Part 1: Wrangling

In [ ]:
```
import sqlite3
import pandas as pd
import numpy as np
```

Problem 1

In [ ]:
```
# Import sqlite file from drive
sqlite_file = '/content/drive/MyDrive/Colab Notebooks/CMSC320/Project2/lahman2(
conn = sqlite3.connect(sqlite_file)

# Total payroll for each team
salary_query = "SELECT yearID, teamID, sum(salary) as total_payroll FROM Salari
team_salaries = pd.read_sql(salary_query, conn)

# Winning percentage for each team
winning_query = "SELECT yearID, teamID, franchID as franchiseID, W AS num_wins,
team_winning = pd.read_sql(winning_query, conn)

# Check missing values
print("The year range of team_salaries table is from %d to %d." %(team_salaries
print("The year range of team_winning table is from %d to %d." %(team_winning['

print("There are %d unique teams in the team_salaries table." %team_salaries['t
print("There are %d unique teams in the team_winning table." %team_winning['tea

# merge two tables
op = team_salaries.merge(team_winning, how = "inner")
op
```

```
The year range of team_salaries table is from 1985 to 2014.
The year range of team_winning table is from 1871 to 2014.
There are 37 unique teams in the team_salaries table.
There are 149 unique teams in the team_winning table.
```

Out[ ]:

	yearID	teamID	total_payroll	franchiseID	num_wins	num_games	winning_percentage
0	1985	ATL	14807000.0	ATL	66	162	40.740741
1	1985	BAL	11560712.0	BAL	83	161	51.552795
2	1985	BOS	10897560.0	BOS	81	163	49.693252
3	1985	CAL	14427894.0	ANA	90	162	55.555556
4	1985	CHA	9846178.0	CHW	85	163	52.147239
...	...	...	...	...	...	...	...
853	2014	SLN	120693000.0	STL	90	162	55.555556
854	2014	TBA	72689100.0	TBD	77	162	47.530864
855	2014	TEX	112255059.0	TEX	67	162	41.358025
856	2014	TOR	109920100.0	TOR	83	162	51.234568
857	2014	WAS	131983680.0	WSN	96	162	59.259259

858 rows × 7 columns

It appears that the time range and team numbers in the two tables are not the same. When we merged the two tables, we only use yearID and teamID to included the matching data and omitted the unmatched data. The resulting merged table has 858 rows, which is less than either of the original tables, indicating that both tables are missing some data.

To perform the merge, we used an inner merge based on the intersection of the columns in both tables. The output of the merge is shown above.

Part 2: Exploratory Data Analysis

Payroll Distribution

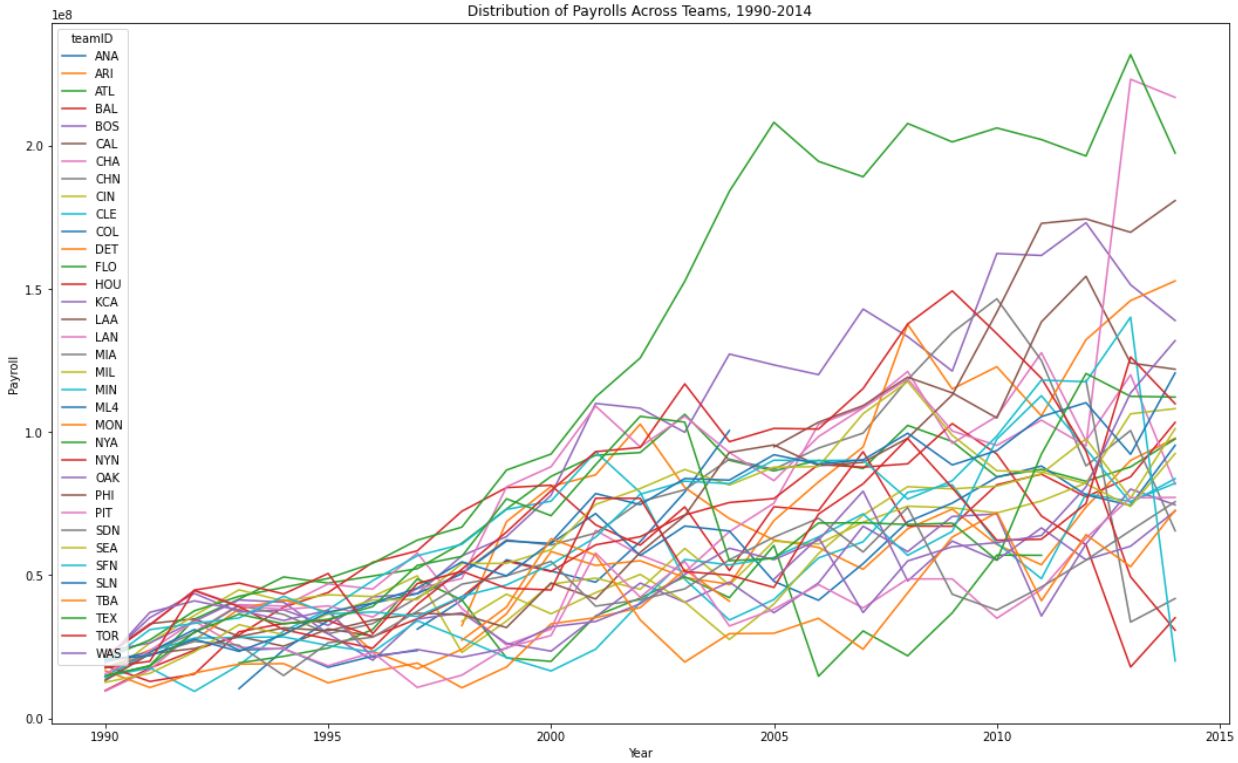
Problem 2

In [ ]:

```
import matplotlib.pyplot as plt

# Create a new dataframe that only contains rows with yearID between 1990 and 2
op_new = op[op['yearID'] >= 1990]

# Format the data
op_new_pivot = pd.pivot(op_new, index = 'yearID', columns = 'teamID', values =

# Plot
op_new_pivot.plot(figsize=(18,11))
plt.xlabel('Year')
plt.ylabel('Payroll')
plt.title('Distribution of Payrolls Across Teams, 1990-2014')
```

Out[ ]:

```
Text(0.5, 1.0, 'Distribution of Payrolls Across Teams, 1990-2014')
```



Question 1

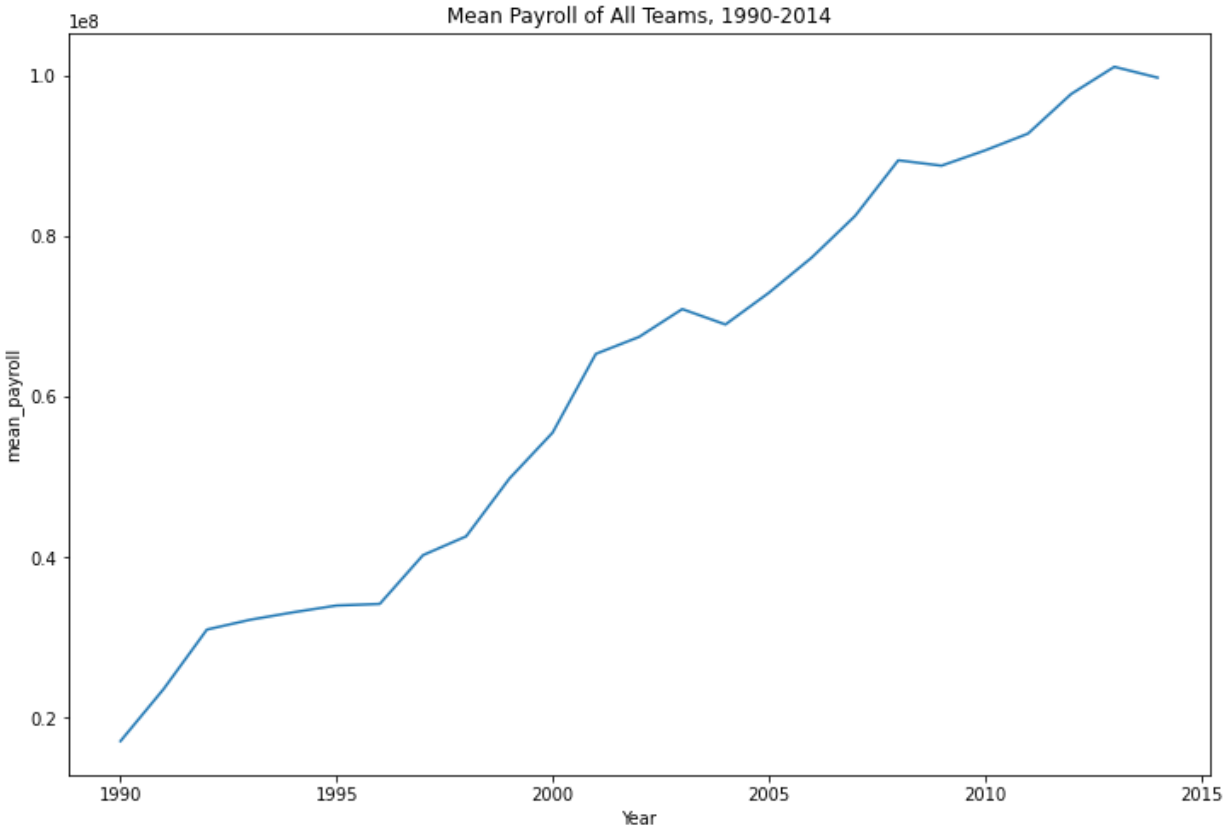
Based on the plot, we can conclude that the trend of total payroll for each team is increasing over time, which means they are paying more to hire players. Also, the difference of total payroll between teams is getting larger over time, this likely indicates a growing disparity in financial resources across teams.

Problem 3

Using mean payroll of teams to show the trend of total payroll increasing.

In [ ]:
```
# Calculate the mean payroll of each year
op_new_pivot['mean_payroll'] = op_new_pivot.mean(axis=1)

# Plot mean payroll vs. time
op_new_pivot['mean_payroll'].plot(figsize=(12,8))
plt.xlabel('Year')
plt.ylabel('mean_payroll')
plt.title('Mean Payroll of All Teams, 1990-2014')
```

Out[ ]:
```
Text(0.5, 1.0, 'Mean Payroll of All Teams, 1990-2014')
```

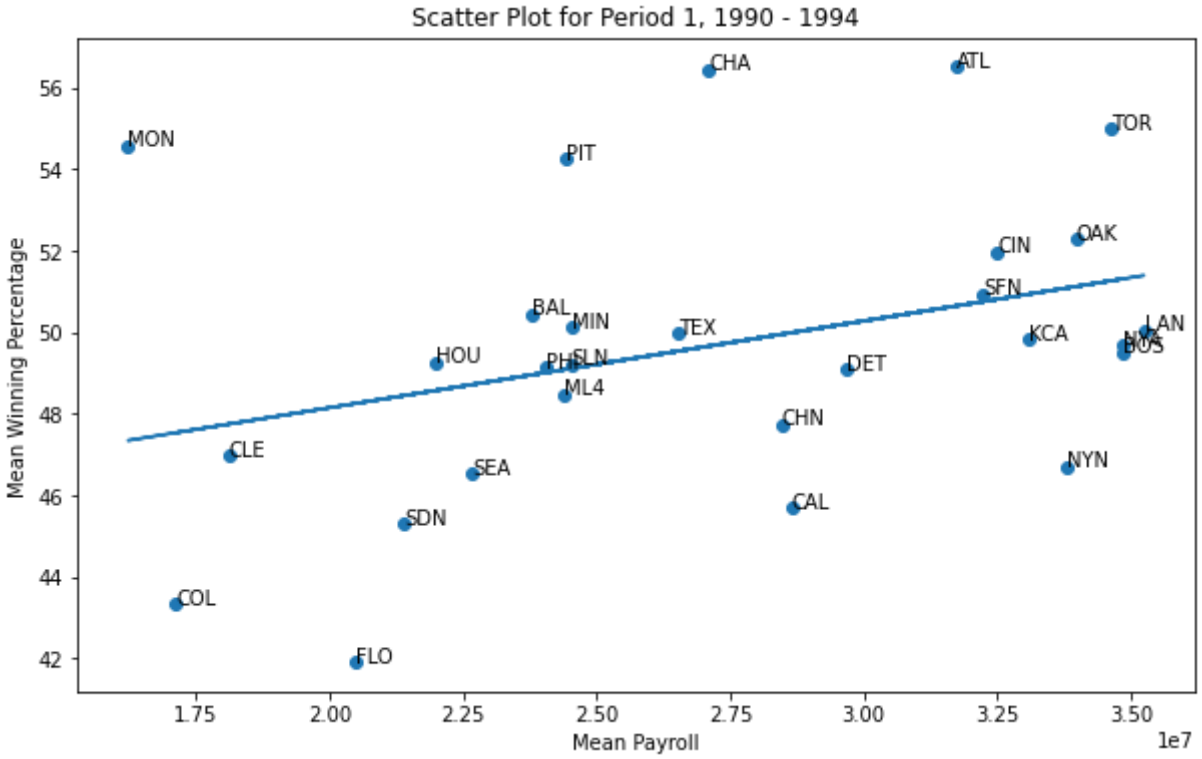


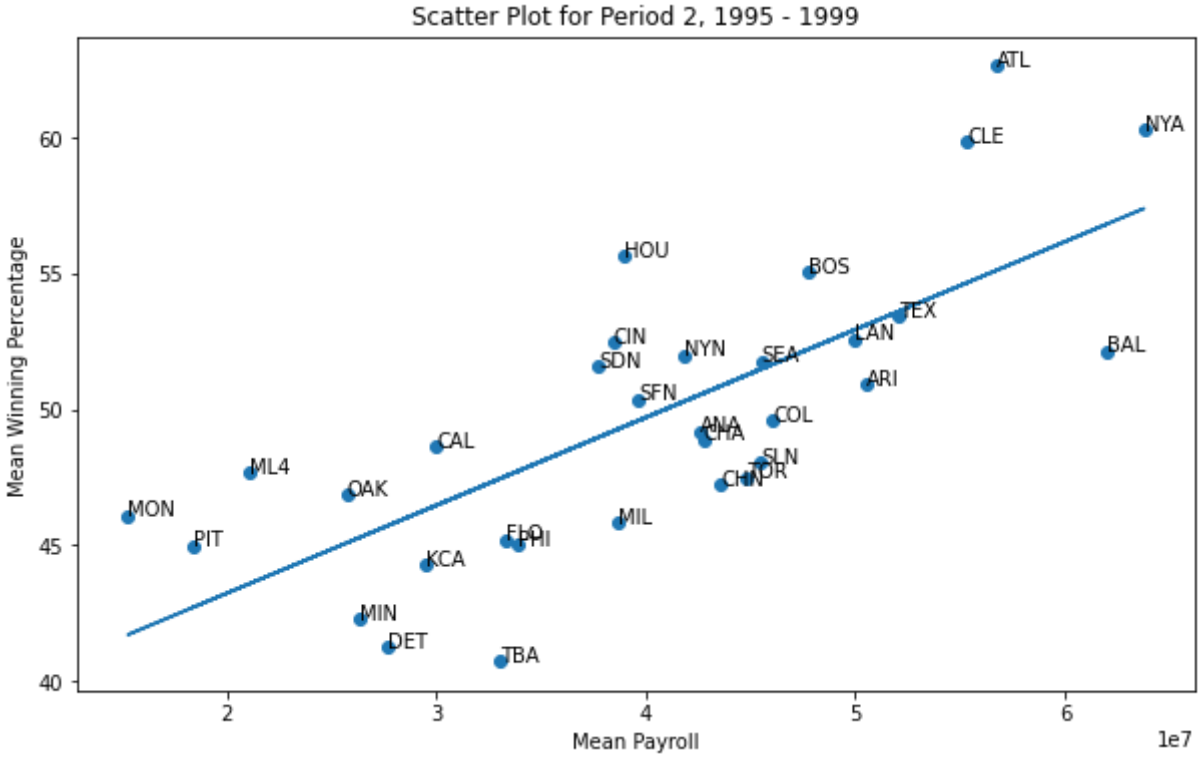
Correlation between payroll and winning percentage

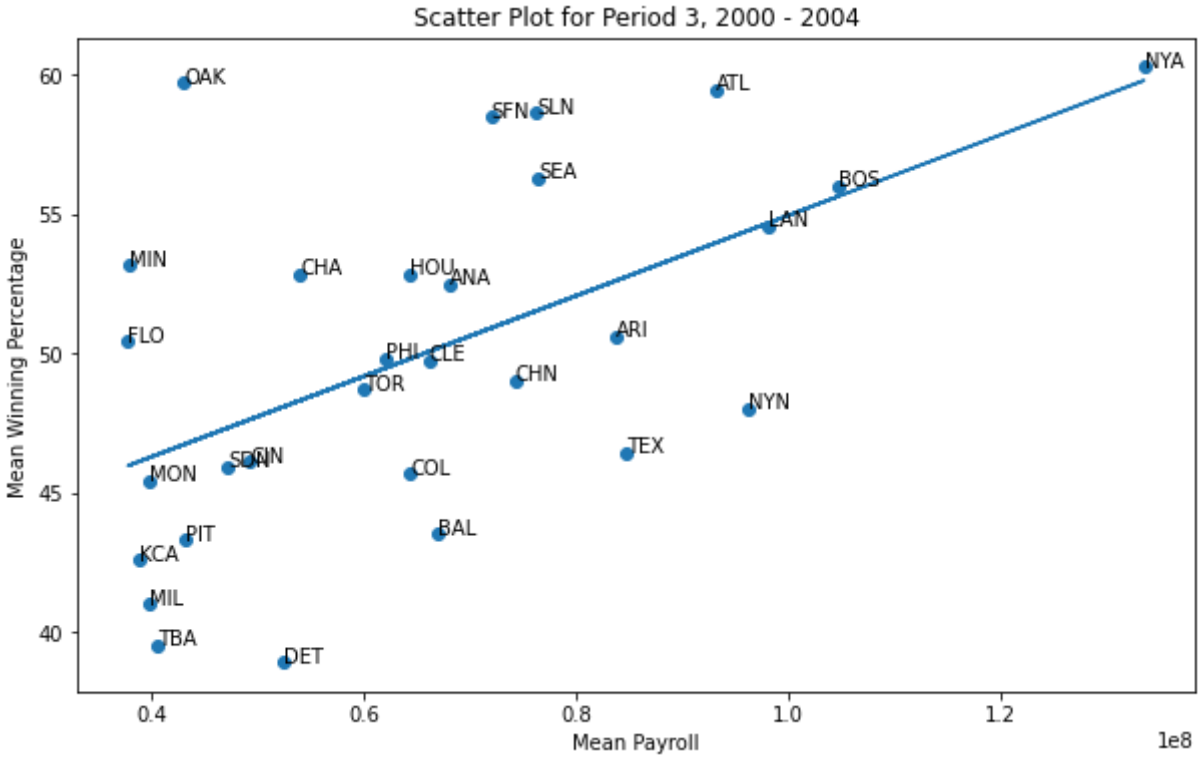
Problem 4

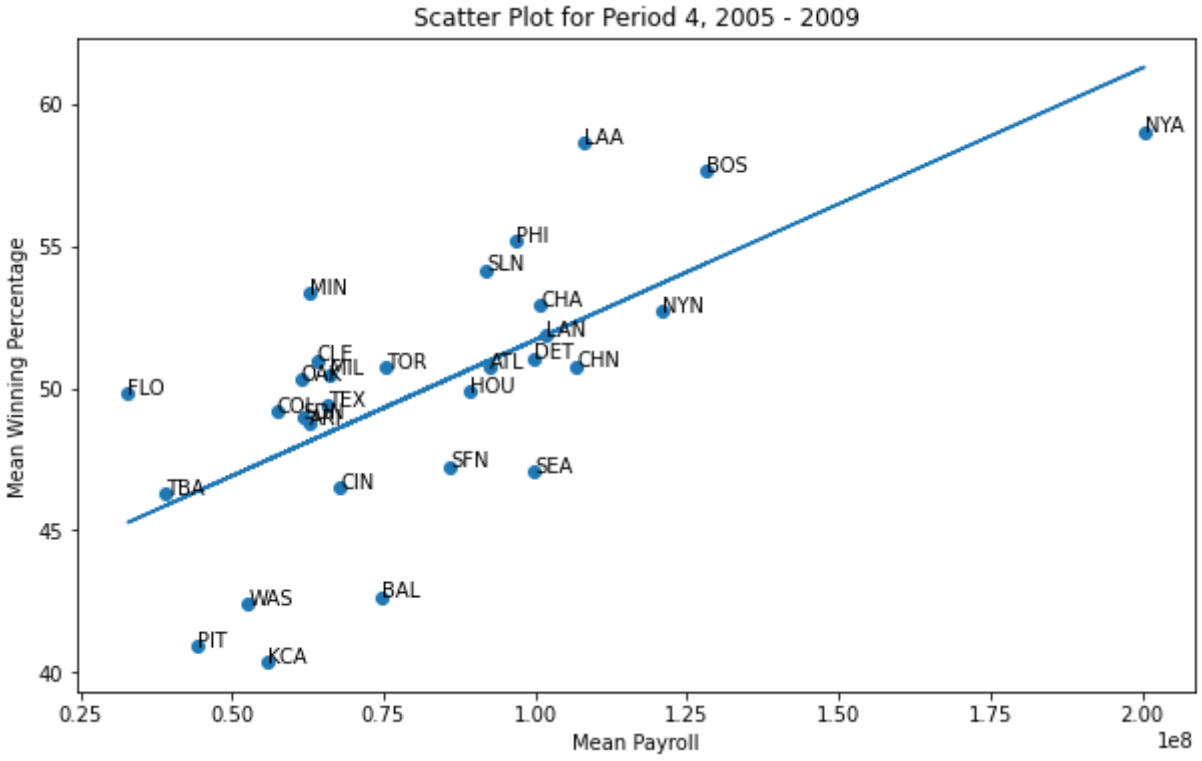
In [ ]:

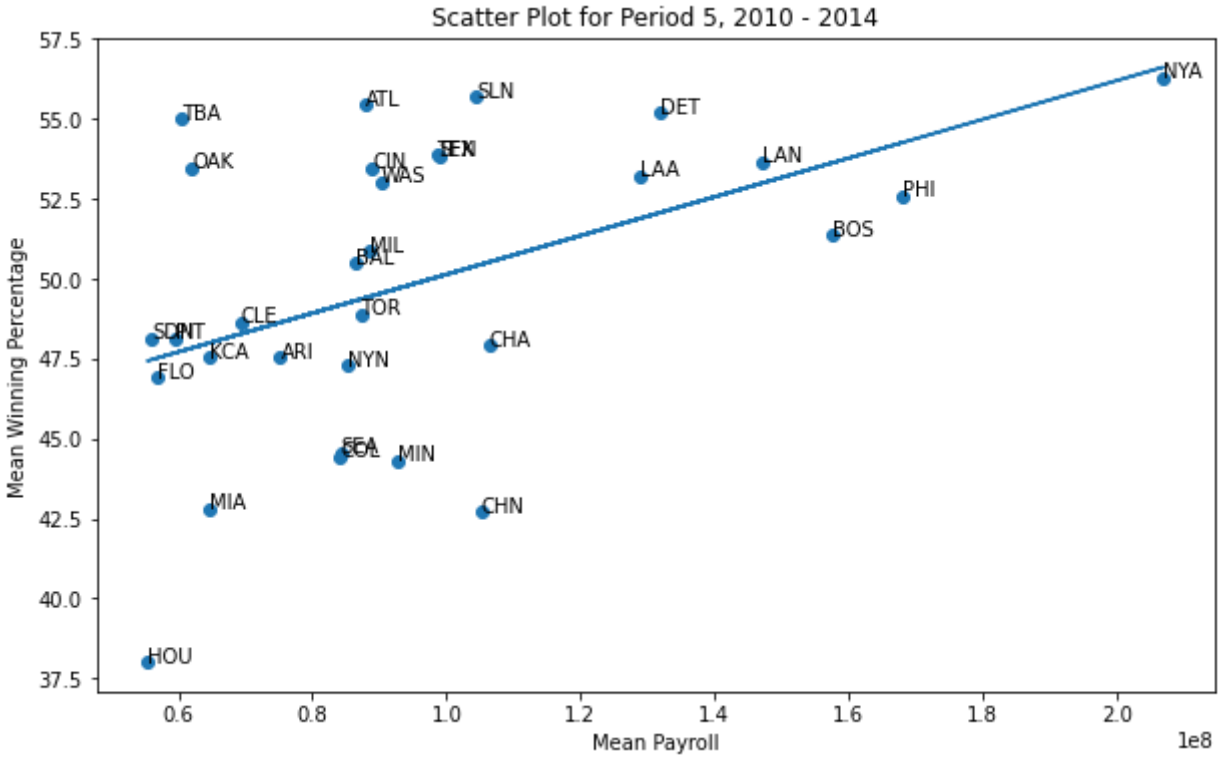
```
# Discretize year into five time periods, and make a new column to indicate whi
op_new = op[op['yearID'] >= 1990].copy()
op_new['period'] = pd.cut(op_new['yearID'], bins = 5, labels = ['Period 1', 'Pe

# Plot data for each period
for period in ['Period 1', 'Period 2', 'Period 3', 'Period 4', 'Period 5']:

  subset = op_new[op_new['period'] == period]

  # Compute mean winning percentage and mean payroll for each team at such peri
  mean_winning = subset.groupby(['teamID'])['winning_percentage'].mean()
  mean_payroll = subset.groupby(['teamID'])['total_payroll'].mean()

  # Plot the scatterplot with regression line
  plt.figure(figsize=(10,6))
  plt.scatter(mean_payroll, mean_winning)
  p = np.polyfit(mean_payroll, mean_winning, 1)
  plt.plot(mean_payroll, np.polyval(p, mean_payroll))
  plt.title('Scatter Plot for %s, %d - %d' %(period, subset['yearID'].min(), su
  plt.xlabel('Mean Payroll')
  plt.ylabel('Mean Winning Percentage')

  # Label points in the plot
  for team, x, y in zip(mean_payroll.index, mean_payroll, mean_winning):
    plt.text(x, y, team)

  plt.show()
```


Scatter Plot for Period 1, 1990 - 1994
Mean Winning Percentage
Mean Payroll
1e7
CHA
ATL
TOR
MON
PIT
CIN
OAK
SFN
BAL
MIN
TEX
LAN
KCA
HOU
PHI
SLN
DET
ML4
CHN
CLE
SEA
NYN
CAL
SDN
COL
FLO


Scatter Plot for Period 2, 1995 - 1999
Mean Winning Percentage
Mean Payroll
1e7
ATL
NYA
CLE
HOU
BOS
TEX
CIN
LAN
BAL
NYN
SEA
SDN
ARI
SFN
COL
ANA
CHA
CAL
SLN
ML4
CHN
OAK
MON
MIL
PIT
PHI
KCA
MIN
DET
TBA

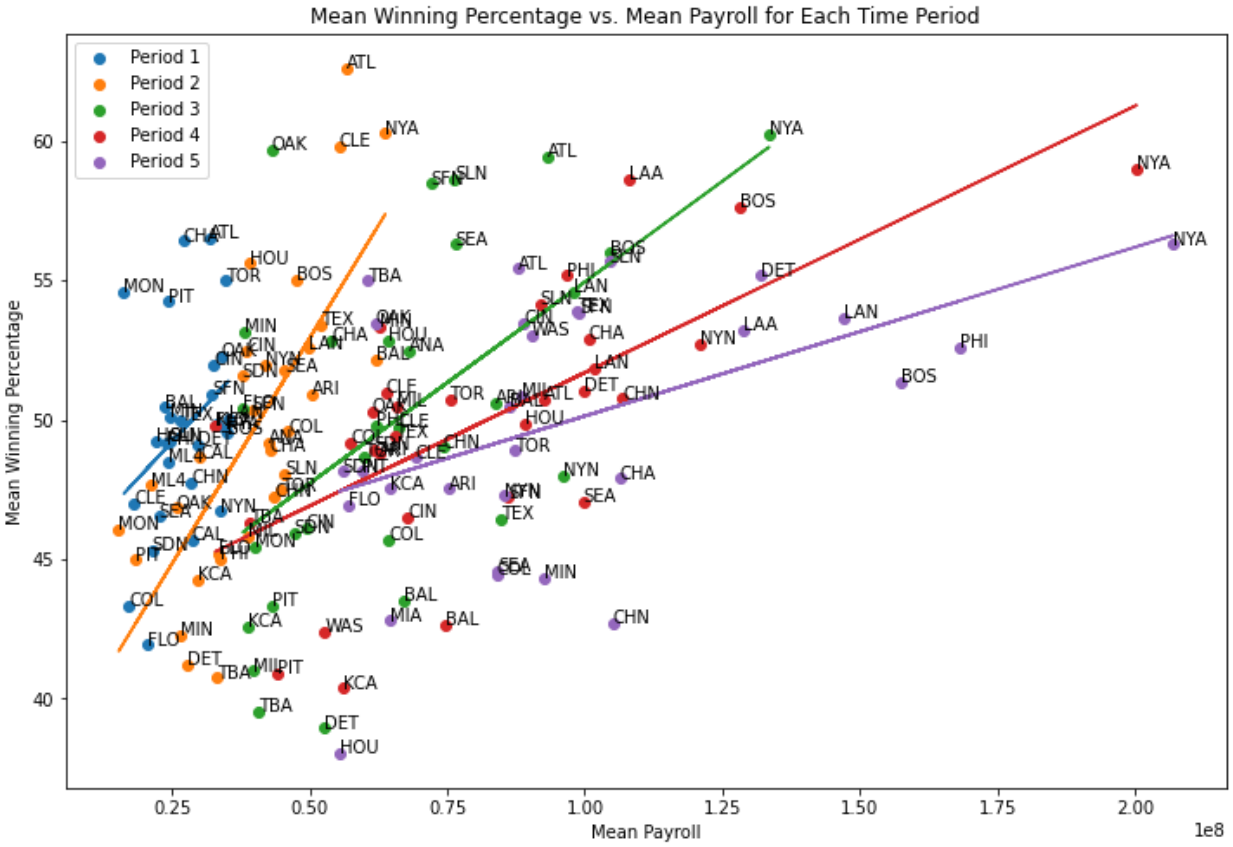

Scatter Plot for Period 3, 2000 - 2004
Mean Winning Percentage
Mean Payroll
1e8
NYA
OAK
ATL
SFN
SLN
SEA
BOS
LAN
MIN
CHA
HOU
ANA
FLO
ARI
PHI
CLE
CHN
TOR
NYN
TEX
SDN
CIN
MON
COL
PIT
BAL
KCA
MIL
TBA
DET


Scatter Plot for Period 4, 2005 - 2009
Mean Winning Percentage
Mean Payroll
1e8
NYA
LAA
BOS
PHI
SLN
MIN
CHA
NYN
LAN
CLE
DET
CHN
TOR
ATL
OAK
MIL
FLO
HOU
TEX
COL
SFN
SEA
TBA
CIN
WAS
BAL
PIT
KCA



In [ ]:
```
# Plot all data on one figure
plt.figure(figsize=(12,8))
for period in ['Period 1', 'Period 2', 'Period 3', 'Period 4', 'Period 5']:

  subset = op_new[op_new['period'] == period]

  # Compute mean winning percentage and mean payroll for each team at such peri
  mean_winning = subset.groupby(['teamID'])['winning_percentage'].mean()
  mean_payroll = subset.groupby(['teamID'])['total_payroll'].mean()

  # Plot the scatterplot with regression line
  plt.scatter(mean_payroll, mean_winning, label=period)
  p = np.polyfit(mean_payroll, mean_winning, 1)
  plt.plot(mean_payroll, np.polyval(p, mean_payroll))

  # Label points in the plot
  for team, x, y in zip(mean_payroll.index, mean_payroll, mean_winning):
    plt.text(x, y, team)

plt.xlabel('Mean Payroll')
plt.ylabel('Mean Winning Percentage')
plt.title('Mean Winning Percentage vs. Mean Payroll for Each Time Period')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f077cd63e20>
```



Question 2

It is clear that higher mean payroll means higher mean winning percentage regardless of time period, as the regression line of each plot clearly shows this.

According to the last figure we plotted (combining all data into one plot), we can identify some excellent teams such as ATL and NYA. Both teams had high winning percentages in period 2 (1995 - 1999), and their winning percentages are above the regression line for most periods, indicating that they were good at paying for wins across these time periods.

In terms of Oakland A's (OAK) spending efficiency across these time periods, we can see that for the later time periods (period 3 - period 5), they achieved high winning percentages despite spending significantly less than other teams. This suggests that they serve as a good example of high spending efficiency.

Part 3: Data transformations

Standardizing across years

Problem 5

In [ ]:
```
op_new = op_new.copy()

# Calculate avg_payroll_j and S_j for each year
avg_payroll_j = op_new.groupby('yearID')['total_payroll'].mean()
```

```
S_j = op_new.groupby('yearID')['total_payroll'].std()

# Create new column
op_new['standardized_payroll'] = (op_new['total_payroll'] - op_new['yearID'].ma
op_new
```

Out[ ]:

	yearID	teamID	total_payroll	franchiseID	num_wins	num_games	winning_percentage	p
130	1990	ATL	14555501.0	ATL	65	162	40.123457	
131	1990	BAL	9680084.0	BAL	76	161	47.204969	
132	1990	BOS	20558333.0	BOS	88	162	54.320988	
133	1990	CAL	21720000.0	ANA	80	162	49.382716	
134	1990	CHA	9491500.0	CHW	94	162	58.024691	
...	...	...	...	...	...	...	...	
853	2014	SLN	120693000.0	STL	90	162	55.555556	
854	2014	TBA	72689100.0	TBD	77	162	47.530864	
855	2014	TEX	112255059.0	TEX	67	162	41.358025	
856	2014	TOR	109920100.0	TOR	83	162	51.234568	
857	2014	WAS	131983680.0	WSN	96	162	59.259259	

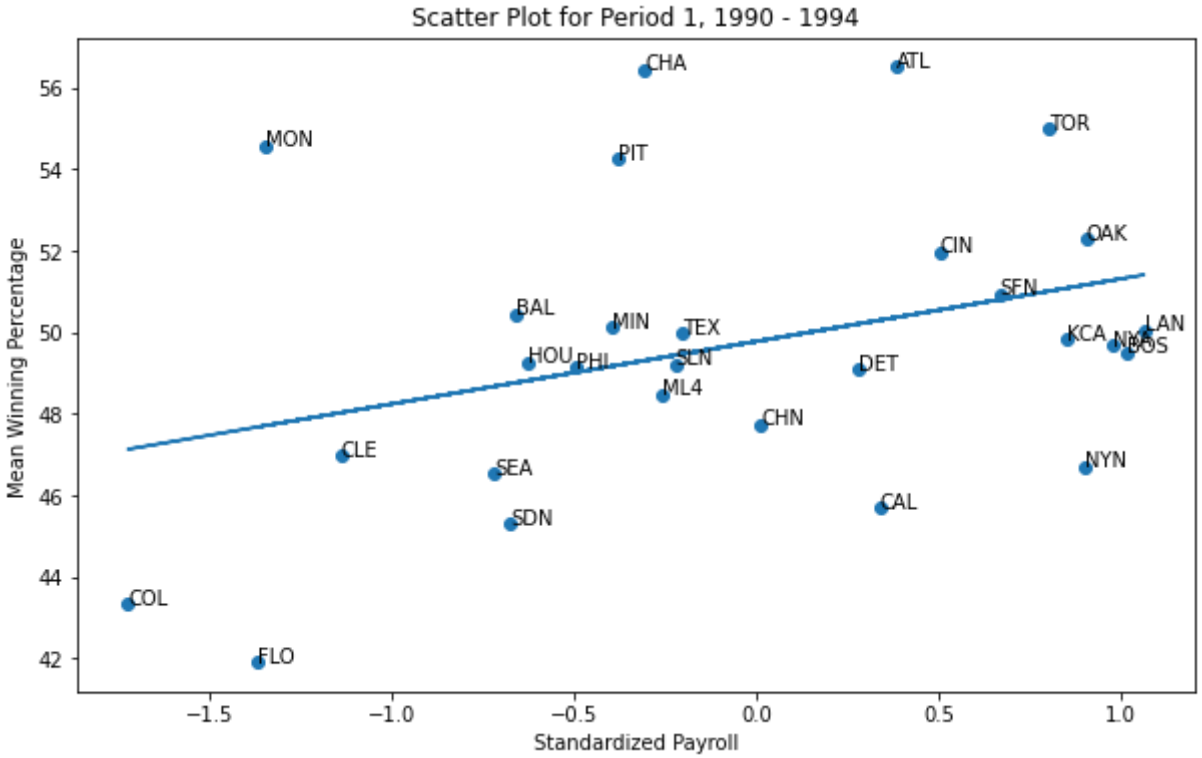
728 rows × 9 columns

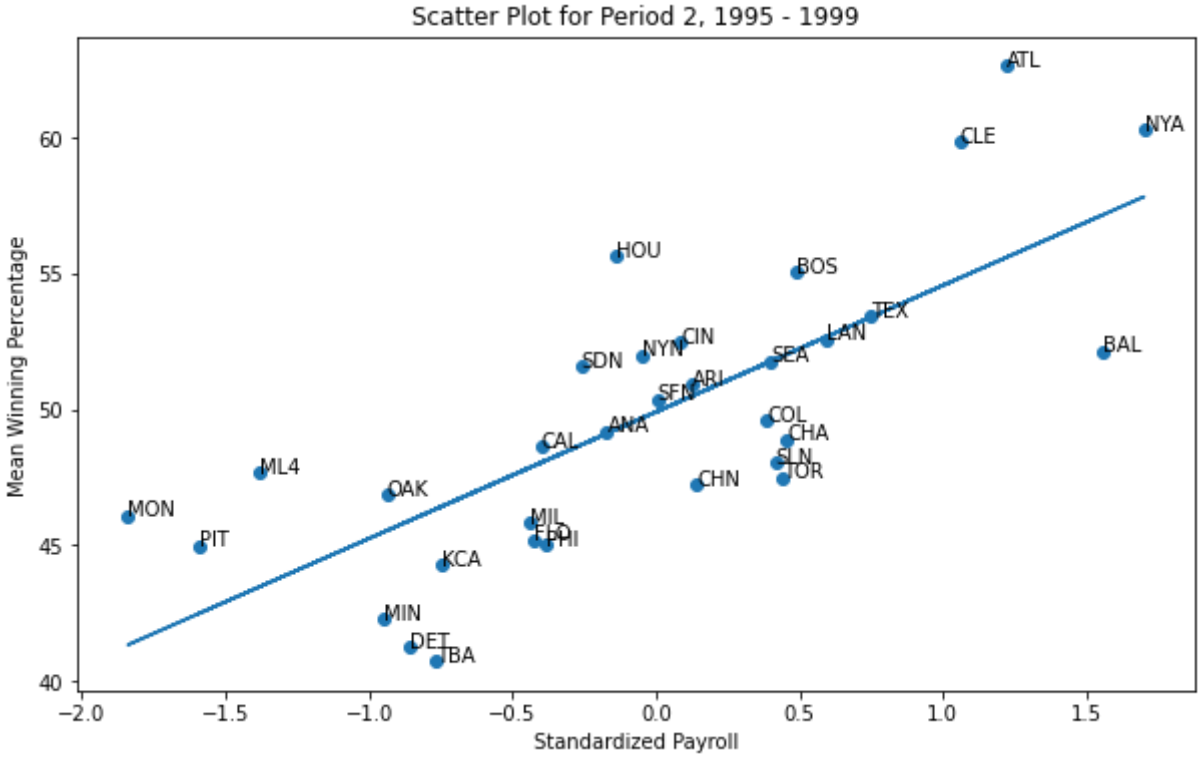
Problem 6

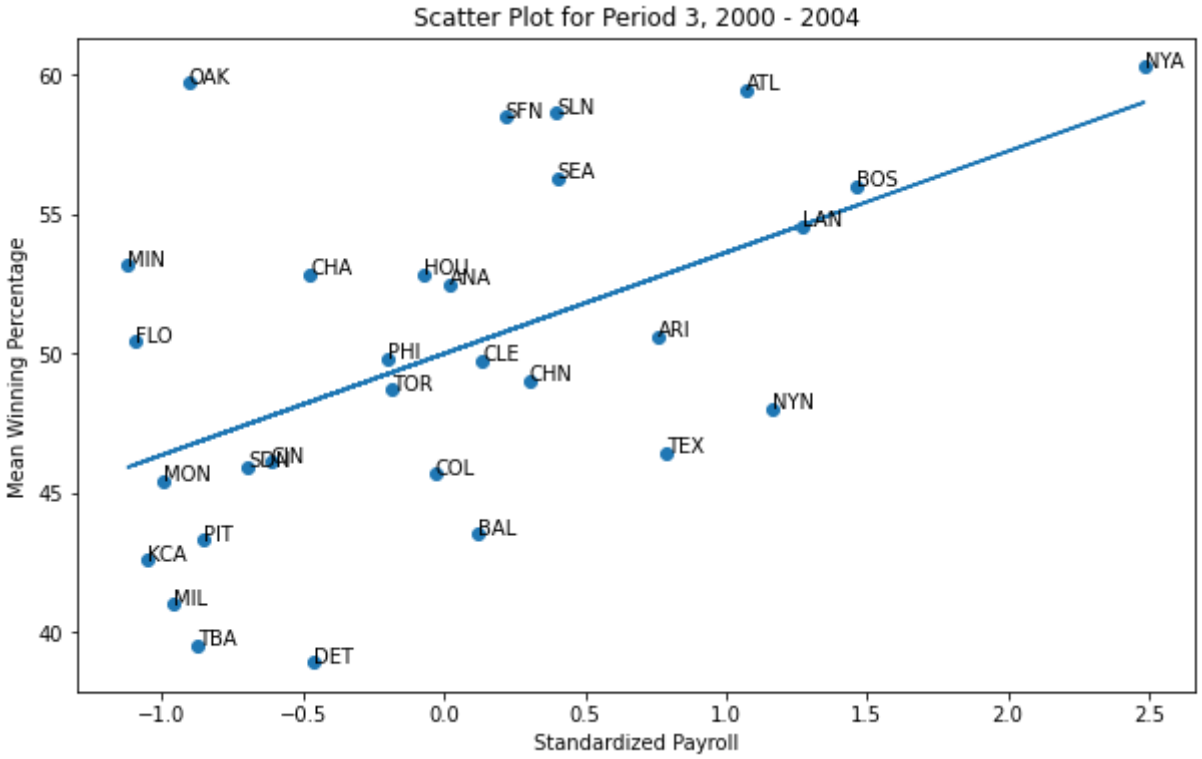
In [ ]:

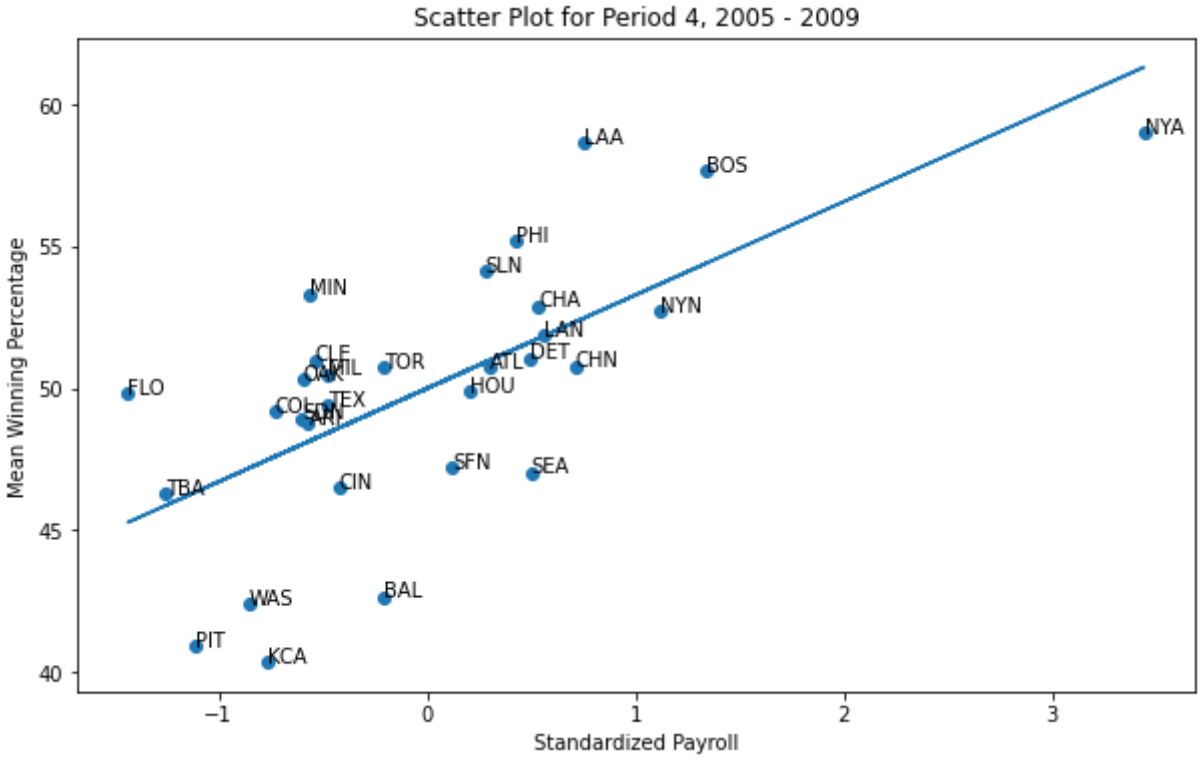
```
# Plot data for each period
for period in ['Period 1', 'Period 2', 'Period 3', 'Period 4', 'Period 5']:

  subset = op_new[op_new['period'] == period]

  # Compute mean winning percentage and mean standardized payroll for each tean
  mean_winning = subset.groupby(['teamID'])['winning_percentage'].mean()
  mean_payroll = subset.groupby(['teamID'])['standardized_payroll'].mean()

  # Plot the scatterplot with regression line
  plt.figure(figsize=(10,6))
  plt.scatter(mean_payroll, mean_winning)
  p = np.polyfit(mean_payroll, mean_winning, 1)
  plt.plot(mean_payroll, np.polyval(p, mean_payroll))
  plt.title('Scatter Plot for %s, %d - %d' %(period, subset['yearID'].min(), su
  plt.xlabel('Standardized Payroll')
  plt.ylabel('Mean Winning Percentage')
```

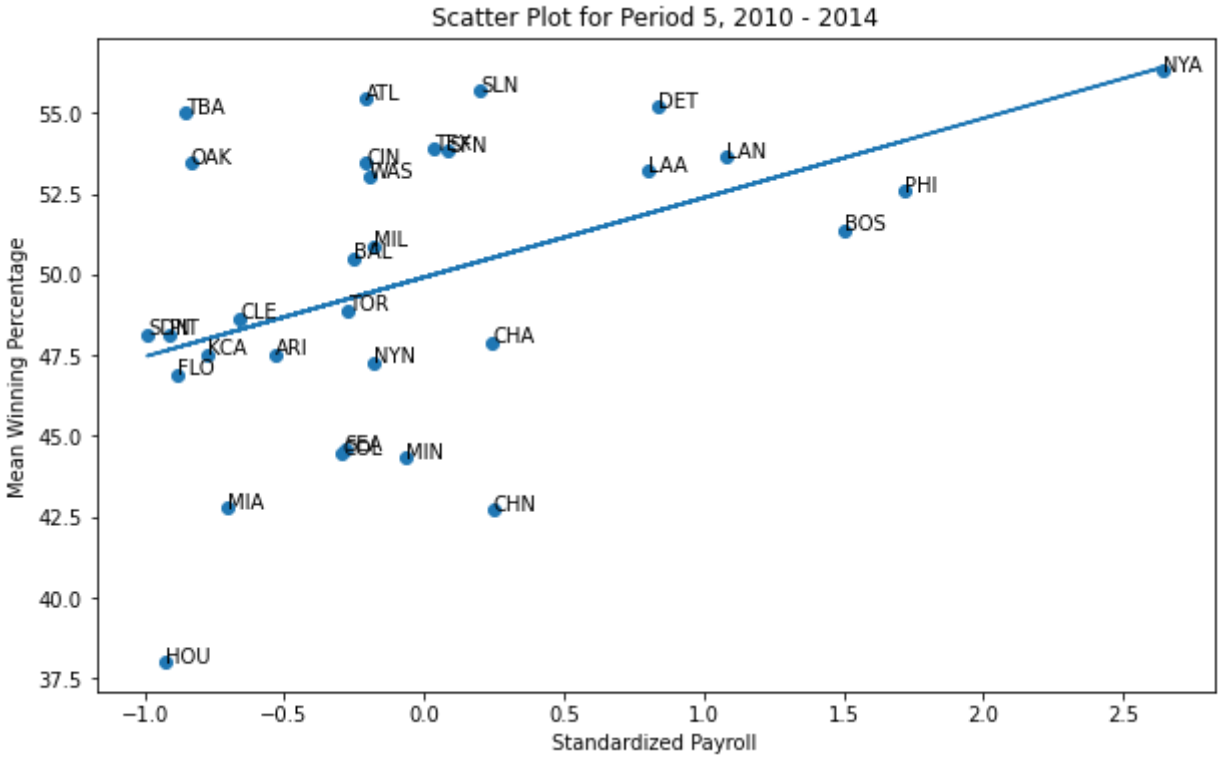
```
  # Label points in the plot
  for team, x, y in zip(mean_payroll.index, mean_payroll, mean_winning):
    plt.text(x, y, team)

  plt.show()
```






Scatter Plot for Period 3, 2000 - 2004
Mean Winning Percentage
Standardized Payroll
OAK
ATL
NYA
SFN
SLN
SEA
BOS
LAN
MIN
CHA
HOU
ANA
FLO
ARI
PHI
CLE
CHN
TOR
NYN
TEX
SDN
CIN
MON
COL
PIT
BAL
KCA
MIL
TBA
DET


Scatter Plot for Period 4, 2005 - 2009
Mean Winning Percentage
Standardized Payroll
LAA
NYA
BOS
PHI
SLN
MIN
CHA
NYN
LAN
CLE
DET
CHN
TOR
ATL
OAK
MIL
FLO
HOU
TEX
COL
SFN
SEA
TBA
CIN
WAS
BAL
PIT
KCA



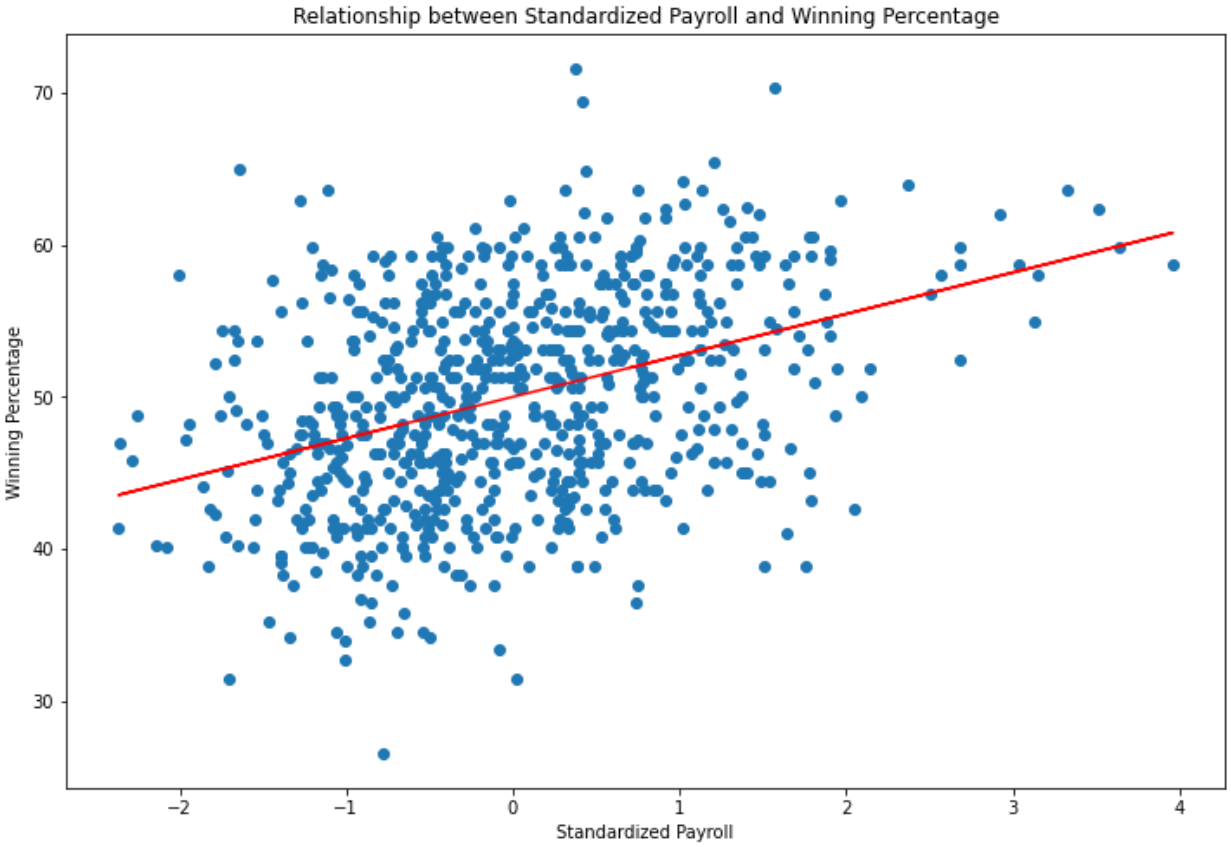
Question 3

The plots we derived from Question 4 and Question 6 are nearly identical in increasing trend, and the scatter of the data is similar in each time period. However, the transformation we applied in Question 6 made it easier to compare teams' spending efficiency across time periods. By standardizing the payroll variable, we can observe how many standard deviations each team's payroll is from the average payroll. This allows us to identify teams that were spending efficiently even with a lower payroll than other teams in the same time period.

Expected wins

Problem 7

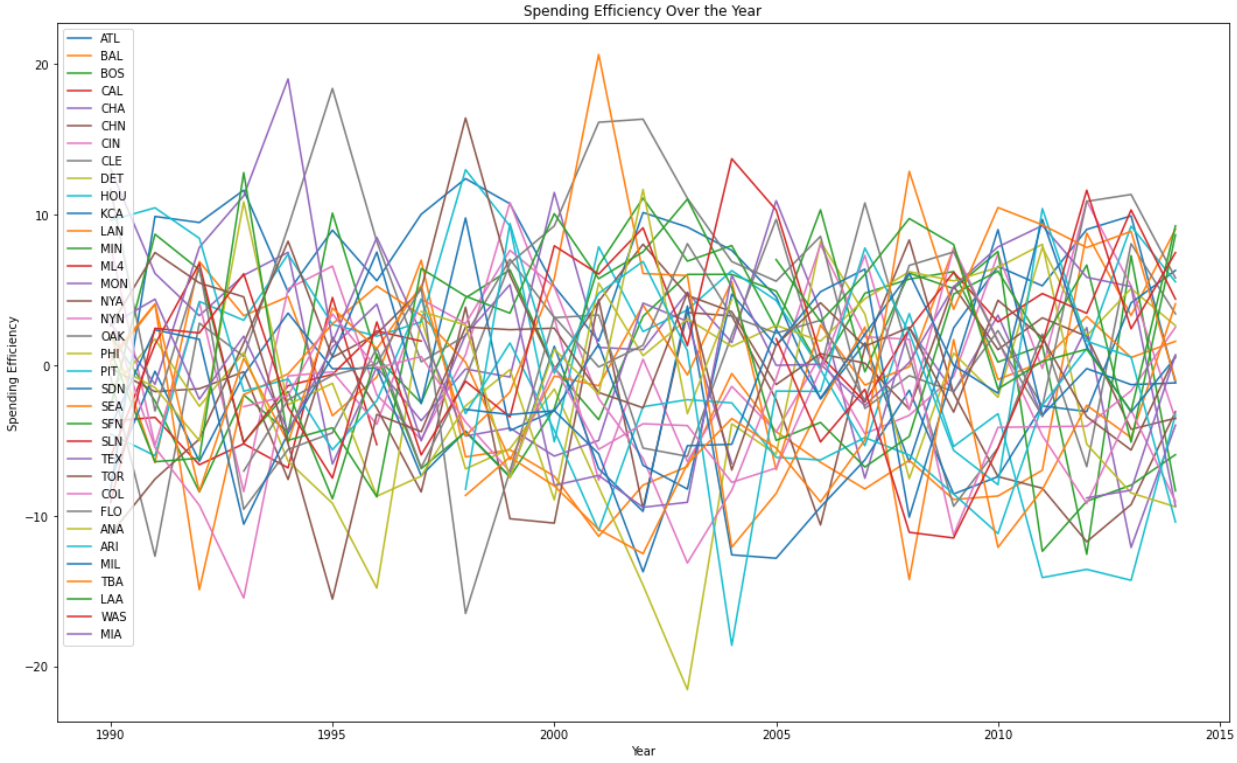
In [ ]:
```
# Calculate regression line using polyfit
slope, intercept = np.polyfit(op_new['standardized_payroll'], op_new['winning_p

# Plot scatter plot and regression line
plt.figure(figsize=(12,8))
plt.scatter(op_new['standardized_payroll'], op_new['winning_percentage'])
plt.plot(op_new['standardized_payroll'], slope * op_new['standardized_payroll']
plt.xlabel('Standardized Payroll')
plt.ylabel('Winning Percentage')
plt.title('Relationship between Standardized Payroll and Winning Percentage')
plt.show()
```



Spending efficiency

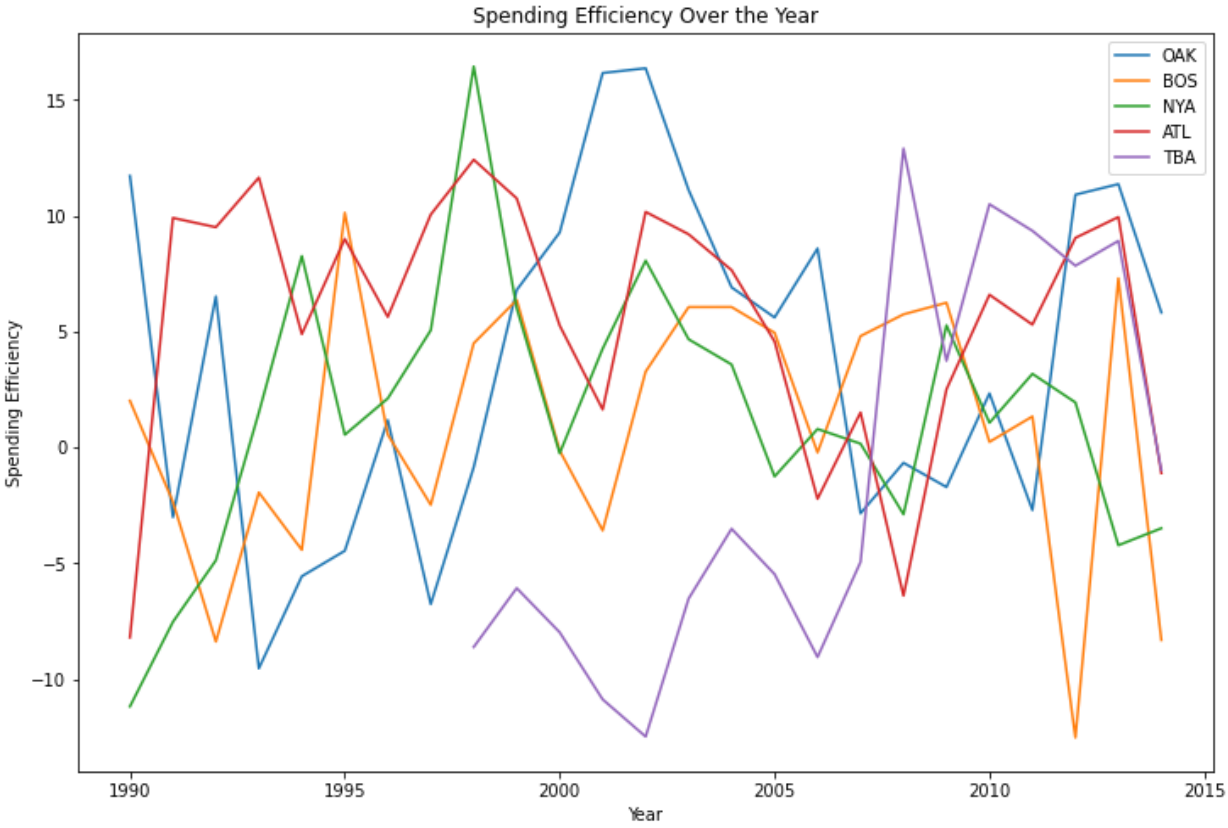
Problem 8

In [ ]:
```
op_new = op_new.copy()
# Calculate spending effiency
op_new['spending_effiency'] = op_new['winning_percentage'] - (50 + 2.5 * op_new

# Getting all team names
teams = op_new['teamID'].unique()

plt.figure(figsize=(18,11))
# Plot each team
for team in teams:
    team_data = op_new[op_new['teamID'] == team]
    plt.plot(team_data['yearID'], team_data['spending_effiency'], label=team)

# Add a legend and axis labels
plt.legend()
plt.xlabel('Year')
plt.ylabel('Spending Efficiency')
plt.title('Spending Efficiency Over the Year')

plt.show()
```



In [ ]:

```
teams = ['OAK', 'BOS', 'NYA', 'ATL', 'TBA']

plt.figure(figsize=(12,8))
# Plot each team
for team in teams:
    team_data = op_new[op_new['teamID'] == team]
    plt.plot(team_data['yearID'], team_data['spending_effiency'], label=team)

# Add a legend and axis labels
plt.legend()
plt.xlabel('Year')
plt.ylabel('Spending Efficiency')
plt.title('Spending Efficiency Over the Year')

plt.show()
```



Question 4

We can know that although some teams have a very high winning rate, their spending efficiency is not as satisfactory as the winning rate, which means winning percentage alone does not give us the full picture of a team's perfomance. Also, spending efficiency is not a fixed metric and can fluctuate over time. Teams must analyze previous data to make adjustments for their future spending to ensure they can get high efficiency.

During the Moneyball period, the Oakland Athletics(OAK) had a relatively good spending efficiency overall. However, there were fluctuations in their efficiency during certain years. From 1993 to 1997 and 2007 to 2011, Oakland's spending efficiency was notably lower or only slightly above zero, despite their on-field success. This suggests that during those years, the team may have overpaid for talent or relied too heavily on high-cost players. On the other hand, during the years of 1999 to 2006, Oakland had a significantly higher spending efficiency, which can be considered a positive indicator of their performance.