtype='<U11')
```

```
[16]: print(np.array([r1,r2,r3]))
```

```
['I' 'AM' 'HAPPY']  
['WHAT' 'A' 'DAY']  
['1' '2' '3']
```

---

## 2 Dictionaries

```
[17]: Games
```

```
[17]: array([[80, 77, 82, 82, 73, 82, 58, 78,  6, 35],  
          [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],  
          [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],  
          [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],  
          [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],  
          [70, 69, 67, 77, 70, 77, 57, 74, 79, 44],  
          [78, 64, 80, 78, 45, 80, 60, 70, 62, 82],  
          [35, 35, 80, 74, 82, 78, 66, 81, 81, 27],  
          [40, 40, 40, 81, 78, 81, 39,  0, 10, 51],  
          [75, 51, 51, 79, 77, 76, 49, 69, 54, 62]])
```

```
[18]: Games[0]
```

```
[18]: array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

```
[19]: Games[2][9]
```

```
[19]: 69
```

```
[20]: Games[2][-1]
```

```
[20]: 69
```

```
[21]: Games[2,-1]
```

```
[21]: 69
```

```
[22]: Games[2,9]
```

```
[22]: 69
```

```
[23]: Points
```

```
[23]: array([[2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133, 83, 782],
           [1653, 1426, 1779, 1688, 1619, 1312, 1129, 1170, 1245, 1154],
           [2478, 2132, 2250, 2304, 2258, 2111, 1683, 2036, 2089, 1743],
           [2122, 1881, 1978, 1504, 1943, 1970, 1245, 1920, 2112, 966],
           [1292, 1443, 1695, 1624, 1503, 1784, 1113, 1296, 1297, 646],
           [1572, 1561, 1496, 1746, 1678, 1438, 1025, 1232, 1281, 928],
           [1258, 1104, 1684, 1781, 841, 1268, 1189, 1186, 1185, 1564],
           [ 903, 903, 1624, 1871, 2472, 2161, 1850, 2280, 2593, 686],
           [ 597, 597, 597, 1361, 1619, 2026, 852, 0, 159, 904],
           [2040, 1397, 1254, 2386, 2045, 1941, 1082, 1463, 1028, 1331]])
```

```
[24]: Points[6]
```

```
[24]: array([1258, 1104, 1684, 1781, 841, 1268, 1189, 1186, 1185, 1564])
```

```
[25]: Points[6,1]
```

```
[25]: 1104
```

```
[26]: dict1={"key1":"val1","key2":"val2","key3":"val3"}
      dict1
```

```
[26]: {'key1': 'val1', 'key2': 'val2', 'key3': 'val3'}
```

```
[27]: dict1["key1"]
```

```
[27]: 'val1'
```

```
[28]: dict1["key2"]
```

```
[28]: 'val2'
```

```
[29]: dict2={"Germany":"I have been here","France":2,"Spain":True}
      dict2
```

```
[29]: {'Germany': 'I have been here', 'France': 2, 'Spain': True}
```

```
[30]: dict2["France"]
```

```
[30]: 2
```

```
[31]: print(Pdict)
```

```
{'KobeBryant': 0, 'JoeJohnson': 1, 'LeBronJames': 2, 'CarmeloAnthony': 3,
'DwightHoward': 4, 'ChrisBosh': 5, 'ChrisPaul': 6, 'KevinDurant': 7,
'DerrickRose': 8, 'DwayneWade': 9}
```



```
[32]: print(Sdict)

{'2005': 0, '2006': 1, '2007': 2, '2008': 3, '2009': 4, '2010': 5, '2011': 6,
'2012': 7, '2013': 8, '2014': 9}
```

```
[33]: Pdict["KobeBryant"]
```

```
[33]: 0
```

```
[34]: Games[0]
```

```
[34]: array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

```
[35]: Points[0]
```

```
[35]: array([2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133,  83, 782])
```

```
[36]: Games[Pdict["KobeBryant"]]
```

```
[36]: array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

```
[37]: Points[Pdict["KobeBryant"]]
```

```
[37]: array([2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133,  83, 782])
```

```
[38]: print(Salary[Pdict["KobeBryant"]])
```

```
[15946875 17718750 19490625 21262500 23034375 24806250 25244493 27849149
 30453805 23500000]
```

```
[39]: Games[Pdict["DerrickRose"]]
```

```
[39]: array([40, 40, 40, 81, 78, 81, 39,  0, 10, 51])
```

```
[40]: Games[Pdict["DerrickRose"]][7]
```

```
[40]: 0
```

```
[41]: Games[Pdict["DerrickRose"]][Sdict["2012"]]
```

```
[41]: 0
```

```
[42]: Points[Pdict["JoeJohnson"]][Sdict["2010"]]
```

```
[42]: 1312
```

```
[43]: Points[Pdict["JoeJohnson"]]
```

```
[43]: array([1653, 1426, 1779, 1688, 1619, 1312, 1129, 1170, 1245, 1154])
```

```
[44]: Points[Pdict["JoeJohnson"]][Sdict['2014']]
```

```
[44]: 1154
```

---

### 3 Metrice Operations

```
[45]: print(Pdict["LeBronJames"])  
      print(Sdict["2010"])
```

```
2  
5
```

```
[46]: Salary[2][5]
```

```
[46]: 14500000
```

```
[47]: Salary[Pdict["LeBronJames"]][Sdict["2010"]]
```

```
[47]: 14500000
```

```
[48]: FieldGoals
```

```
[48]: array([[978, 813, 775, 800, 716, 740, 574, 738, 31, 266],  
            [632, 536, 647, 620, 635, 514, 423, 445, 462, 446],  
            [875, 772, 794, 789, 768, 758, 621, 765, 767, 624],  
            [756, 691, 728, 535, 688, 684, 441, 669, 743, 358],  
            [468, 526, 583, 560, 510, 619, 416, 470, 473, 251],  
            [549, 543, 507, 615, 600, 524, 393, 485, 492, 343],  
            [407, 381, 630, 631, 314, 430, 425, 412, 406, 568],  
            [306, 306, 587, 661, 794, 711, 643, 731, 849, 238],  
            [208, 208, 208, 574, 672, 711, 302, 0, 58, 338],  
            [699, 472, 439, 854, 719, 692, 416, 569, 415, 509]])
```

```
[49]: Games
```

```
[49]: array([[80, 77, 82, 82, 73, 82, 58, 78, 6, 35],  
            [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],  
            [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],  
            [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],  
            [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],  
            [70, 69, 67, 77, 70, 77, 57, 74, 79, 44],  
            [78, 64, 80, 78, 45, 80, 60, 70, 62, 82],  
            [35, 35, 80, 74, 82, 78, 66, 81, 81, 27],  
            [40, 40, 40, 81, 78, 81, 39, 0, 10, 51],  
            [75, 51, 51, 79, 77, 76, 49, 69, 54, 62]])
```

```
[50]: import warnings
warnings.filterwarnings('ignore')
FieldGoals/Games
```

```
[50]: array([[12.225      , 10.55844156,  9.45121951,  9.75609756,  9.80821918,
           9.02439024,  9.89655172,  9.46153846,  5.16666667,  7.6        ],
          [ 7.70731707,  9.40350877,  7.8902439 ,  7.84810127,  8.35526316,
           7.13888889,  7.05        ,  6.18055556,  5.84810127,  5.575        ],
          [11.07594937,  9.8974359 , 10.58666667,  9.74074074, 10.10526316,
           9.59493671, 10.01612903, 10.06578947,  9.96103896,  9.04347826],
          [ 9.45        , 10.63076923,  9.45454545,  8.10606061,  9.97101449,
           8.88311688,  8.01818182,  9.98507463,  9.64935065,  8.95        ],
          [ 5.70731707,  6.41463415,  7.1097561 ,  7.08860759,  6.2195122 ,
           7.93589744,  7.7037037 ,  6.18421053,  6.66197183,  6.12195122],
          [ 7.84285714,  7.86956522,  7.56716418,  7.98701299,  8.57142857,
           6.80519481,  6.89473684,  6.55405405,  6.2278481 ,  7.79545455],
          [ 5.21794872,  5.953125 ,  7.875        ,  8.08974359,  6.97777778,
           5.375        ,  7.08333333,  5.88571429,  6.5483871 ,  6.92682927],
          [ 8.74285714,  8.74285714,  7.3375        ,  8.93243243,  9.68292683,
           9.11538462,  9.74242424,  9.02469136, 10.48148148,  8.81481481],
          [ 5.2        ,  5.2        ,  5.2        ,  7.08641975,  8.61538462,
           8.77777778,  7.74358974,          nan,  5.8        ,  6.62745098],
          [ 9.32        ,  9.25490196,  8.60784314, 10.81012658,  9.33766234,
           9.10526316,  8.48979592,  8.24637681,  7.68518519,  8.20967742]])
```

```
[51]: FieldGoalPerGame=np.matrix.round(FieldGoals/Games)
```

```
[52]: print(FieldGoalPerGame)
```

```
[[12. 11.  9. 10. 10.  9. 10.  9.  5.  8.]
 [ 8.  9.  8.  8.  8.  7.  7.  6.  6.  6.]
 [11. 10. 11. 10. 10. 10. 10. 10. 10.  9.]
 [ 9. 11.  9.  8. 10.  9.  8. 10. 10.  9.]
 [ 6.  6.  7.  7.  6.  8.  8.  6.  7.  6.]
 [ 8.  8.  8.  8.  9.  7.  7.  7.  6.  8.]
 [ 5.  6.  8.  8.  7.  5.  7.  6.  7.  7.]
 [ 9.  9.  7.  9. 10.  9. 10.  9. 10.  9.]
 [ 5.  5.  5.  7.  9.  9.  8. nan  6.  7.]
 [ 9.  9.  9. 11.  9.  9.  8.  8.  8.  8.]]
```

```
[53]: FieldGoalPerGame[Pdict["DerrickRose"]][Sdict["2013"]]
```

```
[53]: 6.0
```

```
[54]: MinutesPlayed
```

```
[54]: array([[3277, 3140, 3192, 2960, 2835, 2779, 2232, 3013, 177, 1207],
          [3340, 2359, 3343, 3124, 2886, 2554, 2127, 2642, 2575, 2791],
```

```
[3361, 3190, 3027, 3054, 2966, 3063, 2326, 2877, 2902, 2493],
[2941, 2486, 2806, 2277, 2634, 2751, 1876, 2482, 2982, 1428],
[3021, 3023, 3088, 2821, 2843, 2935, 2070, 2722, 2396, 1223],
[2751, 2658, 2425, 2928, 2526, 2795, 2007, 2454, 2531, 1556],
[2808, 2353, 3006, 3002, 1712, 2880, 2181, 2335, 2171, 2857],
[1255, 1255, 2768, 2885, 3239, 3038, 2546, 3119, 3122, 913],
[1168, 1168, 1168, 3000, 2871, 3026, 1375, 0, 311, 1530],
[2892, 1931, 1954, 3048, 2792, 2823, 1625, 2391, 1775, 1971]])
```

```
[55]: MinutesPlayedPerGame=np.matrix.round(MinutesPlayed/Games)
```

```
[56]: print(MinutesPlayedPerGame)
```

```
[[41. 41. 39. 36. 39. 34. 38. 39. 30. 34.]
 [41. 41. 41. 40. 38. 35. 35. 37. 33. 35.]
 [43. 41. 40. 38. 39. 39. 38. 38. 38. 36.]
 [37. 38. 36. 34. 38. 36. 34. 37. 39. 36.]
 [37. 37. 38. 36. 35. 38. 38. 36. 34. 30.]
 [39. 39. 36. 38. 36. 36. 35. 33. 32. 35.]
 [36. 37. 38. 38. 38. 36. 36. 33. 35. 35.]
 [36. 36. 35. 39. 40. 39. 39. 39. 39. 34.]
 [29. 29. 29. 37. 37. 37. 35. nan 31. 30.]
 [39. 38. 38. 39. 36. 37. 33. 35. 33. 32.]]
```

```
[57]: AVGSalaryPerGame=np.matrix.round(Salary/Games)
```

```
[58]: print(AVGSalaryPerGame)
```

```
[[ 199336.  230114.  237691.  259299.  315539.  302515.  435250.  357040.
   5075634.  671429.]
 [ 146341.  223582.  164492.  180159.  197063.  226729.  300643.  274342.
   271731.  289760.]
 [  58504.   74719.  173883.  177908.  207630.  183544.  258427.  230855.
   247630.  299194.]
 [  46420.   72216.  169367.  218342.  228694.  222717.  336701.  290299.
   291006.  561450.]
 [  54795.   58619.   73918.  174152.  185397.  213425.  335033.  257057.
   288918.  522836.]
 [  47829.   61380.  185896.  187150.  225427.  188312.  281096.  237095.
   241361.  469191.]
 [  40311.   52815.   45200.   58643.  300456.  186752.  272663.  253992.
   301104.  244739.]
 [    0.    0.   52140.   60595.   58499.   77611.  234949.  205798.
   220156.  703542.]
 [    0.    0.    0.   59541.   66468.   68471.  179326.    inf
   1763269.  369860.]
 [  40426.   75322.  255711.  182412.  204934.  186842.  320224.  249014.
   345796.  241935.]]
```

```
[59]: AccuracyPerGame=np.matrix.round(FieldGoals/FieldGoalAttempts,2)*100
```

```
[60]: print(AccuracyPerGame)
```

```
[45. 46. 46. 47. 46. 45. 43. 46. 42. 37.]  
[45. 47. 43. 44. 46. 44. 45. 42. 45. 44.]  
[48. 48. 48. 49. 50. 51. 53. 56. 57. 49.]  
[48. 48. 49. 44. 46. 46. 43. 45. 45. 44.]  
[53. 60. 60. 57. 61. 59. 57. 58. 59. 59.]  
[51. 50. 49. 49. 52. 50. 49. 53. 52. 46.]  
[43. 44. 49. 50. 49. 46. 48. 48. 47. 49.]  
[47. 47. 43. 48. 48. 46. 50. 51. 50. 51.]  
[48. 48. 48. 48. 49. 45. 43. nan 35. 40.]  
[49. 49. 47. 49. 48. 50. 50. 52. 55. 47.]
```

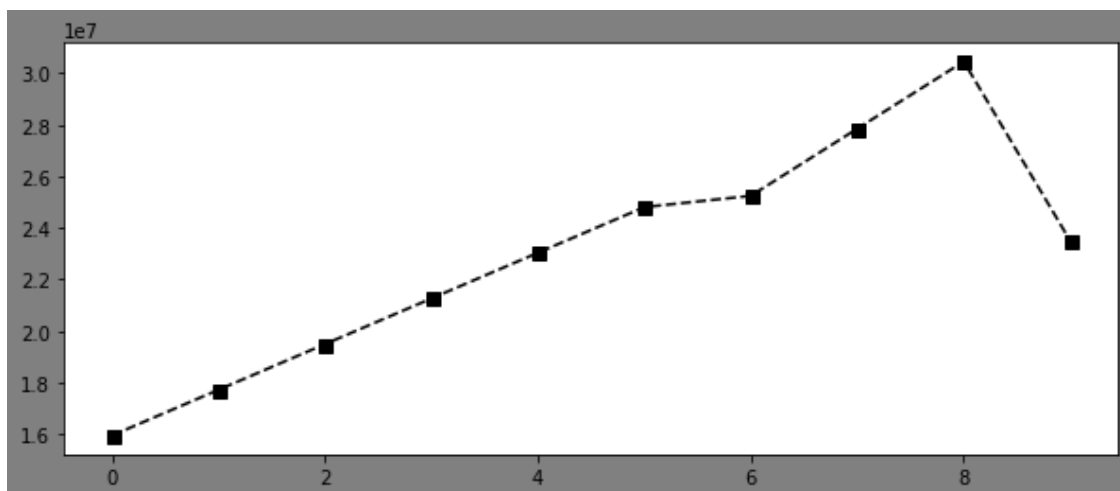
---

## 4 Virtualization

```
[61]: import numpy as np  
import matplotlib.pyplot as plt
```

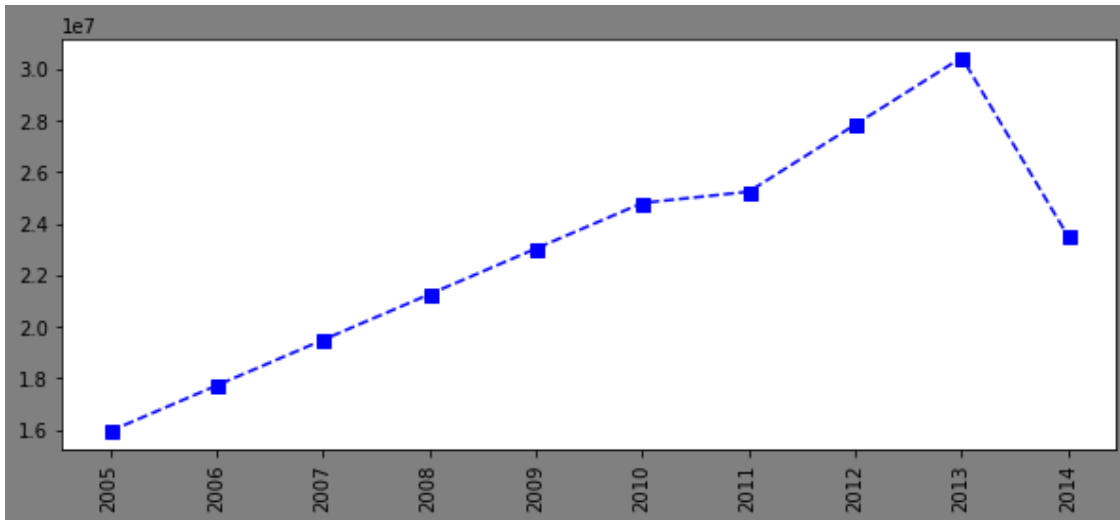
```
[62]: %matplotlib inline  
plt.rcParams['figure.figsize']=10,4  
plt.rcParams['figure.facecolor']='Gray'
```

```
[63]: plt.plot(Salary[0],c='Black',ls='--',marker='s',ms=7,label=Players[0])  
plt.show()
```



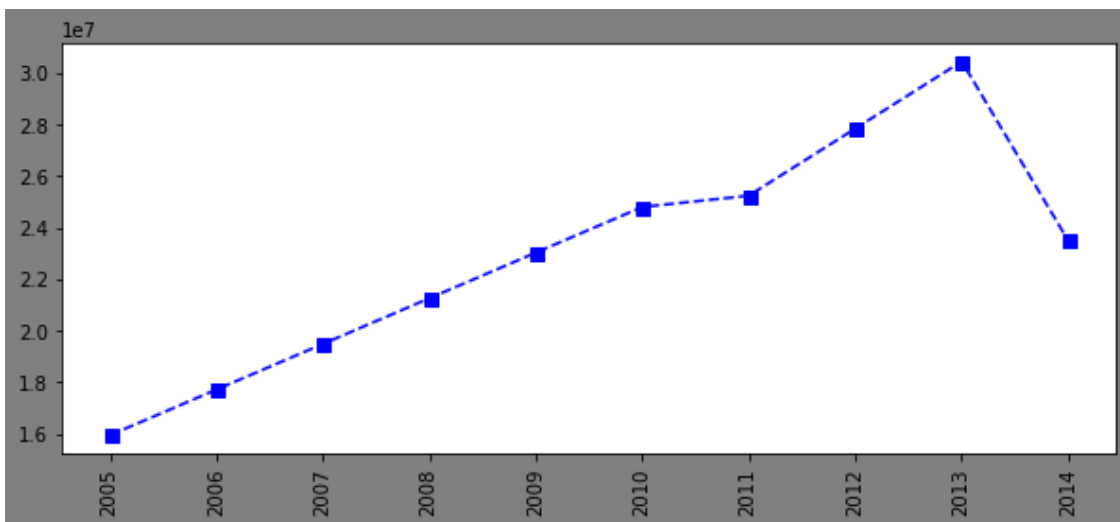
```
[64]: plt.plot(Salary[0],c="Blue",ls='--',marker='s',ms=7,label=Players[0])
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
print(Players[0])
plt.show()
```

KobeBryant



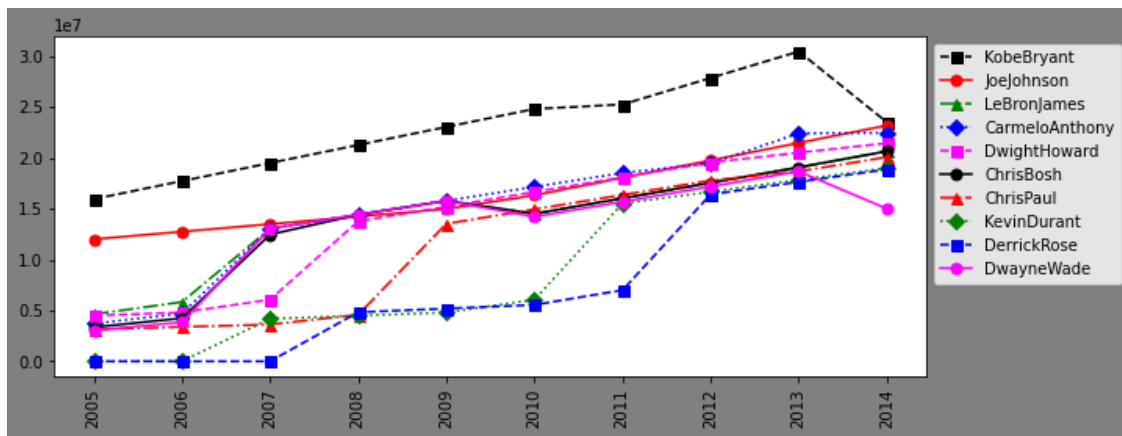
```
[65]: plt.plot(Salary[0],c="Blue",ls='--',marker='s',ms=7,label=Players[0])
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
print(Players[0])
plt.show()
```

KobeBryant



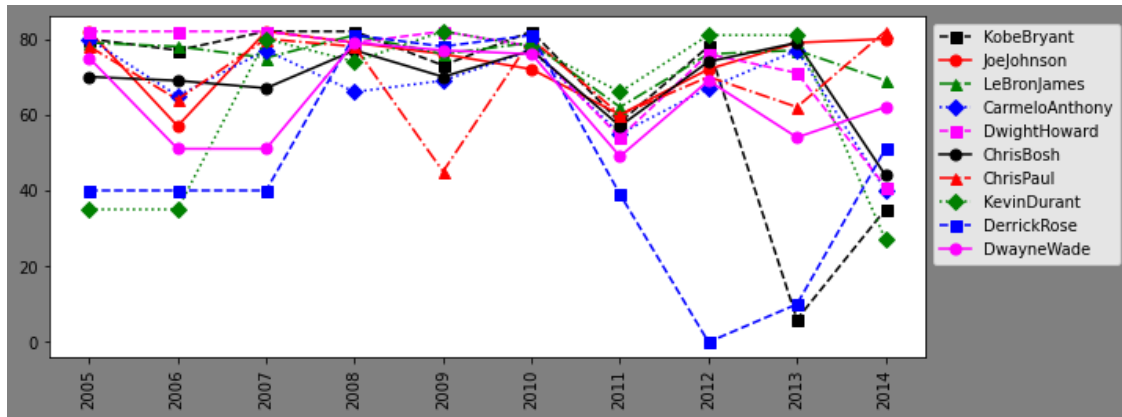
```
[66]: plt.plot(Salary[0],c="Black",ls='--',marker='s',ms=7,label=Players[0])
plt.plot(Salary[1],c="Red",ls='-',marker='o',ms=7,label=Players[1])
plt.plot(Salary[2],c="Green",ls='-.',marker='^',ms=7,label=Players[2])
plt.plot(Salary[3],c="Blue",ls=':',marker='D',ms=7,label=Players[3])
plt.plot(Salary[4],c="Magenta",ls='--',marker='s',ms=7,label=Players[4])
plt.plot(Salary[5],c="Black",ls='-',marker='o',ms=7,label=Players[5])
plt.plot(Salary[6],c="Red",ls='-.',marker='^',ms=7,label=Players[6])
plt.plot(Salary[7],c="Green",ls=':',marker='D',ms=7,label=Players[7])
plt.plot(Salary[8],c="Blue",ls='--',marker='s',ms=7,label=Players[8])
plt.plot(Salary[9],c="Magenta",ls='-',marker='o',ms=7,label=Players[9])

plt.xticks(list(range(0,10)),Seasons,rotation='vertical')# rotation='vertical'
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.show()
```



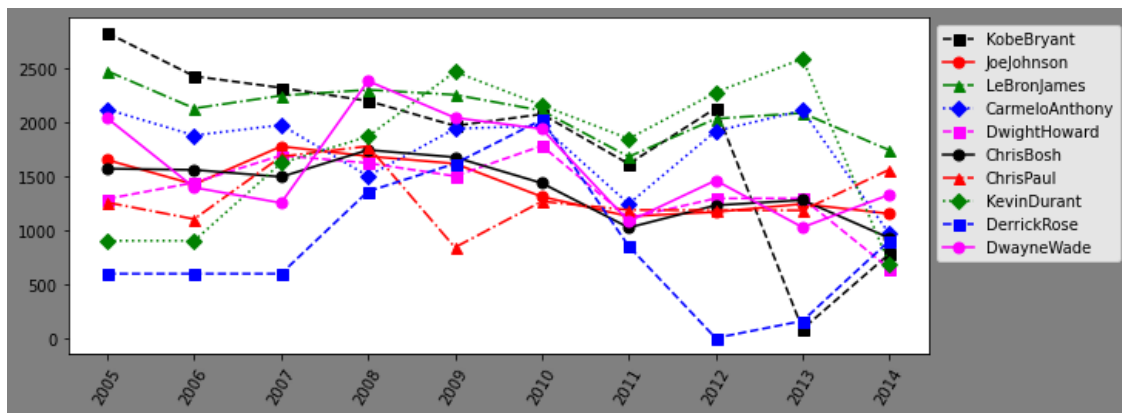
```
[67]: plt.plot(Games[0],c="Black",ls='--',marker='s',ms=7,label=Players[0])
plt.plot(Games[1],c="Red",ls='-',marker='o',ms=7,label=Players[1])
plt.plot(Games[2],c="Green",ls='-.',marker='^',ms=7,label=Players[2])
plt.plot(Games[3],c="Blue",ls=':',marker='D',ms=7,label=Players[3])
plt.plot(Games[4],c="Magenta",ls='--',marker='s',ms=7,label=Players[4])
plt.plot(Games[5],c="Black",ls='-',marker='o',ms=7,label=Players[5])
plt.plot(Games[6],c="Red",ls='-.',marker='^',ms=7,label=Players[6])
plt.plot(Games[7],c="Green",ls=':',marker='D',ms=7,label=Players[7])
plt.plot(Games[8],c="Blue",ls='--',marker='s',ms=7,label=Players[8])
plt.plot(Games[9],c="Magenta",ls='-',marker='o',ms=7,label=Players[9])

plt.xticks(list(range(0,10)),Seasons,rotation='vertical')# rotation='vertical'
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.show()
```



```
[68]: plt.plot(Points[0],c="Black",ls='--',marker='s',ms=7,label=Players[0])
plt.plot(Points[1],c="Red",ls='-',marker='o',ms=7,label=Players[1])
plt.plot(Points[2],c="Green",ls='-.',marker='^',ms=7,label=Players[2])
plt.plot(Points[3],c="Blue",ls=':',marker='D',ms=7,label=Players[3])
plt.plot(Points[4],c="Magenta",ls='--',marker='s',ms=7,label=Players[4])
plt.plot(Points[5],c="Black",ls='-',marker='o',ms=7,label=Players[5])
plt.plot(Points[6],c="Red",ls='-.',marker='^',ms=7,label=Players[6])
plt.plot(Points[7],c="Green",ls=':',marker='D',ms=7,label=Players[7])
plt.plot(Points[8],c="Blue",ls='--',marker='s',ms=7,label=Players[8])
plt.plot(Points[9],c="Magenta",ls='-',marker='o',ms=7,label=Players[9])

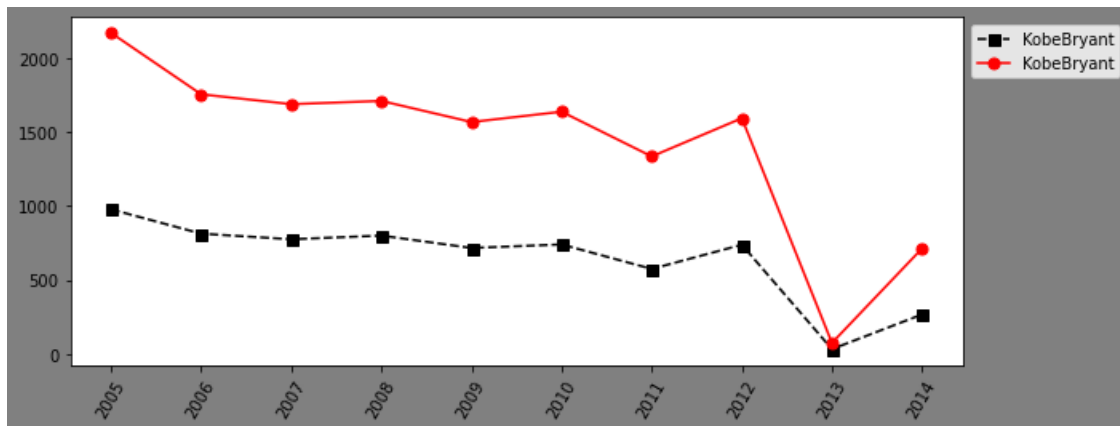
plt.xticks(list(range(0,10)),Seasons,rotation=60)# rotation='vertical'
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.show()
```



```
[69]: plt.plot(FieldGoals[0],c="Black",ls='--',marker='s',ms=7,label=Players[0])
plt.plot(FieldGoalAttempts[0],c="Red",ls='-',marker='o',ms=7,label=Players[0])
```



```
plt.xticks(list(range(0,10)),Seasons,rotation=60)# rotation='vertical'
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.show()
```

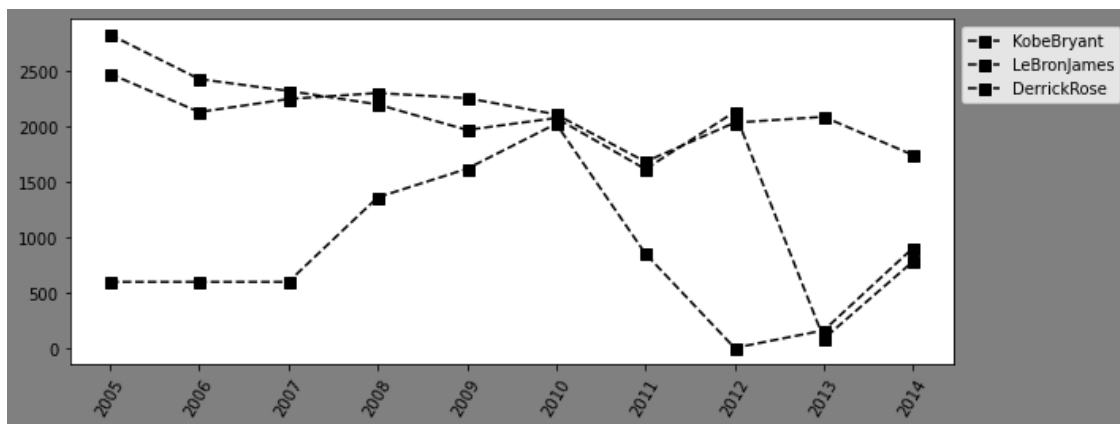


## 5 Functions Design

```
[70]: def myplot(PlayersList):
        for name in PlayersList:
            plt.
            ↪plot(Points[Pdict[name]],c="Black",ls='--',marker='s',ms=7,label=Players[Pdict[name]])

        plt.xticks(list(range(0,10)),Seasons,rotation=60)# rotation='vertical'
        plt.legend(loc='upper left',bbox_to_anchor=(1,1))
        plt.show()
```

```
[71]: myplot(["KobeBryant","LeBronJames","DerrickRose"])
```



## 6 Advanced Functions

```
[72]: def myplot(PlayersList):
    Col={"KobeBryant":"Black","JoeJohnson":"Red","LeBronJames":
    ↪ "Green","CarmeloAnthony":"Blue","DwightHoward":"Magenta",
        "ChrisBosh":"Black","ChrisPaul":"Red","KevinDurant":
    ↪ "Green","DerrickRose":"Blue","DwayneWade":"Magenta"}

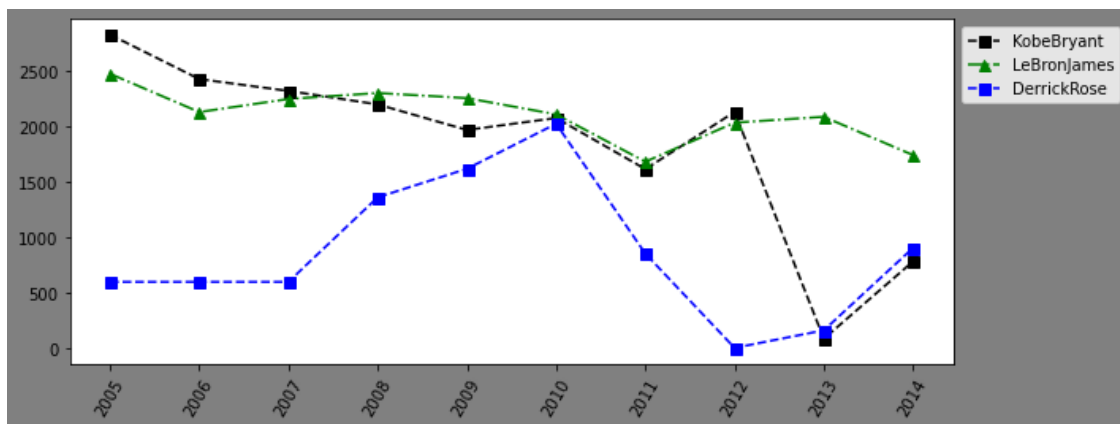
    Mkr={"KobeBryant":"s","JoeJohnson":"o","LeBronJames":"^","CarmeloAnthony":
    ↪ "D","DwightHoward":"s",
        "ChrisBosh":"o","ChrisPaul":"^","KevinDurant":"D","DerrickRose":
    ↪ "s","DwayneWade":"o"}

    Ls={"KobeBryant":'--',"JoeJohnson":'-',"LeBronJames":'-.', "CarmeloAnthony":
    ↪ ':',"DwightHoward":'--',
        "ChrisBosh":'-',"ChrisPaul":'-.', "KevinDurant":':',"DerrickRose":
    ↪ '--',"DwayneWade":'-'}

    for name in PlayersList:
        plt.
    ↪ plot(Points[Pdict[name]],c=Col[name],ls=Ls[name],marker=Mkr[name],ms=7,label=Players[Pdict[

    plt.xticks(list(range(0,10)),Seasons,rotation=60)# rotation='vertical'
    plt.legend(loc='upper left',bbox_to_anchor=(1,1))
    plt.show()
```

```
[73]: myplot(["KobeBryant","LeBronJames","DerrickRose"])
```



## 7 Fix up the Input

```
[74]: def myplot(data, PlayersList=Players):
    Col={"KobeBryant":"Black","JoeJohnson":"Red","LeBronJames":
    ↪"Green","CarmeloAnthony":"Blue","DwightHoward":"Magenta",
        "ChrisBosh":"Black","ChrisPaul":"Red","KevinDurant":
    ↪"Green","DerrickRose":"Blue","DwayneWade":"Magenta"}

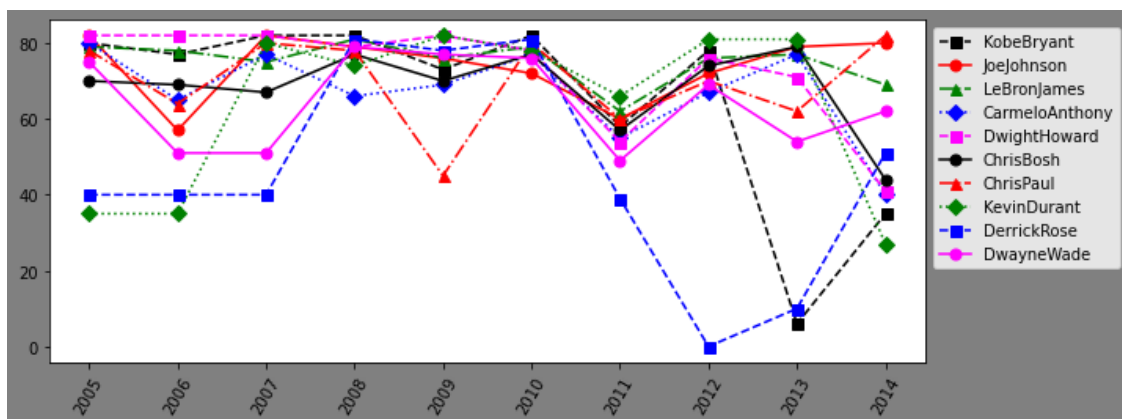
    Mkr={"KobeBryant":"s","JoeJohnson":"o","LeBronJames":"^","CarmeloAnthony":
    ↪"D","DwightHoward":"s",
        "ChrisBosh":"o","ChrisPaul":"^","KevinDurant":"D","DerrickRose":
    ↪"s","DwayneWade":"o"}

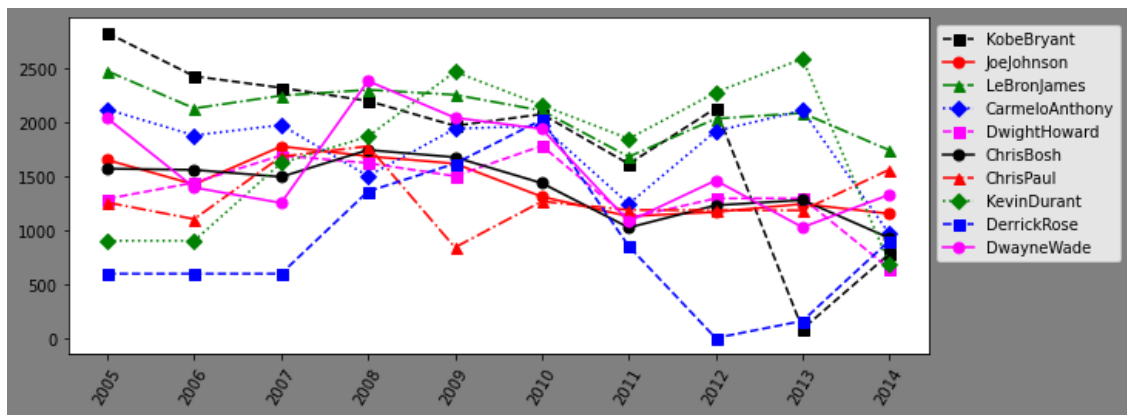
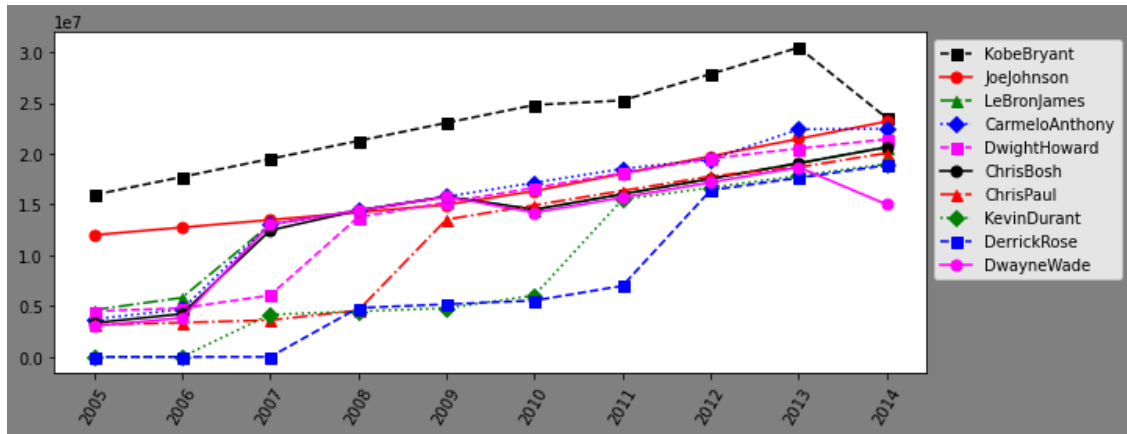
    Ls={"KobeBryant":'--',"JoeJohnson":'-',"LeBronJames":'-.', "CarmeloAnthony":
    ↪'::', "DwightHoward":'--',
        "ChrisBosh":'-',"ChrisPaul":'-.', "KevinDurant":'::', "DerrickRose":
    ↪'--',"DwayneWade":'-'}

    for name in PlayersList:
        plt.
    ↪plot(data[Pdict[name]],c=Col[name],ls=Ls[name],marker=Mkr[name],ms=7,label=Players[Pdict[name]])

    plt.xticks(list(range(0,10)),Seasons,rotation=60)# rotation='vertical'
    plt.legend(loc='upper left',bbox_to_anchor=(1,1))
    plt.show()
```

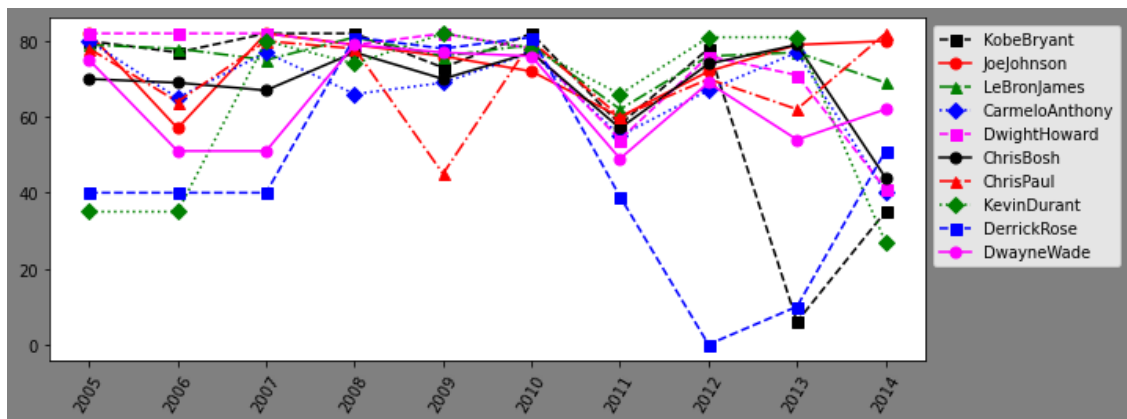
```
[75]: myplot(Games)
myplot(Salary)
myplot(Points)
```





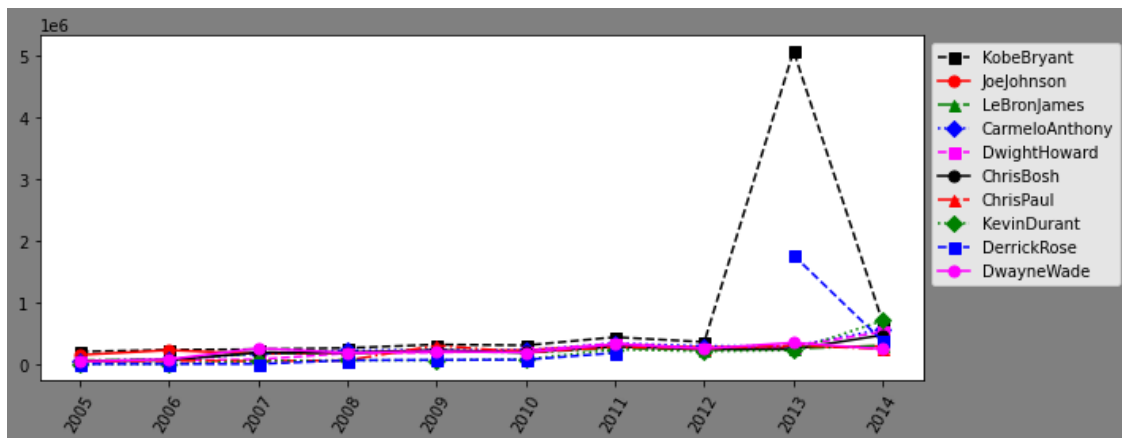
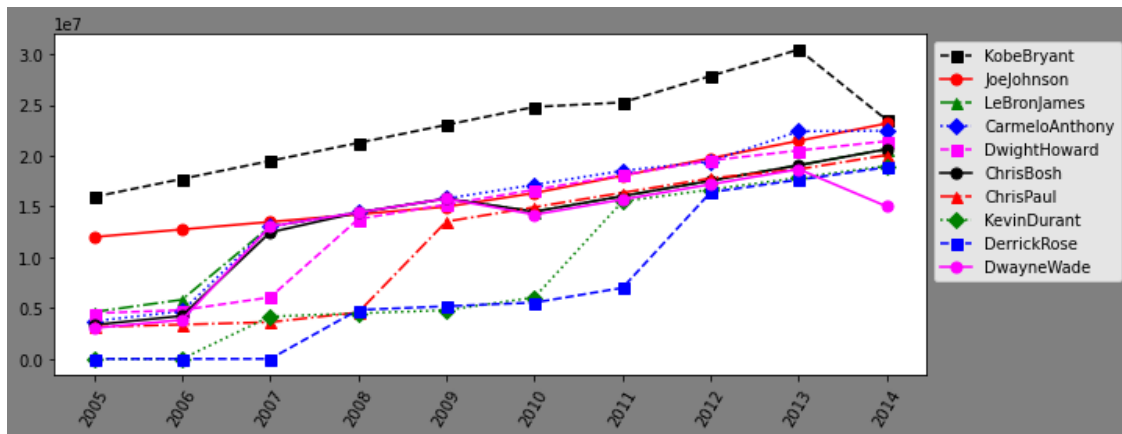
## 8 Insights

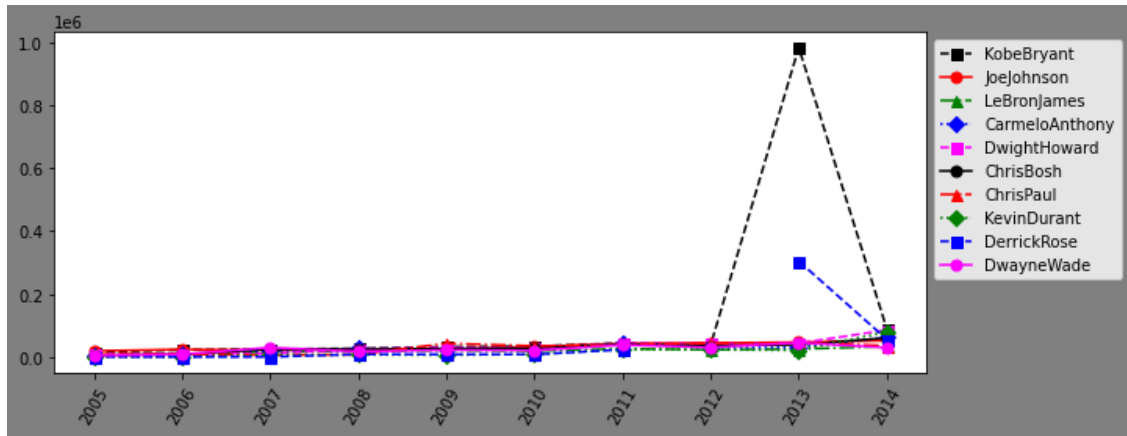
[76]: `myplot(Games)`



## 9 Salary

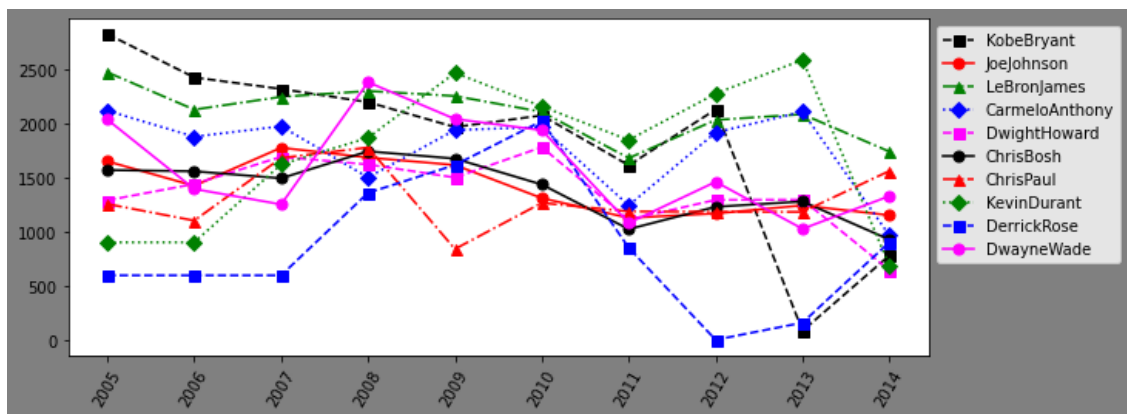
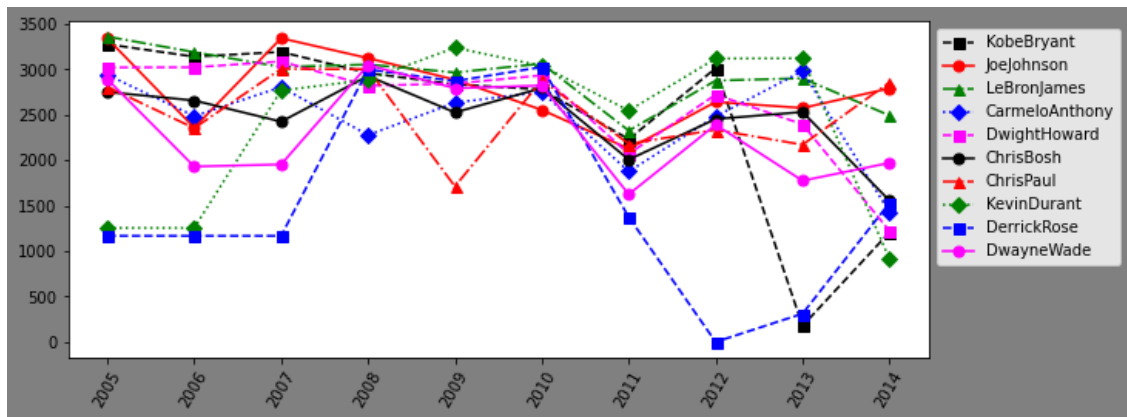
```
[77]: myplot(Salary)
      myplot(Salary/Games)
      myplot(Salary/FieldGoals)
```





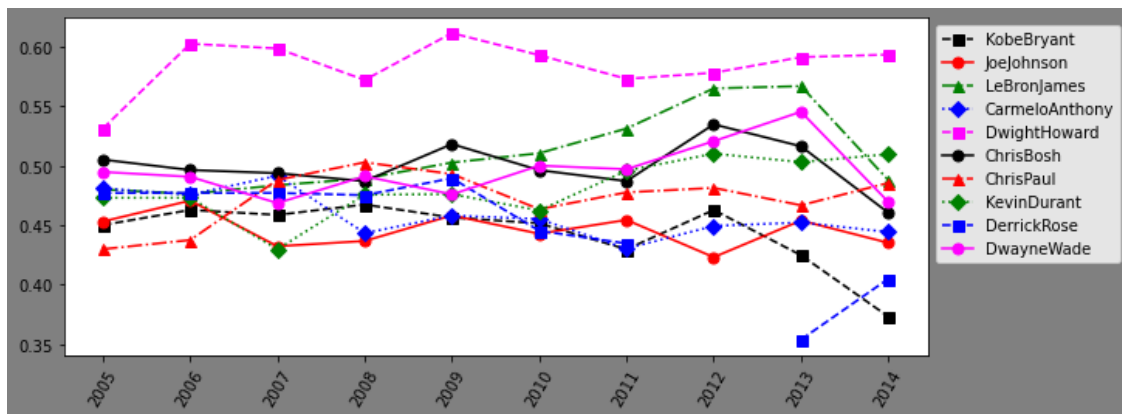
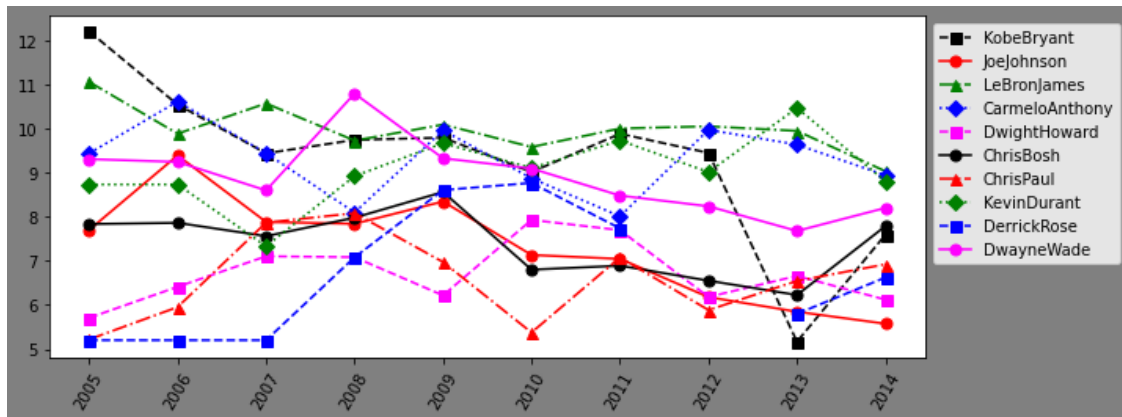
## 10 In Games Matrices

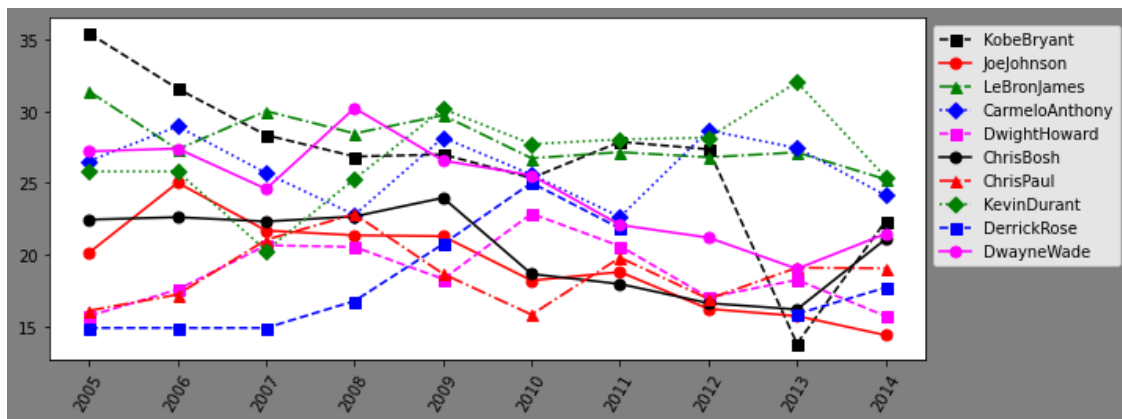
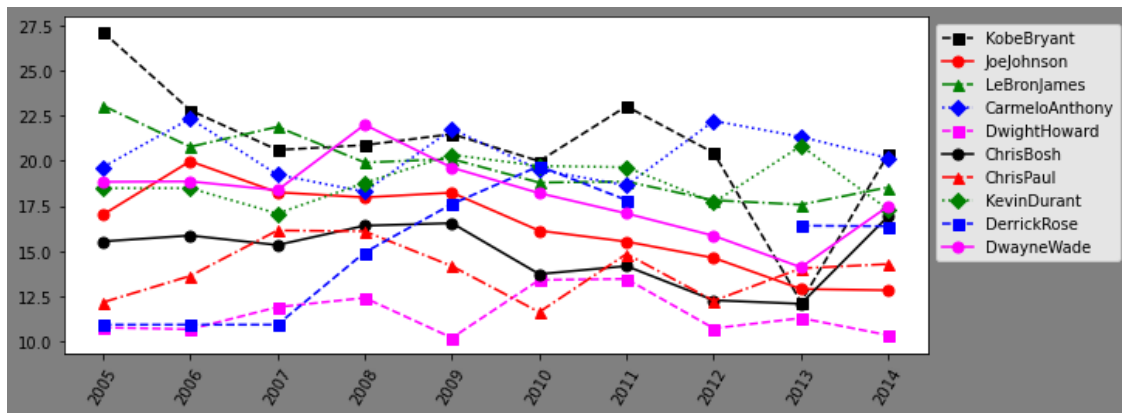
```
[78]: myplot(MinutesPlayed)
      myplot(Points)
```



## 11 In Games Matrices Normalized

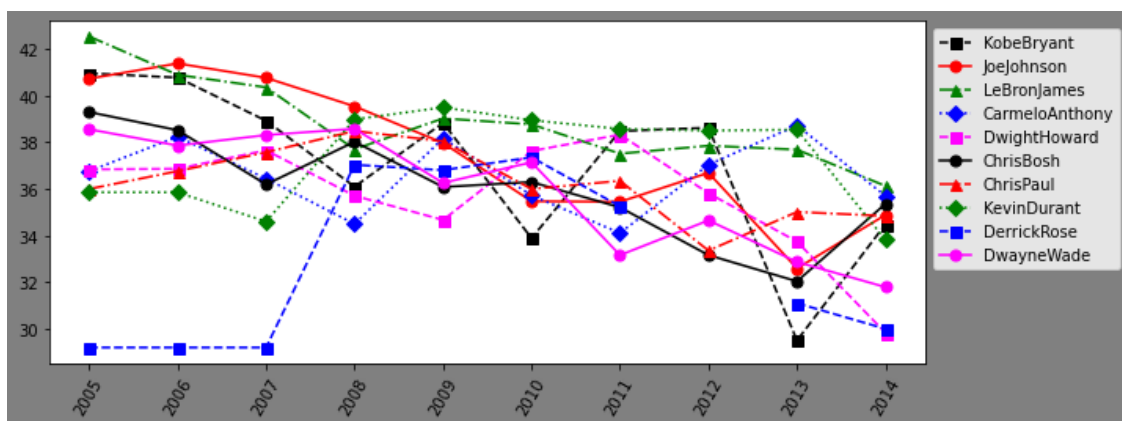
```
[79]: myplot(FieldGoals/Games)
      myplot(FieldGoals/FieldGoalAttempts)
      myplot(FieldGoalAttempts/Games)
      myplot(Points/Games)
```



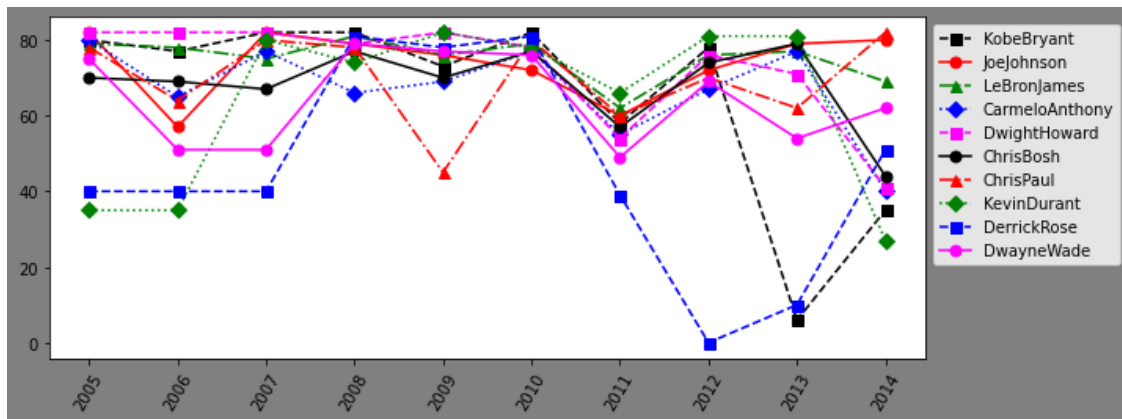


## 12 Interesting Observations

```
[80]: myplot(MinutesPlayed/Games)
      myplot(Games)
```

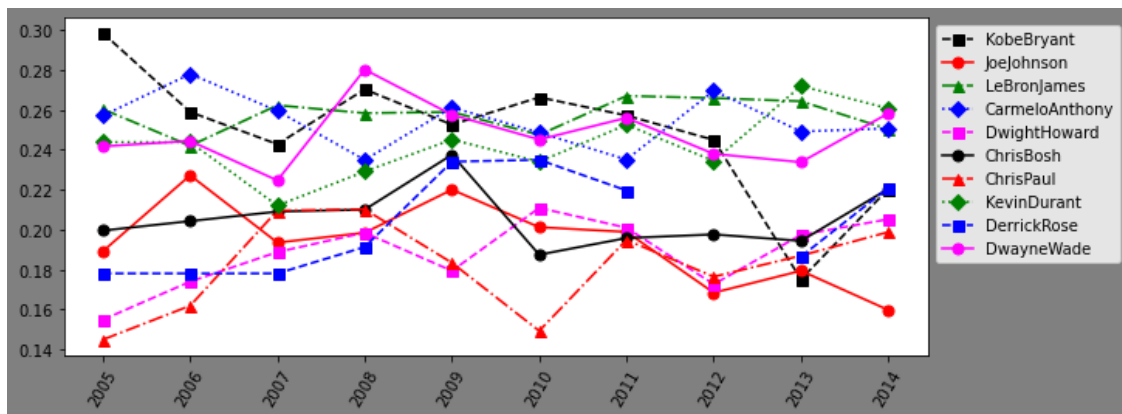






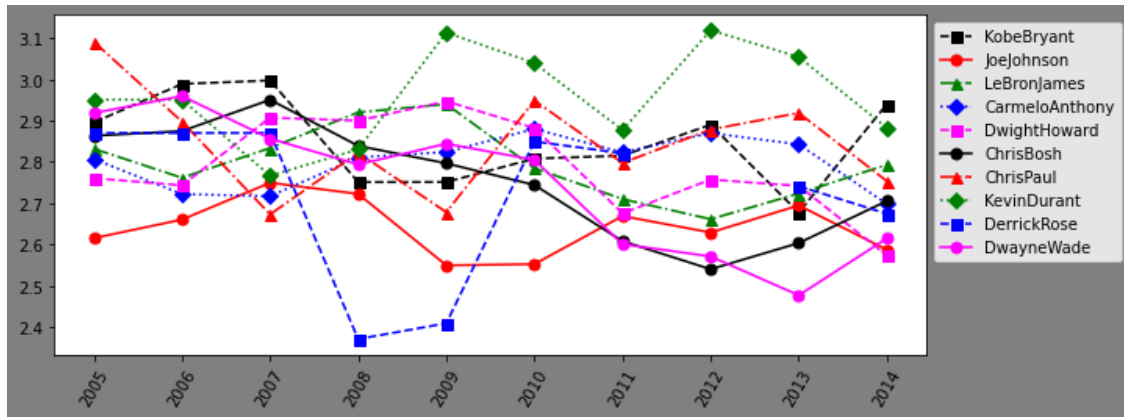
### 13 Time is Valuable

```
[81]: myplot(FieldGoals/MinutesPlayed)
```



### 14 Players Style

```
[82]: myplot(Points/FieldGoals)
```



[ ]:

[ ]: