In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import numpy as np
import pickle
```

In [2]:
```
mlb_df = pd.read_pickle('final_df1.pkl')
mlb_df.head(25)
```

Out[2]:

playerID	g	ab	r	h	2b	3b	hr	rbi	sb	bb	...	k_percentage	bb_
aaronha01	3298	12364	2174	3771	624	98	755	2297.0	240.0	1402	...	16.200000	
abbated01	827	2942	346	748	95	43	11	310.0	138.0	281	...	15.766667	
abbotku01	702	2044	273	523	109	23	62	242.0	22.0	133	...	10.900000	
abreubo01	2425	8480	1453	2470	574	59	288	1363.0	400.0	1476	...	22.020000	
abreujo02	901	3547	483	1038	218	14	179	611.0	10.0	245	...	6.800000	
ackledu01	635	2125	261	512	94	18	46	216.0	31.0	194	...	21.133333	
adairje01	1165	4019	378	1022	163	19	57	366.0	29.0	208	...	9.773333	
adamsbo03	1281	4019	591	1082	188	49	37	303.0	67.0	414	...	18.600000	
adamsbu01	576	2003	282	532	96	12	50	249.0	12.0	234	...	15.333333	
adamsgl01	661	1617	152	452	79	5	34	225.0	6.0	111		8.700000	

In [3]:
```
mlb_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2316 entries, aaronha01 to zuninmi01
Data columns (total 37 columns):
g                           2316 non-null int64
ab                          2316 non-null int64
r                           2316 non-null int64
h                           2316 non-null int64
2b                          2316 non-null int64
3b                          2316 non-null int64
hr                          2316 non-null int64
rbi                         2316 non-null float64
sb                          2316 non-null float64
bb                          2316 non-null int64
so                          2316 non-null float64
asg_mvp                     2316 non-null float64
baberuth_award              2316 non-null float64
baseball_magazine_allstar   2316 non-null float64
comeback_poy                2316 non-null float64
gold_glove_award            2316 non-null float64
hankaaron_award             2316 non-null float64
hutch_award                 2316 non-null float64
lougehrig_award             2316 non-null float64
mvp                         2316 non-null float64
nlcs_mvp                    2316 non-null float64
robertoclemente_award       2316 non-null float64
roy                         2316 non-null float64
silver_slugger              2316 non-null float64
tsn_allstar                 2316 non-null float64
triple_crown                2316 non-null float64
ws_mvp                      2316 non-null float64
k_percentage                2316 non-null float64
bb_percentage               2316 non-null float64
ba                          2316 non-null float64
slg_percent                 2316 non-null float64
obp                         2316 non-null float64
ops                         2316 non-null float64
iso                         2316 non-null float64
tb                          2316 non-null float64
gidp                        2316 non-null float64
inducted_y                  2316 non-null float64
dtypes: float64(29), int64(8)
memory usage: 687.6+ KB
```

In [4]:
```
mlb_df.describe()
```

Out[4]:

	g	ab	r	h	2b	3b	h
count	2316.000000	2316.000000	2316.000000	2316.000000	2316.000000	2316.000000	2316.00000
mean	1256.772884	4262.729275	587.518135	1164.006908	205.221934	37.377807	105.43523
std	534.119150	2072.247123	347.233691	623.185015	118.988868	31.602529	108.07855
min	420.000000	789.000000	95.000000	182.000000	26.000000	0.000000	0.00000
25%	847.750000	2664.750000	326.000000	688.000000	114.000000	16.000000	29.00000
50%	1177.000000	3908.000000	506.500000	1046.000000	180.000000	28.000000	72.00000
75%	1568.000000	5433.000000	755.250000	1494.000000	266.000000	49.000000	140.00000
max	3562.000000	14053.000000	2295.000000	4256.000000	792.000000	302.000000	762.00000

8 rows × 37 columns

In [5]:
```
mlb_df.inducted_y.value_counts()
```

Out[5]:
```
0.0    2171
1.0     145
Name: inducted_y, dtype: int64
```

In [6]:
```
# top 50 in hits
top10_hits = mlb_df.groupby(['playerID'])['h'].max()
top10hits = top10_hits.sort_values(ascending=False).head(10)
top10hits
```

Out[6]:
```
playerID
rosepe01     4256
cobbty01     4189
aaronha01    3771
musiast01    3630
speaktr01    3514
jeterde01    3465
yastrca01    3419
molitpa01    3319
collied01    3315
mayswi01     3283
Name: h, dtype: int64
```

In [7]:
```
mlb_df.reset_index(inplace=True)
```

In [8]:
```
df_by_year = pd.read_pickle('df_by_year.pkl')
df_by_year.head()
```

Out[8]:

	playerID	yearID	level_0	teamID	lgID	G	AB	R	H	2B	...	lougehrig_award	mvp	n
0	aaronha01	1954	0	ML1	NL	122	468	58	131	27	...	0.0	0.0	
1	aaronha01	1955	1	ML1	NL	153	602	105	189	37	...	0.0	0.0	
2	aaronha01	1956	2	ML1	NL	153	609	106	200	34	...	0.0	0.0	
3	aaronha01	1957	3	ML1	NL	151	615	118	198	27	...	0.0	1.0	
4	aaronha01	1958	4	ML1	NL	153	601	109	196	34	...	0.0	0.0	

5 rows × 42 columns

In [9]:
```
df_by_year.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 29152 entries, 0 to 29151
Data columns (total 42 columns):
playerID                    29152 non-null object
yearID                      29152 non-null int64
level_0                     29152 non-null int64
teamID                      29152 non-null object
lgID                        29152 non-null object
G                           29152 non-null int64
AB                          29152 non-null int64
R                           29152 non-null int64
H                           29152 non-null int64
2B                          29152 non-null int64
3B                          29152 non-null int64
HR                          29152 non-null int64
RBI                         29152 non-null float64
SB                          29152 non-null float64
BB                          29152 non-null int64
SO                          29152 non-null float64
birthCountry                29152 non-null object
```

In [10]:
```
# move target column to the end of df
cols = list(df_by_year.columns.values)
cols.pop(cols.index('inducted'))
df_by_year = df_by_year[cols+['inducted']]
```

In [11]:
```
df_by_year.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 29152 entries, 0 to 29151
Data columns (total 42 columns):
playerID                    29152 non-null object
yearID                      29152 non-null int64
level_0                     29152 non-null int64
teamID                      29152 non-null object
lgID                        29152 non-null object
G                           29152 non-null int64
AB                          29152 non-null int64
R                           29152 non-null int64
H                           29152 non-null int64
2B                          29152 non-null int64
3B                          29152 non-null int64
HR                          29152 non-null int64
RBI                         29152 non-null float64
SB                          29152 non-null float64
BB                          29152 non-null int64
SO                          29152 non-null float64
```

In [12]:
```
df_by_year.inducted.value_counts()
```

Out[12]:
```
0    26552
Y     2600
Name: inducted, dtype: int64
```

In [13]:
```
# df_by_year['inducted'] = df_by_year['inducted'].map({'Y': 1, '0': 0})
# df_by_year.iloc[0, df.columns.get_loc('inducted')] = 1
```

In [14]:
```
# df_by_year.inducted.value_counts()
```

In [15]:
```
# HOF by position - plot scatter where inducted=Y in different color than everyone
# put labels of players by plots
hof_by_pos = df_by_year.groupby(['POS','inducted'])['H','HR','RBI','R'].mean()
hof_by_pos.sort_values(by='H', ascending=False)
```

Out[15]:

		H	HR	RBI	R
POS	**inducted**				
OF	**Y**	137.146866	14.890552	70.098223	76.314312
3B	**Y**	136.689516	14.322581	70.088710	71.064516
2B	**Y**	133.498462	8.775385	59.172308	72.301538
SS	**Y**	132.207650	6.464481	56.439891	68.336066
1B	**Y**	126.994100	19.277286	77.542773	70.879056
C	**Y**	104.292490	12.802372	59.086957	51.312253
1B	**0**	96.132454	11.401558	51.848966	47.663171
3B	**0**	92.110317	8.911746	44.978247	45.876942
OF	**0**	90.164284	8.813993	42.473387	47.153376
2B	**0**	89.000301	4.941212	34.317154	44.101598
SS	**0**	85.822336	4.658383	33.516138	41.250500
C	**0**	71.681336	6.512377	35.104519	30.315521

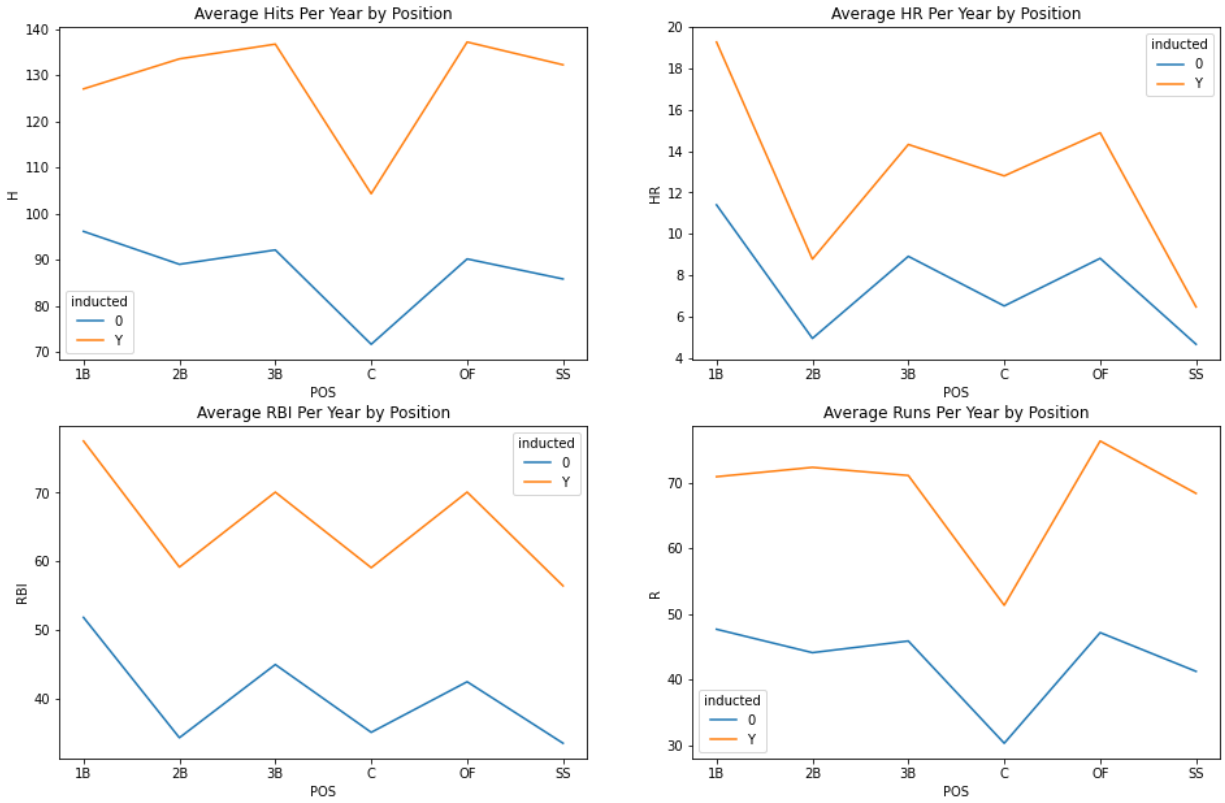
In [16]:
```
hof_by_pos.reset_index(inplace=True)
```

In [17]:
```
def plot_stats():
    statstoplot = ['h','r','hr','rbi','ba','ops']
    for stat in statstoplot:
        plt.figure(figsize=(16,16))
        plt.subplot(stat+1)
        sns.lineplot(hof_by_pos.POS, hof_by_pos.stat, hue=hof_by_pos.inducted)
        plt.title('Average {stat} Per Year by Position')
    return statstoplot
```

In [18]:
```
plt.figure(figsize=(16,16))

plt.subplot(321)
sns.lineplot(hof_by_pos.POS, hof_by_pos.H, hue=hof_by_pos.inducted)
plt.title('Average Hits Per Year by Position')
plt.subplot(322)
sns.lineplot(hof_by_pos.POS, hof_by_pos.HR, hue=hof_by_pos.inducted)
plt.title('Average HR Per Year by Position')
plt.subplot(323)
sns.lineplot(hof_by_pos.POS, hof_by_pos.RBI, hue=hof_by_pos.inducted)
plt.title('Average RBI Per Year by Position')
plt.subplot(324)
sns.lineplot(hof_by_pos.POS, hof_by_pos.R, hue=hof_by_pos.inducted)
plt.title('Average Runs Per Year by Position')
plt.savefig('./images/stats_subplots.png')
```

```
C:\Users\15514\anaconda3\envs\learn-env\lib\site-packages\seaborn\_decorators.p
y:43: FutureWarning: Pass the following variables as keyword args: x, y. From v
ersion 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpre
tation.
  FutureWarning
C:\Users\15514\anaconda3\envs\learn-env\lib\site-packages\seaborn\_decorators.p
y:43: FutureWarning: Pass the following variables as keyword args: x, y. From v
ersion 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpre
tation.
  FutureWarning
C:\Users\15514\anaconda3\envs\learn-env\lib\site-packages\seaborn\_decorators.p
y:43: FutureWarning: Pass the following variables as keyword args: x, y. From v
ersion 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpre
tation.
  FutureWarning
C:\Users\15514\anaconda3\envs\learn-env\lib\site-packages\seaborn\_decorators.p
y:43: FutureWarning: Pass the following variables as keyword args: x, y. From v
ersion 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpre
tation.
  FutureWarning
```



In [19]:
```
hof_by_pos2 = mlb_df.groupby(['inducted_y'])['h','r','rbi','hr','ba','ops'].mean(
hof_by_pos2.sort_values(by='ba', ascending=False)
```

Out[19]:

inducted_y	h	r	rbi	hr	ba	ops
1.0	2356.696552	1273.951724	1196.144828	237.668966	0.273870	0.759253
0.0	1084.347766	541.671580	502.041916	96.603409	0.271942	0.763444

In [20]:
```
hof_by_pos2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Float64Index: 2 entries, 0.0 to 1.0
Data columns (total 6 columns):
h      2 non-null float64
r      2 non-null float64
rbi    2 non-null float64
hr     2 non-null float64
ba     2 non-null float64
ops    2 non-null float64
dtypes: float64(6)
memory usage: 112.0 bytes
```

In [21]:
```
hof_by_pos2.reset_index(inplace=True)
```

In [22]:
```
y = hof_by_pos2['inducted_y']
X = hof_by_pos2.drop('inducted_y', axis=1, inplace=True)
```

In [23]:
```
mlb_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2316 entries, 0 to 2315
Data columns (total 38 columns):
playerID                    2316 non-null object
g                           2316 non-null int64
ab                          2316 non-null int64
r                           2316 non-null int64
h                           2316 non-null int64
2b                          2316 non-null int64
3b                          2316 non-null int64
hr                          2316 non-null int64
rbi                         2316 non-null float64
sb                          2316 non-null float64
bb                          2316 non-null int64
so                          2316 non-null float64
asg_mvp                     2316 non-null float64
baberuth_award              2316 non-null float64
baseball_magazine_allstar   2316 non-null float64
comeback_poy                2316 non-null float64
gold_glove_award            2316 non-null float64
hankaaron_award             2316 non-null float64
hutch_award                 2316 non-null float64
lougehrig_award             2316 non-null float64
mvp                         2316 non-null float64
nlcs_mvp                    2316 non-null float64
robertoclemente_award       2316 non-null float64
roy                         2316 non-null float64
silver_slugger              2316 non-null float64
tsn_allstar                 2316 non-null float64
triple_crown                2316 non-null float64
ws_mvp                      2316 non-null float64
k_percentage                2316 non-null float64
bb_percentage               2316 non-null float64
ba                          2316 non-null float64
slg_percent                 2316 non-null float64
obp                         2316 non-null float64
ops                         2316 non-null float64
iso                         2316 non-null float64
tb                          2316 non-null float64
gidp                        2316 non-null float64
inducted_y                  2316 non-null float64
dtypes: float64(29), int64(8), object(1)
memory usage: 687.7+ KB
```

find top 5 in ['h','r','hr','rbi','ba','ops'] and annotate them on graph

In [24]:
```
# points to show in plots
see_jeter = mlb_df[mlb_df['playerID'].str.contains('jeterde01')]
plot_jeter_x = see_jeter.iloc[0,0]
plot_jeter_y = see_jeter.iloc[0,4]
plot_jeter_x
```

Out[24]:
```
'jeterde01'
```

In [25]:
```
see_prose = mlb_df[mlb_df['playerID'].str.contains('rosepe01')]
plot_prose_x = see_prose.iloc[0,0]
plot_prose_y = see_prose.iloc[0,4]
plot_prose_y
```

Out[25]:
```
4256
```

In [26]:
```
x = top10hits
x
```

Out[26]:
```
playerID
rosepe01     4256
cobbty01     4189
aaronha01    3771
musiast01    3630
speaktr01    3514
jeterde01    3465
yastrca01    3419
molitpa01    3319
collied01    3315
mayswi01     3283
Name: h, dtype: int64
```

In [27]:
```
top10hits_df = pd.DataFrame(data=top10hits)
top10hits_df.head()
```

Out[27]:

playerID	h
rosepe01	4256
cobbty01	4189
aaronha01	3771
musiast01	3630
speaktr01	3514

In [28]:
```
names_t10h = top10hits_df.index.tolist()
names_t10h
```

Out[28]:
```
['rosepe01',
 'cobbty01',
 'aaronha01',
 'musiast01',
 'speaktr01',
 'jeterde01',
 'yastrca01',
 'molitpa01',
 'collied01',
 'mayswi01']
```

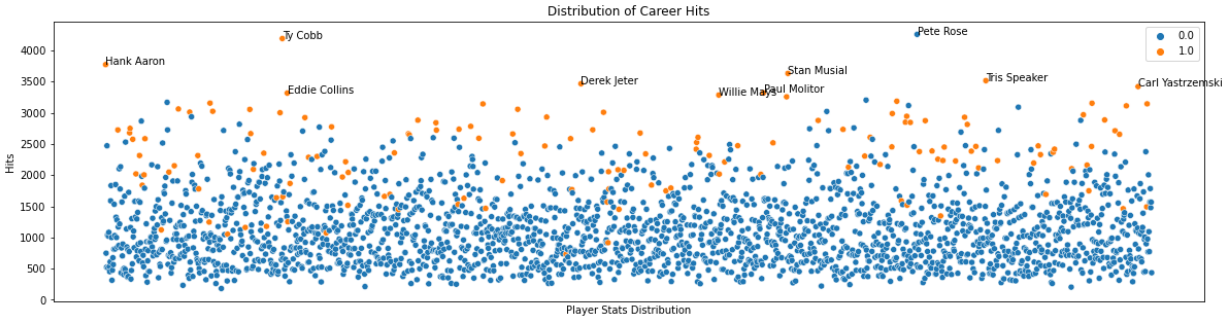
In [29]:
```
hits_t10 = top10hits_df['h'].tolist()
hits_t10
```

Out[29]:
```
[4256, 4189, 3771, 3630, 3514, 3465, 3419, 3319, 3315, 3283]
```

In [30]:
```
pp = ['Pete Rose','Ty Cobb','Hank Aaron','Stan Musial','Tris Speaker',
     'Derek Jeter','Carl Yastrzemski','Paul Molitor','Eddie Collins','Willie Mays
pp
```

Out[30]:
```
['Pete Rose',
 'Ty Cobb',
 'Hank Aaron',
 'Stan Musial',
 'Tris Speaker',
 'Derek Jeter',
 'Carl Yastrzemski',
 'Paul Molitor',
 'Eddie Collins',
 'Willie Mays']
```

In [31]:
```
plt.figure(figsize=(20,5))
plt.title('Distribution of Career Hits')
plt.xlabel('Player Stats Distribution')
plt.ylabel('Hits')
plt.tick_params(axis='x', which='both', bottom=False, labelbottom=False)
sns.scatterplot(data=mlb_df, x=mlb_df['playerID'], y=mlb_df['h'], hue=mlb_df['ind
plt.legend(loc='upper right')
for name, hit, text in zip(names_t10h,hits_t10,pp):
    plt.text(x=name,y=hit,s=text)
plt.savefig('./images/hits.png')
plt.show()
```



In [32]:
```
# top 50 in hr
top10_hr = mlb_df.groupby(['playerID'])['hr'].max()
homeruns_t10 = top10_hr.sort_values(ascending=False).head(10)
homeruns_t10
```

Out[32]:
```
playerID
bondsba01    762
aaronha01    755
ruthba01     714
rodrial01    696
mayswi01     660
pujolal01    656
griffke02    630
thomeji01    612
sosasa01     609
robinfr02    586
Name: hr, dtype: int64
```

In [33]:
```
homeruns_t10_df = pd.DataFrame(data=homeruns_t10)
homeruns_t10_df.head(10)
```

Out[33]:

playerID	hr
bondsba01	762
aaronha01	755
ruthba01	714
rodrial01	696
mayswi01	660
pujolal01	656
griffke02	630
thomeji01	612
sosasa01	609
robinfr02	586

In [34]:
```
names_t10hrs = homeruns_t10_df.index.tolist()
names_t10hrs
```

Out[34]:
```
['bondsba01',
 'aaronha01',
 'ruthba01',
 'rodrial01',
 'mayswi01',
 'pujolal01',
 'griffke02',
 'thomeji01',
 'sosasa01',
 'robinfr02']
```

In [35]:
```
hrs_t10 = homeruns_t10_df['hr'].tolist()
hrs_t10
```

Out[35]:
```
[762, 755, 714, 696, 660, 656, 630, 612, 609, 586]
```

In [36]:
```
hr_pp = ['Barry Bonds', 'Hank Aaron','Babe Ruth','Alex Rodriguez',
     'Willie Mays','Albert Pujols','Ken Griffey Jr.','Jim Thome','Sammy Sosa','Fr
hr_pp
```

Out[36]:
```
['Barry Bonds',
 'Hank Aaron',
 'Babe Ruth',
 'Alex Rodriguez',
 'Willie Mays',
 'Albert Pujols',
 'Ken Griffey Jr.',
 'Jim Thome',
 'Sammy Sosa',
 'Frank Robinson']
```

In [37]:
```
plt.figure(figsize=(20,5))
plt.title('Distribution of Career Home Runs')
plt.xlabel('Player Stats Distribution')
plt.ylabel('Home Runs')
plt.tick_params(axis='x', which='both', bottom=False, labelbottom=False)
sns.scatterplot(data=mlb_df, x=mlb_df['playerID'], y=mlb_df['hr'], hue=mlb_df['in
plt.legend(loc='upper right')
for name, hit, text in zip(names_t10hrs,hrs_t10,hr_pp):
    plt.text(x=name,y=hit,s=text)
plt.savefig('./images/homeruns.png')
plt.show()
```