Assignment 1 with Batting Dataset

Github Links: https://github.com/TrangNguyen95/BattingDecriptiveAnalysis- (https://github.com/TrangNguyen95/BattingDecriptiveAnalysis-)

In [1]:
```
import numpy as np
import pandas as pd
```

1) Load in the appropriate csv file as a pandas dataframe (batting.csv)

In [2]:
```
df=pd.read_csv('Batting.csv')
```

2) Print out the dimensions and info about the dataframe you just created

The dataframe (df) has 102,816 rows and 25 columns.

In [3]:
```
df.shape
```

Out[3]: (102816, 25)

In [4]:
```
df.ndim
```

Out[4]: 2

In [5]:
```
df.size
```

Out[5]: 2570400

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102816 entries, 0 to 102815
Data columns (total 25 columns):
playerID     102816 non-null object
nameFirst    102816 non-null object
nameLast     102816 non-null object
birthYear    102816 non-null int64
yearID       102816 non-null int64
stint        102816 non-null int64
teamID       102816 non-null object
lgID         102079 non-null object
G            102816 non-null int64
AB           102816 non-null int64
R            102816 non-null int64
H            102816 non-null int64
2B           102816 non-null int64
3B           102816 non-null int64
HR           102816 non-null int64
RBI          102392 non-null float64
SB           101516 non-null float64
CS           79360 non-null float64
BB           102816 non-null int64
SO           94978 non-null float64
IBB          66251 non-null float64
HBP          100006 non-null float64
SH           96478 non-null float64
SF           66782 non-null float64
GIDP         76706 non-null float64
dtypes: float64(9), int64(11), object(5)
memory usage: 19.6+ MB
```

3) How many players have hit 40 or more HRs in one single season? (Number only)

In [7]:
```
df[df.HR>=40].count()["playerID"]
```

Out[7]: 326

4) How many players have hit more than 600 HRs for their career? (Dataframe)

In [8]:
```
df4_HR=df.groupby(["playerID","nameFirst","nameLast"]).agg({"HR":"sum"}).reset_index()
```

In [9]:
```
df4_HR.groupby('playerID').filter(lambda x:x['HR']>600).sort_values('HR',ascending=False)
```

Out[9]:

	playerID	nameFirst	nameLast	HR
1542	bondsba01	Barry	Bonds	762
1	aaronha01	Hank	Aaron	755
14865	ruthba01	Babe	Ruth	714
14528	rodrial01	Alex	Rodriguez	696
10857	mayswi01	Willie	Mays	660
6633	griffke02	Ken	Griffey	630
17004	thomeji01	Jim	Thome	612
16103	sosasa01	Sammy	Sosa	609

5) How many players have hit 40 2Bs, 10 3Bs, 200 Hits, and 30 HRs (inclusive) in one season? (Number Only)

In [10]:
```
#Final answer
df5=df.groupby(["playerID","nameFirst","nameLast","yearID"]).agg({"2B":"sum","3B":"sum","H":"sum","HR":"sum"}).reset_index()
df5_1=df5[(df5['2B']>=40) & (df5['3B']>=10) & (df5['H']>=200) & (df5['HR']>=30)].reset_index()
df5_1['playerID'].nunique()
```

Out[10]: 11

6) How many players have had 100 or more SBs in a season? (Dataframe)

In [11]:
```
df6=df[df.SB>=100].groupby(["playerID","nameFirst","nameLast","yearID"]).agg({"SB":"sum"})
df6
```

Out[11]:

				SB
playerID	**nameFirst**	**nameLast**	**yearID**	
brocklo01	**Lou**	**Brock**	**1974**	118.0
brownpe01	**Pete**	**Browning**	**1887**	103.0
brownto01	**Tom**	**Brown**	**1891**	106.0
colemvi01	**Vince**	**Coleman**	**1985**	110.0
			1986	107.0
			1987	109.0
comisch01	**Charlie**	**Comiskey**	**1887**	117.0
fogarji01	**Jim**	**Fogarty**	**1887**	102.0
hamilbi01	**Billy**	**Hamilton**	**1889**	111.0
			1890	102.0
			1891	111.0
henderi01	**Rickey**	**Henderson**	**1980**	100.0
			1982	130.0
			1983	108.0
lathaar01	**Arlie**	**Latham**	**1887**	129.0
			1888	109.0
nicolhu01	**Hugh**	**Nicol**	**1887**	138.0
			1888	103.0
wardjo01	**John**	**Ward**	**1887**	111.0
willsma01	**Maury**	**Wills**	**1962**	104.0

7) How many players in the 1960s have hit more than 200 HRs? (Dataframe)

In [12]:
```
df7=df[(df.yearID>=1960) & (df.yearID<=1969)]
```

In [13]:
```
df7_1=df7.groupby(["playerID","nameFirst","nameLast"]).agg({"HR":"sum"})
df7_1[df7_1.HR>200]
```

Out[13]:

			HR
playerID	**nameFirst**	**nameLast**	
aaronha01	**Hank**	**Aaron**	375
allisbo01	**Bob**	**Allison**	225
bankser01	**Ernie**	**Banks**	269
cashno01	**Norm**	**Cash**	278
cepedor01	**Orlando**	**Cepeda**	254
colavro01	**Rocky**	**Colavito**	245
howarfr01	**Frank**	**Howard**	288
kalinal01	**Al**	**Kaline**	210
killeha01	**Harmon**	**Killebrew**	393
mantlmi01	**Mickey**	**Mantle**	256
marisro01	**Roger**	**Maris**	217
matheed01	**Eddie**	**Mathews**	213
mayswi01	**Willie**	**Mays**	350
mccovwi01	**Willie**	**McCovey**	300
powelbo01	**Boog**	**Powell**	202
robinfr02	**Frank**	**Robinson**	316
santoro01	**Ron**	**Santo**	253
willibi01	**Billy**	**Williams**	249
yastrca01	**Carl**	**Yastrzemski**	202

8) Who has hit the most HRs in history? (Dataframe)

In [14]:
```
df8=df.groupby(["playerID","nameFirst","nameLast"]).agg({"HR":"sum"})
df8[df8["HR"]==max(df8["HR"])]
```

Out[14]:

			HR
playerID	**nameFirst**	**nameLast**	
bondsba01	**Barry**	**Bonds**	762

9) Who had the most hits in the 1970s? (Dataframe)

In [15]:
```
df9=df[(df.yearID>=1970)&(df.yearID<=1979)]
```

In [16]:
```
df9_1=df9.groupby(['playerID','nameFirst','nameLast']).agg({"H":"sum"})
df9_1[df9_1["H"]==max(df9_1["H"])]
```

Out[16]:

			H
playerID	**nameFirst**	**nameLast**	
rosepe01	**Pete**	**Rose**	2045

10) Top 5 highest OBP (on base percentage) with at least 500 PAs in 1977? (Dataframe)

In [17]:
```
df10=df[(df.yearID==1977)].groupby(['playerID','nameFirst','nameLast']).agg({"H":"sum","BB":"sum","IBB":"sum","SH":"sum","SF":'
```

In [18]:
```
df10_1=pd.DataFrame({'H':df10["H"],'BB':df10["BB"],'IBB':df10["IBB"],'SH':df10["SH"],'SF':df10["SF"],'AB':df10["AB"],'PA':(df10
```

In [19]:
```
df10_2=df10_1[(df10_1["PA"]>=500)]
df10_2.sort_values('OBP',ascending=False).head(n=5)
```

Out[19]:

			AB	BB	H	IBB	OBP	PA	SF	SH
playerID	nameFirst	nameLast								
singlke01	Ken	Singleton	536	107	176	13.0	0.563433	662.0	6.0	0.0
smithre06	Reggie	Smith	488	104	150	11.0	0.559426	611.0	7.0	1.0
tenacge01	Gene	Tenace	437	125	102	10.0	0.556064	578.0	4.0	2.0
hargrmi01	Mike	Hargrove	525	103	160	7.0	0.540952	649.0	6.0	8.0
carewro01	Rod	Carew	616	69	239	15.0	0.534091	706.0	5.0	1.0

11) Top 8 highest averages in 2013 with at least 300 PAs? (Dataframe)

In [20]:
```
df11=df[(df.yearID==2013)].groupby(['playerID','nameFirst','nameLast']).agg({"H":"sum","BB":"sum","IBB":"sum","SH":"sum","SF":'
```

In [21]:
```
df11_1=pd.DataFrame({'H':df11["H"],'BB':df11["BB"],'IBB':df11["IBB"],'SH':df11["SH"],'SF':df11["SF"],'AB':df11["AB"],'PA':(df11
```

In [22]:
```
df11_2=df11_1[(df11_1["PA"]>=300)]
df11_2.sort_values('Average',ascending=False).head(n=8)
```

Out[22]:

			AB	Average	BB	H	IBB	PA	SF	SH
playerID	nameFirst	nameLast								
cabremi01	Miguel	Cabrera	555	0.347748	90	193	19.0	666.0	2.0	0.0
ramirha01	Hanley	Ramirez	304	0.345395	27	105	3.0	336.0	2.0	0.0
cuddymi01	Michael	Cuddyer	489	0.331288	46	162	5.0	543.0	3.0	0.0
mauerjo01	Joe	Mauer	445	0.323596	61	144	7.0	515.0	2.0	0.0
troutmi01	Mike	Trout	589	0.322581	110	190	10.0	717.0	8.0	0.0
johnsch05	Chris	Johnson	514	0.321012	29	165	5.0	550.0	2.0	0.0
freemfr01	Freddie	Freeman	551	0.319419	66	176	10.0	632.0	5.0	0.0
puigya01	Yasiel	Puig	382	0.319372	36	122	6.0	427.0	3.0	0.0

12) Leaders in hits from 1940 up to and including 1949. (Dataframe) (Top 10, sorted by hit)

In [23]:
```
df12=df[(df.yearID>=1940) & (df.yearID<=1949)].groupby(['playerID','nameFirst','nameLast']).agg({"H":"sum"}).sort_values('H',as
df12
```

Out[23]:

	playerID	nameFirst	nameLast	H
0	boudrlo01	Lou	Boudreau	1578
1	elliobo01	Bob	Elliott	1563
2	walkedi02	Dixie	Walker	1512
3	musiast01	Stan	Musial	1432
4	doerrbo01	Bobby	Doerr	1407
5	holmeto01	Tommy	Holmes	1402
6	applilu01	Luke	Appling	1376
7	nichobi01	Bill	Nicholson	1328
8	marioma01	Marty	Marion	1310
9	cavarph01	Phil	Cavarretta	1304

13) Who led MLB with the most hits the most times? And how many times? (Dataframe, Number)

(column for the player that led MLB ID, name, year, how many hits, how many times)

In [24]:
```
df13=df.groupby(["yearID","playerID","nameFirst","nameLast"]).agg({"H":"sum"}).reset_index()
```

In [25]:
```
df13_1=df13.groupby(["yearID"]).apply(lambda g:g[g["H"]==g["H"].max()]).sort_values('H',ascending=False)
df13_1.set_index('yearID').reset_index()
```

Out[25]:

	yearID	playerID	nameFirst	nameLast	H
0	2004	suzukic01	Ichiro	Suzuki	262
1	1920	sislege01	George	Sisler	257
2	1930	terrybi01	Bill	Terry	254
3	1929	odoulle01	Lefty	O'Doul	254
4	1925	simmoal01	Al	Simmons	253
5	1922	hornsro01	Rogers	Hornsby	250
6	1911	cobbty01	Ty	Cobb	248
7	2001	suzukic01	Ichiro	Suzuki	242
8	1928	manushe01	Heinie	Manush	241
9	1896	burkeje01	Jesse	Burkett	240
10	2000	erstada01	Darin	Erstad	240
11	1985	boggswa01	Wade	Boggs	240
12	1897	keelewi01	Willie	Keeler	239
13	1977	carewro01	Rod	Carew	239
14	1899	delahed01	Ed	Delahanty	238
15	2007	suzukic01	Ichiro	Suzuki	238
16	1986	mattido01	Don	Mattingly	238
17	1927	wanerpa01	Paul	Waner	237
18	1937	medwijo01	Joe	Medwick	237
19	1921	heilmha01	Harry	Heilmann	237
20	1894	duffyhu01	Hugh	Duffy	237
21	1988	puckeki01	Kirby	Puckett	234
22	1901	lajoina01	Nap	Lajoie	232
23	1936	averiea01	Earl	Averill	232
24	1969	alouma01	Matty	Alou	231
25	1948	musiast01	Stan	Musial	230
26	1973	rosepe01	Pete	Rose	230

	yearID	playerID	nameFirst	nameLast	H
27	1980	wilsowi02	Willie	Wilson	230
28	1962	davisto02	Tommy	Davis	230
29	1971	torrejo01	Joe	Torre	230
...	...	...	...	...	...
129	1995	bicheda01	Dante	Bichette	197
130	1995	gwynnto01	Tony	Gwynn	197
131	1913	jacksjo01	Shoeless Joe	Jackson	197
132	1952	musiast01	Stan	Musial	194
133	1902	hickmch01	Charlie	Hickman	193
134	1902	beaumgi01	Ginger	Beaumont	193
135	1886	orrda01	Dave	Orr	193
136	1919	cobbty01	Ty	Cobb	191
137	1919	veachbo01	Bobby	Veach	191
138	1960	mayswi01	Willie	Mays	190
139	1891	brownto01	Tom	Brown	189
140	1884	dunlafr01	Fred	Dunlap	185
141	1888	ryanji01	Jimmy	Ryan	182
142	1918	burnsge02	George	Burns	178
143	1885	brownpe01	Pete	Browning	174
144	1994	gwynnto01	Tony	Gwynn	165
145	1883	broutda01	Dan	Brouthers	159
146	1879	hinespa01	Paul	Hines	146
147	1875	barnero01	Ross	Barnes	143
148	1981	rosepe01	Pete	Rose	140
149	1876	barnero01	Ross	Barnes	138
150	1881	ansonca01	Cap	Anson	137
151	1873	barnero01	Ross	Barnes	137
152	1882	broutda01	Dan	Brouthers	129
153	1880	dalryab01	Abner	Dalrymple	126
154	1874	mcveyca01	Cal	McVey	123

	yearID	playerID	nameFirst	nameLast	H
155	1877	whitede01	Deacon	White	103
156	1878	startjo01	Joe	Start	100
157	1872	barnero01	Ross	Barnes	99
158	1871	mcveyca01	Cal	McVey	66

159 rows × 5 columns

In [26]:
```
df13_2a=df13_1.groupby(['playerID']).agg({"H":"sum"}).reset_index()
```

In [27]:
```
df13_2b=pd.DataFrame({'playerID':df13_2a["playerID"],'Total_Hits':df13_2a["H"]})
```

In [28]:
```
df13_3a=df13_1.groupby(['playerID']).count().sort_values('H',ascending=False).reset_index()
```

In [29]:
```
df13_3b=pd.DataFrame({'playerID':df13_3a["playerID"],'Total_Number_of_Hits':df13_3a["H"]})
```

In [30]:
```
df13_4a=pd.merge(df13_1,df13_3b,how='right',on=['playerID']).sort_values('H',ascending=False)
```

In [31]:
```
df13_4b=pd.merge(df13_4a,df13_2b,how='left',on=['playerID'])
```

In [32]:
```
df13_5=df13_4b[df13_4b['Total_Hits']==df13_4b['Total_Hits'].max()].groupby(['playerID','nameFirst','nameLast','yearID']).agg({'
df13_5
```

Out[32]:

				H	Total_Number_of_Hits	Total_Hits
playerID	**nameFirst**	**nameLast**	**yearID**			
suzukic01	**Ichiro**	**Suzuki**	**2001**	242	7	1618
			2004	262	7	1618
			2006	224	7	1618
			2007	238	7	1618
			2008	213	7	1618
			2009	225	7	1618
			2010	214	7	1618

In [33]:
```
df13_5['Total_Hits'].max()
```

Out[33]: 1618

In [34]:
```
df13_5['Total_Number_of_Hits'].max()
```

Out[34]: 7

14) Which players have played the most games for their careers? Top 5, descending by games played presented as a dataframe

In [35]:
```
df14=df.groupby(['playerID','nameFirst','nameLast']).agg({'yearID':'count'}).sort_values('yearID',ascending=False).head(n=5).re
df14_1=pd.DataFrame({'playerID':df14['playerID'],'FirstName':df14['nameFirst'],'LastName':df14['nameLast'],'Games':df14['yearID
df14_1
```

Out[35]:

	playerID	FirstName	LastName	Games
0	mcguide01	Deacon	McGuire	31
1	henderi01	Rickey	Henderson	29
2	newsobo01	Bobo	Newsom	29
3	johnto01	Tommy	John	28
4	kaatji01	Jim	Kaat	28

15) How many players have had more than 3000 hits for their careers while also hitting 500 or more HRs? Just a number is okay here

In [36]:
```
df15_1=df.groupby(['playerID','nameFirst','nameLast']).agg({'H':'sum','HR':'sum'}).reset_index()
```

In [37]:
```
df15_2=df15_1[(df15_1['H']>3000)&(df15_1['HR']>=500)]
df15_2['playerID'].count()
```

Out[37]: 5

16) How many HRs were hit during the entire 1988 season? Just a number is okay here

In [38]:
```
df16=df[(df.yearID==1988)]['HR'].sum()
df16
```

Out[38]: 3180

17) Please filter out and show me the top 3 average seasons by Wade Boggs during his career in seasons in which he had at least 500 ABs. I would like a dataframe sorted by average.

In [39]:
```
df17=df[(df['nameFirst'].str.contains("Wade")&df['nameLast'].str.contains("Boggs"))]
df17_1=pd.DataFrame({'PlayerID':df17['playerID'],'FirstName':df17['nameFirst'],'LastName':df17['nameLast'],'Year':df17['yearID'
df17_2=df17_1[(df17_1['AB']>=500)]
df17_2.sort_values('Average',ascending=False).head(n=3)
```

Out[39]:

				Average	AB
PlayerID	**FirstName**	**LastName**	**Year**		
boggswa01	**Wade**	**Boggs**	**1985**	0.367534	653
			1988	0.366438	584
			1987	0.362976	551

18) Please filter out the top OBPs for the 1995 season with at least 400 PAs, sorted by OBP. I would like a dataframe for this

In [40]:
```
df18=df[(df.yearID==1995)].groupby(['playerID','nameFirst','nameLast','yearID']).agg({"H":"sum","BB":"sum","IBB":"sum","SH":"su
df18_1=pd.DataFrame({'PA':(df18["AB"]+df18["BB"]+df18["IBB"]+df18["SH"]+df18["SF"]),'OBP':((df18["H"]+df18["BB"]+df18["IBB"]+df
```

In [41]:
```
df18_2=df18_1[(df18_1["PA"]>=400)]
df18_2.sort_values('OBP',ascending=False)
```

Out[41]:

				OBP	PA
playerID	**nameFirst**	**nameLast**	**yearID**		
thomafr04	**Frank**	**Thomas**	**1995**	0.667343	670.0
martied01	**Edgar**	**Martinez**	**1995**	0.628180	650.0
mcgwima01	**Mark**	**McGwire**	**1995**	0.586751	416.0
bondsba01	**Barry**	**Bonds**	**1995**	0.583004	652.0
davisch01	**Chili**	**Davis**	**1995**	0.577830	534.0
magadda01	**Dave**	**Magadan**	**1995**	0.551724	431.0
thomeji01	**Jim**	**Thome**	**1995**	0.542035	555.0
baineha01	**Harold**	**Baines**	**1995**	0.524675	472.0
weisswa01	**Walt**	**Weiss**	**1995**	0.524590	540.0
boggswa01	**Wade**	**Boggs**	**1995**	0.510870	546.0
naehrti01	**Tim**	**Naehring**	**1995**	0.510393	521.0
salmoti01	**Tim**	**Salmon**	**1995**	0.510242	634.0
joynewa01	**Wally**	**Joyner**	**1995**	0.509677	558.0
tettlmi01	**Mickey**	**Tettleton**	**1995**	0.508159	545.0
bagweje01	**Jeff**	**Bagwell**	**1995**	0.506696	545.0
oneilpa01	**Paul**	**O'Neill**	**1995**	0.495652	550.0
phillto02	**Tony**	**Phillips**	**1995**	0.491429	646.0
henderi01	**Rickey**	**Henderson**	**1995**	0.491400	485.0
ramirma02	**Manny**	**Ramirez**	**1995**	0.489669	572.0
clarkwi02	**Will**	**Clark**	**1995**	0.488987	539.0
knoblch01	**Chuck**	**Knoblauch**	**1995**	0.488848	622.0
venturo01	**Robin**	**Ventura**	**1995**	0.487805	587.0
olerujo01	**John**	**Olerud**	**1995**	0.483740	587.0
gantro01	**Ron**	**Gant**	**1995**	0.482927	495.0
biggicr01	**Craig**	**Biggio**	**1995**	0.481013	652.0
valenjo02	**John**	**Valentin**	**1995**	0.476923	613.0

playerID	nameFirst	nameLast	yearID	OBP	PA
gracema01	**Mark**	**Grace**	**1995**	0.474638	634.0
coninje01	**Jeff**	**Conine**	**1995**	0.474120	566.0
alicelu01	**Luis**	**Alicea**	**1995**	0.472554	504.0
offerjo01	**Jose**	**Offerman**	**1995**	0.470862	508.0
...	**...**	**...**	**...**	...	...
steinte01	**Terry**	**Steinbach**	**1995**	0.362069	440.0
grissma02	**Marquis**	**Grissom**	**1995**	0.359347	607.0
manwaki01	**Kirt**	**Manwaring**	**1995**	0.358839	420.0
galaran01	**Andres**	**Galarraga**	**1995**	0.357401	597.0
blausje01	**Jeff**	**Blauser**	**1995**	0.357309	494.0
greensh01	**Shawn**	**Green**	**1995**	0.356201	405.0
kentje01	**Jeff**	**Kent**	**1995**	0.355932	509.0
hillgl01	**Glenallen**	**Hill**	**1995**	0.354125	542.0
jordabr01	**Brian**	**Jordan**	**1995**	0.353061	518.0
girarjo01	**Joe**	**Girardi**	**1995**	0.352814	504.0
zeileto01	**Todd**	**Zeile**	**1995**	0.349765	470.0
rodriiv01	**Ivan**	**Rodriguez**	**1995**	0.349593	515.0
lewisda01	**Darren**	**Lewis**	**1995**	0.349576	519.0
kellyro01	**Roberto**	**Kelly**	**1995**	0.347222	539.0
durhara01	**Ray**	**Durham**	**1995**	0.346072	513.0
beckeri01	**Rich**	**Becker**	**1995**	0.344388	434.0
dunstsh01	**Shawon**	**Dunston**	**1995**	0.343816	500.0
macfami01	**Mike**	**Macfarlane**	**1995**	0.340659	406.0
sanchre01	**Rey**	**Sanchez**	**1995**	0.338785	454.0
cartejo01	**Joe**	**Carter**	**1995**	0.336918	605.0
easleda01	**Damion**	**Easley**	**1995**	0.336134	400.0
gomezch02	**Chris**	**Gomez**	**1995**	0.334107	479.0
claytro01	**Royce**	**Clayton**	**1995**	0.333988	555.0
walbema01	**Matt**	**Walbeck**	**1995**	0.333333	423.0

				OBP	PA
playerID	**nameFirst**	**nameLast**	**yearID**		
colbrgr01	**Greg**	**Colbrunn**	**1995**	0.333333	558.0
mearepa01	**Pat**	**Meares**	**1995**	0.330769	414.0
lansimi01	**Mike**	**Lansing**	**1995**	0.327623	501.0
gilbe01	**Benji**	**Gil**	**1995**	0.310843	453.0
cedenan01	**Andujar**	**Cedeno**	**1995**	0.302564	426.0
guilloz01	**Ozzie**	**Guillen**	**1995**	0.293976	434.0

171 rows × 2 columns

In [42]:
```
#Top 10
df18_2.sort_values('OBP',ascending=False).head(n=10)
```

Out[42]:

				OBP	PA
playerID	**nameFirst**	**nameLast**	**yearID**		
thomafr04	**Frank**	**Thomas**	**1995**	0.667343	670.0
martied01	**Edgar**	**Martinez**	**1995**	0.628180	650.0
mcgwima01	**Mark**	**McGwire**	**1995**	0.586751	416.0
bondsba01	**Barry**	**Bonds**	**1995**	0.583004	652.0
davisch01	**Chili**	**Davis**	**1995**	0.577830	534.0
magadda01	**Dave**	**Magadan**	**1995**	0.551724	431.0
thomeji01	**Jim**	**Thome**	**1995**	0.542035	555.0
baineha01	**Harold**	**Baines**	**1995**	0.524675	472.0
weisswa01	**Walt**	**Weiss**	**1995**	0.524590	540.0
boggswa01	**Wade**	**Boggs**	**1995**	0.510870	546.0

19) Who had the most 3Bs (in total) in 1922, 1925, 1926, and 1928? I would like a dataframe with just the leader

In [43]:
```
df19=df[(df['yearID']==1922) | (df['yearID']==1925) | (df['yearID']==1926) | (df['yearID']==1928)].groupby(['playerID','nameFir
df19_1=df19[(df19['3B']==df19['3B'].max())]
df19_1
```

Out[43]:

			3B
playerID	**nameFirst**	**nameLast**	
walkecu01	**Curt**	**Walker**	59

20) How many players have hit 30 or more HRs in season while also stealing (SB) 30 more or bases? A number is okay here

In [44]:
```
df20_1=df[(df['HR']>=30)&(df['SB']>=30)].reset_index()
df20_1['playerID'].nunique()
```

Out[44]: 37

21) Who had the highest OBP is 1986 with at least 400 PAs? (Dataframe)

In [45]:
```
df21_1=df[(df['yearID']==1986)].groupby(['playerID','nameFirst','nameLast','yearID']).agg({"H":"sum","BB":"sum","IBB":"sum","SH
df21_2=pd.DataFrame({'PA':(df21_1["AB"]+df21_1["BB"]+df21_1["IBB"]+df21_1["SH"]+df21_1["SF"]),'OBP':((df21_1["H"]+df21_1["BB"]+
```

In [46]:
```
df21_3=df21_2[(df21_2['PA']>=400)]
df21_3.sort_values('OBP',ascending=False).head(n=1)
```

Out[46]:

				OBP	PA
playerID	**nameFirst**	**nameLast**	**yearID**		
boggswa01	**Wade**	**Boggs**	**1986**	0.575862	707.0

22) Same question but for 1997 and only in the NL (check league ID)? (Dataframe)

In [47]:
```
df22_1=df[(df['yearID']==1997)&(df['lgID'].str.contains('NL'))].groupby(['playerID','nameFirst','nameLast','lgID','yearID']).ag
df22_2=pd.DataFrame({'PA':(df22_1["AB"]+df22_1["BB"]+df22_1["IBB"]+df22_1["SH"]+df22_1["SF"]),'OBP':((df22_1["H"]+df22_1["BB"]+
```

In [48]:
```
df22_3=df22_2[(df22_2['PA']>=400)]
df22_3.sort_values('OBP',ascending=False).head(n=1)
```

Out[48]:

	playerID	nameFirst	nameLast	lgID	yearID	OBP	PA
60	bondsba01	Barry	Bonds	NL	1997	0.637218	716.0

23) Who had more than the league average HRs in 2012 (filter out all players with less 500 PAs)? (Dataframe)

In [49]:
```
df23_1=df[(df['yearID']==2012)].groupby(['playerID','nameFirst','nameLast','yearID']).agg({'HR':'sum','AB':'sum','BB':'sum','IB
df23_2=pd.DataFrame({'HR':df23_1['HR'],'PA':(df23_1["AB"]+df23_1["BB"]+df23_1["IBB"]+df23_1["SH"]+df23_1["SF"]),'AverageHR':df2
df23_2[(df23_2['HR']>=df23_2['HR'].mean())&(df23_2['PA']<500)].sort_values('PA',ascending=False)
```

Out[49]:

	playerID	nameFirst	nameLast	yearID	AverageHR	HR	PA
106	boescbr01	Brennan	Boesch	2012	3.842679	12	499.0
639	kotchca01	Casey	Kotchman	2012	3.842679	12	494.0
1270	youklke01	Kevin	Youkilis	2012	3.842679	19	494.0
758	mccanbr01	Brian	McCann	2012	3.842679	20	493.0
584	jayjo02	Jon	Jay	2012	3.842679	4	490.0
1000	roberry01	Ryan	Roberts	2012	3.842679	12	489.0
1220	vottojo01	Joey	Votto	2012	3.842679	14	488.0
55	barmecl01	Clint	Barmes	2012	3.842679	8	488.0
279	davisra01	Rajai	Davis	2012	3.842679	8	484.0
256	crawfbr01	Brandon	Crawford	2012	3.842679	4	479.0
752	maybejo02	John	Mayberry	2012	3.842679	14	479.0
740	martiru01	Russell	Martin	2012	3.842679	21	477.0
81	beltbr01	Brandon	Belt	2012	3.842679	7	474.0
689	loneyja01	James	Loney	2012	3.842679	6	472.0
724	markani01	Nick	Markakis	2012	3.842679	13	470.0
702	ludwiry01	Ryan	Ludwick	2012	3.842679	26	470.0
1203	venabwi01	Will	Venable	2012	3.842679	9	467.0
394	frazito01	Todd	Frazier	2012	3.842679	19	463.0
939	plouftr01	Trevor	Plouffe	2012	3.842679	24	461.0
608	joycema01	Matthew	Joyce	2012	3.842679	17	460.0
909	pennicl01	Cliff	Pennington	2012	3.842679	6	460.0
344	ellisma01	Mark	Ellis	2012	3.842679	7	457.0
326	dudalu01	Lucas	Duda	2012	3.842679	15	455.0
619	kempma01	Matt	Kemp	2012	3.842679	23	454.0
99	blancgr01	Gregor	Blanco	2012	3.842679	5	453.0
605	jonesch06	Chipper	Jones	2012	3.842679	14	453.0

	playerID	nameFirst	nameLast	yearID	AverageHR	HR	PA
1048	saltaja01	Jarrod	Saltalamacchia	2012	3.842679	25	447.0
434	gomezca01	Carlos	Gomez	2012	3.842679	19	445.0
1056	sandopa01	Pablo	Sandoval	2012	3.842679	12	445.0
237	colvity01	Tyler	Colvin	2012	3.842679	18	444.0
...	...	...	...	...	...	...	...
781	mesorde01	Devin	Mesoraco	2012	3.842679	5	187.0
180	carpmi01	Mike	Carp	2012	3.842679	5	187.0
899	pearcst01	Steve	Pearce	2012	3.842679	4	186.0
198	cedenro02	Ronny	Cedeno	2012	3.842679	4	185.0
1106	snidetr01	Travis	Snider	2012	3.842679	4	184.0
731	martest01	Starling	Marte	2012	3.842679	5	179.0
28	ankieri01	Rick	Ankiel	2012	3.842679	5	174.0
570	ishiktr01	Travis	Ishikawa	2012	3.842679	4	173.0
614	kearnau01	Austin	Kearns	2012	3.842679	4	170.0
801	moorety01	Tyler	Moore	2012	3.842679	10	170.0
827	nadyxa01	Xavier	Nady	2012	3.842679	4	167.0
377	flahery01	Ryan	Flaherty	2012	3.842679	6	163.0
468	gutiefr01	Franklin	Gutierrez	2012	3.842679	4	161.0
643	kratzer01	Erik	Kratz	2012	3.842679	9	157.0
1095	sierrmo01	Moises	Sierra	2012	3.842679	6	155.0
213	chiselo01	Lonnie	Chisenhall	2012	3.842679	5	150.0
380	flowety01	Tyler	Flowers	2012	3.842679	7	149.0
244	coopeda01	David	Cooper	2012	3.842679	4	144.0
576	jacksbr01	Brett	Jackson	2012	3.842679	4	142.0
610	kaaihki01	Kila	Ka'aihue	2012	3.842679	4	138.0
125	brownan02	Andrew	Brown	2012	3.842679	5	126.0
734	martife02	Fernando	Martinez	2012	3.842679	6	125.0
313	dominma01	Matt	Dominguez	2012	3.842679	5	114.0
937	pillbr01	Brett	Pill	2012	3.842679	4	113.0
433	gomesya01	Yan	Gomes	2012	3.842679	4	108.0

	playerID	nameFirst	nameLast	yearID	AverageHR	HR	PA
239	conrabr01	Brooks	Conrad	2012	3.842679	4	105.0
599	johnsni01	Nick	Johnson	2012	3.842679	4	98.0
438	gonzaal02	Alex	Gonzalez	2012	3.842679	4	87.0
248	corpoca01	Carlos	Corporan	2012	3.842679	4	84.0
983	reimono01	Nolan	Reimold	2012	3.842679	5	69.0

190 rows × 7 columns

24) Who is the youngest player to hit 50 or more HRs in a single season? (Dataframe)

In [50]:
```
df24_1=pd.DataFrame({'playerID':df['playerID'],'yearID':df['yearID'],'Age':(2018-df['birthYear'])})
df24_2=pd.merge(df,df24_1,how='left',on=['playerID','yearID']).reset_index()
```

In [51]:
```
df24_3=df24_2[(df24_2['HR']>=50)].groupby(['playerID','nameFirst','nameLast','Age','yearID']).agg({'HR':'sum'}).reset_index().s
df24_4=df24_3[(df24_3['Age']==df24_3['Age'].min())]
df24_4
```

Out[51]:

	playerID	nameFirst	nameLast	Age	yearID	HR
4	davisch02	Chris	Davis	32	2013	53

25) Who are the five youngest players to hit 300 or more HRs for their career? (Dataframe)

In [52]:
```
df25_1=pd.DataFrame({'playerID':df['playerID'],'yearID':df['yearID'],'Age':(2018-df['birthYear']),'HR':df['HR']})
df25_2=pd.merge(df,df25_1,how='left',on=['playerID','yearID','HR'])
```

In [53]:
```
df25_3=df25_2.groupby(['playerID','nameFirst','nameLast']).agg({'HR':'sum','Age':'mean'}).reset_index()
```

In [54]:
```
df25_4=df25_3[(df25_3['HR']>=300)].sort_values(['Age','HR']).head(n=5)
df25_4
```

Out[54]:

	playerID	nameFirst	nameLast	HR	Age
5287	fieldpr01	Prince	Fielder	319	34
4953	encared01	Edwin	Encarnacion	310	35
2385	cabremi01	Miguel	Cabrera	446	35
6329	gonzaad01	Adrian	Gonzalez	308	36
947	bautijo02	Jose	Bautista	308	38

In [55]:
```
#since there are 3 people born in 1980 who turned 38 in 2018 and without sorting for HR, the 5th person on the list could be ar
df25_5=df25_3[(df25_3['HR']>=300) &(df25_3['Age']==38)]
df25_5
```

Out[55]:

	playerID	nameFirst	nameLast	HR	Age
947	bautijo02	Jose	Bautista	308	38
13782	pujolal01	Albert	Pujols	591	38
16879	teixema01	Mark	Teixeira	409	38

BONUS1: Graph total HRs per season using bar graph.

In [56]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [57]:
```
from plotly import __version__
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot

print(__version__)
```

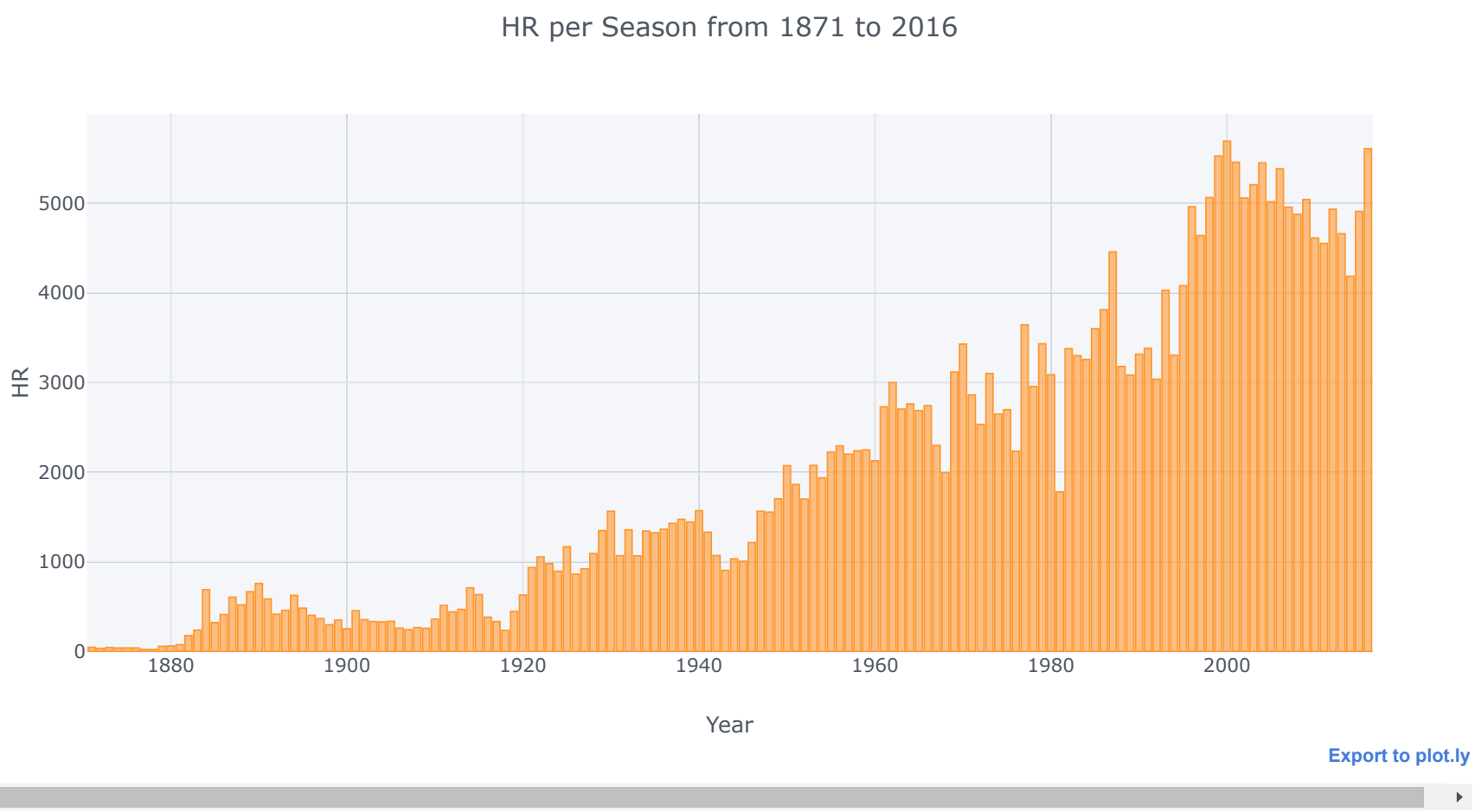
```
2.3.0
```

In [58]:
```
import cufflinks as cf
init_notebook_mode(connected=True)
cf.go_offline()
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.
```

In [59]:
```
dfB1=df.groupby(['yearID']).agg({'HR':'sum'}).reset_index()
```

In [60]:
```
dfB1.iplot(kind='bar',x='yearID',y='HR',xTitle="Year",yTitle="HR",title="HR per Season from 1871 to 2016")
```

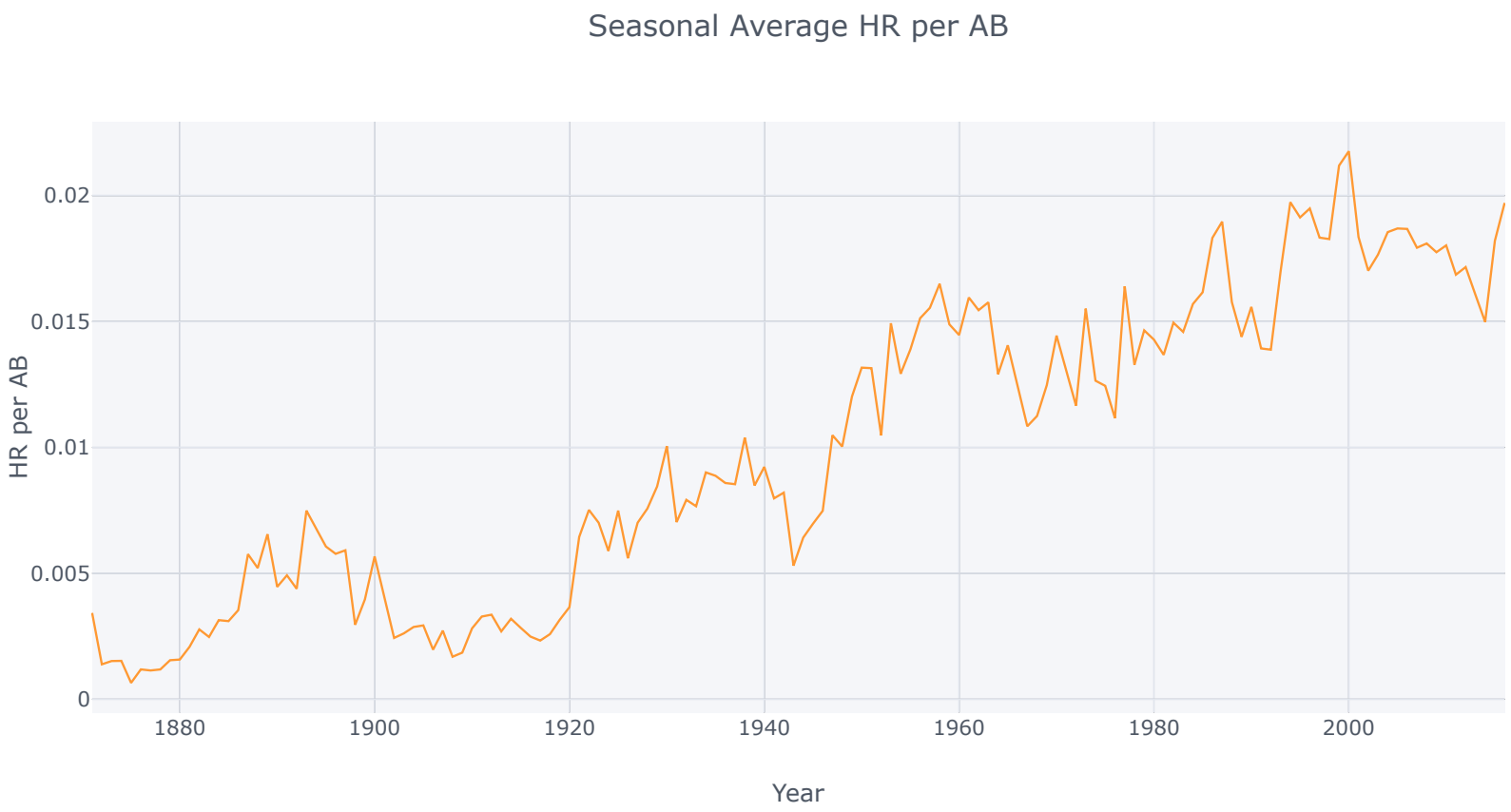


BONUS2: Using a line graph please graph the average HRs per AB (think about this) per season.

In [61]:
```
dfB2_1=pd.DataFrame({"Year":df['yearID'],"playerID":df["playerID"],"HRperAB":(df['HR']/df['AB'])})
```

In [62]:
```
dfB2_2=dfB2_1.groupby(['Year']).agg({'HRperAB':'mean'}).reset_index()
```

In [63]:
```
dfB2_2.iplot(kind='line',x='Year',y='HRperAB',xTitle='Year',yTitle='HR per AB',title="Seasonal Average HR per AB")
```



Thank you!