project2

October 22, 2020

[1]:
```
import sqlite3
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

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

[2]:
```
salary_query = "SELECT yearID, teamID, sum(salary) as total_payroll FROM␣
↪Salaries GROUP BY yearID, teamID"

team_salaries = pd.read_sql(salary_query, conn)
team_salaries.head()
```

[2]:

	yearID	teamID	total_payroll
0	1985	ATL	14807000.0
1	1985	BAL	11560712.0
2	1985	BOS	10897560.0
3	1985	CAL	14427894.0
4	1985	CHA	9846178.0

1 Part 1

2 Problem 1

[3]:
```
# From above table, we see that the first table starts from 1985 while the␣
↪second table starts from year 1901.

winning_query = "SELECT yearID, teamID, franchID, W, G, (W*1.0/G*1.0)*100 as␣
↪winning_percentage FROM Teams WHERE yearID >= 1985 GROUP BY yearID, teamID"

team_winning = pd.read_sql(winning_query, conn)
team_winning.head()
```

[3]:

	yearID	teamID	franchID	W	G	winning_percentage
0	1985	ATL	ATL	66	162	40.740741

```
1     1985     BAL       BAL   83   161            51.552795
2     1985     BOS       BOS   81   163            49.693252
3     1985     CAL       ANA   90   162            55.555556
4     1985     CHA       CHW   85   163            52.147239
```

[4]:
```
print ("The winning percentage table has " + str(len(team_winning)) + " data,␣
↪the salary table has " +
       str(len(team_salaries)) + " data. " + str(len(team_winning) -␣
↪len(team_salaries)) + " data are missing.")
```

```
The winning percentage table has 858 data, the salary table has 860 data. -2
data are missing.
```

[5]:
```
team_query = "SELECT Teams.yearID, Teams.teamID, Teams.franchID, Teams.W, Teams.
↪G, \
(Teams.W*1.0/Teams.G*1.0)*100 as winning_percentage, sum(Salaries.salary) as␣
↪total_payroll \
FROM Salaries \
INNER JOIN Teams \
ON Salaries.yearID = Teams.yearID AND Salaries.teamID = Teams.teamID \
WHERE Teams.yearID >= 1990 AND Teams.yearID <= 2014 \
GROUP BY Salaries.yearID, Salaries.teamID"

team = pd.read_sql(team_query, conn)
team
```

[5]:
```
     yearID teamID franchID   W    G  winning_percentage  total_payroll
0      1990    ATL      ATL  65  162           40.123457     14555501.0
1      1990    BAL      BAL  76  161           47.204969      9680084.0
2      1990    BOS      BOS  88  162           54.320988     20558333.0
3      1990    CAL      ANA  80  162           49.382716     21720000.0
4      1990    CHA      CHW  94  162           58.024691      9491500.0
..      …      …        …   ..   …                  …              …
723    2014    SLN      STL  90  162           55.555556    120693000.0
724    2014    TBA      TBD  77  162           47.530864     72689100.0
725    2014    TEX      TEX  67  162           41.358025    112255059.0
726    2014    TOR      TOR  83  162           51.234568    109920100.0
727    2014    WAS      WSN  96  162           59.259259    131983680.0

[728 rows x 7 columns]
```

[6]:
```
# The two table differs because they start from different year. During the same␣
↪time period, they contains different number of
# data. 2 data are missed in the winning table.
# We can join the franchID, W, G, winning percentage from wining table with␣
↪salary from salary table.
# Only the data from 199
```

```python
# The new table's size is:
print (str(len(team)))
```

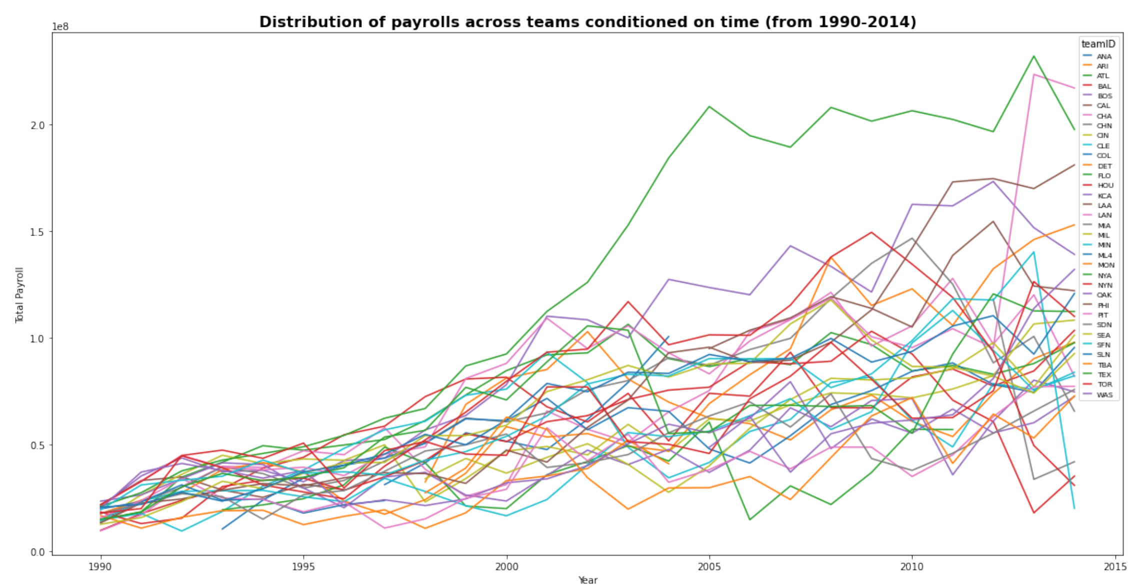
728

3 Part 2

4 Problem 2

[7]:
```python
distribution = team.pivot(index='yearID', columns='teamID', values='total_payroll')
plt.rcParams["figure.figsize"] = [20,10]
params = {'legend.fontsize': 7.8,
          'legend.handlelength': 1}
plt.rcParams.update(params)
ax = distribution.plot()
ax.set_xlabel("Year")
ax.set_ylabel("Total Payroll")
plt.title('Distribution of payrolls across teams conditioned on time (from 1990-2014)', size=16, weight='bold')
```

[7]:
```
Text(0.5, 1.0, 'Distribution of payrolls across teams conditioned on time (from
1990-2014)')
```



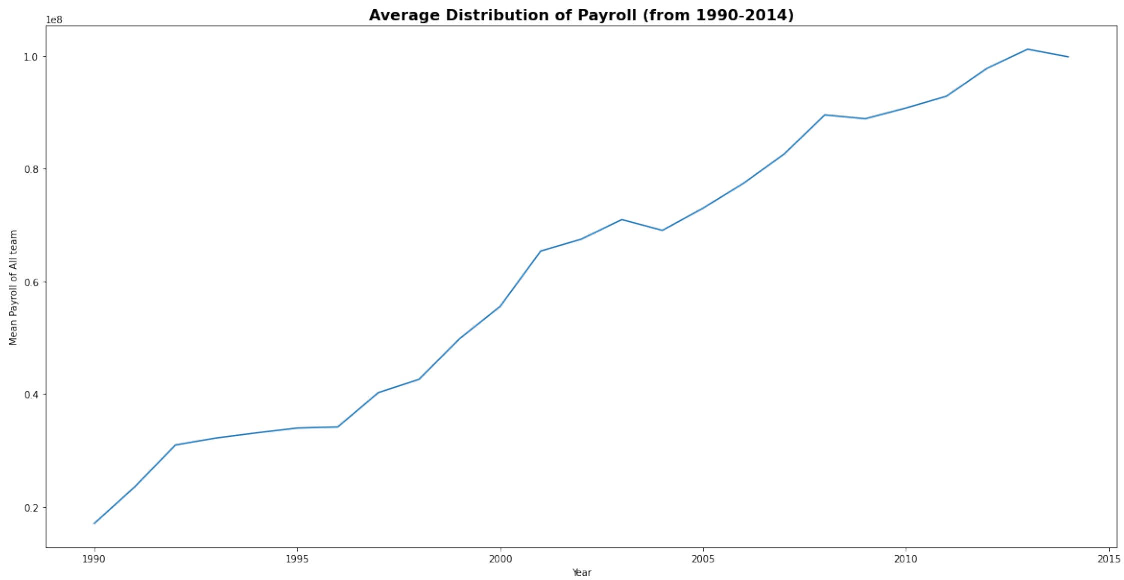
5 Question 1

[8]:
```
# From the graph above, we see that from 1990 to 2014, the total payroll of␣
↪each team is increasing generally.
# In the year of 1990, the salary of each teams seemed to be low and centered.␣
↪But in 2014, the income gap of each team seems
# is greater.
```

6 Problem 3

[9]:
```
mean_payroll = distribution.mean(axis = 1)
mean_ax = mean_payroll.plot()
mean_ax.set_ylabel("Mean Payroll of All team")
mean_ax.set_xlabel("Year")
plt.title("Average Distribution of Payroll (from 1990-2014)", size=16,␣
↪weight='bold')
```

[9]:
```
Text(0.5, 1.0, 'Average Distribution of Payroll (from 1990-2014)')
```



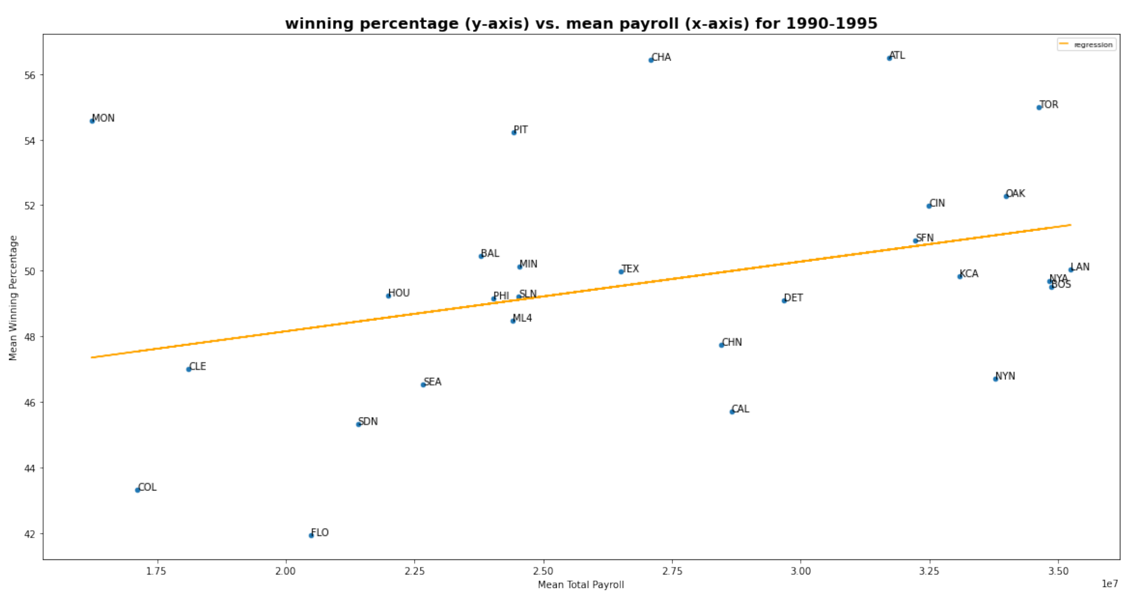
7 Problem 4

[10]:
```
bins = [1990, 1995, 2000, 2005, 2010, 2015]
team['Bin'] = pd.cut(team['yearID'], bins, right=False,␣
↪labels=['1990-1995','1995-2000','2000-2005','2005-2010','2010-2015'])
team1990 = team[team['Bin'] == '1990-1995']
team1995 = team[team['Bin'] == '1995-2000']
```

```
team2000 = team[team['Bin'] == '2000-2005']
team2005 = team[team['Bin'] == '2005-2010']
team2010 = team[team['Bin'] == '2010-2015']
```

[11]:
```
# Year 1990-1995
team1990 = team1990.groupby(['teamID']).mean()
team1990 = team1990.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team1990['total_payroll'],team1990['winning_percentage'],1)
f = np.poly1d(d)
team1990.insert(2,'regression',f(team1990['total_payroll']))
team1990r = team1990[['total_payroll','regression']].copy()

team1990_ax = team1990.plot(x = 'total_payroll', y = 'winning_percentage',␣
↪kind='scatter')
for index, row in team1990.iterrows():
    team1990_ax.annotate(index, (row['total_payroll'],␣
↪row['winning_percentage']))
team1990r.plot(x = 'total_payroll', y = 'regression', ax = team1990_ax, color =␣
↪'orange')

team1990_ax.set_ylabel("Mean Winning Percentage")
team1990_ax.set_xlabel("Mean Total Payroll")
plt.title("winning percentage (y-axis) vs. mean payroll (x-axis) for␣
↪1990-1995", size=16, weight='bold')
```

[11]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean payroll (x-axis) for
1990-1995')
```

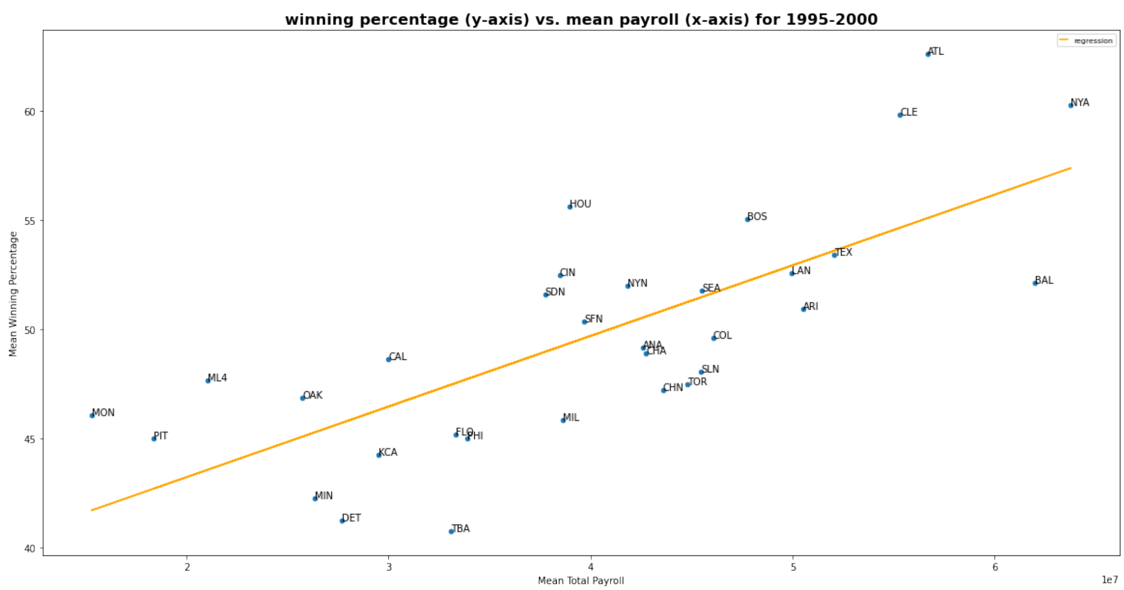


[12]:

```
# Year 1995-2000
team1995 = team1995.groupby(['teamID']).mean()
team1995 = team1995.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team1995['total_payroll'],team1995['winning_percentage'],1)
f = np.poly1d(d)
team1995.insert(2,'regression',f(team1995['total_payroll']))
team1995r = team1995[['total_payroll','regression']].copy()

team1995_ax = team1995.plot(x = 'total_payroll', y = 'winning_percentage',␣
↪kind='scatter')
for index, row in team1995.iterrows():
    team1995_ax.annotate(index, (row['total_payroll'],␣
↪row['winning_percentage']))
team1995r.plot(x = 'total_payroll', y = 'regression', ax = team1995_ax, color =␣
↪'orange')

team1995_ax.set_ylabel("Mean Winning Percentage")
team1995_ax.set_xlabel("Mean Total Payroll")
plt.title("winning percentage (y-axis) vs. mean payroll (x-axis) for␣
↪1995-2000", size=16, weight='bold')
```

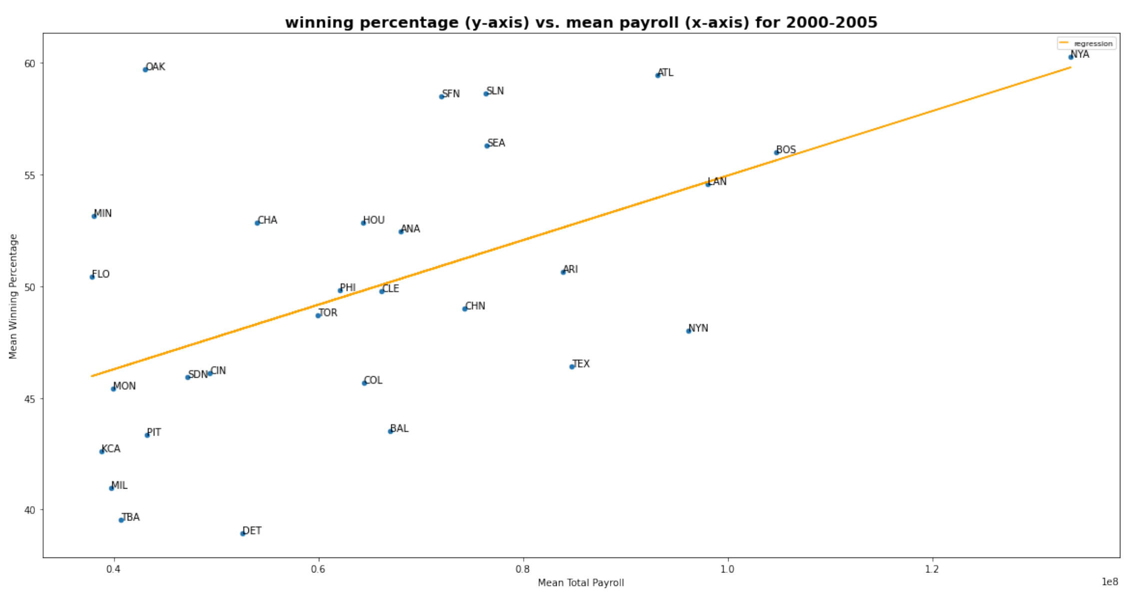
[12]:

```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean payroll (x-axis) for
1995-2000')
```



[13]:
```python
# Year 2000-2005
team2000 = team2000.groupby(['teamID']).mean()
team2000 = team2000.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team2000['total_payroll'],team2000['winning_percentage'],1)
f = np.poly1d(d)
team2000.insert(2,'regression',f(team2000['total_payroll']))
team2000r = team2000[['total_payroll','regression']].copy()

team2000_ax = team2000.plot(x = 'total_payroll', y = 'winning_percentage', kind='scatter')
for index, row in team2000.iterrows():
    team2000_ax.annotate(index, (row['total_payroll'], row['winning_percentage']))
team2000r.plot(x = 'total_payroll', y = 'regression', ax = team2000_ax, color = 'orange')

team2000_ax.set_ylabel("Mean Winning Percentage")
team2000_ax.set_xlabel("Mean Total Payroll")
plt.title("winning percentage (y-axis) vs. mean payroll (x-axis) for 2000-2005", size=16, weight='bold')
```

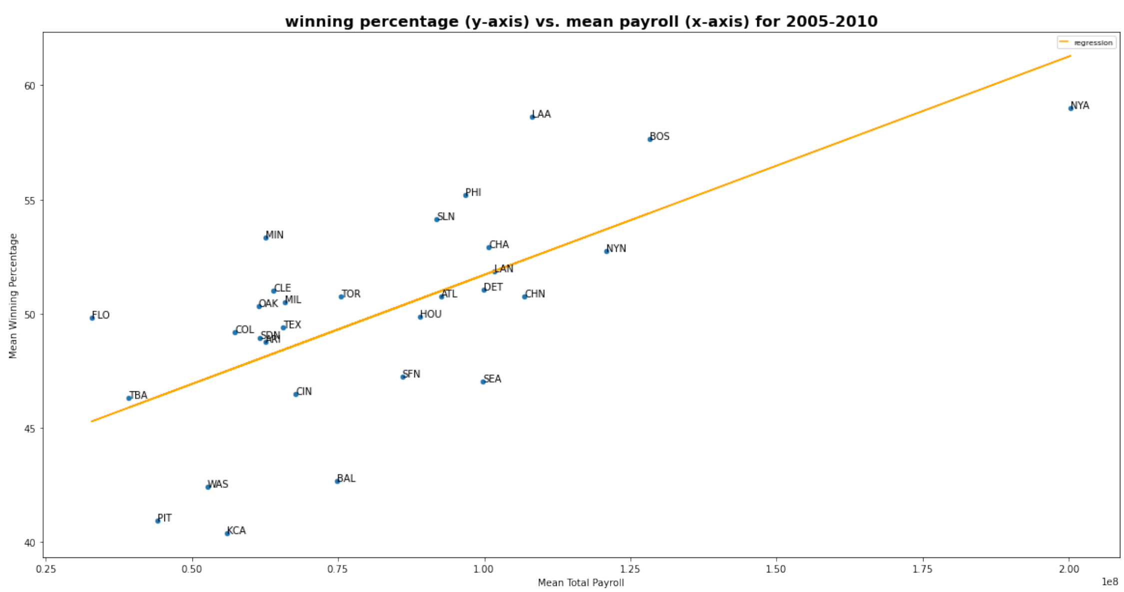
[13]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean payroll (x-axis) for
2000-2005')
```



[14]:
```python
# Year 2005-2010
team2005 = team2005.groupby(['teamID']).mean()
team2005 = team2005.drop(['yearID','W','G'], axis=1)
```

```python
d = np.polyfit(team2005['total_payroll'],team2005['winning_percentage'],1)
f = np.poly1d(d)
team2005.insert(2,'regression',f(team2005['total_payroll']))
team2005r = team2005[['total_payroll','regression']].copy()

team2005_ax = team2005.plot(x = 'total_payroll', y = 'winning_percentage', kind='scatter')
for index, row in team2005.iterrows():
    team2005_ax.annotate(index, (row['total_payroll'], row['winning_percentage']))
team2005r.plot(x = 'total_payroll', y = 'regression', ax = team2005_ax, color = 'orange')

team2005_ax.set_ylabel("Mean Winning Percentage")
team2005_ax.set_xlabel("Mean Total Payroll")
plt.title("winning percentage (y-axis) vs. mean payroll (x-axis) for 2005-2010", size=16, weight='bold')
```

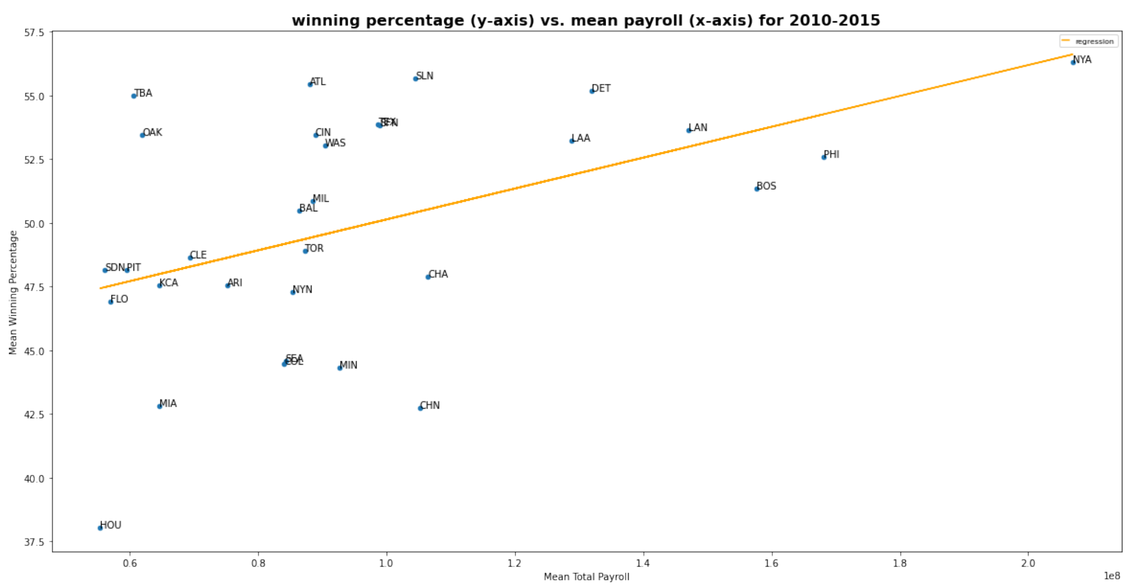
[14]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean payroll (x-axis) for
2005-2010')
```



[15]:
```python
# Year 2010-2015
team2010 = team2010.groupby(['teamID']).mean()
team2010 = team2010.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team2010['total_payroll'],team2010['winning_percentage'],1)
```

```
f = np.poly1d(d)
team2010.insert(2,'regression',f(team2010['total_payroll']))
team2010r = team2010[['total_payroll','regression']].copy()

team2010_ax = team2010.plot(x = 'total_payroll', y = 'winning_percentage',␣
↪kind='scatter')
for index, row in team2010.iterrows():
    team2010_ax.annotate(index, (row['total_payroll'],␣
↪row['winning_percentage']))
team2010r.plot(x = 'total_payroll', y = 'regression', ax = team2010_ax, color =␣
↪'orange')

team2010_ax.set_ylabel("Mean Winning Percentage")
team2010_ax.set_xlabel("Mean Total Payroll")
plt.title("winning percentage (y-axis) vs. mean payroll (x-axis) for␣
↪2010-2015", size=16, weight='bold')
```

[15]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean payroll (x-axis) for
2010-2015')
```



8 Question 2

[16]:
```
# The points generally fit the regression line, so basically the higher the␣
↪pay, the higher the winning percentage.
# The team NYA is particularly good at paying for wins across these time␣
↪periods because they are paid most and they also has
```

```
# one of the highest wining percentage.
# Oakland A's spending efficiency is high because the points are alwasy above␣
↪the regression line, and Since 1995,
# Oakland A's points are on the upper left on the figure; which means they␣
↪should be paid more given the high winning
# percentage. Hence they have a high spending efficiency.
```

9 Part 3

10 Problem 5

[17]:
```
payroll_mean = team.groupby(team['yearID']).mean()
payroll_mean.columns = ['W', 'G', 'winning_percentage', 'mean_payroll']
payroll_std = team.groupby(team['yearID']).std()
payroll_std.columns = ['W', 'G', 'winning_percentage', 'std_payroll']
team['std_payroll'] = np.nan

for index, row in team.iterrows():
    team.at[index, 'std_payroll'] = \
                ((row['total_payroll'] -␣
↪payroll_mean['mean_payroll'][row["yearID"]]) /␣
↪payroll_std['std_payroll'][row["yearID"]])
team
```

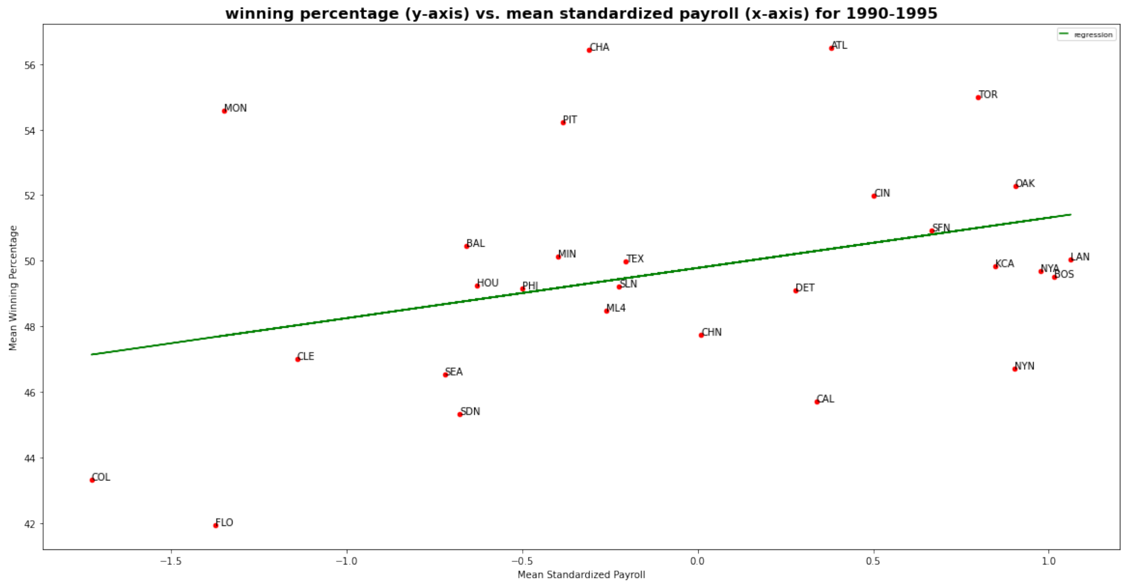
[17]:
```
     yearID teamID franchID   W    G  winning_percentage  total_payroll  \
0      1990    ATL      ATL  65  162           40.123457     14555501.0
1      1990    BAL      BAL  76  161           47.204969      9680084.0
2      1990    BOS      BOS  88  162           54.320988     20558333.0
3      1990    CAL      ANA  80  162           49.382716     21720000.0
4      1990    CHA      CHW  94  162           58.024691      9491500.0
..      …      …        …  ..  …                   …             …
723    2014    SLN      STL  90  162           55.555556    120693000.0
724    2014    TBA      TBD  77  162           47.530864     72689100.0
725    2014    TEX      TEX  67  162           41.358025    112255059.0
726    2014    TOR      TOR  83  162           51.234568    109920100.0
727    2014    WAS      WSN  96  162           59.259259    131983680.0

           Bin  std_payroll
0    1990-1995    -0.667275
1    1990-1995    -1.959861
2    1990-1995     0.924213
3    1990-1995     1.232198
4    1990-1995    -2.009859
..         …            …
723  2010-2015     0.457126
724  2010-2015    -0.593171
```

```
725  2010-2015      0.272509
726  2010-2015      0.221422
727  2010-2015      0.704160

[728 rows x 9 columns]
```

11 Problem 6

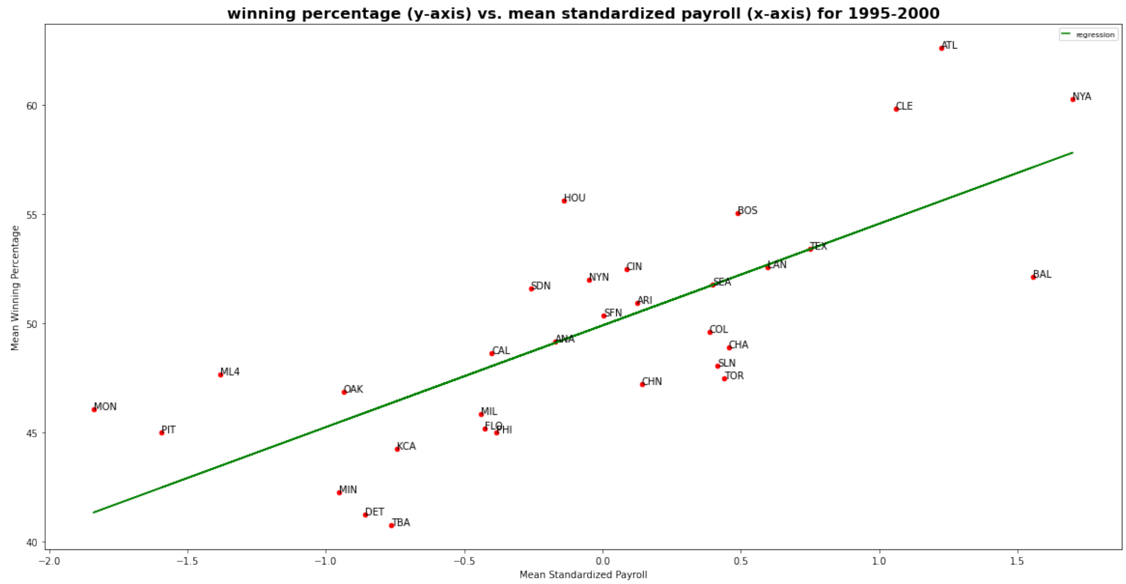
[18]:
```
team1990_std = team[team['Bin'] == '1990-1995']
team1995_std = team[team['Bin'] == '1995-2000']
team2000_std = team[team['Bin'] == '2000-2005']
team2005_std = team[team['Bin'] == '2005-2010']
team2010_std = team[team['Bin'] == '2010-2015']

# Year 1990-1995
team1990_std = team1990_std.groupby(['teamID']).mean()
team1990_std = team1990_std.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team1990_std['std_payroll'],team1990_std['winning_percentage'],1)
f = np.poly1d(d)
team1990_std.insert(2,'regression',f(team1990_std['std_payroll']))
team1990r_std = team1990_std[['std_payroll','regression']].copy()

team1990_ax = team1990_std.plot(x = 'std_payroll', y = 'winning_percentage',␣
 ↪kind='scatter', color = 'red')
for index, row in team1990_std.iterrows():
    team1990_ax.annotate(index, (row['std_payroll'], row['winning_percentage']))
team1990r_std.plot(x = 'std_payroll', y = 'regression', ax = team1990_ax, color␣
 ↪= 'green')

team1990_ax.set_ylabel("Mean Winning Percentage")
team1990_ax.set_xlabel("Mean Standardized Payroll")

plt.title("winning percentage (y-axis) vs. mean standardized payroll (x-axis)␣
 ↪for 1990-1995", size=16, weight='bold')
```

[18]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean standardized payroll
(x-axis) for 1990-1995')
```



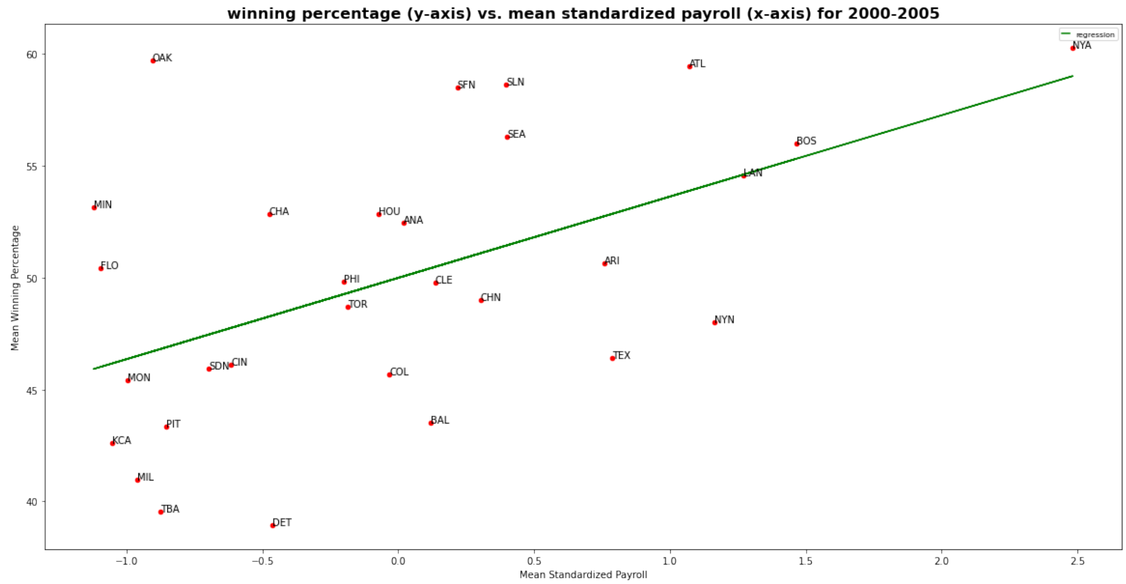
[19]:
```python
team1995_std = team1995_std.groupby(['teamID']).mean()
team1995_std = team1995_std.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team1995_std['std_payroll'],team1995_std['winning_percentage'],1)
f = np.poly1d(d)
team1995_std.insert(2,'regression',f(team1995_std['std_payroll']))
team1995r_std = team1995_std[['std_payroll','regression']].copy()

team1995_ax = team1995_std.plot(x = 'std_payroll', y = 'winning_percentage',␣
 ↪kind='scatter', color = 'red')
for index, row in team1995_std.iterrows():
    team1995_ax.annotate(index, (row['std_payroll'], row['winning_percentage']))
team1995r_std.plot(x = 'std_payroll', y = 'regression', ax = team1995_ax, color␣
 ↪= 'green')

team1995_ax.set_ylabel("Mean Winning Percentage")
team1995_ax.set_xlabel("Mean Standardized Payroll")

plt.title("winning percentage (y-axis) vs. mean standardized payroll (x-axis)␣
 ↪for 1995-2000", size=16, weight='bold')
```

[19]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean standardized payroll
(x-axis) for 1995-2000')
```



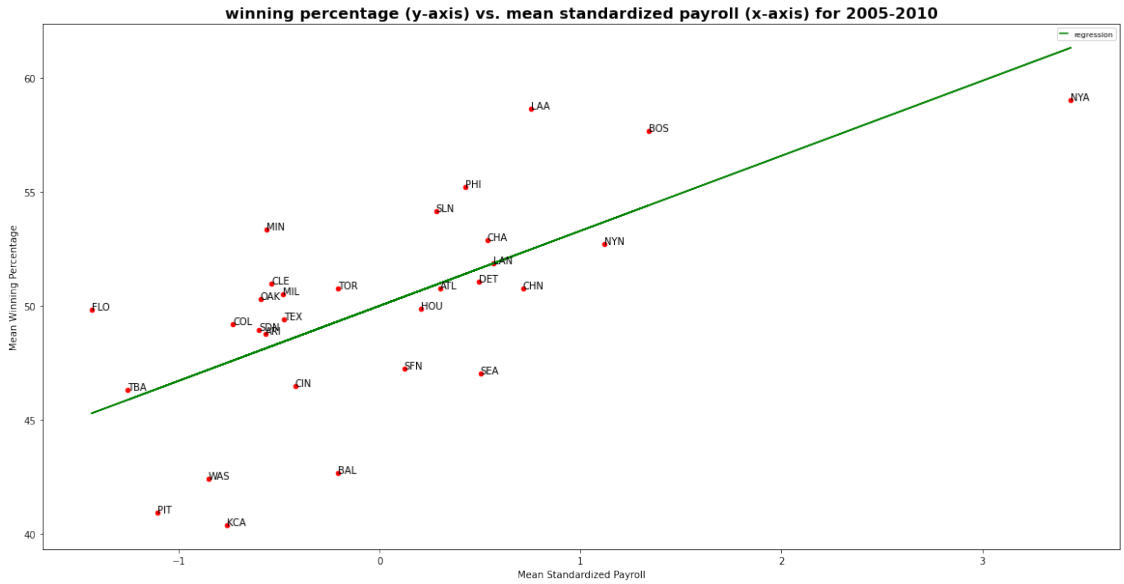
[20]:
```
team2000_std = team2000_std.groupby(['teamID']).mean()
team2000_std = team2000_std.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team2000_std['std_payroll'],team2000_std['winning_percentage'],1)
f = np.poly1d(d)
team2000_std.insert(2,'regression',f(team2000_std['std_payroll']))
team2000r_std = team2000_std[['std_payroll','regression']].copy()

team2000_ax = team2000_std.plot(x = 'std_payroll', y = 'winning_percentage',␣
↪kind='scatter', color = 'red')
for index, row in team2000_std.iterrows():
    team2000_ax.annotate(index, (row['std_payroll'], row['winning_percentage']))
team2000r_std.plot(x = 'std_payroll', y = 'regression', ax = team2000_ax, color␣
↪= 'green')

team2000_ax.set_ylabel("Mean Winning Percentage")
team2000_ax.set_xlabel("Mean Standardized Payroll")

plt.title("winning percentage (y-axis) vs. mean standardized payroll (x-axis)␣
↪for 2000-2005", size=16, weight='bold')
```

[20]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean standardized payroll
(x-axis) for 2000-2005')
```



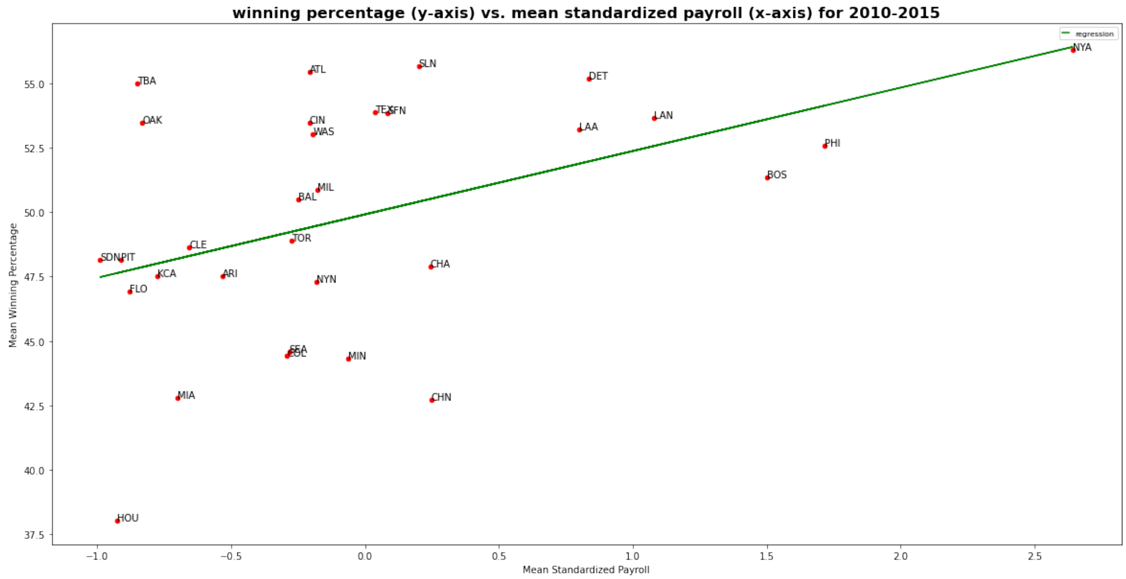
[21]:
```python
team2005_std = team2005_std.groupby(['teamID']).mean()
team2005_std = team2005_std.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team2005_std['std_payroll'],team2005_std['winning_percentage'],1)
f = np.poly1d(d)
team2005_std.insert(2,'regression',f(team2005_std['std_payroll']))
team2005r_std = team2005_std[['std_payroll','regression']].copy()

team2005_ax = team2005_std.plot(x = 'std_payroll', y = 'winning_percentage',␣
↪kind='scatter', color = 'red')
for index, row in team2005_std.iterrows():
    team2005_ax.annotate(index, (row['std_payroll'], row['winning_percentage']))
team2005r_std.plot(x = 'std_payroll', y = 'regression', ax = team2005_ax, color␣
↪= 'green')

team2005_ax.set_ylabel("Mean Winning Percentage")
team2005_ax.set_xlabel("Mean Standardized Payroll")

plt.title("winning percentage (y-axis) vs. mean standardized payroll (x-axis)␣
↪for 2005-2010", size=16, weight='bold')
```

[21]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean standardized payroll
(x-axis) for 2005-2010')
```



[22]:
```python
team2010_std = team2010_std.groupby(['teamID']).mean()
team2010_std = team2010_std.drop(['yearID','W','G'], axis=1)

d = np.polyfit(team2010_std['std_payroll'],team2010_std['winning_percentage'],1)
f = np.poly1d(d)
team2010_std.insert(2,'regression',f(team2010_std['std_payroll']))
team2010r_std = team2010_std[['std_payroll','regression']].copy()

team2010_ax = team2010_std.plot(x = 'std_payroll', y = 'winning_percentage',␣
 ↪kind='scatter', color = 'red')
for index, row in team2010_std.iterrows():
    team2010_ax.annotate(index, (row['std_payroll'], row['winning_percentage']))
team2010r_std.plot(x = 'std_payroll', y = 'regression', ax = team2010_ax, color␣
 ↪= 'green')

team2010_ax.set_ylabel("Mean Winning Percentage")
team2010_ax.set_xlabel("Mean Standardized Payroll")

plt.title("winning percentage (y-axis) vs. mean standardized payroll (x-axis)␣
 ↪for 2010-2015", size=16, weight='bold')
```

[22]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. mean standardized payroll
(x-axis) for 2010-2015')
```



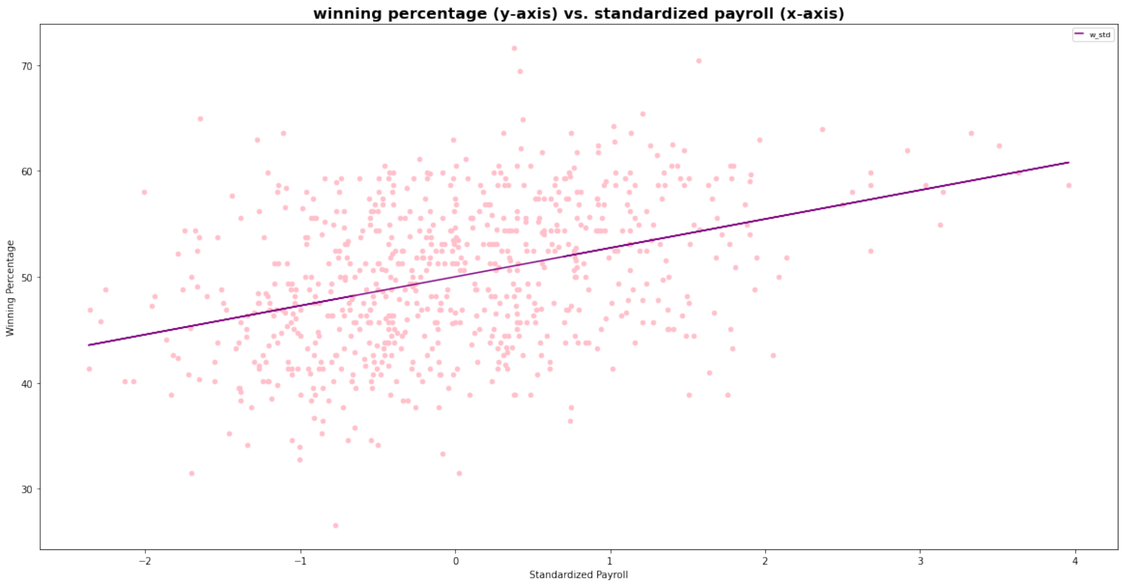
12 Question 3

[23]:
```
# By doing the transformation in problem 6, we make the standard deviation 1 and the mean 0. This helps to normalized the graph.
```

13 Problem 7

[24]:
```
d = np.polyfit(team['std_payroll'],team['winning_percentage'],1)
f = np.poly1d(d)
team.insert(9,'w_std',f(team['std_payroll']))
team_std = team[['std_payroll','w_std']].copy()

team_ax = team.plot(x = 'std_payroll', y = 'winning_percentage', kind='scatter', color = 'pink')

team.plot(x = 'std_payroll', y = 'w_std', ax = team_ax, color = 'purple')

team_ax.set_ylabel("Winning Percentage")
team_ax.set_xlabel("Standardized Payroll")

plt.title("winning percentage (y-axis) vs. standardized payroll (x-axis)", size=16, weight='bold')
```

[24]:
```
Text(0.5, 1.0, 'winning percentage (y-axis) vs. standardized payroll (x-axis)')
```



14 Problem 8

[25]:
```python
team['expected_win_pct'] = np.nan
team['efficiency'] = np.nan

for index, row in team.iterrows():
    team.at[index, 'expected_win_pct'] = row['std_payroll']*2.5+50

for index, row in team.iterrows():
    team.at[index, 'efficiency'] = row['winning_percentage'] -␣
 ↪row['expected_win_pct']
team
```

[25]:
```
     yearID teamID franchID   W    G  winning_percentage  total_payroll  \
0      1990    ATL      ATL  65  162           40.123457     14555501.0
1      1990    BAL      BAL  76  161           47.204969      9680084.0
2      1990    BOS      BOS  88  162           54.320988     20558333.0
3      1990    CAL      ANA  80  162           49.382716     21720000.0
4      1990    CHA      CHW  94  162           58.024691      9491500.0
..      …      …        …  ..  …                   …              …
723    2014    SLN      STL  90  162           55.555556    120693000.0
724    2014    TBA      TBD  77  162           47.530864     72689100.0
725    2014    TEX      TEX  67  162           41.358025    112255059.0
726    2014    TOR      TOR  83  162           51.234568    109920100.0
727    2014    WAS      WSN  96  162           59.259259    131983680.0

           Bin  std_payroll      w_std  expected_win_pct  efficiency
```

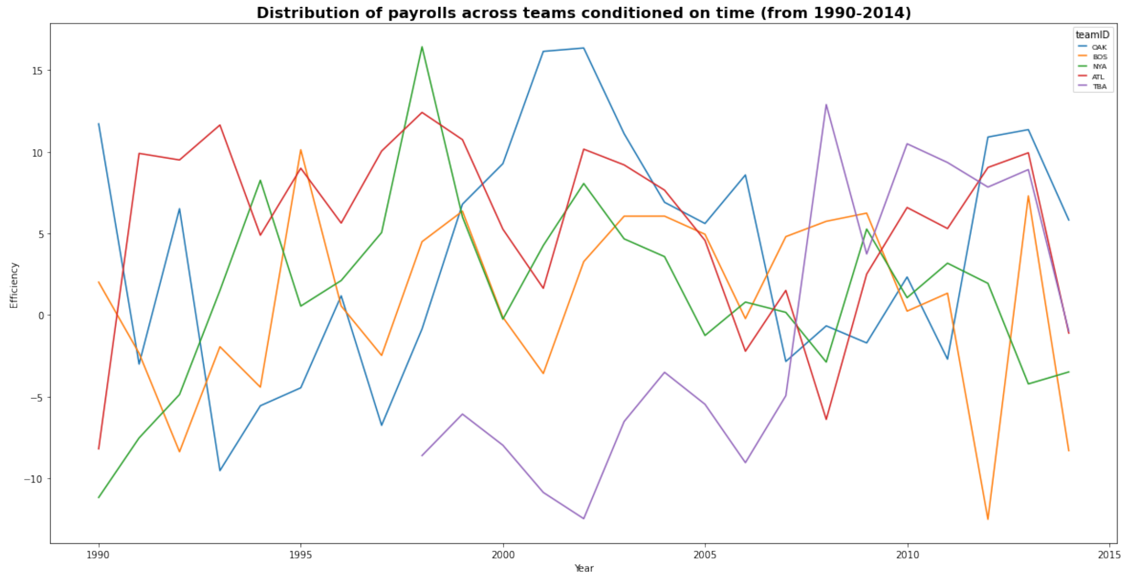
```
0     1990-1995    -0.667275  48.170158         48.331811   -8.208354
1     1990-1995    -1.959861  44.647730         45.100348    2.104621
2     1990-1995     0.924213  52.507130         52.310533    2.010454
3     1990-1995     1.232198  53.346420         53.080495   -3.697779
4     1990-1995    -2.009859  44.511480         44.975353   13.049338
..          …           …         …               …         …
723   2010-2015     0.457126  51.234270         51.142816    4.412740
724   2010-2015    -0.593171  48.372100         48.517072   -0.986208
725   2010-2015     0.272509  50.731169         50.681273   -9.323248
726   2010-2015     0.221422  50.591950         50.553554    0.681014
727   2010-2015     0.704160  51.907462         51.760400    7.498860

[728 rows x 12 columns]
```

[26]:

```
distribution = team.pivot(index='yearID', columns='teamID', values='efficiency')

distribution = distribution.filter([ 'OAK', 'BOS', 'NYA', 'ATL', 'TBA'])

plt.rcParams["figure.figsize"] = [20,10]
params = {'legend.fontsize': 7.8,
          'legend.handlelength': 1}
plt.rcParams.update(params)

ax = distribution.plot()

ax.set_xlabel("Year")
ax.set_ylabel("Efficiency")
plt.title('Distribution of payrolls across teams conditioned on time (from␣
 ↪1990-2014)', size=16, weight='bold')
```

[26]:

```
Text(0.5, 1.0, 'Distribution of payrolls across teams conditioned on time (from
1990-2014)')
```



15 Question 4

[27]:

```
# Base on the graph of Question 2, 3 ,and 4, we see that the salary is not the only factor that influence performance. Oakland's
# performance fluctuate during the Moneyball period. Its performance increases in about 1995 and reach its peak between 2000
# and 2005.
```