Report:

Data Understanding:

We are going to work on the NBA Basketball data. We mainly work on two datasets.

1. "basketball_players.csv"

   It contains statistics of a player. column names of this data frame are

   ```
   'playerID', 'year', 'stint', 'tmID', 'lgID', 'GP', 'GS',
   'minutes', 'points', 'oRebounds', 'dRebounds', 'rebounds',
   'assists', 'steals', 'blocks', 'turnovers', 'PF',
   'fgAttempted', 'fgMade', 'ftAttempted', 'ftMade',
   'threeAttempted', 'threeMade', 'PostGP', 'PostGS',
   'PostMinutes', 'PostPoints', 'PostoRebounds', 'PostdRebounds',
   'PostRebounds', 'PostAssists', 'PostSteals', 'PostBlocks',
   'PostTurnovers', 'PostPF', 'PostfgAttempted', 'PostfgMade',
   'PostftAttempted', 'PostftMade', 'PostthreeAttempted',
   'PostthreeMade', 'note'
   ```

2. "basketball_master.csv”

   It contains demographic data of a player

   'bioID', 'useFirst', 'firstName', 'middleName', 'lastName', 'nameGiven', 'fullGivenName', 'nameSuffix', 'nameNick', 'pos', 'firstseason', 'lastseason', 'height', 'weight', 'college', 'collegeOther', 'birthDate', 'birthCity', 'birthState', 'birthCountry', 'highSchool', 'hsCity', 'hsState', 'hsCountry', 'deathDate', 'race'

ASSIGNMENT QUESTION AND IT’S SOLUTION:

PART 1:

The following answer will cover 1st and 4th question of part1

calculate the mean and median of number of points scored per season and also showing how the number of points change over time

	year	mean(points)
0	1937	42.985915
1	1938	67.959677
2	1939	78.223214
3	1940	72.043478
4	1941	74.189474

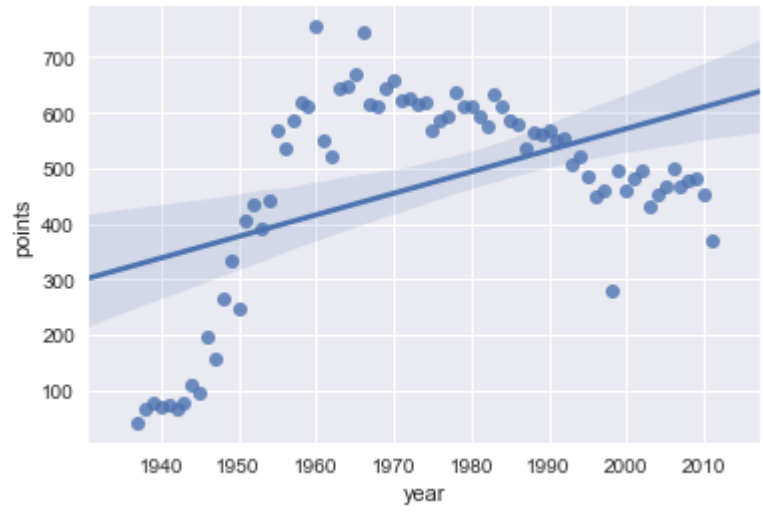
5	1942	65.500000
6	1943	79.021739
7	1944	111.573171
8	1945	94.722222
9	1946	196.050000
10	1947	158.304536
11	1948	263.963068
12	1949	333.888476
13	1950	247.460000
14	1951	404.370079
15	1952	434.694656
16	1953	393.280992
17	1954	442.103448
18	1955	569.560000
19	1956	537.380952
20	1957	584.980952
21	1958	617.247525
22	1959	611.351351
23	1960	754.060606
24	1961	550.060377
25	1962	520.721053
26	1963	644.524194
27	1964	647.487805
28	1965	670.685484
29	1966	743.226562
..	...	...
45	1982	576.501408

46	1983	633.134146
47	1984	611.230994
48	1985	587.217514
49	1986	580.778711
50	1987	536.823684
51	1988	565.143939
52	1989	562.665083
53	1990	567.156627
54	1991	548.600000
55	1992	553.617577
56	1993	506.164414
57	1994	522.134884
58	1995	483.883436
59	1996	450.945205
60	1997	459.129293
61	1998	280.151899
62	1999	495.275641
63	2000	460.120408
64	2001	483.070213
65	2002	495.837719
66	2003	429.588008
67	2004	454.579848
68	2005	466.095703
69	2006	498.765914
70	2007	467.975332
71	2008	478.815891
72	2009	482.617188

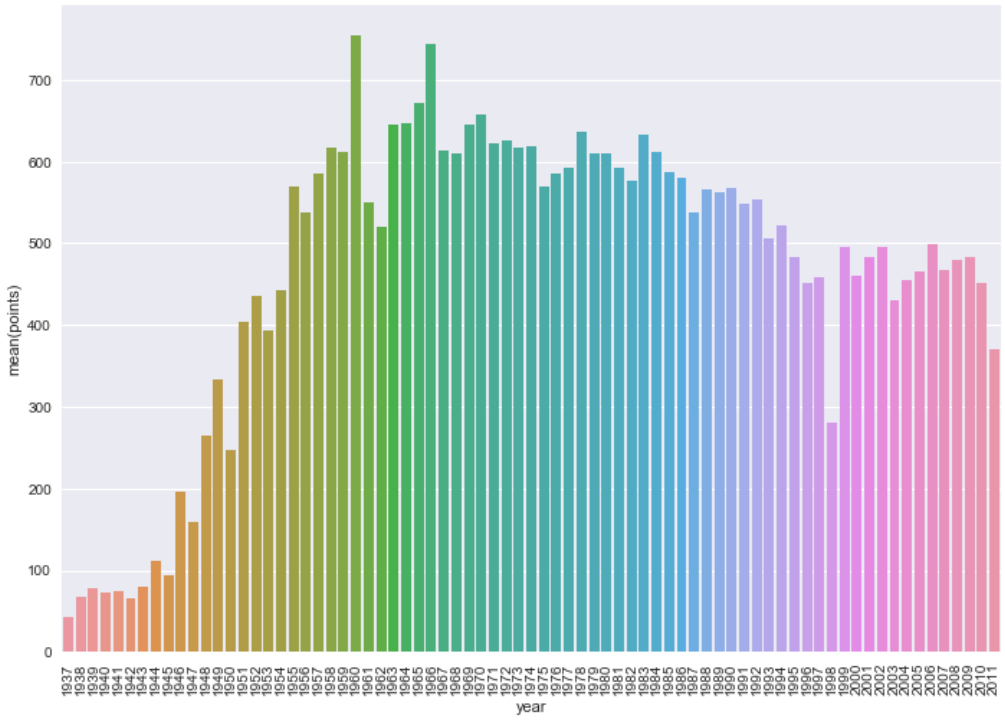
73 2010 451.533088

74 2011 369.899225

Regplot of year vs mean(points)



Barplot of year vs mean(points)



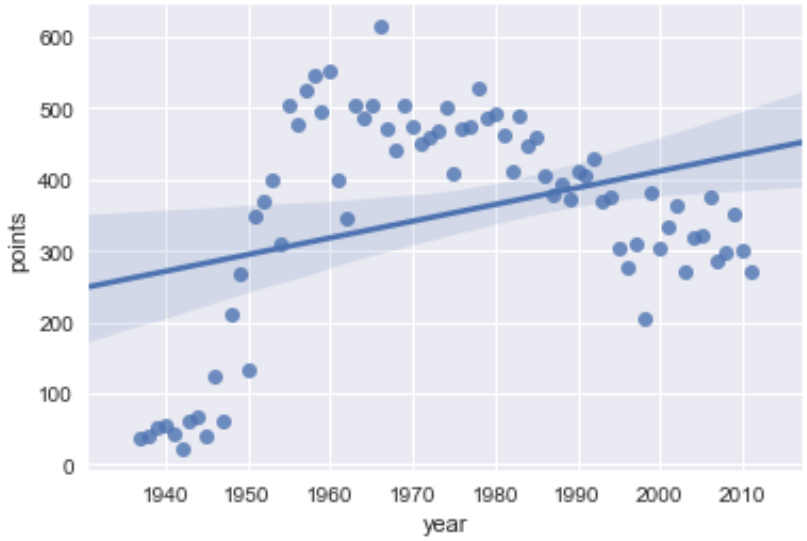
Result of year and median(points)

	year	median(points)
0	1937	36.0
1	1938	39.0
2	1939	53.5
3	1940	54.5
4	1941	43.0
5	1942	23.0
6	1943	61.0
7	1944	68.5
8	1945	39.5
9	1946	125.0
10	1947	60.0
11	1948	209.5
12	1949	267.0
13	1950	133.0
14	1951	347.0
15	1952	369.0
16	1953	400.0
17	1954	310.0
18	1955	503.0
19	1956	476.0
20	1957	525.0
21	1958	546.0
22	1959	494.0
23	1960	552.0
24	1961	399.0
25	1962	344.5
26	1963	504.0
27	1964	486.0

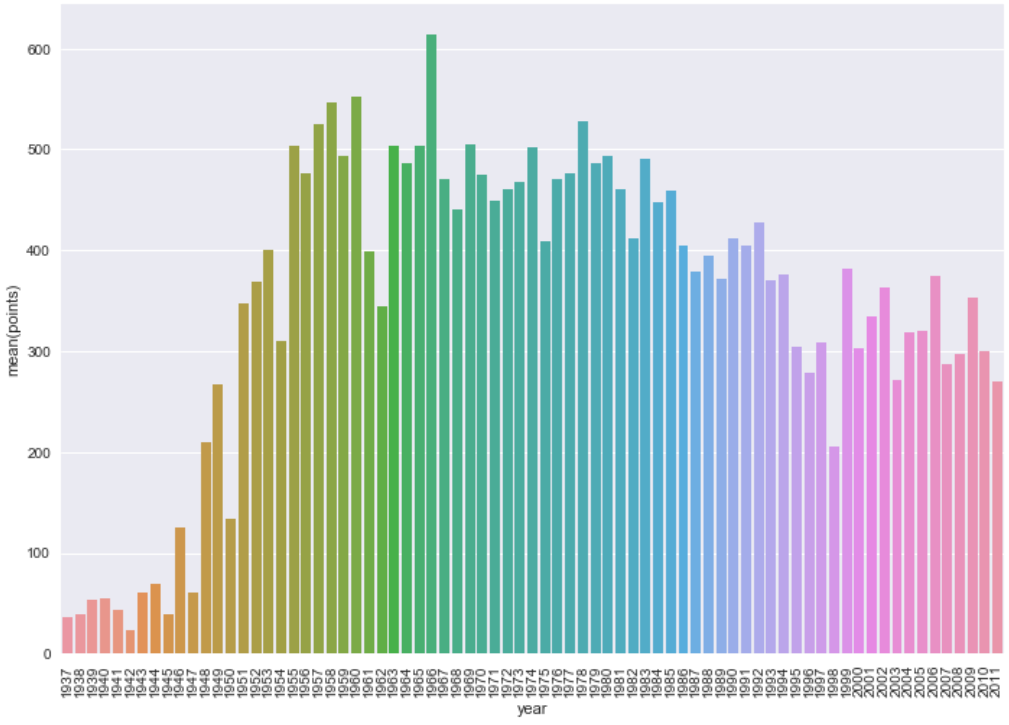
28	1965	503.5
29	1966	613.5
..	...	...
45	1982	411.0
46	1983	490.5
47	1984	448.0
48	1985	458.5
49	1986	404.0
50	1987	379.0
51	1988	394.5
52	1989	372.0
53	1990	411.0
54	1991	405.0
55	1992	428.0
56	1993	370.0
57	1994	376.0
58	1995	304.0
59	1996	278.0
60	1997	309.0
61	1998	206.0
62	1999	381.0
63	2000	303.0
64	2001	334.5
65	2002	363.0
66	2003	271.0
67	2004	318.0
68	2005	320.5
69	2006	374.0
70	2007	287.0
71	2008	297.0

72 2009 352.5
73 2010 300.5
74 2011 269.5

Regplot of Year vs median(points)



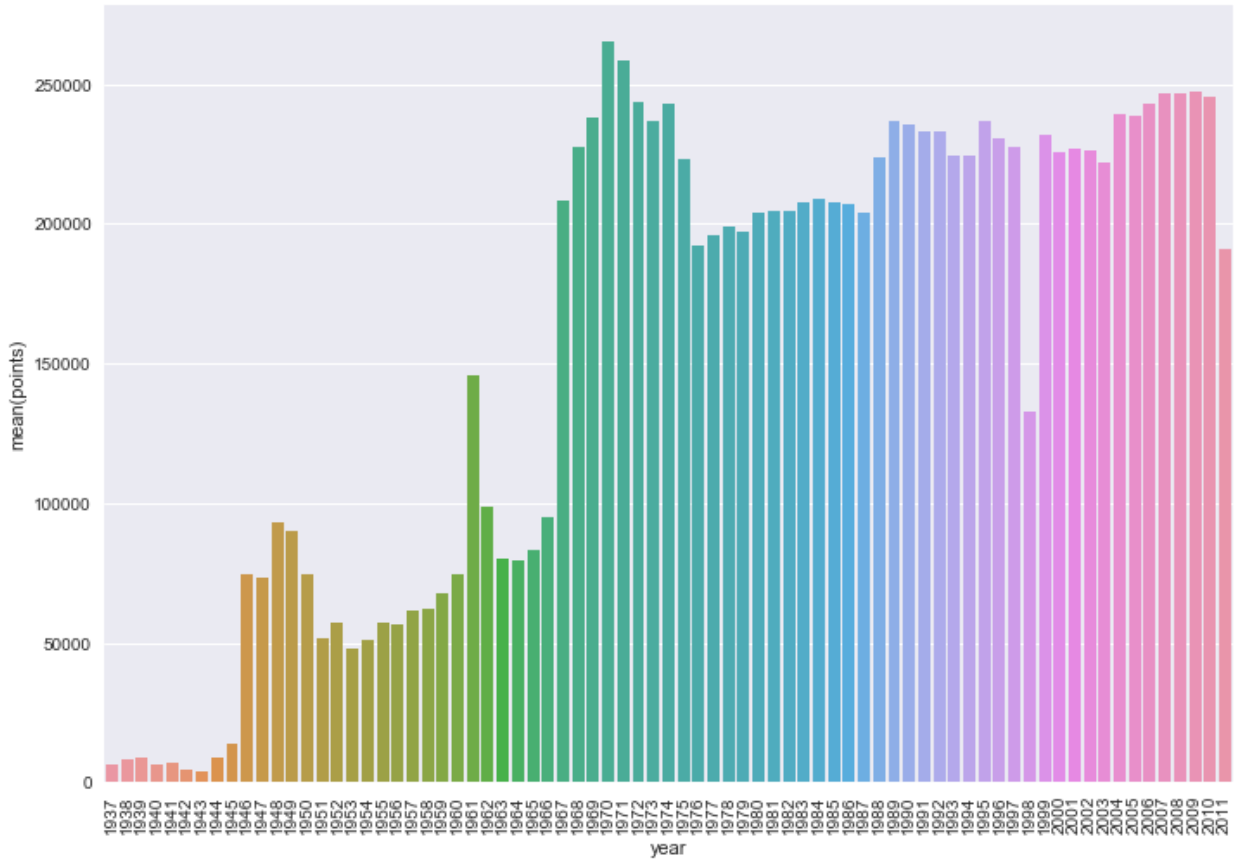
Bar plot of Year vs median(points)



The following answer will cover 2nd question of part1

Question: Determine the highest number of points recorded in single season. Identify who scored those points and the year they did so.

Bar plot of year vs sum(points)



As we can see clearly

maximum points scored in year 1970 and the maximum points is 265257.

And these are the name of the players who did it.

hese are the name of the players

	useFirst	lastName
3887	Kareem	Abdul-Jabbar
3888	Mahdi	Abdul-Rahman

3889	Zaid	Abdul-Aziz
3890	Don	Adams
3891	Rick	Adelman
3892	Lucius	Allen
3893	Cliff	Anderson
3894	Cliff	Anderson
3895	Nate	Archibald
3896	Jim	Ard
3897	Bob	Arnzen
3898	Al	Attles
3899	Dennis	Awtrey
3900	Walker	Banks
3901	Dick	Barnett
3902	Jim	Barnes
3903	Jim	Barnett
3904	John	Barnhill
3905	John	Barnhill
3906	Mike	Barrett
3907	Moe	Barr
3908	Rick	Barry
3909	Johnny	Baum
3910	Elgin	Baylor
3911	Charlie	Beasley
3912	Charlie	Beasley
3913	John	Beasley
3914	Zelmo	Beaty
3915	Byron	Beck

3916	Art	Becker
...	...	...
4260	Hubie	White
4261	Jo Jo	White
4262	Lenny	Wilkens
4263	Al	Williams
4264	Art	Williams
4265	Bernie	Williams
4266	Charles	Williams
4267	Charles	Williams
4268	Chuck	Williams
4269	Milt	Williams
4270	Ron	Williams
4271	Vann	Williford
4272	Willie	Williams
4273	Willie	Williams
4274	George	Wilson
4275	Jim	Wilson
4276	Steve	Wilson
4277	Lee	Winfield
4278	Marv	Winkler
4279	Willie	Wise
4280	Greg	Wittman
4281	Greg	Wittman
4282	Tom	Workman
4283	Tom	Workman
4284	Howie	Wright

4285 Lonnie Wright

4286 Lonnie Wright

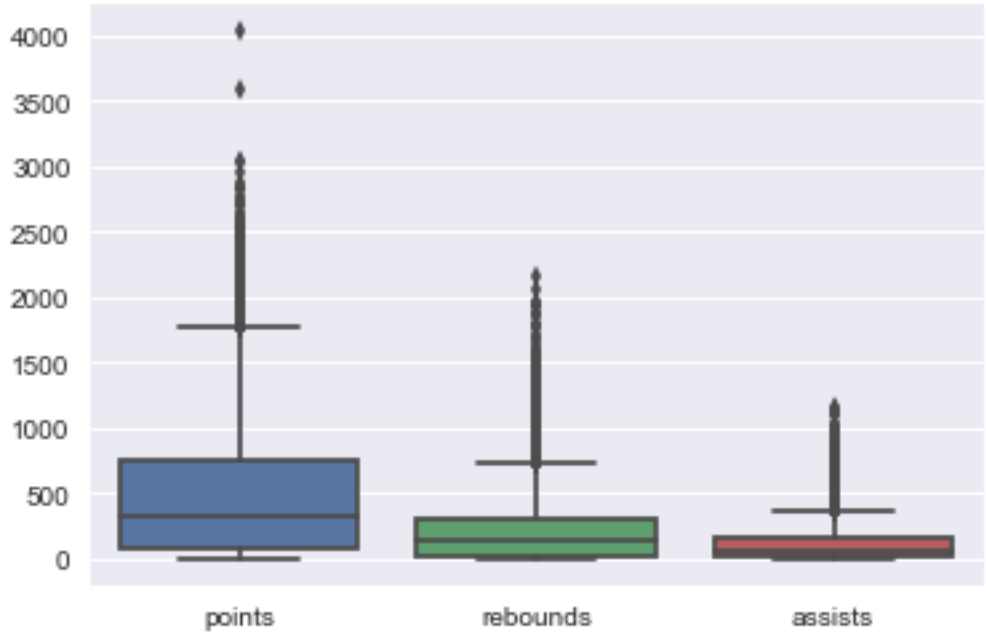
4287 Gary Zeller

4288 Bill Zopf

21817 Henry Logan

PART1 -3RD QUESTION

We have to make boxplot



PART 2:

Question 1: most efficient players

I think the most efficient players are those players who score more points in less attempt. Here the idea is firstly group by player and then finding those players who score more in less attempt.

these are the name of the most efficient players

	useFirst	lastName
0	Troy	Murphy
1	Shareef	Abdur-Rahim
2	Mike	Dunleavy
3	Terry	Dischinger
4	Sidney	Wicks
5	Calbert	Cheaney
6	Kevin	Martin
7	Steve	Francis
8	Kenny	Sears
9	Clarence	Weatherspoon

2nd part- 2nd question

Question: - I have to find out the player who is allrounder

Simple logic I applied here is firstly I selected top 500 points scorer player and then applied filter on it to choose best 100 rebounders and then top 50 assist score holder on those 100 rebounders and lastly on those 50 I find out top 5 who steals ball very well.

these are top 5 allrounders

	useFirst	lastName
0	Clarence	Weatherspoon
1	Steve	Francis
2	Shareef	Abdur-Rahim
3	Calbert	Cheaney

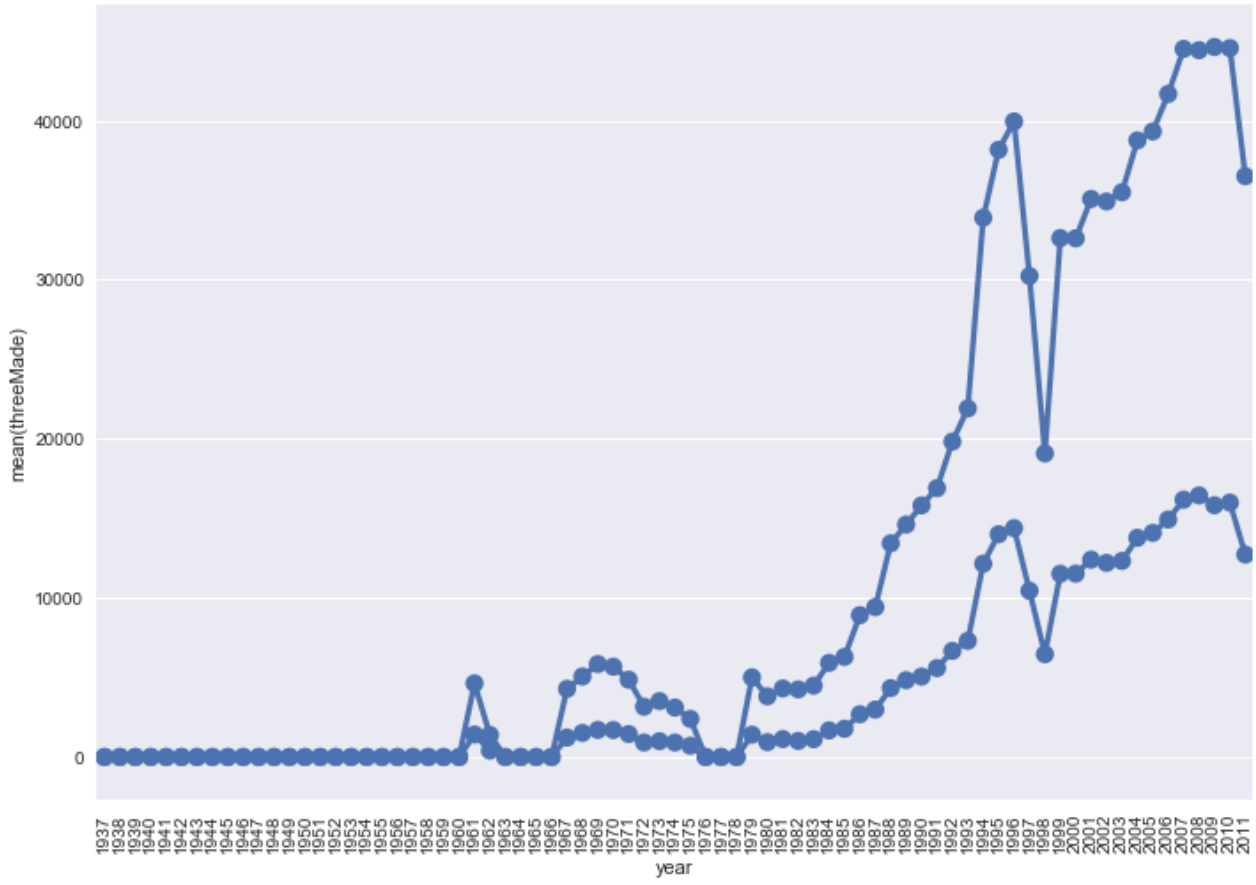
4 Sidney Wicks

2nd part- 3rd question

Question: -popularity of three-point shots

Below in the picture one line indicates attempt for three shots over year and 2nd line player successfully makes it.

We can see by the graph popularity of three shots started being popular dramatically after 1978.



PART 3:

Question 1: to find GOAT player.

The basic concept I used here is firstly I grouped the dataframe by player then on merged dataset I added four more column which is "points_per_match","rebounds_per_match","steals_per_match" and "assists_per_match".

Firstly I sort the data frame on the points per match and then the result is

useFirst	lastName	points_per_match	rebounds_per_match	steals_per_match	assists_per_match	
818	Mike	Conley	30.224215	4.318386	4.327354	7.946188
2248	Michael	Jordan	30.123134	6.223881	2.345149	5.254664
726	Wilt	Chamberlain	30.066029	22.893780	0.000000	4.443062
2088	LeBron	James	27.641509	7.174165	1.732946	6.895501
266	Elgin	Baylor	27.362884	13.549645	0.000000	4.314421
4634	Jerry	West	27.030043	5.757511	0.086910	6.693133
2040	Allen	Iverson	26.660832	3.713348	2.169584	6.153173
3400	Bob	Pettit	26.363636	16.223485	0.000000	2.991162
1153	Kevin	Durant	26.263158	6.613158	1.213158	2.823684
3652	Oscar	Robertson	25.682692	7.503846	0.074038	9.506731
570	Kobe	Bryant	25.395349	5.289406	1.483204	4.667528
4503	Dwyane	Wade	25.154362	5.067114	1.770134	6.203020
1501	George	Gervin	25.089623	5.284906	1.210377	2.639623
2695	Karl	Malone	25.018970	10.140921	1.412602	3.555556
4692	Dominique	Wilkins	24.830540	6.675047	1.283054	2.492551
241	Rick	Barry	24.783333	6.728431	1.082353	4.854902
116	Carmelo	Anthony	24.653251	6.334365	1.117647	3.111455
1	Kareem	Abdul-Jabbar	24.607051	11.179487	0.743590	3.628205

useFirst	lastName	points_per_match	rebounds_per_match	steals_per_match	assists_per_match	
349	Larry	Bird	24.293200	10.004459	1.734671	6.348941
944	Adrian	Dantley	24.269110	5.712042	0.988482	2.963351
2710	Pete	Maravich	24.237082	4.174772	0.892097	5.414894
1253	Julius	Erving	24.156074	8.467418	1.827836	4.164119
3243	Shaquille	O'Neal	23.691798	10.852527	0.612262	2.507042
3186	Dirk	Nowitzki	22.875829	8.278673	0.877725	2.645498
130	Paul	Arizin	22.813464	8.596073	0.000000	2.335203
4110	NaN	Spivey	22.696078	10.705882	0.000000	1.607843
4321	David	Thompson	22.672297	4.131757	1.005068	3.275338
2932	George	Mikan	22.645833	7.892045	0.000000	2.357955
2038	Dan	Issel	22.563218	9.140394	0.774220	2.386700
1660	Alex	Groza	22.500000	5.453846	0.000000	2.446154
2341	Bernard	King	22.488558	5.789474	0.990847	3.275744
213	Charles	Barkley	22.140727	11.692451	1.535881	3.928239
2779	Bob	McAdoo	22.050469	9.446009	0.881455	2.289906
3419	Paul	Pierce	22.040000	6.009756	1.464390	3.839024

Conclusion-mike conley has the best points per match and micheal jordan at 2nd position

Then sort on assist per match and the result will be

	useFirst	lastName	points_per_match	rebounds_per_match	steals_per_match	assists_per_match
3652	Oscar	Robertson	25.682692	7.503846	0.074038	9.506731
818	Mike	Conley	30.224215	4.318386	4.327354	7.946188

	useFirst	lastName	points_per_match	rebounds_per_match	steals_per_match	assists_per_match
2088	LeBron	James	27.641509	7.174165	1.732946	6.895501
3711	Derrick	Rose	20.996416	3.838710	0.870968	6.849462
4634	Jerry	West	27.030043	5.757511	0.086910	6.693133
349	Larry	Bird	24.293200	10.004459	1.734671	6.348941
4503	Dwyane	Wade	25.154362	5.067114	1.770134	6.203020
2040	Allen	Iverson	26.660832	3.713348	2.169584	6.153173
2710	Pete	Maravich	24.237082	4.174772	0.892097	5.414894
2248	Michael	Jordan	30.123134	6.223881	2.345149	5.254664
3866	Charlie	Scott	20.693166	3.969317	0.847978	4.902371
241	Rick	Barry	24.783333	6.728431	1.082353	4.854902
1803	John	Havlicek	20.783465	6.304724	0.374803	4.814173
570	Kobe	Bryant	25.395349	5.289406	1.483204	4.667528
3396	Geoff	Petrie	21.820628	2.849776	0.553812	4.612108
726	Wilt	Chamberlain	30.066029	22.893780	0.000000	4.443062

here also mike conley came at 2nd position and micheal jordon at 10th position and then on steals per match

	useFirst	lastName	points_per_match	rebounds_per_match	steals_per_match	assists_per_match
818	Mike	Conley	30.224215	4.318386	4.327354	7.946188
2248	Michael	Jordan	30.123134	6.223881	2.345149	5.254664
2040	Allen	Iverson	26.660832	3.713348	2.169584	6.153173
1253	Julius	Erving	24.156074	8.467418	1.827836	4.164119
4503	Dwyane	Wade	25.154362	5.067114	1.770134	6.203020
3225	Hakeem	Olajuwon	21.765751	11.104200	1.746365	2.470113
349	Larry	Bird	24.293200	10.004459	1.734671	6.348941
2088	LeBron	James	27.641509	7.174165	1.732946	6.895501
213	Charles	Barkley	22.140727	11.692451	1.535881	3.928239
570	Kobe	Bryant	25.395349	5.289406	1.483204	4.667528

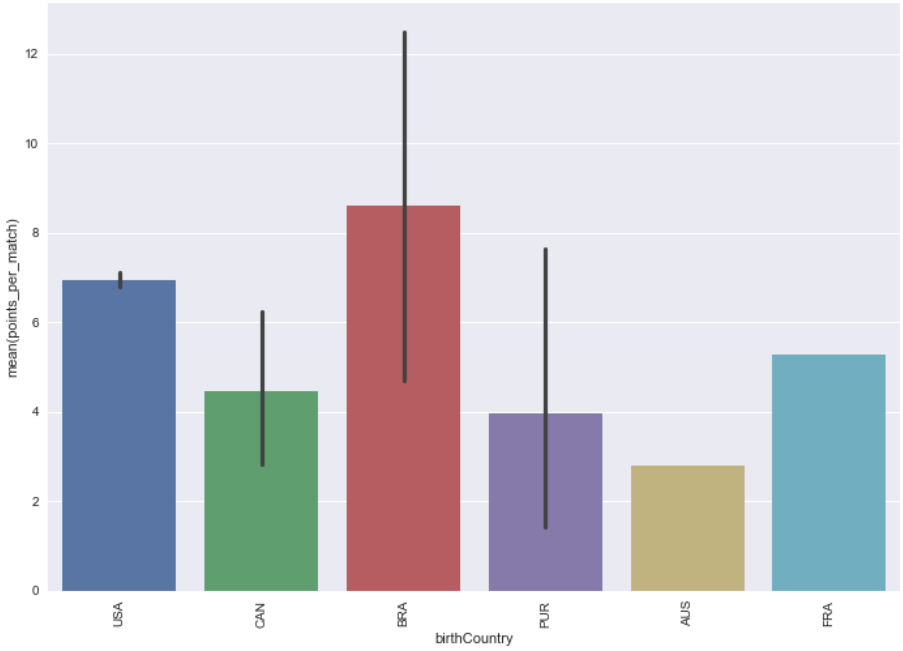
Here also mike conley came at 1st position and micheal jordon at 2nd position.

Part3 -Question 2:

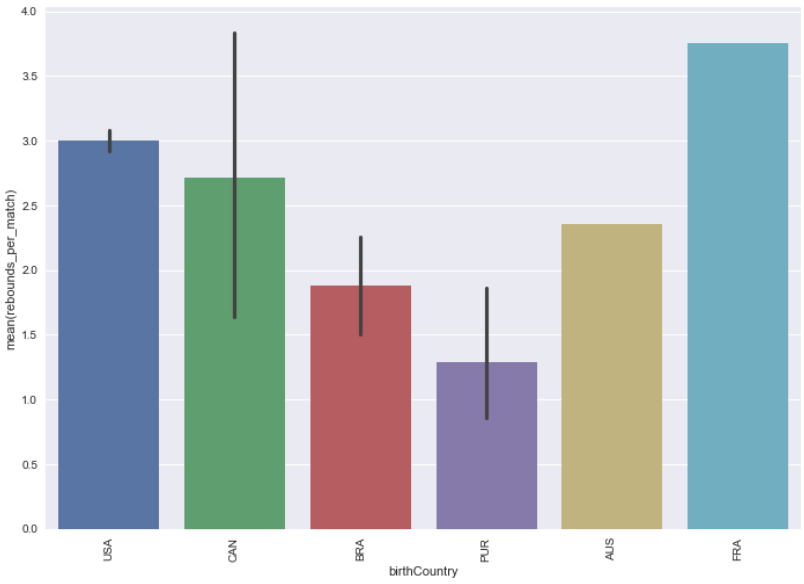
Question: - Find out anything interesting that come from same region.

The following figure is showing players points per match vs country. people who born in brazil are the players who scores most point and then USA.

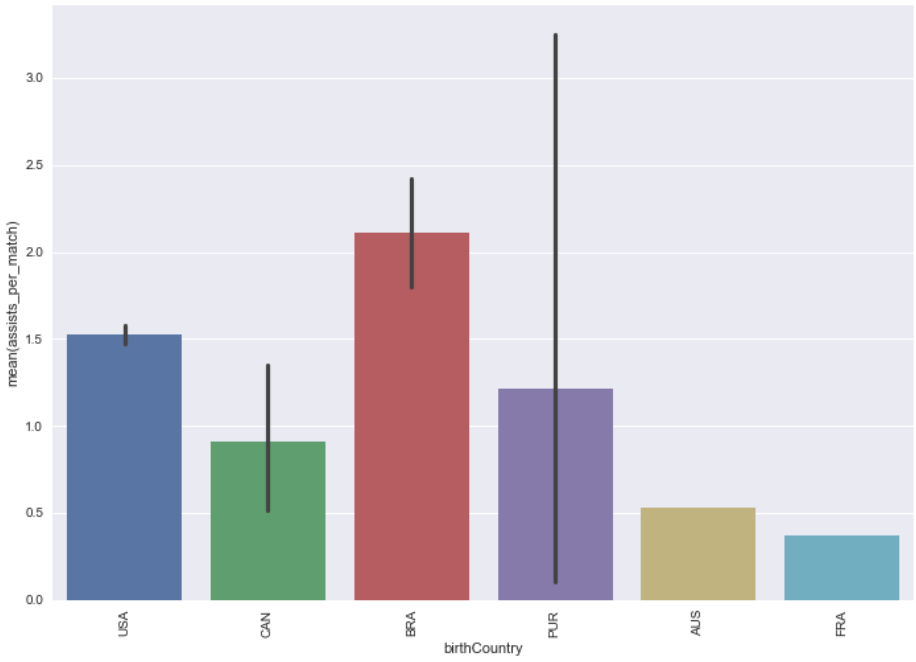
people born in Australia are the players who scores less than other countries

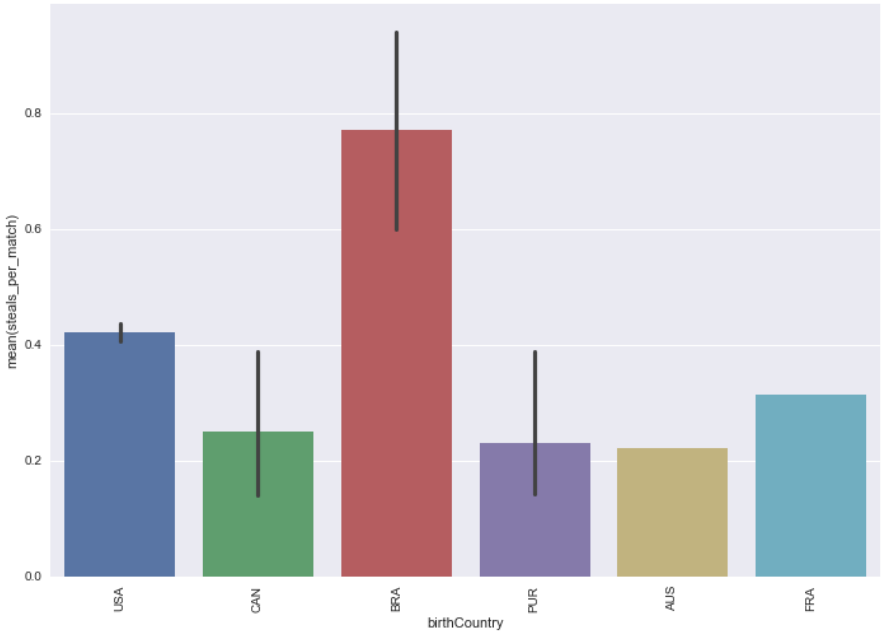


The following figures is showing players rebounds per match vs country



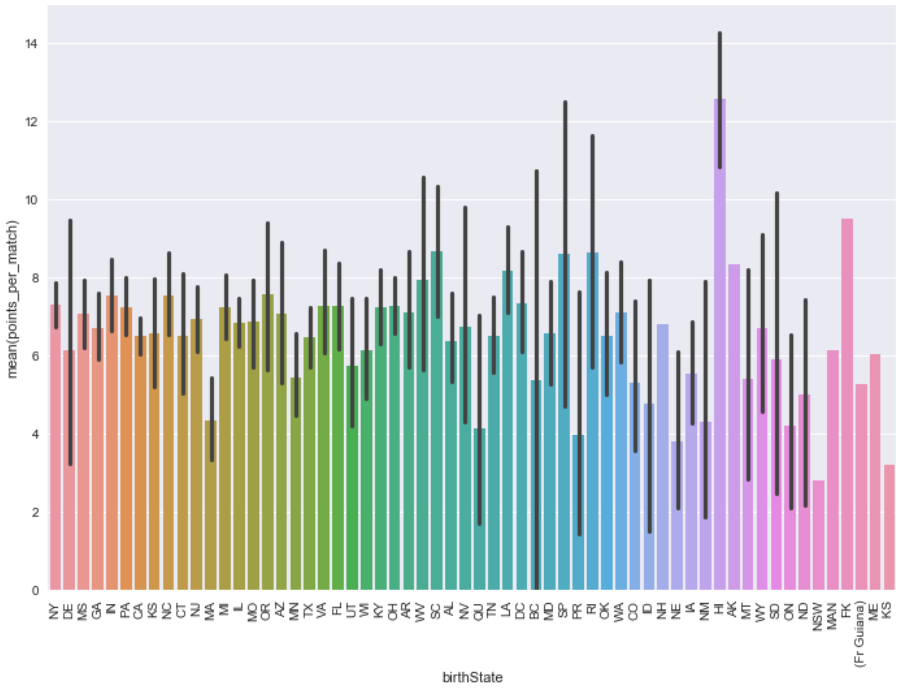
By looking at the graph we can say that Brazilian scores more point but rebounding capability of Brazilian players is at the 2nd last position. people born in France is has good rebounding capability

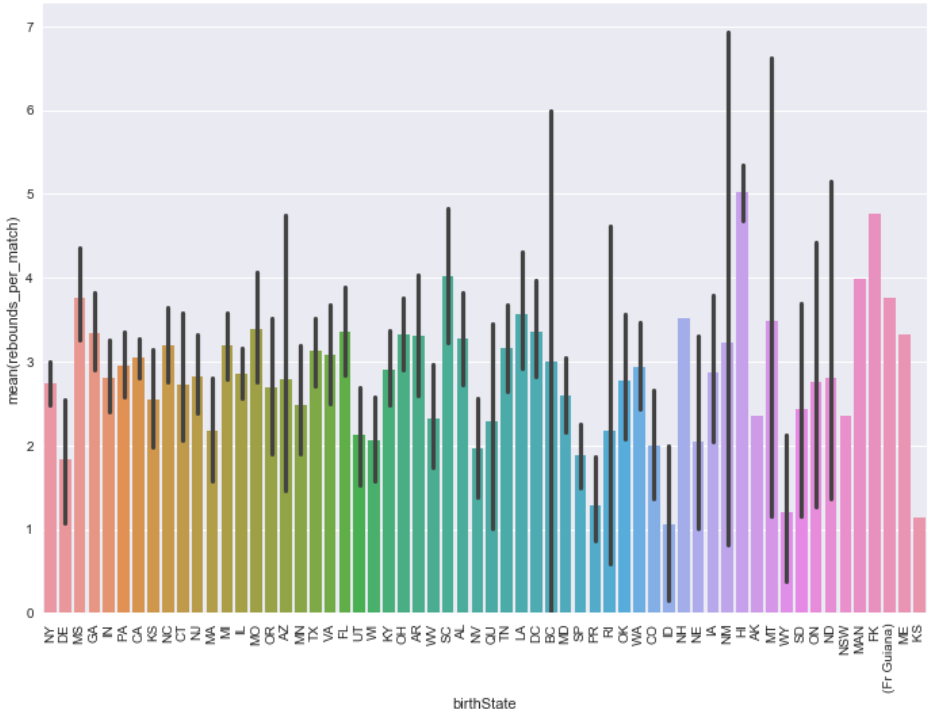


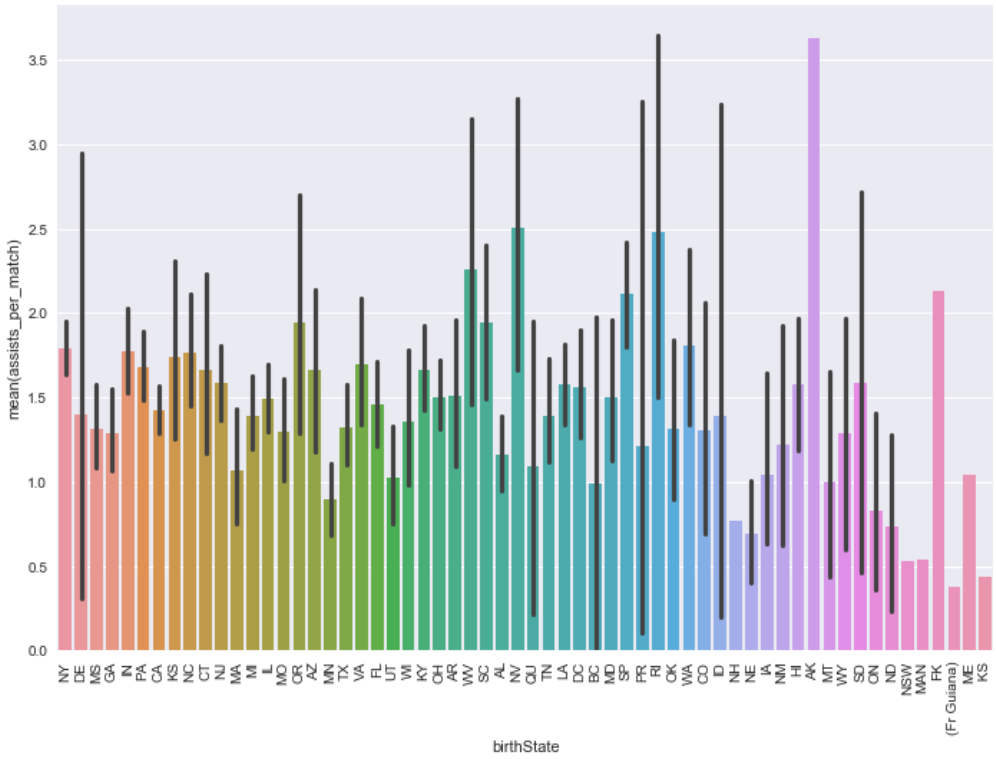


Brazilian are also good in assistant point per match and steals per match.

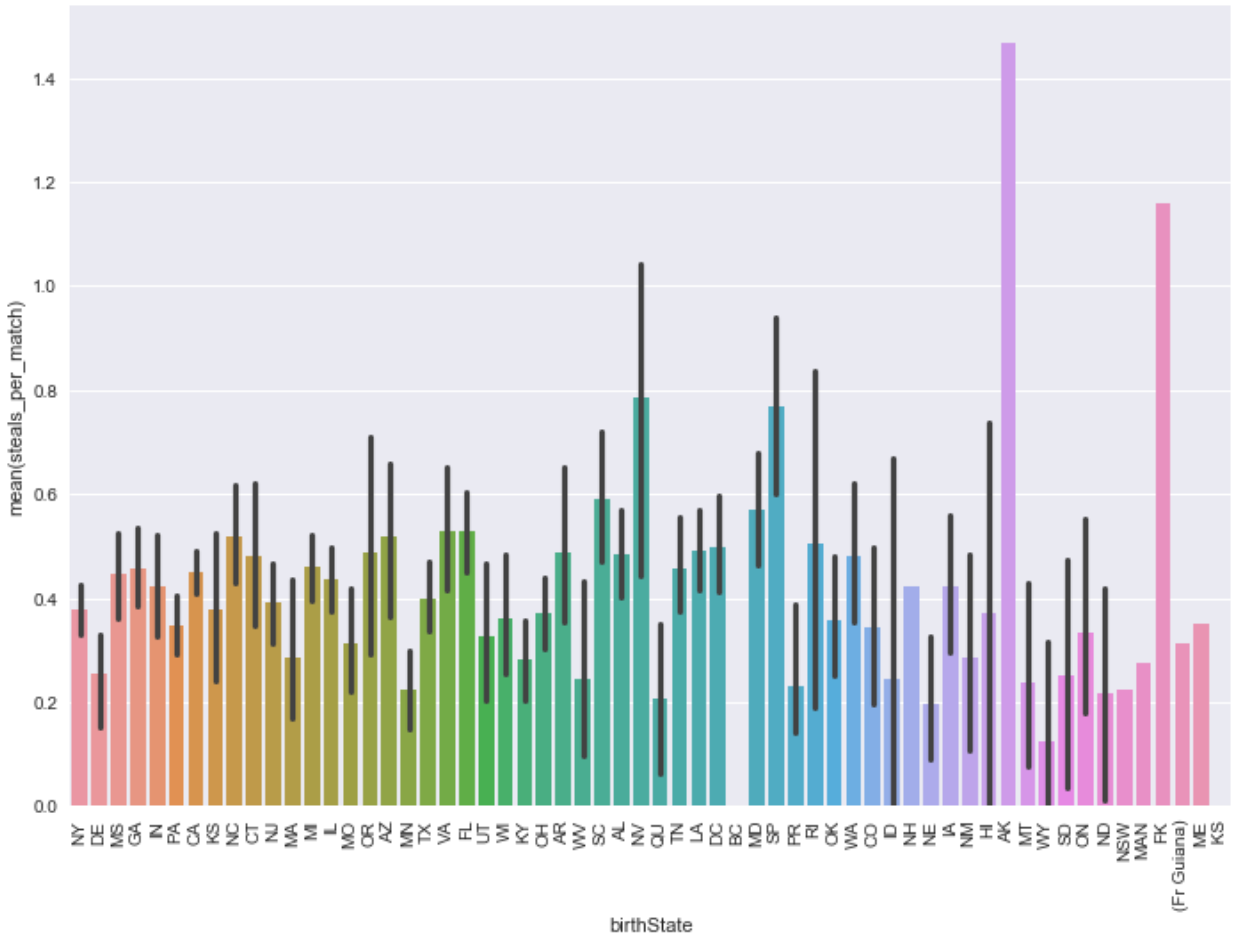
Let’s do the same statistics on the birth city.







There is nothing so interesting to find out by birth city but Brazilian have the best record in the NBA.



Part3 -Question 3

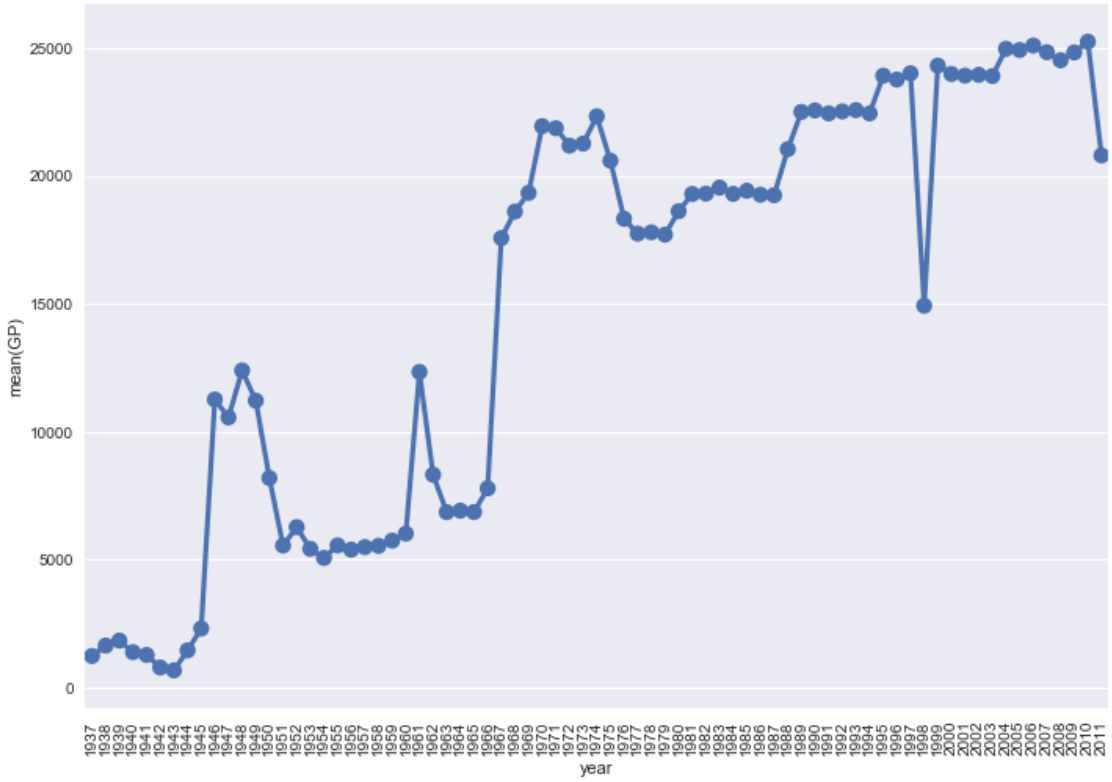
Question: - Find something else in dataset that you find interesting.

I want to see how playing style and other things changes over time

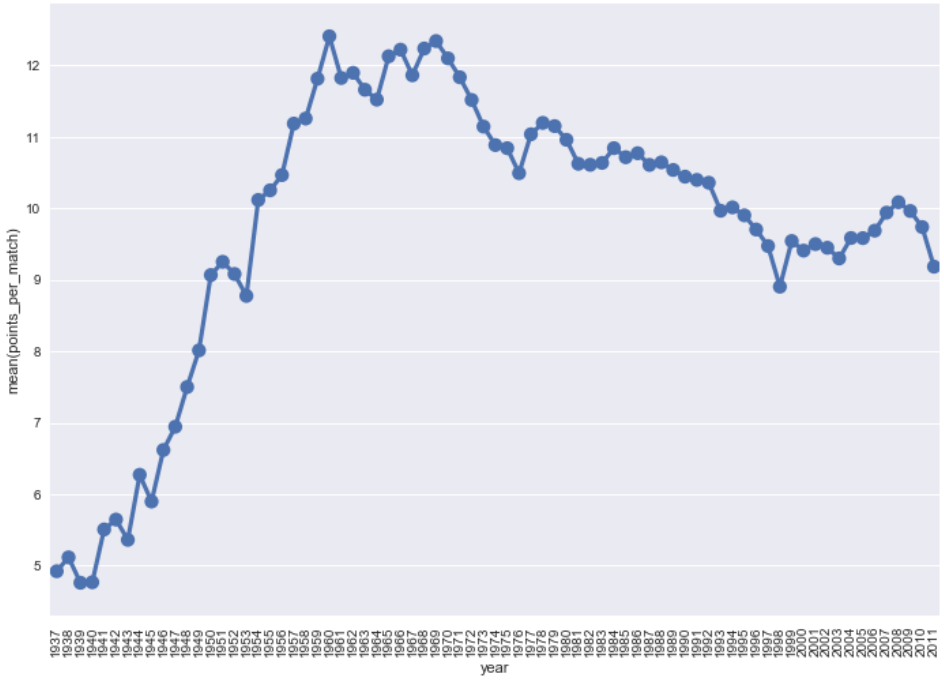
Snippet of the program

These snippets are in reverse order (last part of program is first snippet and so on).after plotting all the graph I found game is improving and being faster and faster year by year.

I found out 2 types of pattern one which is trending up slowly and others which suddenly grows.

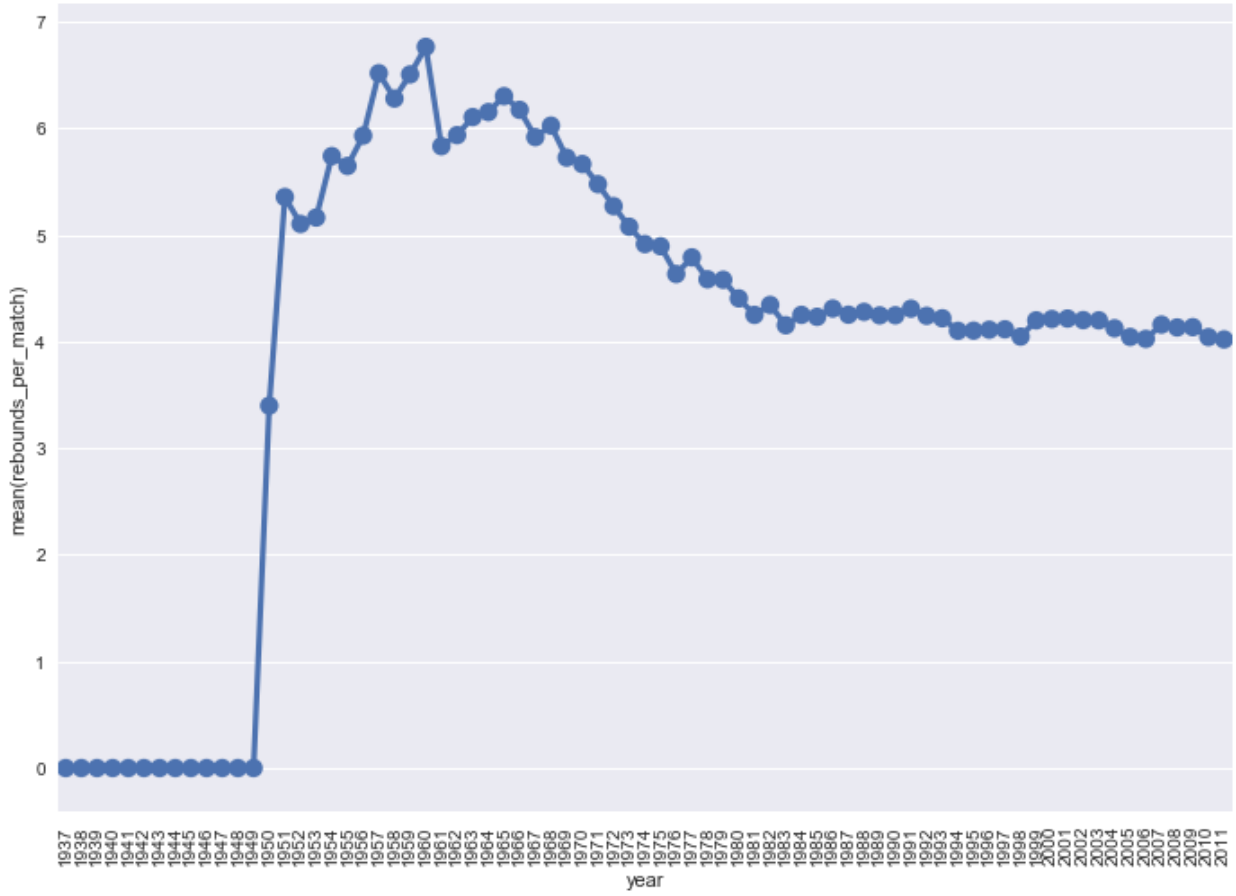


See how number of games played every year increase. In 1937 around 1000 games played every year and now it is 25000.

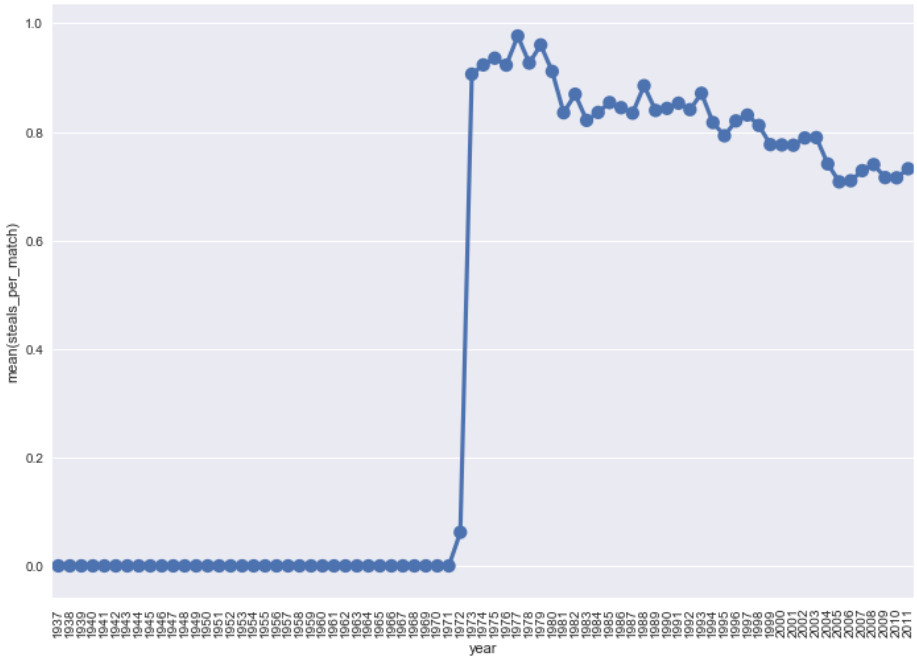


Points taken per match is also increasing with increase in time.

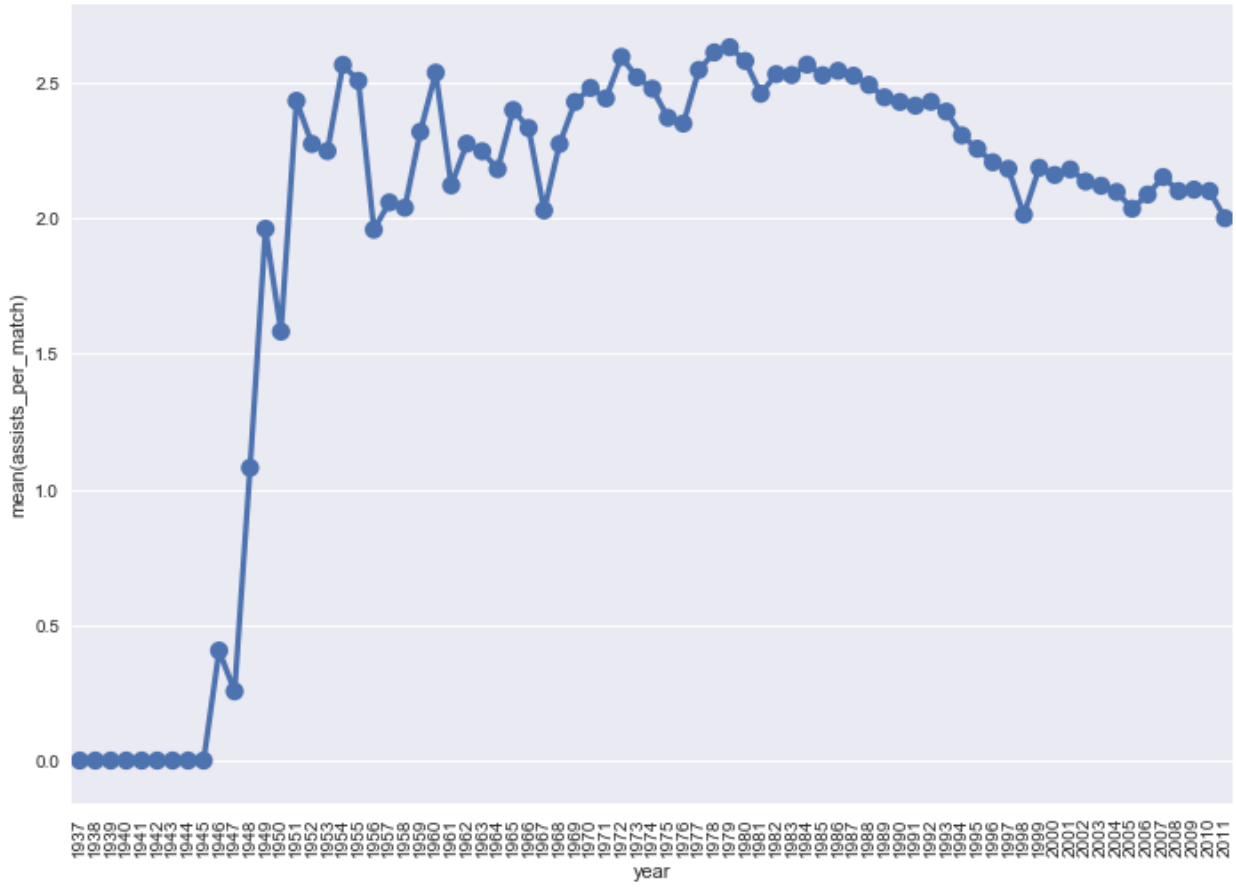
in



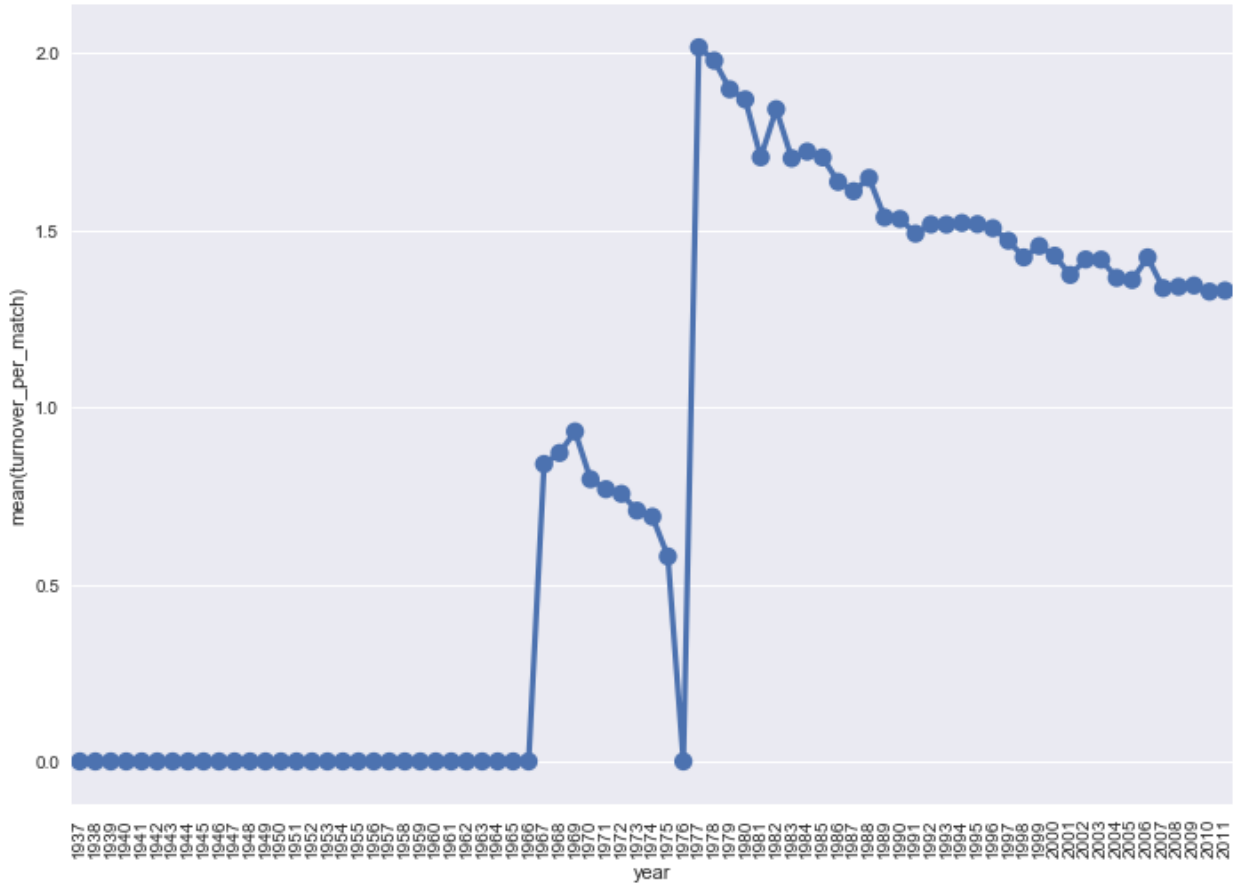
Till 1949 nobody uses rebound and in 1950 it suddenly becomes in trend



People do not used to steal before 1972



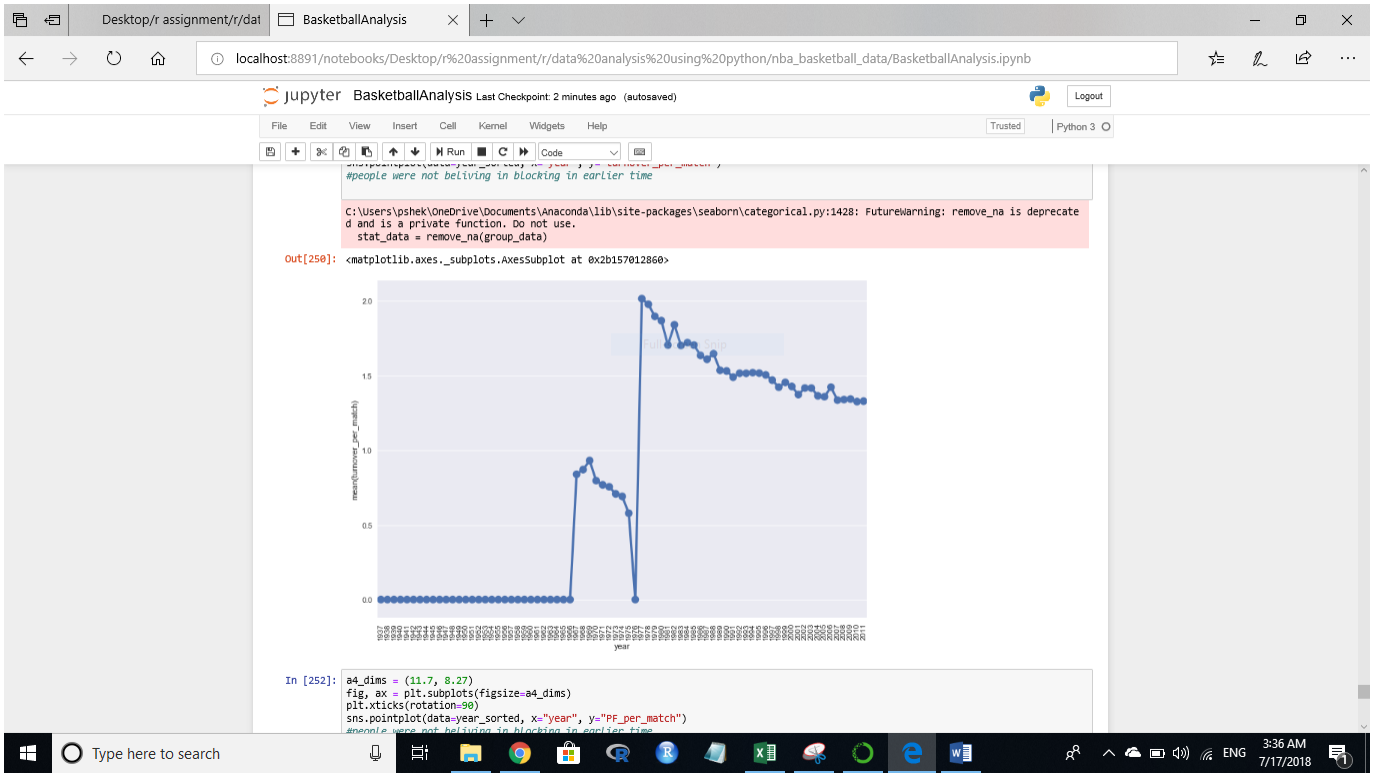
Till 1945 no players believes in assists .



Till 1945 no players believes in turnover.

Screenshots of the program

All the screen shots is in reverse order of pgm(last part of pgm at first page)



Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: a minute ago (autosaved)

Logout

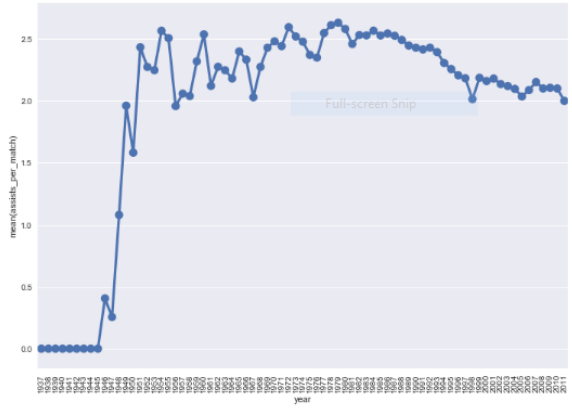
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[244]: <matplotlib.axes._subplots.AxesSubplot at 0x2b156b8f898>



In [250]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.pointplot(data=year_sorted, x="year", y="turnover_per_match")
#people were not beliving in blocking in earlier time
```

Type here to search

ENG 3:36 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: a minute ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

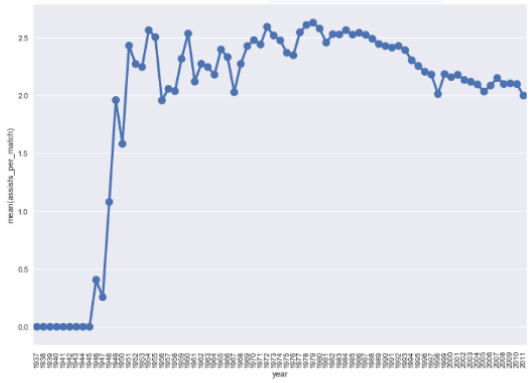
Trusted | Python 3

Run | Code

In [239]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.pointplot(data=year_sorted, x="year", y="assists_per_match")
#people were not beliving in assisting in earlier time
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[239]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1549348d0>



Type here to search

ENG 3:36 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: a minute ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.pointplot(data=year_sorted, x="year", y="steals_per_match")
#no stealing till 1972
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[238]: <matplotlib.axes._subplots.AxesSubplot at 0x2b15506ac50>

In [239]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
```

Type here to search ENG 3:35 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: a few seconds ago (autosaved) Logout

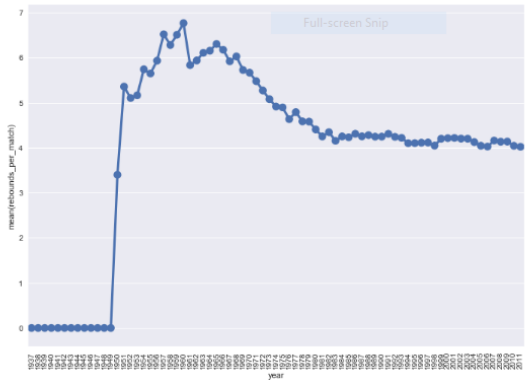
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [237]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.pointplot(data=year_sorted, x="year", y="rebounds_per_match")
#we are not seeing any rebound till 1949 and then a dragistic growth in number of rebounds in 1950
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[237]: <matplotlib.axes._subplots.AxesSubplot at 0x2b154e82710>



In [238]:
```
a4_dims = (11.7, 8.27)
```

Type here to search ENG 3:35 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: a few seconds ago (autosaved) Logout

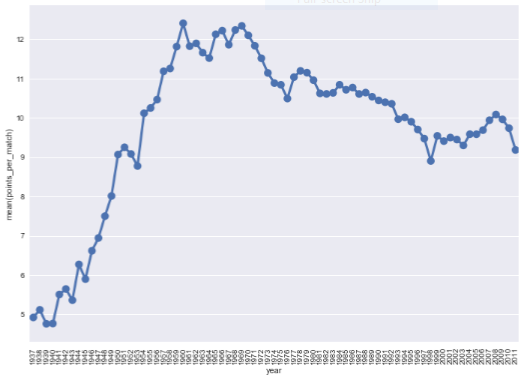
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [236]:
```
#as we can see how frequency in the number of games played is trending by time
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.pointplot(data=year_sorted, x="year", y="points_per_match")
#we are seeing trend
#it seems like  people do not score more points in ayear
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[236]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1549deda0>



Type here to search ENG 3:35 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 7 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

23751 rows × 42 columns

In [249]:
```
#part 3 -3rd question
#main motive of this is
#i want to see how playing style and other things changes over time


year_sorted=players.groupby("year").sum()
year_sorted=year_sorted.reset_index()
year_sorted['points_per_match']=year_sorted["points"] / year_sorted["GP"]
year_sorted['rebounds_per_match']=year_sorted["rebounds"] / year_sorted["GP"]
year_sorted['steals_per_match']=year_sorted["steals"] / year_sorted["GP"]
year_sorted['assists_per_match']=year_sorted["assists"] / year_sorted["GP"]
year_sorted['blocks_per_match']=year_sorted["blocks"] / year_sorted["GP"]
year_sorted['turnover_per_match']=year_sorted["turnovers"] / year_sorted["GP"]
year_sorted['PF_per_match']=year_sorted["PF"] / year_sorted["GP"]
```

In [235]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.pointplot(data=year_sorted, x="year", y="GP")
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[235]: <matplotlib.axes._subplots.AxesSubplot at 0x2b154536c18>

Type here to search ENG 3:34 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 6 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

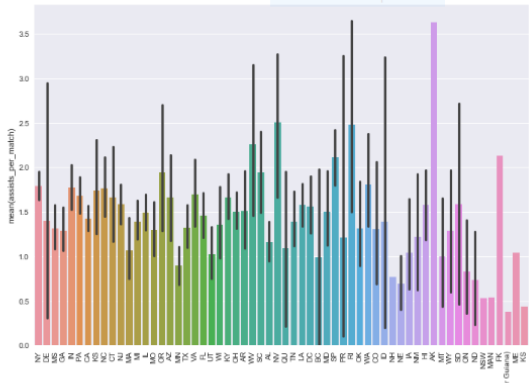
Run | Code

birthState

In [222]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=country, x="birthState", y="assists_per_match")
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[222]: <matplotlib.axes._subplots.AxesSubplot at 0x2b15369cd68>



Type here to search | ENG 3:33 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 5 minutes ago (autosaved)

Logout

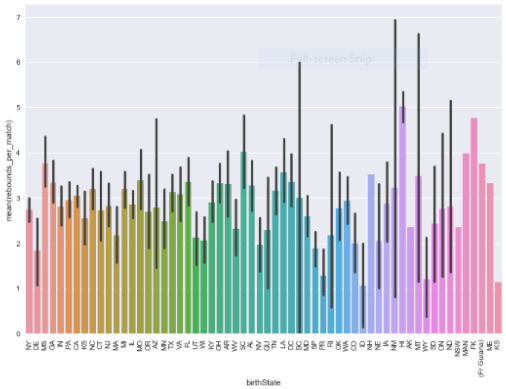
File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

Run | Code

```
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=country, x="birthState", y="rebounds_per_match")
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[220]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1532452b0>



In [222]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
```

Type here to search | ENG 3:32 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 5 minutes ago (autosaved)

Logout

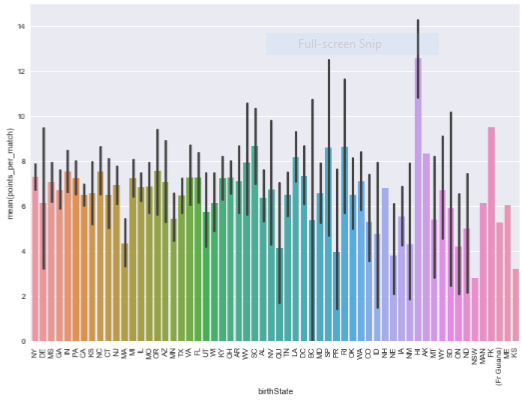
File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

Run | Code

In [219]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=country, x="birthState", y="points_per_match")
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[219]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1529b0588>



In [220]:
```
a4_dims = (11.7, 8.27)
```

Type here to search | ENG 3:32 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 4 minutes ago (autosaved)

Logout

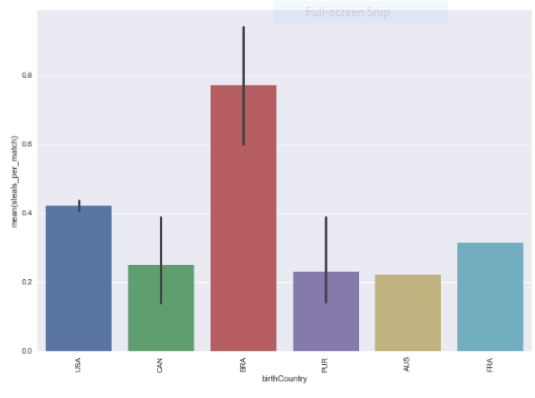
File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

Run | Code

In [218]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=country, x="birthCountry", y="steals_per_match")
#again brazil
#one thing here to notice is brazalian are the top players of nba
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[218]: <matplotlib.axes._subplots.AxesSubplot at 0x2b152cc4eb8>



Type here to search | ENG 3:31 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 4 minutes ago (autosaved) Logout

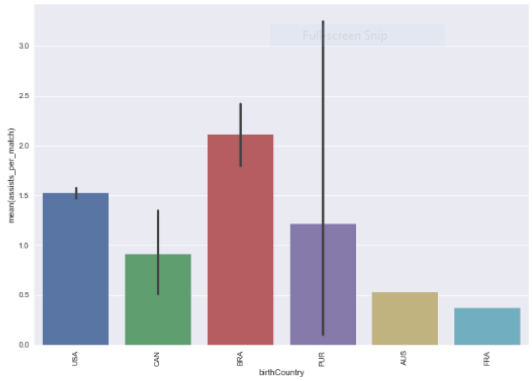
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [217]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=country, x="birthCountry", y="assists_per_match")
#people born in brazilare good in assist point
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[217]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1528cd748>



In [218]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
```

Type here to search ENG 3:31 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 4 minutes ago (autosaved) Logout

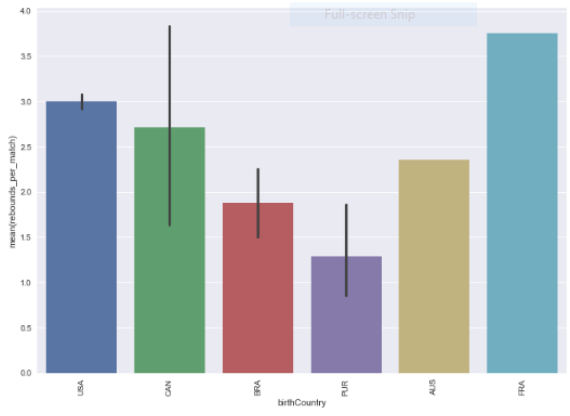
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [216]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=country, x="birthCountry", y="rebounds_per_match")
# by looking at the graph we can say that brazilian scores more point but rebounding capability of brazalian players is at the 2n
#people born in france is has good rebounding capability
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[216]: <matplotlib.axes._subplots.AxesSubplot at 0x2b152844080>



Type here to search ENG 3:31 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 2 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

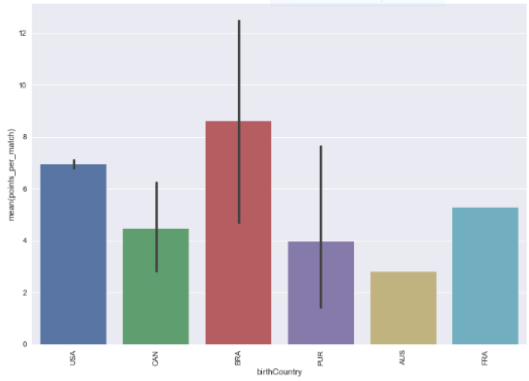
Run Code

```
#people born in autrailia are the players who scores less than other countries

country=country.dropna()
country=country.reset_index()
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=country, x="birthCountry", y="points_per_match")
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecated and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[215]: <matplotlib.axes._subplots.AxesSubplot at 0x2b151fd17b8>



Type here to search ENG 3:29 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: a minute ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

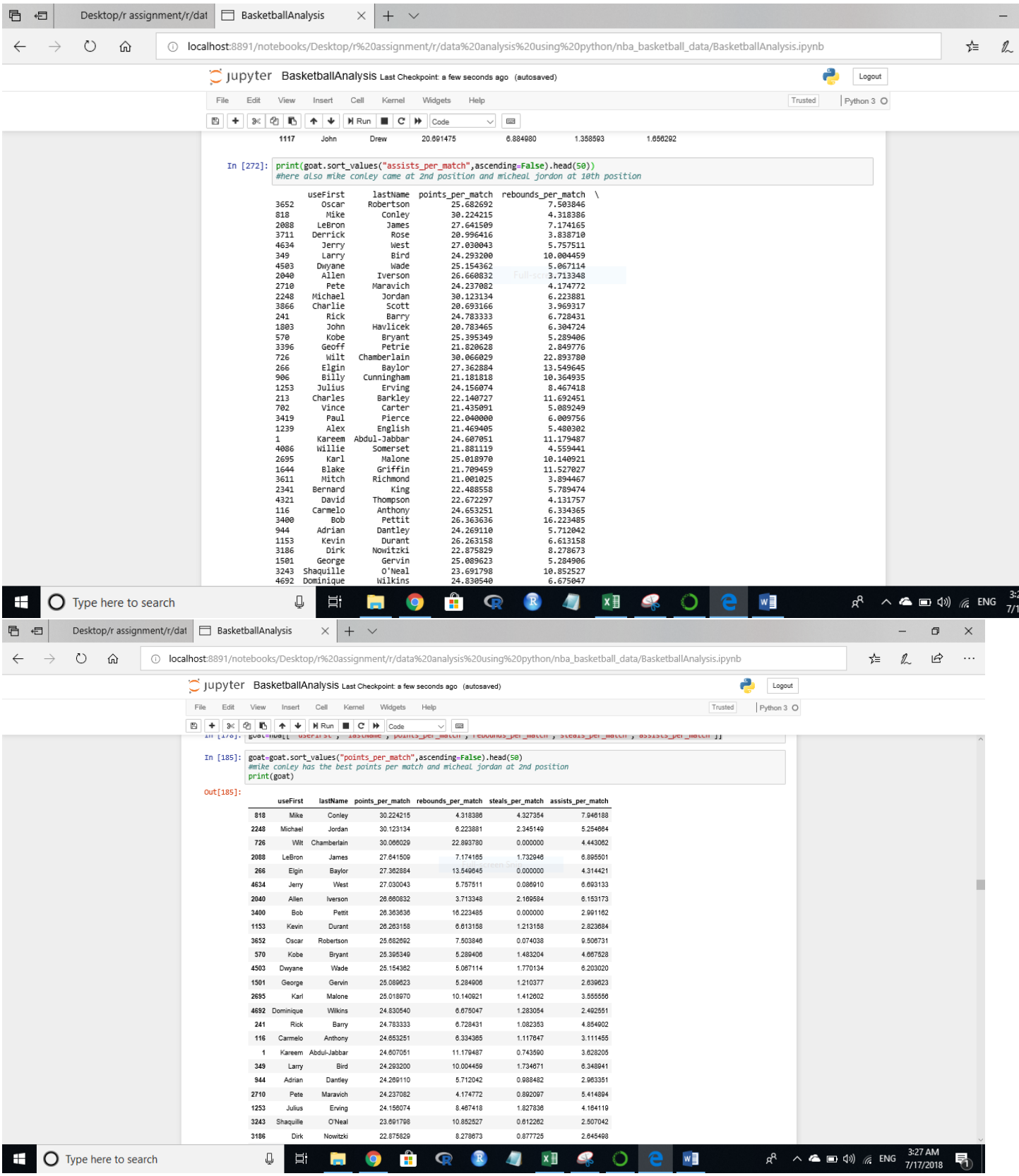
Run Code

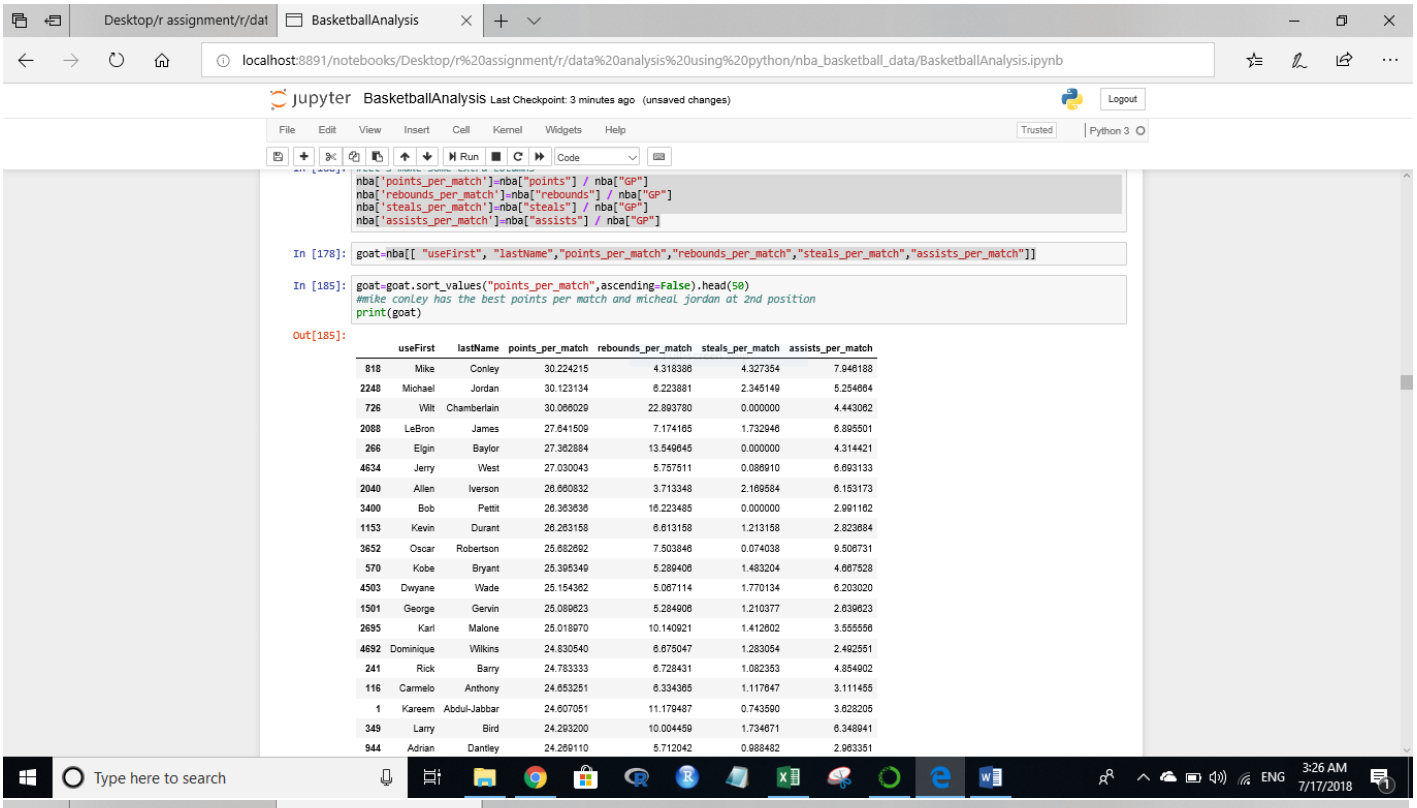
In [186]:
```
print(goat.sort_values("steals_per_match",ascending=False).head(50))
#here also mike conley came at 1st position
#and micheal jordon at 2nd position
```

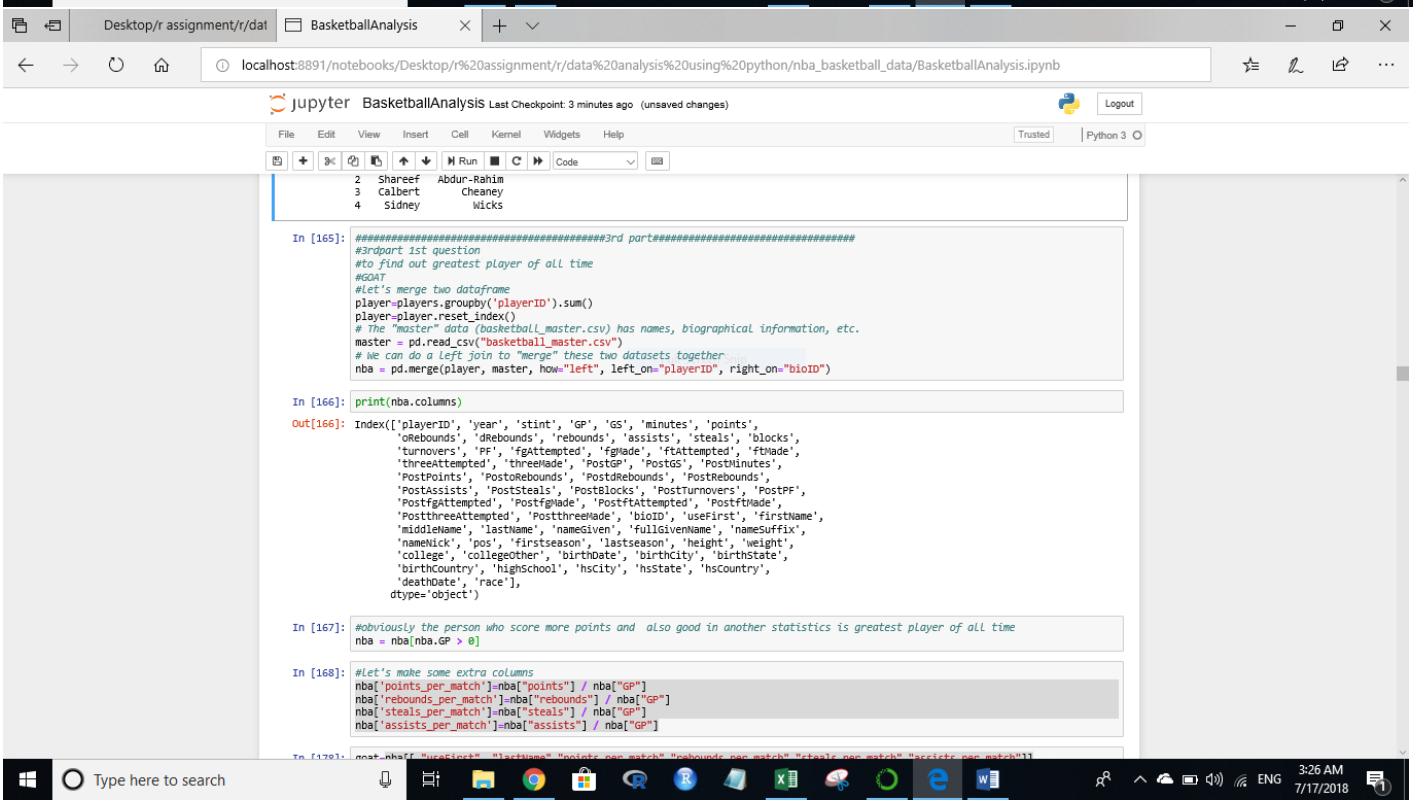
Out[186]:

	useFirst	lastName	points_per_match	rebounds_per_match	steals_per_match	assists_per_match
818	Mike	Conley	30.224215	4.318386	4.327354	7.946188
2248	Michael	Jordan	30.123134	6.223881	2.345149	5.254684
2040	Allen	Iverson	26.660832	3.713348	2.169584	6.153173
1253	Julius	Erving	24.156074	8.467418	1.827836	4.164119
4503	Dwyane	Wade	25.154362	5.067114	1.770134	6.203020
3225	Hakeem	Olajuwon	21.765751	11.104200	1.746365	2.470113
349	Larry	Bird	24.293200	10.004459	1.734671	6.348941
2088	LeBron	James	27.641509	7.174165	1.732946	6.895501
213	Charles	Barkley	22.140727	11.692451	1.535881	3.928239
570	Kobe	Bryant	25.395349	5.289406	1.483204	4.667528
3419	Paul	Pierce	22.040000	6.009756	1.464390	3.839024
2695	Karl	Malone	25.018970	10.140921	1.412602	3.555556
3665	David	Robinson	21.063830	10.635258	1.406282	2.473151
1117	John	Drew	20.691475	6.884980	1.358593	1.656292
4692	Dominique	Wilkins	24.830540	6.675047	1.283054	2.492551
3611	Mitch	Richmond	21.001025	3.894467	1.240779	3.481557
1153	Kevin	Durant	26.263158	6.613158	1.213158	2.823684
1501	George	Gervin	25.089623	5.284906	1.210377	2.639623
702	Vince	Carter	21.435091	5.089249	1.161258	3.890467
116	Carmelo	Anthony	24.653251	6.334365	1.117647	3.111455
241	Rick	Barry	24.783333	6.728431	1.082353	4.854902
4321	David	Thompson	22.672297	4.131757	1.005068	3.275338
2341	Bernard	King	22.488558	5.789474	0.990847	3.275744

Type here to search ENG 3:28 AM 7/17/2018


Desktop/r assignment/r/dat
BasketballAnalysis
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb
jupyter BasketballAnalysis Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
1117 John Drew 20.691475 6.884980 1.358593 1.656292
In [272]:
print(goat.sort_values("assists_per_match",ascending=False).head(50))
#here also mike conley came at 2nd position and micheal jordon at 10th position
useFirst lastName points_per_match rebounds_per_match \
3652 Oscar Robertson 25.682692 7.503846
818 Mike Conley 30.224215 4.318386
2088 LeBron James 27.641509 7.174165
3711 Derrick Rose 20.996416 3.838710
4634 Jerry West 27.030043 5.757511
349 Larry Bird 24.293200 10.004459
4503 Dwyane Wade 25.154362 5.067114
2040 Allen Iverson 26.660832 3.713348
2710 Pete Maravich 24.237082 4.174772
2248 Michael Jordan 30.123134 6.223881
3866 Charlie Scott 20.693166 3.969317
241 Rick Barry 24.783333 6.728431
1803 John Havlicek 20.783465 6.304724
570 Kobe Bryant 25.395349 5.289406
3396 Geoff Petrie 21.820628 2.849776
726 Wilt Chamberlain 30.066029 22.893780
266 Elgin Baylor 27.362884 13.549645
906 Billy Cunningham 21.181818 10.364935
1253 Julius Erving 24.156074 8.467418
213 Charles Barkley 22.140727 11.692451
702 Vince Carter 21.435091 5.089249
3419 Paul Pierce 22.040000 6.009756
1239 Alex English 21.469405 5.480302
1 Kareem Abdul-Jabbar 24.607051 11.179487
4086 Willie Somerset 21.881119 4.559441
2695 Karl Malone 25.018970 10.140921
1644 Blake Griffin 21.709459 11.527027
3611 Mitch Richmond 21.001025 3.894467
2341 Bernard King 22.488558 5.789474
4321 David Thompson 22.672297 4.131757
116 Carmelo Anthony 24.653251 6.334365
3400 Bob Pettit 26.363636 16.223485
944 Adrian Dantley 24.269110 5.712042
1153 Kevin Durant 26.263158 6.613158
3186 Dirk Nowitzki 22.875829 8.278673
1501 George Gervin 25.089623 5.284906
3243 Shaquille O'Neal 23.691798 10.852527
4692 Dominique Wilkins 24.830540 6.675047
Type here to search
ENG
3:2
7/1
Desktop/r assignment/r/dat
BasketballAnalysis
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb
jupyter BasketballAnalysis Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [185]:
goat=goat.sort_values("points_per_match",ascending=False).head(50)
#mike conley has the best points per match and micheal jordan at 2nd position
print(goat)
Out[185]:
useFirst lastName points_per_match rebounds_per_match steals_per_match assists_per_match
818 Mike Conley 30.224215 4.318386 4.327354 7.946188
2248 Michael Jordan 30.123134 6.223881 2.345149 5.254664
726 Wilt Chamberlain 30.066029 22.893780 0.000000 4.443062
2088 LeBron James 27.641509 7.174165 1.732946 6.895501
266 Elgin Baylor 27.362884 13.549645 0.000000 4.314421
4634 Jerry West 27.030043 5.757511 0.086910 6.693133
2040 Allen Iverson 26.660832 3.713348 2.169584 6.153173
3400 Bob Pettit 26.363636 16.223485 0.000000 2.991162
1153 Kevin Durant 26.263158 6.613158 1.213158 2.823684
3652 Oscar Robertson 25.682692 7.503846 0.074038 9.506731
570 Kobe Bryant 25.395349 5.289406 1.483204 4.667528
4503 Dwyane Wade 25.154362 5.067114 1.770134 6.203020
1501 George Gervin 25.089623 5.284906 1.210377 2.839623
2695 Karl Malone 25.018970 10.140921 1.412602 3.555556
4692 Dominique Wilkins 24.830540 6.675047 1.283054 2.492551
241 Rick Barry 24.783333 6.728431 1.082353 4.854902
116 Carmelo Anthony 24.653251 6.334365 1.117647 3.111455
1 Kareem Abdul-Jabbar 24.607051 11.179487 0.743590 3.628205
349 Larry Bird 24.293200 10.004459 1.734671 6.348941
944 Adrian Dantley 24.269110 5.712042 0.988482 2.963351
2710 Pete Maravich 24.237082 4.174772 0.892097 5.414894
1253 Julius Erving 24.156074 8.467418 1.827836 4.164119
3243 Shaquille O'Neal 23.691798 10.852527 0.612262 2.507042
3186 Dirk Nowitzki 22.875829 8.278673 0.877725 2.645498
Type here to search
ENG
3:27 AM
7/17/2018


Desktop/r assignment/r/dat
BasketballAnalysis
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb
jupyter BasketballAnalysis Last Checkpoint: 3 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
nba['points_per_match']=nba["points"] / nba["GP"]
nba['rebounds_per_match']=nba["rebounds"] / nba["GP"]
nba['steals_per_match']=nba["steals"] / nba["GP"]
nba['assists_per_match']=nba["assists"] / nba["GP"]
In [178]: goat=nba[["useFirst", "lastName","points_per_match","rebounds_per_match","steals_per_match","assists_per_match"]]
In [185]: goat=goat.sort_values("points_per_match",ascending=False).head(50)
#mike conley has the best points per match and micheal jordan at 2nd position
print(goat)
Out[185]:
useFirst lastName points_per_match rebounds_per_match steals_per_match assists_per_match
818 Mike Conley 30.224215 4.318386 4.327354 7.946188
2248 Michael Jordan 30.123134 6.223881 2.345149 5.254664
726 Wilt Chamberlain 30.066029 22.893780 0.000000 4.443062
2088 LeBron James 27.641509 7.174165 1.732946 6.895501
266 Elgin Baylor 27.362884 13.549645 0.000000 4.314421
4634 Jerry West 27.030043 5.757511 0.086910 6.693133
2040 Allen Iverson 26.660832 3.713348 2.169584 6.153173
3400 Bob Pettit 26.363636 16.223485 0.000000 2.991162
1153 Kevin Durant 26.263158 6.613158 1.213158 2.823684
3652 Oscar Robertson 25.682692 7.503846 0.074038 9.506731
570 Kobe Bryant 25.395349 5.289406 1.483204 4.667528
4503 Dwyane Wade 25.154362 5.067114 1.770134 6.203020
1501 George Gervin 25.089623 5.284906 1.210377 2.639623
2695 Karl Malone 25.018970 10.140921 1.412602 3.555556
4692 Dominique Wilkins 24.830540 6.675047 1.283054 2.492551
241 Rick Barry 24.783333 6.728431 1.082353 4.854902
116 Carmelo Anthony 24.653251 6.334365 1.117647 3.111455
1 Kareem Abdul-Jabbar 24.607051 11.179487 0.743590 3.628205
349 Larry Bird 24.293200 10.004459 1.734671 6.348941
944 Adrian Dantley 24.269110 5.712042 0.988482 2.963351
Type here to search
ENG
3:26 AM
7/17/2018


Desktop/r assignment/r/dat
BasketballAnalysis
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb
jupyter BasketballAnalysis Last Checkpoint: 3 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
2 Shareef Abdur-Rahim
3 Calbert Cheaney
4 Sidney Wicks
In [165]: ##3rd part#################################
#3rdpart 1st question
#to find out greatest player of all time
#GOAT
#Let's merge two dataframe
player=players.groupby('playerID').sum()
player=player.reset_index()
# The "master" data (basketball_master.csv) has names, biographical information, etc.
master = pd.read_csv("basketball_master.csv")
# We can do a left join to "merge" these two datasets together
nba = pd.merge(player, master, how="left", left_on="playerID", right_on="bioID")
In [166]: print(nba.columns)
Out[166]: Index(['playerID', 'year', 'stint', 'GP', 'GS', 'minutes', 'points',
'oRebounds', 'dRebounds', 'rebounds', 'assists', 'steals', 'blocks',
'turnovers', 'PF', 'fgAttempted', 'fgMade', 'ftAttempted', 'ftMade',
'threeAttempted', 'threeMade', 'PostGP', 'PostGS', 'PostMinutes',
'PostPoints', 'PostoRebounds', 'PostdRebounds', 'PostRebounds',
'PostAssists', 'PostSteals', 'PostBlocks', 'PostTurnovers', 'PostPF',
'PostfgAttempted', 'PostfgMade', 'PostftAttempted', 'PostftMade',
'PostthreeAttempted', 'PostthreeMade', 'bioID', 'useFirst', 'firstName',
'middleName', 'lastName', 'nameGiven', 'fullGivenName', 'nameSuffix',
'nameNick', 'pos', 'firstseason', 'lastseason', 'height', 'weight',
'college', 'collegeOther', 'birthDate', 'birthCity', 'birthState',
'birthCountry', 'highSchool', 'hsCity', 'hsState', 'hsCountry',
'deathDate', 'race'],
dtype='object')
In [167]: #obviously the person who score more points and also good in another statistics is greatest player of all time
nba = nba[nba.GP > 0]
In [168]: #Let's make some extra columns
nba['points_per_match']=nba["points"] / nba["GP"]
nba['rebounds_per_match']=nba["rebounds"] / nba["GP"]
nba['steals_per_match']=nba["steals"] / nba["GP"]
nba['assists_per_match']=nba["assists"] / nba["GP"]
Type here to search
ENG
3:26 AM
7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: 2 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

Run | Code

In [271]:

```
#2nd part=2nd question

#I have to find out the allrounder
# Firstly priority=points
# second priority=rebounds
# third priority =assist
#fourth proirity=steals

#since i have a dataframe group_playerid which have top 500 player who hasgood points
#next step is to find out top 200 who has good assist value
#these are the top 10 allrounders in point assisand block
top_point_rebounds=group_playerid.sort_values("rebounds",ascending=False).head(100)
top_point_rebounds_assists=top_point_rebounds.sort_values("assists",ascending=False).head(20)
top_point_rebounds_assists_steals=top_point_rebounds_assists.sort_values("steals",ascending=False).head(5)
merge = pd.merge(top_point_rebounds_assists_steals, master, how="left", left_on="playerID", right_on="bioID")
print("these are top 5 allrounders")
print(merge[["useFirst", "lastName"]])
```

```
these are top 5 allrounders
   useFirst      lastName
0  Clarence  Weatherspoon
1     Steve       Francis
2   Shareef   Abdur-Rahim
3   Calbert       Cheaney
4    Sidney         Wicks
```

In [165]:

```
#########################################3rd part#################################
#3rdpart 1st question
#to find out greatest player of all time
#GOAT
#Let's merge two dataframe
player=players.groupby('playerID').sum()
player=player.reset_index()
# The "master" data (basketball_master.csv) has names, biographical information, etc.
master = pd.read_csv("basketball_master.csv")
# We can do a left join to "merge" these two datasets together
nba = pd.merge(player, master, how="left", left_on="playerID", right_on="bioID")
```

In [166]:

```
print(nba.columns)
```

Out[166]: Index(['playerID', 'year', 'stint', 'GP', 'GS', 'minutes', 'points'

Type here to search | ENG 3:25 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter BasketballAnalysis Last Checkpoint: a few seconds ago (autosaved)

Logout

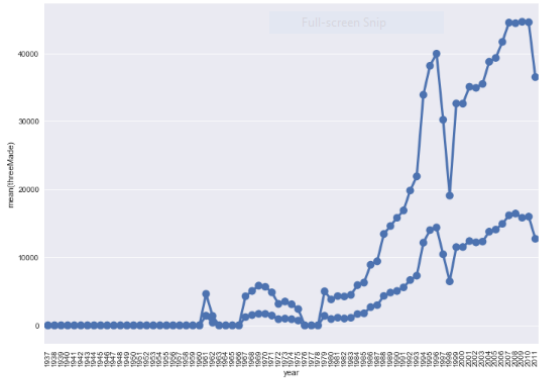
File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

Run | Code

```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.pointplot(data=years_group, x="year", y="threeAttempted")
sns.pointplot(data=years_group, x="year", y="threeMade")
#yes we are seeing a trend in attempt of three point across the league after 1978
```

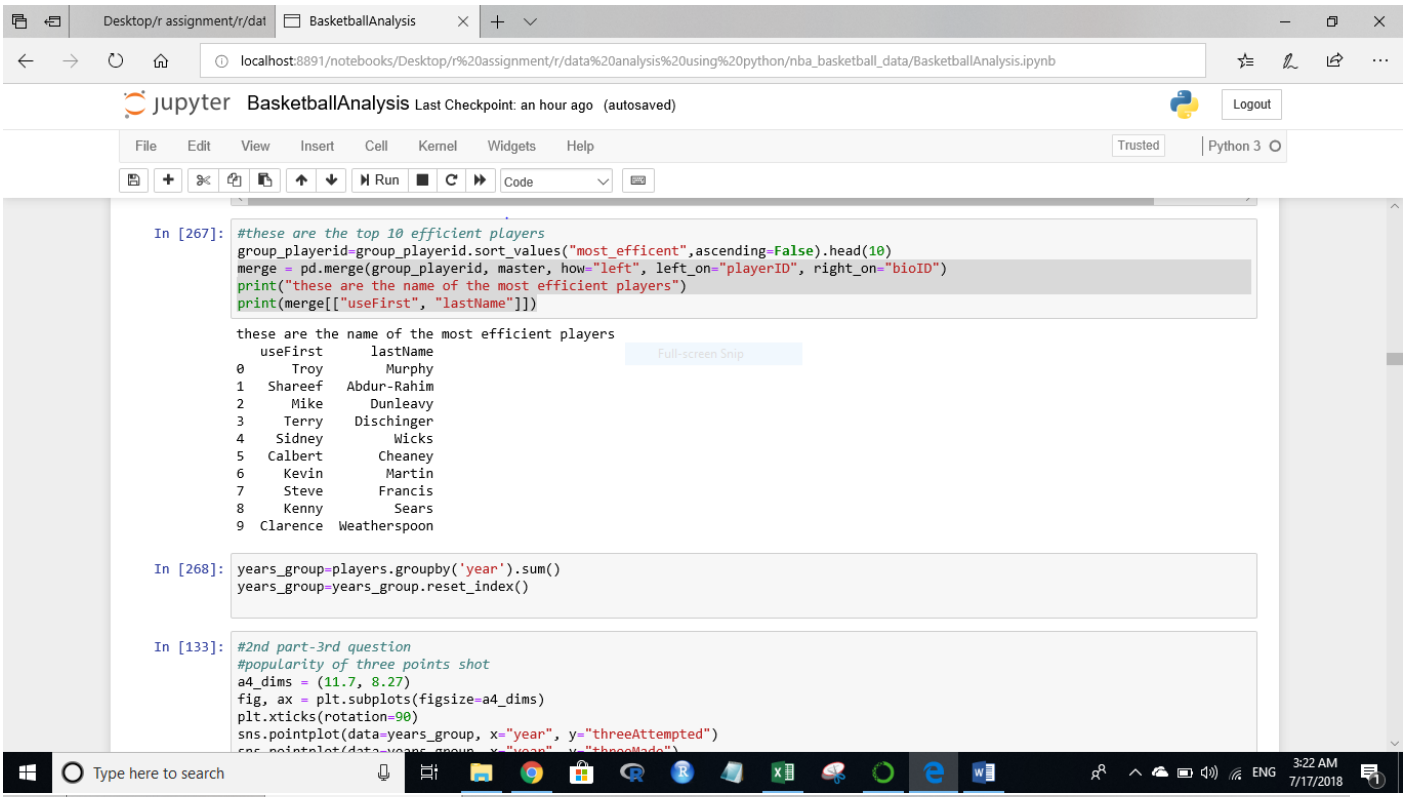
```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecated and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

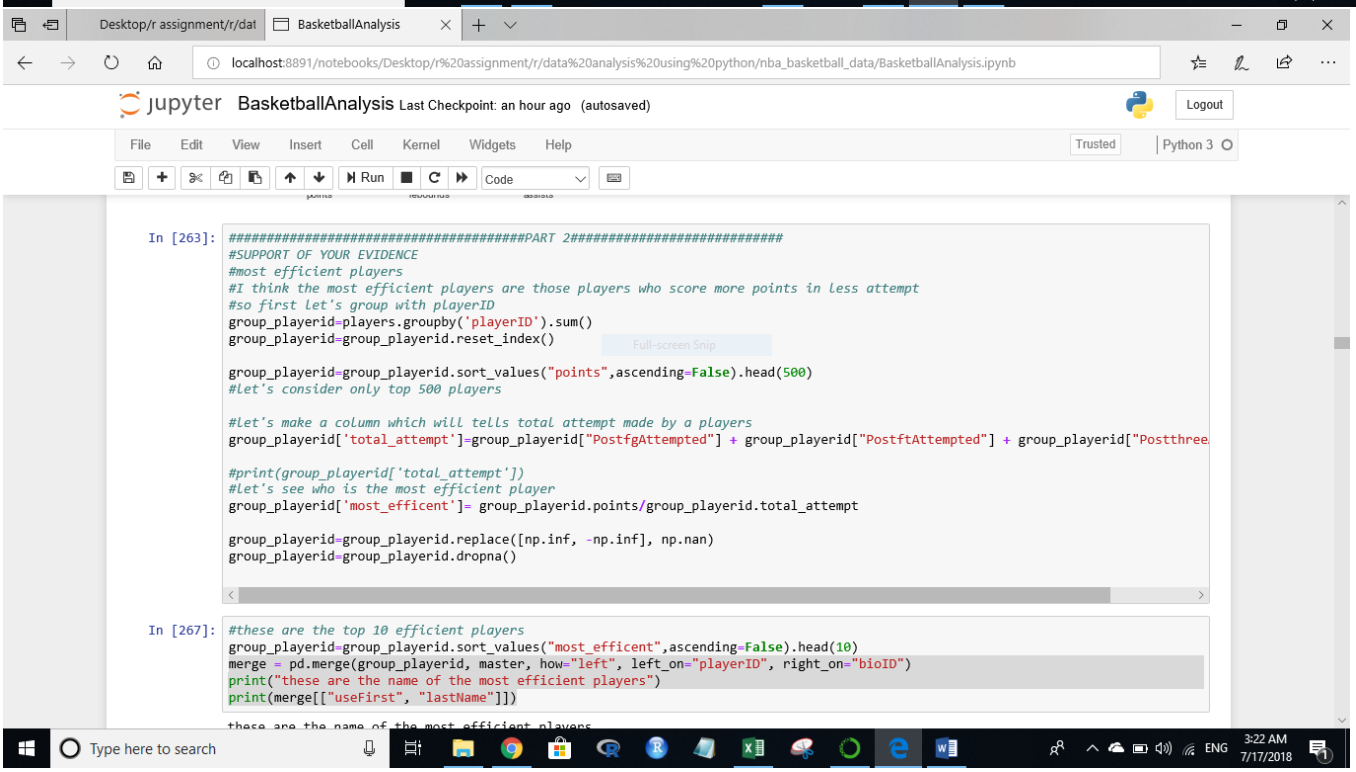
Out[133]: <matplotlib.axes._subplots.AxesSubplot at 0x2b14a8e37b8>



In [271]:

Type here to search | ENG 3:23 AM 7/17/2018


Desktop/r assignment/r/dat
BasketballAnalysis
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb
jupyter BasketballAnalysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [267]:
#these are the top 10 efficient players
group_playerid=group_playerid.sort_values("most_efficent",ascending=False).head(10)
merge = pd.merge(group_playerid, master, how="left", left_on="playerID", right_on="bioID")
print("these are the name of the most efficient players")
print(merge[["useFirst", "lastName"]])
these are the name of the most efficient players
useFirst lastName
0 Troy Murphy
1 Shareef Abdur-Rahim
2 Mike Dunleavy
3 Terry Dischinger
4 Sidney Wicks
5 Calbert Cheaney
6 Kevin Martin
7 Steve Francis
8 Kenny Sears
9 Clarence Weatherspoon
In [268]:
years_group=players.groupby('year').sum()
years_group=years_group.reset_index()
In [133]:
#2nd part-3rd question
#popularity of three points shot
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.pointplot(data=years_group, x="year", y="threeAttempted")
Type here to search
ENG
3:22 AM
7/17/2018


Desktop/r assignment/r/dat
BasketballAnalysis
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb
jupyter BasketballAnalysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [263]:
##PART 2#########################
#SUPPORT OF YOUR EVIDENCE
#most efficient players
#I think the most efficient players are those players who score more points in less attempt
#so first let's group with playerID
group_playerid=players.groupby('playerID').sum()
group_playerid=group_playerid.reset_index()
group_playerid=group_playerid.sort_values("points",ascending=False).head(500)
#Let's consider only top 500 players
#Let's make a column which will tells total attempt made by a players
group_playerid['total_attempt']=group_playerid["PostfgAttempted"] + group_playerid["PostftAttempted"] + group_playerid["Postthree
#print(group_playerid['total_attempt'])
#Let's see who is the most efficient player
group_playerid['most_efficent']= group_playerid.points/group_playerid.total_attempt
group_playerid=group_playerid.replace([np.inf, -np.inf], np.nan)
group_playerid=group_playerid.dropna()
In [267]:
#these are the top 10 efficient players
group_playerid=group_playerid.sort_values("most_efficent",ascending=False).head(10)
merge = pd.merge(group_playerid, master, how="left", left_on="playerID", right_on="bioID")
print("these are the name of the most efficient players")
print(merge[["useFirst", "lastName"]])
Type here to search
ENG
3:22 AM
7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter **BasketballAnalysis** Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

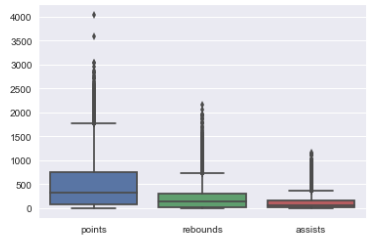
```
[403 rows x 2 columns]
```

In [78]:
```
#it is 3rd question of first part

sns.boxplot(data=players[["points", "rebounds","assists"]])
#outliers present in all the three columns
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:454: FutureWarning: remove_na is deprecated
and is a private function. Do not use.
  box_data = remove_na(group_data)
```

Out[78]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2b144ebc2e8>
```



In [263]:
```
########################################PART 2############################
#SUPPORT OF YOUR EVIDENCE
```

Type here to search ENG 3:21 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter **BasketballAnalysis** Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [70]:
```
maximum_points=sorted_total_points.head(1)
#maximum points in a year
print("maximum points scored in year",maximum_points.year.iloc[0])
print("and the maximum points is",maximum_points.points.iloc[0])
```

```
maximum points scored in year 1970
and the maximum points is 265257
```

In [259]:
```
#now let's look the player name who contributed in those points
master = pd.read_csv("basketball_master.csv")
# We can do a left join to "merge" these two datasets together
merge = pd.merge(players, master, how="left", left_on="playerID", right_on="bioID")
max_year=merge['year']==1970
merge=merge[max_year]
print("these are the name of the players")
print(merge[["useFirst", "lastName"]])
#and the player name will be
```

```
these are the name of the players
      useFirst      lastName
3887    Kareem  Abdul-Jabbar
3888     Mahdi  Abdul-Rahman
3889      Zaid    Abdul-Aziz
3890       Don         Adams
3891      Rick       Adelman
3892    Lucius         Allen
3893     Cliff      Anderson
3894     Cliff      Anderson
3895      Nate     Archibald
3896       Jim           Ard
3897       Bob        Arnzen
```

Type here to search ENG 3:20 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

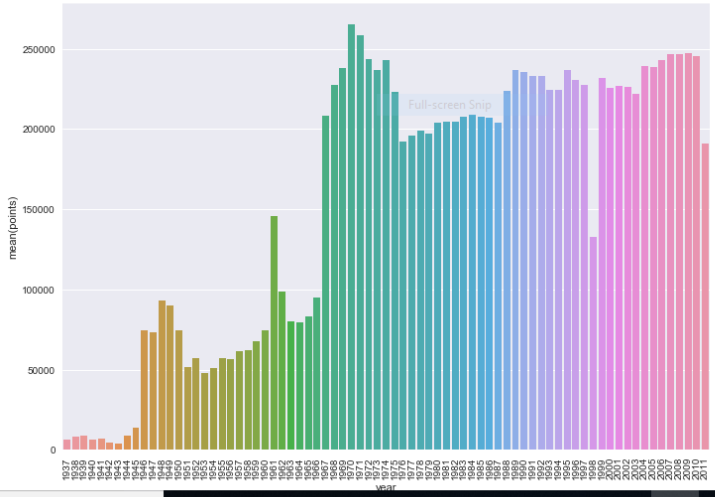
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter **BasketballAnalysis** Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Out[254]: <matplotlib.axes._subplots.AxesSubplot at 0x2b157e15198>



Type here to search ENG 3:20 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

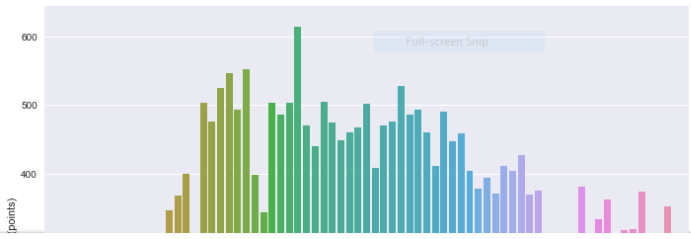
jupyter **BasketballAnalysis** Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecated and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1443fdeb8>



In [254]:
```
#highest scoring year is
total_points=players[['points','year']].groupby('year')['points'].sum()
total_points=total_points.reset_index()
sorted_total_points=total_points.sort_values("points",ascending=False)
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=sorted_total_points, x="year", y="points")
#as we can  see by the plot maximum points scored in 1970
```

Type here to search ENG 3:19 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

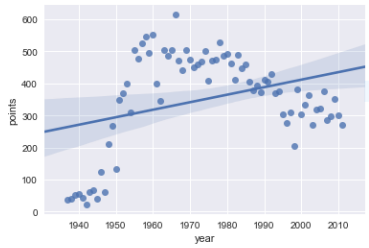
jupyter **BasketballAnalysis** Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
[75 rows x 2 columns]
```

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x2b144526a20>



In [36]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=median_grouped, x="year", y="points")
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecated and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1443fdeb8>

Type here to search ENG 3:19 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

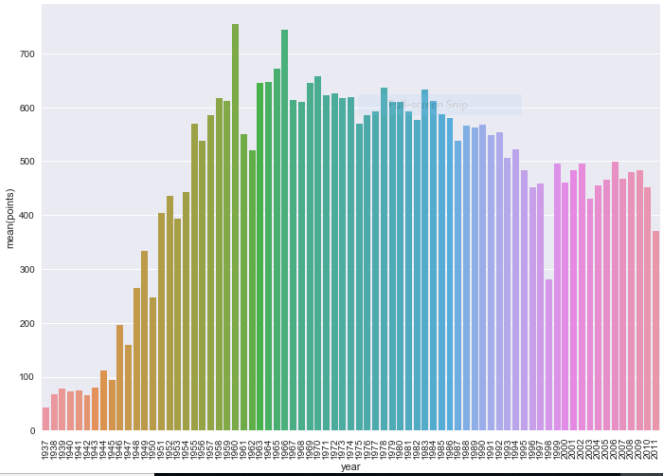
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter **BasketballAnalysis** Last Checkpoint: 44 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1433aca58>



Type here to search ENG 3:18 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter **BasketballAnalysis** Last Checkpoint: 44 minutes ago (autosaved) Logout

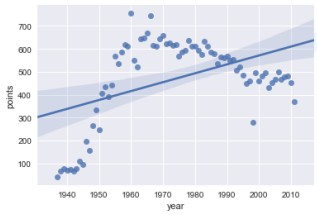
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
74  2011  369.899225

[75 rows x 2 columns]
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x2b1437aa7b8>



In [32]:
```
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
plt.xticks(rotation=90)
sns.barplot(data=mean_grouped, x="year", y="points")
```

```
C:\Users\pshek\OneDrive\Documents\Anaconda\lib\site-packages\seaborn\categorical.py:1428: FutureWarning: remove_na is deprecate
d and is a private function. Do not use.
  stat_data = remove_na(group_data)
```

Type here to search ENG 3:18 AM 7/17/2018

Desktop/r assignment/r/dat | BasketballAnalysis

localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb

jupyter **BasketballAnalysis** Last Checkpoint: 43 minutes ago (autosaved) Logout

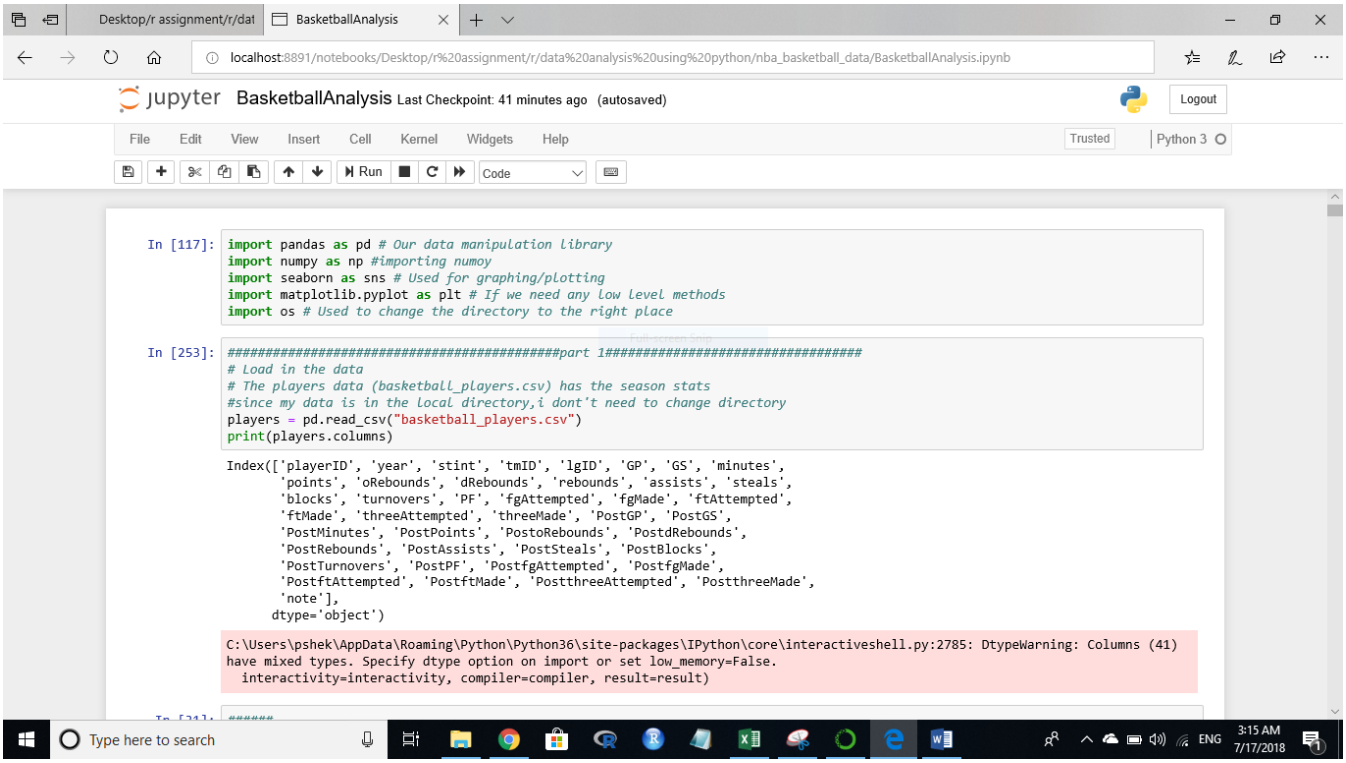
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [31]:
```
######
#part1
#the following answer will cover 1st and 4th question of part1
#calculate the mean and median  of number of points scored per season and also showing  how the number of points change over time
mean_grouped=players[['points','year']].groupby('year')['points'].mean()
mean_grouped=mean_grouped.reset_index()
print(mean_grouped)
sns.regplot(data=mean_grouped, x="year", y="points")
```

```
    year      points
0   1937   42.985915
1   1938   67.959677
2   1939   78.223214
3   1940   72.043478
4   1941   74.189474
5   1942   65.500000
6   1943   79.021739
7   1944  111.573171
8   1945   94.722222
9   1946  196.050000
10  1947  158.304536
11  1948  263.963068
12  1949  333.888476
13  1950  247.460000
14  1951  404.370079
15  1952  434.694656
16  1953  393.280992
17  1954  442.103448
18  1955  569.560000
19  1956  537.380952
```

Type here to search ENG 3:17 AM 7/17/2018


Desktop/r assignment/r/dat
BasketballAnalysis
localhost:8891/notebooks/Desktop/r%20assignment/r/data%20analysis%20using%20python/nba_basketball_data/BasketballAnalysis.ipynb
jupyter BasketballAnalysis Last Checkpoint: 41 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [117]:
import pandas as pd # Our data manipulation library
import numpy as np #importing numoy
import seaborn as sns # Used for graphing/plotting
import matplotlib.pyplot as plt # If we need any low level methods
import os # Used to change the directory to the right place
In [253]:
##part 1#################################
Load in the data
The players data (basketball_players.csv) has the season stats
#since my data is in the local directory,i dont't need to change directory
players = pd.read_csv("basketball_players.csv")
print(players.columns)
Index(['playerID', 'year', 'stint', 'tmID', 'lgID', 'GP', 'GS', 'minutes',
'points', 'oRebounds', 'dRebounds', 'rebounds', 'assists', 'steals',
'blocks', 'turnovers', 'PF', 'fgAttempted', 'fgMade', 'ftAttempted',
'ftMade', 'threeAttempted', 'threeMade', 'PostGP', 'PostGS',
'PostMinutes', 'PostPoints', 'PostoRebounds', 'PostdRebounds',
'PostRebounds', 'PostAssists', 'PostSteals', 'PostBlocks',
'PostTurnovers', 'PostPF', 'PostfgAttempted', 'PostfgMade',
'PostftAttempted', 'PostftMade', 'PostthreeAttempted', 'PostthreeMade',
'note'],
dtype='object')
C:\Users\pshek\AppData\Roaming\Python\Python36\site-packages\IPython\core\interactiveshell.py:2785: DtypeWarning: Columns (41)
have mixed types. Specify dtype option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)
Type here to search
ENG
3:15 AM
7/17/2018