Exploratory Data Analysis for Machine Learning

Brief description of the data set and a summary of its attributes

The data set contains historical baseball statistics for Major League Baseball teams from 1871 through 2014. The statistics cover all personal performance data such as batting, pitching, and also “general” team performance data. In this study, I am going to focus on team performance data and explore it as the baseline for later modeling and predicting stages.

Initial plan for data exploration

The data exploration plan is described as following:

1. Describe the data and its insight.
    a. The source of data.
    b. Feature identification (whether data contains Categorical or Numerical variables or a mix of both).
    c. Relationship between features (How features are dependent on each other).
2. Data cleaning and data integrity
    a. To ensure all data present. (If we have collected data for three years, any week missing can be a problem in later stages.)
    b. Are there any missing values present?
    c. Are there any outliers in the dataset?
3. Feature engineering.
    a. To create new features from the existing data set if necessary.

Actions taken for data cleaning and feature engineering

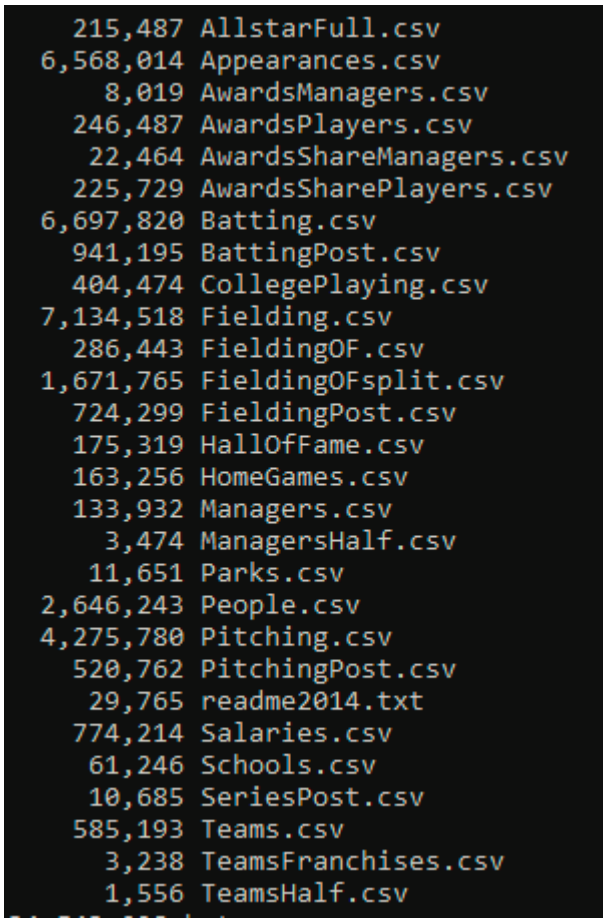
Based on the data exploratory plan, here are the actions I have taken for the data exploratory:

1. Describe the data set.
    As a baseball fan, it is always fun to analyze what a winning team is made of. How do batting, pitching and defense performance contribute to a team’s success? Is it purely a money game of acquiring superstars, or we can develop some team chemistry with highly potential but less expensive players?

Therefore, I start my study with a baseball data set which contains comprehensive Major League Baseball records from 1871 through 2014.

The full data set can be retrieved here:
http://www.seanlahman.com/baseball-archive/statistics/

The data set includes 27 CSV data files and one readme TXT file.



2. Identify the features and check the data types.
   The study will start from the team's overall performance. Through the data set I would like to figure out key offensive and defensive factors that contribute to a team's annual winnings and ranking. Furthermore, it is also interesting to see how a team's overall payroll correlates to its performance. Therefore, two CSV files, Teams.csv and Salaries.csv will become the major data source of analytic.

   Let's check the columns and data types of these 2 CSV files:
   a. Teams.csv

Columns	Description	Data Type

yearID	Year	int64
lgID	League	object
teamID	Team	object
franchID	Franchise	object
divID	Team's division	object
Rank	Position in final standings	int64
G	Games played	int64
GHome	Games played at home	float64
W	Wins	int64
L	Losses	int64
DivWin	Division Winner (Y or N)	object
WCWin	Wild Card Winner (Y or N)	object
LgWin	League Champion(Y or N)	object
WSWin	World Series Winner (Y or N)	object
R	Runs scored	int64
AB	At bats	int64
H	Hits by batters	int64
2B	Doubles	int64
3B	Triples	int64
HR	Homeruns by batters	int64
BB	Walks by batters	int64
SO	Strikeouts by batters	float64
SB	Stolen bases	float64

CS	Caught stealing	float64
HBP	Batters hit by pitch	float64
SF	Sacrifice flies	float64
RA	Opponents runs scored	int64
ER	Earned runs allowed	int64
ERA	Earned run average	float64
CG	Complete games	int64
SHO	Shutouts	int64
SV	Saves	int64
IPOuts	Outs Pitched (innings pitched x 3)	int64
HA	Hits allowed	int64
HRA	Homeruns allowed	int64
BBA	Walks allowed	int64
SOA	Strikeouts by pitchers	int64
E	Errors	int64
DP	Double Plays	float64
FP	Fielding percentage	float64
name	Team's full name	object
park	Name of team's home ballpark	object
attendance	Home attendance total	float64
BPF	Three-year park factor for batters	int64
PPF	Three-year park factor for pitchers	int64

teamIDBR	Team ID used by Baseball Reference website	object
teamIDlahman45	Team ID used in Lahman database version 4.5	object
teamIDretro	Team ID used by Retrosheet	object

b. Salaries.csv

Columns	Description	Data Type
yearID	Year	int64
teamID	Team	object
lgID	League	object
playerID	Player ID code	object
salary	Salary	int64

Through a simple correlation check method, some positive correlation exists between team's wins and key performance indexes. For example, wins and runs scored, wins and game saves, wins and hits, etc.
The summary is as following:

	W	L	DivWin	WCWin	LgWin	WSWin	R	AB	H	2B	...	SHO	SV	IPouts	I
W	1.000000	-0.701145	0.580709	0.230986	0.348168	0.232024	0.582912	0.411887	0.501754	0.330347	...	0.484777	0.687579	0.451171	-0.0847
L	-0.701145	1.000000	-0.520748	-0.203047	-0.312479	-0.206917	-0.290985	0.338459	0.030939	-0.006686	...	-0.330802	-0.440848	0.312185	0.5871
DivWin	0.580709	-0.520748	1.000000	-0.104925	0.444408	0.308692	0.308543	0.091223	0.201641	0.112982	...	0.296759	0.385054	0.115989	-0.1301
WCWin	0.230986	-0.203047	-0.104925	1.000000	0.174031	0.120884	0.202187	0.052385	0.120521	0.168941	...	0.046959	0.107683	0.045902	-0.0378
LgWin	0.348168	-0.312479	0.444408	0.174031	1.000000	0.694615	0.162831	0.053000	0.116135	0.053798	...	0.195257	0.240168	0.068610	-0.1182
WSWin	0.232024	-0.206917	0.308692	0.120884	0.694615	1.000000	0.110100	0.034746	0.088074	0.048713	...	0.151786	0.149955	0.042283	-0.0906
R	0.582912	-0.290985	0.308543	0.202187	0.162831	0.110100	1.000000	0.510762	0.808361	0.655176	...	-0.086385	0.243771	0.374562	0.4544
AB	0.411887	0.338459	0.091223	0.052385	0.053000	0.034746	0.510762	1.000000	0.815225	0.525947	...	0.143260	0.300335	0.967633	0.7010
H	0.501754	0.030939	0.201641	0.120521	0.116135	0.088074	0.808361	0.815225	1.000000	0.692192	...	0.013288	0.273602	0.680635	0.6385
2B	0.330347	-0.006686	0.112982	0.168941	0.053798	0.048713	0.655176	0.525947	0.692192	1.000000	...	-0.043044	0.152670	0.402041	0.4544
3B	0.058798	0.051323	0.036179	0.019339	0.037891	0.034793	0.039864	0.142016	0.152699	0.041459	...	0.064810	0.058087	0.142943	0.0666
HR	0.379475	-0.214472	0.221195	0.158893	0.102427	0.055517	0.753188	0.307530	0.469499	0.458746	...	-0.142948	0.139213	0.203495	0.3232
BB	0.492629	-0.220231	0.255003	0.185609	0.131853	0.080957	0.627639	0.370549	0.409526	0.313116	...	0.022756	0.233556	0.371707	0.2503
SO	0.084654	0.264296	-0.006062	0.085035	-0.053807	-0.058187	0.195971	0.426403	0.186067	0.353040	...	-0.006942	0.090461	0.435803	0.3436
SB	0.165103	-0.065174	0.078533	-0.068462	0.072640	0.028146	-0.033737	0.088560	0.020890	-0.180286	...	0.127280	0.177574	0.161984	-0.0663
CS	0.006205	0.084064	-0.043449	-0.078715	0.021012	0.009903	-0.155463	0.048963	-0.036883	-0.303612	...	0.022980	0.071232	0.139914	-0.0170
HBP	0.144878	0.014690	0.072785	0.119975	0.028260	0.031098	0.374504	0.243170	0.289446	0.502490	...	-0.084185	0.102404	0.186133	0.2797
SF	0.358583	-0.167736	0.195684	0.099032	0.064267	0.056822	0.450140	0.292902	0.453236	0.287313	...	0.087340	0.212791	0.257387	0.1961
RA	-0.360718	0.638532	-0.273232	-0.066782	-0.187599	-0.121338	0.408075	0.430382	0.468330	0.389674	...	-0.552667	-0.279144	0.280258	0.8501
ER	-0.334843	0.606295	-0.255780	-0.051526	-0.180797	-0.112826	0.432861	0.427191	0.482777	0.421101	...	-0.545333	-0.259921	0.272797	0.8439
ERA	-0.552889	0.471308	-0.312633	-0.073171	-0.214883	-0.134111	0.262595	-0.019148	0.169725	0.237525	...	-0.667103	-0.437138	-0.190536	0.5802
CG	0.101213	-0.079924	0.043832	-0.109609	0.072538	0.044264	-0.125715	-0.011223	-0.082923	-0.344077	...	0.256191	-0.140880	0.049230	-0.1538
SHO	0.484777	-0.330802	0.296759	0.046959	0.195257	0.151786	-0.086385	0.143260	0.013288	-0.043044	...	1.000000	0.308667	0.244667	-0.3297
SV	0.687579	-0.440848	0.385054	0.107683	0.240168	0.149955	0.243771	0.300335	0.273602	0.152670	...	0.308667	1.000000	0.369150	-0.0426
IPouts	0.451171	0.312185	0.115989	0.045902	0.068610	0.042283	0.374562	0.967633	0.680635	0.402041	...	0.244667	0.369150	1.000000	0.5962
HA	-0.084732	0.587115	-0.130126	-0.037854	-0.118221	-0.090614	0.454480	0.701022	0.638599	0.454493	...	-0.329741	-0.042668	0.596211	1.0000
HRA	-0.154842	0.356043	-0.129637	0.001851	-0.116112	-0.076633	0.477781	0.332728	0.425406	0.419049	...	-0.442013	-0.152591	0.198567	0.6134
BBA	-0.199850	0.486949	-0.212610	-0.052623	-0.103021	-0.034211	0.219414	0.392797	0.310606	0.181648	...	-0.278316	-0.155606	0.335077	0.4239
SOA	0.418006	-0.071258	0.214272	0.160927	0.108110	0.086503	0.280061	0.450999	0.359692	0.443135	...	0.277807	0.276517	0.464603	0.0630
E	-0.173751	0.378242	-0.148163	-0.138290	-0.090254	-0.079711	-0.098035	0.226610	0.046945	-0.163969	...	-0.103493	-0.086181	0.256558	0.2497
DP	0.022857	0.300158	-0.090399	0.008806	-0.056016	-0.046240	0.285082	0.441310	0.394881	0.243803	...	-0.101042	-0.008994	0.393767	0.5463
FP	0.331059	-0.276190	0.184339	0.160951	0.111149	0.093103	0.220742	0.100353	0.180594	0.294092	...	0.191568	0.210195	0.084153	-0.0467

32 rows × 32 columns

3. Missing value treatment and outlier removal.
    a. Data selection and segmentation

Although the Teams.csv data file provides us 143 year-long team performance data (1871 to 2014), I will have my focus on modern baseball statistics ranging from 1985 to 2012. The reasons are as following:

i. Recent baseball statistics are more representative of modern baseball competition.
ii. The salary data will also be part of the data set, and the earliest team salary data recorded in the Salaries.csv is 1985 year - data.

Therefore, the data set is streamlined to 798 records.

By examining the data columns, I have found that some of them are less relevant to the study subject. Columns such as 'park', 'BPF', 'PPF', 'teamIDBR', 'teamIDlahman45', and 'teamIDretro' are either name or ID related which I believe could be removed from the data set.

As a result, the streamlined Teams data set contains 798 records with 42 columns.

In [3]:
```
# Extract 1985-2012 data

# Make a data copy
team_df = original_teams_df.copy(deep=True)
team_df = team_df.loc[(team_df['yearID']>1984) & (team_df['yearID']<2013)]
# print(team_df.shape)

# Get necessary columns
unnecessary_columns = ['park', 'BPF', 'PPF', 'teamIDBR', 'teamIDlahman45', 'teamIDretro']
team_df.drop(columns=unnecessary_columns, axis=1, inplace=True)
print(team_df.shape)
```

```
(798, 42)
```

As mentioned, the team payroll data is part of the data set. Therefore I merge team statistics with each team's 22-men roster payroll data.

In [16]:
```
# Payroll data
payroll_data_path = 'C:\\Users\\mikee\\Downloads\\Coursera\\IBM Machine Learning\\EDA\\baseballdatabank-master\\core\\Salaries.cs

payroll_df_backup = pd.read_csv(payroll_data_path)
payroll_df = payroll_df_backup.copy(deep=True)

# payroll_df.info()
payroll_df = payroll_df.loc[payroll_df['yearID']<=2012]

payroll_df = payroll_df.groupby(by=['yearID', 'teamID']).sum()
payroll_df = payroll_df.reset_index()

# merge payroll data with team data
team_df = team_df.merge(payroll_df, on=['yearID', 'teamID'])
team_df.head()
```

Out[16]:

	yearID	lgID	teamID	franchID	divID	Rank	G	Ghome	W	L	...	HA	HRA	BBA	SOA	E	DP	FP	name	attendance	salary
0	1985	NL	ATL	ATL	W	5	162	81.0	66	96	...	1512	134	642	776	159	197	0.976	Atlanta Braves	1350137.0	14807000
1	1985	AL	BAL	BAL	E	4	161	81.0	83	78	...	1480	160	568	793	129	168	0.979	Baltimore Orioles	2132387.0	11560712
2	1985	AL	BOS	BOS	E	5	163	81.0	81	81	...	1487	130	540	913	145	161	0.977	Boston Red Sox	1786633.0	10897560
3	1985	AL	CAL	ANA	W	2	162	79.0	90	72	...	1453	171	514	767	112	202	0.982	California Angels	2567427.0	14427894
4	1985	AL	CHA	CHW	W	3	163	81.0	85	77	...	1411	161	569	1023	111	152	0.982	Chicago White Sox	1669888.0	9846178

5 rows × 43 columns

The dimension of the selected data set is 798 records with 43 columns.

In [17]:
```
team_df.shape
```

Out[17]:
```
(798, 43)
```

In [21]:
```
team_df.columns
```

Out[21]:
```
Index(['yearID', 'lgID', 'teamID', 'franchID', 'divID', 'Rank', 'G', 'Ghome',
       'W', 'L', 'DivWin', 'WCWin', 'LgWin', 'WSWin', 'R', 'AB', 'H', '2B',
       '3B', 'HR', 'BB', 'SO', 'SB', 'CS', 'HBP', 'SF', 'RA', 'ER', 'ERA',
       'CG', 'SHO', 'SV', 'IPouts', 'HA', 'HRA', 'BBA', 'SOA', 'E', 'DP', 'FP',
       'name', 'attendance', 'salary'],
      dtype='object')
```

b. Missing data treatment

Basically, the 2 data files, Teams.csv and Salaries.csv, are relatively "clean" in terms of missing value. However, they are not that perfect and ready to be analyzed.

The missing value is due to the strike in 1994. Since the regular season was interrupted due to the strike, there were no division, league and world series winners respectively for the entire MLB teams.

The outcome is obvious and straightforward: there are no values from DivWin (division winner), WCWin (wild card winner), LgWin (league winner), and WSWin (world series winner) columns. Since these are categorical type columns, I simply update these missing values into ‘N’ for the sake of consistency.

Another missing part is the WCWin missing prior to 1994. This is because the wild card was first instituted in MLB in 1994, so it is reasonable that there would be no such value from 1985 to 1993. And even though the wild card rule was set in 1994, the season was unfortunately interrupted due to the strike which led to no wild card value in that year too.

In conclusion, for missing DivWin, LgWin, and WSWin values in 1994, I reset them to ‘N’. For missing WCWin values from 1985 to 1994, I also reset them to N.

In [22]:
```
# Reset the values.
# Reset missing DivWin values in 1994
team_df.loc[team_df['yearID']==1994, 'DivWin'] = 'N'
# Reset missing WSWin values in and before 1994
team_df.loc[team_df['yearID']<=1994, 'WCWin'] = 'N'
# Reset missing LgWin values in 1994
team_df.loc[team_df['yearID']==1994, 'LgWin'] = 'N'
# Reset missing WSWin values in 1994
team_df.loc[team_df['yearID']==1994, 'WSWin'] = 'N'
team_df.loc[team_df['yearID']==1994][['DivWin', 'WCWin', 'LgWin', 'WSWin']]
```

Out[22]:

	DivWin	WCWin	LgWin	WSWin
236	N	N	N	N
237	N	N	N	N
238	N	N	N	N
239	N	N	N	N
240	N	N	N	N
241	N	N	N	N
242	N	N	N	N
243	N	N	N	N
244	N	N	N	N
245	N	N	N	N
246	N	N	N	N
247	N	N	N	N

Let’s have a final check on missing values:

```
In [11]: print(check_missing_value(team_df))
         # team_df.reset_index(inplace=True, drop=True)
         # team_df
```

```
yearID        0
lgID          0
teamID        0
franchID      0
divID         0
Rank          0
G             0
Ghome         0
W             0
L             0
DivWin        0
WCWin         0
LgWin         0
WSWin         0
R             0
AB            0
H             0
2B            0
3B            0
HR            0
BB            0
SO            0
SB            0
CS            0
HBP           0
SF            0
RA            0
ER            0
ERA           0
CG            0
SHO           0
SV            0
IPouts        0
HA            0
HRA           0
BBA           0
SOA           0
E             0
DP            0
FP            0
name          0
attendance    0
salary        0
```

Now the data set does not contain any missing value.

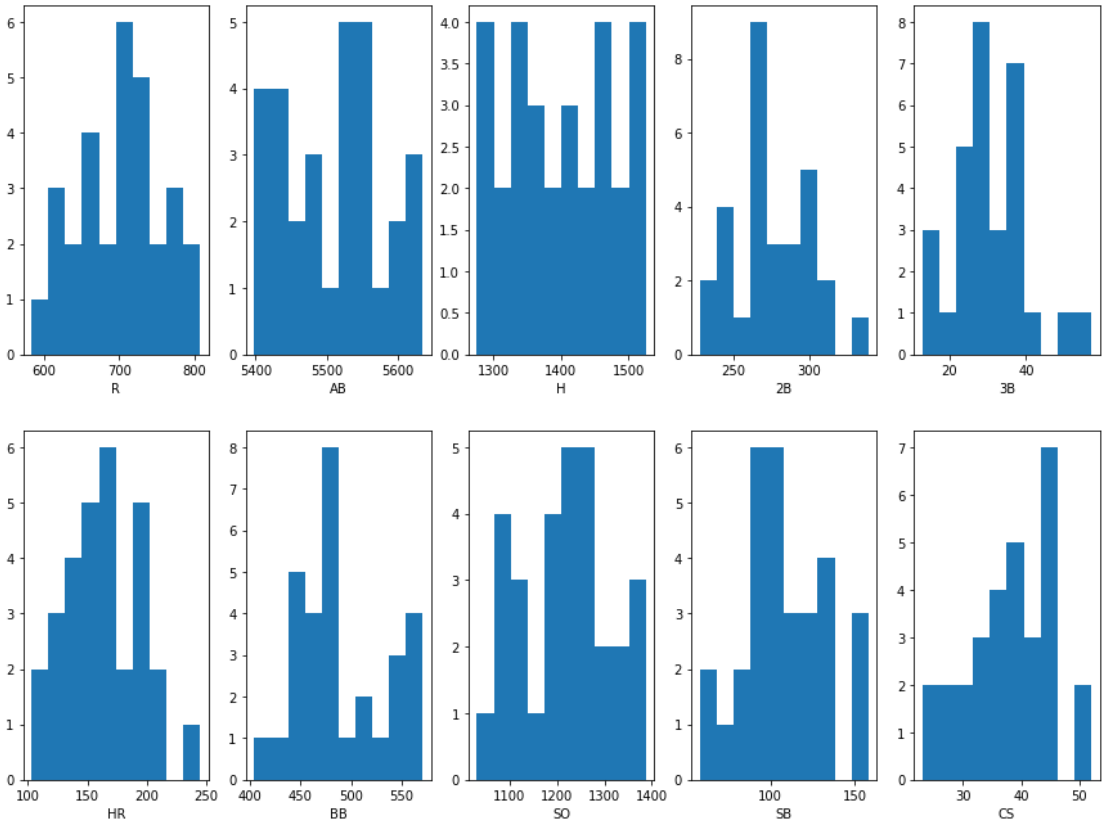
c. Outlier

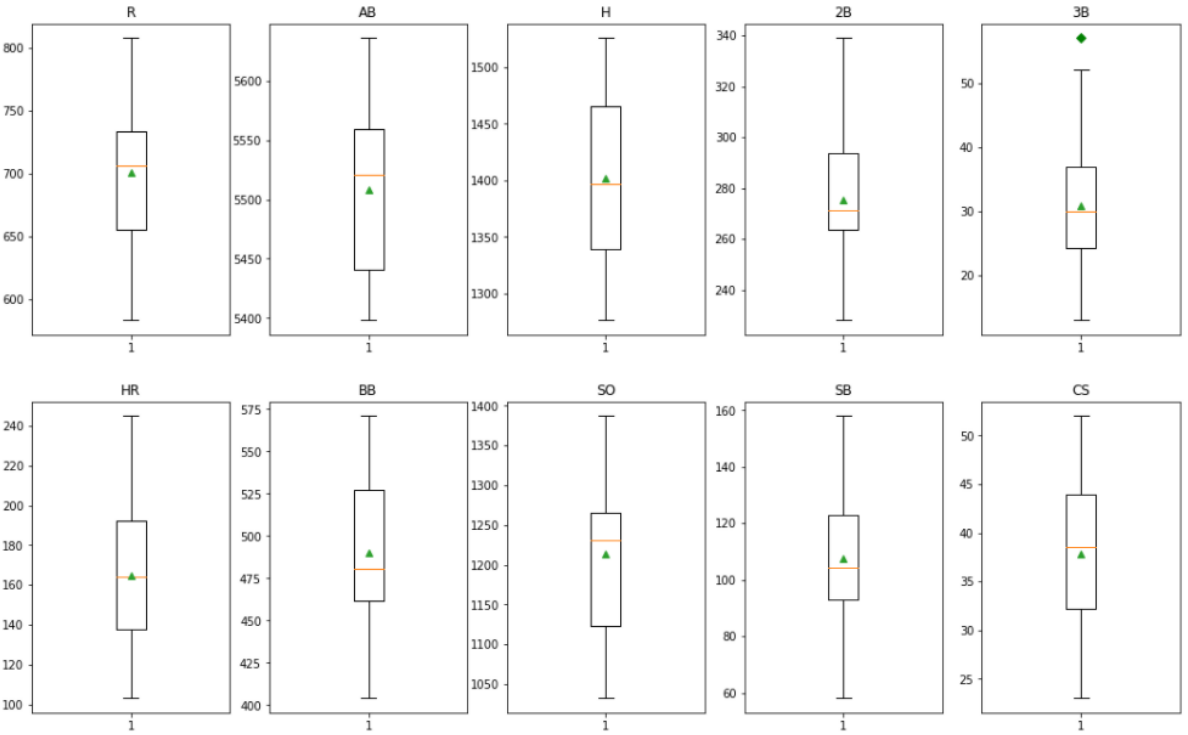
After checking the missing data, it comes to identify the potential outliers in the data set. Firstly, I pick up all numeric columns and get rid of non - numeric ones. Then, for some numeric columns, such as ID related and attendance, are less statistically meaningful, therefore these columns are excluded from the checking list as well.

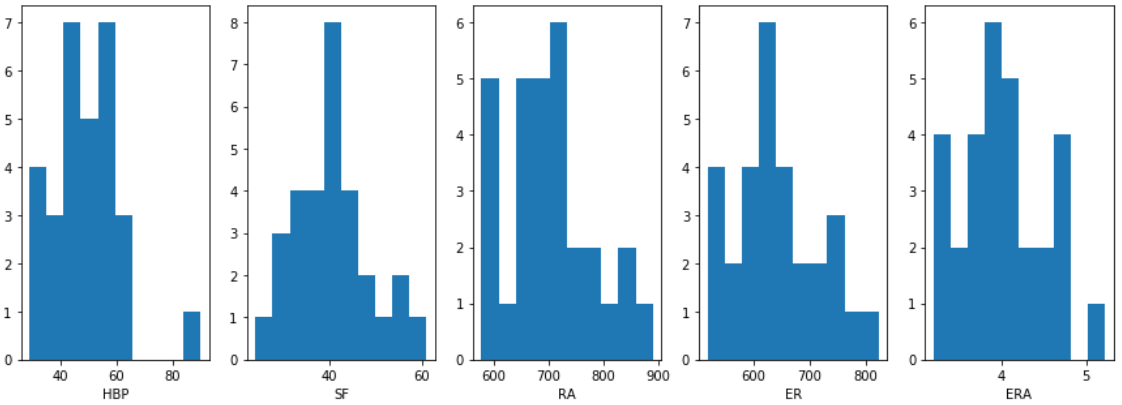
As a result, the candidate columns to be verified are: 'R', 'AB', 'H', '2B', '3B', 'HR', 'BB', 'SO', 'SB', 'CS', 'HBP', 'SF', 'RA', 'ER', 'ERA', 'CG', 'SHO', 'SV', 'IPouts', 'HA', 'HRA', 'BBA', 'SOA', 'E', 'DP', 'FP', 'salary', 'OBP', 'SLG', and 'OPS' respectively. Even though some of them may not be utilized for modeling and prediction, it is no harm to have a check to each of them.

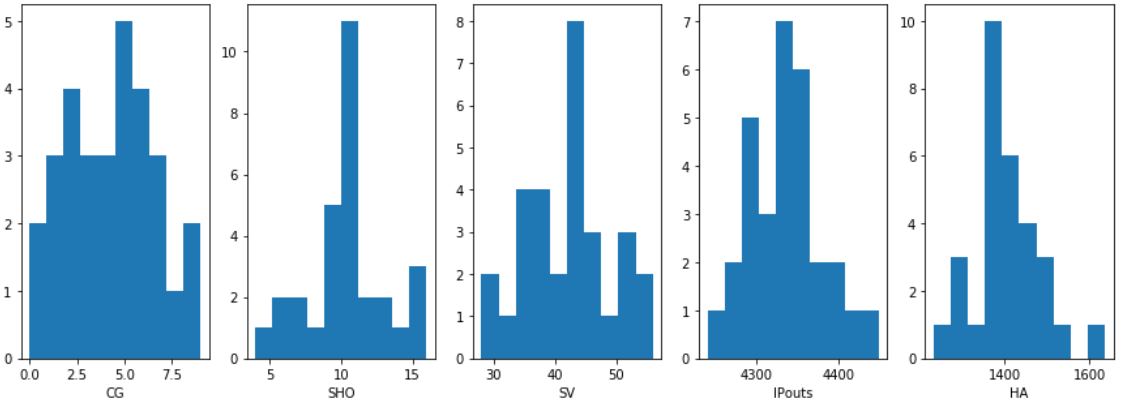
In addition, since the data set covers MLB team statistics ranging from 1985 to 2012, it is more reasonable to verify outliers separately on a yearly manner instead of checking them on a mixed, multi - year data set to prevent misleading.

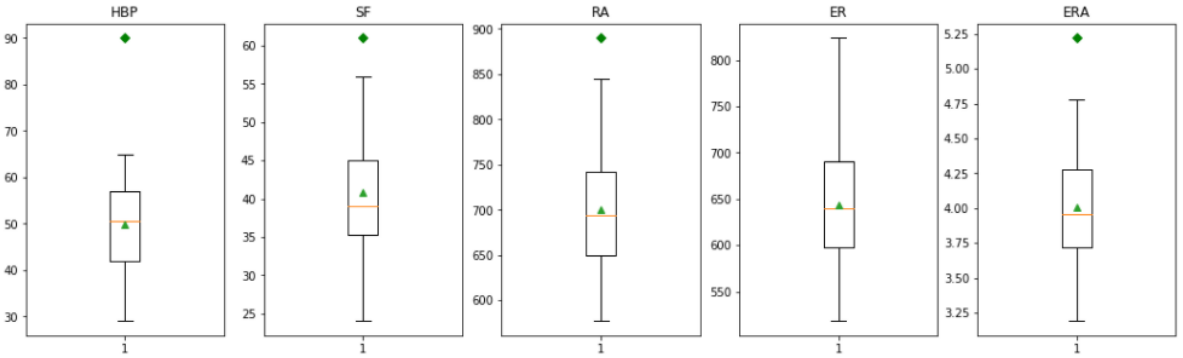
I will use visualization tools such as histogram and boxplot to identify potential outliers. Here is some samples of visualization on 2012 year data:

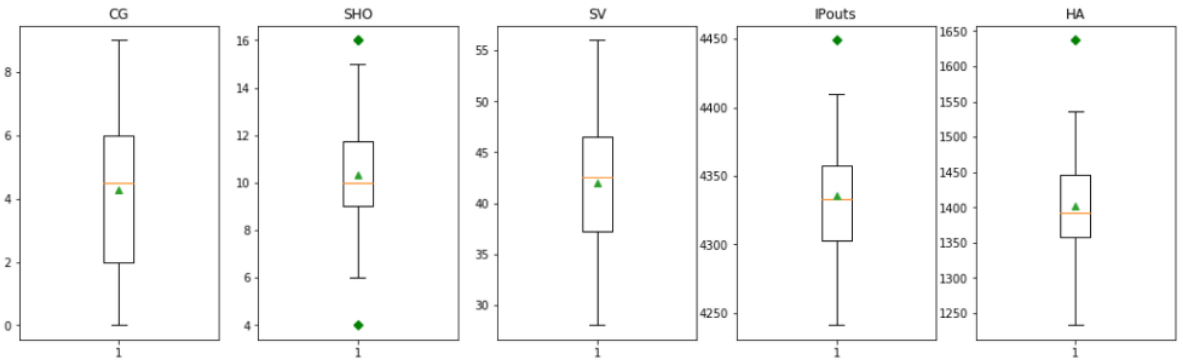





HBP
SF
RA
ER
ERA


CG
SHO
SV
IPouts
HA


HBP
SF
RA
ER
ERA


CG
SHO
SV
IPouts
HA

There is an interesting fact regarding the outlier verification. There are some outliers within each yearly data, however, if we compare the results from multiple yearly data, we can see the features that contain outliers are changing among the years. For example, in 1989 data there are outliers in the ‘H’ feature and no outliers in ‘SB’ feature. However it is just the opposite in 1990 data since the outlier appears in ‘SB’ but not in ‘H’.

I believe it is the nature of baseball game statistics. Let’s say, some teams were focusing on producing short - range hits so that they had outstanding ‘H’ figures, and in 1990 some teams had recruited players with astonishing speed so that the ‘SB’ (Stolen Base) figures were sky - high. Therefore, the outliers on baseball statistics may be regarded as the combination of talent players, tactics and other factors and it is a part of the game. I would rather keep these figures instead of removing them.

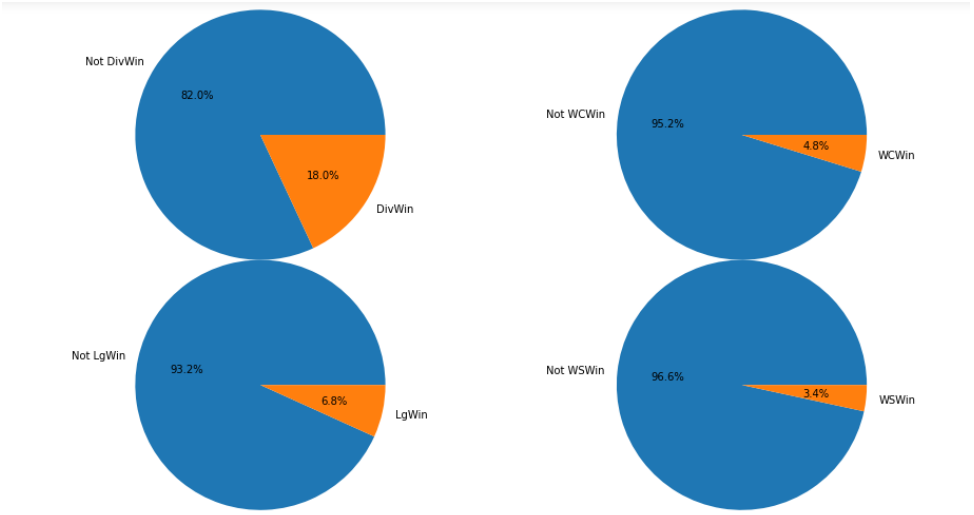
4. Single variable analysis
   Single variable analysis is to analyze if the variables are with continuous or categorical features and see the characteristics of each feature.

   According to the outlier verification results described above, it is obvious that except some categorical variables most variables in the data set are continuous ones.

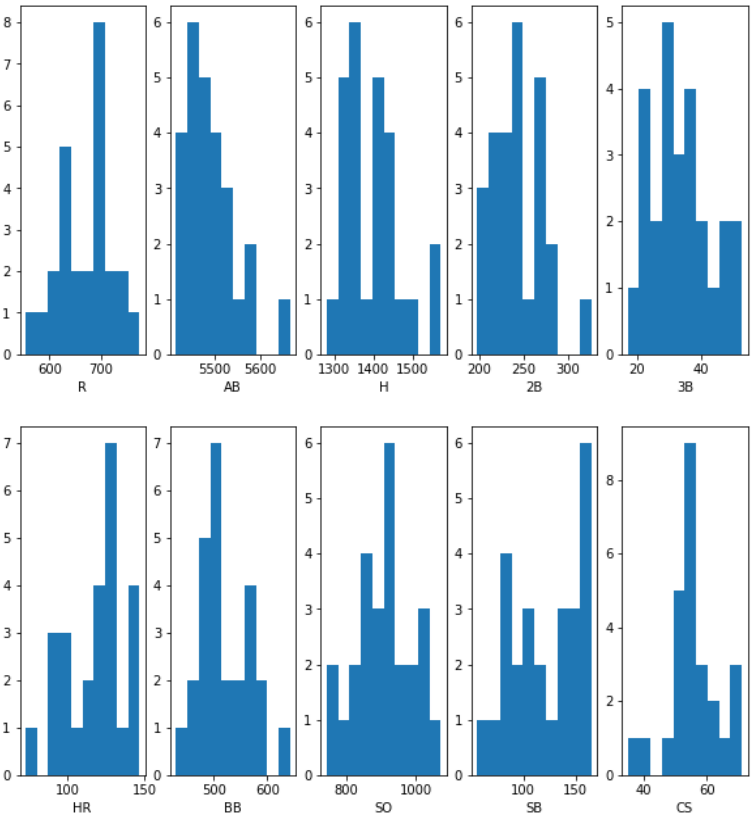
   A simple method to examine continuous and categorical variable attributes is to visualize them through proper plots:

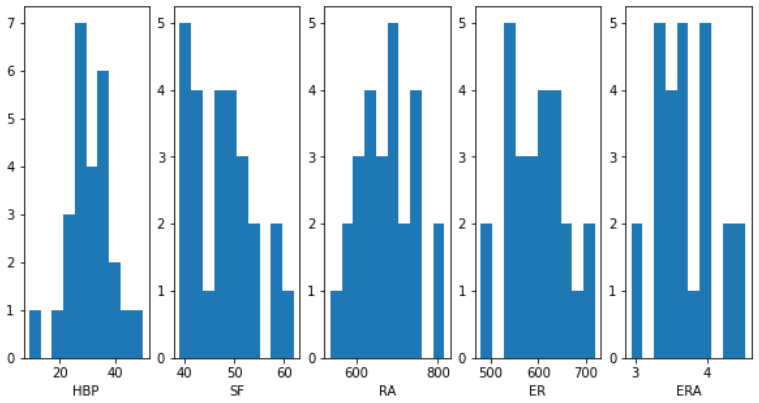
Type	Plot
Continuous variable	histogram, boxplot
Categorical variable	bar chart, pie chart

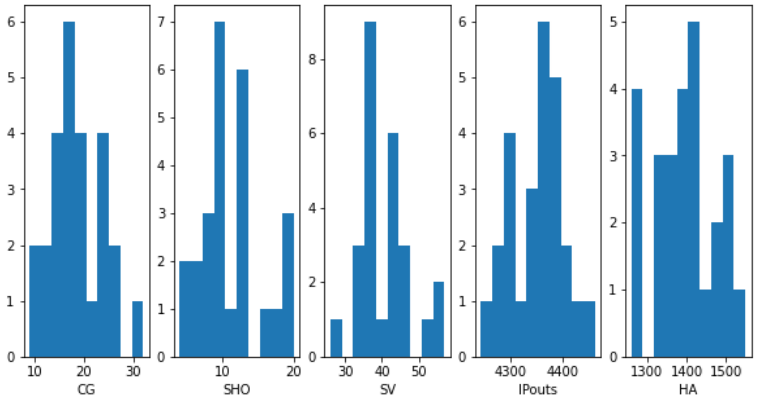
Visualize categorical samples ('DivWin', 'WCWin', 'LgWin', 'WSWin'):

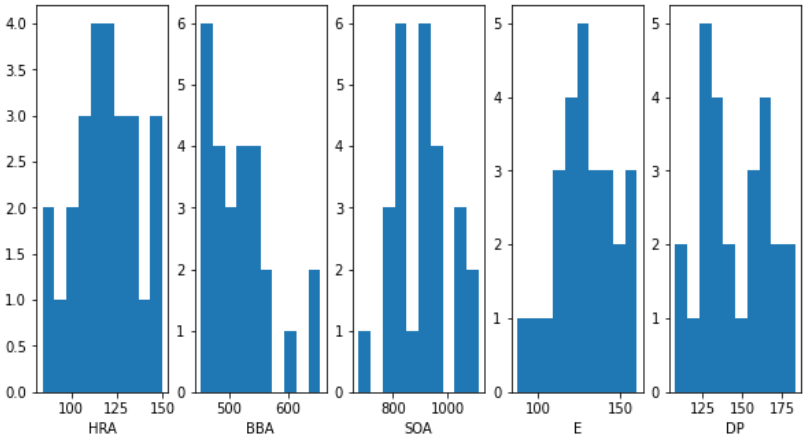


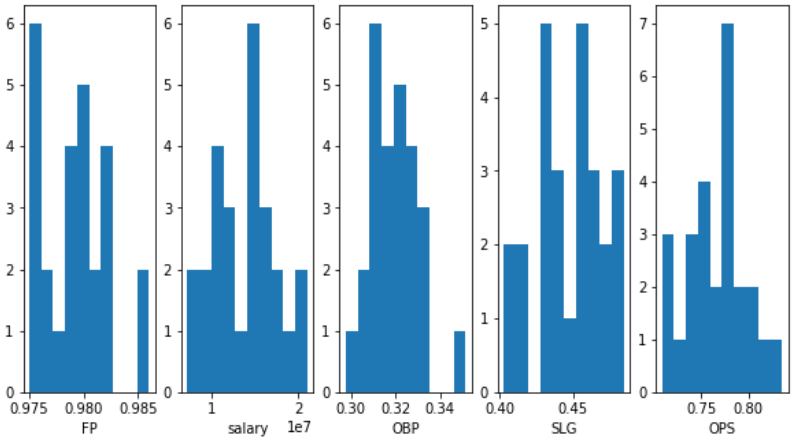
Visualize continuous examples:




HBP
SF
RA
ER
ERA


CG
SHO
SV
IPouts
HA


HRA
BBA
SOA
E
DP


FP
salary
1e7
OBP
SLG
OPS

5. Feature engineering.
    a. Transform categorical data

Inside the data set, there are four categorical columns that may have effects on future analysis: DivWin, WCWin, LgWin, and WSWin. These columns are indicators of postseason winnings to a team and are represented in value Y (winner) and N.

For the sake of simplicity, I will transform value Y into numeric value 1 and value N to so that I can directly utilize these features for modeling.

In [62]:
```
# Transform categorical columns to numeric features.

team_df['DivWin'].replace(to_replace={'N':0, 'Y':1}, inplace=True)
team_df['WCWin'].replace(to_replace={'N':0, 'Y':1}, inplace=True)
team_df['LgWin'].replace(to_replace={'N':0, 'Y':1}, inplace=True)
team_df['WSWin'].replace(to_replace={'N':0, 'Y':1}, inplace=True)
team_df[['yearID', 'name', 'DivWin', 'WCWin', 'LgWin', 'WSWin']]
```

Out[62]:

	yearID	name	DivWin	WCWin	LgWin	WSWin
0	1985	Atlanta Braves	0	0	0	0
1	1985	Baltimore Orioles	0	0	0	0
2	1985	Boston Red Sox	0	0	0	0
3	1985	California Angels	0	0	0	0
4	1985	Chicago White Sox	0	0	0	0
...	...	...	...	...	...	...
793	2012	St. Louis Cardinals	0	1	0	0
794	2012	Tampa Bay Rays	0	0	0	0
795	2012	Texas Rangers	0	1	0	0
796	2012	Toronto Blue Jays	0	0	0	0
797	2012	Washington Nationals	1	0	0	0

798 rows × 6 columns

    b. Transform text (object) value to dummies

There are also four text columns in the data set, which are teamID, franchID, lgID and divID. For the teamID and franchID I believe they are for the team identification usage only and have little effect on modeling. Therefore I tend not to do anything to them at this moment.

However for lgID and divID, since there would be a scenario to analyze teams' performance within and across the leagues and divisions, so I decide to transform these columns into dummy features:

- lgID_AL (American League)
- lgID_NL (National League)
- divID_C (Central Division)
- divID_E (Eastern Division)
- divID_W (Western Division)

```
lgID_AL          uint8
lgID_NL          uint8
divID_C          uint8
divID_E          uint8
divID_W          uint8
```

Here is the example:

	teamID	franchID	name	lgID_AL	lgID_NL	divID_C	divID_E	divID_W
0	ATL	ATL	Atlanta Braves	0	1	0	0	1
1	BAL	BAL	Baltimore Orioles	1	0	0	1	0
2	BOS	BOS	Boston Red Sox	1	0	0	1	0
3	CAL	ANA	California Angels	1	0	0	0	1
4	CHA	CHW	Chicago White Sox	1	0	0	0	1
...	...	...	...	...	...	...	...	...
793	SLN	STL	St. Louis Cardinals	0	1	1	0	0
794	TBA	TBD	Tampa Bay Rays	1	0	0	1	0
795	TEX	TEX	Texas Rangers	1	0	0	0	1
796	TOR	TOR	Toronto Blue Jays	1	0	0	1	0
797	WAS	WSN	Washington Nationals	0	1	0	1	0

798 rows × 8 columns

c. Create domain data

Based on modern baseball game statistics and observation, several interesting indicators have been introduced as key indexes to evaluate offensive performance. The most widely adapted one is On-base Plus Slugging (OPS), which is the summary of On - Base Percentage (OBP) and Slugger (SLG).

The formula is described as below:

- OBP (On-Base Percentage) =（H + BB + HBP）/（AB + BB + HBP + SF）
- SLG (Slugging Percentage) = ((1*H)+(2*2B)+(3*3B)+(4*HR))/AB
- OPS = OBP + SLG

To provide more comprehensive review of teams' performance, I will add these 3 features into the data set:

name	attendance	salary	OBP	SLG	OPS
Atlanta Braves	1350137.0	14807000	0.314881	0.429425	0.744306
Baltimore Orioles	2132387.0	11560712	0.335599	0.514954	0.850552
Boston Red Sox	1786633.0	10897560	0.346522	0.513986	0.860508
California Angels	2567427.0	14427894	0.332738	0.459206	0.791945
Chicago White Sox	1669888.0	9846178	0.315143	0.470750	0.785893
...	...	...	...	...	...
St. Louis Cardinals	3262109.0	110300862	0.337542	0.507471	0.845013
Tampa Bay Rays	1559681.0	64173500	0.316691	0.478511	0.795202
Texas Rangers	3460280.0	120510974	0.333603	0.541682	0.875285
Toronto Blue Jays	2099663.0	75009200	0.309241	0.491708	0.800949
Washington Nationals	2370794.0	80855143	0.322152	0.520214	0.842366

d. Data normalization

To better fit the data set into modeling and prediction, The major numeric columns are to be normalized. I will use standard normalization to transform the columns as they are expected to be governed by the normal distribution (as illustrated in “Outlier” section).

Here is the sample result of normalized data set:

	yearID	teamID	franchID	name	lgID_AL	lgID_NL	divID_C	divID_E	divID_W	R	...	HRA	BBA	SOA	E	DP
0	1985	ATL	ATL	Atlanta Braves	0	1	0	0	1	-1.146699	...	-0.759223	1.552650	-1.677700	2.389722	2.459523
1	1985	BAL	BAL	Baltimore Orioles	1	0	0	1	0	0.894389	...	0.078668	0.501904	-1.557607	0.886294	0.998956
2	1985	BOS	BOS	Boston Red Sox	1	0	0	1	0	0.696864	...	-0.888129	0.104324	-0.709896	1.688122	0.646406
3	1985	CAL	ANA	California Angels	1	0	0	0	1	-0.049340	...	0.433161	-0.264857	-1.741278	0.034352	2.711345
4	1985	CHA	CHW	Chicago White Sox	1	0	0	0	1	-0.005446	...	0.110895	0.516103	0.067172	-0.015763	0.193127
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
793	2012	SLN	STL	St. Louis Cardinals	0	1	1	0	0	0.312789	...	-0.759223	-1.372401	1.444703	-0.216220	0.042033
794	2012	TBA	TBD	Tampa Bay Rays	1	0	0	1	0	-0.433415	...	-0.598090	-0.903825	2.610306	0.134580	0.344220
795	2012	TEX	TEX	Texas Rangers	1	0	0	0	1	0.784653	...	0.562067	-1.230408	1.925073	-1.318734	-0.612703
796	2012	TOR	TOR	Toronto Blue Jays	1	0	0	1	0	-0.224917	...	1.496638	0.587099	0.907819	-0.516905	0.948592
797	2012	WAS	WSN	Washington Nationals	0	1	0	1	0	-0.060313	...	-0.920356	-0.506245	2.200579	-0.867705	-0.713432

798 rows × 39 columns

Key Findings and Insights, which synthesizes the results of Exploratory Data Analysis in an insightful and actionable manner

Now the data set is cleaned and without missing data. Through feature engineering I also supplement some performance indicators based on modern baseball theory. This provides us a better and simpler view of data analytics.

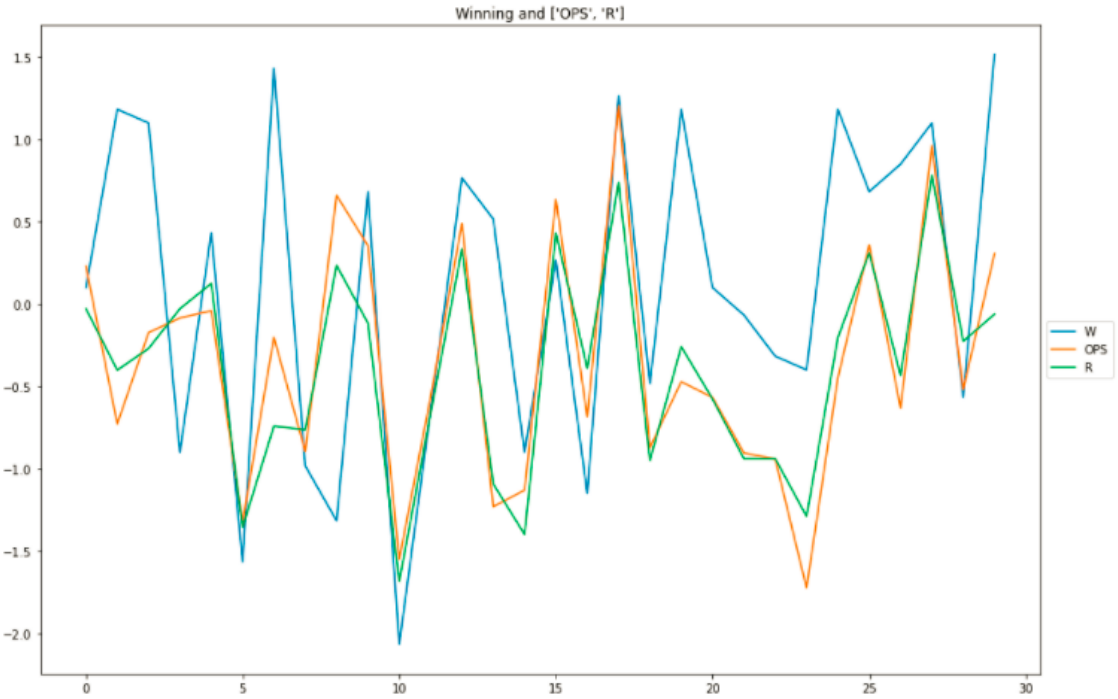
Here I start to cross - check the relationships between multiple features that I am interested in. As described earlier, my study will be focusing on the factors that contribute to a winning baseball team. Therefore the key factors definitely come from three perspectives: batting, pitching and defense.

The observation is based on 2012 teams' statistics as the first sample year.

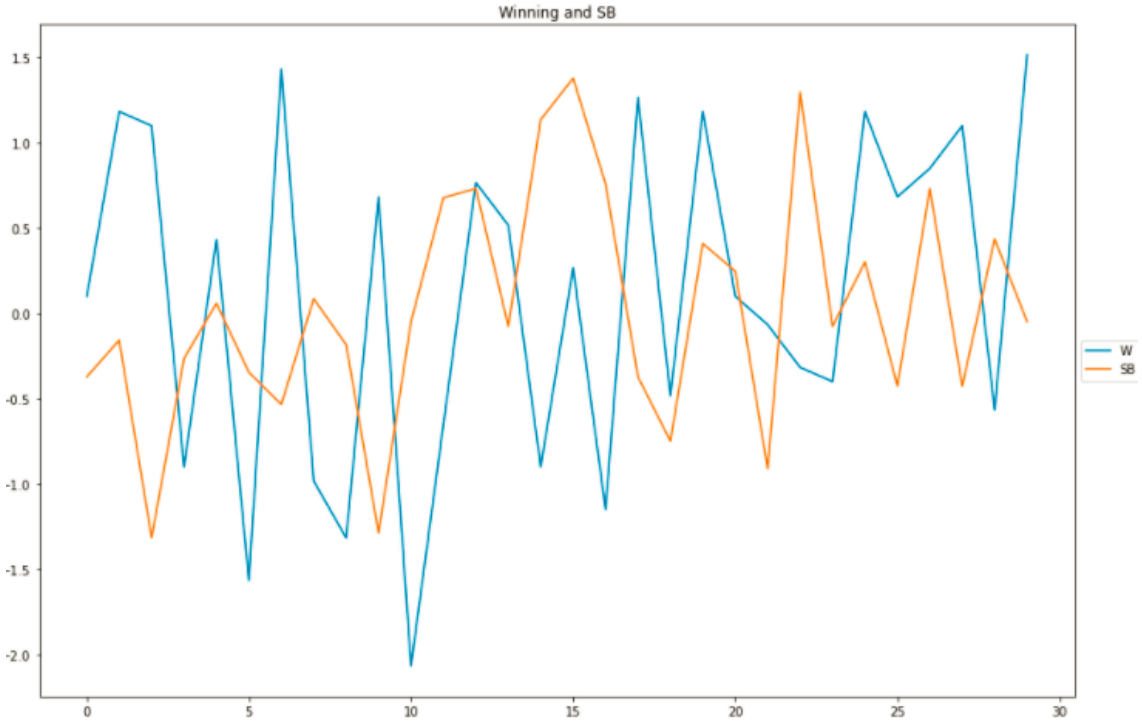
1. Offensive performance and winning

I will observe the relationships between features such as On-base Plus Slugging (OPS), and Stolen Base (SB) and team's winning.

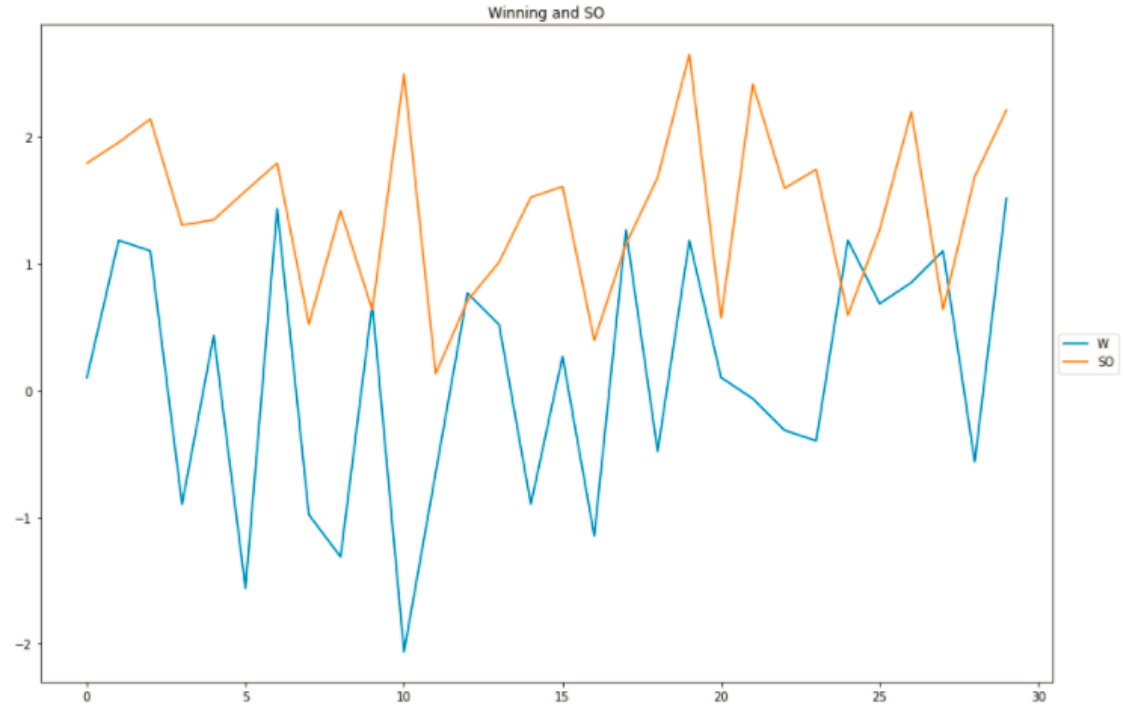
- Winning, OPS and R



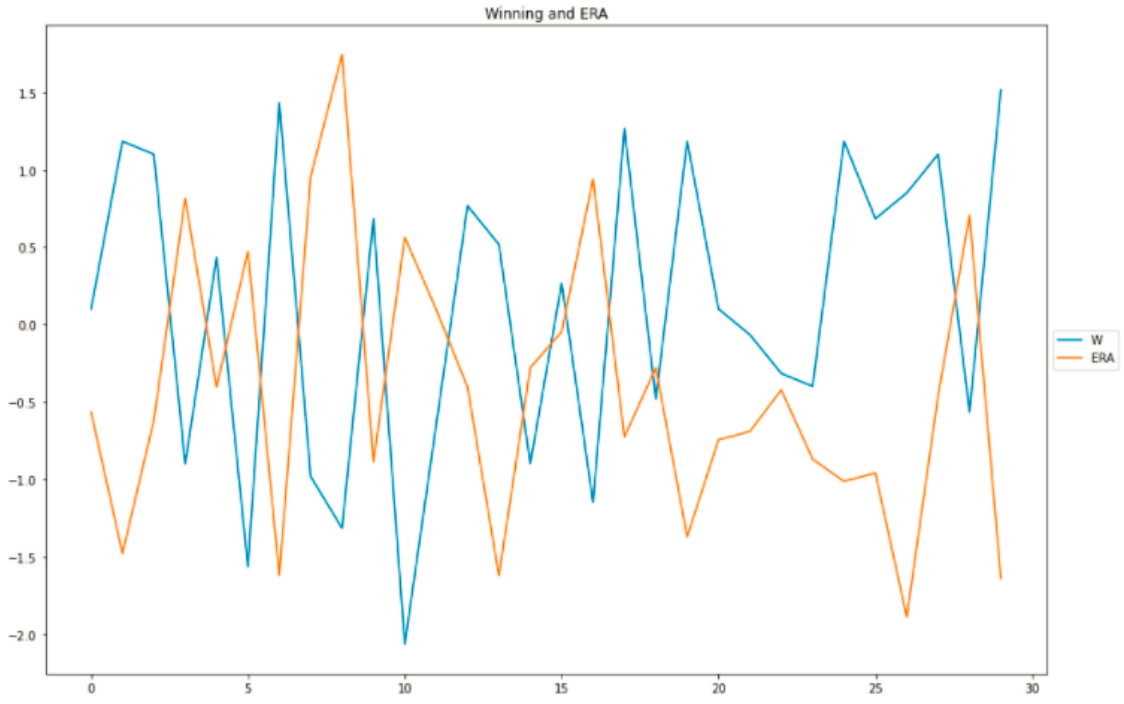
- Winning and SB



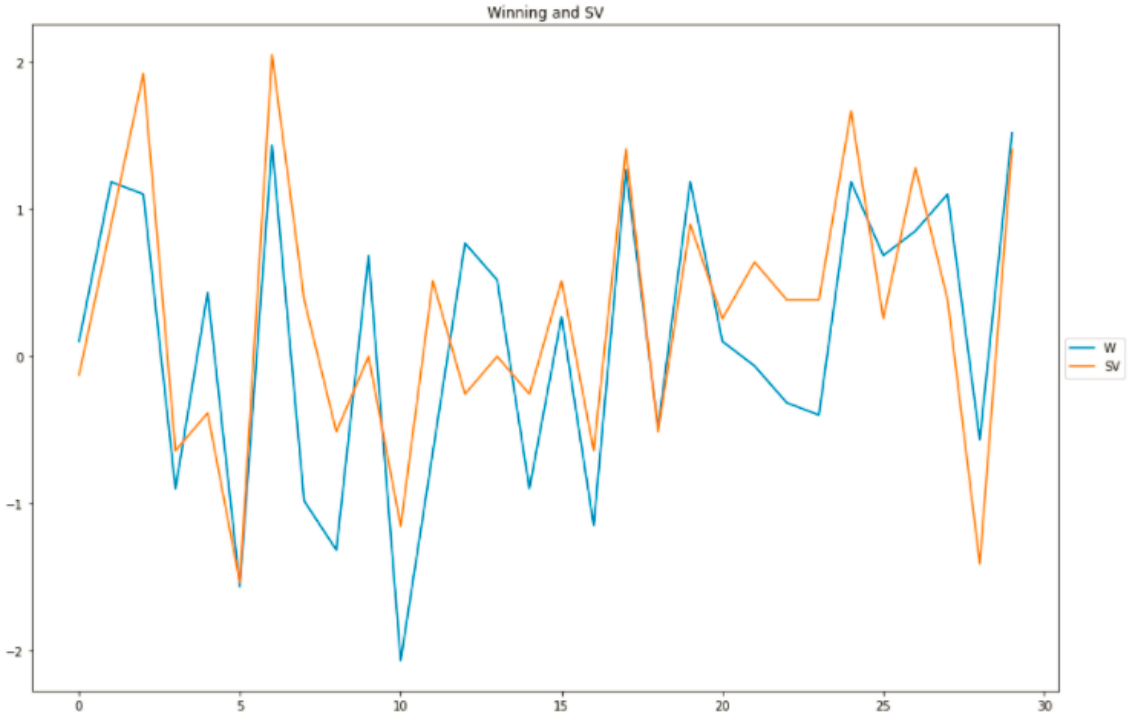
- Winning and SO (Strike out by batters)



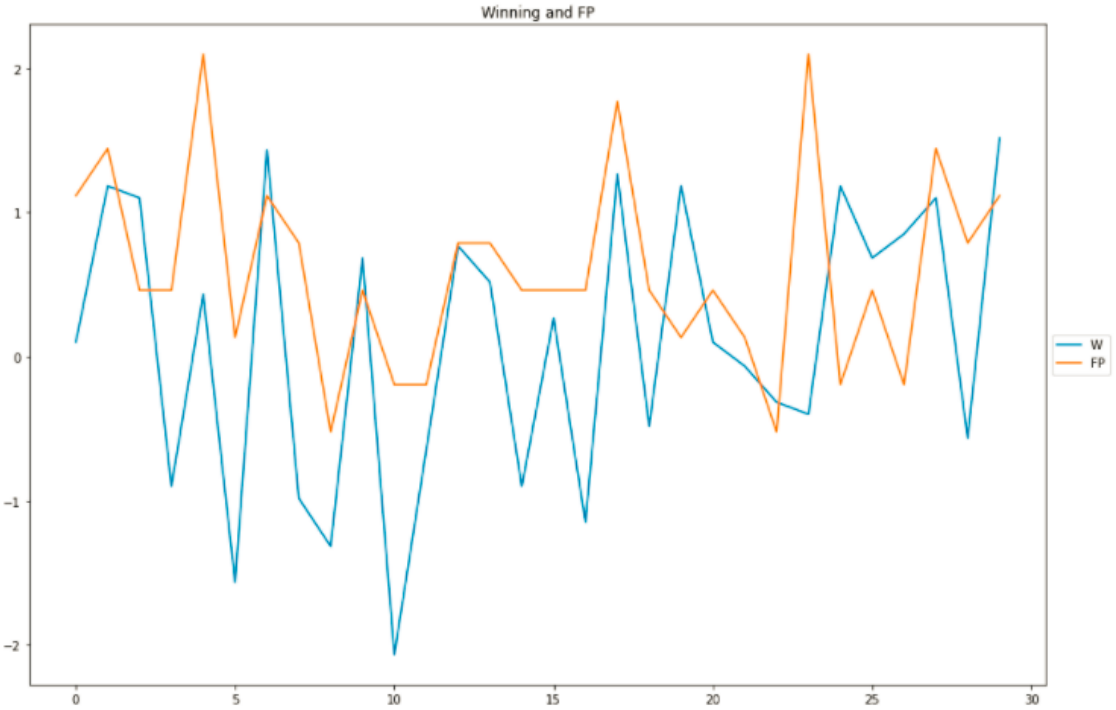
2. Pitching performance and winning
    - Winning and ERA



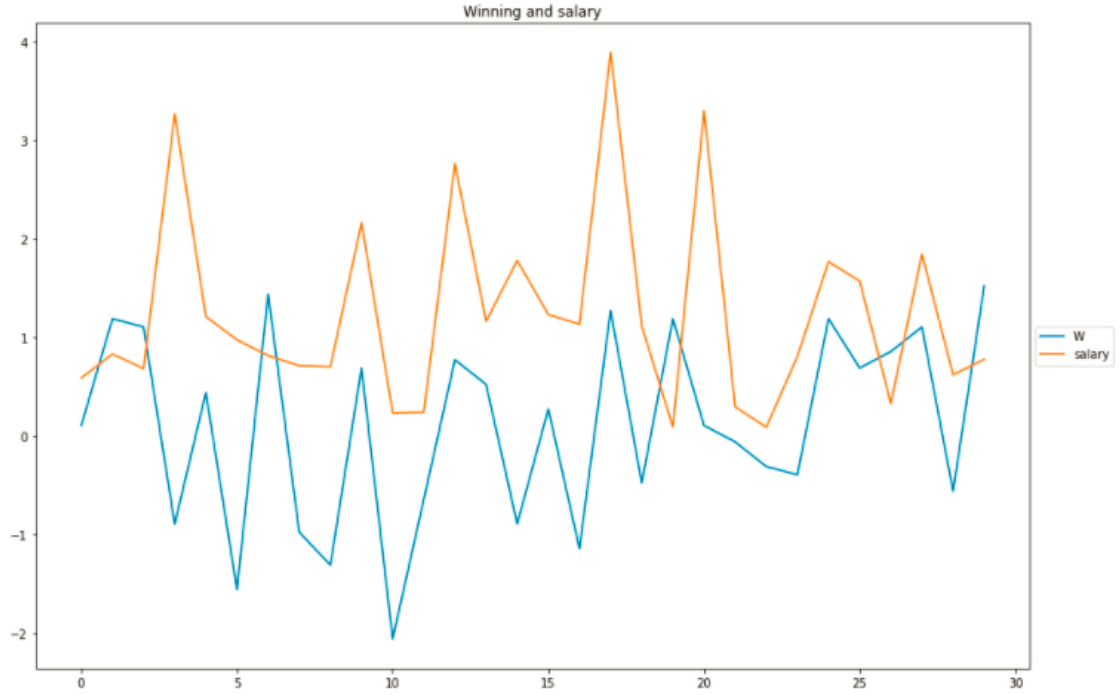
- Winning and SV



3. Defensive performance and winning
    - Winning and Field Percentage



4. Team salary and winning
    - Winning and salary



Through the observations above, I have found some interesting insights that can lead to further investigation and analysis.

1. Positive correlation between teams' success and offensive performance. Generally from the plot it is clear that the higher offensive performance,the more winnings.
2. Team's success also positively associates with pitching and defence.

3. However, it is not the case for the team's payroll. In other words, acquiring highly - paid players does not guarantee the team's winning.

Formulating at least 3 hypothesis about this data

Based on the purpose of data exploratory, I would like to see the "trend" of key indicators, including batting, pitching, defence and team payroll, and furthermore to see how these indicators affect the team's winning in the later analysis.

Therefore, in order to examine whether these indicators are different along with time, I come up with some hypotheses to see the facts from the statistics perspective.

1. Batting and offensive hypothesis test based on OPS.
    - H0: Average team batting performance (OPS) of a particular team before year 2001 is similar to that of after year 2001.
    - H1: Reject H0. Average team batting performance (OPS) of a particular team before year 2001 is significantly different from that of after year 2001.
2. Pitching hypothesis test based on ERA.
    - H0: Average team pitching performance (ERA) of a particular team before year 2001 is similar to that of after year 2001.
    - H1: Reject H0. Average team pitching performance (OPS) of a particular team before year 2001 is significantly different from that of after year 2001.
3. Team payroll hypothesis test based on salary.
    - H0: Average team payroll (salary) of a particular team before year 2001 is similar to that of after year 2001.
    - H1: Reject H0. Average team payroll (salary) of a particular team before year 2001 is significantly different from that of after year 2001.

I will divide the population into 2 samples by the year 2001 and set the significant level at 5% for 2 - tailed tests.

Conducting a formal significance test for one of the hypotheses and discuss the results

In the professional sports industry, payroll is always a critical factor of team management. Hence, I would like to see whether team payroll expenditure has been changing over time.

Therefore, for the significance test, I would like to verify the hypothesis that the average team payroll of a particular MLB team has NOT changed dramatically over time.

- H0: Average team payroll (salary) of a particular team before year 2001 is similar to that of after year 2001.
- H1: Reject H0. Average team payroll (salary) of a particular team before year 2001 is significantly different from that of after year 2001.

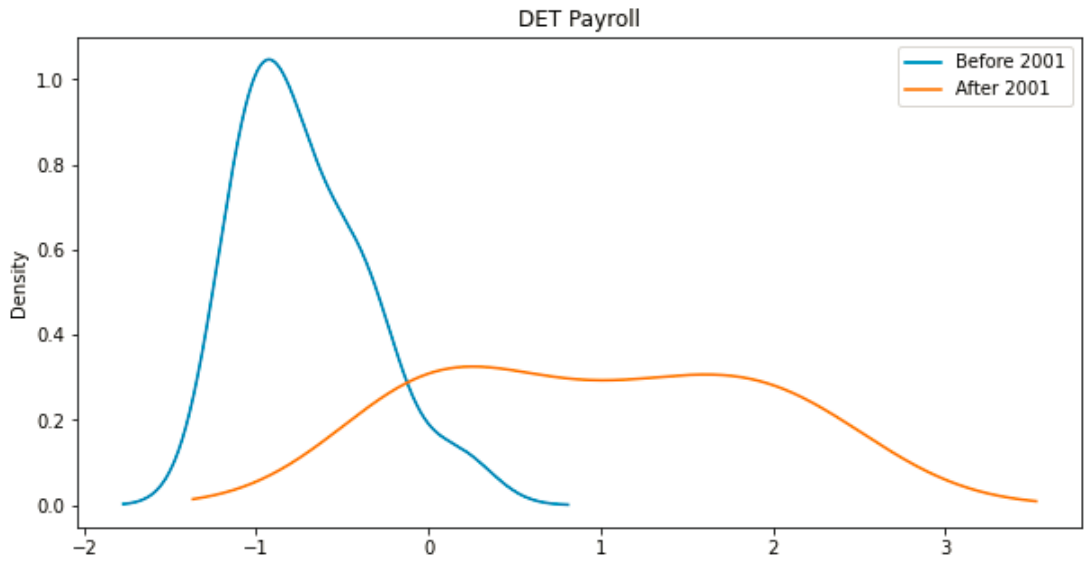
The significant level is set to 5% (α=0.05) for a two - tailed test.
The result is listed as following:

Team	p-value
ATL	7.18E-08
BAL	0.000219605
BOS	1.10E-11
ANA	4.69E-09
CHW	5.16E-08
CHC	6.46E-09
CIN	1.22E-06
CLE	0.001238561
DET	6.19E-07
HOU	1.20E-09
KCR	7.25E-06
LAD	2.04E-09
MIN	5.60E-07
MIL	1.47E-06
WSN	5.71E-10
NYY	2.29E-12
NYM	9.60E-11
OAK	2.87E-07
PHI	2.35E-07

PIT	1.72E-08
SDP	4.50E-05
SEA	5.66E-10
SFG	2.80E-10
STL	4.12E-11
TEX	2.06E-05
TOR	3.33E-06
COL	0.004206409
FLA	0.151941205
ARI	0.801993361
TBD	1.975554433

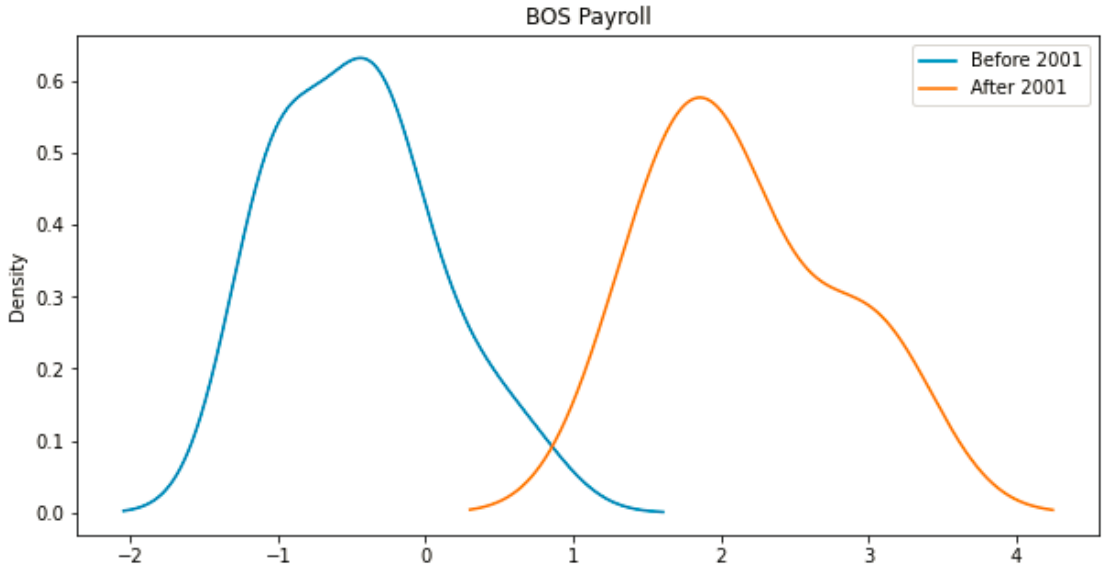
The observation is that except 3 teams, which are Miami Marlins, Arizona Diamondbacks, and Tampa Bay Devil Rays, all other teams have very small alpha values that lead to rejecting the null hypothesis. In other words, almost every team has a different scale in its payroll before and after 2001.

Let's take Detroit Tigers (DET) as an example:



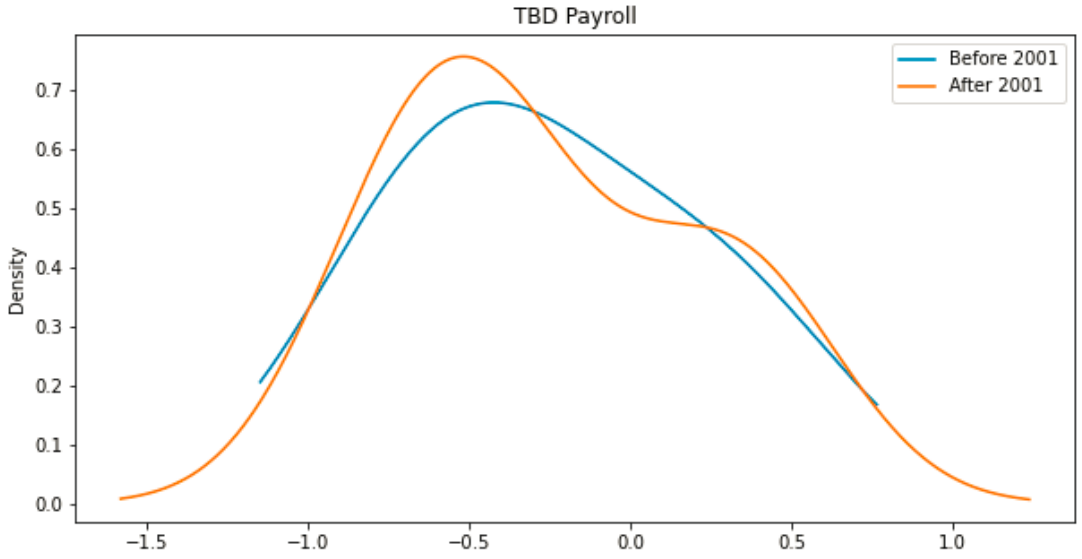
It is clear that DET has been more ambitious since 2001 in recruiting high market value players or provided big contracts to outstanding rosters in order to boost the teams' performance.

Another “rich” example is Boston Red Sox (BOS):



It seems after 2001, the team has invested much more than before trying to enhance its competitiveness.

On the other hand, how about small - mid market teams such as Tampa Bay Devil Rays (TBD)?



As a young team joined the league in 1998, TBD tends to keep the payroll budget over years under the overall league standard.

Suggestions for next steps in analyzing this data

So far the data set has been merged, cleaned, normalized and featured. In addition, a basic but essential data exploratory has also been conducted to figure out the insights of important features. It would be more interesting to

begin the data modeling and prediction to see how these features affect the teams' performance. Some questions that are worth to study are:

1. How do offensive indicators such as OPS contribute to winning?
2. And how about pitching or defense performance?
3. Does higher payroll make a winning team?

The expected deliverable will include a model that contains factors to develop a competitive MLB team. It would be a good predictive tool for team managers to analyze team status and to establish team building plans.

A paragraph that summarizes the quality of this data set and a request for additional data if needed

In general, the data set that is acquired from Sean Lahman's site is quite comprehensive and with good quality. There are very few missing values that appear in the data set and are easy to solve.

Currently the main data set consists of two types of data, which are team performance data and salary data. It might be sufficient for the follow up analysis and modeling usage. Individual play statistics, such as Batting.csv, Pitching.csv and Fielding.csv can also be included in the later stage so that we can extend our analysis scope to see who are the best fit players for a particular team.