In [ ]:
```
'''
Project 2: SQLlite, pandas, and data wrangling
'''
```

In [ ]:
```
'''
Problem 1: Using SQL computed a relation containing the total payroll
'''
```

In [2]:
```
import sqlite3
import pandas

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

salary_query = "SELECT yearID, sum(salary) as total_payroll FROM Salar

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

Out[2]:

	yearID	total_payroll
0	1985	134401120.0
1	1986	157716444.0
2	1987	136088747.0
3	1988	157049812.0
4	1989	188771688.0

In [4]:
```
#handles float division and calculates winning percertange
calculations = "SELECT yearID, teamID, G, W, ((CAST(W AS float)/CAST(G
cursor.execute(calculations)
calculations_query = pandas.read_sql(calculations, conn)

#calculates total salary for each yearID for each teamID
salary_query = "SELECT yearID, sum(salary) as total_payroll FROM Salar
cursor.execute(salary_query)
```

Out[4]: <sqlite3.Cursor at 0x7fd0d341a030>

In [283]:

```
#merge Salaries and Teams table

'''
Dealt with missing data by finding columns in each table that had null
only created a new merged table where a team's wins and games were not
I merged the table with left join and union in order to account for mi
float values.

'''

select = "SELECT T.yearID, T.teamID, T.W, T.G, ((CAST(T.W AS float)/CA
table = pandas.read_sql(select, conn)
table.head()
```

Out[283]:

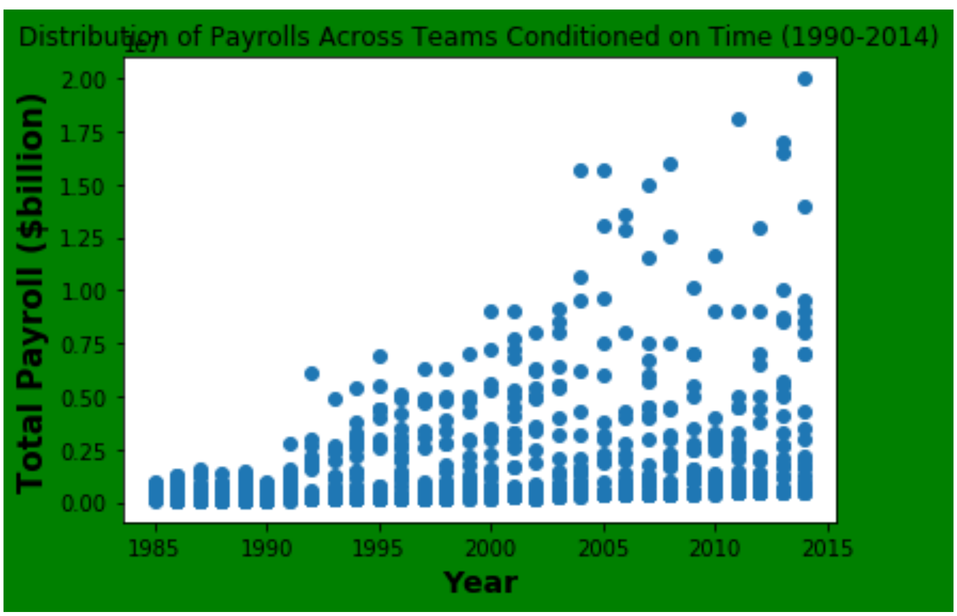
	yearID	teamID	W	G	Winning_Percentage	salary	Total_Payroll	Avg_Payroll	Std_Payı
0	1985	ATL	66	162	40.740741	870000.0	14807000.0	0.058756	0.000(
1	1985	BAL	83	161	51.552795	625000.0	11560712.0	0.054062	0.000(
2	1985	BOS	81	163	49.693252	915000.0	10897560.0	0.083964	0.000(
3	1985	CAL	90	162	55.555556	365000.0	14427894.0	0.025298	0.000(
4	1985	CHA	85	163	52.147239	147500.0	9846178.0	0.014980	0.000(

In [ ]:

```
#We want to understand how efficient teams have been historically at s
#and getting wins in return

'''
PART 2
Problem 2: Plots to explain the distribution of payrolls
across teams conditioned on time (from 1990-2014).

'''
```

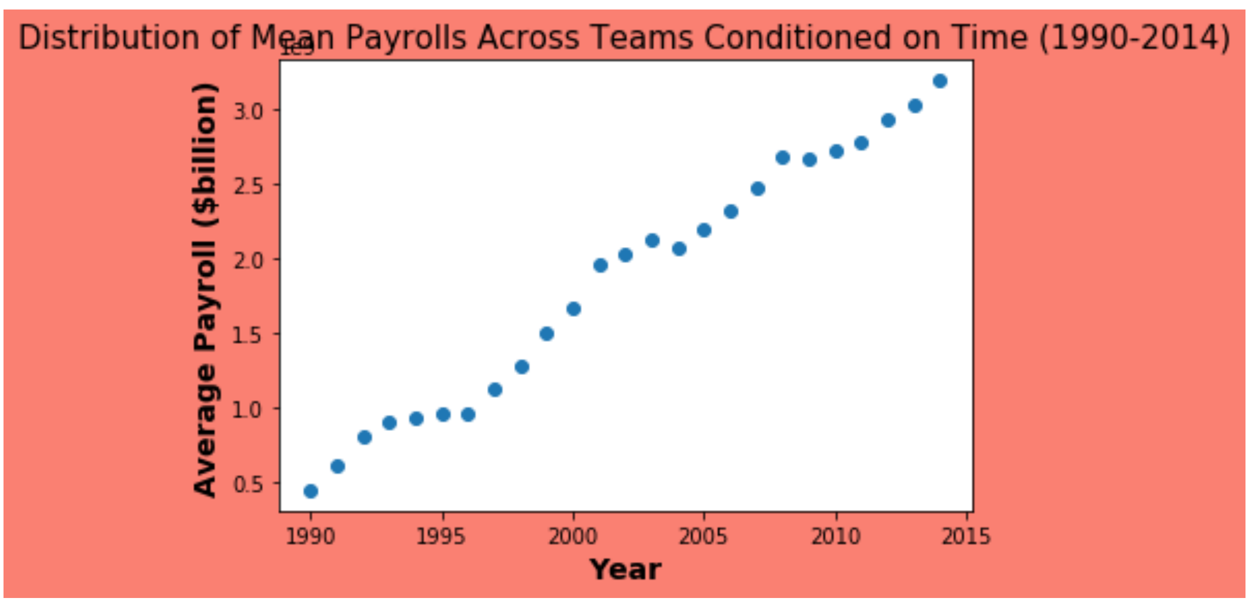
In [14]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

select = "SELECT yearID, salary from Salaries"
query = pandas.read_sql(select, conn)

payroll = pd.DataFrame(data=query, index=pd.date_range(start=pd.datet
payroll = payroll.cumsum()

x=table['yearID']
y=table['salary']

plt.figure()
plt.scatter(x,y)
plt.title("Distribution of Payrolls Across Teams Conditioned on Time (
plt.xlabel("Year", size=14, weight='bold')
plt.ylabel("Total Payroll ($billion)", size = 15, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'green'
plt.show()
```

```
/Users/khushibhansali/opt/anaconda3/lib/python3.7/site-packages/ipyke
rnel_launcher.py:8: FutureWarning: The pandas.datetime class is depre
cated and will be removed from pandas in a future version. Import fro
m datetime module instead.
```

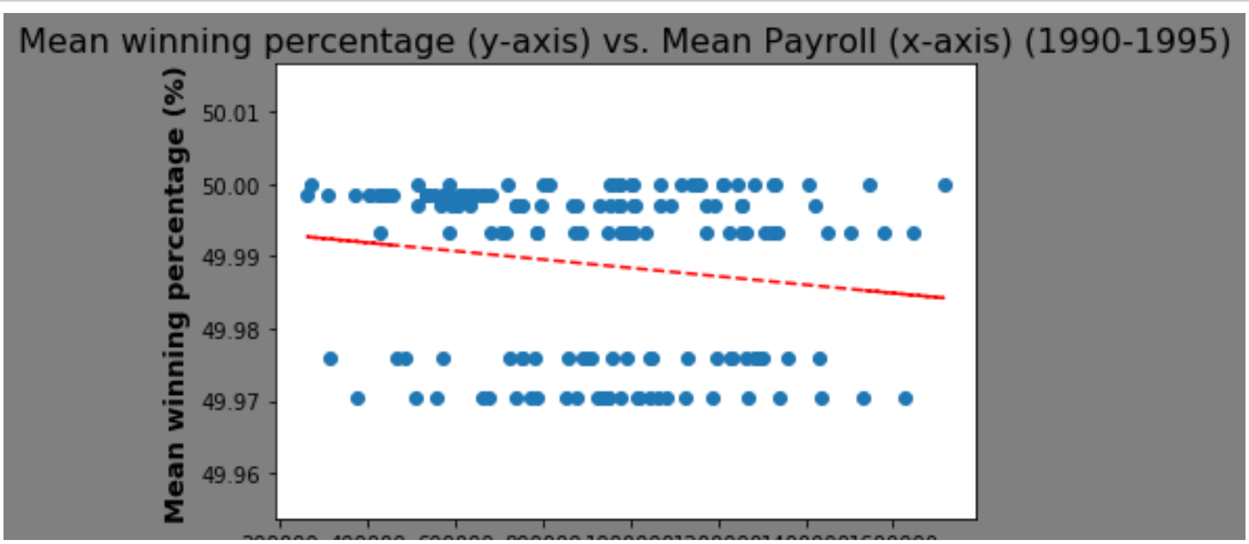


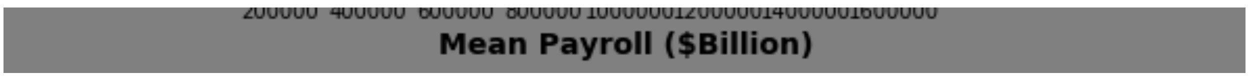
In [ ]:
```
'''
Question 1: As time increased, total payroll also increased. This migh
money to pay for better value players as time increased. The data bein
of this. However, because this scatterplot just measures total payroll
on the fact that teams may/may not be more efficent in the amount they
released.
'''
```

In [138]:
```
#Problem 3: shows mean payrolls across teams from 1990-2014
import datetime as dt
from datetime import datetime

mean_salary = "SELECT yearID, sum(salary) as sum_of_year, avg(sum(sala
query = pd.read_sql(mean_salary, conn)

years = "SELECT yearID from Salaries where yearID between 1990 and 201
query2 = pd.read_sql(years, conn)

x=query2['yearID']
y=query['avg_sum']

plt.figure()
plt.scatter(x,y)
z = np.polyfit(y, x, 1)
p = np.poly1d(z)

plt.title("Distribution of Mean Payrolls Across Teams Conditioned on T
plt.xlabel("Year", size=14, weight='bold')
plt.ylabel("Average Payroll ($billion)", size = 14, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'salmon'
plt.show()
```



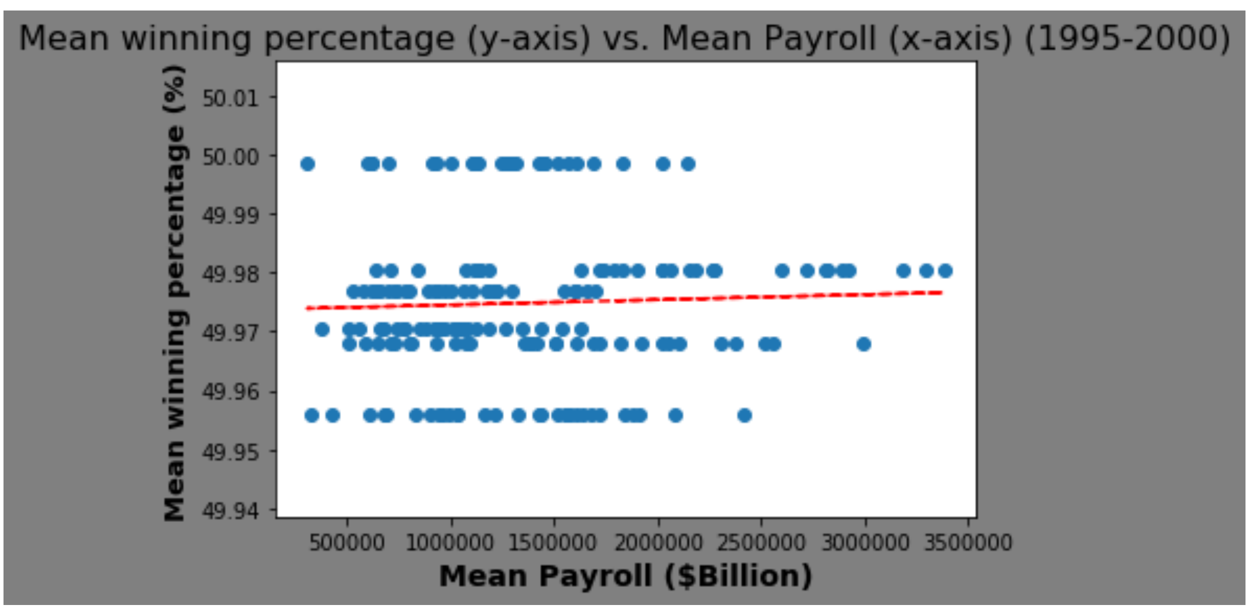
In [136]:
```
'''
Problem 4:
Scatterplot showing mean winning percentage
(y-axis) vs. mean payroll (x-axis) for each of the five time.

'''
#shows mean payrolls across teams from 1990-1995

import numpy

mean_salary = "SELECT avg(salary) as salary from Salaries where yearID
x_query = pd.read_sql(mean_salary, conn)

mean_wins = "SELECT yearID, teamID, ((CAST(W AS float)/CAST(G AS float
y_query = pd.read_sql(mean_wins, conn)

x=x_query['salary']
y=numpy.asarray(y_query['mean_win'])

plt.figure()

#create basic scatterplot
plt.scatter(x, y)

#obtain m (slope) and b(intercept) of linear regression line
m, b = np.polyfit(x, y, 1)

#add linear regression line to scatterplot
plt.plot(x, m*x+b,'r--')

plt.title("Mean winning percentage (y-axis) vs. Mean Payroll (x-axis)
plt.xlabel("Mean Payroll ($Billion)", size=14, weight='bold')
plt.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'grey'
plt.show()
```



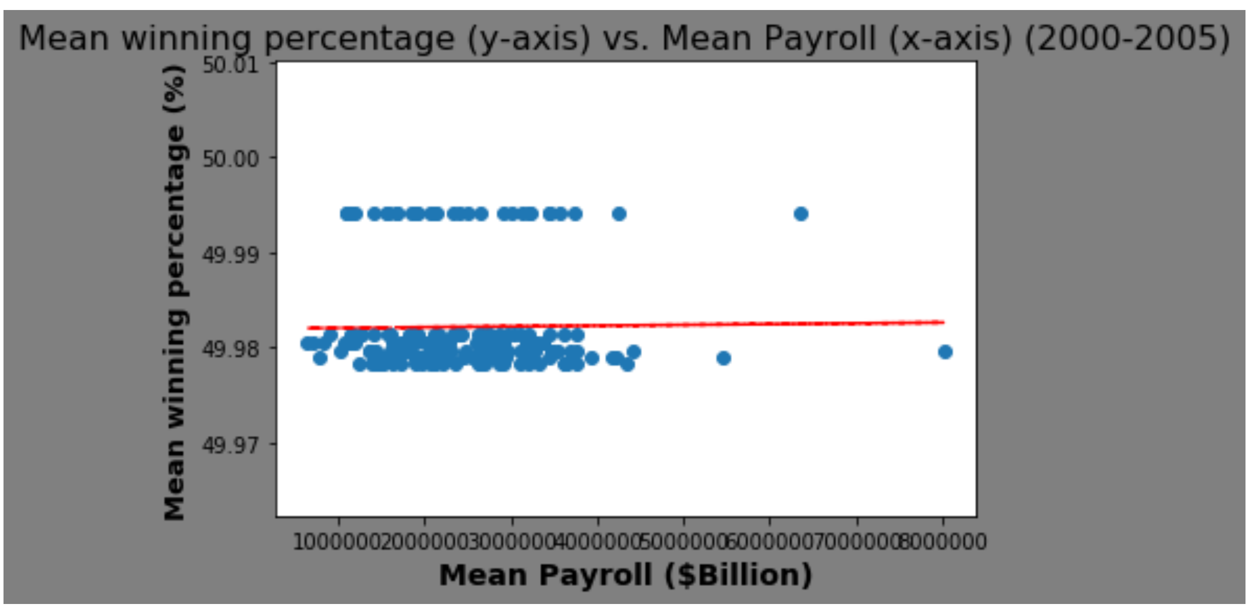


In [140]:

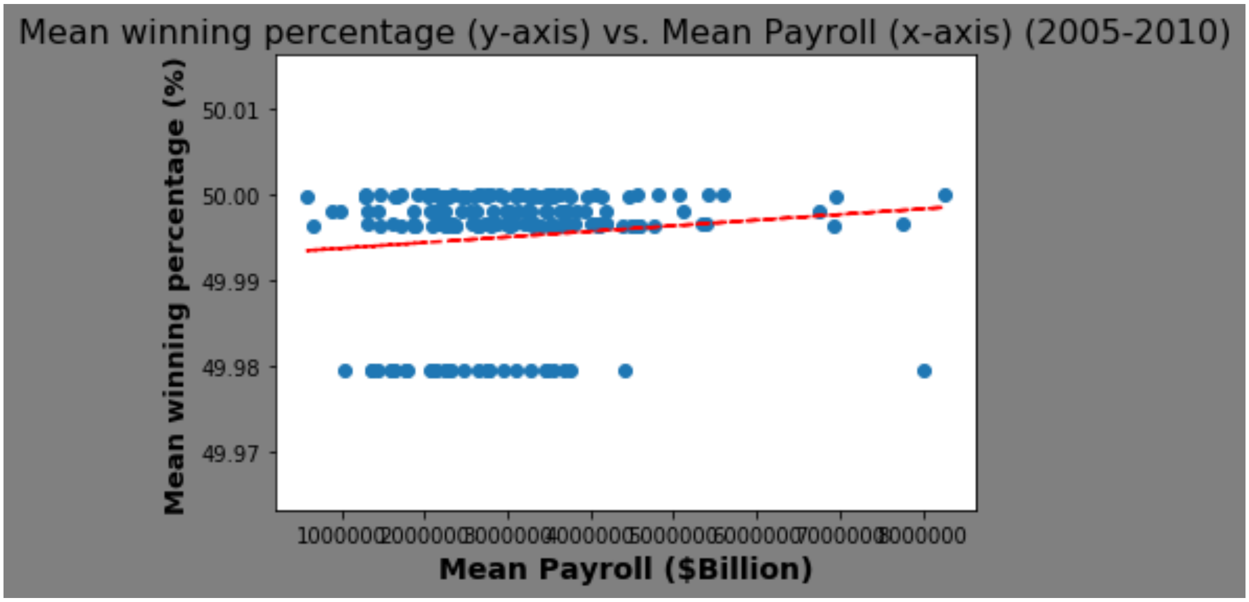
```
#Problem 4: mean winning percentage (y-axis) vs. mean payroll (x-axis)
#shows mean payrolls across teams from 1995-2000

mean_salary = "SELECT avg(salary) as salary from Salaries where yearID
x_query = pd.read_sql(mean_salary, conn)

mean_wins = "SELECT yearID, teamID, ((CAST(W AS float)/CAST(G AS float
y_query = pd.read_sql(mean_wins, conn)

x=x_query['salary']
y=numpy.asarray(y_query['mean_win'])

plt.figure()

#create basic scatterplot
plt.scatter(x, y)

#obtain m (slope) and b(intercept) of linear regression line
m, b = np.polyfit(x, y, 1)

#add linear regression line to scatterplot
plt.plot(x, m*x+b,'r--')

plt.title("Mean winning percentage (y-axis) vs. Mean Payroll (x-axis)
plt.xlabel("Mean Payroll ($Billion)", size=14, weight='bold')
plt.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'grey'
plt.show()
```



In [141]:
```
roblem 4: mean winning percentage (y-axis) vs. mean payroll (x-axis) fc
hows mean payrolls across teams from 2000-2005
port numpy

an_salary = "SELECT avg(salary) as salary from Salaries where yearID be
query = pd.read_sql(mean_salary, conn)

an_wins = "SELECT yearID, teamID, ((CAST(W AS float)/CAST(G AS float))*
query = pd.read_sql(mean_wins, conn)

x_query['salary']
numpy.asarray(y_query['mean_win'])

t.figure()

reate basic scatterplot
t.scatter(x, y)

btain m (slope) and b(intercept) of linear regression line
 b = np.polyfit(x, y, 1)

dd linear regression line to scatterplot
t.plot(x, m*x+b,'r--')

t.title("Mean winning percentage (y-axis) vs. Mean Payroll (x-axis) (20
t.xlabel("Mean Payroll ($Billion)", size=14, weight='bold')
t.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
t.rcParams["figure.facecolor"] = 'grey'
t.show()
```

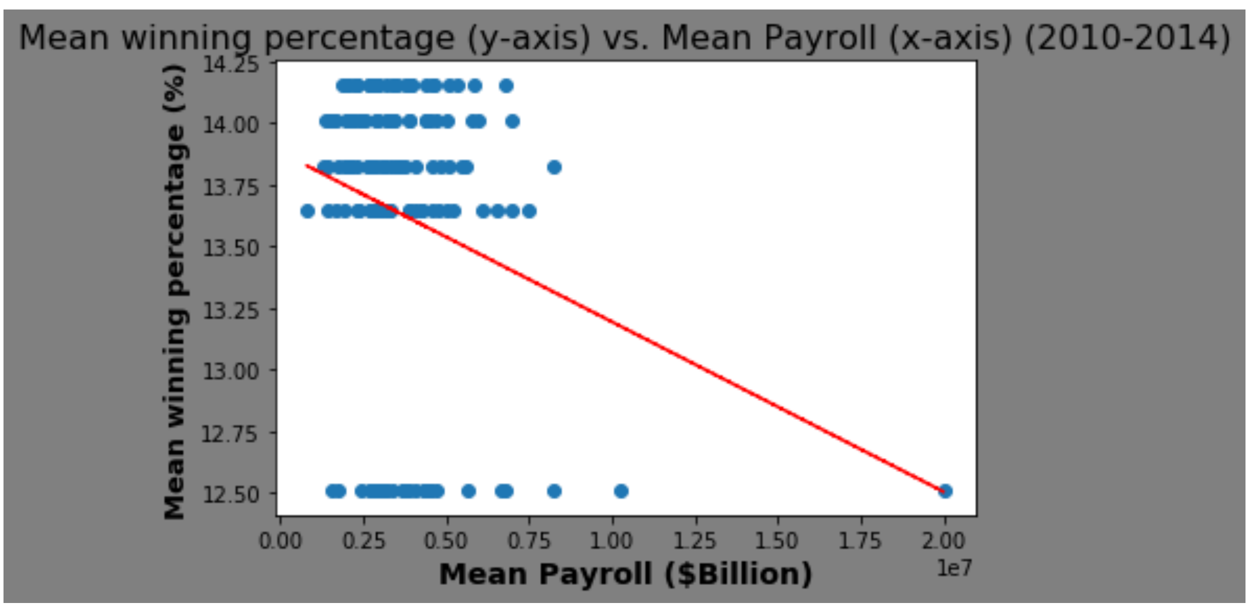


In [142]:
```
#Problem 4: mean winning percentage (y-axis) vs. mean payroll (x-axis)
#shows mean payrolls across teams from 2005-2010

mean_salary = "SELECT avg(salary) as salary from Salaries where yearID
x_query = pd.read_sql(mean_salary, conn)

mean_wins = "SELECT yearID, teamID, ((CAST(W AS float)/CAST(G AS float
y_query = pd.read_sql(mean_wins, conn)

x=x_query['salary']
y=numpy.asarray(y_query['mean_win'])

plt.figure()

#create basic scatterplot
plt.scatter(x, y)

#obtain m (slope) and b(intercept) of linear regression line
m, b = np.polyfit(x, y, 1)

#add linear regression line to scatterplot
plt.plot(x, m*x+b,'r--')

plt.title("Mean winning percentage (y-axis) vs. Mean Payroll (x-axis)
plt.xlabel("Mean Payroll ($Billion)", size=14, weight='bold')
plt.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'grey'
plt.show()
```



In [273]:

```
#Problem 4: mean winning percentage (y-axis) vs. mean payroll (x-axis)
#shows mean payrolls across teams from 2010-2014

select = "SELECT avg(S.salary) as salary, avg(sum(((CAST(T.W AS float)
query = pd.read_sql(select, conn)

x=query['salary']
y=numpy.asarray(query['mean_win'])

plt.figure()

#create basic scatterplot
plt.scatter(x, y)

#obtain m (slope) and b(intercept) of linear regression line
m, b = np.polyfit(x, y, 1)

#add linear regression line to scatterplot
plt.plot(x, m*x+b,'r--')

plt.title("Mean winning percentage (y-axis) vs. Mean Payroll (x-axis)
plt.xlabel("Mean Payroll ($Billion)", size=14, weight='bold')
plt.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'grey'
plt.show()
```



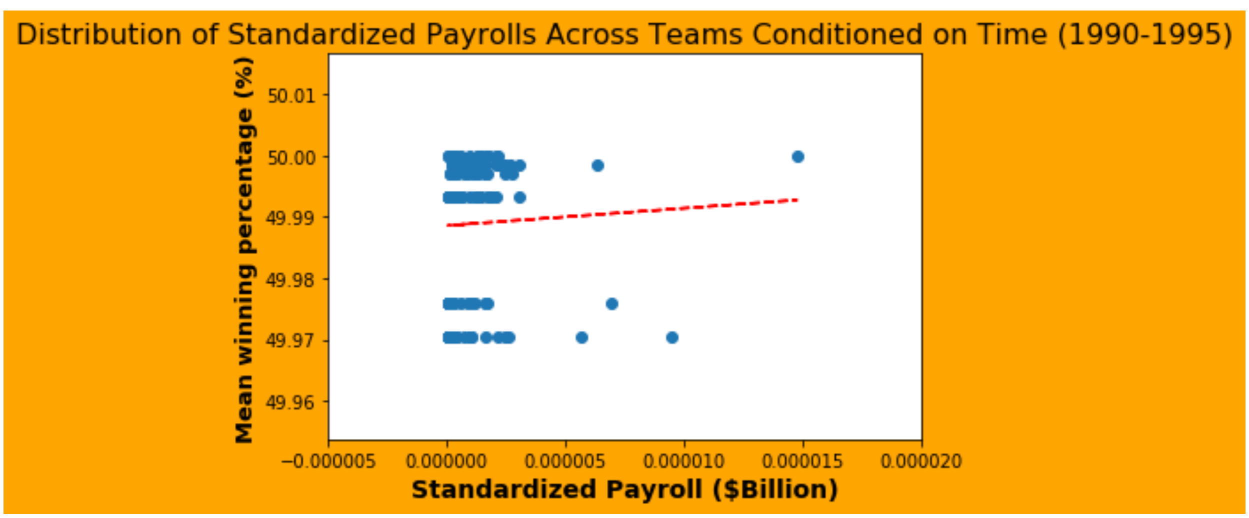
In [ ]:
```
'''
Question 2:
During 1990 to 1995, I see that as mean payroll increases, mean winnir
teams thought they were spending more efficently but actually weren't
During 1995 to 2000, as mean payroll increased, mean winning percentag
spending more to be able to afford better performing players.
During 2000 to 2005, as mean payroll increased, mean winning percentag
spending more to be able to afford better performing players.
During 2005 to 2010, as mean payroll increased, mean winning percentag
spending more to be able to afford better performing players.
During 2010 to 2014, as mean payroll increased, mean winning percentag
have to spend as much to afford high performing players. This was alsc
so maybe teams watched Oakland's performance to payroll ratio and lear

During these times, OAK and TBA standout at being particularly good at
In particular, Oakland spending efficiency significantly improved acrc
'''
```

In [284]:
```
'''

Part 3: Calculating standardized payroll

'''

#Problem 5: Create a new variable in dataset that standardizes payroll
year = table['yearID']
payroll = table['salary']
avg_payroll = table['Avg_Payroll']
std_payroll = table['Std_Payroll']
table.head()
```

Out[284]:

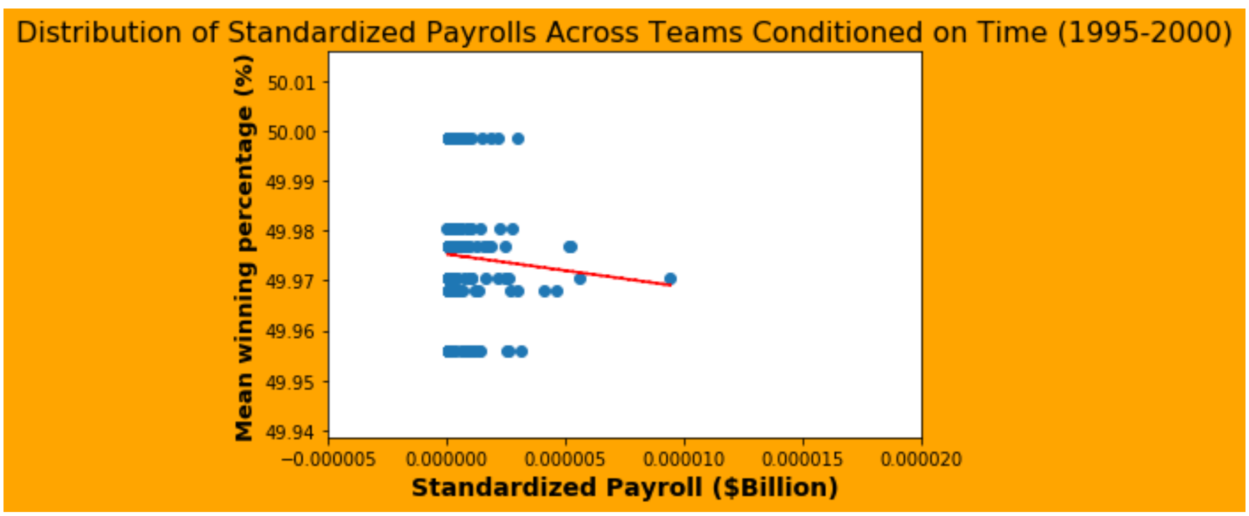
	yearID	teamID	W	G	Winning_Percentage	salary	Total_Payroll	Avg_Payroll	Std_Payı
0	1985	ATL	66	162	40.740741	870000.0	14807000.0	0.058756	0.0000
1	1985	BAL	83	161	51.552795	625000.0	11560712.0	0.054062	0.0000
2	1985	BOS	81	163	49.693252	915000.0	10897560.0	0.083964	0.0000
3	1985	CAL	90	162	55.555556	365000.0	14427894.0	0.025298	0.0000
4	1985	CHA	85	163	52.147239	147500.0	9846178.0	0.014980	0.0000

In [220]:
```
#Problem 6: shows standardized payrolls across teams from 1990-2014
#shows mean payrolls across teams from 1990-1995

mean_salary = "SELECT (CAST (salary-(CAST(salary AS float)/CAST(sum(sa
x_query = pd.read_sql(mean_salary, conn)

mean_wins = "SELECT yearID, teamID, ((CAST(W AS float)/CAST(G AS float
y_query = pd.read_sql(mean_wins, conn)

x=x_query['Std_Payroll']
y=numpy.asarray(y_query['mean_win'])

plt.figure()

#create basic scatterplot
plt.scatter(x, y)
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b,'r--')

plt.xlim([-0.000005, 0.000020])
plt.title("Distribution of Standardized Payrolls Across Teams Conditio
plt.xlabel("Standardized Payroll ($Billion)", size=14, weight='bold')
plt.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'orange'
plt.show()
```



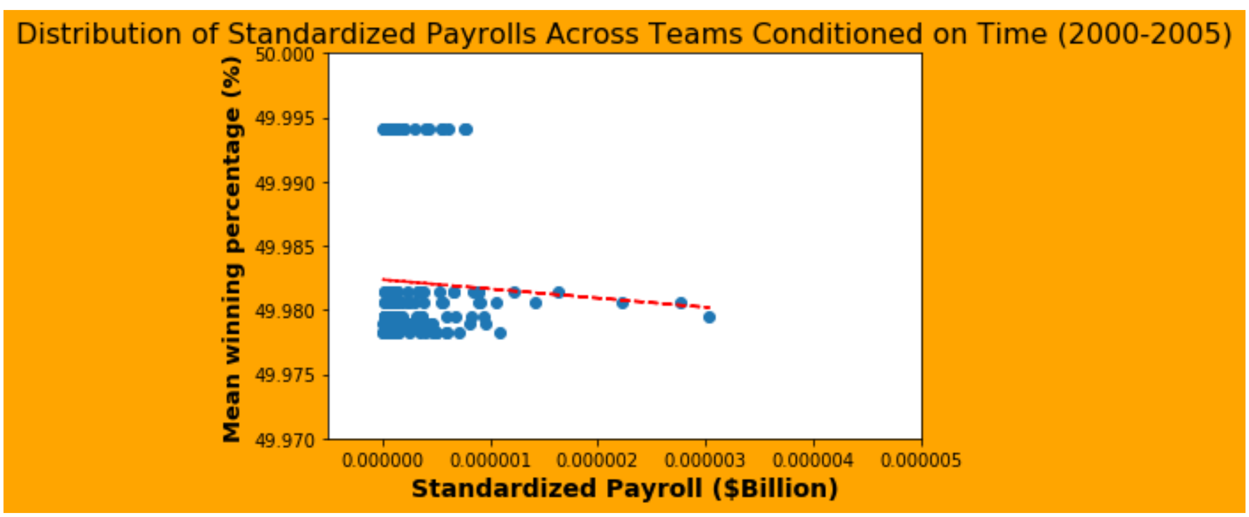
In [221]:

```
#Problem 6: shows standardized payrolls across teams
#shows mean payrolls across teams from 1995-2000

mean_salary = "SELECT (CAST (salary-(CAST(salary AS float)/CAST(sum(sa
x_query = pd.read_sql(mean_salary, conn)

mean_wins = "SELECT yearID, teamID, ((CAST(W AS float)/CAST(G AS float
y_query = pd.read_sql(mean_wins, conn)

x=x_query['Std_Payroll']
y=numpy.asarray(y_query['mean_win'])

plt.figure()

#create basic scatterplot
plt.scatter(x, y)
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b,'r--')

plt.xlim([-0.000005, 0.000020])
plt.title("Distribution of Standardized Payrolls Across Teams Conditic
plt.xlabel("Standardized Payroll ($Billion)", size=14, weight='bold')
plt.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'orange'
plt.show()
```



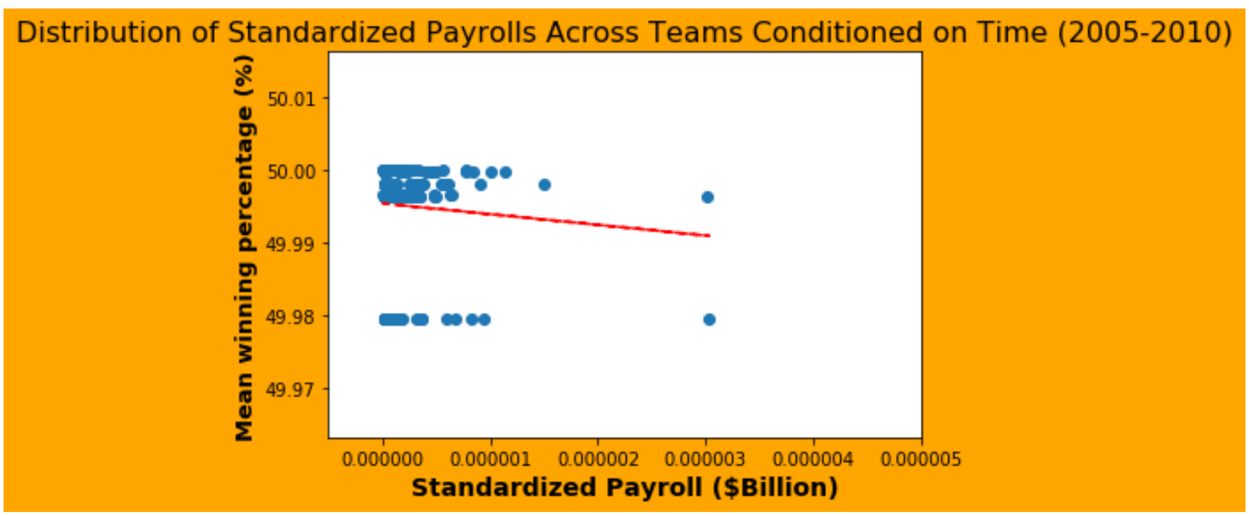
In [275]:

```
lem 6: shows standardized payrolls across teams
s mean payrolls across teams from 2000-2005

salary = "SELECT (CAST (salary-(CAST(salary AS float)/CAST(sum(salary)
ry = pd.read_sql(mean_salary, conn)

wins = "SELECT yearID, teamID, ((CAST(W AS float)/CAST(G AS float))*10
ry = pd.read_sql(mean_wins, conn)

uery['Std_Payroll']
py.asarray(y_query['mean_win'])

igure()

te basic scatterplot
catter(x, y)
= np.polyfit(x, y, 1)
lot(x, m*x+b,'r--')

lim([-0.0000005, 0.000005])
lim([49.97, 50])
itle("Distribution of Standardized Payrolls Across Teams Conditioned o
label("Standardized Payroll ($Billion)", size=14, weight='bold')
label("Mean winning percentage (%)", size = 13, weight = 'bold')
cParams["figure.facecolor"] = 'orange'
how()
```

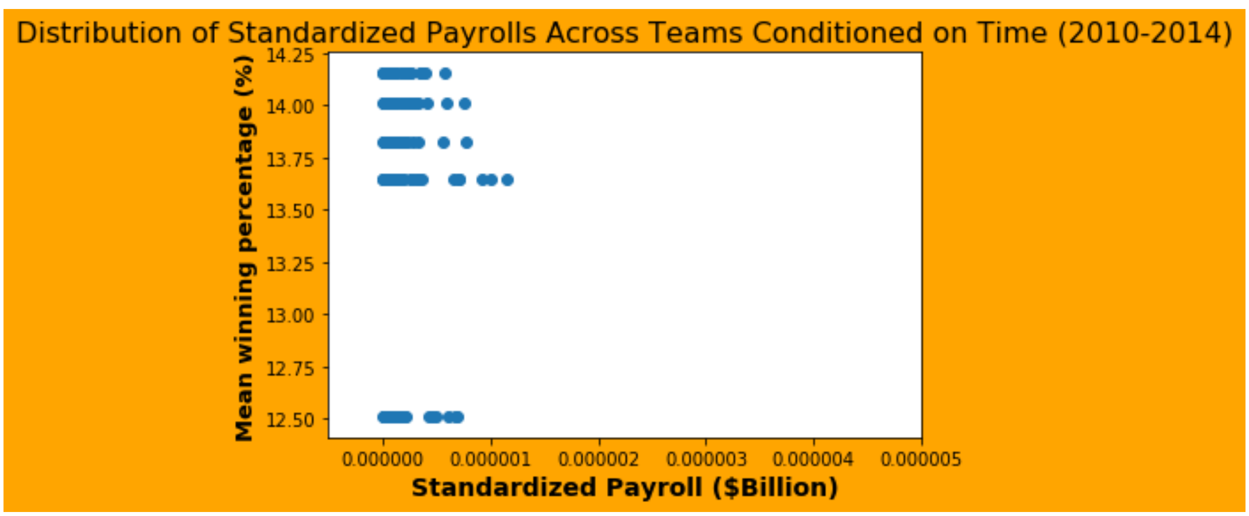


In [223]:

```
#Problem 6: shows standardized payrolls across teams
#shows mean payrolls across teams from 2005-2010

mean_salary = "SELECT (CAST (salary-(CAST(salary AS float)/CAST(sum(sa
x_query = pd.read_sql(mean_salary, conn)

mean_wins = "SELECT yearID, teamID, ((CAST(W AS float)/CAST(G AS float
y_query = pd.read_sql(mean_wins, conn)

x=x_query['Std_Payroll']
y=numpy.asarray(y_query['mean_win'])

plt.figure()

#create basic scatterplot
plt.scatter(x, y)
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b,'r--')

plt.xlim([-0.0000005, 0.000005])
plt.title("Distribution of Standardized Payrolls Across Teams Conditic
plt.xlabel("Standardized Payroll ($Billion)", size=14, weight='bold')
plt.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'orange'
plt.show()
```



In [224]:
```
Problem 6: shows standardized payrolls across teams
Shows mean payrolls across teams from 2010-2014

elect = "SELECT (CAST (S.salary-(CAST(S.salary AS float)/CAST(sum(S.sal
uery = pd.read_sql(select, conn)

=query['Std_Payroll']
=numpy.asarray(query['mean_win'])

lt.figure()

create basic scatterplot
lt.scatter(x, y)
n, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b,'r--')

lt.xlim([-0.0000005, 0.000005])
lt.title("Distribution of Standardized Payrolls Across Teams Conditione
lt.xlabel("Standardized Payroll ($Billion)", size=14, weight='bold')
lt.ylabel("Mean winning percentage (%)", size = 13, weight = 'bold')
lt.rcParams["figure.facecolor"] = 'orange'
lt.show()
```

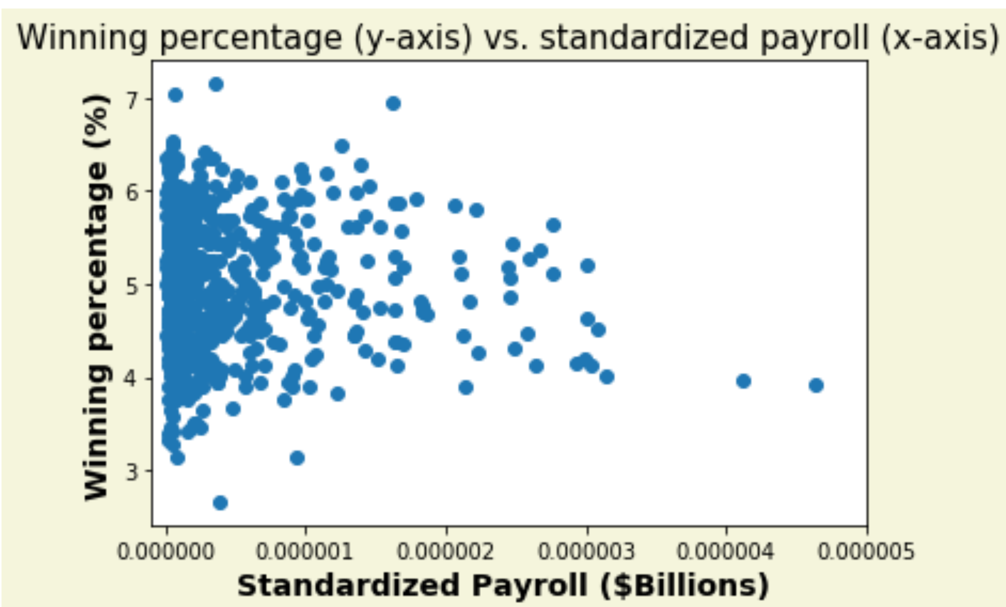


In [ ]:

```
on 3:

 1990 to 1995, I see that as standardized payroll increases, mean winn
ng more to be able to afford better performing players. .
 1995 to 2000, as standardized payroll increased, mean winning percent
o spend as much to afford high performing players.
 2000 to 2005, as standardized payroll increased, mean winning percent
o spend as much to afford high performing players. This was also aroun
be teams watched Oakland's performance to payroll ratio and learned fr
 2005 to 2010, as standardized payroll increased, mean winning percent
o spend as much to afford high performing players.
 2010 to 2014, as standardized payroll increased, mean winning percent
o spend as much to afford high performing players.

l, by changing the plots from mean payroll to standardized payroll, we
h year, mean winning percentage was highest when payroll was lowest be
erforming players. The teams were probably able to understand what pla
en optimally pay them for most wins.
```

In [297]:

```
#Problem 7
#Make a single scatter plot of winning percentage (y-axis) vs. standar
#Add a regression line to highlight the relationship.

#DataFrame.dropna, or turn it into a df
select = "SELECT (CAST (S.salary-(CAST(S.salary AS float)/CAST(sum(S.s
query = pd.read_sql(select, conn)

x = query['Std_Payroll']
y = query['Winning_Percentage']

plt.figure()
plt.scatter(x, y)

#Regression line
#z = np.polyfit(x, y, 1)
#p = np.poly1d(z)
#plt.plot(x, p(x),'r--')

plt.xlim([-0.0000001, 0.000005])

plt.title("Winning percentage (y-axis) vs. standardized payroll (x-axi
plt.xlabel("Standardized Payroll ($Billions) ", size=14, weight='bold'
plt.ylabel("Winning percentage (%)", size = 14, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'beige'
plt.show()
```

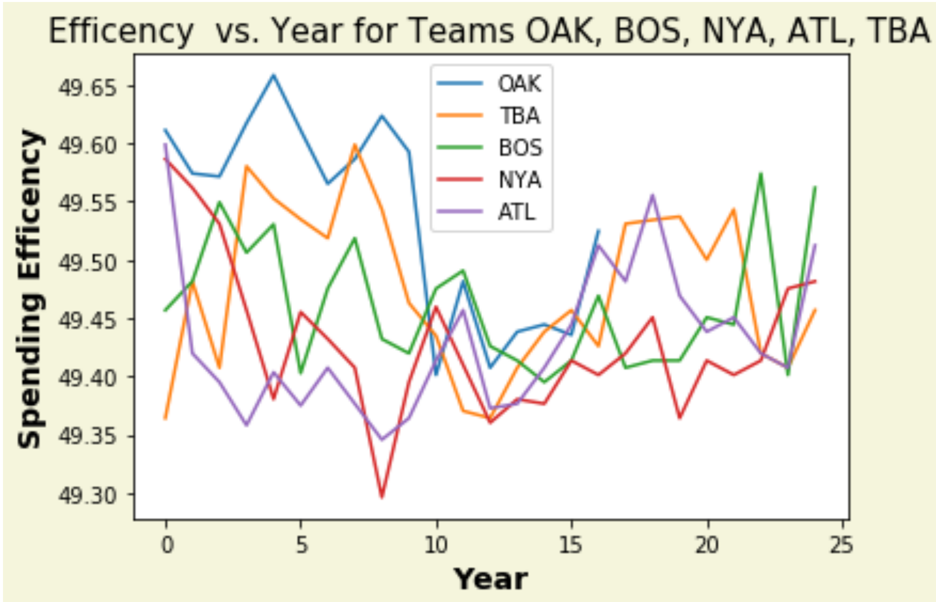


In [329]:

```
# Problem 8
# Line plot with year(x-axis) and efficiency(y-axis)
# Teams plotted are Oakland, the New York Yankees, Boston, Atlanta and

select = "SELECT S.yearID, S.teamID, (CAST (S.salary-(CAST(S.salary AS
```

```
query = pd.read_sql(select, conn)

expected_win_pct = []
std_pay = query['Std_Payroll']

for val in std_pay:
    expected_win_pct.append(50 + 2.5*val)

efficency = []
win_pct = query['Winning_Percentage']

for (a, b) in zip(win_pct, expected_win_pct):
    efficency.append(abs(a-b))

teams = query['teamID']
select_teams=['OAK', 'BOS', 'NYA', 'ATL', 'TBA']
efficency_select_teams = []
all_years = query['yearID']
year = []

oak=[]
bos=[]
nya=[]
atl=[]
tba=[]

for (teamID, efficent, yr) in zip(teams, efficency, all_years):
    if(teamID == 'OAK'):
        oak.append(efficent)
        year.append(yr)
    if(teamID=='BOS'):
        bos.append(efficent)
    if(teamID=='NYA'):
        nya.append(efficent)
    if(teamID=='ATL'):
        atl.append(efficent)
    if(teamID=='TBA'):
        tba.append(efficent)


plt.figure()
plt.plot(tba, label = "OAK")
plt.plot(oak, label = "TBA")
plt.plot(bos, label = "BOS")
plt.plot(nya, label = "NYA")
plt.plot(atl, label = "ATL")

plt.legend()
plt.title("Efficency  vs. Year for Teams OAK, BOS, NYA, ATL, TBA", siz
plt.xlabel("Year", size=14, weight='bold')
```

```
plt.ylabel("Spending Efficency", size = 14, weight = 'bold')
plt.rcParams["figure.facecolor"] = 'beige'
plt.show()
```



In [ ]:

```
'''
Question 4:
From this plot, I can see that as years the general trend was that eff
This could be because some teams figured out what characteristics resu

From this plot, I can see teams didn't have to spend much and still
recieved wins and had decent efficency. In particular, Oakland had the
spending efficency from 2000 to 2005 which was the year moneyball was 
Their spending efficency peaked above all other teams and they had hig
percentages as well as seen in graphs from question 2 and 3. In genera
proves that moneyball was a worthy movie that helped teams improve spe
over the years.

'''
```