Moneyball_An_Analysis_of_Efficiency_in_Baseball_Teams_Spending

June 11, 2023

[25]:
```python
import sqlite3
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

sqlite_file = 'lahman2014.sqlite'
conn = sqlite3.connect(sqlite_file)
```

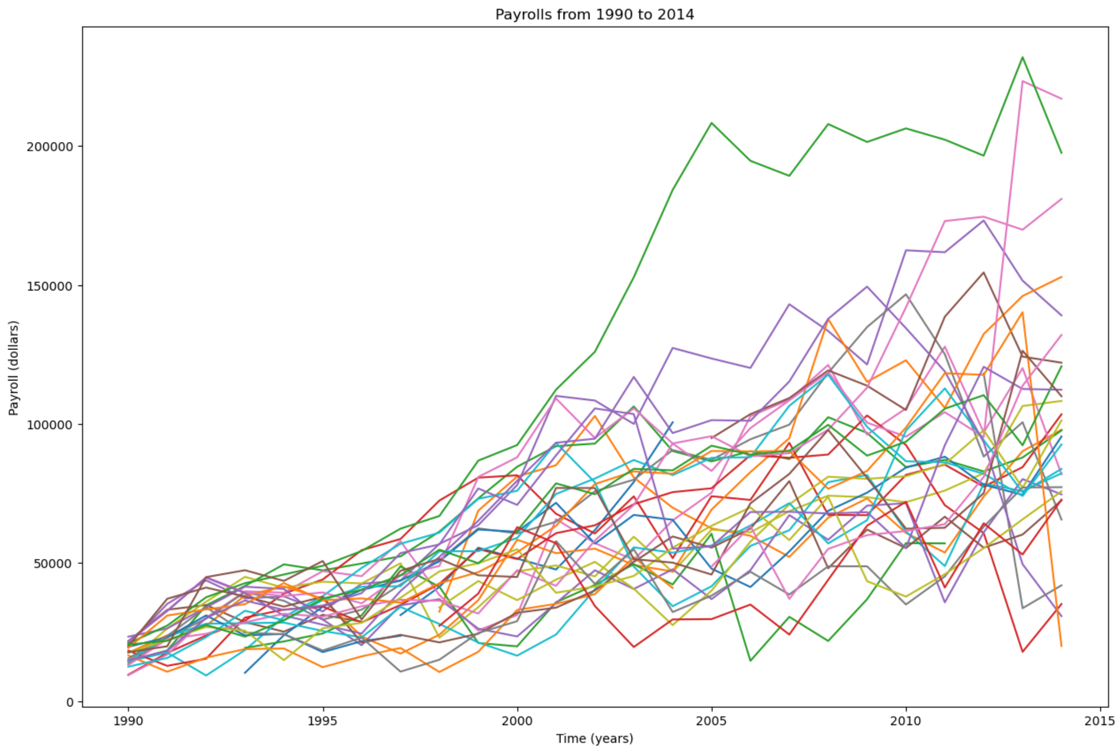
[26]:
```python
salaries = pd.read_sql("SELECT teamID, yearID, sum(salary)/1000 as␣
 ↪payroll_by_thousand, sum(salary)/count(salary) as mean_payroll FROM Salaries␣
 ↪GROUP BY teamID, yearID ORDER BY teamID"
, conn)

wins = pd.read_sql("SELECT teamID, yearID, W as wins, G as games, ((W*100) /␣
 ↪(G)) as winnings, franchID FROM teams WHERE yearID >= 1990 GROUP BY teamID,␣
 ↪yearID ORDER BY teamID", conn)

final = salaries.merge(wins)

final
```

[26]:
```
    teamID  yearID  payroll_by_thousand  mean_payroll  wins  games  winnings  \
0      ANA    1997            31135.472  1.004370e+06    84    162        51
1      ANA    1998            41281.000  1.214147e+06    85    162        52
2      ANA    1999            55388.166  1.384704e+06    70    162        43
3      ANA    2000            51464.167  1.715472e+06    82    162        50
4      ANA    2001            47535.167  1.584506e+06    75    162        46
..     …       …                   …           …     …      …          …
723    WAS    2010            61400.000  2.046667e+06    69    162        42
724    WAS    2011            63856.928  2.201963e+06    80    161        49
725    WAS    2012            80855.143  2.695171e+06    98    162        60
726    WAS    2013           113703.270  4.548131e+06    86    162        53
727    WAS    2014           131983.680  4.399456e+06    96    162        59

    franchID
0        ANA
```

```
1         ANA
2         ANA
3         ANA
4         ANA
..        …
723       WSN
724       WSN
725       WSN
726       WSN
727       WSN

[728 rows x 8 columns]
```

[27]:

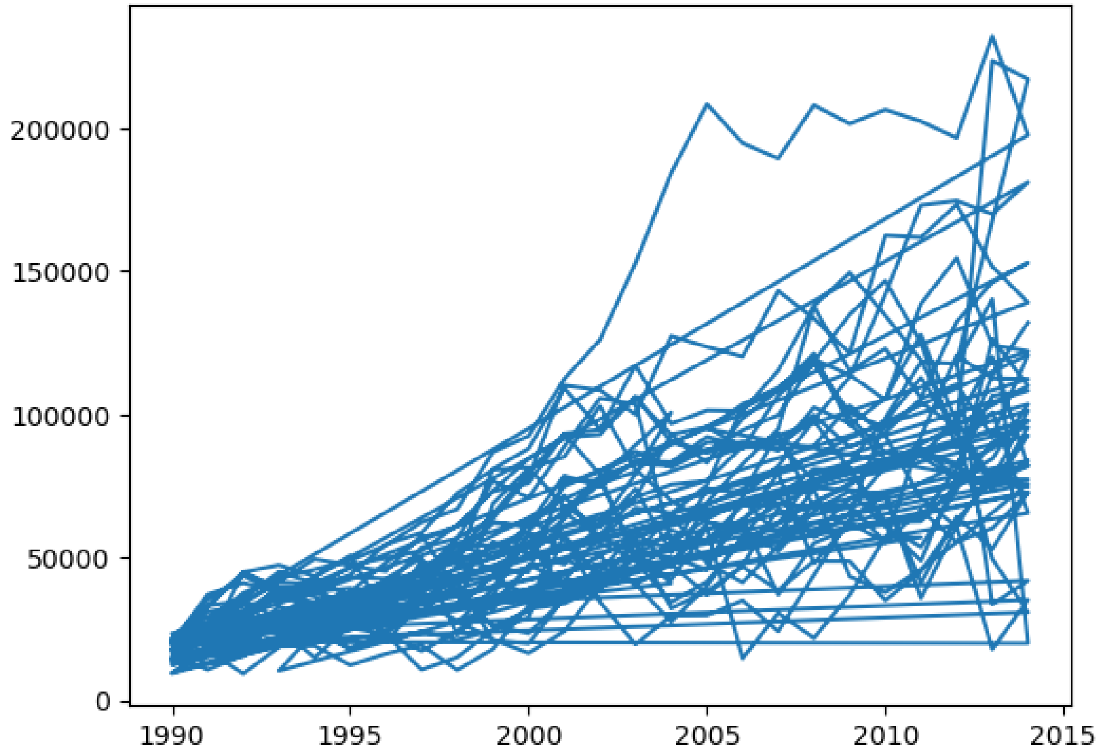
```
group = np.unique(salaries.iloc[:,0].values)

years = pd.DataFrame(columns = ['yearID'], data = np.arange(1990, 2015))

data = {}

plt.figure(figsize = (15, 10))
plt.xlabel("Time (years)")
plt.ylabel("Payroll (dollars)")
plt.title("Payrolls from 1990 to 2014")

for x in group:
    data[x] = years.merge(salaries[['yearID','teamID','payroll_by_thousand']].
 ↪groupby(['teamID']).get_group(x))
    plt.plot(data[x]['yearID'], data[x]['payroll_by_thousand'])

plt.show()
```



We can see a positive correlation between payroll and time per team. The spread of the data increases with time which can be seen by the range of the data, when looking at the starting payroll quanitiy and the ending.

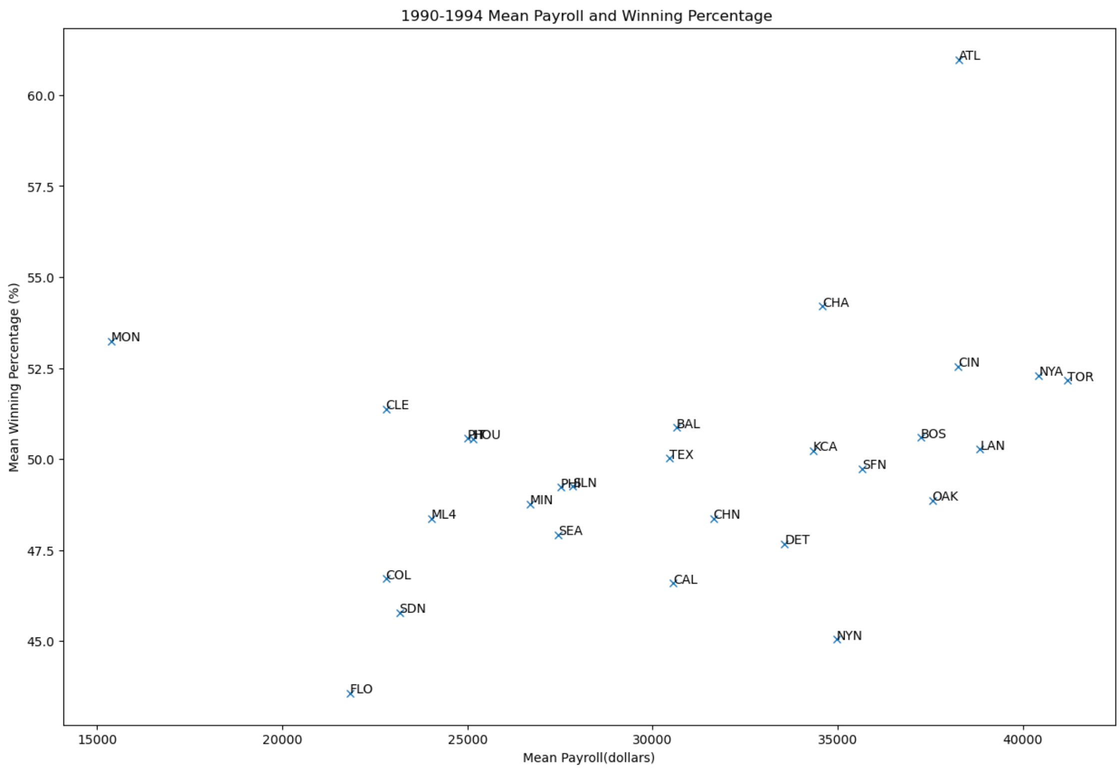
[28]:
```
plt.plot(final['yearID'],final['payroll_by_thousand'])
```

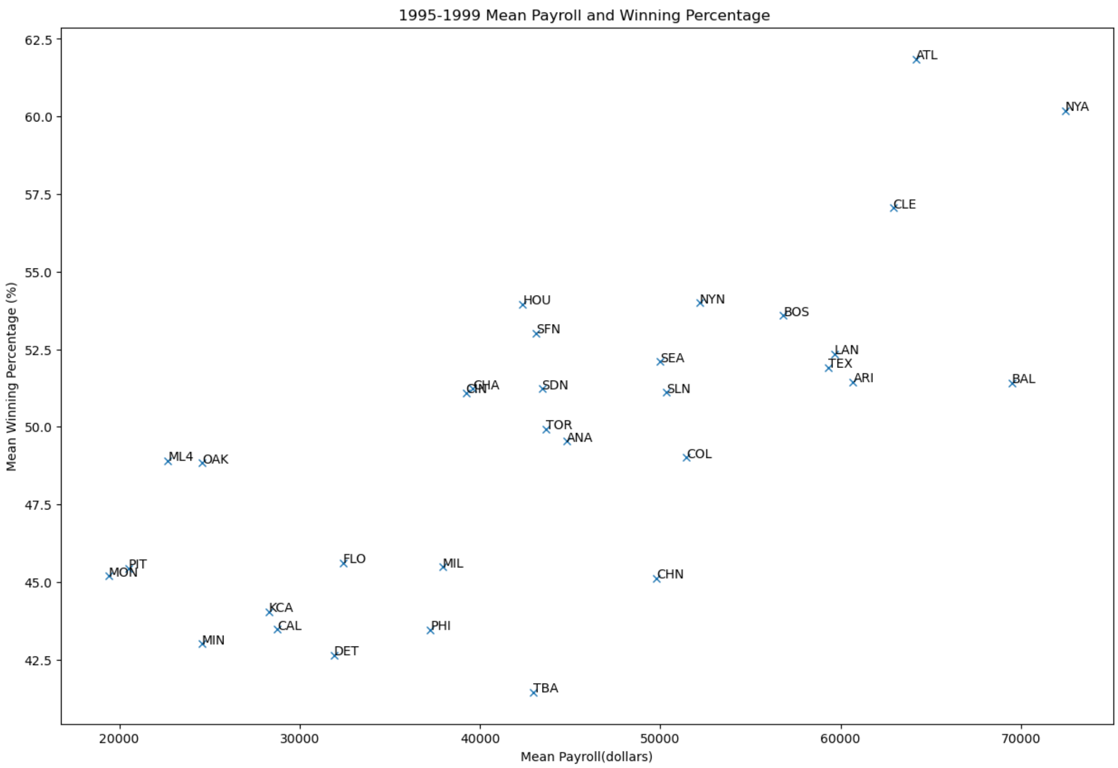
[28]: [<matplotlib.lines.Line2D at 0xffff41a33220>]

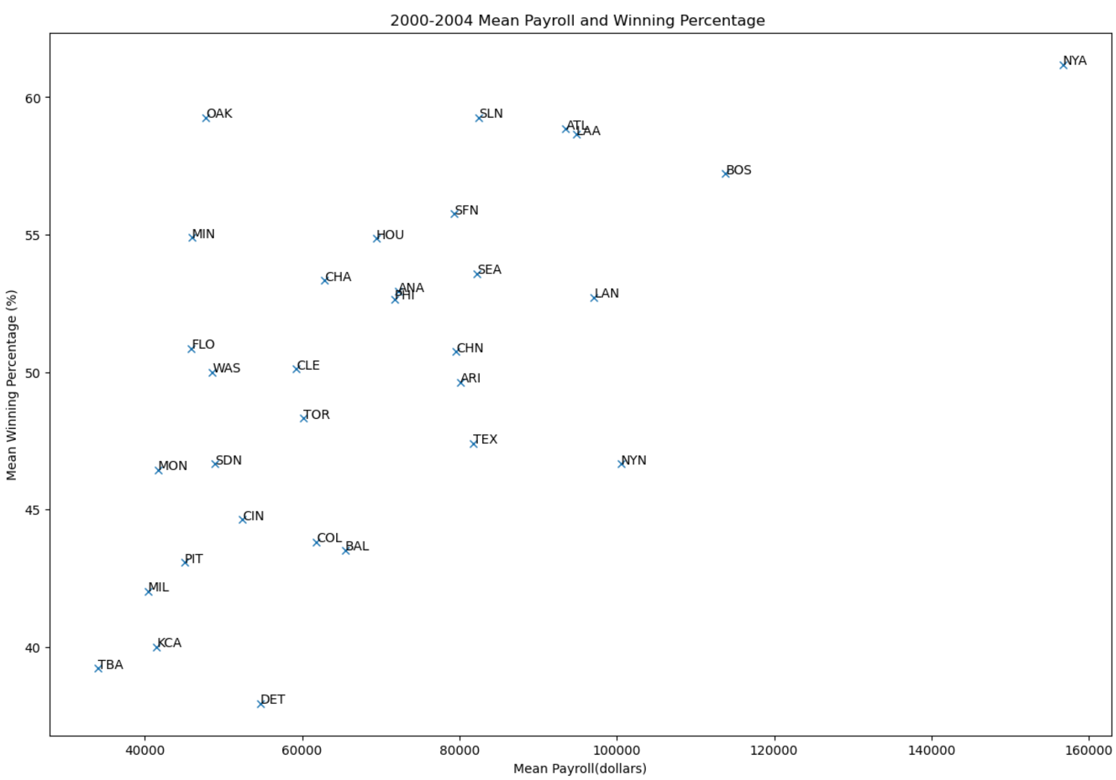


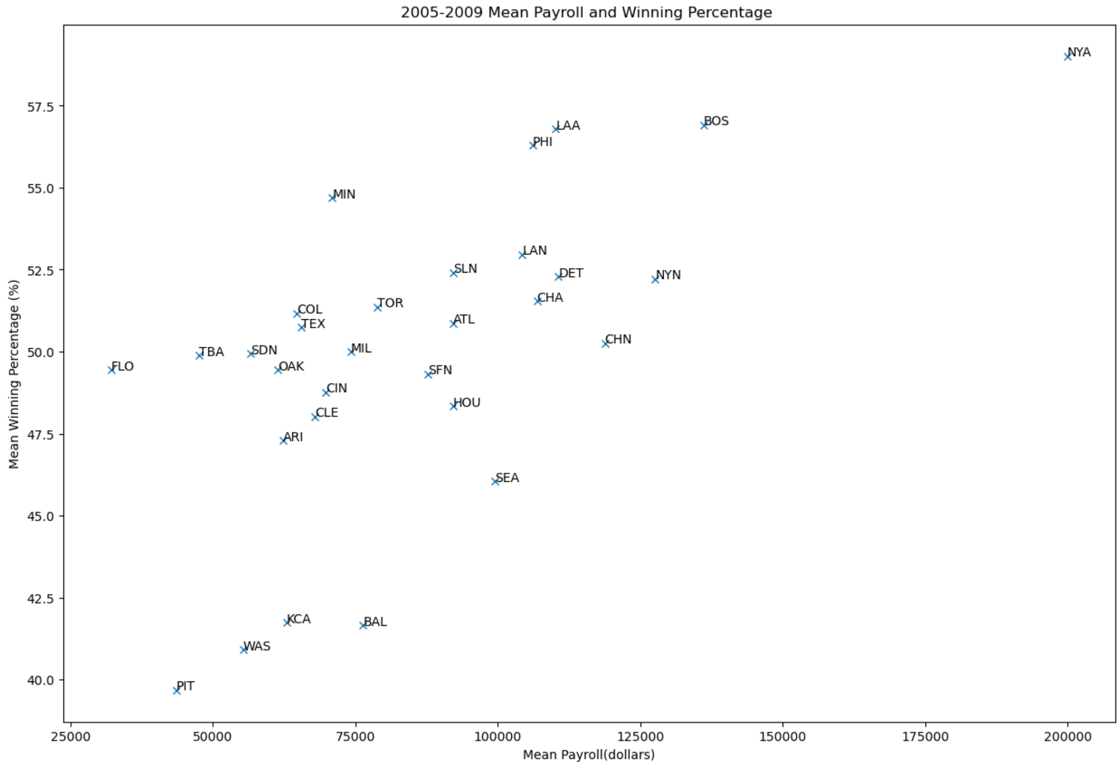
[29]:
```
years = pd.DataFrame(columns=['yearID'], data=np.arange(1990, 2015))
table = years.merge(final[['yearID', 'teamID', 'payroll_by_thousand', 'wins',␣
 ↪'games']])

# set bin labels as periods
periods = ['1990-1994', '1995-1999', '2000-2004', '2005-2009', '2010-2014']
bins = [1990, 1995, 2000, 2005, 2010, 2015]
table['period'] = pd.cut(table['yearID'], bins=bins, labels=periods)

for x in periods:
    tbl = table[table['period'] == x].copy()
    tbl['win_rate'] = (100 * tbl['wins']) / (tbl['games'])


    pay = (tbl.groupby(['teamID']))['payroll_by_thousand'].mean().to_frame()
    win = (tbl.groupby(['teamID']))['win_rate'].mean().to_frame()

    pay['teamID'] = pay.index
    win['teamID'] = win.index
    pay.reset_index(drop=True, inplace=True)
    win.reset_index(drop=True, inplace=True)
    result = pay.merge(win)
```

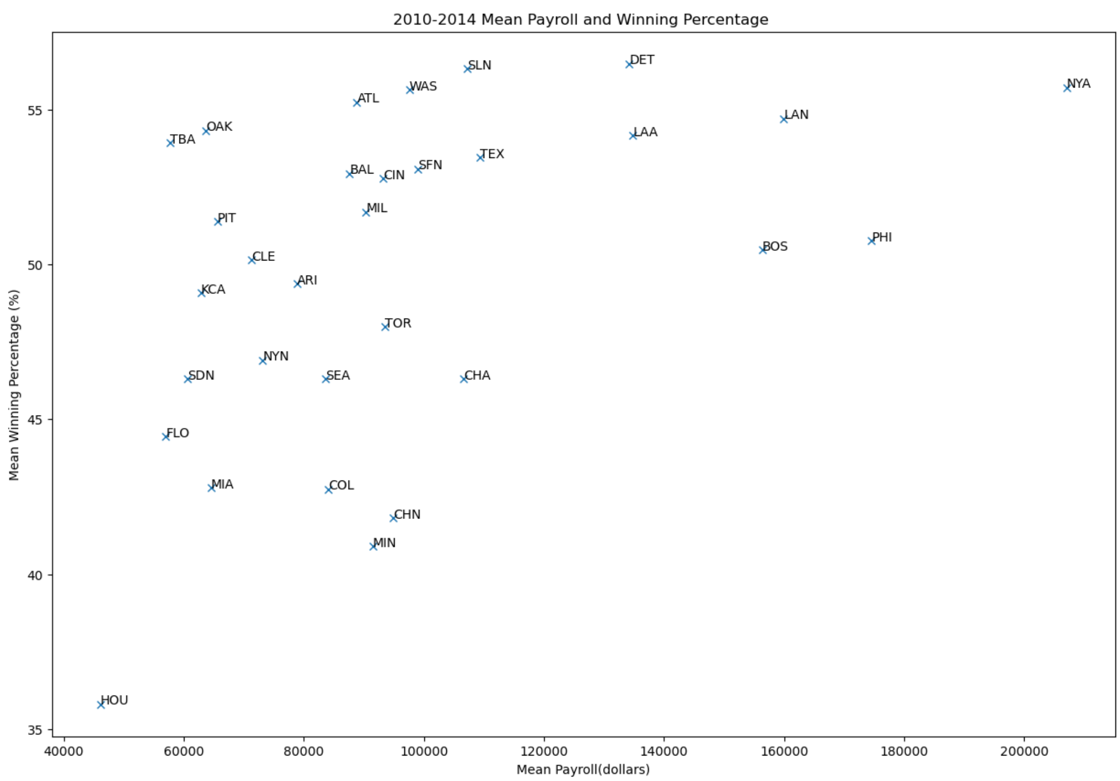
```
    result.columns = ['mean_pay', 'teamID', 'mean_win']

    x_data = result['mean_pay'].values
    y_data = result['mean_win'].values
    plt.figure(figsize=(15, 10))
    plt.plot(x_data, y_data, 'x')

    plt.title(x + " Mean Payroll and Winning Percentage ")
    plt.ylabel("Mean Winning Percentage (%)")
    plt.xlabel("Mean Payroll(dollars)")

    for y, txt in enumerate(result['teamID']):
        plt.annotate(txt, (x_data[y], y_data[y]), size=10)

plt.show()
```




1995-1999 Mean Payroll and Winning Percentage
Mean Winning Percentage (%)
Mean Payroll(dollars)
62.5
60.0
57.5
55.0
52.5
50.0
47.5
45.0
42.5
20000
30000
40000
50000
60000
70000
ATL
NYA
CLE
HOU
NYN
BOS
SFN
LAN
SEA
TEX
ARI
BAL
CHA
CIN
SDN
SLN
TOR
ANA
COL
ML4
OAK
PIT
MON
FLO
MIL
CHN
KCA
CAL
PHI
MIN
DET
TBA


2000-2004 Mean Payroll and Winning Percentage
Mean Winning Percentage (%)
Mean Payroll(dollars)
60
55
50
45
40
40000
60000
80000
100000
120000
140000
160000
NYA
OAK
SLN
ATL
LAA
BOS
SFN
MIN
HOU
CHA
SEA
ANA
PHI
LAN
FLO
CHN
WAS
CLE
ARI
TOR
TEX
MON
SDN
NYN
CIN
COL
BAL
PIT
MIL
KCA
TBA
DET


2005-2009 Mean Payroll and Winning Percentage
Mean Winning Percentage (%)
Mean Payroll(dollars)
NYA
BOS
LAA
PHI
MIN
LAN
SLN
DET
NYN
CHA
TOR
COL
TEX
ATL
CHN
MIL
SDN
TBA
FLO
OAK
SFN
CIN
HOU
CLE
ARI
SEA
KCA
BAL
WAS
PIT


2010-2014 Mean Payroll and Winning Percentage
Mean Winning Percentage (%)
Mean Payroll(dollars)
DET
SLN
WAS
NYA
ATL
LAN
OAK
TBA
LAA
TEX
SFN
BAL
CIN
MIL
PIT
PHI
BOS
CLE
ARI
KCA
TOR
NYN
SDN
SEA
CHA
FLO
MIA
COL
CHN
MIN
HOU

The graphs above show a positive correlation between mean payroll and mean winning percentage. As time goes on, the average starts to move towards a From all the graphs of the different time periods, NYA seems to stand out for being good at paying for wins across these time periods. Oakland A tends to be on the lower end of the mean payroll as the years go on while still having relatively high mean winning percentages.

[30]:
```
std_col = []
salaries = pd.read_sql("SELECT Teams.teamID as team_ID, Salaries.yearID,␣
↪AVG(100.00*W/G) as win, AVG(salary) as mean FROM Salaries LEFT JOIN Teams ON␣
↪Teams.teamID = Salaries.teamID WHERE Salaries.yearID >= 1990 and Salaries.
↪yearID <= 2014 GROUP BY Salaries.yearID, Teams.teamID", conn)
payroll = pd.read_sql("SELECT yearID, AVG(salary) as mean FROM Salaries WHERE␣
↪yearID <= 2014  and yearID >= 1990", conn)

for r in salaries.iterrows():
    standard = 0
    if (np.std(salaries[salaries['yearID'] == r[1]['yearID']]['mean']) != 0):
        standard = (r[1]['mean']-payroll.loc('yearID' == r[1]['yearID'])[0][1])/
↪np.std(salaries[salaries['yearID'] == r[1]['yearID']]['mean'])
    std_col.append(standard)
salaries['standard'] = std_col
salaries = salaries.drop(labels=0, axis=0)
salaries
```

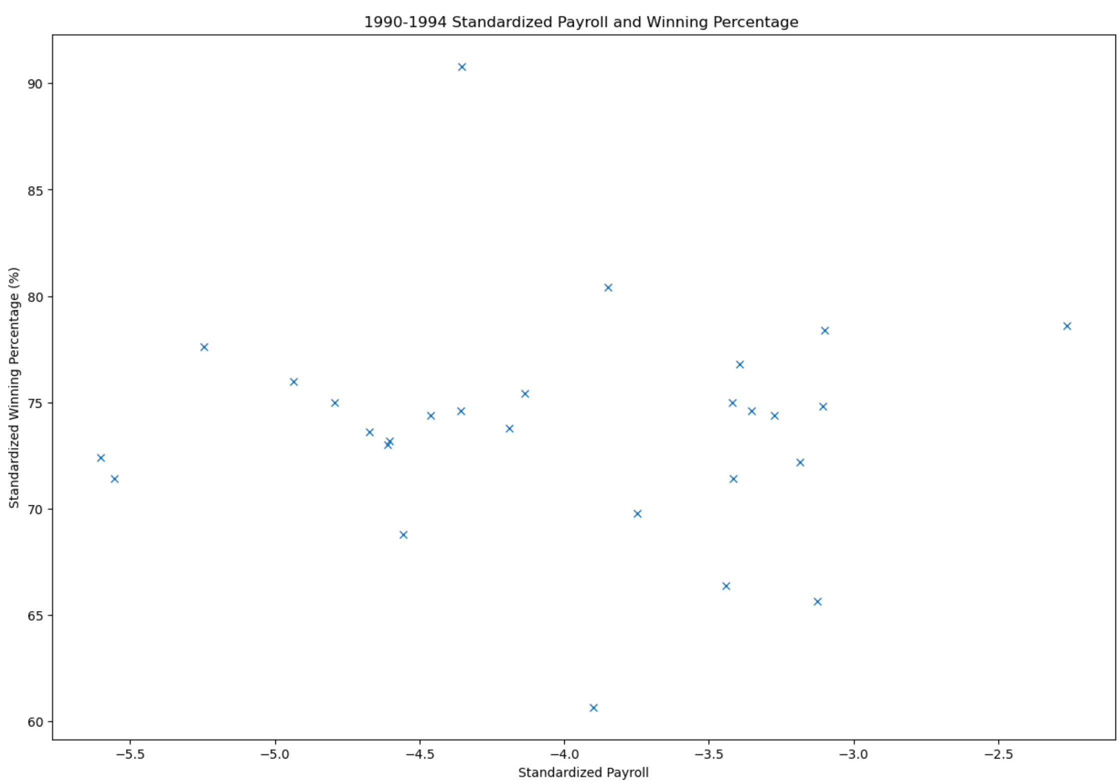
[30]:

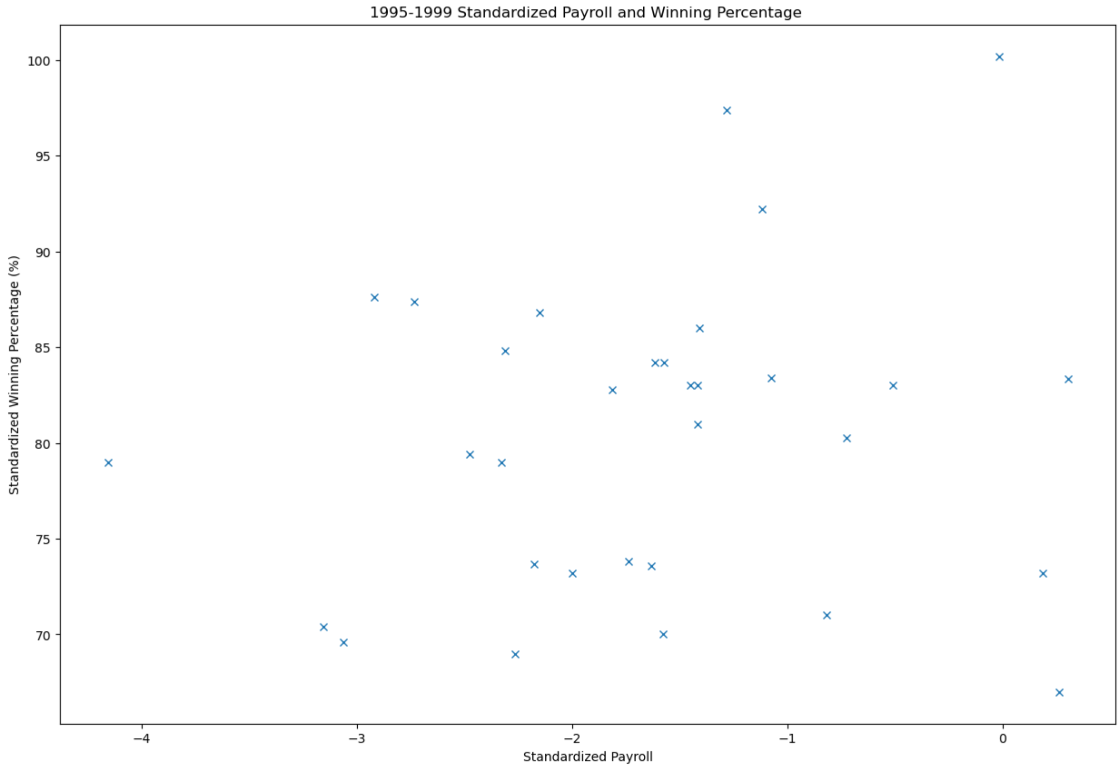
	team_ID	yearID	win	mean	standard
1	BAL	1990	51.260522	2.616239e+05	-16.797818
2	BOS	1990	51.427450	6.424479e+05	-13.426350
3	CAL	1990	48.171550	6.205714e+05	-13.620024
4	CHA	1990	50.195230	3.061774e+05	-16.403382
5	CHN	1990	51.115255	4.394839e+05	-15.223208
..	…	…	…	…	…
724	SLN	2014	50.560881	4.310464e+06	0.647272
725	TBA	2014	46.210623	2.907564e+06	0.225203
726	TEX	2014	49.075709	4.677294e+06	0.757635
727	TOR	2014	49.287603	4.396804e+06	0.673248
728	WAS	2014	42.719816	4.399456e+06	0.674046

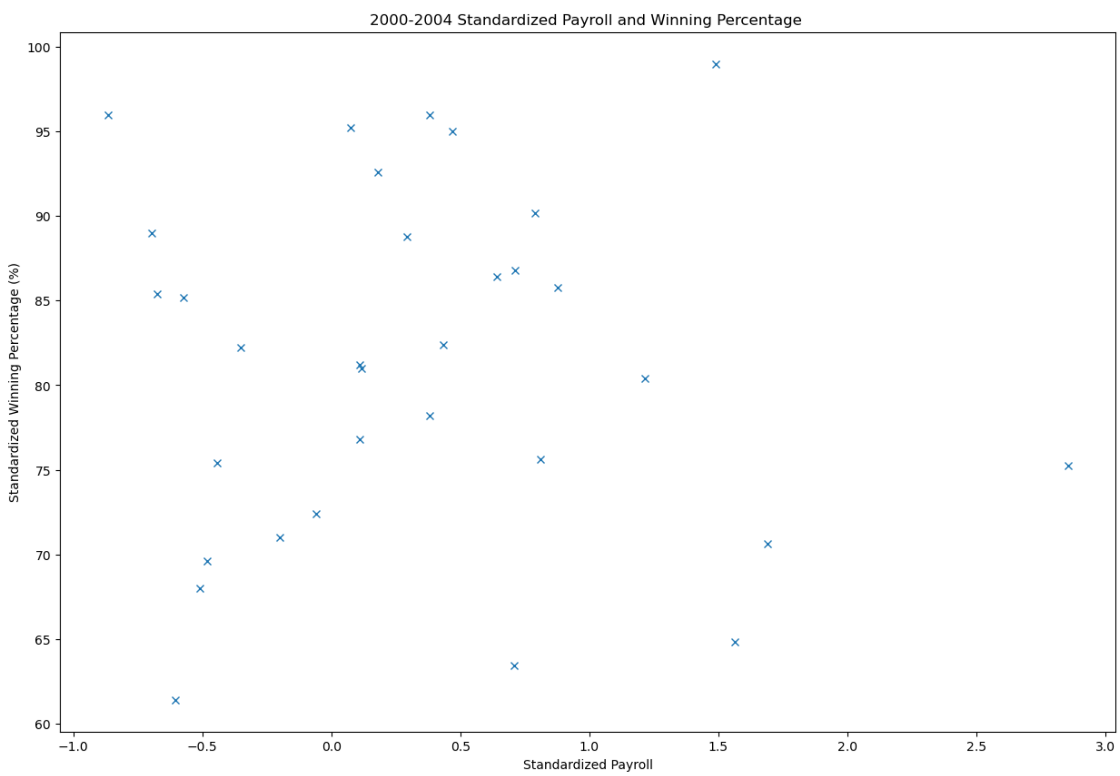
[728 rows x 5 columns]

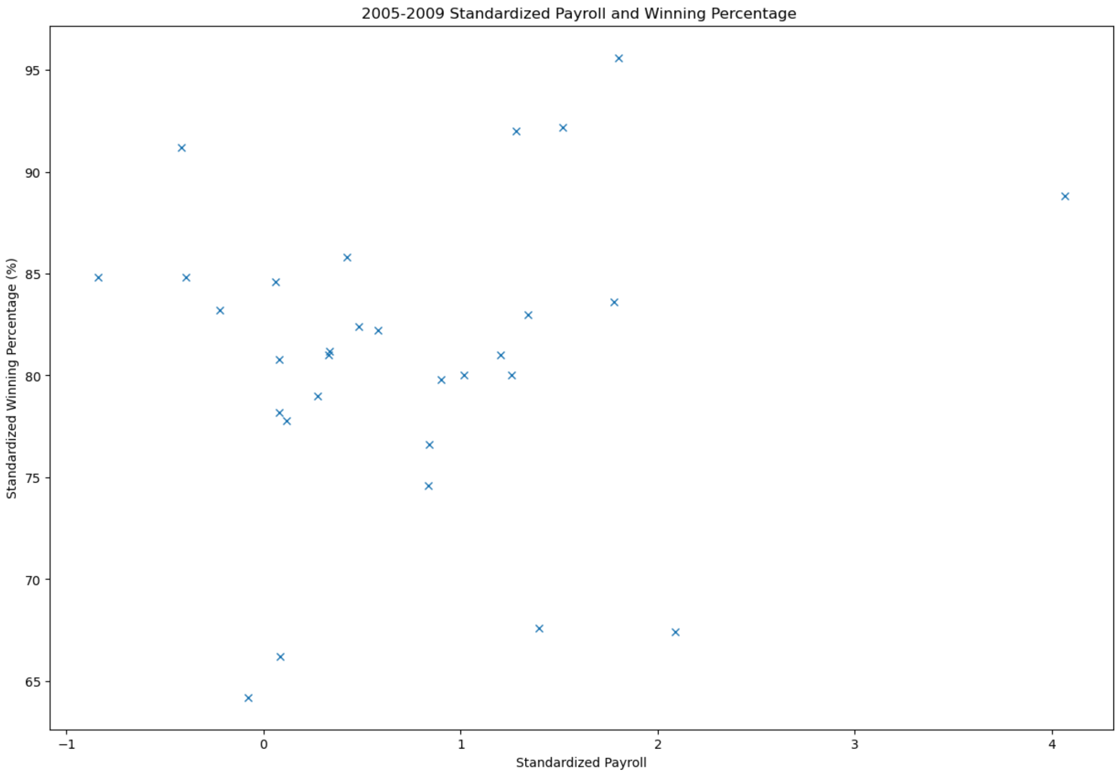
[31]:
```
table['standard_pay'] = salaries['standard'].values

for x in periods:
    tbl = table[table['period'] == x].copy()
    tbl['winnings'] = (100*tbl['wins']) / (tbl['games'])
```

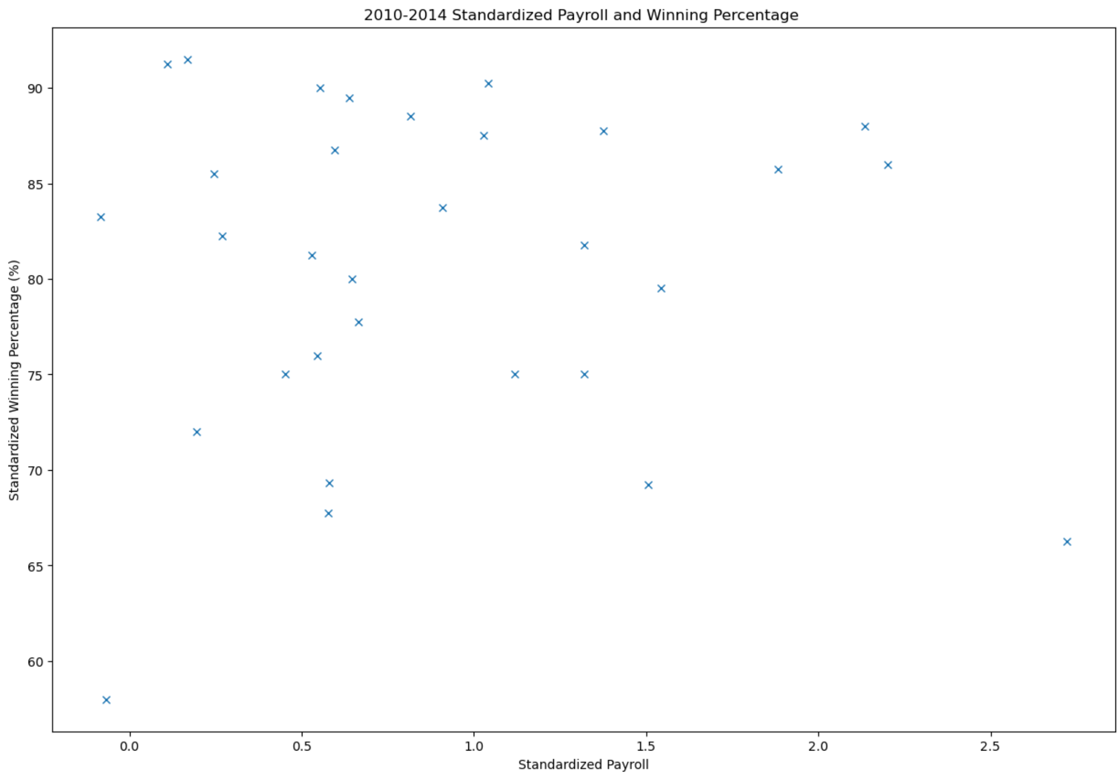
```
    payroll = tbl.groupby(['teamID'])['standard_pay'].mean().to_frame()
    win = tbl.groupby(['teamID'])['wins'].mean().to_frame()

    payroll['teamID'] = payroll.index
    win['teamID'] = win.index
    payroll.columns = ['standard_pay', 'teamID']
    win.columns = ['standard_win', 'teamID']

    payroll.reset_index(drop=True, inplace=True)
    win.reset_index(drop=True, inplace=True)
    result = payroll.merge(win)
    result.columns = ['standard_pay', 'teamID', 'standard_win']

    x_data = result['standard_pay'].values
    y_data = result['standard_win'].values
    plt.figure(figsize=(15,10))
    plt.plot(x_data, y_data,'x')

    plt.title(x + " Standardized Payroll and Winning Percentage ")
    plt.xlabel("Standardized Payroll")
    plt.ylabel("Standardized Winning Percentage (%)")

plt.show()
```




1995-1999 Standardized Payroll and Winning Percentage
Standardized Winning Percentage (%)
Standardized Payroll
100
95
90
85
80
75
70
−4
−3
−2
−1
0

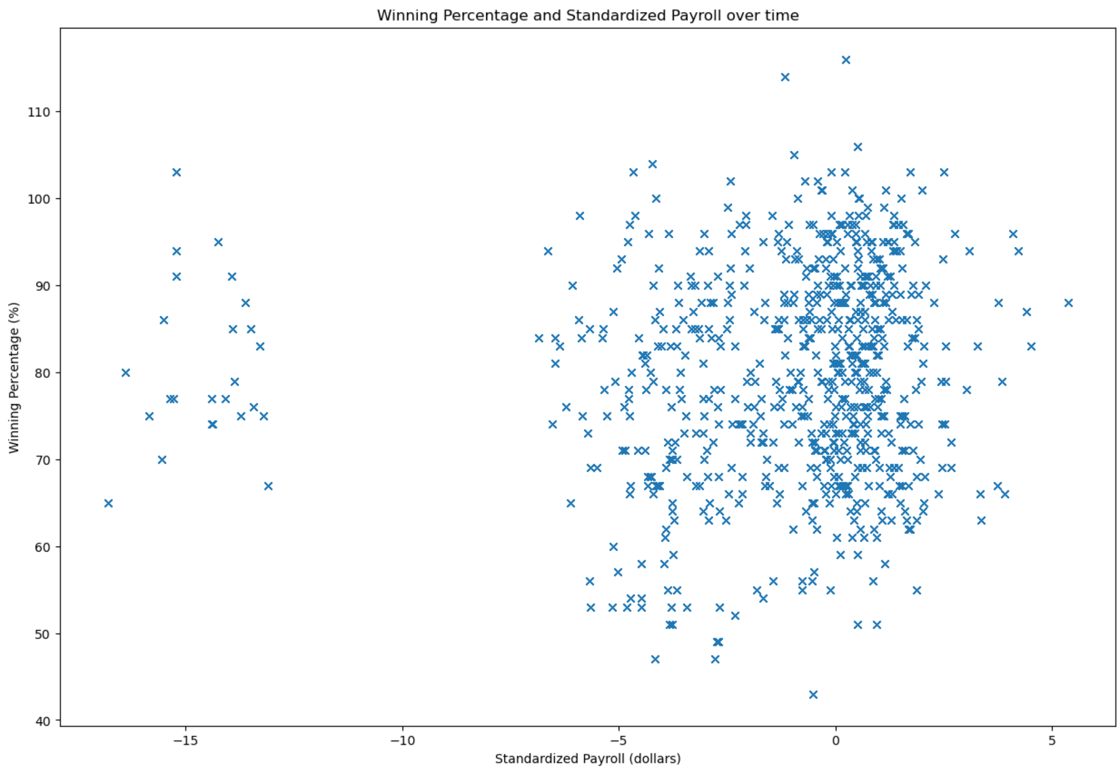

2000-2004 Standardized Payroll and Winning Percentage
Standardized Winning Percentage (%)
Standardized Payroll
100
95
90
85
80
75
70
65
60
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0


2005-2009 Standardized Payroll and Winning Percentage
Standardized Winning Percentage (%)
Standardized Payroll
95
90
85
80
75
70
65
−1
0
1
2
3
4

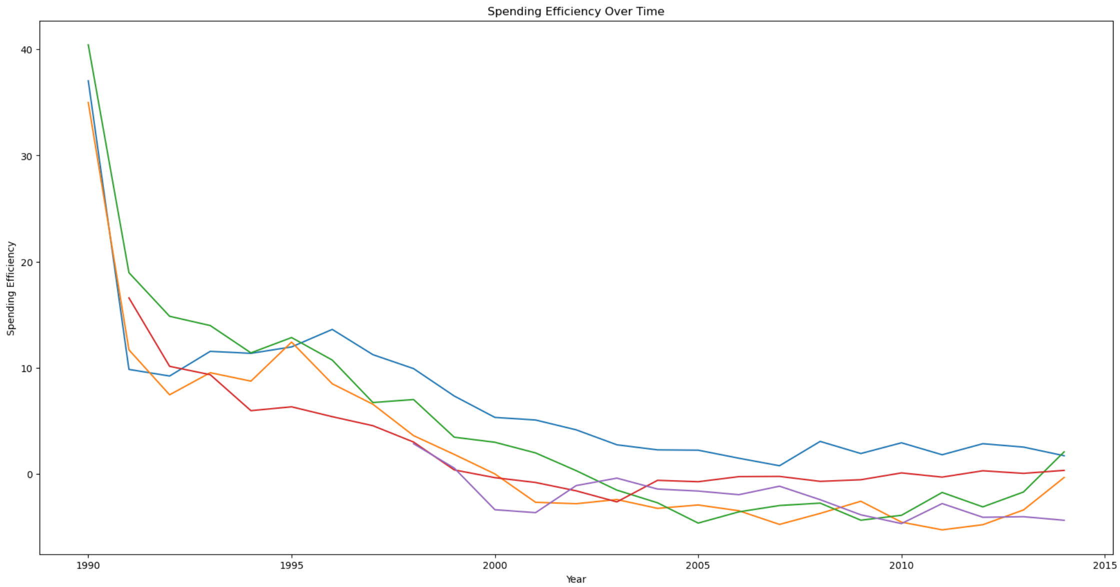


Just like problem 4, we continue to see a positive correlation between mean payroll and winning percentages when using the standard payroll. It is however less significant or drastic as it was previously. In the first few periods however, the x-axis of mean payroll shifts to include some negative values which we did not previously see.

[32]:
```
winning = pd.DataFrame(columns = ['yearID', 'win_percentage'])
winning['yearID'] = table['yearID']

table = table.merge(winning)

x_data = table['standard_pay'].values
y_data = table['wins'].values

plt.figure(figsize=(15,10))
plt.plot(x_data, y_data,'x')

plt.title("Winning Percentage and Standardized Payroll over time")
plt.xlabel("Standardized Payroll (dollars)")
plt.ylabel("Winning Percentage (%)")
plt.show()
```



[33]:
```
col = []
my_teams = ["OAK", "BOS", "NYA", "ATL", "TBA"]
for r in salaries.iterrows():
    eff = r[1]['win']- (50 + (2.5 * r[1]['standard']))
    col.append(eff)

salaries['efficiency'] = col

plt.figure(figsize=(20,10))
plt.xlabel('Year')
plt.ylabel('Spending Efficiency')
plt.title('Spending Efficiency Over Time')


for i in my_teams:
    i = salaries.loc[(salaries['team_ID'] == i)]
    plt.plot(i.yearID,i.efficiency, label=i)

plt.show()
```



Unlike the other plots, this plot shows a downward trend for all the teams between year and spending efficiency. Oakland's efficiency starts off as the second highest and ends the same. It takes the lead in efficiency around the years 1996-2013.