In [ ]:
```
# CS 5010 Project Code
#Michael Kastanowski(mrk9fx), Rehan Merchant (rm2bt), Matthew Nicklas (mmn4sv),
#Group 15
```

In [1]:
```
#Query 1
```

In [2]:
```
import requests
import csv
from numpy import *
import pandas as pd
```

In [3]:
```
bat_df =pd.read_csv("Batting.csv")#import batting
```

In [4]:
```
list(bat_df.columns)#check batting variables
```

Out[4]:
```
['playerID',
 'yearID',
 'stint',
 'teamID',
 'lgID',
 'G',
 'AB',
 'R',
 'H',
 '2B',
 '3B',
 'HR',
 'RBI',
 'SB',
 'CS',
 'BB',
 'SO',
 'IBB',
 'HBP',
 'SH',
 'SF',
 'GIDP']
```

In [5]:
```
pitch_df = pd.read_csv("Pitching.csv")#import pitching
```

In [6]:
```
list(pitch_df.columns)#check pitching variables
```

Out[6]:
```
['playerID',
 'yearID',
 'stint',
 'teamID',
 'lgID',
 'W',
 'L',
 'G',
 'GS',
 'CG',
 'SHO',
 'SV',
 'IPouts',
 'H',
 'ER',
 'HR',
 'BB',
 'SO',
```

```
'BAOpp',
'ERA',
'IBB',
'WP',
'HBP',
'BK',
'BFP',
'GF',
'R',
'SH',
'SF',
'GIDP']
```

In [7]:
```
sal_df = pd.read_csv("Salaries.csv")#import salary
```

In [8]:
```
sal_bat_1 =pd.merge(bat_df, sal_df, on =['playerID', 'yearID',  'teamID', 'lgID'
```

In [9]:
```
sal_bat_fin=sal_bat_1[['playerID', 'yearID',  'teamID', 'lgID', 'R', 'H', '2B',
```

In [10]:
```
sal_bat_fin
```

Out[10]:

	playerID	yearID	teamID	lgID	R	H	2B	3B	HR	salary
1	agostju01	1985	CHA	AL	0.0	0.0	0.0	0.0	0.0	147500
2	aguaylu01	1985	PHI	NL	27.0	46.0	7.0	3.0	6.0	237000
4	allenne01	1985	SLN	NL	0.0	0.0	0.0	0.0	0.0	750000
5	almonbi01	1985	PIT	NL	33.0	66.0	17.0	0.0	6.0	255000
6	anderla02	1985	PHI	NL	1.0	0.0	0.0	0.0	0.0	250500
...	...	...	...	...	...	...	...	...	...	...
24621	zieglbr01	2015	ARI	NL	0.0	0.0	0.0	0.0	0.0	5000000
24622	zimmejo02	2015	WAS	NL	4.0	10.0	1.0	0.0	0.0	16500000
24623	zimmery01	2015	WAS	NL	43.0	86.0	25.0	1.0	16.0	14000000
24624	zobribe01	2015	OAK	AL	39.0	63.0	20.0	2.0	6.0	7500000
24625	zuninmi01	2015	SEA	AL	28.0	61.0	11.0	0.0	11.0	523500

22749 rows × 10 columns

In [11]:
```
sal_pitch_1 =pd.merge(pitch_df, sal_df, on =['playerID', 'yearID',  'teamID', 'l
```

In [12]:
```
sal_pitch_fin =sal_pitch_1[['playerID', 'yearID',  'teamID', 'lgID', 'W', 'L'  ,'
sal_pitch_fin
```

Out[12]:

	playerID	yearID	teamID	lgID	W	L	R	H	ERA	salary
0	ackerji01	1985	TOR	AL	7	2	35	86	3.23	170000
1	agostju01	1985	CHA	AL	4	3	27	45	3.58	147500
2	alexado01	1985	TOR	AL	17	10	105	268	3.45	875000
3	allenne01	1985	SLN	NL	1	4	22	32	5.59	750000
4	anderla02	1985	PHI	NL	3	3	41	78	4.32	250500

	playerID	yearID	teamID	lgID	W	L	R	H	ERA	salary
...	...	...	...	...	...	...	...	...	...	...
11526	wrighst01	2015	BOS	AL	5	4	38	67	4.09	510500
11527	yateski01	2015	TBA	AL	1	0	18	23	7.97	512800
11528	youngch03	2015	KCA	AL	11	6	44	91	3.06	675000
11529	zieglbr01	2015	ARI	NL	0	3	17	48	1.85	5000000
11530	zimmejo02	2015	WAS	NL	13	10	89	204	3.66	16500000

11526 rows × 10 columns

In [13]:
```
sal_pitch_fin=sal_pitch_fin.rename(columns={ 'W': 'Wins', 'L':'Losses' ,'R':"Run
#rename colums
```

In [14]:
```
sal_bat_fin = sal_bat_fin.rename(columns = {'R':'Runs', 'H':'Hits', '2B': 'Doubl
#rename columns
```

In [15]:
```
sal_bat_fin.to_csv('Salary_Batting.csv',index=False) #save
```

In [16]:
```
sal_pitch_fin.to_csv('Salary_Pitching.csv', index =False)#save
```

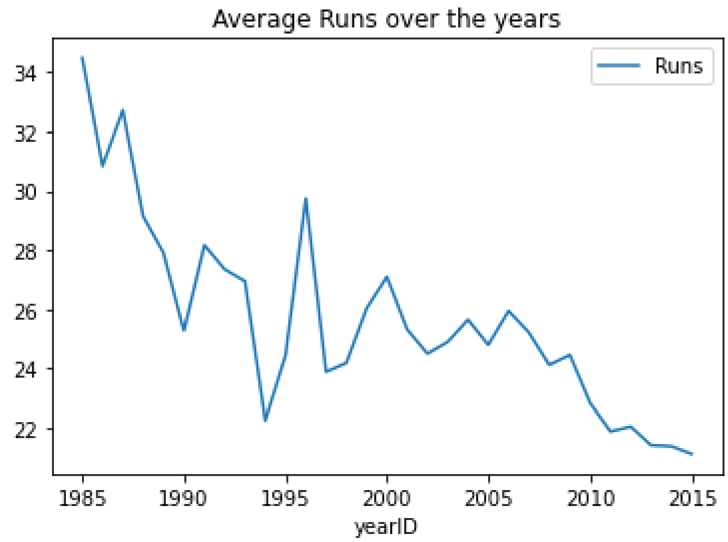
In [17]:
```
#import important packages
%matplotlib inline
import requests
import csv
import numpy as np
import pandas
import pandas as pd
```

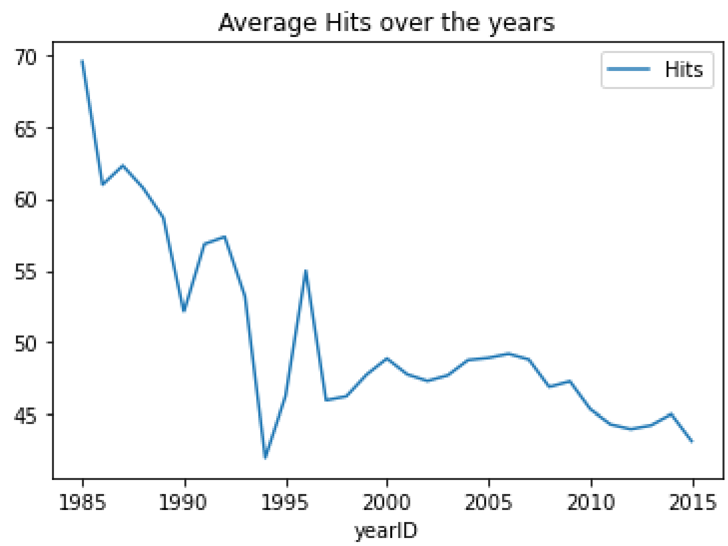
In [18]:
```
sal_bat = pd.read_csv('Salary_Batting.csv') #import batting statistics
sal_pitch = pd.read_csv('Salary_Pitching.csv') #import pitching statistics
```

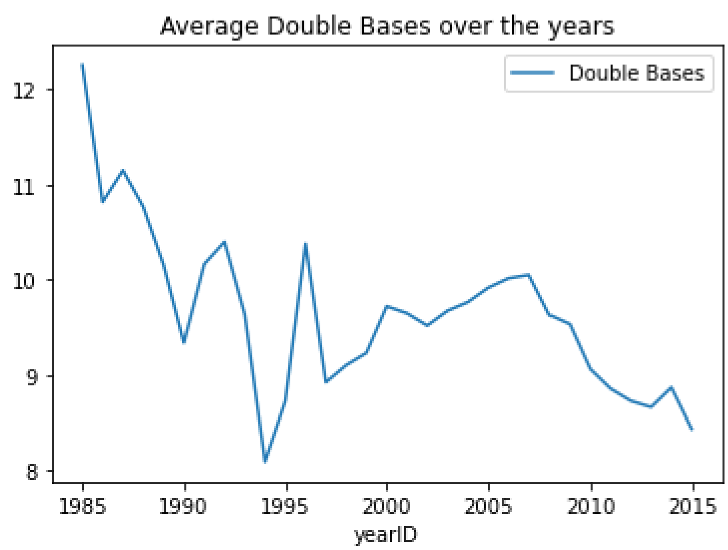
In [19]:
```
#drop null values
#sal_bat = sal_bat.dropna
#sal_pitch = sal_pitch.dropna
```

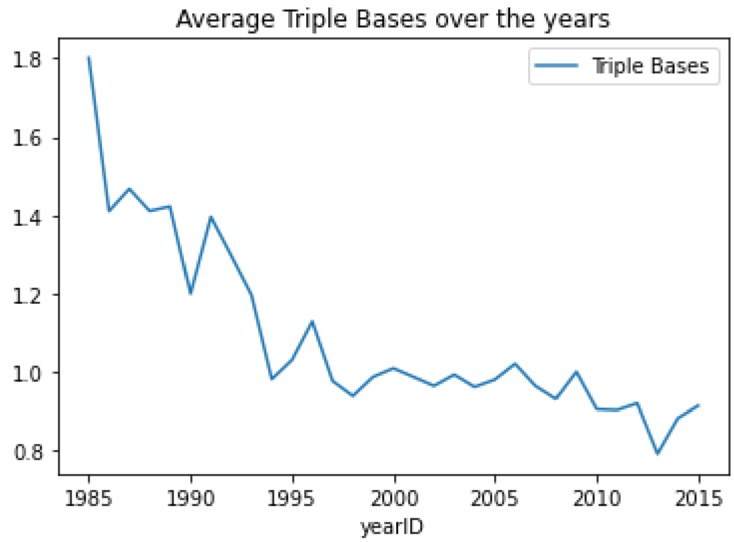
In [20]:
```
def group_avg(df, group): #used to create average df
    avg_df= df.groupby(group).mean()
    return avg_df #creates plot of yearly means

def group_reg(df, group):
    group_df = (df, group)
    return group_df

def year_plot(df): #used to plot
    for column in df:
        df.plot(use_index=True, y = column, title = ('Average ' + str(column)+ '
                #plots each column against the index(years)
```

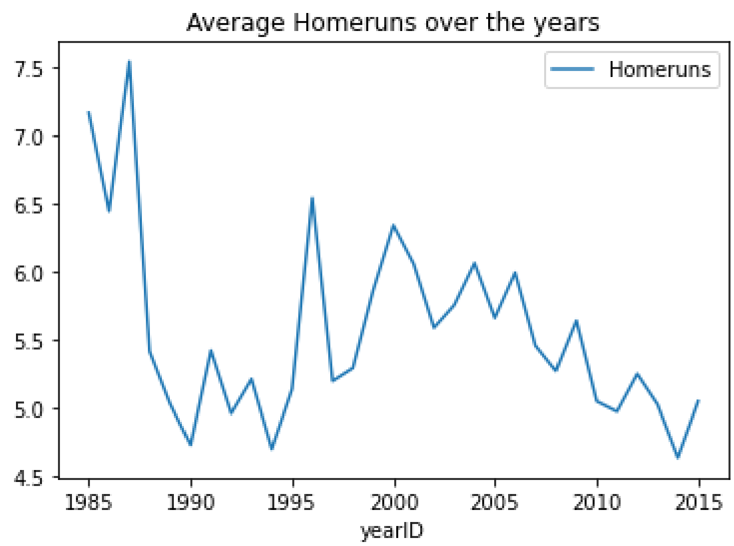
In [21]:
```
sal_bat.pipe(group_avg, 'yearID').pipe(year_plot) #plot batting statistics
```

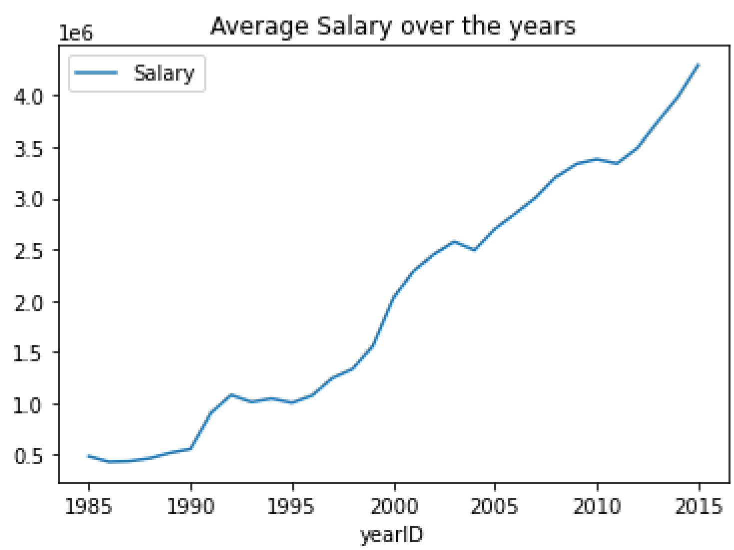

Average Runs over the years
Runs
34
32
30
28
26
24
22
1985
1990
1995
2000
2005
2010
2015
yearID


Average Hits over the years
Hits
70
65
60
55
50
45
1985
1990
1995
2000
2005
2010
2015
yearID


Average Double Bases over the years
Double Bases
12
11
10
9
8
1985
1990
1995
2000
2005
2010
2015
yearID

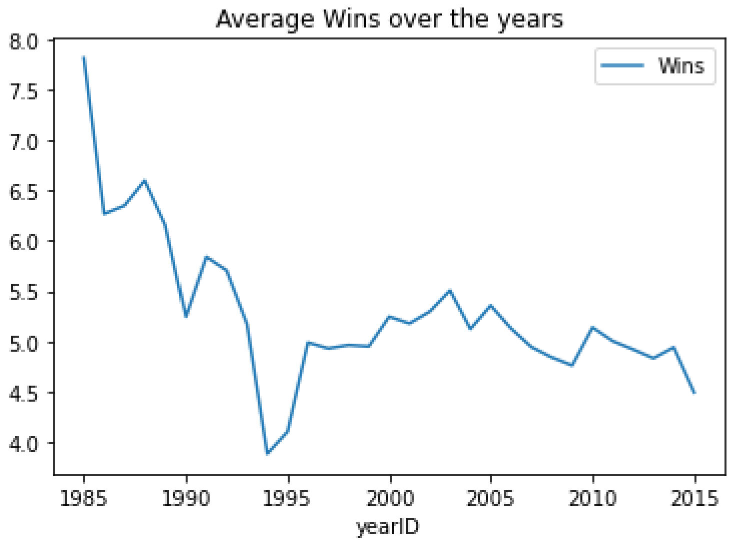


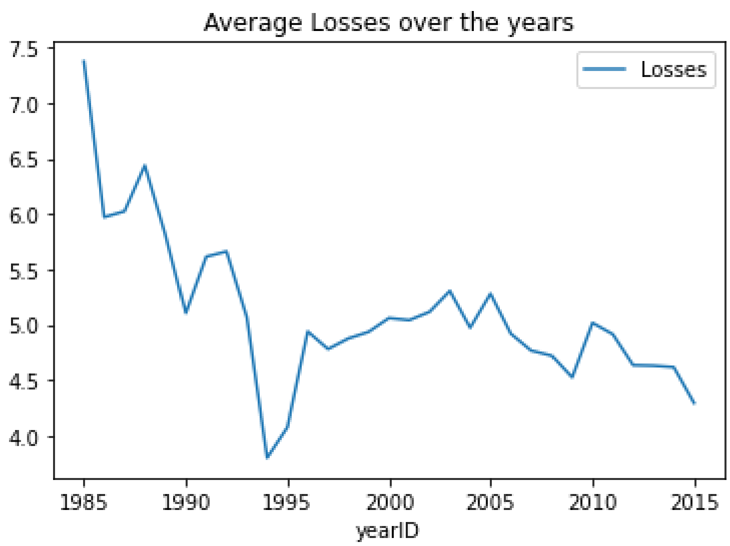


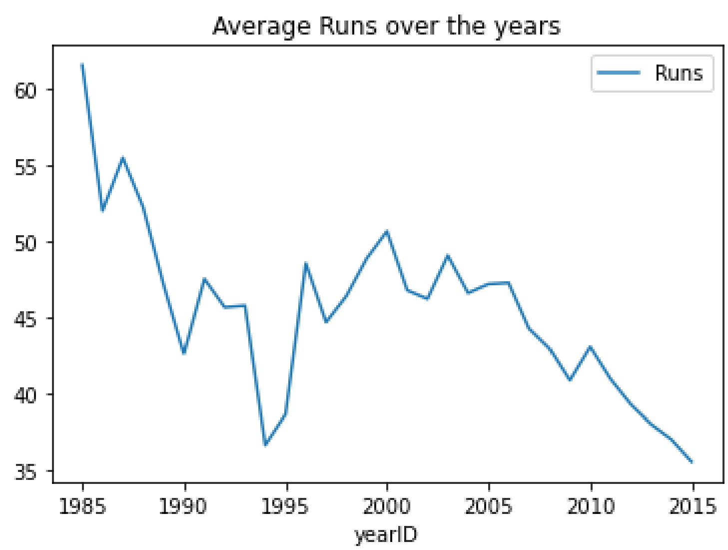


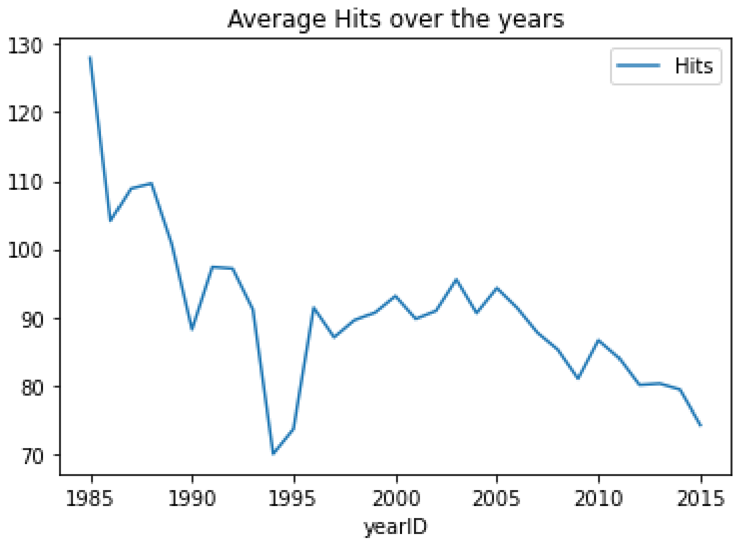
In [22]:
```
# Average salary increases over the years but the average batting statistic drop
```

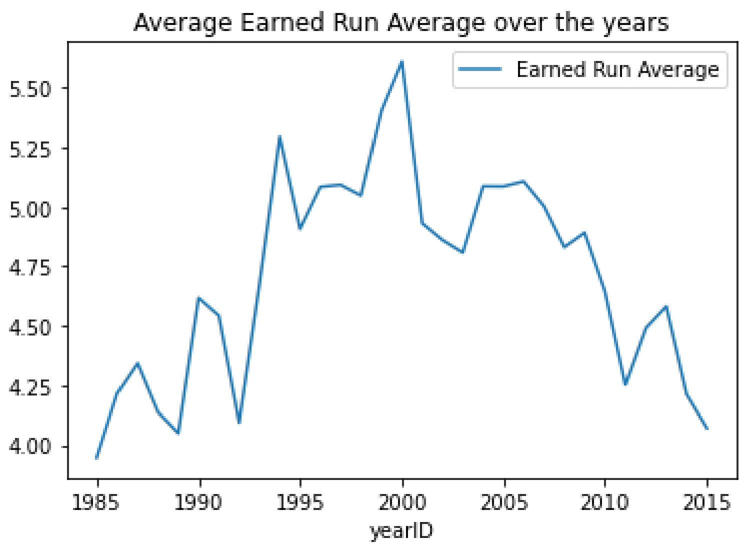
In [23]:
```
sal_pitch.pipe(group_avg, 'yearID').pipe(year_plot) #plot pitching statistics
```

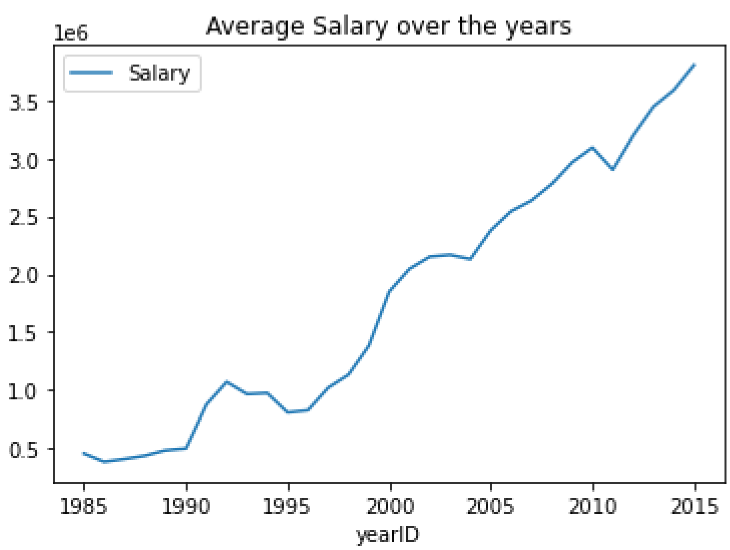

Average Wins over the years
Wins
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
4.0
1985
1990
1995
2000
2005
2010
2015
yearID


Average Losses over the years
Losses
7.5
7.0
6.5
6.0
5.5
5.0
4.5
4.0
1985
1990
1995
2000
2005
2010
2015
yearID


Average Runs over the years
Runs
60
55
50
45
40
35
1985
1990
1995
2000
2005
2010
2015
yearID







In [24]:
```
# Average salary increases over the years but the average pitchingg statistic dr
```

In [25]:
```
#Query 2
```

In [26]:
```
import matplotlib.pyplot as plt
from IPython.display import display
```

In [27]:
```
# Which active team has the most World Series Wins
```

In [28]:
```
#importing data in Pandas DF
batting = pd.read_csv('Batting.csv')
salaries = pd.read_csv('Salaries.csv')
teams = pd.read_csv('Teams.csv')
teamsFranchise = pd.read_csv('TeamsFranchises.csv')
```

In [29]:
```
#removing batting stats before 1985 since we dont have salary data for before 19
indexBatting = batting[batting['yearID']<1985].index
batting.drop(indexBatting,inplace=True)
```

In [30]:
```
#which team has the most chamiponships out of the active teams today
#only WS winners
indexteams = teams[teams['WSWin']!='Y'].index
teams.drop(indexteams,inplace=True)
```

In [31]:
```
#keep only active teams
indexFranchise = teamsFranchise[teamsFranchise['active']!='Y'].index
teamsFranchise.drop(indexFranchise,inplace=True)

i = teamsFranchise.franchID.isin(teams.franchID)
print(i)
WSwinningteams = teamsFranchise[i]
WSwinningteams=WSwinningteams.drop('NAassoc',1)
WSwinningteams=WSwinningteams.drop('active',1)
WSwinningteams= WSwinningteams.set_index('franchID')
print(WSwinningteams)
```

```
1       True
2       True
4       True
5       True
13      True
25      True
28      True
29      True
31      True
37     False
40      True
43      True
46     False
53      True
56      True
61     False
62      True
71      True
74      True
75      True
80      True
83      True
92     False
93     False
94      True
99      True
103    False
104    False
107     True
118    False
Name: franchID, dtype: bool
                      franchName
```

```
franchID
ANA       Los Angeles Angels of Anaheim
ARI                Arizona Diamondbacks
ATL                      Atlanta Braves
BAL                   Baltimore Orioles
BOS                      Boston Red Sox
CHC                        Chicago Cubs
CHW                   Chicago White Sox
CIN                     Cincinnati Reds
CLE                   Cleveland Indians
DET                      Detroit Tigers
FLA                     Florida Marlins
KCR                  Kansas City Royals
LAD                 Los Angeles Dodgers
MIN                     Minnesota Twins
NYM                       New York Mets
NYY                    New York Yankees
OAK                   Oakland Athletics
PHI               Philadelphia Phillies
PIT                  Pittsburgh Pirates
SFG                San Francisco Giants
STL                 St. Louis Cardinals
TOR                   Toronto Blue Jays
```

In [32]:

```
#groupby franchise ID
numWSwins =teams.groupby('franchID').size()
print(numWSwins)

#assigning WSWins to numWSwins
WSWinsAndTeams = WSwinningteams.assign(WSWins = numWSwins)
print(WSWinsAndTeams)
```

```
franchID
ANA     1
ARI     1
ATL     3
BAL     3
BOS     8
CHC     2
CHW     3
CIN     5
CLE     2
DET     4
DTN     1
FLA     2
KCR     2
LAD     6
MIN     3
NYM     2
NYY    27
OAK     9
PHI     2
PIT     5
PRO     1
SFG    10
STL    12
TOR     2
dtype: int64
                              franchName  WSWins
franchID
ANA       Los Angeles Angels of Anaheim       1
ARI                Arizona Diamondbacks       1
ATL                      Atlanta Braves       3
BAL                   Baltimore Orioles       3
BOS                      Boston Red Sox       8
```

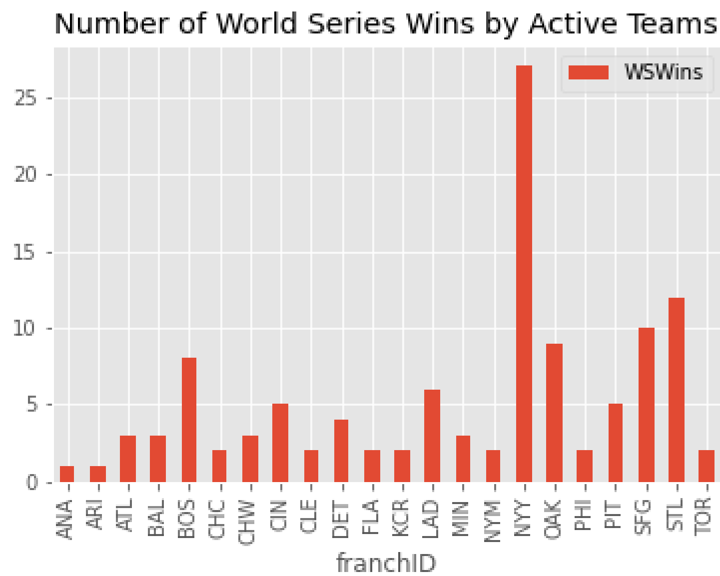
```
CHC             Chicago Cubs        2
CHW         Chicago White Sox       3
CIN           Cincinnati Reds       5
CLE         Cleveland Indians       2
DET            Detroit Tigers       4
FLA           Florida Marlins       2
KCR        Kansas City Royals       2
LAD       Los Angeles Dodgers       6
MIN           Minnesota Twins       3
NYM             New York Mets       2
NYY          New York Yankees      27
OAK         Oakland Athletics       9
PHI     Philadelphia Phillies       2
PIT        Pittsburgh Pirates       5
SFG      San Francisco Giants      10
STL       St. Louis Cardinals      12
TOR         Toronto Blue Jays       2
```

In [110…

```
WSWinsAndTeams.plot.bar()
plt.style.use('ggplot')
plt.title('Number of World Series Wins by Active Teams')
```

Out[110…

```
Text(0.5, 1.0, 'Number of World Series Wins by Active Teams')
```



In [34]:

```
#Query 3
```

In [35]:

```
# Which team won the world championship each year from 1985-2015 and how much sa
```

In [36]:

```
#WS winning teams 1985 to 2015
print(teams)
teams1985_2015 = teams
teams1985_2015index = teams1985_2015[teams1985_2015['yearID']<1985].index
teams1985_2015.drop(teams1985_2015index, inplace = True)
print(teams1985_2015)
print(salaries)
```

```
     yearID lgID teamID franchID divID  Rank    G  Ghome   W   L  ...     DP  \
147    1884   NL    PRO      PRO   NaN     1  114    NaN  84  28  ...    NaN
186    1886   AA    SL4      STL   NaN     1  139    NaN  93  46  ...    NaN
195    1887   NL    DTN      DTN   NaN     1  127    NaN  79  45  ...    NaN
215    1888   NL    NY1      SFG   NaN     1  138    NaN  84  47  ...    NaN
231    1889   NL    NY1      SFG   NaN     1  131    NaN  83  43  ...    NaN
```

```
...      ...  ...     ...       ...   ...    ...  ...     ...  ..  ..  ...      ...
2680    2011   NL     SLN       STL     C      2  162    81.0  90  72  ...    167.0
2709    2012   NL     SFN       SFG     W      1  162    81.0  94  68  ...    134.0
2718    2013   AL     BOS       BOS     E      1  162    81.0  97  65  ...    142.0
2769    2014   NL     SFN       SFG     W      2  162    81.0  88  74  ...    155.0
2775    2015   AL     KCA       KCR     C      1  162    81.0  95  67  ...    138.0

         FP                   name                    park  attendance  BPF  \
147   0.910       Providence Grays  Messer Street Grounds          NaN   99
186   0.910      St. Louis Browns     Sportsman's Park I          NaN  105
195   0.920     Detroit Wolverines       Recreation Park          NaN  104
215   0.920        New York Giants        Polo Grounds I          NaN   99
231   0.920        New York Giants       Polo Grounds II          NaN  104
...     ...                    ...                    ...          ...  ...
2680  0.982  St. Louis Cardinals     Busch Stadium III    3093954.0   95
2709  0.981  San Francisco Giants            AT&T Park    3377371.0   88
2718  0.987       Boston Red Sox        Fenway Park II    2833333.0  102
2769  0.984  San Francisco Giants            AT&T Park    3368697.0   95
2775  0.985    Kansas City Royals      Kauffman Stadium   2708549.0  104

      PPF  teamIDBR  teamIDlahman45  teamIDretro
147    96       PRO             PRO          PRO
186   100       STL             SL4          SL4
195   100       DTN             DTN          DTN
215    96       NYG             NY1          NY1
231   101       NYG             NY1          NY1
...   ...       ...             ...          ...
2680   94       STL             SLN          SLN
2709   88       SFG             SFN          SFN
2718  102       BOS             BOS          BOS
2769   95       SFG             SFN          SFN
2775  103       KCR             KCA          KCA

[116 rows x 48 columns]
      yearID lgID teamID franchID divID  Rank    G  Ghome    W   L  ...  \
1927    1985   AL    KCA      KCR     W     1  162   82.0   91  71  ...
1959    1986   NL    NYN      NYM     E     1  162   81.0  108  54  ...
1981    1987   AL    MIN      MIN     W     1  162   81.0   85  77  ...
2006    1988   NL    LAN      LAD     W     1  162   81.0   94  67  ...
2038    1989   AL    OAK      OAK     W     1  162   81.0   99  63  ...
2053    1990   NL    CIN      CIN     W     1  162   81.0   91  71  ...
2085    1991   AL    MIN      MIN     W     1  162   81.0   95  67  ...
2124    1992   AL    TOR      TOR     E     1  162   81.0   96  66  ...
2152    1993   AL    TOR      TOR     E     1  162   81.0   95  67  ...
2181    1995   NL    ATL      ATL     E     1  144   72.0   90  54  ...
2226    1996   AL    NYA      NYY     E     1  162   80.0   92  70  ...
2247    1997   NL    FLO      FLA     E     2  162   81.0   92  70  ...
2283    1998   AL    NYA      NYY     E     1  162   81.0  114  48  ...
2313    1999   AL    NYA      NYY     E     1  162   81.0   98  64  ...
2343    2000   AL    NYA      NYY     E     1  161   80.0   87  74  ...
2356    2001   NL    ARI      ARI     W     1  162   81.0   92  70  ...
2385    2002   AL    ANA      ANA     W     2  162   81.0   99  63  ...
2426    2003   NL    FLO      FLA     E     2  162   81.0   91  71  ...
2449    2004   AL    BOS      BOS     E     2  162   81.0   98  64  ...
2479    2005   AL    CHA      CHW     C     1  162   81.0   99  63  ...
2530    2006   NL    SLN      STL     C     1  161   80.0   83  78  ...
2538    2007   AL    BOS      BOS     E     1  162   81.0   96  66  ...
2585    2008   NL    PHI      PHI     E     1  162   81.0   92  70  ...
2612    2009   AL    NYA      NYY     E     1  162   81.0  103  59  ...
2649    2010   NL    SFN      SFG     W     1  162   81.0   92  70  ...
2680    2011   NL    SLN      STL     C     2  162   81.0   90  72  ...
2709    2012   NL    SFN      SFG     W     1  162   81.0   94  68  ...
2718    2013   AL    BOS      BOS     E     1  162   81.0   97  65  ...
2769    2014   NL    SFN      SFG     W     2  162   81.0   88  74  ...
2775    2015   AL    KCA      KCR     C     1  162   81.0   95  67  ...
```

	DP	FP	name	park	\
1927	160.0	0.980	Kansas City Royals	Royals Stadium	
1959	145.0	0.970	New York Mets	Shea Stadium	
1981	147.0	0.980	Minnesota Twins	Hubert H Humphrey Metrodome	
2006	126.0	0.970	Los Angeles Dodgers	Dodger Stadium	
2038	159.0	0.970	Oakland Athletics	Oakland Coliseum	
2053	126.0	0.980	Cincinnati Reds	Riverfront Stadium	
2085	161.0	0.980	Minnesota Twins	Hubert H Humphrey Metrodome	
2124	109.0	0.980	Toronto Blue Jays	Skydome	
2152	144.0	0.980	Toronto Blue Jays	Skydome	
2181	113.0	0.980	Atlanta Braves	Atlanta-Fulton County Stadium	
2226	146.0	0.980	New York Yankees	Yankee Stadium II	
2247	167.0	0.980	Florida Marlins	Joe Robbie Stadium	
2283	146.0	0.980	New York Yankees	Yankee Stadium II	
2313	130.0	0.980	New York Yankees	Yankee Stadium II	
2343	132.0	0.981	New York Yankees	Yankee Stadium II	
2356	148.0	0.986	Arizona Diamondbacks	Bank One Ballpark	
2385	151.0	0.986	Anaheim Angels	Edison International Field	
2426	162.0	0.987	Florida Marlins	Pro Player Stadium	
2449	129.0	0.981	Boston Red Sox	Fenway Park II	
2479	166.0	0.985	Chicago White Sox	U.S. Cellular Field	
2530	170.0	0.984	St. Louis Cardinals	Busch Stadium III	
2538	145.0	0.986	Boston Red Sox	Fenway Park II	
2585	142.0	0.985	Philadelphia Phillies	Citizens Bank Park	
2612	131.0	0.985	New York Yankees	Yankee Stadium III	
2649	110.0	0.988	San Francisco Giants	AT&T Park	
2680	167.0	0.982	St. Louis Cardinals	Busch Stadium III	
2709	134.0	0.981	San Francisco Giants	AT&T Park	
2718	142.0	0.987	Boston Red Sox	Fenway Park II	
2769	155.0	0.984	San Francisco Giants	AT&T Park	
2775	138.0	0.985	Kansas City Royals	Kauffman Stadium	

	attendance	BPF	PPF	teamIDBR	teamIDlahman45	teamIDretro
1927	2162717.0	100	100	KCR	KCA	KCA
1959	2767601.0	98	96	NYM	NYN	NYN
1981	2081976.0	103	103	MIN	MIN	MIN
2006	2980262.0	98	97	LAD	LAN	LAN
2038	2667225.0	97	95	OAK	OAK	OAK
2053	2400892.0	105	105	CIN	CIN	CIN
2085	2293842.0	105	104	MIN	MIN	MIN
2124	4028318.0	105	104	TOR	TOR	TOR
2152	4057947.0	101	100	TOR	TOR	TOR
2181	2561831.0	103	102	ATL	ATL	ATL
2226	2250877.0	101	100	NYY	NYA	NYA
2247	2364387.0	95	96	FLA	FLO	FLO
2283	2955193.0	97	95	NYY	NYA	NYA
2313	3292736.0	98	97	NYY	NYA	NYA
2343	3055435.0	99	98	NYY	NYA	NYA
2356	2736451.0	108	107	ARI	ARI	ARI
2385	2305547.0	100	99	ANA	ANA	ANA
2426	1303215.0	98	98	FLA	FLO	FLO
2449	2837294.0	106	105	BOS	BOS	BOS
2479	2342833.0	103	103	CHW	CHA	CHA
2530	3407104.0	99	99	STL	SLN	SLN
2538	2970755.0	106	105	BOS	BOS	BOS
2585	3422583.0	103	102	PHI	PHI	PHI
2612	3719358.0	105	103	NYY	NYA	NYA
2649	3037443.0	101	101	SFG	SFN	SFN
2680	3093954.0	95	94	STL	SLN	SLN
2709	3377371.0	88	88	SFG	SFN	SFN
2718	2833333.0	102	102	BOS	BOS	BOS
2769	3368697.0	95	95	SFG	SFN	SFN
2775	2708549.0	104	103	KCR	KCA	KCA

```
[30 rows x 48 columns]
       yearID teamID lgID   playerID    salary
0        1985    ATL   NL  barkele01    870000
1        1985    ATL   NL  bedrost01    550000
2        1985    ATL   NL  benedbr01    545000
3        1985    ATL   NL   campri01    633333
4        1985    ATL   NL  ceronri01    625000
...       ...    ...  ...        ...       ...
25570    2015    WAS   NL  treinbl01    512800
25571    2015    WAS   NL  ugglada01    507500
25572    2015    WAS   NL  werthja01  21000000
25573    2015    WAS   NL  zimmejo02  16500000
25574    2015    WAS   NL  zimmery01  14000000

[25575 rows x 5 columns]
```

In [37]:

```
# legue avg salaries by year
groupSalaries_avg = salaries.groupby('yearID').mean()
print(groupSalaries_avg)
```

```
              salary
yearID
1985    4.762994e+05
1986    4.171470e+05
1987    4.347295e+05
1988    4.531711e+05
1989    5.063231e+05
1990    5.119737e+05
1991    8.949612e+05
1992    1.047521e+06
1993    9.769666e+05
1994    1.049589e+06
1995    9.649791e+05
1996    1.027909e+06
1997    1.218687e+06
1998    1.280845e+06
1999    1.485317e+06
2000    1.992985e+06
2001    2.279841e+06
2002    2.392527e+06
2003    2.573473e+06
2004    2.491776e+06
2005    2.633831e+06
2006    2.834521e+06
2007    2.941436e+06
2008    3.136517e+06
2009    3.277647e+06
2010    3.278747e+06
2011    3.318838e+06
2012    3.458421e+06
2013    3.723344e+06
2014    3.980446e+06
2015    4.301276e+06
```

In [38]:

```
# legue avg salaries by year
groupSalaries_avg = salaries.groupby('yearID').mean()
print(groupSalaries_avg)
```

```
              salary
yearID
1985    4.762994e+05
1986    4.171470e+05
1987    4.347295e+05
1988    4.531711e+05
1989    5.063231e+05
```

```
1990    5.119737e+05
1991    8.949612e+05
1992    1.047521e+06
1993    9.769666e+05
1994    1.049589e+06
1995    9.649791e+05
1996    1.027909e+06
1997    1.218687e+06
1998    1.280845e+06
1999    1.485317e+06
2000    1.992985e+06
2001    2.279841e+06
2002    2.392527e+06
2003    2.573473e+06
2004    2.491776e+06
2005    2.633831e+06
2006    2.834521e+06
2007    2.941436e+06
2008    3.136517e+06
2009    3.277647e+06
2010    3.278747e+06
2011    3.318838e+06
2012    3.458421e+06
2013    3.723344e+06
2014    3.980446e+06
2015    4.301276e+06
```

In [47]:

```
#grouping salaries by year and team
grouped_multiple = salaries.groupby(['yearID', 'teamID']).agg({'salary':['mean']
grouped_multiple.columns = grouped_multiple.columns.droplevel(-1)
print(grouped_multiple)
```

```
                      salary
yearID teamID
1985   ATL     6.730455e+05
       BAL     5.254869e+05
       BOS     4.359024e+05
       CAL     5.152819e+05
       CHA     4.688656e+05
...                     ...
2015   SLN     4.586212e+06
       TBA     2.224870e+06
       TEX     4.791426e+06
       TOR     4.519696e+06
       WAS     5.365085e+06

[888 rows x 1 columns]
```

In [48]:

```
#merge both dataframes on yearID and teamID
avg_salary_WSWinning_teams =pd.merge(teams1985_2015, grouped_multiple, on =['yea
```

In [49]:

```
#merge tables on yearID
avg_salary_WSWinning_teams_avg_league_salary = pd.merge(avg_salary_WSWinning_tea
print(avg_salary_WSWinning_teams_avg_league_salary)
```

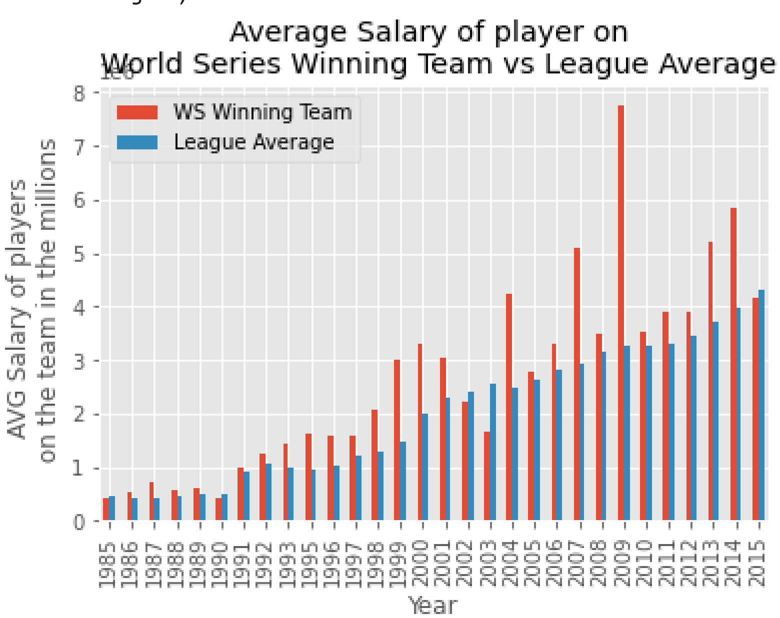
```
    yearID lgID teamID franchID divID  Rank    G  Ghome    W   L  ...  \
0     1985   AL    KCA      KCR     W     1  162   82.0   91  71  ...
1     1986   NL    NYN      NYM     E     1  162   81.0  108  54  ...
2     1987   AL    MIN      MIN     W     1  162   81.0   85  77  ...
3     1988   NL    LAN      LAD     W     1  162   81.0   94  67  ...
4     1989   AL    OAK      OAK     W     1  162   81.0   99  63  ...
5     1990   NL    CIN      CIN     W     1  162   81.0   91  71  ...
6     1991   AL    MIN      MIN     W     1  162   81.0   95  67  ...
7     1992   AL    TOR      TOR     E     1  162   81.0   96  66  ...
```

```
8     1993   AL    TOR      TOR     E     1  162   81.0   95  67  ...
9     1995   NL    ATL      ATL     E     1  144   72.0   90  54  ...
10    1996   AL    NYA      NYY     E     1  162   80.0   92  70  ...
11    1997   NL    FLO      FLA     E     2  162   81.0   92  70  ...
12    1998   AL    NYA      NYY     E     1  162   81.0  114  48  ...
13    1999   AL    NYA      NYY     E     1  162   81.0   98  64  ...
14    2000   AL    NYA      NYY     E     1  161   80.0   87  74  ...
15    2001   NL    ARI      ARI     W     1  162   81.0   92  70  ...
16    2002   AL    ANA      ANA     W     2  162   81.0   99  63  ...
17    2003   NL    FLO      FLA     E     2  162   81.0   91  71  ...
18    2004   AL    BOS      BOS     E     2  162   81.0   98  64  ...
19    2005   AL    CHA      CHW     C     1  162   81.0   99  63  ...
20    2006   NL    SLN      STL     C     1  161   80.0   83  78  ...
21    2007   AL    BOS      BOS     E     1  162   81.0   96  66  ...
22    2008   NL    PHI      PHI     E     1  162   81.0   92  70  ...
23    2009   AL    NYA      NYY     E     1  162   81.0  103  59  ...
24    2010   NL    SFN      SFG     W     1  162   81.0   92  70  ...
25    2011   NL    SLN      STL     C     2  162   81.0   90  72  ...
26    2012   NL    SFN      SFG     W     1  162   81.0   94  68  ...
27    2013   AL    BOS      BOS     E     1  162   81.0   97  65  ...
28    2014   NL    SFN      SFG     W     2  162   81.0   88  74  ...
29    2015   AL    KCA      KCR     C     1  162   81.0   95  67  ...

                     name                            park attendance  BPF  PPF  \
0      Kansas City Royals                  Royals Stadium  2162717.0  100  100
1         New York Mets                      Shea Stadium  2767601.0   98   96
2       Minnesota Twins     Hubert H Humphrey Metrodome  2081976.0  103  103
3     Los Angeles Dodgers                  Dodger Stadium  2980262.0   98   97
4       Oakland Athletics                Oakland Coliseum  2667225.0   97   95
5        Cincinnati Reds               Riverfront Stadium  2400892.0  105  105
6       Minnesota Twins     Hubert H Humphrey Metrodome  2293842.0  105  104
7     Toronto Blue Jays                           Skydome  4028318.0  105  104
8     Toronto Blue Jays                           Skydome  4057947.0  101  100
9        Atlanta Braves  Atlanta-Fulton County Stadium  2561831.0  103  102
10     New York Yankees               Yankee Stadium II  2250877.0  101  100
11      Florida Marlins               Joe Robbie Stadium  2364387.0   95   96
12     New York Yankees               Yankee Stadium II  2955193.0   97   95
13     New York Yankees               Yankee Stadium II  3292736.0   98   97
14     New York Yankees               Yankee Stadium II  3055435.0   99   98
15  Arizona Diamondbacks             Bank One Ballpark  2736451.0  108  107
16       Anaheim Angels     Edison International Field  2305547.0  100   99
17      Florida Marlins               Pro Player Stadium  1303215.0   98   98
18       Boston Red Sox                  Fenway Park II  2837294.0  106  105
19    Chicago White Sox           U.S. Cellular Field  2342833.0  103  103
20   St. Louis Cardinals              Busch Stadium III  3407104.0   99   99
21       Boston Red Sox                  Fenway Park II  2970755.0  106  105
22 Philadelphia Phillies             Citizens Bank Park  3422583.0  103  102
23     New York Yankees              Yankee Stadium III  3719358.0  105  103
24 San Francisco Giants                      AT&T Park  3037443.0  101  101
25   St. Louis Cardinals              Busch Stadium III  3093954.0   95   94
26 San Francisco Giants                      AT&T Park  3377371.0   88   88
27       Boston Red Sox                  Fenway Park II  2833333.0  102  102
28 San Francisco Giants                      AT&T Park  3368697.0   95   95
29    Kansas City Royals                Kauffman Stadium  2708549.0  104  103

   teamIDBR  teamIDlahman45  teamIDretro      salary_x      salary_y
0       KCR             KCA          KCA  4.236900e+05  4.762994e+05
1       NYM             NYN          NYN  5.497755e+05  4.171470e+05
2       MIN             MIN          MIN  7.108333e+05  4.347295e+05
3       LAD             LAN          LAN  5.616838e+05  4.531711e+05
4       OAK             OAK          OAK  6.245228e+05  5.063231e+05
5       CIN             CIN          CIN  4.226471e+05  5.119737e+05
6       MIN             MIN          MIN  9.734097e+05  8.949612e+05
7       TOR             TOR          TOR  1.244130e+06  1.047521e+06
8       TOR             TOR          TOR  1.432702e+06  9.769666e+05
```

```
9         ATL              ATL           ATL   1.628808e+06  9.649791e+05
10        NYY              NYA           NYA   1.593876e+06  1.027909e+06
11        FLA              FLO           FLO   1.570726e+06  1.218687e+06
12        NYY              NYA           NYA   2.087715e+06  1.280845e+06
13        NYY              NYA           NYA   2.990840e+06  1.485317e+06
14        NYY              NYA           NYA   3.297795e+06  1.992985e+06
15        ARI              ARI           ARI   3.038679e+06  2.279841e+06
16        ANA              ANA           ANA   2.204345e+06  2.392527e+06
17        FLA              FLO           FLO   1.648333e+06  2.573473e+06
18        BOS              BOS           BOS   4.243283e+06  2.491776e+06
19        CHW              CHA           CHA   2.784370e+06  2.633831e+06
20        STL              SLN           SLN   3.292273e+06  2.834521e+06
21        BOS              BOS           BOS   5.108079e+06  2.941436e+06
22        PHI              PHI           PHI   3.495710e+06  3.136517e+06
23        NYY              NYA           NYA   7.748046e+06  3.277647e+06
24        SFG              SFN           SFN   3.522905e+06  3.278747e+06
25        STL              SLN           SLN   3.904947e+06  3.318838e+06
26        SFG              SFN           SFN   3.920689e+06  3.458421e+06
27        BOS              BOS           BOS   5.225172e+06  3.723344e+06
28        SFG              SFN           SFN   5.839649e+06  3.980446e+06
29        KCR              KCA           KCA   4.152112e+06  4.301276e+06

[30 rows x 50 columns]
```

In [50]:
```
avg_salary_WSWinning_teams_avg_league_salary_plot = avg_salary_WSWinning_teams_a
avg_salary_WSWinning_teams_avg_league_salary_plot.legend(["WS Winning Team", "Le
plt.style.use('ggplot')
plt.ylabel('AVG Salary of players \n on the team in the millions')
plt.xlabel('Year')
title = 'Average Salary of player on \n World Series Winning Team vs League Aver
plt.title(title)
```

Out[50]:
```
Text(0.5, 1.0, 'Average Salary of player on \n World Series Winning Team vs Leag
ue Average')
```



In [51]:
```
#create winners table
WinnerTable = avg_salary_WSWinning_teams_avg_league_salary[['yearID','franchID']
display(WinnerTable)
```

	yearID	franchID
0	1985	KCR

	yearID	franchID
1	1986	NYM
2	1987	MIN
3	1988	LAD
4	1989	OAK
5	1990	CIN
6	1991	MIN
7	1992	TOR
8	1993	TOR
9	1995	ATL
10	1996	NYY
11	1997	FLA
12	1998	NYY
13	1999	NYY
14	2000	NYY
15	2001	ARI
16	2002	ANA
17	2003	FLA
18	2004	BOS
19	2005	CHW
20	2006	STL
21	2007	BOS
22	2008	PHI
23	2009	NYY
24	2010	SFG
25	2011	STL
26	2012	SFG
27	2013	BOS
28	2014	SFG
29	2015	KCR

In [52]:

```
#Query 4
```

In [84]:

```
import csv
import pandas as pd
import numpy as np
import os
```

```
reg_pitch = pd.read_csv('Pitching.csv')
```

In [85]:
```
post_pitch = pd.read_csv('PitchingPost.csv')
```

In [86]:
```
reg_pitch.head()
len(reg_pitch)
```

Out[86]: 44139

In [87]:
```
len(post_pitch)
```

Out[87]: 5109

In [88]:
```
print(reg_pitch.columns.values)
print(post_pitch.columns.values)
```

```
['playerID' 'yearID' 'stint' 'teamID' 'lgID' 'W' 'L' 'G' 'GS' 'CG' 'SHO'
 'SV' 'IPouts' 'H' 'ER' 'HR' 'BB' 'SO' 'BAOpp' 'ERA' 'IBB' 'WP' 'HBP' 'BK'
 'BFP' 'GF' 'R' 'SH' 'SF' 'GIDP']
['playerID' 'yearID' 'round' 'teamID' 'lgID' 'W' 'L' 'G' 'GS' 'CG' 'SHO'
 'SV' 'IPouts' 'H' 'ER' 'HR' 'BB' 'SO' 'BAOpp' 'ERA' 'IBB' 'WP' 'HBP' 'BK'
 'BFP' 'GF' 'R' 'SH' 'SF' 'GIDP']
```

In [89]:
```
post_colnames = []
reg_colnames = []
for i in post_pitch.columns.values:
    post_colnames.append(i)
for i in reg_pitch.columns.values:
    reg_colnames.append(i)

post_colnames == reg_colnames is True #check if column names are the same. Since
```

Out[89]: False

In [90]:
```
differences = []
for i in post_colnames:
    if i not in reg_colnames:
        differences.append(i)
for i in reg_colnames:
    if i not in post_colnames:
        differences.append(i)
differences

# All this code is meant to do is tell which of the columns are different betwee
# this query, we will drop these two columns. This is sort of an optional step j
```

Out[90]: ['round', 'stint']

In [91]:
```
post_new = post_pitch[['yearID', 'teamID', 'BB', 'ERA']]
reg_new = reg_pitch[['yearID', 'teamID', 'BB', 'ERA']] # In this selection, we k
# These are two important factors in determining a pitcher's efficiency: ERA is
# batters will go to first base on a walk (four balls).
```

In [92]:
```
post_new['REG/POST'] = 'POST'
reg_new['REG/POST'] = 'REG'
# Add new column to differentiate between post season and preseason statistics b
```

```
<ipython-input-92-1ff8f2556536>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy
  post_new['REG/POST'] = 'POST'
<ipython-input-92-1ff8f2556536>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy
  reg_new['REG/POST'] = 'REG'
```

In [93]:

```
post_new=post_new.dropna()
reg_new=reg_new.dropna() # drop NA values from the categories that we need data
```

In [94]:

```
post_new.max()
```

Out[94]:

```
yearID        2015
teamID         WS1
BB              32
ERA            inf
REG/POST      POST
dtype: object
```

In [95]:

```
post_new.sort_values(by='ERA')
```

Out[95]:

	yearID	teamID	BB	ERA	REG/POST
1958	1990	PIT	0	0.0	POST
2183	1995	CLE	2	0.0	POST
2184	1995	CLE	0	0.0	POST
2186	1995	CLE	2	0.0	POST
2188	1995	CLE	1	0.0	POST
...	...	...	...	...	...
5050	2015	KCA	0	108.0	POST
4829	2014	DET	0	108.0	POST
1389	1979	PIT	2	108.0	POST
4402	2011	ARI	2	108.0	POST
4991	2015	NYN	0	inf	POST

4841 rows × 5 columns

In [96]:

```
reg_new=reg_new[reg_new.ERA != 'inf'] # drop rows with inf ERA
```

In [97]:

```
post_new=post_new[post_new.ERA != 'inf']
post_new=post_new.drop(4991)
```

In [98]:

```
print('Since 1884, the average MLB regular season ERA is: ', reg_new['ERA'].mean
print('Since 1884, the average MLB post season ERA is: ', post_new['ERA'].mean()
```

```
Since 1884, the average MLB regular season ERA is: 5.070029058548439
Since 1884, the average MLB post season ERA is: 4.7373822314049745
```

In [99]:

```
print('Since 1884, the average MLB regular season BB (walks) per pitcher per gam
```

```
print('Since 1884, the average MLB post season BB (walks) per pitcher per game i
print('')
print('It makes sense that the regular season BB is higher than the postseason B
 Since there are more games played per player on average in the regular season,
```

```
Since 1884, the average MLB regular season BB (walks) per pitcher per game is: 3
0.180390020204772
Since 1884, the average MLB post season BB (walks) per pitcher per game is: 1.92
91322314049586

It makes sense that the regular season BB is higher than the postseason BB (walk
s). Because this stat is not a percentage like ERA, numbers will likely steadily
increase over time. Since there are more games played per player on average in t
he regular season, that means that the totals of them walking batters are much m
ore likely to be higher.
```

In [100…

```
# groupby year and average era...
```

In [101…

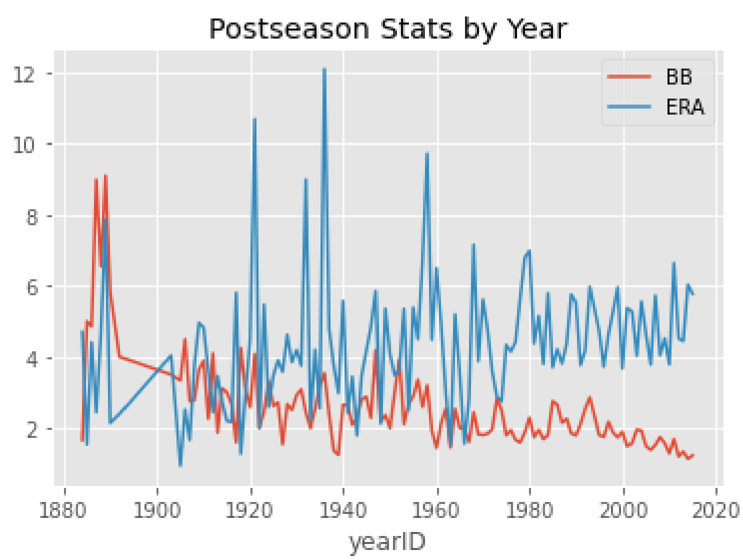
```
post_by_year = post_new.groupby(by='yearID', dropna=False).mean()
reg_by_year = reg_new.groupby(by='yearID', dropna=False).mean()
```

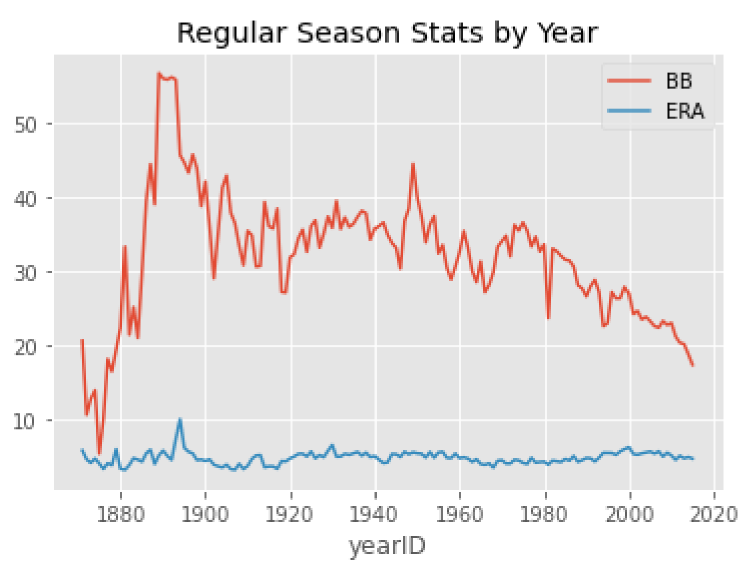
In [102…

```
post_by_year.plot(title='Postseason Stats by Year')
reg_by_year.plot(title='Regular Season Stats by Year')
```

Out[102…

```
&lt;AxesSubplot:title={'center':'Regular Season Stats by Year'}, xlabel='yearID'&gt;
```





In [103…
```
both=pd.merge(reg_by_year, post_by_year, on=['yearID']) # combine into one df
both.columns=['BB_reg', 'ERA_reg', 'BB_post', 'ERA_post']
both
```

Out[103…

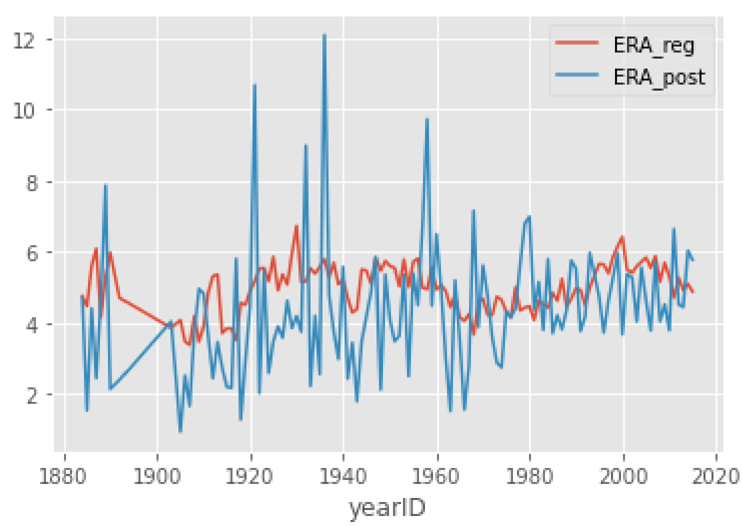
	BB_reg	ERA_reg	BB_post	ERA_post
yearID				
1884	21.051064	4.751234	1.666667	4.700000
1885	29.906780	4.467966	5.000000	1.542500
1886	39.524823	5.598369	4.857143	4.410000
1887	44.492647	6.092941	9.000000	2.436667
1888	39.057851	4.159339	6.555556	4.670000
...	...	...	...	...
2011	21.241867	4.713847	1.693333	6.645267
2012	20.399445	5.274563	1.201149	4.515690
2013	20.216851	4.920207	1.343558	4.443190
2014	18.858681	5.101319	1.134146	6.035305
2015	17.417079	4.880149	1.233129	5.776933

119 rows × 4 columns

In [104…
```
del both['BB_reg']
del both['BB_post']
```

In [105…
```
both.plot() # plot showing ERAs in regular season vs. post season
```

Out[105… <AxesSubplot:xlabel='yearID'>



In [106…
```
both=pd.merge(reg_by_year, post_by_year, on=['yearID']) # combine into one df
both.columns=['BB_reg', 'ERA_reg', 'BB_post', 'ERA_post']
del both['ERA_reg']
del both['ERA_post']
both
```

Out[106…

	BB_reg	BB_post
yearID		
1884	21.051064	1.666667
1885	29.906780	5.000000
1886	39.524823	4.857143
1887	44.492647	9.000000
1888	39.057851	6.555556
...	...	...
2011	21.241867	1.693333
2012	20.399445	1.201149
2013	20.216851	1.343558
2014	18.858681	1.134146
2015	17.417079	1.233129

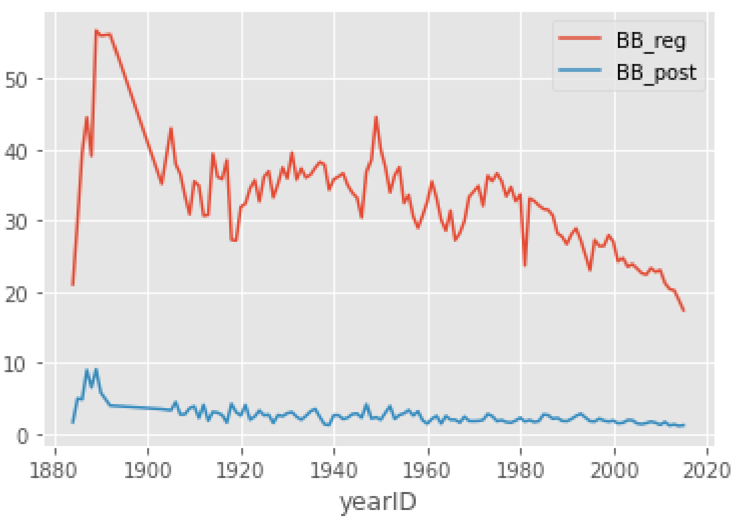
119 rows × 2 columns

In [107…

```
both.plot() # total BB over the years
```

Out[107…

```
<AxesSubplot:xlabel='yearID'>
```



In [108…

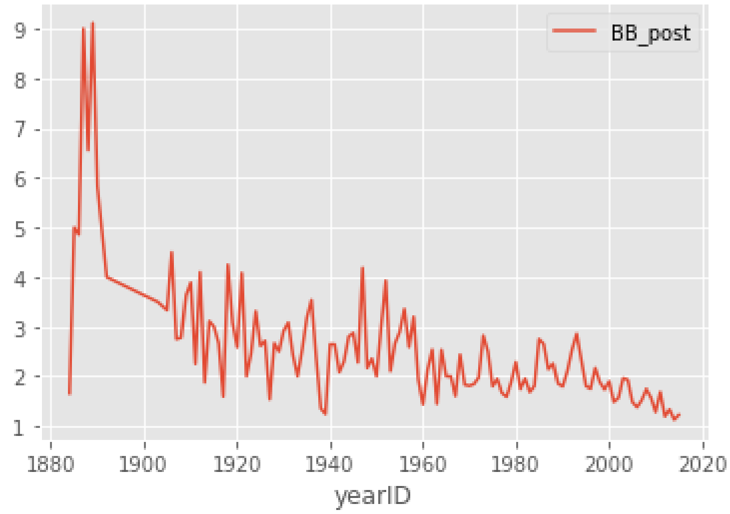
```
del both['BB_reg']
```

In [109…

```
both.plot() # just postseason walks
```

Out[109…

```
<AxesSubplot:xlabel='yearID'>
```


BB_post
yearID
1
2
3
4
5
6
7
8
9
1880
1900
1920
1940
1960
1980
2000
2020

In [ ]: