Final Project

December 20, 2021

[168]:
```
import pandas as pd
from sklearn import model_selection
from sklearn import linear_model
from statsmodels import api as sm

data = pd.read_csv("Batting.csv", sep=',')
people = pd.read_csv("People.csv", sep = ',')
data.head()
```

[168]:

	playerID	yearID	stint	teamID	lgID	G	AB	R	H	2B	…	RBI	SB	\
0	abercda01	1871	1	TRO	NaN	1	4	0	0	0	…	0.0	0.0	
1	addybo01	1871	1	RC1	NaN	25	118	30	32	6	…	13.0	8.0	
2	allisar01	1871	1	CL1	NaN	29	137	28	40	4	…	19.0	3.0	
3	allisdo01	1871	1	WS3	NaN	27	133	28	44	10	…	27.0	1.0	
4	ansonca01	1871	1	RC1	NaN	25	120	29	39	11	…	16.0	6.0	

	CS	BB	SO	IBB	HBP	SH	SF	GIDP
0	0.0	0	0.0	NaN	NaN	NaN	NaN	0.0
1	1.0	4	0.0	NaN	NaN	NaN	NaN	0.0
2	1.0	2	5.0	NaN	NaN	NaN	NaN	1.0
3	1.0	0	2.0	NaN	NaN	NaN	NaN	0.0
4	2.0	2	1.0	NaN	NaN	NaN	NaN	0.0

[5 rows x 22 columns]

[59]:
```
data = data.loc[data["yearID"] > 2010]
data = data.loc[data['yearID'] < 2020]
data = data.loc[data["AB"]>100]
data
```

[59]:

	playerID	yearID	stint	teamID	lgID	G	AB	R	H	2B	…	\
94222	abreubo01	2011	1	LAA	AL	142	502	54	127	30	…	
94226	ackledu01	2011	1	SEA	AL	90	333	39	91	16	…	
94237	allenbr01	2011	2	OAK	AL	41	146	18	30	9	…	
94240	altuvjo01	2011	1	HOU	NL	57	221	26	61	10	…	
94242	alvarpe01	2011	1	PIT	NL	74	235	18	45	9	…	
…	…	…	…	…	…	…	…	…	…	..	…	

```
107413  yastrmi01    2019      1    SFN   NL  107  371   64  101  22  …
107415  yelicch01    2019      1    MIL   NL  130  489  100  161  29  …
107426  zimmery01    2019      1    WAS   NL   52  171   20   44   9  …
107427  zobribe01    2019      1    CHN   NL   47  150   24   39   5  …
107428  zuninmi01    2019      1    TBA   AL   90  266   30   44  10  …

         RBI    SB   CS  BB     SO   IBB  HBP   SH   SF  GIDP
94222   60.0  21.0  5.0  78  113.0   5.0  1.0  1.0  3.0   8.0
94226   36.0   6.0  0.0  40   79.0   1.0  0.0  0.0  3.0   3.0
94237   11.0   2.0  0.0  11   55.0   1.0  0.0  0.0  1.0   0.0
94240   12.0   7.0  3.0   5   29.0   0.0  2.0  5.0  1.0   5.0
94242   19.0   1.0  0.0  24   80.0   1.0  2.0  1.0  0.0  11.0
…       …    …   …   ..    …    …   …   …   …    …
107413  55.0   2.0  4.0  32  107.0   1.0  4.0  1.0  3.0   4.0
107415  97.0  30.0  2.0  80  118.0  16.0  8.0  0.0  3.0   8.0
107426  27.0   0.0  0.0  17   39.0   0.0  0.0  0.0  2.0   4.0
107427  17.0   0.0  0.0  23   24.0   0.0  1.0  0.0  2.0   6.0
107428  32.0   0.0  0.0  20   98.0   0.0  3.0  0.0  0.0   4.0

[3993 rows x 22 columns]
```

[60]:
```
data['nameLast'] = people['nameLast']
data['nameGiven'] = people['nameGiven']
data
```

[60]:
```
         playerID  yearID  stint teamID lgID    G   AB    R    H  2B  …  \
94222   abreubo01    2011      1    LAA   AL  142  502   54  127  30  …
94226   ackledu01    2011      1    SEA   AL   90  333   39   91  16  …
94237   allenbr01    2011      2    OAK   AL   41  146   18   30   9  …
94240   altuvjo01    2011      1    HOU   NL   57  221   26   61  10  …
94242   alvarpe01    2011      1    PIT   NL   74  235   18   45   9  …
…           …     …    …     …   …  …  …  …  …  …  …  …
107413  yastrmi01    2019      1    SFN   NL  107  371   64  101  22  …
107415  yelicch01    2019      1    MIL   NL  130  489  100  161  29  …
107426  zimmery01    2019      1    WAS   NL   52  171   20   44   9  …
107427  zobribe01    2019      1    CHN   NL   47  150   24   39   5  …
107428  zuninmi01    2019      1    TBA   AL   90  266   30   44  10  …

         CS  BB     SO   IBB  HBP   SH   SF  GIDP  nameLast  nameGiven
94222   5.0  78  113.0   5.0  1.0  1.0  3.0   8.0       NaN        NaN
94226   0.0  40   79.0   1.0  0.0  0.0  3.0   3.0       NaN        NaN
94237   0.0  11   55.0   1.0  0.0  0.0  1.0   0.0       NaN        NaN
94240   3.0   5   29.0   0.0  2.0  5.0  1.0   5.0       NaN        NaN
94242   0.0  24   80.0   1.0  2.0  1.0  0.0  11.0       NaN        NaN
…     …  ..    …    …  …  …  …  …    …         …          …
107413  4.0  32  107.0   1.0  4.0  1.0  3.0   4.0       NaN        NaN
107415  2.0  80  118.0  16.0  8.0  0.0  3.0   8.0       NaN        NaN
```

```
107426  0.0  17   39.0   0.0  0.0  0.0  2.0   4.0       NaN        NaN
107427  0.0  23   24.0   0.0  1.0  0.0  2.0   6.0       NaN        NaN
107428  0.0  20   98.0   0.0  3.0  0.0  0.0   4.0       NaN        NaN

[3993 rows x 24 columns]
```

[61]:
```
for i in range(data.shape[0]):
    lastName = people.loc[people.playerID == data.iat[i,0],'nameLast'].values[0]
    givenName = people.loc[people.playerID == data.iat[i,0],'nameGiven'].
 ↪values[0]
    data.iat[i,22] = lastName
    data.iat[i,23] = givenName

data.head()
```

[61]:
```
        playerID  yearID  stint teamID lgID    G   AB   R    H  2B  …   CS  \
94222  abreubo01    2011      1    LAA   AL  142  502  54  127  30  …  5.0
94226  ackledu01    2011      1    SEA   AL   90  333  39   91  16  …  0.0
94237  allenbr01    2011      2    OAK   AL   41  146  18   30   9  …  0.0
94240  altuvjo01    2011      1    HOU   NL   57  221  26   61  10  …  3.0
94242  alvarpe01    2011      1    PIT   NL   74  235  18   45   9  …  0.0

       BB     SO  IBB  HBP   SH   SF  GIDP  nameLast       nameGiven
94222  78  113.0  5.0  1.0  1.0  3.0   8.0     Abreu       Bob Kelly
94226  40   79.0  1.0  0.0  0.0  3.0   3.0    Ackley  Dustin Michael
94237  11   55.0  1.0  0.0  0.0  1.0   0.0     Allen  Brandon Durell
94240   5   29.0  0.0  2.0  5.0  1.0   5.0    Altuve     Jose Carlos
94242  24   80.0  1.0  2.0  1.0  0.0  11.0   Alvarez    Pedro Manuel

[5 rows x 24 columns]
```

[62]:
```
data['nameFull'] = data['nameGiven'] + ' '+ data['nameLast']
```

[74]:
```
BA = []
OBP = []
SLG = []
OPS = []
for i in range(data.shape[0]):
    Hits = data.iat[i,8]
    AB = data.iat[i,6]
    Walks = data.iat[i,15]
    HBP = data.iat[i,18]
    SF = data.iat[i,20]
    TB = data.iat[i,9] + data.iat[i,10]*2 +\
        data.iat[i,11]*3 + data.iat[i,8]
    OBP.append(round((Hits+Walks+HBP)/(AB+Walks+SF+HBP),3))
    BA.append(round(Hits/(AB-Walks),3))
```

```
    SLG.append(round(TB/AB,3))
    OPS.append(OBP[i]+SLG[i])


data['BA'] = BA
data['OBP'] = OBP
data['SLG'] = SLG
data['OPS'] = OPS
```

[75]:
```
data
```

[75]:

	playerID	yearID	stint	teamID	lgID	G	AB	R	H	2B	…	\
94222	abreubo01	2011	1	LAA	AL	142	502	54	127	30	…	
94226	ackledu01	2011	1	SEA	AL	90	333	39	91	16	…	
94237	allenbr01	2011	2	OAK	AL	41	146	18	30	9	…	
94240	altuvjo01	2011	1	HOU	NL	57	221	26	61	10	…	
94242	alvarpe01	2011	1	PIT	NL	74	235	18	45	9	…	
…	…	…	…	…	…	…	…	…	…	..	…	
107413	yastrmi01	2019	1	SFN	NL	107	371	64	101	22	…	
107415	yelicch01	2019	1	MIL	NL	130	489	100	161	29	…	
107426	zimmery01	2019	1	WAS	NL	52	171	20	44	9	…	
107427	zobribe01	2019	1	CHN	NL	47	150	24	39	5	…	
107428	zuninmi01	2019	1	TBA	AL	90	266	30	44	10	…	

	SH	SF	GIDP	nameLast	nameGiven	\
94222	1.0	3.0	8.0	Abreu	Bob Kelly	
94226	0.0	3.0	3.0	Ackley	Dustin Michael	
94237	0.0	1.0	0.0	Allen	Brandon Durell	
94240	5.0	1.0	5.0	Altuve	Jose Carlos	
94242	1.0	0.0	11.0	Alvarez	Pedro Manuel	
…	…	…	…	…	…	
107413	1.0	3.0	4.0	Yastrzemski	Michael Andrew	
107415	0.0	3.0	8.0	Yelich	Christian Stephen	
107426	0.0	2.0	4.0	Zimmerman	Ryan Wallace	
107427	0.0	2.0	6.0	Zobrist	Benjamin Thomas	
107428	0.0	0.0	4.0	Zunino	Michael Accorsi	

	nameFull	BA	OBP	SLG	OPS
94222	Bob Kelly Abreu	0.300	0.353	0.365	0.718
94226	Dustin Michael Ackley	0.311	0.348	0.417	0.765
94237	Brandon Durell Allen	0.222	0.259	0.356	0.615
94240	Jose Carlos Altuve	0.282	0.297	0.357	0.654
94242	Pedro Manuel Alvarez	0.213	0.272	0.289	0.561
…	…	…	…	…	…
107413	Michael Andrew Yastrzemski	0.298	0.334	0.518	0.852
107415	Christian Stephen Yelich	0.394	0.429	0.671	1.100
107426	Ryan Wallace Zimmerman	0.286	0.321	0.415	0.736

```
107427     Benjamin Thomas Zobrist  0.307  0.358  0.313  0.671
107428      Michael Accorsi Zunino  0.179  0.232  0.312  0.544

[3993 rows x 29 columns]
```

[77]:
```
battingAverageTotal = [0]*9
batterYear = [0]*9
averageBAYear = [0]*9
for i in range(data.shape[0]):
    if data.iat[i,1] == 2011:
        battingAverageTotal[0] += data.iat[i,25]
        batterYear[0] += 1
    elif data.iat[i,1] == 2012:
        battingAverageTotal[1] += data.iat[i,25]
        batterYear[1] += 1
    elif data.iat[i,1] == 2013:
        battingAverageTotal[2] += data.iat[i,25]
        batterYear[2] += 1
    elif data.iat[i,1] == 2014:
        battingAverageTotal[3] += data.iat[i,25]
        batterYear[3] += 1
    elif data.iat[i,1] == 2015:
        battingAverageTotal[4] += data.iat[i,25]
        batterYear[4] += 1
    elif data.iat[i,1] == 2016:
        battingAverageTotal[5] += data.iat[i,25]
        batterYear[5] += 1
    elif data.iat[i,1] == 2017:
        battingAverageTotal[6] += data.iat[i,25]
        batterYear[6] += 1
    elif data.iat[i,1] == 2018:
        battingAverageTotal[7] += data.iat[i,25]
        batterYear[7] += 1
    elif data.iat[i,1] == 2019:
        battingAverageTotal[8] += data.iat[i,25]
        batterYear[8] += 1
for i in range(0,9):
    averageBAYear[i] = round(battingAverageTotal[i]/batterYear[i],3)

years = [2011,2012,2013,2014,2015,2016,2017,2018,2019]
averageBAYear
```

[77]:
```
[0.28, 0.279, 0.278, 0.274, 0.279, 0.28, 0.283, 0.274, 0.28]
```

[78]:
```
silverSlugger= ['garvemi01','realmjt01','santaca01',\
                'freemfr01','lemahdj01','albieoz01',\
                'bregmal01','rendoan01','bogaexa01',\
```

```
                'storytr01','troutmi01','springe01',\
                'bettsmo01','bellico01','yelicch01',\
                'acunaro01','cruzne02']
```

[79]:
```
ssAL = pd.DataFrame()
ssNL = pd.DataFrame()
for player in silverSlugger:
    if (data.loc[data.playerID == player,'lgID'].values[0] == 'AL'):
        ssAL = ssAL.append(data.loc[data['playerID'] == player])
    else:
        ssNL = ssNL.append(data.loc[data['playerID'] == player])
```

[80]:
```
ssAL.head()
```

[80]:
```
         playerID  yearID  stint teamID lgID    G   AB   R    H  2B  …   SH  \
104814  garvemi01    2018      1    MIN   AL  103  302  38   81  19  …  1.0
106376  garvemi01    2019      1    MIN   AL   93  311  70   85  16  …  0.0
95350   santaca01    2011      1    CLE   AL  155  552  84  132  35  …  0.0
96764   santaca01    2012      1    CLE   AL  143  507  72  128  27  …  0.0
98186   santaca01    2013      1    CLE   AL  154  541  75  145  39  …  0.0

         SF  GIDP  nameLast     nameGiven             nameFull     BA    OBP  \
104814  1.0   8.0    Garver  Mitchell Lyn  Mitchell Lyn Garver  0.297  0.335
106376  2.0   5.0    Garver  Mitchell Lyn  Mitchell Lyn Garver  0.315  0.365
95350   7.0  15.0   Santana        Carlos       Carlos Santana  0.290  0.351
96764   8.0  21.0   Santana        Carlos       Carlos Santana  0.308  0.365
98186   4.0   7.0   Santana        Carlos       Carlos Santana  0.324  0.377

          SLG    OPS
104814  0.414  0.749
106376  0.630  0.995
95350   0.457  0.808
96764   0.420  0.785
98186   0.455  0.832

[5 rows x 29 columns]
```

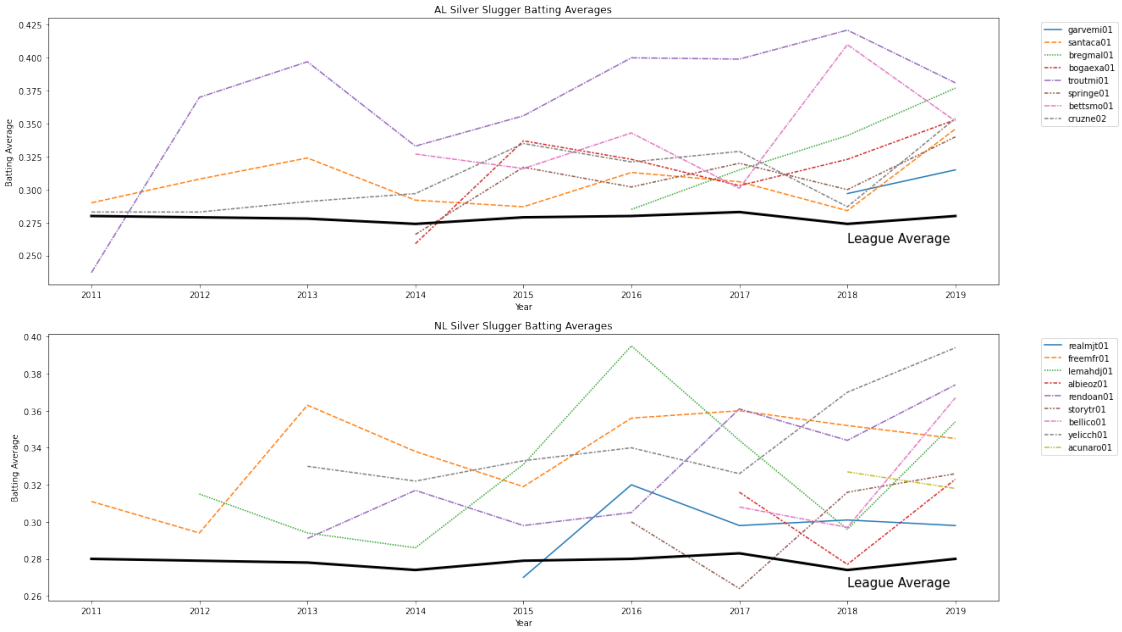
[81]:
```
ssNL.head()
```

[81]:
```
         playerID  yearID  stint teamID lgID    G   AB   R    H  2B  …   SH  \
100974  realmjt01    2015      1    MIA   NL  126  441  49  114  21  …  1.0
102453  realmjt01    2016      1    MIA   NL  137  509  60  154  31  …  0.0
103929  realmjt01    2017      1    MIA   NL  141  532  68  148  31  …  0.0
105446  realmjt01    2018      1    MIA   NL  125  477  74  132  30  …  0.0
106999  realmjt01    2019      1    PHI   NL  145  538  92  148  36  …  0.0

         SF  GIDP  nameLast    nameGiven             nameFull     BA    OBP  \
```

```
100974  4.0  11.0  Realmuto  Jacob Tyler  Jacob Tyler Realmuto  0.270  0.290
102453  3.0  12.0  Realmuto  Jacob Tyler  Jacob Tyler Realmuto  0.320  0.343
103929  3.0  13.0  Realmuto  Jacob Tyler  Jacob Tyler Realmuto  0.298  0.332
105446  4.0   9.0  Realmuto  Jacob Tyler  Jacob Tyler Realmuto  0.301  0.340
106999  8.0  12.0  Realmuto  Jacob Tyler  Jacob Tyler Realmuto  0.298  0.328

          SLG    OPS
100974  0.406  0.696
102453  0.428  0.771
103929  0.451  0.783
105446  0.484  0.824
106999  0.493  0.821

[5 rows x 29 columns]
```

[82]:
```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline
```

[84]:
```
fig, axes = plt.subplots(2,1)
fig.set_size_inches(18.5, 10.5, forward=True)
plot1 =sns.lineplot(ax = axes[0], data=ssAL,x = 'yearID',y = 'BA',hue =␣
 ↪'playerID', style = 'playerID')
plot2 =sns.lineplot(ax = axes[1], data=ssNL,x = 'yearID',y = 'BA',hue =␣
 ↪'playerID', style = 'playerID')
plot1.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plot2.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plot1.title.set_text("AL Silver Slugger Batting Averages")
plot2.title.set_text("NL Silver Slugger Batting Averages")
plot1.set_xlabel("Year")
plot2.set_xlabel("Year")
plot1.set_ylabel("Batting Average")
plot2.set_ylabel("Batting Average")
plot1.plot(years,averageBAYear,color = 'black',linewidth = 3)
plot2.plot(years,averageBAYear,color = 'black',linewidth = 3)
plot1.annotate('League Average', xy= (2018,.260),size = 15)
plot2.annotate('League Average', xy= (2018,.265),size = 15)
fig.tight_layout()
```

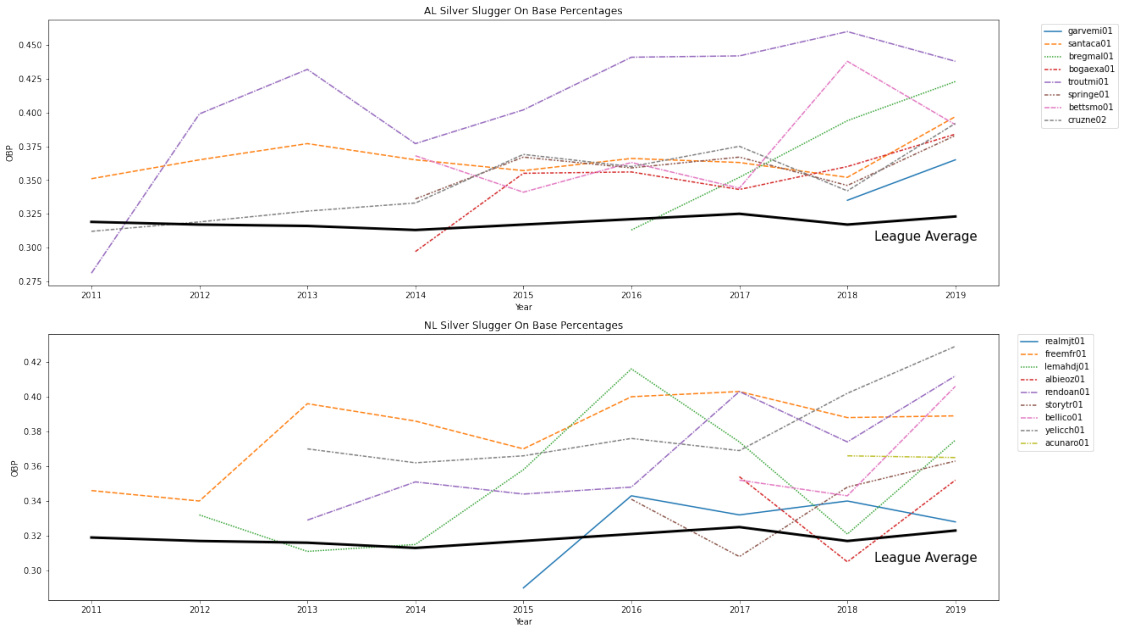


[85]:
```
OBPTotal = [0]*9
averageOBP = [0]*9
for i in range(data.shape[0]):
    if data.iat[i,1] == 2011:
        OBPTotal[0] += data.iat[i,26]
    elif data.iat[i,1] == 2012:
        OBPTotal[1] += data.iat[i,26]
    elif data.iat[i,1] == 2013:
        OBPTotal[2] += data.iat[i,26]
    elif data.iat[i,1] == 2014:
        OBPTotal[3] += data.iat[i,26]
    elif data.iat[i,1] == 2015:
        OBPTotal[4] += data.iat[i,26]
    elif data.iat[i,1] == 2016:
        OBPTotal[5] += data.iat[i,26]
    elif data.iat[i,1] == 2017:
        OBPTotal[6] += data.iat[i,26]
    elif data.iat[i,1] == 2018:
        OBPTotal[7] += data.iat[i,26]
    elif data.iat[i,1] == 2019:
        OBPTotal[8] += data.iat[i,26]

for i in range(0,9):
    averageOBP[i] = round(OBPTotal[i]/batterYear[i],3)

averageOBP
```

[85]: [0.319, 0.317, 0.316, 0.313, 0.317, 0.321, 0.325, 0.317, 0.323]

[86]:
```
fig, axes = plt.subplots(2,1)
fig.set_size_inches(18.5, 10.5, forward=True)
plot1 =sns.lineplot(ax = axes[0], data=ssAL,x = 'yearID',y = 'OBP',hue =␣
 ↪'playerID', style = 'playerID')
plot2 =sns.lineplot(ax = axes[1], data=ssNL,x = 'yearID',y = 'OBP',hue =␣
 ↪'playerID', style = 'playerID')
plot1.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plot2.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plot1.title.set_text("AL Silver Slugger On Base Percentages")
plot2.title.set_text("NL Silver Slugger On Base Percentages")
plot1.set_xlabel("Year")
plot2.set_xlabel("Year")
plot1.set_ylabel("OBP")
plot2.set_ylabel("OBP")
plot1.plot(years,averageOBP,color = 'black',linewidth = 3)
plot2.plot(years,averageOBP,color = 'black',linewidth = 3)
plot1.annotate('League Average', xy= (2018,.260),size = 15)
plot2.annotate('League Average', xy= (2018,.265),size = 15)
fig.tight_layout()
plt.xticks([2011,2012,2013,2014,2015,2016,2017,2018,2019])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plot1.annotate('League Average', xy= (2018.25,.305),size = 15)
plot2.annotate('League Average', xy= (2018.25,.305),size = 15)
```

[86]: Text(2018.25, 0.305, 'League Average')



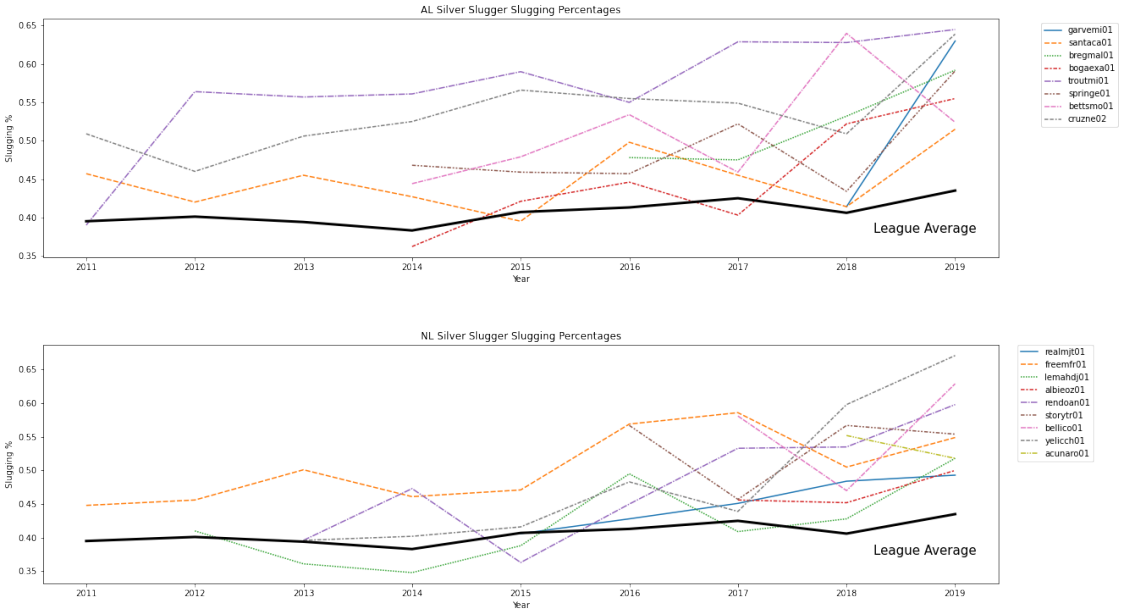
[87]:
```
SluggingTotal = [0]*9
averageSlugging = [0]*9
for i in range(data.shape[0]):
    if data.iat[i,1] == 2011:
        SluggingTotal[0] += data.iat[i,27]
    elif data.iat[i,1] == 2012:
        SluggingTotal[1] += data.iat[i,27]
    elif data.iat[i,1] == 2013:
        SluggingTotal[2] += data.iat[i,27]
    elif data.iat[i,1] == 2014:
        SluggingTotal[3] += data.iat[i,27]
    elif data.iat[i,1] == 2015:
        SluggingTotal[4] += data.iat[i,27]
    elif data.iat[i,1] == 2016:
        SluggingTotal[5] += data.iat[i,27]
    elif data.iat[i,1] == 2017:
        SluggingTotal[6] += data.iat[i,27]
    elif data.iat[i,1] == 2018:
        SluggingTotal[7] += data.iat[i,27]
    elif data.iat[i,1] == 2019:
        SluggingTotal[8] += data.iat[i,27]

for i in range(0,9):
    averageSlugging[i] = round(SluggingTotal[i]/batterYear[i],3)

averageSlugging
```

[87]:
```
[0.395, 0.401, 0.394, 0.383, 0.407, 0.413, 0.425, 0.406, 0.435]
```

[88]:
```
fig, axes = plt.subplots(2,1)
fig.set_size_inches(18.5, 10.5, forward=True)
plot1 =sns.lineplot(ax = axes[0], data=ssAL,x = 'yearID',y = 'SLG',hue =␣
 ↪'playerID', style = 'playerID')
plot2 =sns.lineplot(ax = axes[1], data=ssNL,x = 'yearID',y = 'SLG',hue =␣
 ↪'playerID', style = 'playerID')
plot1.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plot2.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plot1.title.set_text("AL Silver Slugger Slugging Percentages")
plot2.title.set_text("NL Silver Slugger Slugging Percentages")
plot1.set_xlabel("Year")
plot2.set_xlabel("Year")
plot1.set_ylabel("Slugging %")
plot2.set_ylabel("Slugging %")
plot1.plot(years,averageSlugging,color = 'black',linewidth = 3)
plot2.plot(years,averageSlugging,color = 'black',linewidth = 3)
```

```
plot1.annotate('League Average', xy= (2018,.260),size = 15)
plot2.annotate('League Average', xy= (2018,.265),size = 15)
fig.tight_layout()
plt.xticks([2011,2012,2013,2014,2015,2016,2017,2018,2019])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plot1.annotate('League Average', xy= (2018.25,.380),size = 15)
plot2.annotate('League Average', xy= (2018.25,.375),size = 15)
```

[88]: Text(2018.25, 0.375, 'League Average')



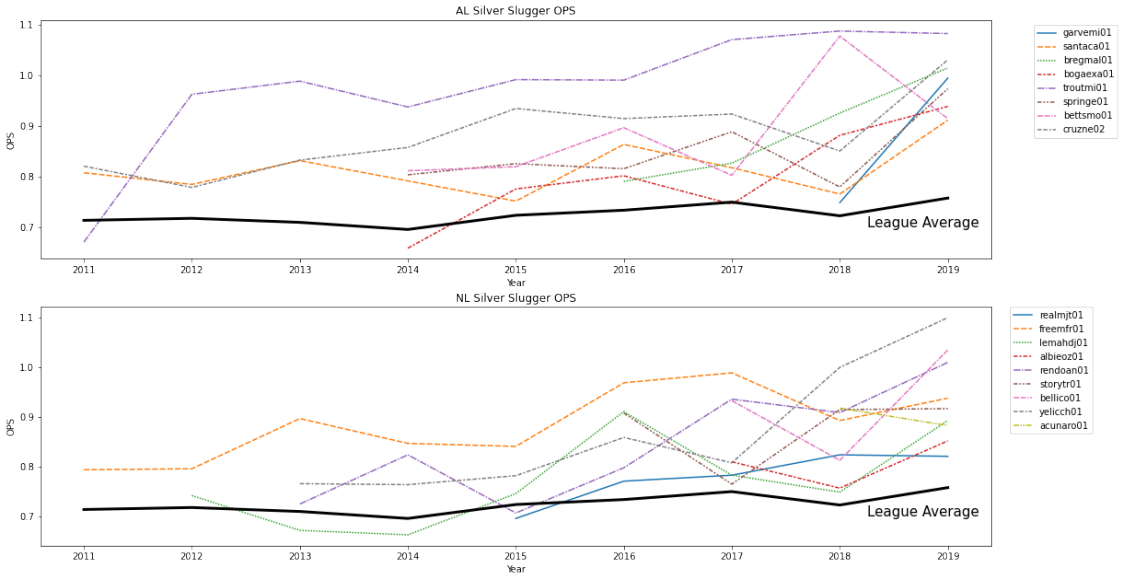
[89]:
```
averageOPS = [0]*9
for i in range(0,9):
    averageOPS[i] = round(averageSlugging[i]+averageOBP[i],3)

averageOPS
```

[89]: [0.714, 0.718, 0.71, 0.696, 0.724, 0.734, 0.75, 0.723, 0.758]

[90]:
```
fig, axes = plt.subplots(2,1)
fig.set_size_inches(18.5, 10.5, forward=True)
plot1 =sns.lineplot(ax = axes[0], data=ssAL,x = 'yearID',y = 'OPS',hue =␣
 ↪'playerID', style = 'playerID')
plot2 =sns.lineplot(ax = axes[1], data=ssNL,x = 'yearID',y = 'OPS',hue =␣
 ↪'playerID', style = 'playerID')
plot1.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plot2.legend(bbox_to_anchor=(1.04,1), loc="upper left")
```

```
plot1.title.set_text("AL Silver Slugger OPS")
plot2.title.set_text("NL Silver Slugger OPS")
plot1.set_xlabel("Year")
plot2.set_xlabel("Year")
plot1.set_ylabel("OPS")
plot2.set_ylabel("OPS")
plot1.plot(years,averageOPS,color = 'black',linewidth = 3)
plot2.plot(years,averageOPS,color = 'black',linewidth = 3)
plot1.annotate('League Average', xy= (2018,.260),size = 15)
plot2.annotate('League Average', xy= (2018,.265),size = 15)
plt.xticks([2011,2012,2013,2014,2015,2016,2017,2018,2019])
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plot1.annotate('League Average', xy= (2018.25,.7),size = 15)
plot2.annotate('League Average', xy= (2018.25,.7),size = 15)
```

[90]: Text(2018.25, 0.7, 'League Average')



[150]:
```
third = pd.read_csv("FanGraphs Leaderboard.csv", sep = ',')
third
```

[150]:

	Given Name	Last Name	Team	G	PA	HR	R	RBI	SB	BB%	…	\
0	Michael	Trout	LAA	134	600	45	110	104	11	18.30%	…	
1	Alex	Bregman	HOU	156	690	41	122	112	5	17.20%	…	

```
2    Christian     Yelich  MIL  130  580  44  100   97  30  13.80%  …
3         Cody  Bellinger  LAD  156  660  47  121  115  15  14.40%  …
4       Marcus     Semien  OAK  162  747  33  123   92  10  11.60%  …
..         …        …   …   …   …  ..  …   …  ..     …   …
130      Eric     Hosmer  SDP  160  667  22   72   99   0   6.00%  …
131    Miguel    Cabrera  DET  136  549  12   41   59   0   8.70%  …
132   Orlando      Arcia  MIL  152  546  15   51   59   8   7.90%  …
133    Albert     Pujols  LAA  131  545  23   55   93   3   7.90%  …
134     Khris      Davis  OAK  133  533  23   61   73   0   8.80%  …

       OBP    SLG   wOBA  xwOBA  wRC+  BsR   Off   Def  WAR  playerid
0    0.438  0.645  0.436  0.451   177  7.1  65.9  -0.3  8.4     10155
1    0.423  0.592  0.418  0.369   167 -2.1  57.3   4.8  8.4     17678
2    0.429  0.671  0.442  0.424   174  8.5  65.2  -3.9  7.8     11477
3    0.406  0.629  0.415  0.424   161  1.4  54.2   3.4  7.7     15998
4    0.369  0.522  0.373  0.353   138  1.7  37.7  14.1  7.6     12533
..     …      …      …      …   …  …   …   …   …       …
130  0.310  0.425  0.309  0.305    92 -5.2 -12.1 -12.4 -0.3      3516
131  0.346  0.398  0.318  0.326    97 -6.3  -8.4 -13.9 -0.3      1744
132  0.283  0.350  0.269  0.271    61  0.9 -27.3   5.4 -0.4     13185
133  0.305  0.430  0.308  0.313    90 -4.4 -11.4 -13.9 -0.6      1177
134  0.293  0.387  0.289  0.312    81 -2.2 -15.0 -13.5 -0.9      9112

[135 rows x 24 columns]
```

[151]:
```
only2019 = pd.DataFrame()
only2019 = only2019.append(data.loc[data['yearID'] == 2019])
only2019['WAR'] = np.nan
only2019 = only2019.drop(columns =␣
 ↪['nameLast','nameGiven','GIDP','SF','SH','HBP','IBB'])
only2019.head()
```

[151]:
```
         playerID  yearID  stint teamID lgID    G   AB    R    H  2B  …  \
105862  abreujo02    2019      1    CHA   AL  159  634   85  180  38  …
105863  acunaro01    2019      1    ATL   NL  156  626  127  175  22  …
105865  adamewi01    2019      1    TBA   AL  152  531   69  135  25  …
105872  adamsma01    2019      1    WAS   NL  111  310   42   70  14  …
105874  adriaeh01    2019      1    MIN   AL   84  202   34   55   8  …

          SB   CS  BB     SO                nameFull     BA    OBP    SLG  \
105862   2.0  2.0  36  152.0       Jose Dariel Abreu  0.301  0.330  0.503
105863  37.0  9.0  76  188.0       Ronald Jose Acuna  0.318  0.365  0.518
105865   4.0  2.0  46  153.0     Willy Rafael Adames  0.278  0.317  0.418
105872   0.0  0.0  20  115.0     Matthew James Adams  0.241  0.276  0.465
105874   0.0  2.0  20   40.0  Ehire Enrique Adrianza  0.302  0.349  0.416

          OPS  WAR
```

```
105862  0.833  NaN
105863  0.883  NaN
105865  0.735  NaN
105872  0.741  NaN
105874  0.765  NaN

[5 rows x 23 columns]
```

[152]:
```
print(only2019.loc[only2019['playerID'] == 'troutmi01'])
```

```
         playerID  yearID  stint teamID lgID    G   AB    R    H  2B  …  \
107287  troutmi01    2019      1    LAA   AL  134  470  110  137  27  …

          SB   CS   BB     SO              nameFull     BA    OBP    SLG  \
107287  11.0  2.0  110  120.0  Michael Nelson Trout  0.381  0.438  0.645

          OPS  WAR
107287  1.083  NaN

[1 rows x 23 columns]
```

[155]:
```
i = 0
only2019WAR = pd.DataFrame()
for i in range(only2019.shape[0]):
    for j in range(third.shape[0]):
        if third.iat[j,0] in only2019.iat[i,17]:
            if third.iat[j,1] in only2019.iat[i,17]:
                only2019.iat[i,22] = third.iat[j,22]
                only2019WAR = only2019WAR.append(only2019.
 ↪loc[only2019['playerID'] == only2019.iat[i,0]])

only2019WAR.head()
```

[155]:
```
          playerID  yearID  stint teamID lgID    G   AB    R    H  2B  …  \
105862  abreujo02    2019      1    CHA   AL  159  634   85  180  38  …
105863  acunaro01    2019      1    ATL   NL  156  626  127  175  22  …
105865  adamewi01    2019      1    TBA   AL  152  531   69  135  25  …
105881  alberha01    2019      1    BAL   AL  139  524   62  160  21  …
105900  alonspe01    2019      1    NYN   NL  161  597  103  155  30  …

           SB   CS  BB     SO              nameFull     BA    OBP    SLG    OPS  \
105862    2.0  2.0  36  152.0    Jose Dariel Abreu  0.301  0.330  0.503  0.833
105863   37.0  9.0  76  188.0    Ronald Jose Acuna  0.318  0.365  0.518  0.883
105865    4.0  2.0  46  153.0  Willy Rafael Adames  0.278  0.317  0.418  0.735
105881    4.0  4.0  16   50.0  Hanser Joel Alberto  0.315  0.329  0.422  0.751
105900    1.0  0.0  72  183.0  Peter Morgan Alonso  0.295  0.358  0.583  0.941
```

```
        WAR
105862  1.7
105863  5.5
105865  2.9
105881  1.9
105900  4.9

[5 rows x 23 columns]
```

[156]:
```
only2019WAR.sort_values(by='WAR', ascending=False)
```

[156]:
```
         playerID  yearID  stint teamID lgID    G   AB    R    H  2B  …  \
106046  bregmal01    2019      1    HOU   AL  156  554  122  164  37  …
107287  troutmi01    2019      1    LAA   AL  134  470  110  137  27  …
107415  yelicch01    2019      1    MIL   NL  130  489  100  161  29  …
105988  bellico01    2019      1    LAN   NL  156  558  121  170  34  …
107131  semiema01    2019      1    OAK   AL  162  657  123  187  43  …
…            …     …    …     …   …  …  …  …  …  …  ..  …
106081  cabremi01    2019      1    DET   AL  136  493   41  139  21  …
106541  hosmeer01    2019      1    SDN   NL  160  619   72  164  29  …
105931  arciaor01    2019      1    MIL   NL  152  494   51  110  16  …
106975  pujolal01    2019      1    LAA   AL  131  491   55  120  22  …
106194  daviskh01    2019      1    OAK   AL  133  481   61  106  11  …

          SB   CS   BB     SO                  nameFull     BA    OBP    SLG  \
106046   5.0  1.0  119   83.0   Alexander David Bregman  0.377  0.423  0.592
107287  11.0  2.0  110  120.0      Michael Nelson Trout  0.381  0.438  0.645
107415  30.0  2.0   80  118.0  Christian Stephen Yelich  0.394  0.429  0.671
105988  15.0  5.0   95  108.0      Cody James Bellinger  0.367  0.406  0.629
107131  10.0  8.0   87  102.0      Marcus Andrew Semien  0.328  0.369  0.522
…       …  …  …    …                         …    …    …    …
106081   0.0  0.0   48  108.0       Jose Miguel Cabrera  0.312  0.346  0.398
106541   0.0  3.0   40  163.0          Eric John Hosmer  0.283  0.310  0.425
105931   8.0  5.0   43  109.0       Orlando Jesus Arcia  0.244  0.283  0.350
106975   3.0  0.0   43   68.0       Jose Alberto Pujols  0.268  0.305  0.430
106194   0.0  0.0   47  146.0  Khristopher Adrian Davis  0.244  0.293  0.387

          OPS  WAR
106046  1.015  8.4
107287  1.083  8.4
107415  1.100  7.8
105988  1.035  7.7
107131  0.891  7.6
…       …  …
106081  0.744 -0.3
106541  0.735 -0.3
105931  0.633 -0.4
```

```
106975  0.735 -0.6
106194  0.680 -0.9

[131 rows x 23 columns]
```

[163]:
```
grouped_attrs = ['BA','OBP','SLG','OPS']
features = only2019WAR[grouped_attrs]
target = only2019WAR[['WAR']]
```

[174]:
```
X = features
y = target['WAR']
lm = linear_model.LinearRegression()
model = lm.fit(X,y)
```

[171]:
```
lm.score(X,y)
```

[171]:
```
0.6679651664503146
```

[175]:
```
sk_coeff = lm.coef_.tolist()
for attr, coef in zip(grouped_attrs, sk_coeff):
    print('Attribute: {}, Coefficient: {}'.format(attr,coef))
```

```
Attribute: BA, Coefficient: -2.319440436602853
Attribute: OBP, Coefficient: 13.309707102003758
Attribute: SLG, Coefficient: 0.4955868336785638
Attribute: OPS, Coefficient: 13.805293935682313
```

[187]:
```
sm_y = y
sm_X = X

#add a constant to the features
sm_X = sm.add_constant(X)

# use Ordinary Least Squares
OLS_model = sm.OLS(y,X).fit()
OLS_model.summary()
```

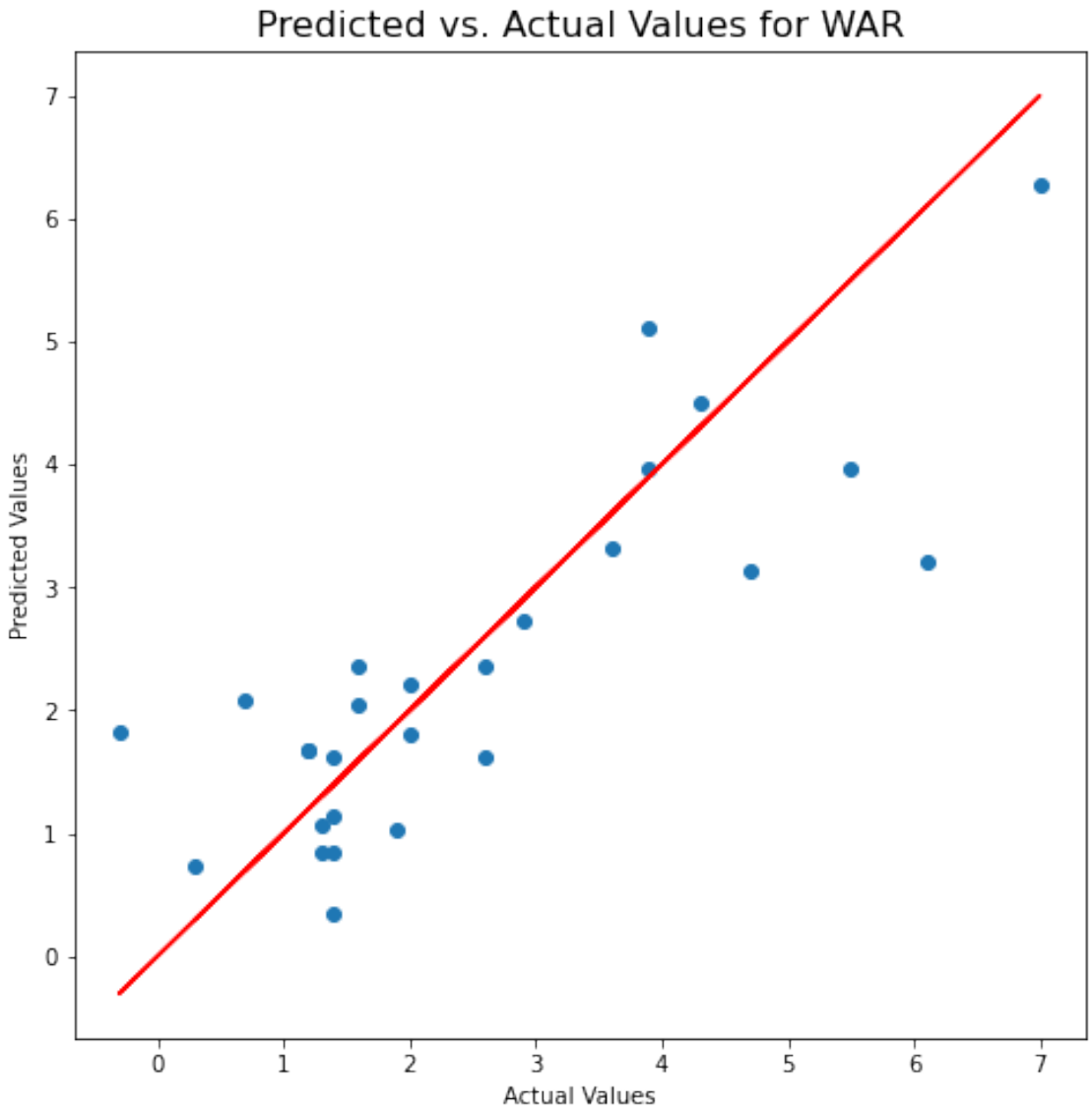
[187]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                                 OLS Regression Results
================================================================================
=======
Dep. Variable:                      WAR   R-squared (uncentered):
0.764
Model:                              OLS   Adj. R-squared (uncentered):
0.759
Method:                 Least Squares   F-statistic:
```

```
138.4
Date:                Mon, 20 Dec 2021   Prob (F-statistic):
5.23e-40
Time:                        02:15:58   Log-Likelihood:
-253.52
No. Observations:                 131   AIC:
513.0
Df Residuals:                     128   BIC:
521.7
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
BA             2.9565     10.951      0.270      0.788     -18.712      24.625
OBP          -13.6900      6.759     -2.025      0.045     -27.065      -0.315
SLG           13.8281      4.168      3.318      0.001       5.581      22.075
OPS            0.1381      3.165      0.044      0.965      -6.125       6.401
==============================================================================
Omnibus:                        5.799   Durbin-Watson:                   1.959
Prob(Omnibus):                  0.055   Jarque-Bera (JB):                5.655
Skew:                           0.509   Prob(JB):                       0.0592
Kurtosis:                       3.041   Cond. No.                     8.19e+15
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not
contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[3] The smallest eigenvalue is 2.19e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```

[179]:
```
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,␣
↪test_size=0.2)

lm = linear_model.LinearRegression()
model = lm.fit(X_train, y_train)

predictions = lm.predict(X_test)

predictions[0:5]
```

[179]:
```
array([3.96862981, 2.05456731, 2.72644231, 1.61706731, 3.96081731])
```

[184]:
```
plt.figure(figsize = (8,8))
plt.title("Predicted vs. Actual Values for WAR", fontsize=16)
plt.scatter(y_test, predictions)
plt.plot(y_test, y_test, color = "Red")
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.show()
```



[186]:
```
print("Accuracy (scale of 0 to 1): {}".format(model.score(X_test, y_test)))
```

```
Accuracy (scale of 0 to 1): 0.6853278323823044
```

[ ]: