In this project we are dealling with baseball players data, we have several csv files which each have speciic information aboutthe players and managers(which mainly were player before).

i was intrested in looking at players data including their demographic, salaries, player characteristics, awards and find the players which have won awards and are common through all those datasets, then finding relationships between their salaries and other variables and also find some relationships between number of wins and loses and other variables

In [1]:

```
import numpy as np
import pandas as pd
import os
%matplotlib inline
import matplotlib.pyplot as plt
import plotly.plotly as py
import seaborn as sns
import statsmodels.api as sm
lowess = sm.nonparametric.lowess
```

change working directory

In [2]:

```
os.chdir('D:/DATA_ANALYSIS_NANO_PLUS_DEGREE/Baseball project/Dataset/')
```

list all available files inside the directory

In [3]:

```
onlyfiles = os.listdir('D:/DATA_ANALYSIS_NANO_PLUS_DEGREE/Baseball project/
Dataset/')
onlyfiles
```

Out[3]:

```
['AllstarFull.csv',
 'Appearances.csv',
 'AwardsManagers.csv',
 'AwardsPlayers.csv',
 'AwardsShareManagers.csv',
 'AwardsSharePlayers.csv',
 'Batting.csv',
 'BattingPost.csv',
 'CollegePlaying.csv',
 'corr.csv',
 'Datasets contents.rtf',
```

```
 'Fielding.csv',
 'FieldingOF.csv',
 'FieldingPost.csv',
 'HallOfFame.csv',
 'HomeGames.csv',
 'Managers.csv',
 'ManagersHalf.csv',
 'Master.csv',
 'Parks.csv',
 'Pitching.csv',
 'PitchingPost.csv',
 'readme2014.txt',
 'Salaries.csv',
 'Schools.csv',
 'SeriesPost.csv',
 'Teams.csv',
 'TeamsFranchises.csv',
 'TeamsHalf.csv',
 'test_baseball.csv',
 'test_baseball1.csv',
 '~$tasets contents.rtf']
```

Reading interesting Datafiles

In [4]:

```
# defining a function to read csv files for simplicity and avoiding
repititive codes
def read_csv(fileName):
    return pd.read_csv(fileName , sep=',')
files = {}
names = ['AllstarFull.csv','Appearances.csv','AwardsPlayers.csv','Salaries.
csv','Batting.csv' , 'Master.csv' , 'FieldingPost.csv', 'Pitching.csv']
for name in names:
    files[name.replace(".csv","")] = read_csv(name)

for key,val in files.items():
    exec(key + '=val')
```

Reading Datafiles with specific columns which i was intrested in

after loading batting table and looking at both variables in batting table and pitching table have decided to only choose variables which are not common in both table for pitching , i have used the following command to get colnames which are not common :

colnames_to_use = Pitching.columns.difference(BattingTable.columns)

In [5]:

```
# Fielding = pd.read_csv('Fielding.csv' , sep = ',' , usecols=
['playerID','A', 'DP', 'E', 'InnOuts', 'PB', 'PO', 'POS', 'ZR'])
# Master = pd.read_csv('Master.csv' , sep = ',' , usecols=
['playerID','birthYear','birthMonth','birthDay','birthCountry','deathDay','
```

```
ght','height','bats','throws','debut','finalGame'])
# Pitching = pd.read_csv('Pitching.csv', sep=',' , usecols=['playerID','yearID','BAOpp', 'BFP', 'BK', 'CG', 'ER', 'ERA', 'GF', 'GS', 'IPouts',
#         'L', 'SHO', 'SV', 'W', 'WP'])
```

Merging DataFiles

I Have merged datasets based on PrimaryKeys which here was PlayerID and for some dataframes as we had duplicate column names i have used those names also for merging, like yearID or teamID

In [23]:

```
merged = Batting.merge(Pitching , how = 'inner' , on = ['playerID','teamID','yearID','stint'])
merged = merged.merge(AwardsPlayers , how = 'inner' , on =  ['playerID'])
merged = merged.merge(FieldingPost, how = 'inner', on = ['playerID','teamID'])
merged = merged.merge(Salaries , how = 'inner' , on = ['playerID','teamID','yearID'])
merged = merged.merge(AllstarFull, how = 'inner' , on = ['playerID','teamID','yearID'])
merged = merged.merge(Master , how = 'inner' , on = ['playerID'])
```

In [9]:

```
merged.to_csv('D:/DATA_ANALYSIS_NANO_PLUS_DEGREE/Baseball project/Dataset/test_baseball1.csv' , sep=',' , index=False)
```

Wrangling and Exploration phase for Merged data

In [26]:

```
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 25165 entries, 0 to 25164
Data columns (total 98 columns):
playerID         25165 non-null object
yearID_x         25165 non-null int64
stint            25165 non-null int64
teamID           25165 non-null object
lgID_x           25165 non-null object
G_x              25165 non-null int64
AB               22527 non-null float64
R_x              22527 non-null float64
H_x              22527 non-null float64
2B               22527 non-null float64
3B               22527 non-null float64
HR_x             22527 non-null float64
RBI              22527 non-null float64
SB_x             22527 non-null float64
CS_x             22527 non-null float64
BB_x             22527 non-null float64
```

```
BB_x            22527 non-null float64
SO_x            22527 non-null float64
IBB_x           22527 non-null float64
HBP_x           22527 non-null float64
SH_x            22527 non-null float64
SF_x            22527 non-null float64
GIDP_x          22527 non-null float64
lgID_y          25165 non-null object
W               25165 non-null int64
L               25165 non-null int64
G_y             25165 non-null int64
GS_x            25165 non-null int64
CG              25165 non-null int64
SHO             25165 non-null int64
SV              25165 non-null int64
IPouts          25165 non-null float64
H_y             25165 non-null int64
ER              25165 non-null int64
HR_y            25165 non-null int64
BB_y            25165 non-null int64
SO_y            25165 non-null int64
BAOpp           23974 non-null float64
ERA             25165 non-null float64
IBB_y           25165 non-null float64
WP              25165 non-null float64
HBP_y           25165 non-null float64
BK              25165 non-null int64
BFP             25165 non-null float64
GF              25165 non-null float64
R_y             25165 non-null int64
SH_y            13227 non-null float64
SF_y            13227 non-null float64
GIDP_y          199 non-null float64
awardID         25165 non-null object
yearID_y        25165 non-null int64
lgID_x          25165 non-null object
tie             1177 non-null object
notes           9017 non-null object
yearID          25165 non-null int64
lgID_y          25165 non-null object
round           25165 non-null object
POS             25165 non-null object
G               25165 non-null int64
GS_y            25037 non-null float64
InnOuts         24568 non-null float64
PO              25165 non-null int64
A               25165 non-null int64
E               25165 non-null int64
DP              25165 non-null int64
TP              25165 non-null int64
PB              333 non-null float64
SB_y            23710 non-null float64
CS_y            23710 non-null float64
lgID_x          25165 non-null object
salary          25165 non-null int64
gameNum         25165 non-null int64
gameID          25165 non-null object
lgID_y          25165 non-null object
GP              25162 non-null float64
startingPos     5996 non-null float64
birthYear       25165 non-null float64
```

```
birthMonth        25165 non-null float64
birthDay          25165 non-null float64
birthCountry      25165 non-null object
birthState        25165 non-null object
birthCity         25165 non-null object
deathYear         64 non-null float64
deathMonth        64 non-null float64
deathDay          64 non-null float64
deathCountry      64 non-null object
deathState        64 non-null object
deathCity         64 non-null object
nameFirst         25165 non-null object
nameLast          25165 non-null object
nameGiven         25165 non-null object
weight            25165 non-null float64
height            25165 non-null float64
bats              25165 non-null object
throws            25165 non-null object
debut             25165 non-null object
finalGame         25165 non-null object
retroID           25165 non-null object
bbrefID           25165 non-null object
dtypes: float64(42), int64(27), object(29)
memory usage: 19.0+ MB
```

by using info function we can get an insight about each variable and if there is any missing value, we can see in general we have 486 records and we can see some variables have missing values, so i am defining a threshhold for missing values, which will be based on ratio of missing values to number of records and consider 25%

In [27]:

```
def getPercentageMissing(data):
    num = merged.isnull().sum()
    return 100*(num/486)
index=getPercentageMissing(merged) > 25.0
merged.drop(index[index.values].keys(), axis=1,inplace=True)
```

In [28]:

```
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 25165 entries, 0 to 25164
Data columns (total 64 columns):
playerID          25165 non-null object
yearID_x          25165 non-null int64
stint             25165 non-null int64
teamID            25165 non-null object
lgID_x            25165 non-null object
G_x               25165 non-null int64
lgID_y            25165 non-null object
W                 25165 non-null int64
L                 25165 non-null int64
```

```
G_y             25165 non-null int64
GS_x            25165 non-null int64
CG              25165 non-null int64
SHO             25165 non-null int64
SV              25165 non-null int64
IPouts          25165 non-null float64
H_y             25165 non-null int64
ER              25165 non-null int64
HR_y            25165 non-null int64
BB_y            25165 non-null int64
SO_y            25165 non-null int64
ERA             25165 non-null float64
IBB_y           25165 non-null float64
WP              25165 non-null float64
HBP_y           25165 non-null float64
BK              25165 non-null int64
BFP             25165 non-null float64
GF              25165 non-null float64
R_y             25165 non-null int64
awardID         25165 non-null object
yearID_y        25165 non-null int64
lgID_x          25165 non-null object
yearID          25165 non-null int64
lgID_y          25165 non-null object
round           25165 non-null object
POS             25165 non-null object
G               25165 non-null int64
PO              25165 non-null int64
A               25165 non-null int64
E               25165 non-null int64
DP              25165 non-null int64
TP              25165 non-null int64
lgID_x          25165 non-null object
salary          25165 non-null int64
gameNum         25165 non-null int64
gameID          25165 non-null object
lgID_y          25165 non-null object
GP              25162 non-null float64
birthYear       25165 non-null float64
birthMonth      25165 non-null float64
birthDay        25165 non-null float64
birthCountry    25165 non-null object
birthState      25165 non-null object
birthCity       25165 non-null object
nameFirst       25165 non-null object
nameLast        25165 non-null object
nameGiven       25165 non-null object
weight          25165 non-null float64
height          25165 non-null float64
bats            25165 non-null object
throws          25165 non-null object
debut           25165 non-null object
finalGame       25165 non-null object
retroID         25165 non-null object
bbrefID         25165 non-null object
dtypes: float64(13), int64(27), object(24)
memory usage: 12.5+ MB
```

we can see there is no more NULL or NA value in the dataset

We can see there is no more NULL or NA value in the dataset

In [33]:

```
merged.isnull()
```

Out[33]:

	playerID	yearID_x	stint	teamID	lgID_x	G_x	lgID_y	W	L	G_y	...	nameLas
0	False	False	False	False	False	False	False	False	False	False	...	False
1	False	False	False	False	False	False	False	False	False	False	...	False
2	False	False	False	False	False	False	False	False	False	False	...	False
3	False	False	False	False	False	False	False	False	False	False	...	False
4	False	False	False	False	False	False	False	False	False	False	...	False
5	False	False	False	False	False	False	False	False	False	False	...	False
6	False	False	False	False	False	False	False	False	False	False	...	False
7	False	False	False	False	False	False	False	False	False	False	...	False
8	False	False	False	False	False	False	False	False	False	False	...	False
9	False	False	False	False	False	False	False	False	False	False	...	False
10	False	False	False	False	False	False	False	False	False	False	...	False
11	False	False	False	False	False	False	False	False	False	False	...	False
12	False	False	False	False	False	False	False	False	False	False	...	False
13	False	False	False	False	False	False	False	False	False	False	...	False
14	False	False	False	False	False	False	False	False	False	False	...	False
15	False	False	False	False	False	False	False	False	False	False	...	False
16	False	False	False	False	False	False	False	False	False	False	...	False
17	False	False	False	False	False	False	False	False	False	False	...	False
18	False	False	False	False	False	False	False	False	False	False	...	False
19	False	False	False	False	False	False	False	False	False	False	...	False
20	False	False	False	False	False	False	False	False	False	False	...	False
21	False	False	False	False	False	False	False	False	False	False	...	False
22	False	False	False	False	False	False	False	False	False	False	...	False
23	False	False	False	False	False	False	False	False	False	False	...	False
24	False	False	False	False	False	False	False	False	False	False	...	False
25	False	False	False	False	False	False	False	False	False	False	...	False
26	False	False	False	False	False	False	False	False	False	False	...	False
27	False	False	False	False	False	False	False	False	False	False	...	False
28	False	False	False	False	False	False	False	False	False	False	...	False
29	False	False	False	False	False	False	False	False	False	False	...	False

...	playerID	yearID_x	stint	teamID	lgID_x	G_x	lgID_y	W	L	G_y	...	nameLas
25135	False	False	False	False	False	False	False	False	False	False	...	False
25136	False	False	False	False	False	False	False	False	False	False	...	False
25137	False	False	False	False	False	False	False	False	False	False	...	False
25138	False	False	False	False	False	False	False	False	False	False	...	False
25139	False	False	False	False	False	False	False	False	False	False	...	False
25140	False	False	False	False	False	False	False	False	False	False	...	False
25141	False	False	False	False	False	False	False	False	False	False	...	False
25142	False	False	False	False	False	False	False	False	False	False	...	False
25143	False	False	False	False	False	False	False	False	False	False	...	False
25144	False	False	False	False	False	False	False	False	False	False	...	False
25145	False	False	False	False	False	False	False	False	False	False	...	False
25146	False	False	False	False	False	False	False	False	False	False	...	False
25147	False	False	False	False	False	False	False	False	False	False	...	False
25148	False	False	False	False	False	False	False	False	False	False	...	False
25149	False	False	False	False	False	False	False	False	False	False	...	False
25150	False	False	False	False	False	False	False	False	False	False	...	False
25151	False	False	False	False	False	False	False	False	False	False	...	False
25152	False	False	False	False	False	False	False	False	False	False	...	False
25153	False	False	False	False	False	False	False	False	False	False	...	False
25154	False	False	False	False	False	False	False	False	False	False	...	False
25155	False	False	False	False	False	False	False	False	False	False	...	False
25156	False	False	False	False	False	False	False	False	False	False	...	False
25157	False	False	False	False	False	False	False	False	False	False	...	False
25158	False	False	False	False	False	False	False	False	False	False	...	False
25159	False	False	False	False	False	False	False	False	False	False	...	False
25160	False	False	False	False	False	False	False	False	False	False	...	False
25161	False	False	False	False	False	False	False	False	False	False	...	False
25162	False	False	False	False	False	False	False	False	False	False	...	False
25163	False	False	False	False	False	False	False	False	False	False	...	False
25164	False	False	False	False	False	False	False	False	False	False	...	False

25165 rows × 64 columns

we can see some of the variables(coloumns) have variance of zero means all the players have same value and we know

there is no meaningful information in those variables so will remove them

Also i have classified players based on their salary in three categories LOW,MEDIUM,HIGH which i got the threshholds from salary column, 0 to first quartile as LOW, first quartile throguh 3rd quartile as MEDIUM, 3rd quartile to maximum value as HIGH

In [47]:

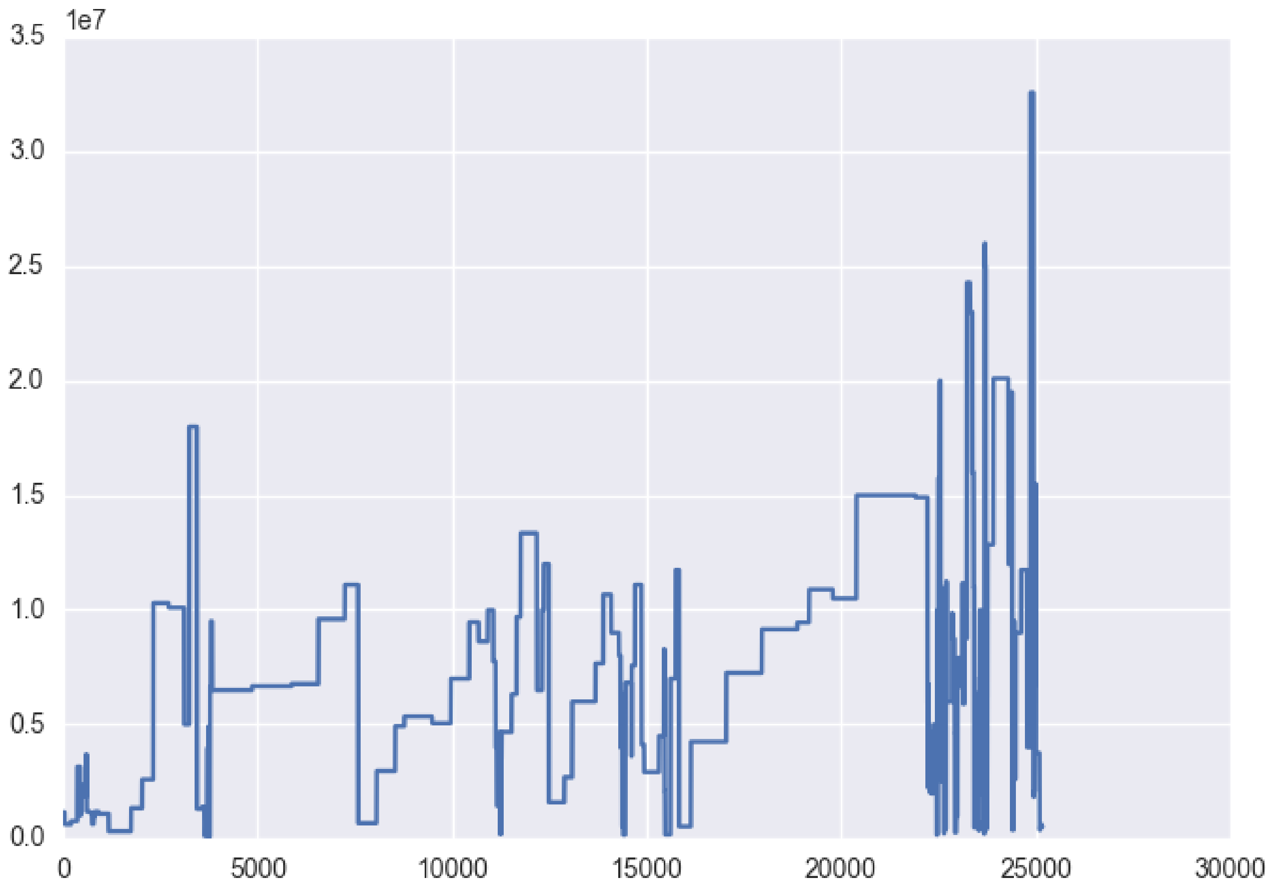
```
bins = [0,4.250000e+06,1.010000e+07,3.257100e+07]
classes = ['LOW','MEDIUM','HIGH']
merged['salary_class'] = pd.cut(merged['salary'], bins, labels=classes)
```

In [53]:

```
plt.plot(merged['salary'])
```

Out[53]:

```
[<matplotlib.lines.Line2D at 0x1743fcf8>]
```



In [54]:

```
index=merged.std() == 0.0
merged.drop(index[index.values].keys(), axis=1,inplace=True)
```

Compute pairwise correlation of columns, excluding NA/null values with .corr function of numpy , only

NA/null values with .corr function of numpy , only considered variables(columns) belonging to pitching and batting data sets

In [55]:

```
merged.corr(method='pearson') >0.75
```

Out[55]:

	yearID_x	stint	G_x	W	L	G_y	GS_x	CG	SHO	SV	...	A	E
yearID_x	True	False	False	False	False	False	False	False	False	False	...	False	False
stint	False	True	False	False	False	False	False	False	False	False	...	False	False
G_x	False	False	True	False	False	True	False	False	False	True	...	False	False
W	False	False	False	True	False	False	True	False	False	False	...	False	False
L	False	False	False	False	True	False	False	False	False	False	...	False	False
G_y	False	False	True	False	False	True	False	False	False	True	...	False	False
GS_x	False	False	False	True	False	False	True	False	False	False	...	False	False
CG	False	False	False	False	False	False	False	True	True	False	...	False	False
SHO	False	False	False	False	False	False	False	True	True	False	...	False	False
SV	False	False	True	False	False	True	False	False	False	True	...	False	False
IPouts	False	False	False	True	False	False	True	False	False	False	...	False	False
H_y	False	False	False	True	True	False	True	False	False	False	...	False	False
ER	False	False	False	True	True	False	True	False	False	False	...	False	False
HR_y	False	False	False	False	False	False	True	False	False	False	...	False	False
BB_y	False	False	False	False	False	False	True	False	False	False	...	False	False
SO_y	False	False	False	True	False	False	True	False	False	False	...	False	False
ERA	False	False	False	False	False	False	False	False	False	False	...	False	False
IBB_y	False	False	False	False	False	False	False	False	False	False	...	False	False
WP	False	False	False	False	False	False	False	False	False	False	...	False	False
HBP_y	False	False	False	False	False	False	False	False	False	False	...	False	False
BK	False	False	False	False	False	False	False	False	False	False	...	False	False
BFP	False	False	False	True	True	False	True	False	False	False	...	False	False
GF	False	False	True	False	False	True	False	False	False	True	...	False	False
R_y	False	False	False	True	True	False	True	False	False	False	...	False	False
yearID_y	False	False	False	False	False	False	False	False	False	False	...	False	False
yearID	False	False	False	False	False	False	False	False	False	False	...	False	False
G	False	False	False	False	False	False	False	False	False	False	...	False	False
PO	False	False	False	False	False	False	False	False	False	False	...	False	False
A	False	False	False	False	False	False	False	False	False	False	...	True	False

	yearID_x	stint	G_x	W	L	G_y	GS_x	CG	SHO	SV	...	A	E
E	False	False	False	False	False	False	False	False	False	False	...	False	True
DP	False	False	False	False	False	False	False	False	False	False	...	False	False
salary	False	False	False	False	False	False	False	False	False	False	...	False	False
GP	False	False	False	False	False	False	False	False	False	False	...	False	False
birthYear	False	False	False	False	False	False	False	False	False	False	...	False	False
birthMonth	False	False	False	False	False	False	False	False	False	False	...	False	False
birthDay	False	False	False	False	False	False	False	False	False	False	...	False	False
weight	False	False	False	False	False	False	False	False	False	False	...	False	False
height	False	False	False	False	False	False	False	False	False	False	...	False	False

38 rows × 38 columns

i have uesd 0.75 as a threshhold which above that define a hoigh correlation, and as we can see for exanple AB is highly correlated with R,H,RBI

i was intrested to see which variables are corrolated with the salary

i have defined a threshhold of 0.25 and we can see only yearID's and BirthYear are correlated with the salary and based on the plots we can see highest salaries is relalted to mainly younger players which make sense as they are playing in later times and salaries expected to be higher.

In [75]:

```
a = merged.corr(method='pearson') >0.25
```

In [76]:

```
a.salary
```

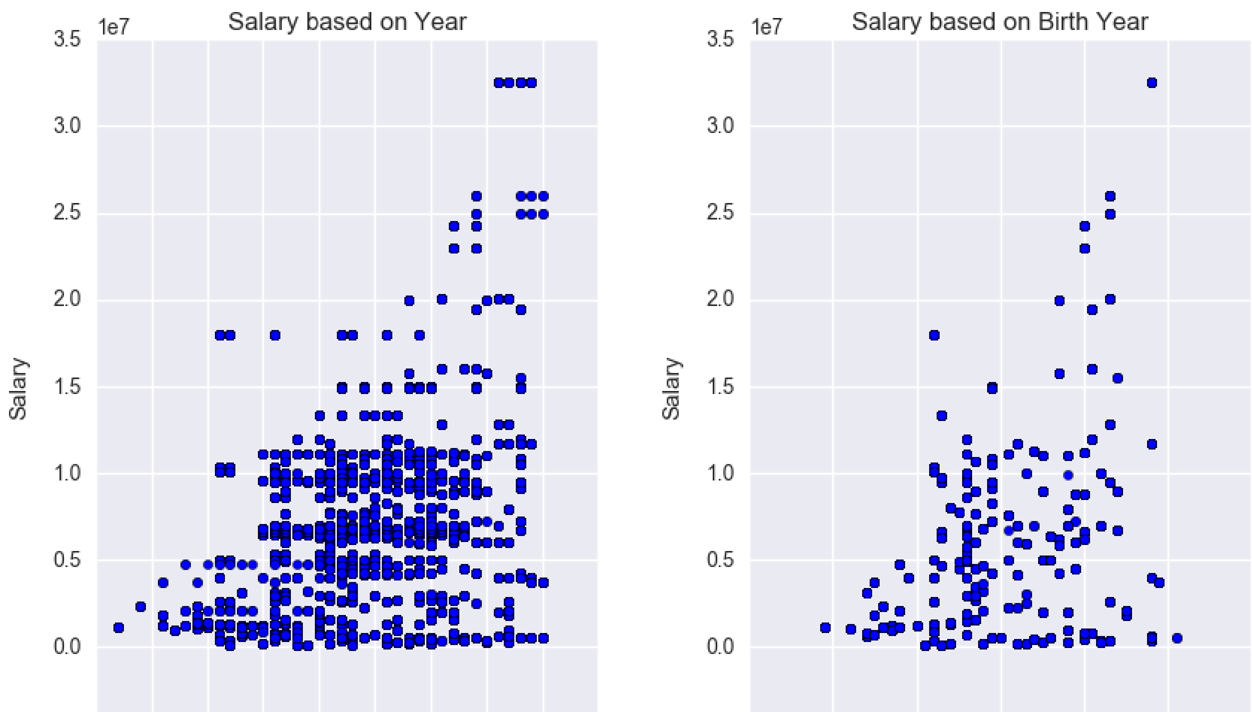
Out[76]:

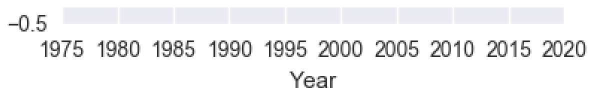
```
yearID_x        True
stint          False
G_x            False
W              False
L              False
G_y            False
GS_x           False
CG             False
SHO            False
SV             False
IPouts         False
H_y            False
ER             False
```

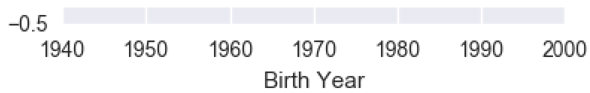
```
HR_y            False
BB_y            False
SO_y            False
ERA             False
IBB_y           False
WP              False
HBP_y           False
BK              False
BFP             False
GF              False
R_y             False
yearID_y         True
yearID           True
G               False
PO              False
A               False
E               False
DP              False
salary           True
GP              False
birthYear        True
birthMonth      False
birthDay        False
weight          False
height          False
Name: salary, dtype: bool
```

In [77]:

```
fig, (a,b) = plt.subplots(1,2,figsize=(10, 6))
a.scatter(merged['yearID_y'],merged['salary'] )
a.set_xlabel("Year")
a.set_ylabel("Salary")
a.set_title("Salary based on Year")
b.scatter(merged['birthYear'],merged['salary'] )
b.set_xlabel("Birth Year")
b.set_ylabel("Salary")
b.set_title("Salary based on Birth Year")
fig.subplots_adjust(hspace = 0.5 , wspace = 0.3)
```







i also considerd same threshold for wining and losing , as we can see wining and losing the game are correlated with GS (games started), and Ipouts (Outs Pitched (innings pitched x 3)), and ER(earned runs) also correlated with both wining and losing which is making sense.

we can see Wining is correlated with AB(at bats) and losing is not.

In [78]:

```
Batting.iloc[:,6:].corr(method='pearson')
```

Out[78]:

	AB	R	H	2B	3B	HR	RBI	SB	CS
AB	1.000000	0.950973	0.987312	0.929086	0.711915	0.689583	0.919150	0.602521	0.68256
R	0.950973	1.000000	0.966463	0.917910	0.742934	0.723842	0.923072	0.657821	0.68321
H	0.987312	0.966463	1.000000	0.945326	0.735753	0.698043	0.934702	0.610866	0.68649
2B	0.929086	0.917910	0.945326	1.000000	0.652233	0.719772	0.915200	0.522245	0.61325
3B	0.711915	0.742934	0.735753	0.652233	1.000000	0.341271	0.659402	0.613622	0.65267
HR	0.689583	0.723842	0.698043	0.719772	0.341271	1.000000	0.833147	0.256216	0.36660
RBI	0.919150	0.923072	0.934702	0.915200	0.659402	0.833147	1.000000	0.501424	0.55577
SB	0.602521	0.657821	0.610866	0.522245	0.613622	0.256216	0.501424	1.000000	0.78932
CS	0.682560	0.683213	0.686499	0.613252	0.652677	0.366609	0.555778	0.789324	1.00000
BB	0.866973	0.889075	0.863757	0.830856	0.589784	0.726364	0.853383	0.534996	0.59444
SO	0.819089	0.768581	0.777413	0.775100	0.449108	0.790554	0.789303	0.435791	0.51434
IBB	0.637314	0.645149	0.650866	0.630030	0.402967	0.665754	0.697137	0.288021	0.35753
HBP	0.625348	0.636861	0.621093	0.598783	0.444970	0.481898	0.601970	0.446851	0.41864
SH	0.500656	0.448621	0.482565	0.392423	0.525831	0.050490	0.361164	0.429918	0.45268
SF	0.803583	0.784240	0.803769	0.786475	0.526516	0.697388	0.829233	0.403140	0.48598
GIDP	0.868972	0.810634	0.861458	0.827089	0.547360	0.690161	0.838809	0.377281	0.49839

when player is in batting position we can see AB = at batting position is highly correlated with variables like(R = runs abd H =hits) which make sense due to being in batting position gives you the chance to hit and then do the run,

also we can see correlation gradually dropping for variables(2B= double hit and 3B = tripple hit respectively) as we can imagine having a double hit is more likely to happen when at batting position than a tripple hits.

looking at correlation between in batting position and losing or wining it shoes that the correlation is close to 0 so it is unlikly that being in a batting position will effect losing or

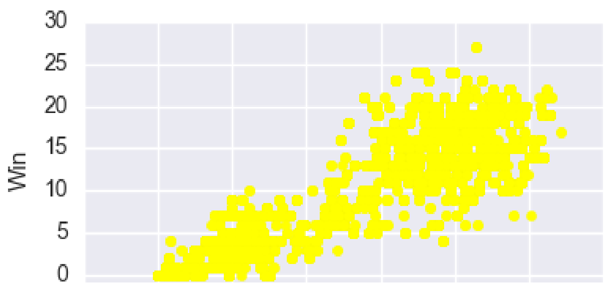
correlation is close to 0 so it is unlikely that being in a batting position will effect losing or wining based on retrospective data.

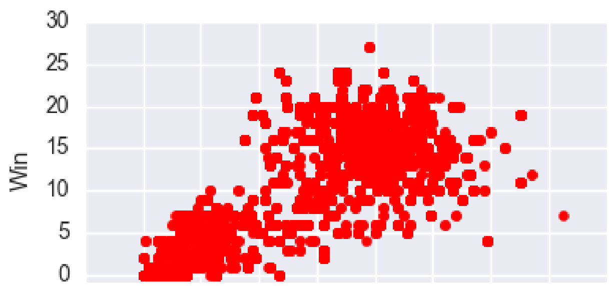
In [79]:

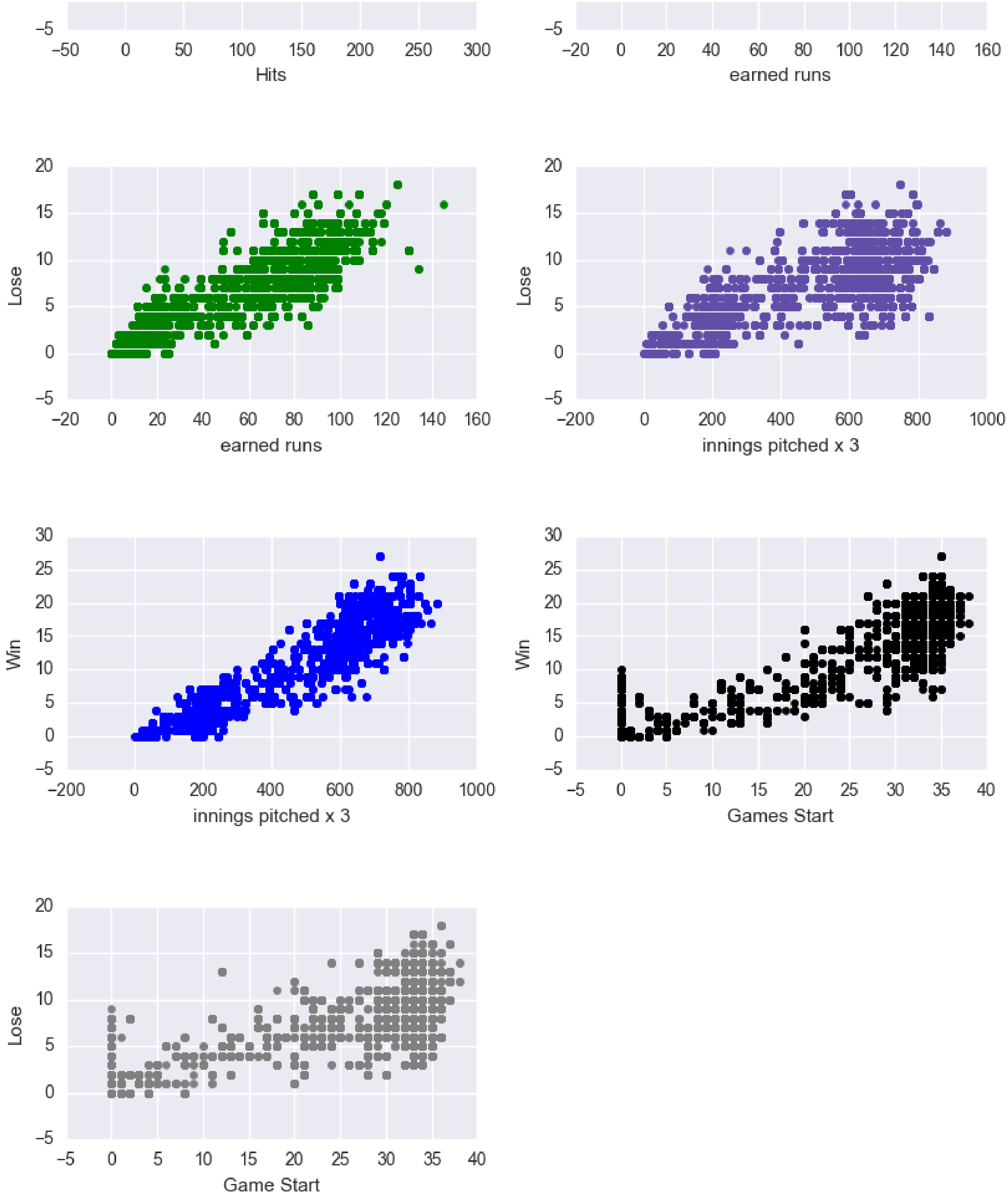
```
data_numeric = merged._get_numeric_data()
data_numeric['salary_class'] = merged.salary_class
```

In [81]:

```
plt.subplot(421)
plt.scatter(data_numeric['H_y'],data_numeric['W'],color= 'yellow')
plt.xlabel('Hits')
plt.ylabel('Win')
plt.grid(True)
plt.tight_layout(pad=0.4, w_pad=4, h_pad=4,rect=[1, 0, 2, 2])

plt.subplot(422)
plt.scatter(data_numeric['ER'],data_numeric['W'],color= 'red')
plt.xlabel('earned runs')
plt.ylabel('Win')
plt.grid(True)

plt.subplot(423)
plt.scatter(data_numeric['ER'],data_numeric['L'],color= 'green')
plt.xlabel('earned runs')
plt.ylabel('Lose')
plt.grid(True)

plt.subplot(424)
plt.scatter(data_numeric['IPouts'],data_numeric['L'],color= '#624ea7')
plt.xlabel('innings pitched x 3')
plt.ylabel('Lose')
plt.grid(True)

plt.subplot(425)
plt.scatter(data_numeric['IPouts'],data_numeric['W'],color= 'blue')
plt.xlabel('innings pitched x 3')
plt.ylabel('Win')
plt.grid(True)

plt.subplot(426)
plt.scatter(data_numeric['GS_x'],data_numeric['W'],color= 'black')
plt.xlabel('Games Start')
plt.ylabel('Win')
plt.grid(True)

plt.subplot(427)
plt.scatter(data_numeric['GS_x'],data_numeric['L'],color= 'grey')
plt.xlabel('Game Start')
plt.ylabel('Lose')
plt.grid(True)
```







as we can see based on the scatter plot, The number of wins increases as number of Hits increase, also more earned runs will casue in both wins and loses which if we look at the numbers increasing the earned runs will ending up in higher numbers of wins than loses, and we can see there are noises in number of loses based on earned runns which might be randoms, which this condition is also for innings pitched and Game Start.

Also we can see for both win and lose based on game start we have increase in number of los and wins while the game

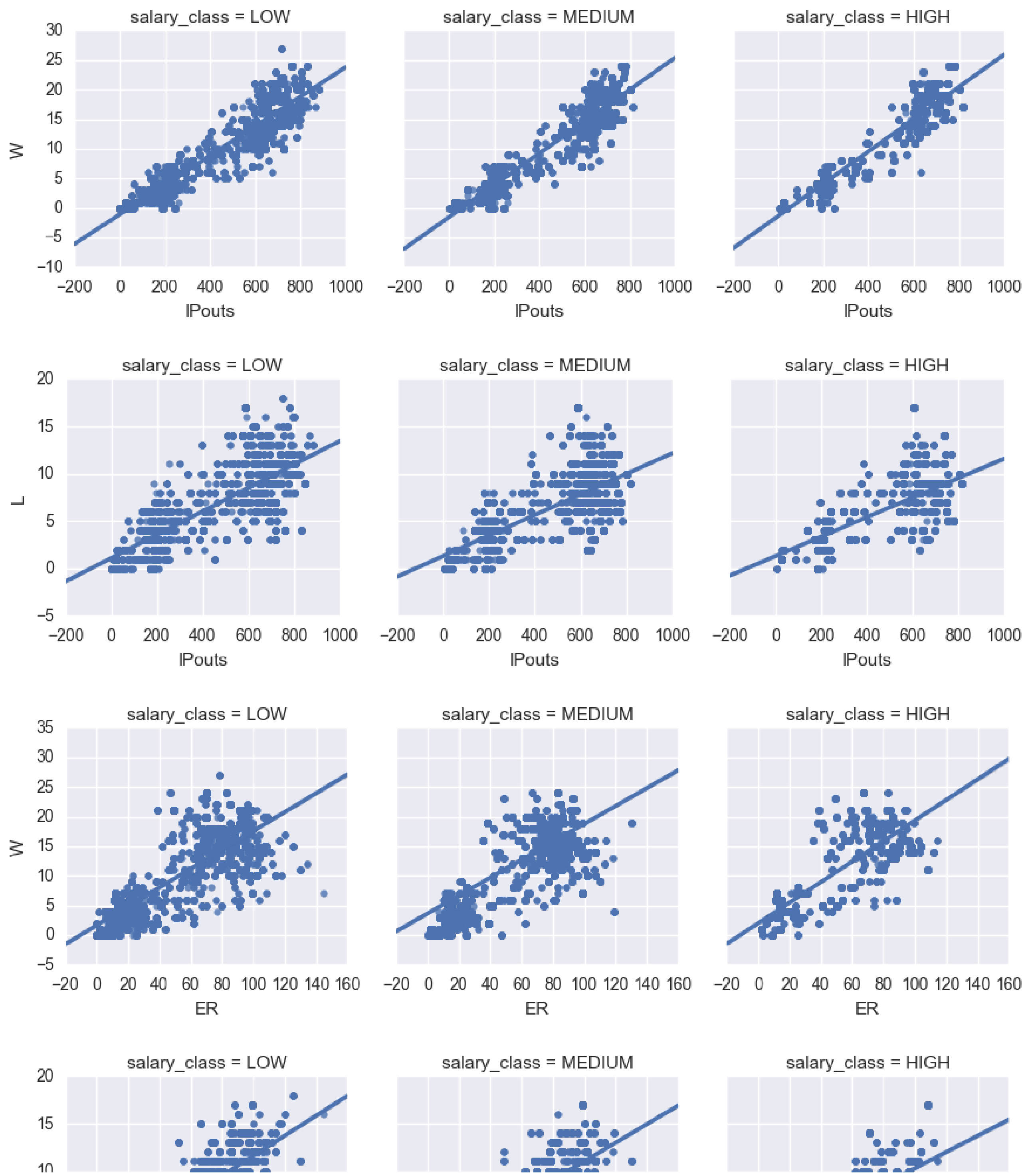
we have increase in number of ies and wins while the game start value is 0 which is because some teams might have been guest and did not start a game for couple a weeks

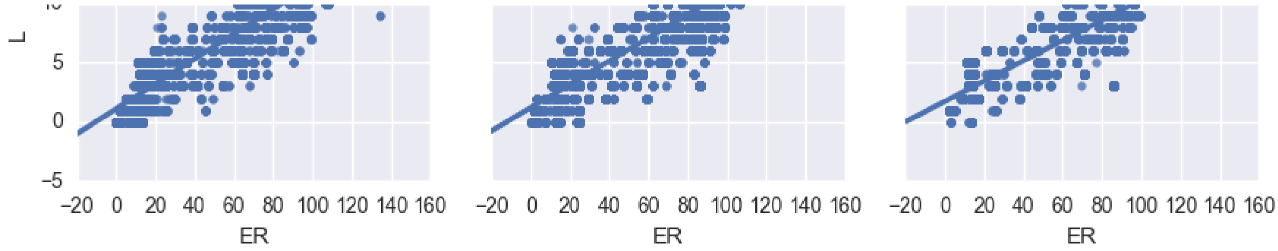
In [71]:

```
sns.lmplot(x = "IPouts" , y ="W" , col="salary_class" , data = data_numeric
, size= 3 , col_wrap=3)
sns.lmplot(x = "IPouts" , y ="L" , col="salary_class" , data = data_numeric
, size= 3 , col_wrap=3)
sns.lmplot(x = "ER" , y ="W" , col="salary_class" , data = data_numeric , si
ze= 3 , col_wrap=3)
sns.lmplot(x = "ER" , y ="L" , col="salary_class" , data = data_numeric , si
ze= 3 , col_wrap=3)
```

Out[71]:

```
<seaborn.axisgrid.FacetGrid at 0x18781780>
```





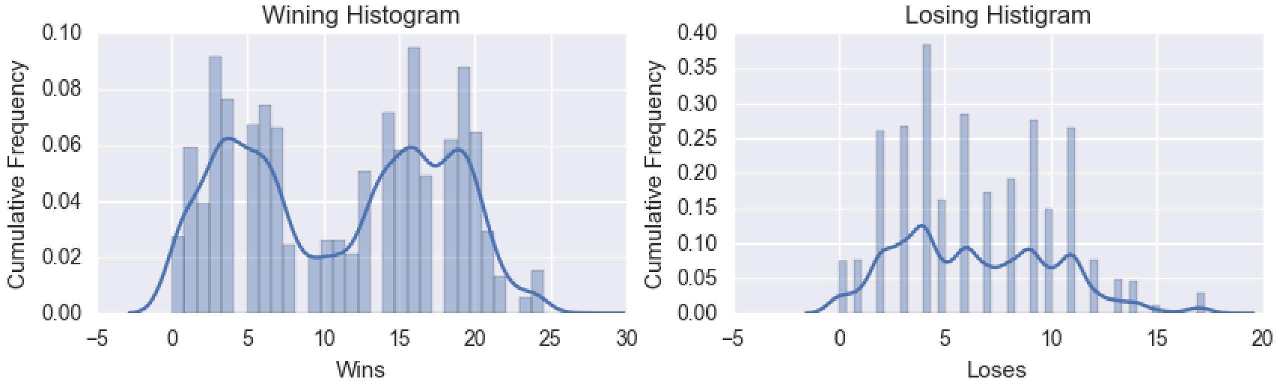
by using salary_class variable and using a linear relation between Los, Win, ER, Ipouts we can see the following results:

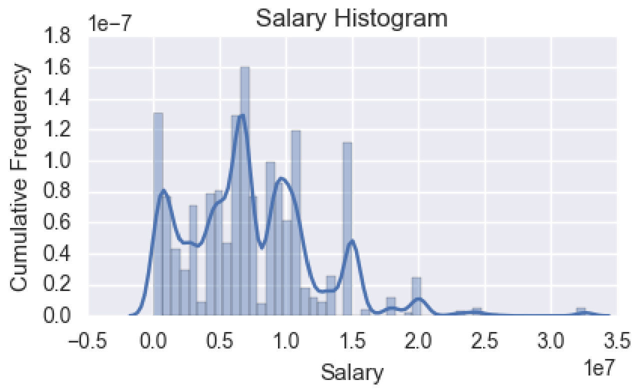
1- we have a good linear relation between IPouts and number of wins but for Ipouts and number of lost games we can see there exists random noises and relation is not linear it can be quadtratic as it seems to have a curve

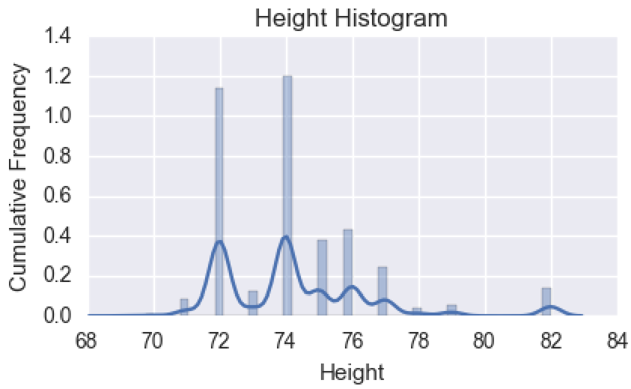
2- we also can find a linear relation between win and ER wich we also have some linear relation between ER and lose but we can see the number of lost are much less than number of wins, basically e=increasing in ER leading to more number of wins than lose

In [72]:

```
# fig, axs = plt.subplots(ncols=2, figsize=(10, 4))
fig, ((a,b),(c,d)) = plt.subplots(2,2,figsize=(10, 6),)
sns.distplot(data_numeric['W'] , ax=a)
a.set_xlabel('Wins')
a.set_ylabel('Cumulative Frequency')
a.set_title('Wining Histogram')
sns.distplot(data_numeric['L'] , ax = b)
b.set_xlabel('Loses')
b.set_ylabel('Cumulative Frequency')
b.set_title('Losing Histigram')
sns.distplot(data_numeric['salary'],ax = c)
c.set_xlabel('Salary')
c.set_ylabel('Cumulative Frequency')
c.set_title('Salary Histogram')
sns.distplot(data_numeric['height'] , ax = d)
d.set_xlabel('Height')
d.set_ylabel('Cumulative Frequency')
d.set_title('Height Histogram')
fig.subplots_adjust(hspace = 0.5,wspace = 0.2)
```







looking at distrubution plot of Salary, Height, Wins, Losses which i was intrested to see, we can see Salary is skewed to right, and Height is the only one close to normal distrubution.

we can see yearID has a positive correlation with salary which make sense as we are getting closer to present the salaries have also increased which we can see for players which are younger we have higher salaries.

Conclusion :

1- Salary of players was not correlated to information we had in the tables, as the highest correlation did not meet 0.25, the approach might be if we use different dataset and look throguh all possible variables, we also noticed age and playing leauge year have some correlation with salary as colser to current date salary mainly increases.

2- For winining we have found several variables which were highly correlated (greater than 0.75), allso we found linear relations between wins and for example number of Hits, Ipouts and Earned Rounds.

3- For Losing also we have found some variables with high correlations (greater than 0.75), and also we have seen Ipouts and Earned rounds have a linear relations with number of loses.

what we can see here is Earned rounds and Ipouts bouth increase in win and lose, but if we look at the plots we can see we have much more increases in number of wins than loses.

In [ ]: