In [10]:
```
#to create a matrix #L11 and L12
```

In [11]:
```
import numpy as np
np.reshape(*,*,'C') #like C language. first row then column
np.reshape(*,*,'F') #like R or Fortran language. first column then row
```

```
  File "<ipython-input-11-c28b659a4de8>", line 2
    np.reshape(*,*,'C') #like C language. first row then column
                ^
SyntaxError: invalid syntax
```

In [12]:
```
np.array() #it combines data elementsinto matrix
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-12-48ae2534db40> in <module>()
----> 1 np.array() #it combines data elementsinto matrix

NameError: name 'np' is not defined
```

In [13]:
```
#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
# - Pdict
#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 Programm
#           by Kirill Eremenko, you may do so by referencing www.superdatascience

#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,

#Players
Players = ["KobeBryant","JoeJohnson","LeBronJames","CarmeloAnthony","DwightHoward
Pdict = {"KobeBryant":0,"JoeJohnson":1,"LeBronJames":2,"CarmeloAnthony":3,"Dwight
```

```
#Salaries
KobeBryant_Salary = [15946875,17718750,19490625,21262500,23034375,24806250,252444
JoeJohnson_Salary = [12000000,12744189,13488377,14232567,14976754,16324500,180385
LeBronJames_Salary = [4621800,5828090,13041250,14410581,15779912,14500000,160225
CarmeloAnthony_Salary = [3713640,4694041,13041250,14410581,15779912,17149243,1851
DwightHoward_Salary = [4493160,4806720,6061274,13758000,15202590,16647180,180917
ChrisBosh_Salary = [3348000,4235220,12455000,14410581,15779912,14500000,16022500,
ChrisPaul_Salary = [3144240,3380160,3615960,4574189,13520500,14940153,16359805,17
KevinDurant_Salary = [0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832
DerrickRose_Salary = [0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,188
DwayneWade_Salary = [3031920,3841443,13041250,14410581,15779912,14200000,15691000
#Matrix
Salary = np.array([KobeBryant_Salary, JoeJohnson_Salary, LeBronJames_Salary, Carm

#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, Dw

#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, CarmeloAn

#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, CarmeloAnth
```

```
#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, Ca

#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, CarmeloAnthon
```

In [14]:
```
Salary
```

Out[14]:
```
array([[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]])
```

In [15]:
```
mydata=np.arange(0,20)
```

In [16]:
```
mydata
```

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

In [17]:
```
np.reshape(mydata,(5,4)) #parameters are passed in the form of tuples. First row
```

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

In [18]:
```
y=np.reshape(mydata,(5,4),order='F')
```

In [19]:
```
x=np.reshape(mydata,(5,4),order='C')
```

In [20]:
```
x[2,2]
```

Out[20]:
```
10
```

In [21]:
```
x[3][2]
```

Out[21]:
```
14
```

In [22]:
```
y[3,1]
```

Out[22]:
```
8
```

In [23]:
```
type(mydata)
```

Out[23]:
```
numpy.ndarray
```

In [24]:
```
#since my data is an array and in OOPS concept it is a object
#HEnce instead of using np.reshape we can also use - mydata.reshape(dimension)
```

In [25]:
```
mydata.reshape((10,2))
```

Out[25]:
```
array([[ 0,  1],
       [ 2,  3],
       [ 4,  5],
       [ 6,  7],
       [ 8,  9],
       [10, 11],
       [12, 13],
       [14, 15],
       [16, 17],
       [18, 19]])
```

In [28]:
```
r1=[1,2,3,34,5]
r2=["jejrg","fuwrg","jfb"]
r3=[1,2,3,"fgg"]
```

In [29]:
```
l=[r1,r2,r3]
```

In [30]:
```
l[2][2]
```

Out[30]:
```
3
```

In [31]:
```
l
```

Out[31]:
```
[[1, 2, 3, 34, 5], ['jejrg', 'fuwrg', 'jfb'], [1, 2, 3, 'fgg']]
```

In [32]:
```
l[2][1]
```

Out[32]:
```
2
```

In [33]:
```
p=np.array([r1,r2,r3])
```

In [34]:
```
p
```

Out[34]:
```
array([list([1, 2, 3, 34, 5]), list(['jejrg', 'fuwrg', 'jfb']),
       list([1, 2, 3, 'fgg'])], dtype=object)
```

In [35]:
```
print(p)
```

```
[list([1, 2, 3, 34, 5]) list(['jejrg', 'fuwrg', 'jfb'])
 list([1, 2, 3, 'fgg'])]
```

In [64]:
```
#Dictionaries start here #L13
```

Games

In [65]:
```
Games
```

Out[65]:
```
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]])
```

In [66]:
```
Games[2]
```

Out[66]:
```
array([79, 78, 75, 81, 76, 79, 62, 76, 77, 69])
```

In [67]:
```
Games[2,-1]
```

Out[67]:
```
69
```

In [68]:
```
Points[6]
```

Out[68]:
```
array([1258, 1104, 1684, 1781,  841, 1268, 1189, 1186, 1185, 1564])
```

In [69]:
```
Points[6,1]
```

Out[69]:
```
1104
```

In [70]:
```
dict1={'key1':'b=valu1','key2':'value2'} #key is like a label for each value
```

In [71]:
```
dict1
```

Out[71]:
```
{'key1': 'b=valu1', 'key2': 'value2'}
```

In [72]:
```
dict1.keys()
```

Out[72]:
```
dict_keys(['key1', 'key2'])
```

In [73]:
```
dict1.values()
```

Out[73]:
```
dict_values(['b=valu1', 'value2'])
```

In [74]:
```
dict1.itername()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-74-dde26b72f6d5> in <module>()
----> 1 dict1.itername()

AttributeError: 'dict' object has no attribute 'itername'
```

In [75]:
```
dict1['key2'] #accessing through keys
```

Out[75]:
```
'value2'
```

In [76]:
```
dict2={'germany':'sgvy','france':2,'spain':True}
```

In [77]:
```
dict2 #order doesnt matter cause accessing is done through keys
```

Out[77]:
```
{'france': 2, 'germany': 'sgvy', 'spain': True}
```

In [78]:
```
Pdict
```

Out[78]:
```
{'CarmeloAnthony': 3,
 'ChrisBosh': 5,
 'ChrisPaul': 6,
 'DerrickRose': 8,
 'DwayneWade': 9,
 'DwightHoward': 4,
 'JoeJohnson': 1,
 'KevinDurant': 7,
 'KobeBryant': 0,
 'LeBronJames': 2}
```

In [79]:
```
Games
```

Out[79]:
```
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]])
```

In [80]:
```
Pdict['KobeBryant']
```

Out[80]:
```
0
```

In [81]:
```
Games[Pdict['KobeBryant']] #Use of dictionary to fing data related to a player by
```

Out[81]:
```
array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

In [82]:
```
Games[Pdict['KobeBryant']][Sdict['2012']]
```

Out[82]:
```
78
```

In [83]:
```
#Matrix Operations starts here #L14
```

In [84]:
```
Salary
```

Out[84]:
```
array([[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]])
```

In [98]:
```
Salary
```

Out[98]:
```
array([[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]])
```

In [99]:
```
FieldGoals
```

Out[99]:
```
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]])
```

In [100]:
```
Games
```

Out[100]:
```
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]])
```

In [101]:
```
import warnings
warnings.filterwarnings('ignore')
c=np.matrix.round(FieldGoals/Games)
```

In [102]:
```
c
```

Out[102]:
```
array([[ 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.]])
```

In [103]:
```
c[Pdict['KobeBryant']][Sdict['2012']]
```

Out[103]:
```
9.0
```

In [104]:
```
x=np.matrix.round(MinutesPlayed/Games)
```

In [105]:
```
x
```

Out[105]:
```
array([[ 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.]])
```

In [112]:
```
z=np.matrix.round(FieldGoals/FieldGoalAttempts,2)*100
```

In [113]:
```
z
```

Out[113]:
```
array([[ 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.]])
```

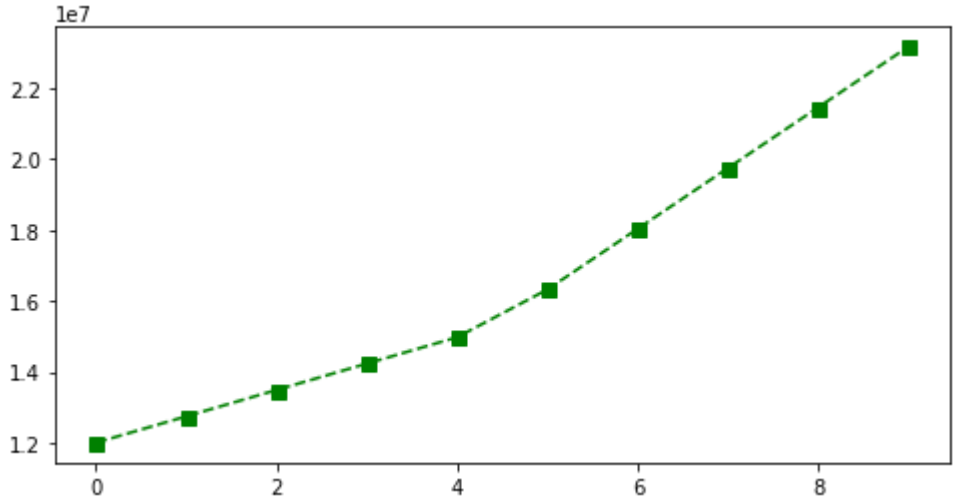
In [114]:
```
#Visualization #L15
```

In [115]:
```
import numpy as np
```

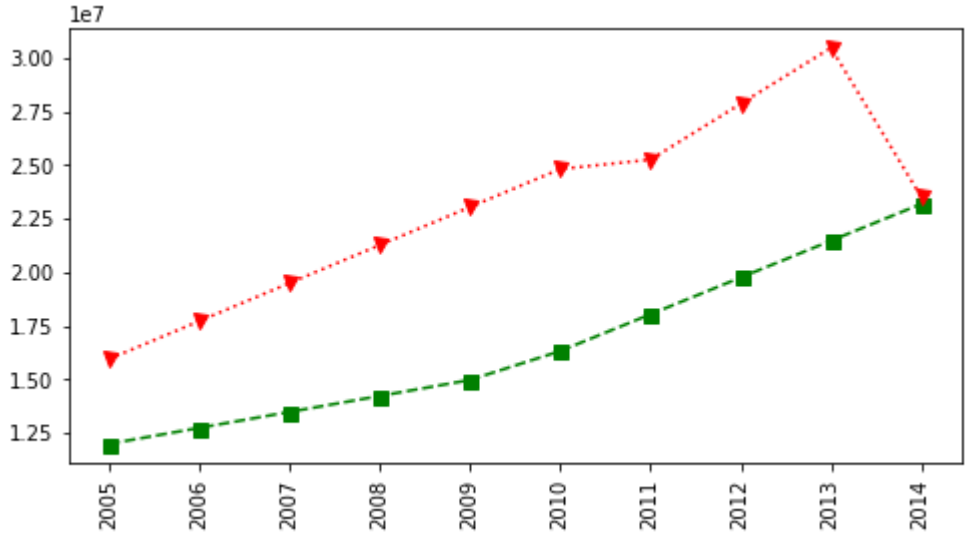
In [116]:
```
import matplotlib.pyplot as plt
```

In [132]:
```
%matplotlib inline
plt.rcParams['figure.figsize']=8,4
```

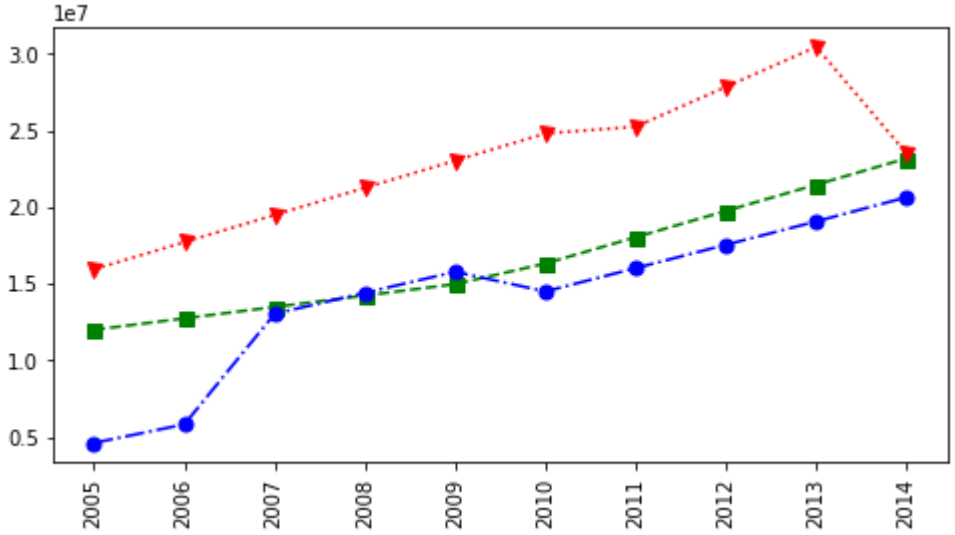
In [133]:
```
plt.plot(Salary[1],c='Green',ls='--',marker='s',ms=7)
plt.show() #to display plots
```



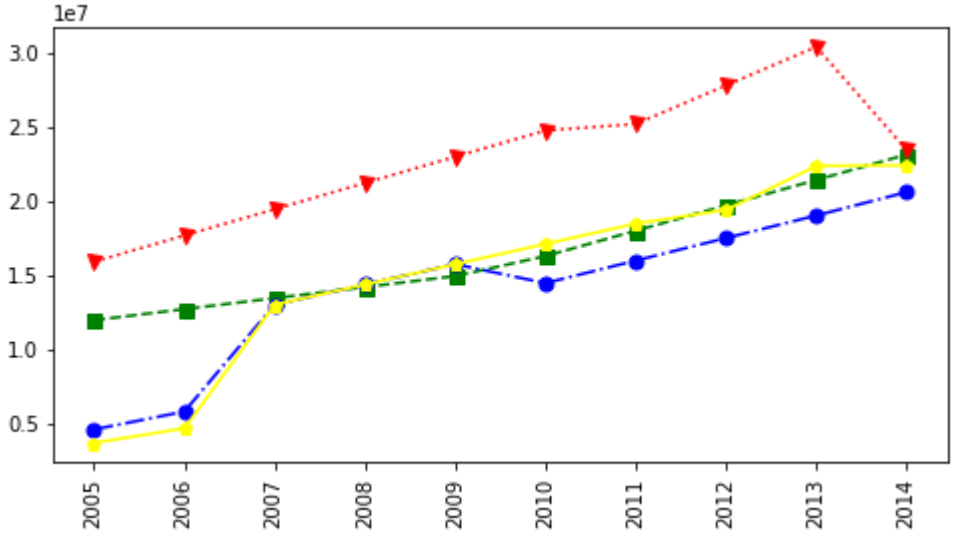
In [145]:
```
plt.plot(Salary[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(Salary[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



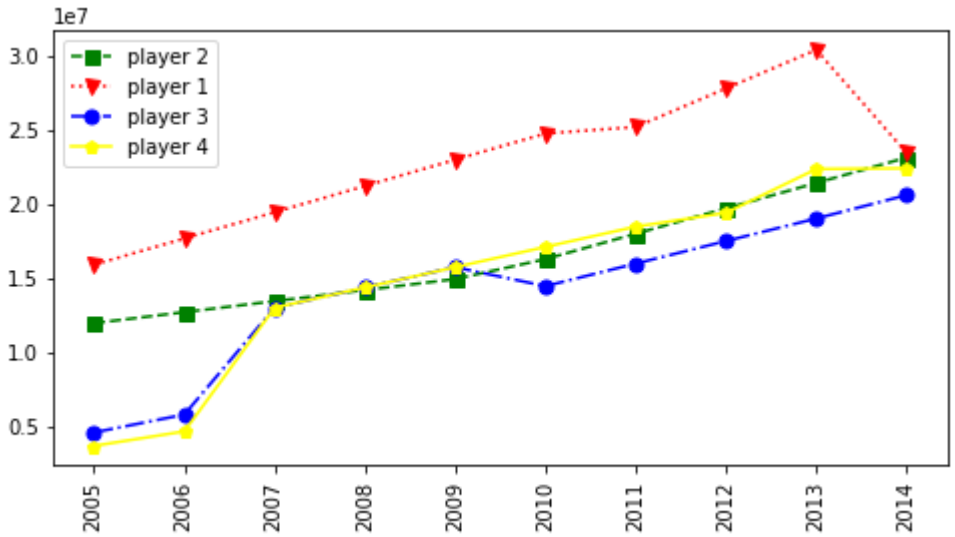
In [146]:
```
plt.plot(Salary[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(Salary[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(Salary[2],c='Blue',ls='-.',marker='o',ms=7,label="player 3")
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



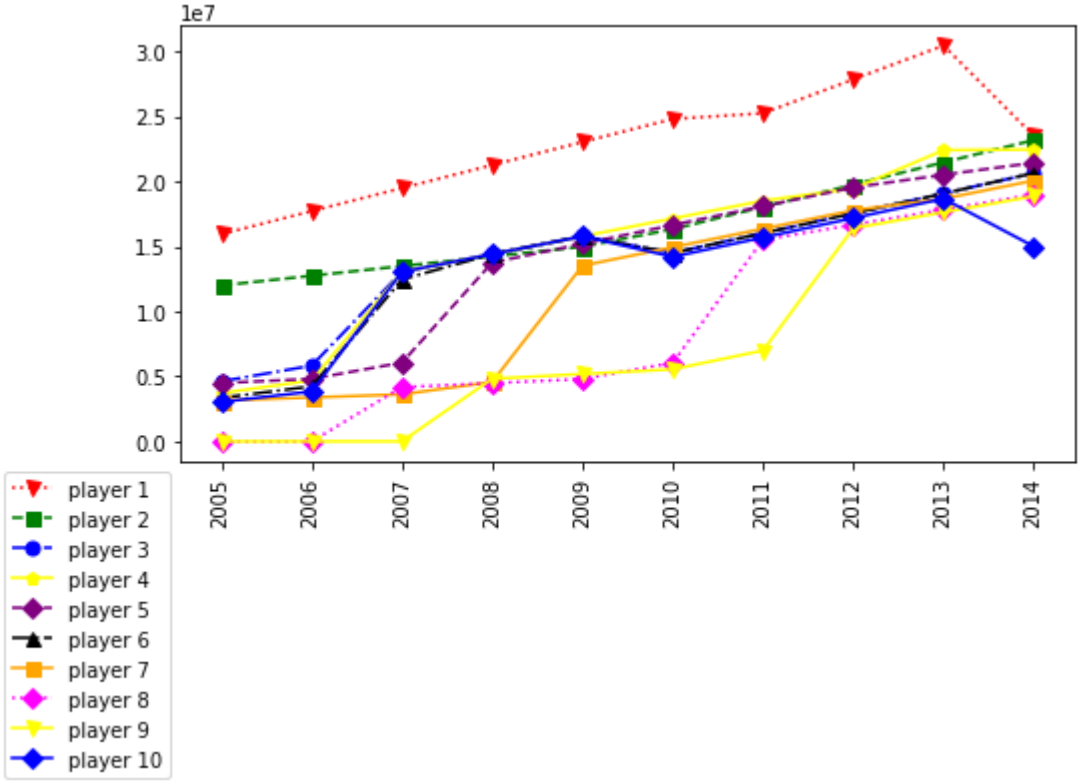
In [147]:
```
plt.plot(Salary[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(Salary[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(Salary[2],c='Blue',ls='-.',marker='o',ms=7,label="player 3")
plt.plot(Salary[3],c='Yellow',ls='-',marker='p',ms=7,label="player 4")
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



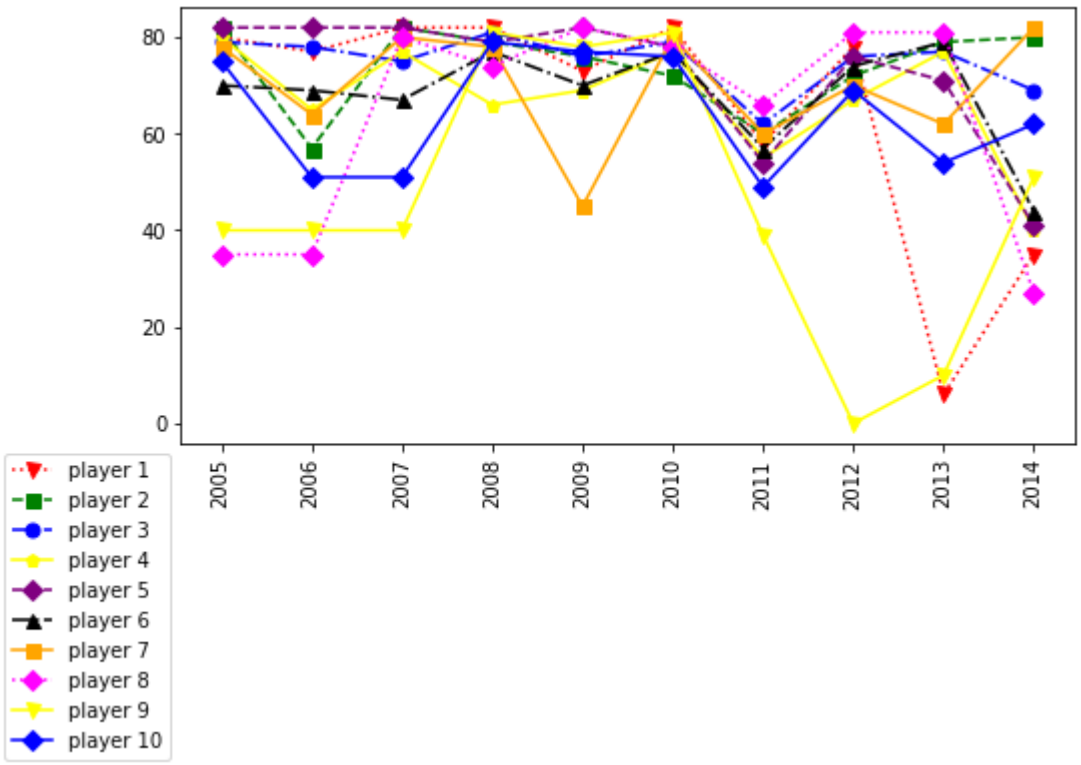
In [148]:
```
plt.plot(Salary[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(Salary[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(Salary[2],c='Blue',ls='-.',marker='o',ms=7,label="player 3")
plt.plot(Salary[3],c='Yellow',ls='-',marker='p',ms=7,label="player 4")
plt.legend()
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



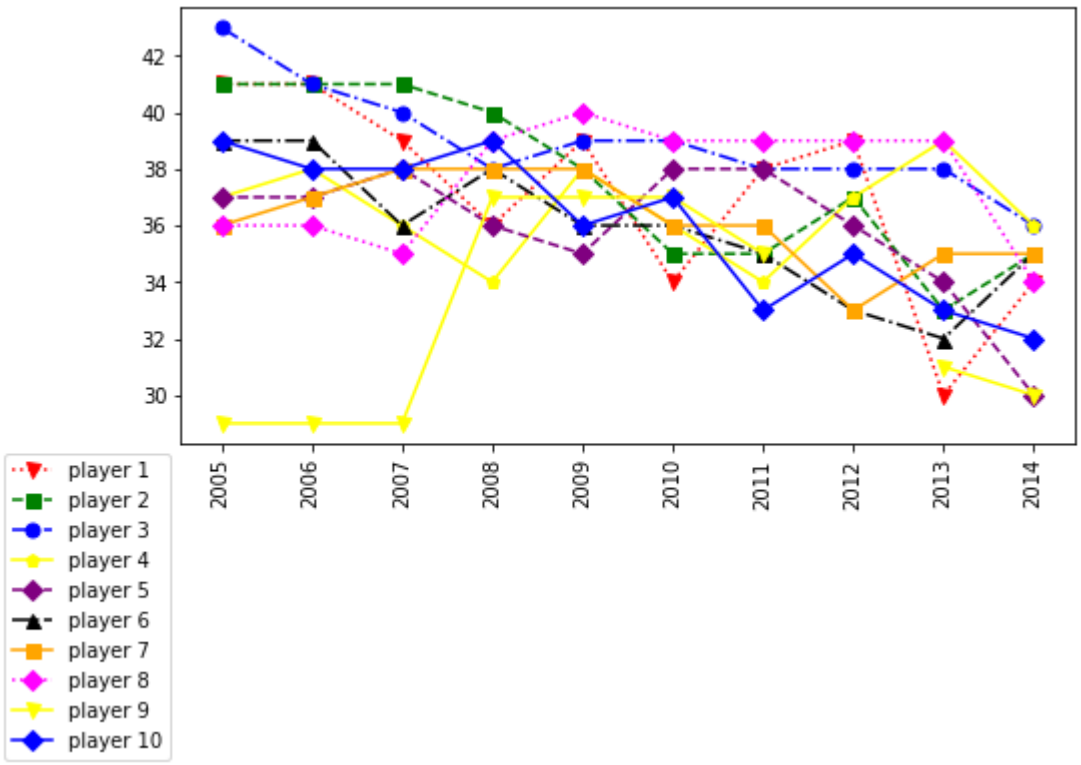
In [163]:
```
#Adding Legends
plt.plot(Salary[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(Salary[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(Salary[2],c='Blue',ls='-.',marker='o',ms=7,label="player 3")
plt.plot(Salary[3],c='Yellow',ls='-',marker='p',ms=7,label="player 4")
plt.plot(Salary[4],c='Purple',ls='--',marker='D',ms=7,label="player 5")
plt.plot(Salary[5],c='Black',ls='-.',marker='^',ms=7,label="player 6")
plt.plot(Salary[6],c='Orange',ls='-',marker='s',ms=7,label="player 7")
plt.plot(Salary[7],c='Magenta',ls=':',marker='D',ms=7,label="player 8")
plt.plot(Salary[8],c='Yellow',ls='-',marker='v',ms=7,label="player 9")
plt.plot(Salary[9],c='Blue',ls='-',marker='D',ms=7,label="player 10")
plt.legend(loc='bottom right', bbox_to_anchor=(0,0)) #location of plot to the box
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



In [164]:
```
#Adding Legends
plt.plot(Games[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(Games[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(Games[2],c='Blue',ls='-.',marker='o',ms=7,label="player 3")
plt.plot(Games[3],c='Yellow',ls='-',marker='p',ms=7,label="player 4")
plt.plot(Games[4],c='Purple',ls='--',marker='D',ms=7,label="player 5")
plt.plot(Games[5],c='Black',ls='-.',marker='^',ms=7,label="player 6")
plt.plot(Games[6],c='Orange',ls='-',marker='s',ms=7,label="player 7")
plt.plot(Games[7],c='Magenta',ls=':',marker='D',ms=7,label="player 8")
plt.plot(Games[8],c='Yellow',ls='-',marker='v',ms=7,label="player 9")
plt.plot(Games[9],c='Blue',ls='-',marker='D',ms=7,label="player 10")
plt.legend(loc='bottom right', bbox_to_anchor=(0,0)) #location of plot to the box
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```

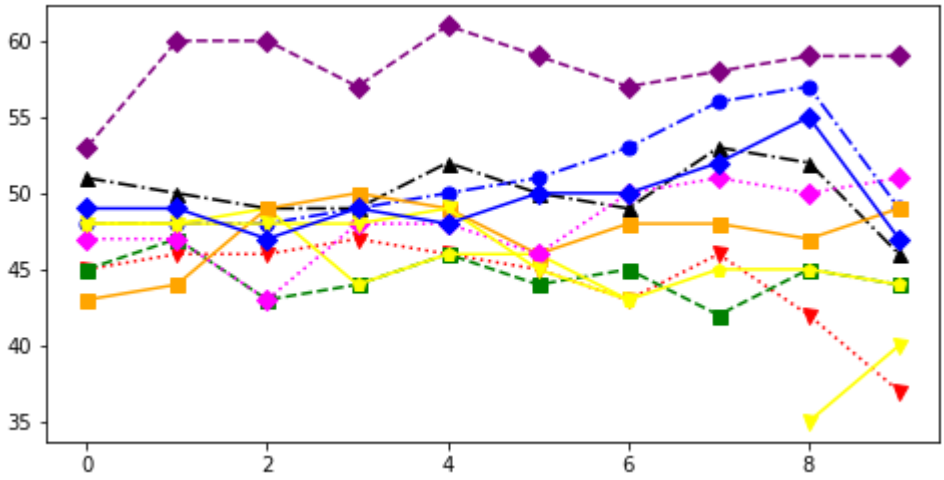


In [165]:
```
#Adding Legends
plt.plot(x[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(x[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(x[2],c='Blue',ls='-.',marker='o',ms=7,label="player 3")
plt.plot(x[3],c='Yellow',ls='-',marker='p',ms=7,label="player 4")
plt.plot(x[4],c='Purple',ls='--',marker='D',ms=7,label="player 5")
plt.plot(x[5],c='Black',ls='-.',marker='^',ms=7,label="player 6")
plt.plot(x[6],c='Orange',ls='-',marker='s',ms=7,label="player 7")
plt.plot(x[7],c='Magenta',ls=':',marker='D',ms=7,label="player 8")
plt.plot(x[8],c='Yellow',ls='-',marker='v',ms=7,label="player 9")
plt.plot(x[9],c='Blue',ls='-',marker='D',ms=7,label="player 10")
plt.legend(loc='bottom right', bbox_to_anchor=(0,0)) #location of plot to the box
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



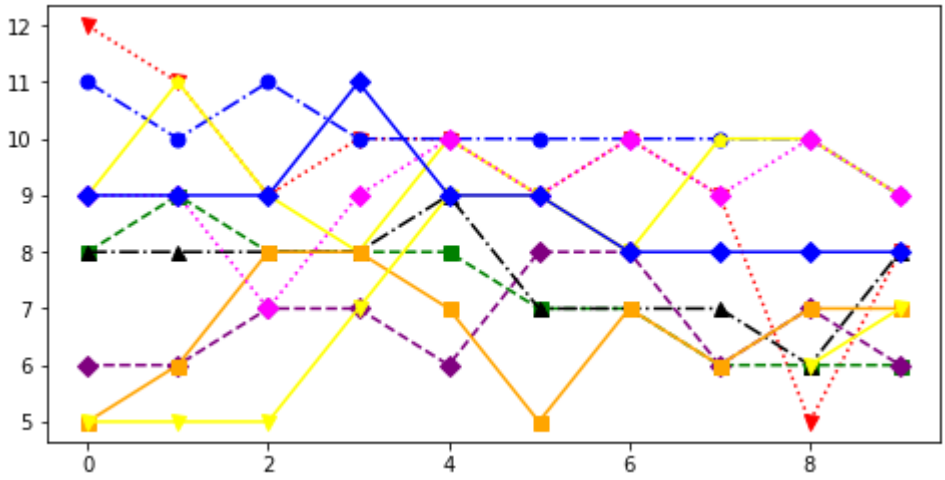
In [166]:
```
#Adding Legends
plt.plot(z[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(z[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(z[2],c='Blue',ls='-.',marker='o',ms=7,label="player 3")
plt.plot(z[3],c='Yellow',ls='-',marker='p',ms=7,label="player 4")
plt.plot(z[4],c='Purple',ls='--',marker='D',ms=7,label="player 5")
plt.plot(z[5],c='Black',ls='-.',marker='^',ms=7,label="player 6")
plt.plot(z[6],c='Orange',ls='-',marker='s',ms=7,label="player 7")
plt.plot(z[7],c='Magenta',ls=':',marker='D',ms=7,label="player 8")
plt.plot(z[8],c='Yellow',ls='-',marker='v',ms=7,label="player 9")
plt.plot(z[9],c='Blue',ls='-',marker='D',ms=7,label="player 10")
plt.legend(loc='bottom right', bboz_to_anchor=(0,0)) #location of plot to the boz
plt.zticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-166-34cdb5c41607> in <module>()
     10 plt.plot(z[8],c='Yellow',ls='-',marker='v',ms=7,label="player 9")
     11 plt.plot(z[9],c='Blue',ls='-',marker='D',ms=7,label="player 10")
---> 12 plt.legend(loc='bottom right', bboz_to_anchor=(0,0)) #location of plot
 to the boz and pass as tuple
     13 plt.zticks(list(range(0,10)),Seasons,rotation='vertical')
     14 plt.show

G:\PYTHON\AnacondaPython\lib\site-packages\matplotlib\pyplot.py in legend(*arg
s, **kwargs)
   3721 @docstring.copy_dedent(Axes.legend)
   3722 def legend(*args, **kwargs):
-> 3723     ret = gca().legend(*args, **kwargs)
   3724     return ret
   3725

G:\PYTHON\AnacondaPython\lib\site-packages\matplotlib\axes\_axes.py in legend(s
elf, *args, **kwargs)
    563             raise TypeError('Invalid arguments to legend.')
    564
--> 565         self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
    566         self.legend_._remove_method = lambda h: setattr(self, 'legend_'
, None)
    567         return self.legend_

TypeError: __init__() got an unexpected keyword argument 'bboz_to_anchor'
```


60
55
50
45
40
35
0
2
4
6
8

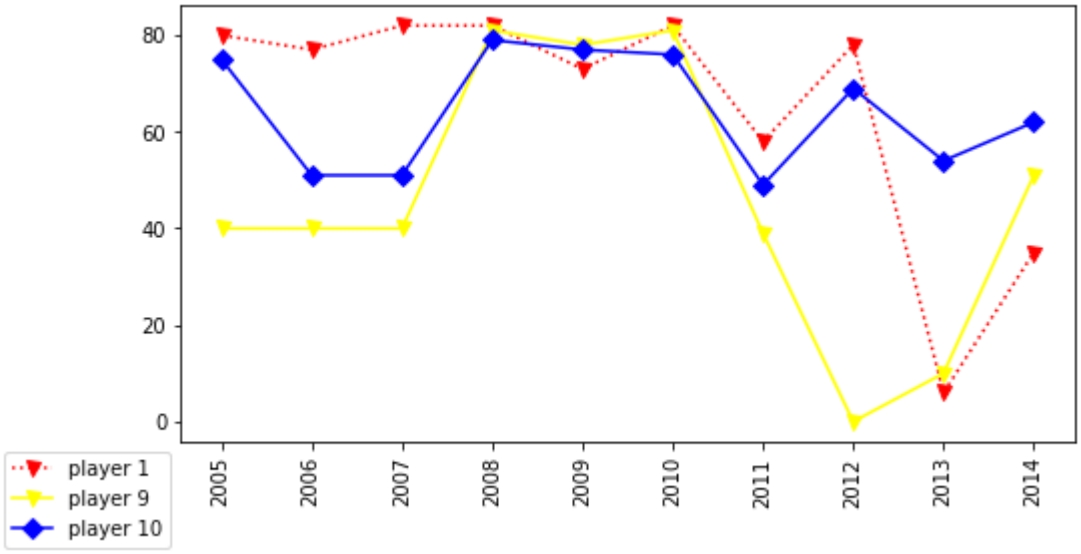
In [167]:
```
#Adding Legends
plt.plot(c[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(c[1],c='Green',ls='--',marker='s',ms=7,label="player 2")
plt.plot(c[2],c='Blue',ls='-.',marker='o',ms=7,label="player 3")
plt.plot(c[3],c='Yellow',ls='-',marker='p',ms=7,label="player 4")
plt.plot(c[4],c='Purple',ls='--',marker='D',ms=7,label="player 5")
plt.plot(c[5],c='Black',ls='-.',marker='^',ms=7,label="player 6")
plt.plot(c[6],c='Orange',ls='-',marker='s',ms=7,label="player 7")
plt.plot(c[7],c='Magenta',ls=':',marker='D',ms=7,label="player 8")
plt.plot(c[8],c='Yellow',ls='-',marker='v',ms=7,label="player 9")
plt.plot(c[9],c='Blue',ls='-',marker='D',ms=7,label="player 10")
plt.legend(loc='bottom right', bboc_to_anchor=(0,0)) #location of plot to the boc
plt.cticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-167-e451bdbce296> in <module>()
     10 plt.plot(c[8],c='Yellow',ls='-',marker='v',ms=7,label="player 9")
     11 plt.plot(c[9],c='Blue',ls='-',marker='D',ms=7,label="player 10")
---> 12 plt.legend(loc='bottom right', bboc_to_anchor=(0,0)) #location of plot
 to the boc and pass as tuple
     13 plt.cticks(list(range(0,10)),Seasons,rotation='vertical')
     14 plt.show

G:\PYTHON\AnacondaPython\lib\site-packages\matplotlib\pyplot.py in legend(*arg
s, **kwargs)
   3721 @docstring.copy_dedent(Axes.legend)
   3722 def legend(*args, **kwargs):
-> 3723     ret = gca().legend(*args, **kwargs)
   3724     return ret
   3725

G:\PYTHON\AnacondaPython\lib\site-packages\matplotlib\axes\_axes.py in legend(s
elf, *args, **kwargs)
    563             raise TypeError('Invalid arguments to legend.')
    564
--> 565         self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
    566         self.legend_._remove_method = lambda h: setattr(self, 'legend_'
, None)
    567         return self.legend_

TypeError: __init__() got an unexpected keyword argument 'bboc_to_anchor'
```



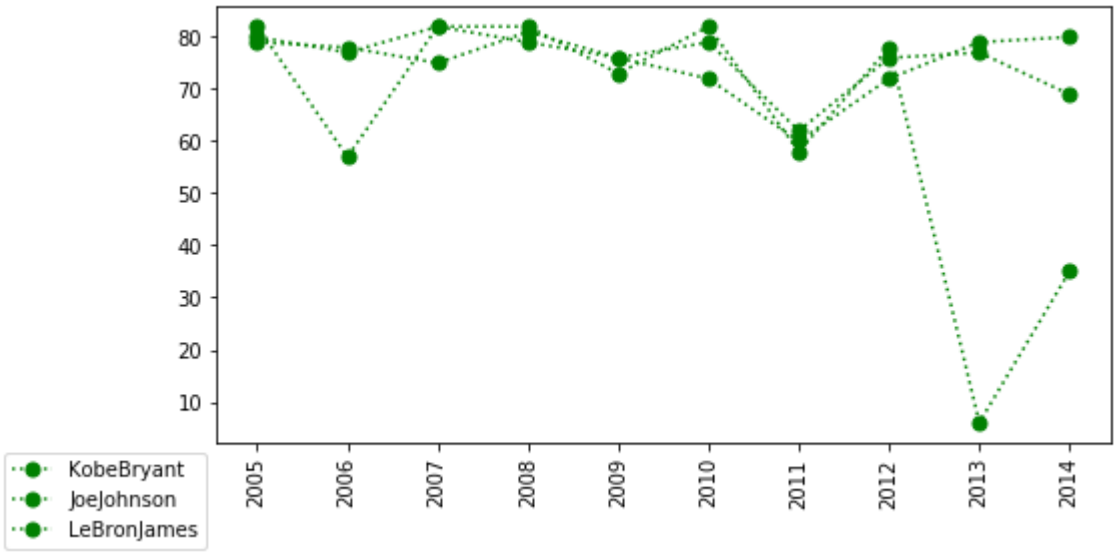
In [168]:
```
#Creating our own Function #L16
```

In [169]:
```
#Adding legends
plt.plot(Games[0],c='Red',ls=':',marker='v',ms=7,label="player 1")
plt.plot(Games[8],c='Yellow',ls='-',marker='v',ms=7,label="player 9")
plt.plot(Games[9],c='Blue',ls='-',marker='D',ms=7,label="player 10")
plt.legend(loc='bottom right', bbox_to_anchor=(0,0)) #location of plot to the box
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



In [178]:
```
def myplot(playerlist):
    for i in playerlist:
        plt.plot(Games[Pdict[i]],c='Green',ls=':',marker='o',ms=7,label=Players[P
    plt.legend(loc='bottom right', bbox_to_anchor=(0,0)) #location of plot to the
    plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
    plt.show()
```

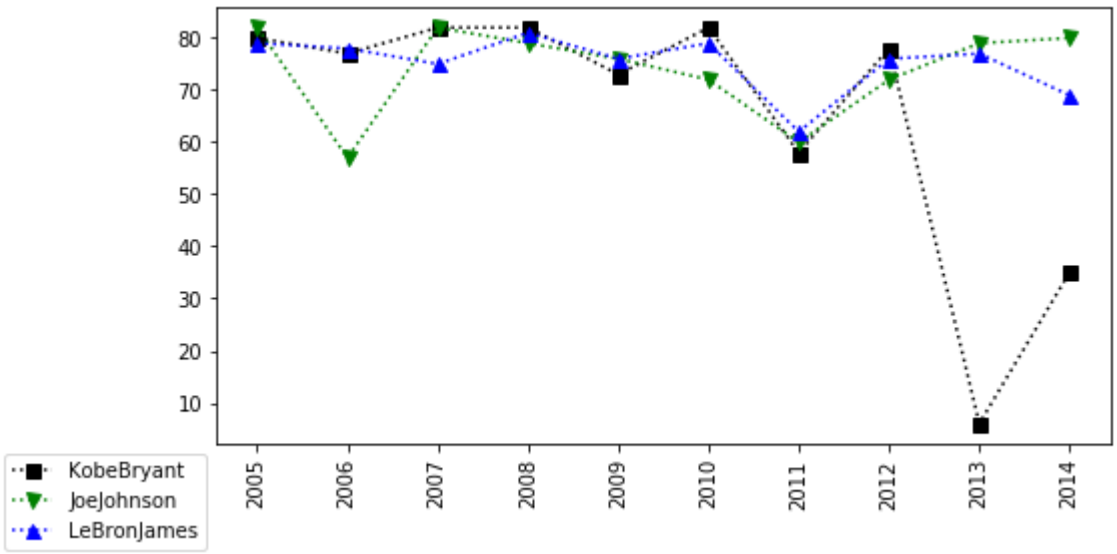
In [179]:
```
myplot(["KobeBryant","JoeJohnson","LeBronJames"])
```



In [180]:
```
#Advanced Function Design #L17
```

In [191]:
```
def myplot(playerlist):
    Col = {"KobeBryant":"Black","JoeJohnson":"Green","LeBronJames":"Blue","Carmel
    Mark = {"KobeBryant":"s","JoeJohnson":"v","LeBronJames":"^","CarmeloAnthony":
    for i in playerlist:
        plt.plot(Games[Pdict[i]],c=Col[i],ls=':',marker=Mark[i],ms=7,label=Player
    plt.legend(loc='bottom right', bbox_to_anchor=(0,0)) #Location of plot to the
    plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
    plt.show()
```

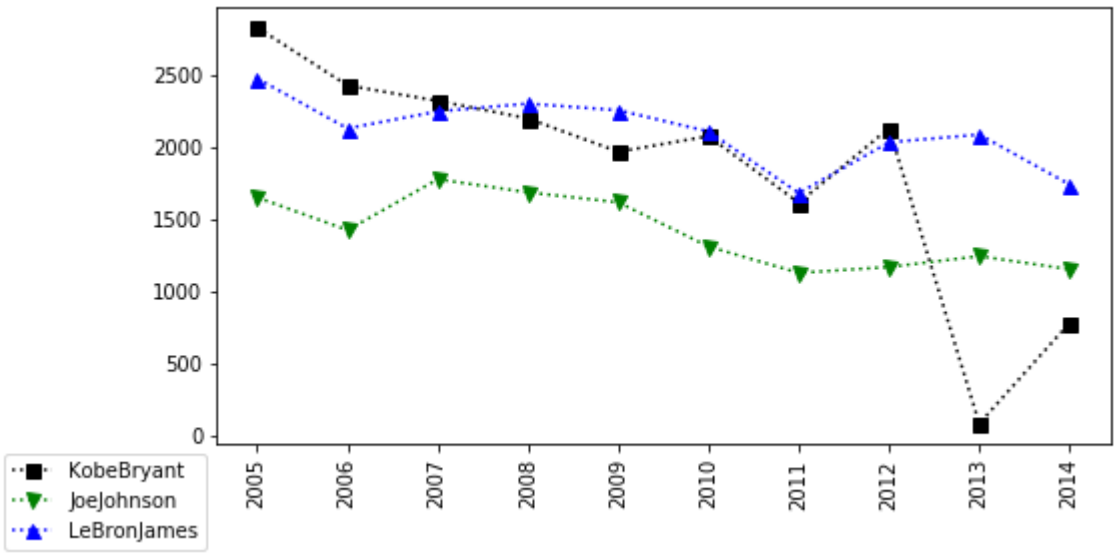
In [192]:
```
myplot(["KobeBryant","JoeJohnson","LeBronJames"])
```



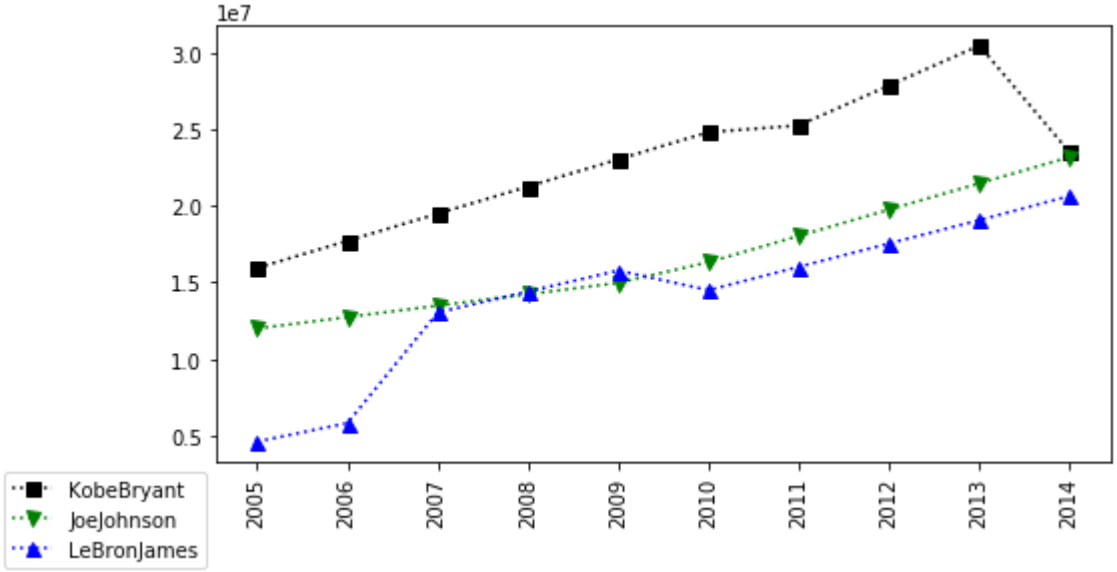
In [193]:
```
#Fix up the inputs
```

In [195]:
```
def myplot(data,playerlist):
    Col = {"KobeBryant":"Black","JoeJohnson":"Green","LeBronJames":"Blue","Carmel
    Mark = {"KobeBryant":"s","JoeJohnson":"v","LeBronJames":"^","CarmeloAnthony":
    for i in playerlist:
        plt.plot(data[Pdict[i]],c=Col[i],ls=':',marker=Mark[i],ms=7,label=Players
    plt.legend(loc='bottom right', bbox_to_anchor=(0,0)) #location of plot to the
    plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
    plt.show()
```

In [196]:
```
myplot(Points,["KobeBryant","JoeJohnson","LeBronJames"])
```

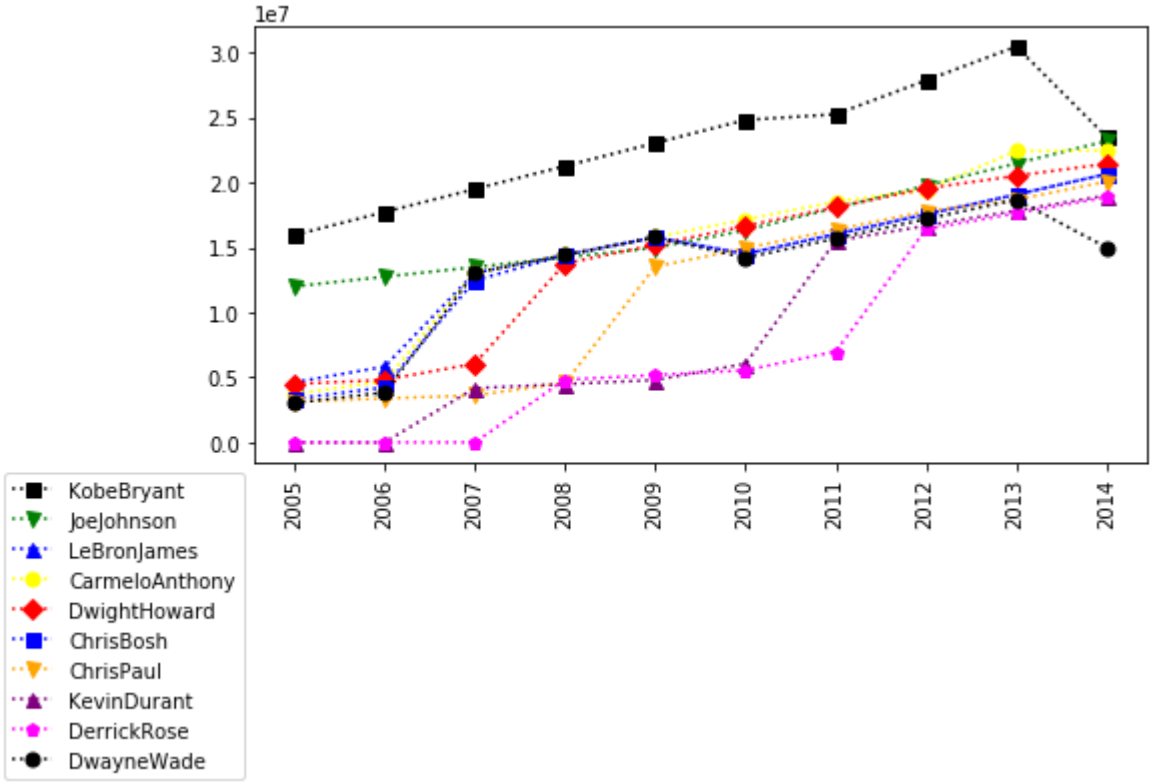


In [197]:
```
myplot(Salary,["KobeBryant","JoeJohnson","LeBronJames"])
```



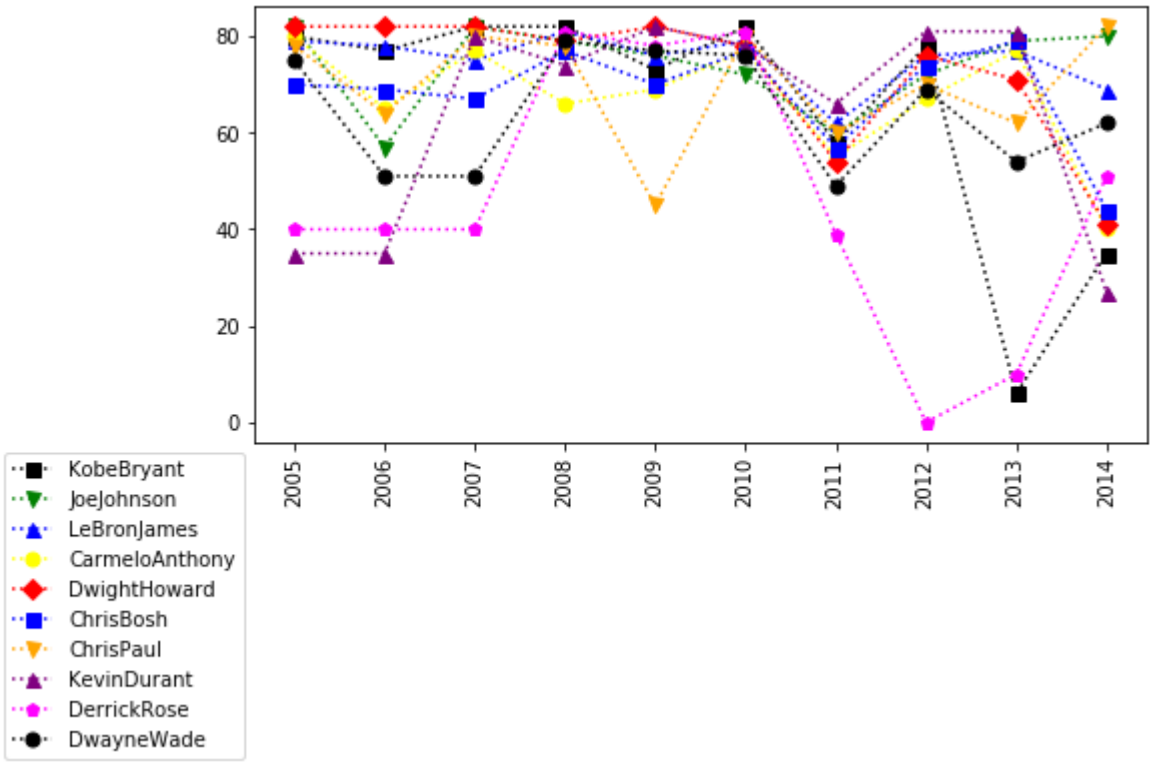
In [198]:
```
def myplot(data,playerlist=Players): #setting up the default value
    Col = {"KobeBryant":"Black","JoeJohnson":"Green","LeBronJames":"Blue","Carmel
    Mark = {"KobeBryant":"s","JoeJohnson":"v","LeBronJames":"^","CarmeloAnthony":
    for i in playerlist:
        plt.plot(data[Pdict[i]],c=Col[i],ls=':',marker=Mark[i],ms=7,label=Players
    plt.legend(loc='bottom right', bbox_to_anchor=(0,0)) #location of plot to the
    plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
    plt.show()
```

In [199]:
```
myplot(Salary)
```

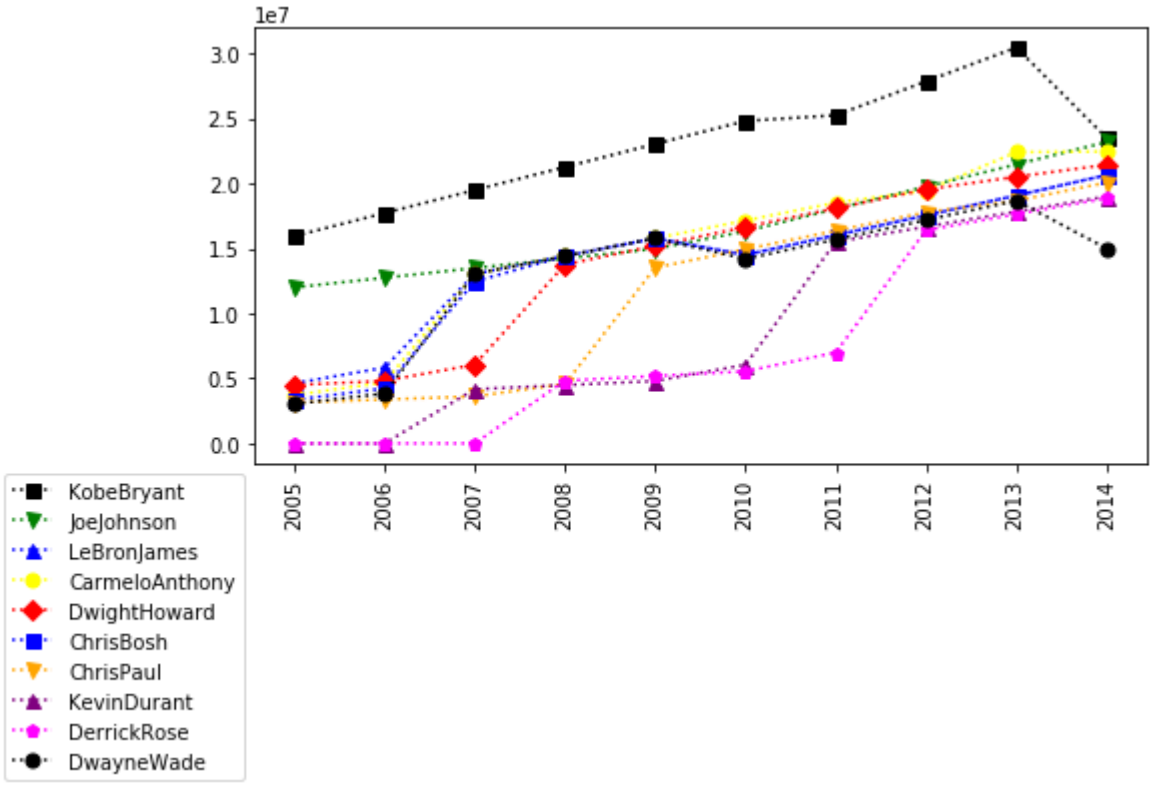


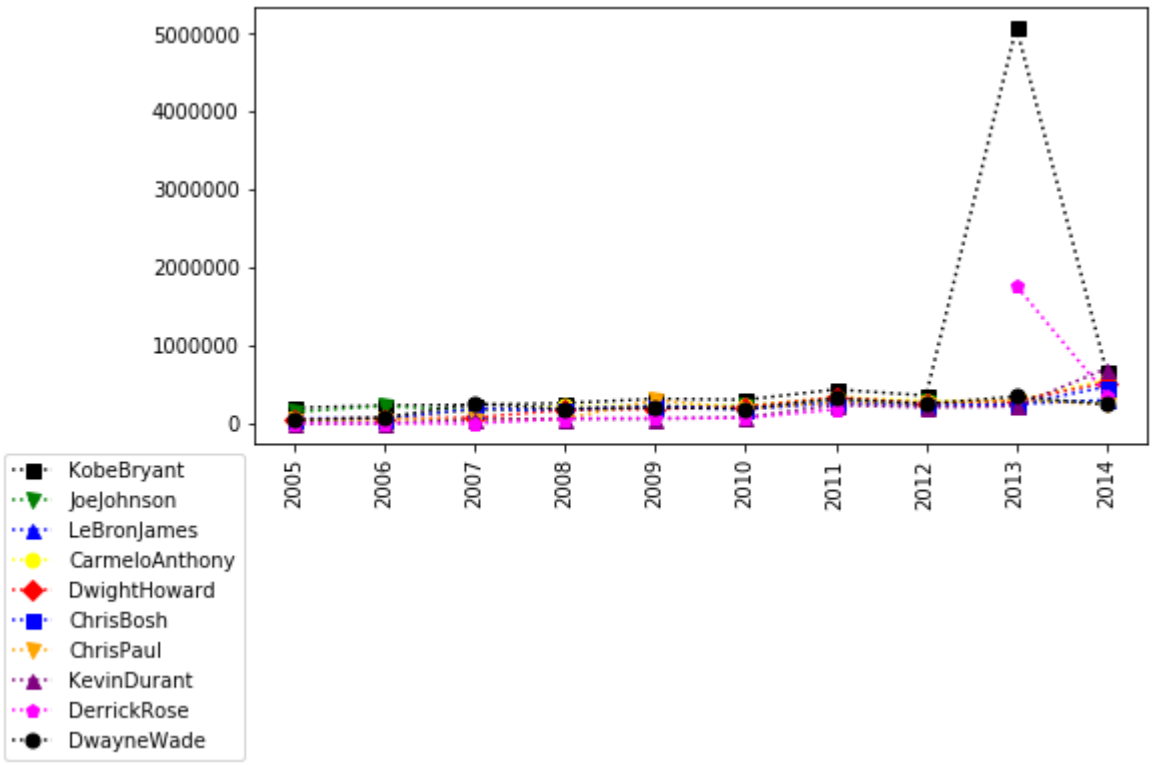
In [200]:
```
#Insights
```

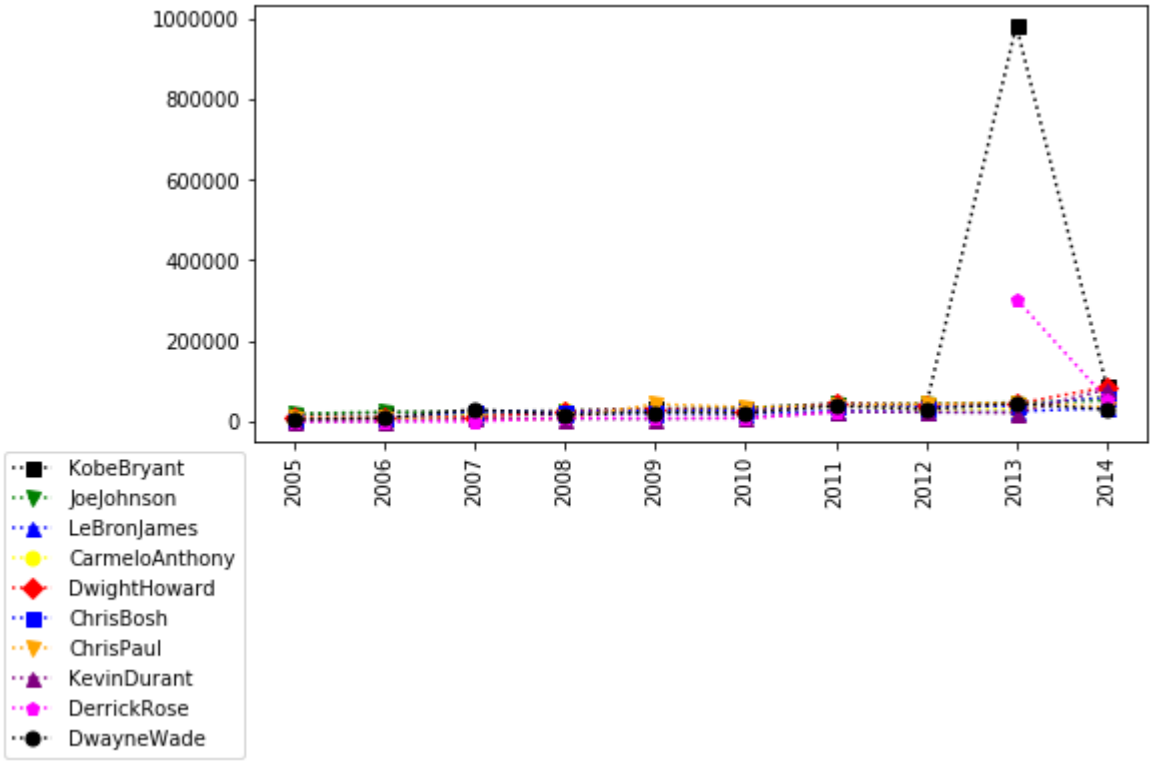
In [201]:
```
myplot(Games)
```



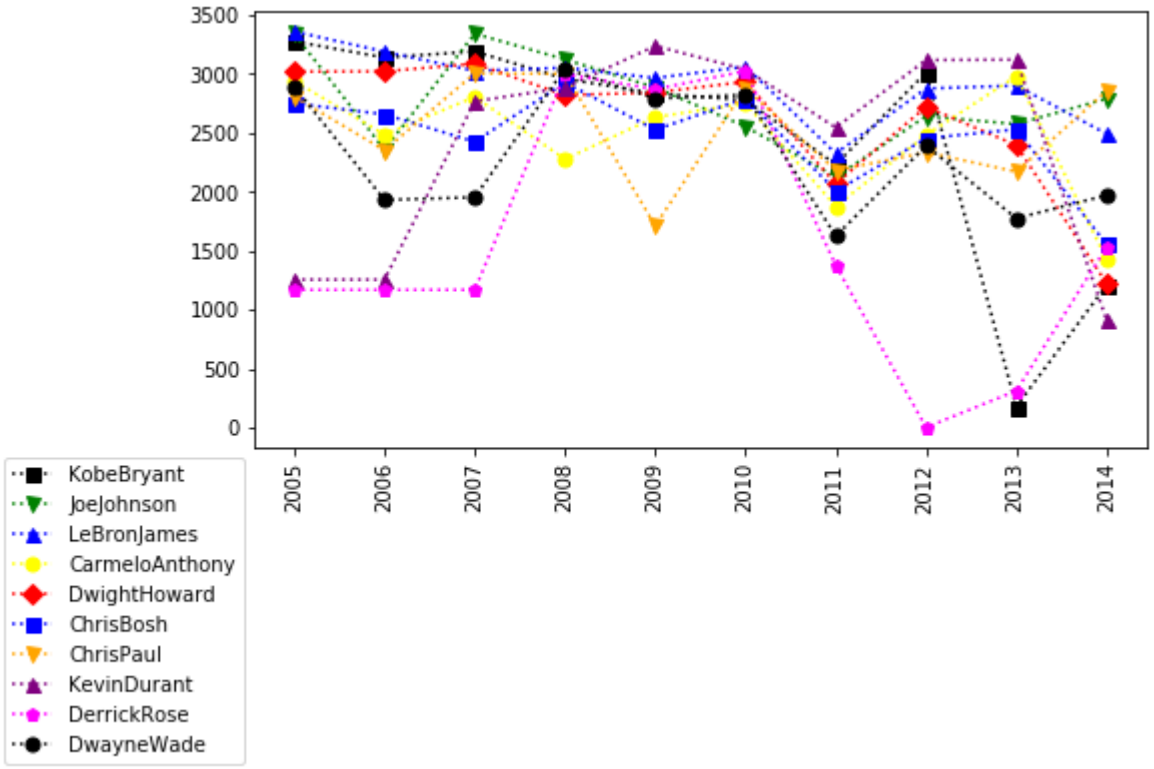
In [206]:
```
myplot(Salary)
myplot(Salary/Games) #Salary per game
myplot(Salary/FieldGoals) #Salary per Goal
```

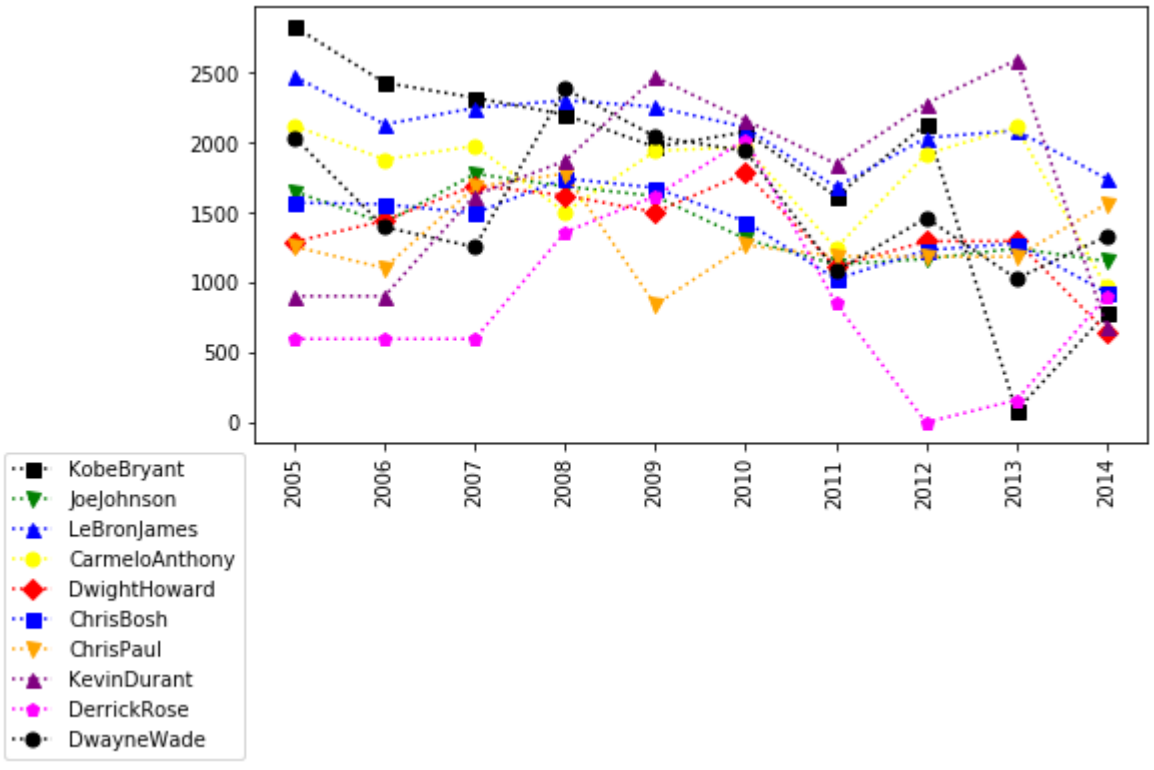





1000000
800000
600000
400000
200000
0
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
KobeBryant
JoeJohnson
LeBronJames
CarmeloAnthony
DwightHoward
ChrisBosh
ChrisPaul
KevinDurant
DerrickRose
DwayneWade

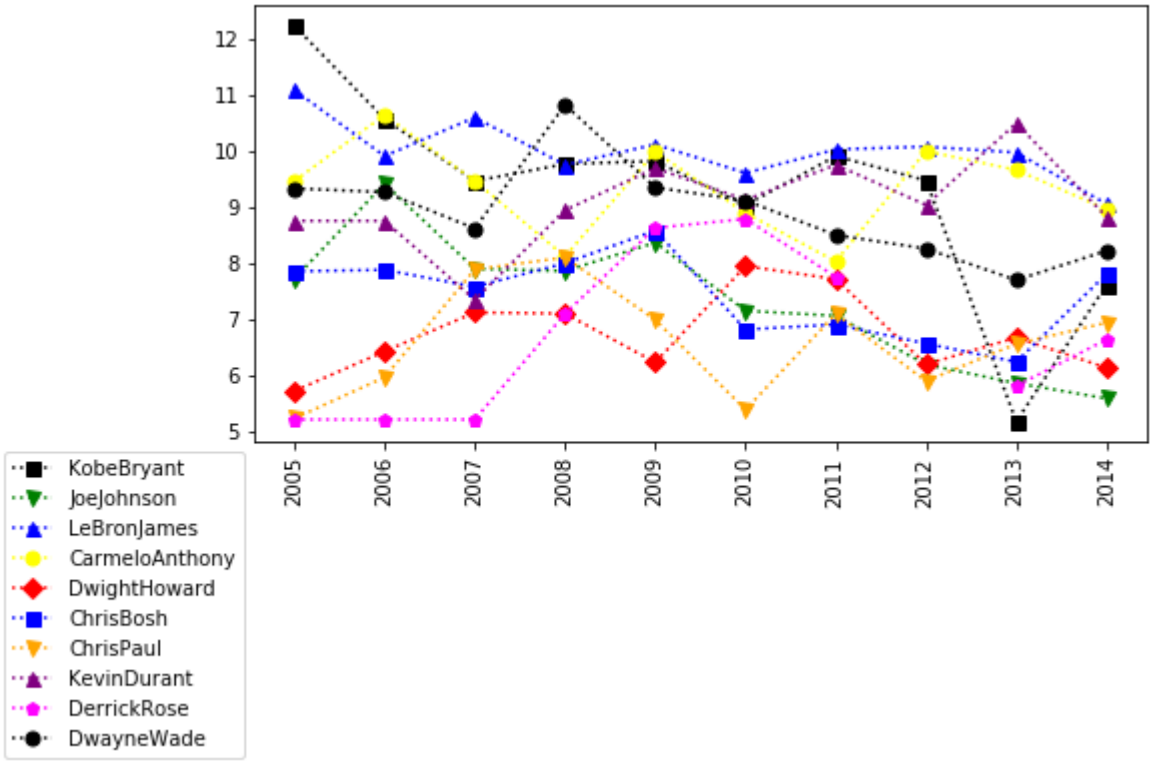
In [207]:
```
#InGameMatrix
myplot(MinutesPlayed)
myplot(Points)
```

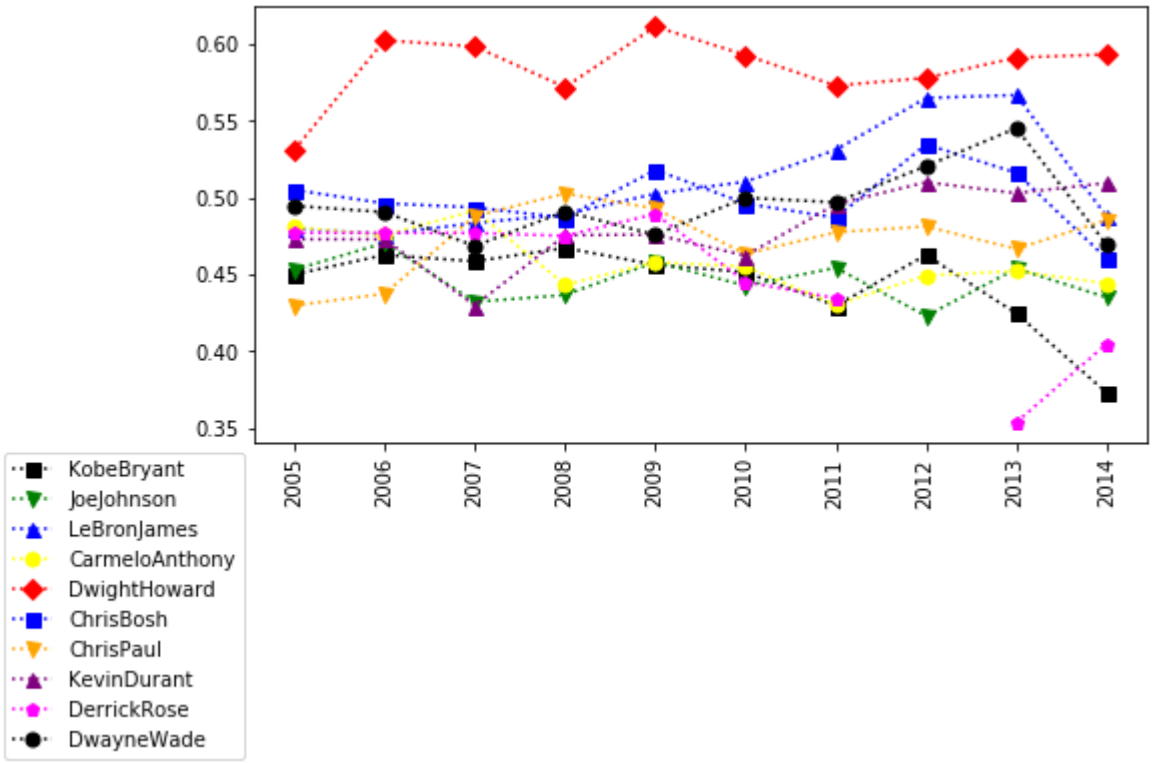


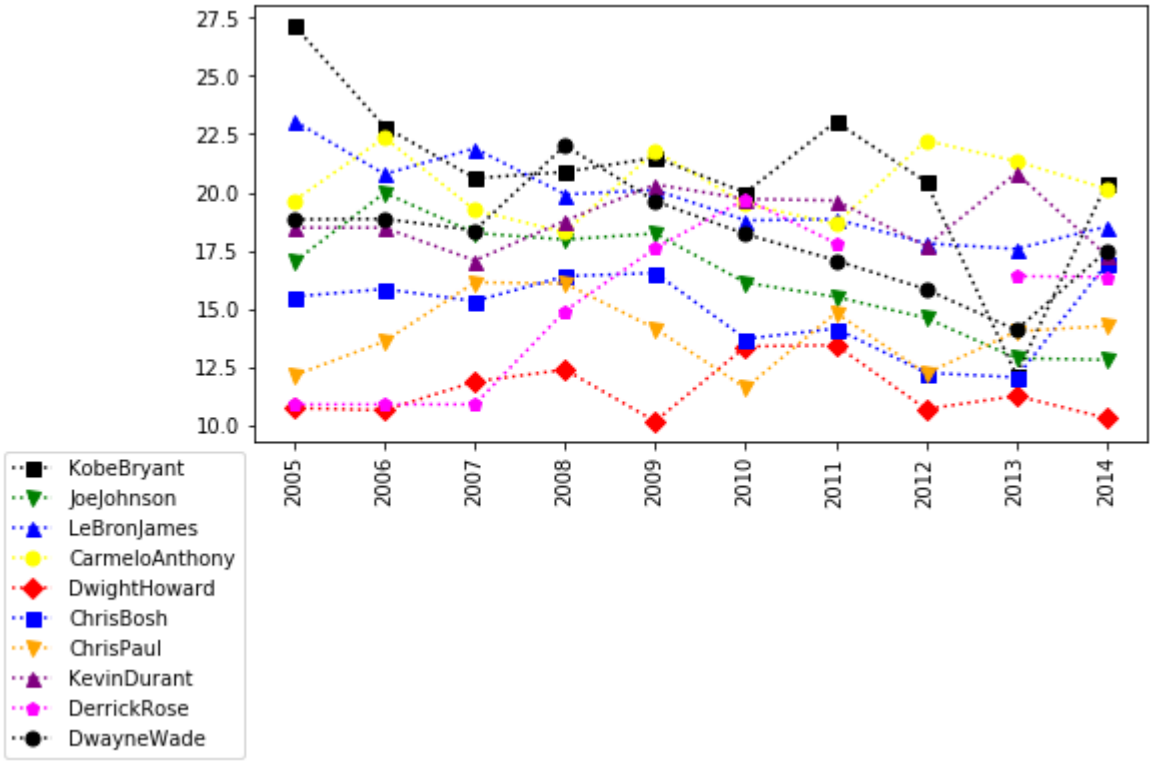


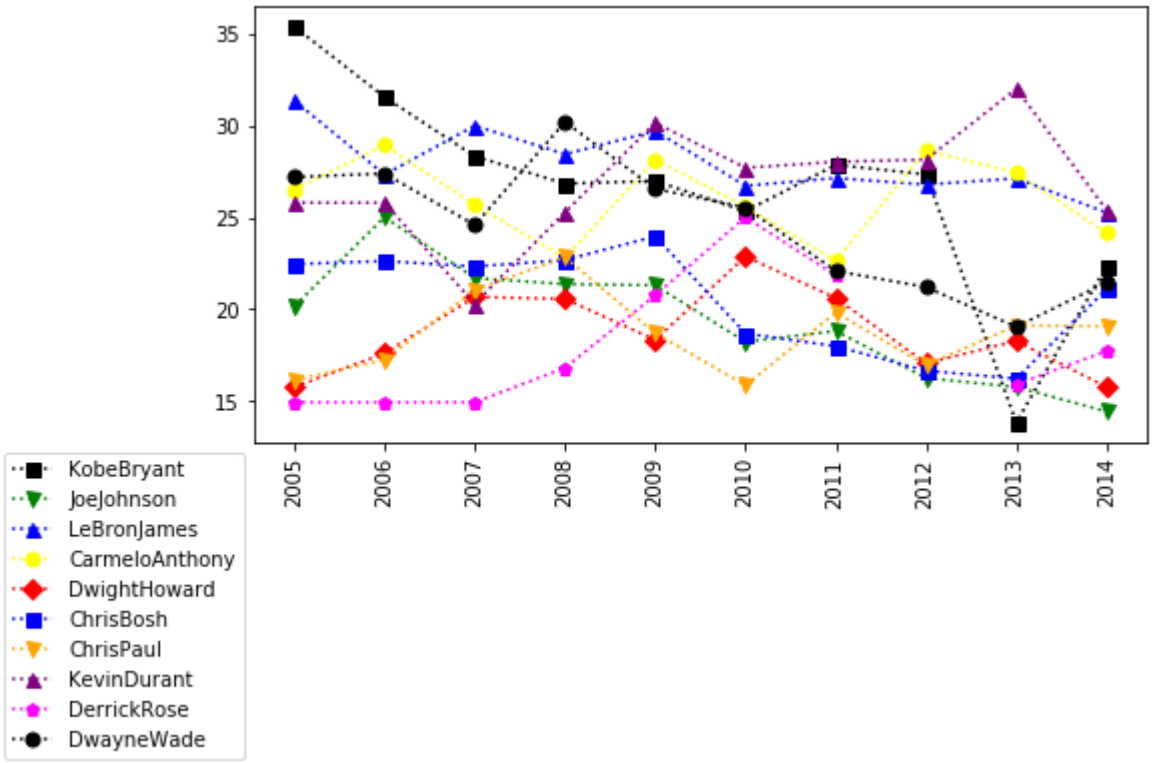
In [208]:
```
#Normalizing InGame Matrix
```

In [209]:
```
myplot(FieldGoals/Games)
myplot(FieldGoals/FieldGoalAttempts)
myplot(FieldGoalAttempts/Games)
myplot(Points/Games)
```

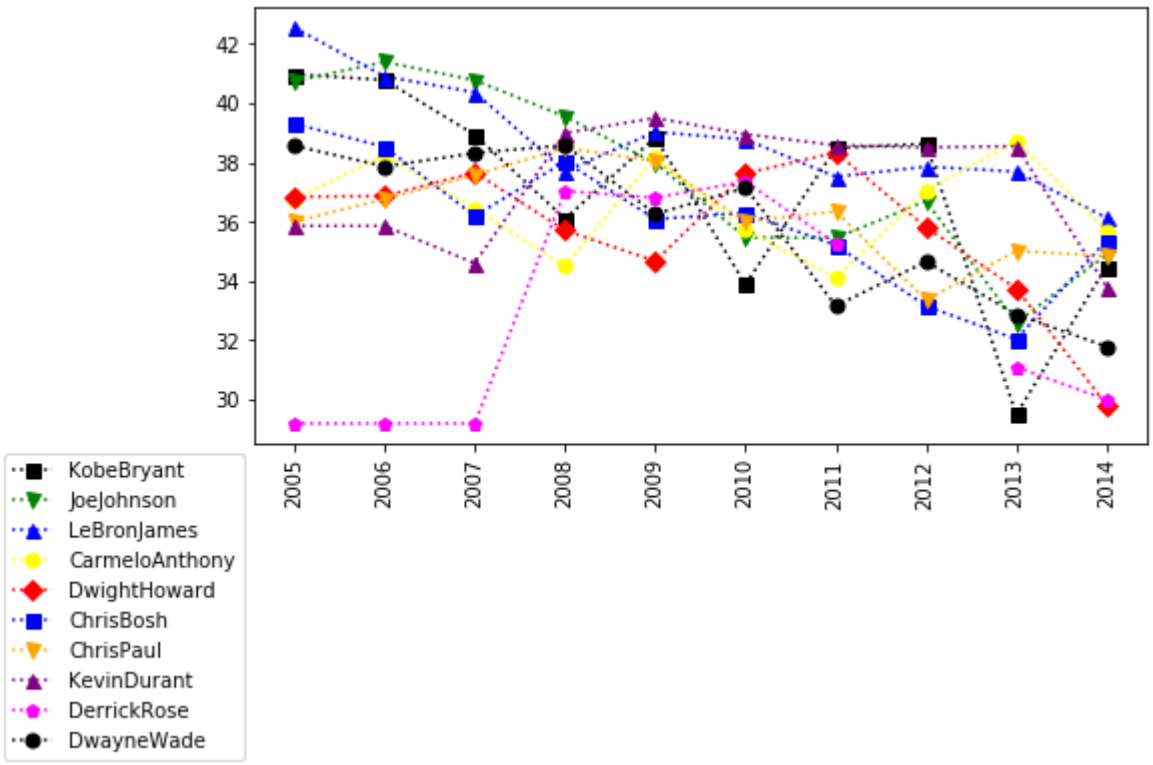


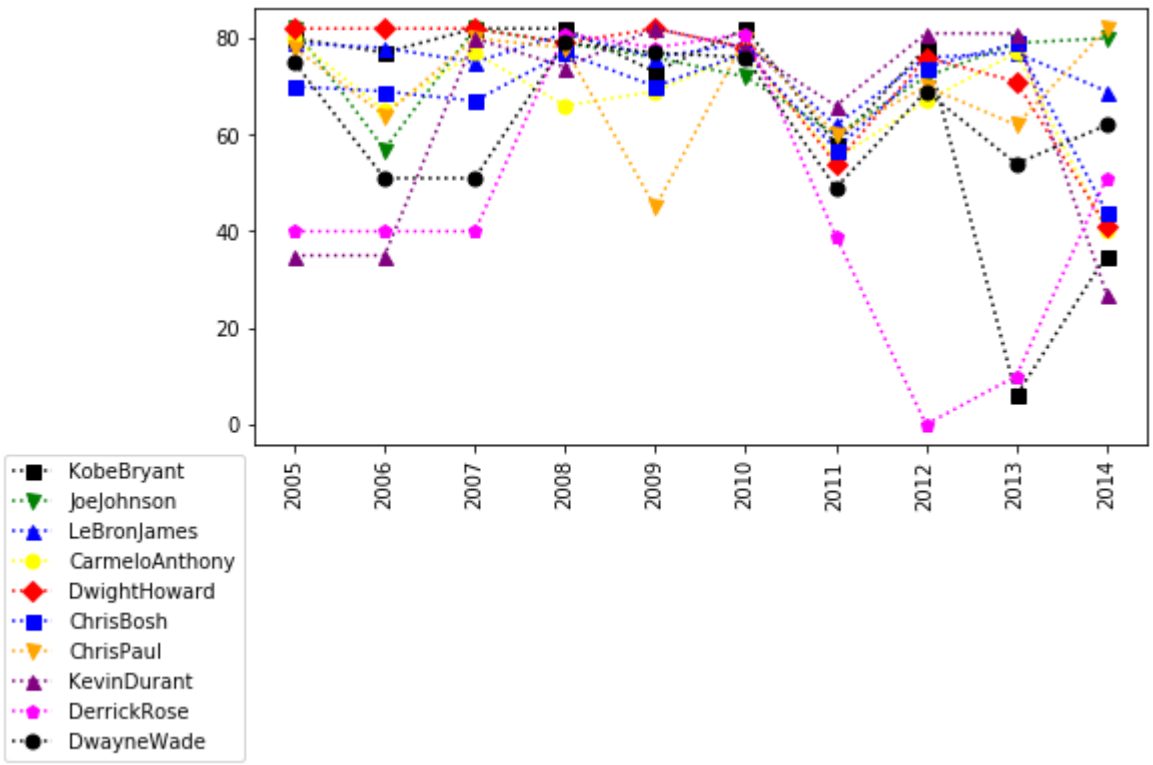



27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
KobeBryant
JoeJohnson
LeBronJames
CarmeloAnthony
DwightHoward
ChrisBosh
ChrisPaul
KevinDurant
DerrickRose
DwayneWade

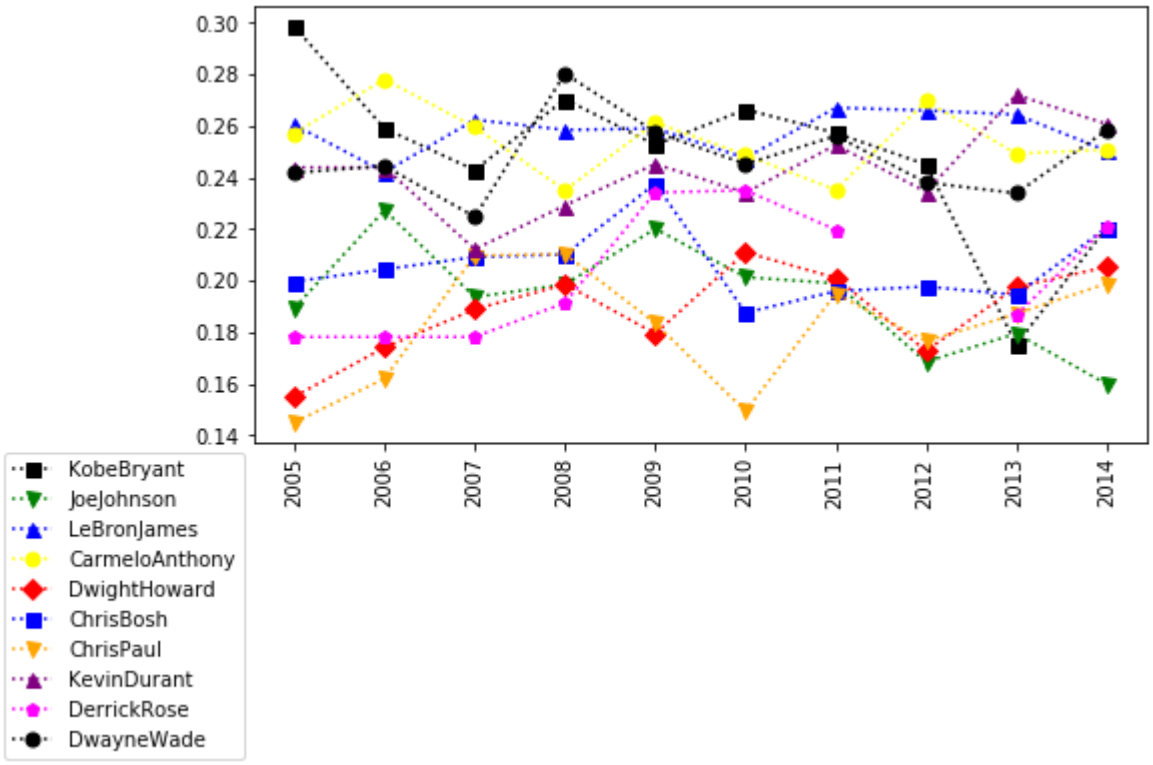

35
30
25
20
15
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
KobeBryant
JoeJohnson
LeBronJames
CarmeloAnthony
DwightHoward
ChrisBosh
ChrisPaul
KevinDurant
DerrickRose
DwayneWade

In [211]:
```
#Interesting Observation
import warnings
warnings.filterwarnings('ignore')
myplot(MinutesPlayed/Games)
myplot(Games)
```

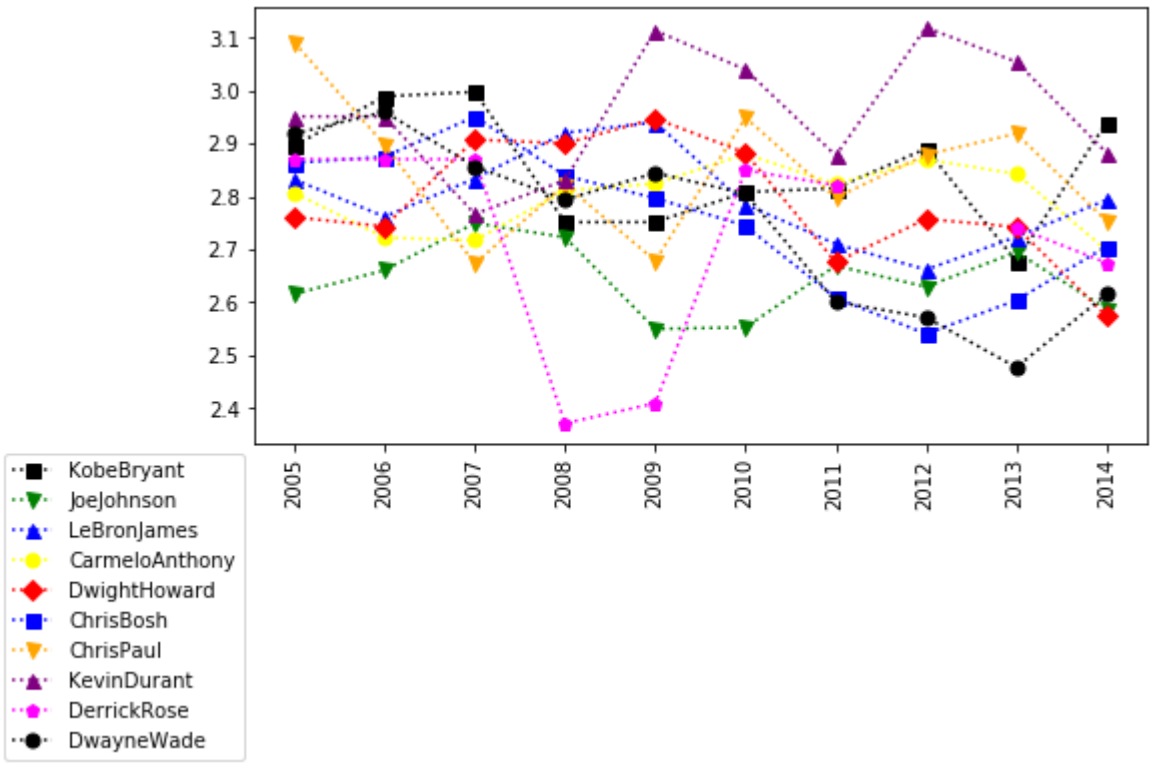




In [213]:
```
myplot(FieldGoals/MinutesPlayed) #Time is Valuable
```



In [214]:
```
myplot(Points/FieldGoals) #Player Style
```



In [ ]:
```

```