In [809]:
```
#load relevant packages
import pandas as pd
from scipy.stats import uniform
import statsmodels.api as sm
import statsmodels.formula.api as smf
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

In [810]:
```
#load baseline coaches file
coaches = pd.read_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/coaches_baseline.csv')
coaches.head(10)
```

Out[810]:

	School	Conference	Coach	SchoolPay	TotalPay	Bonus	BonusPaid	AssistantPay	Buyout
0	Air Force	Mt. West	Troy Calhoun	885000	885000	247000	--	$0	--
1	Akron	MAC	Terry Bowden	$411,000	$412,500	$225,000	$50,000	$0	$688,500
2	Alabama	SEC	Nick Saban	$8,307,000	$8,307,000	$1,100,000	$500,000	$0	$33,600,000
3	Alabama at Birmingham	C-USA	Bill Clark	$900,000	$900,000	$950,000	$165,471	$0	$3,847,500
4	Appalachian State	Sun Belt	Scott Satterfield	$712,500	$712,500	$295,000	$145,000	$0	$2,160,417
5	Arizona	Pac-12	Kevin Sumlin	$1,600,000	$2,000,000	$2,025,000	--	$0	$10,000,000
6	Arizona State	Pac-12	Herm Edwards	$2,000,000	$2,000,000	$3,010,000	--	$0	$8,166,667
7	Arkansas	SEC	Chad Morris	$3,500,000	$3,500,000	$1,000,000	--	$0	$12,500,000
8	Arkansas State	Sun Belt	Blake Anderson	$825,000	$825,000	$185,000	$25,000	$0	$300,000
9	Army	Ind.	Jeff Monken	932521	932521	--	--	$0	--

In [811]:
```
#load ancilary revenue data
revenue = pd.read_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/revenue.csv')
revenue.head(10)
```

Out[811]:

	RK	School	Conference	Revenue	Expenses
0	1	Texas	Big 12	$223,879,781	$204,234,897
1	2	Texas A&M	SEC	$212,748,002	$169,012,456
2	3	Ohio State	Big Ten	$210,548,239	$220,572,956
3	4	Michigan	Big Ten	$197,820,410	$190,952,175
4	5	Georgia	SEC	$174,042,482	$143,299,554
5	6	Penn State	Big Ten	$164,529,326	$160,369,805
6	7	Alabama	SEC	$164,090,889	$185,317,681
7	8	Oklahoma	Big 12	$163,126,695	$157,958,270
8	9	Florida	SEC	$159,706,937	$141,829,002
9	10	LSU	SEC	$157,787,782	$148,977,880

In [812]:

```
#load ancilary stadium size data
size = pd.read_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/Stadium_Size.csv')
size.head(10)
```

Out[812]:

	Stadium	College	Conference	Capacity	Opened
0	Michigan Stadium	Michigan	Big Ten	107,601	1927
1	Beaver Stadium	Penn State	Big Ten	106,572	1960
2	Ohio Stadium	Ohio State	Big Ten	104,944	1922
3	Kyle Field	Texas A&M	SEC	102,733	1904
4	Neyland Stadium	Tennessee	SEC	102,521	1921
5	Bryant Denny Stadium	Alabama	SEC	101,821	1929
6	Tiger Stadium	LSU	SEC	100,500	1924
7	Royal Memorial Stadium	Texas	Big 12	100,119	1924
8	Los Angeles Coliseum	USC	Pac 12	93,607	1923
9	Sanford Stadium	Georgia	SEC	92,746	1929

In [813]:
```
#load ancilary coach win data
coach = pd.read_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/coach_win.csv')
coach.head(10)
```

Out[813]:

	team	conf	coach	firstyear	currwin	currloss	currwin%	win	loss	win_percent	oc	dc	stc
0	Cincinnati Bearcats	The American	Luke Fickell	2017	35.0	14.0	0.714	41	21	0.661	Mike Denbrock	Mike Tressel	Brian Mason
1	East Carolina Pirates	The American	Mike Houston	2019	7.0	14.0	0.333	7	14	0.333	Donnie Kirkpatrick	Blake Harrell	Tim Daoust
2	Houston Cougars	The American	Dana Holgorsen	2019	7.0	13.0	0.35	68	54	0.557	Shannon Dawson	Doug Belk	Mark Scott
3	Memphis Tigers	The American	Ryan Silverfield	2020	8.0	4.0	0.667	8	4	0.667	Kevin Johns	Mike MacIntyre	Charles Bankins
4	Navy Midshipmen	The American	Ken Niumatalolo	2007	101.0	67.0	0.601	101	67	0.601	Ivin Jasper	Brian Newberry	Danny O'Rourke
5	SMU Mustangs	The American	Sonny Dykes	2018	22.0	14.0	0.611	63	59	0.516	Garrett Riley	Jim Leavitt	Kenny Perry
6	South Florida Bulls	The American	Jeff Scott	2020	1.0	8.0	0.111	1	8	0.111	Charlie Weis Jr.	Glenn Spencer	Daniel Da Prato
7	Temple Owls	The American	Rod Carey	2019	9.0	11.0	0.45	61	41	0.598	Mike Uremovich	Jeff Knowles	Brett Diersen
8	Tulane Green Wave	The American	Willie Fritz	2016	29.0	33.0	0.468	46	40	0.535	Chip Long	Chris Hampton	Willie Fritz
9	Tulsa Golden Hurricane	The American	Philip Montgomery	2015	31.0	40.0	0.437	31	40	0.437	Philip Montgomery	Joseph Gillespie	Calvin Lowry

In [814]:
```
#load ancilary grad rate win data
grad = pd.read_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/grad_rates.csv')
grad.head(10)
```

Out[814]:

	x	school	conf	sport	state	gsr	fgr	Unnamed: 7
0	2012	Abilene Christian	Southland Conference	Football	TX	70	47.0	NaN
1	2012	Akron	Mid-American Conference	Football	OH	75	61.0	NaN
2	2012	Alabama A&M	Southwestern Athletic Conf.	Football	AL	59	49.0	NaN
3	2012	Alabama State	Southwestern Athletic Conf.	Football	AL	58	39.0	NaN
4	2012	Alabama	Southeastern Conference	Football	AL	85	65.0	NaN
5	2012	Alabama at Birmingham	Conference USA	Football	AL	71	51.0	NaN
6	2012	University at Albany	Colonial Athletic Association	Football	NY	88	63.0	NaN
7	2012	Alcorn State	Southwestern Athletic Conf.	Football	MS	58	40.0	NaN
8	2012	Appalachian State	Sun Belt Conference	Football	NC	75	67.0	NaN
9	2012	Arizona State	Pac-12 Conference	Football	AZ	75	60.0	NaN

In [815]:

```
#join data sets
temp = pd.merge(coaches,revenue,how='outer',left_on=['School'],right_on=['School'])
temp1 = pd.merge(temp,size,how='outer',left_on=['School'],right_on=['College'])
temp2 = pd.merge(temp1,coach,how='outer',left_on=['Coach'],right_on=['coach'])
temp2 = pd.merge(temp1,coach,how='outer',left_on=['Coach'],right_on=['coach'])

temp2 = pd.merge(temp2,grad,how='outer',left_on=['School'],right_on=['school'])
temp1 = pd.merge(temp1,grad,how='outer',left_on=['School'],right_on=['school'])

temp1.head(10)
#len(df2)
```

Out[815]:

	School	Conference_x	Coach	SchoolPay	TotalPay	Bonus	BonusPaid	AssistantPay	Buyout	RK	...	Capacity	Opened	x
0	Air Force	Mt. West	Troy Calhoun	885000	885000	247000	--	$0	--	57.0	...	52,237	1962.0	2012.0
1	Akron	MAC	Terry Bowden	$411,000	$412,500	$225,000	$50,000	$0	$688,500	84.0	...	30,000	2009.0	2012.0
2	Alabama	SEC	Nick Saban	$8,307,000	$8,307,000	$1,100,000	$500,000	$0	$33,600,000	7.0	...	101,821	1929.0	2012.0
3	Alabama at Birmingham	C-USA	Bill Clark	$900,000	$900,000	$950,000	$165,471	$0	$3,847,500	86.0	...	NaN	NaN	2012.0
4	Appalachian State	Sun Belt	Scott Satterfield	$712,500	$712,500	$295,000	$145,000	$0	$2,160,417	81.0	...	24,150	1962.0	2012.0
5	Arizona	Pac-12	Kevin Sumlin	$1,600,000	$2,000,000	$2,025,000	--	$0	$10,000,000	38.0	...	56,037	1928.0	2012.0
6	Arizona State	Pac-12	Herm Edwards	$2,000,000	$2,000,000	$3,010,000	--	$0	$8,166,667	27.0	...	56,232	1958.0	2012.0
7	Arizona State	Pac-12	Herm Edwards	$2,000,000	$2,000,000	$3,010,000	--	$0	$8,166,667	27.0	...	56,232	1958.0	2012.0
8	Arkansas	SEC	Chad Morris	$3,500,000	$3,500,000	$1,000,000	--	$0	$12,500,000	20.0	...	72,000	1938.0	2012.0
9	Arkansas State	Sun Belt	Blake Anderson	$825,000	$825,000	$185,000	$25,000	$0	$300,000	92.0	...	30,964	2002.0	NaN

10 rows × 26 columns

In [816]:
```python
for col in temp1.columns:
    print(col)
```

```
School
Conference_x
Coach
SchoolPay
TotalPay
Bonus
BonusPaid
AssistantPay
Buyout
RK
Conference_y
Revenue
Expenses
Stadium
College
Conference
Capacity
Opened
x
school
conf
sport
state
gsr
fgr
Unnamed: 7
```

In [817]:

```
#drop meaningless columns
temp2 = temp2.drop(['Conference_y', 'AssistantPay', 'SchoolPay', 'Bonus', 'BonusPaid',
                    'Stadium', 'Buyout','RK','College','Conference','Opened','team',
                    'coach','firstyear','currwin','currloss','win','loss','school','x',
                    'sport','state','Unnamed: 7'], axis=1)
temp1 = temp1.drop(['Conference_y', 'AssistantPay', 'SchoolPay', 'Bonus', 'BonusPaid',
                    'Stadium', 'Buyout','RK','College','Conference','Opened','school','x',
                    'sport','state','Unnamed: 7'], axis=1)

"""
Want to keep two data frames. temp2 has the coaches win%
which could be a helpful variable but the coaches name
is more difficult to join on and limits the dataset to
about 60 rows. I want to be able to test the more limited
dataset; however, it may not be as useful as temp1
"""
for column in list(temp2):
  temp2[column].replace('--', np.nan, inplace=True)
temp2.dropna(inplace=True)

for column in list(temp1):
  temp1[column].replace('--', np.nan, inplace=True)
temp1.dropna(inplace=True)

#convert columns that should be numbers to float
temp2['TotalPay'] = temp2['TotalPay'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).ast
temp1['TotalPay'] = temp1['TotalPay'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).ast
temp2['Revenue'] = temp2['Revenue'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype
temp1['Revenue'] = temp1['Revenue'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype
temp2['Expenses'] = temp2['Expenses'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).ast
temp1['Expenses'] = temp1['Expenses'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).ast
temp1['Capacity'] = temp1['Capacity'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).ast
temp2['Capacity'] = temp2['Capacity'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).ast
temp2['win_percent'] = temp2['win_percent'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',',''
```

In [818]:

```
for col in temp1.columns:
    print(col)

len(temp1)
```

```
School
Conference_x
Coach
TotalPay
Revenue
Expenses
Capacity
conf
gsr
fgr
```

Out[818]: 97

In [781]:

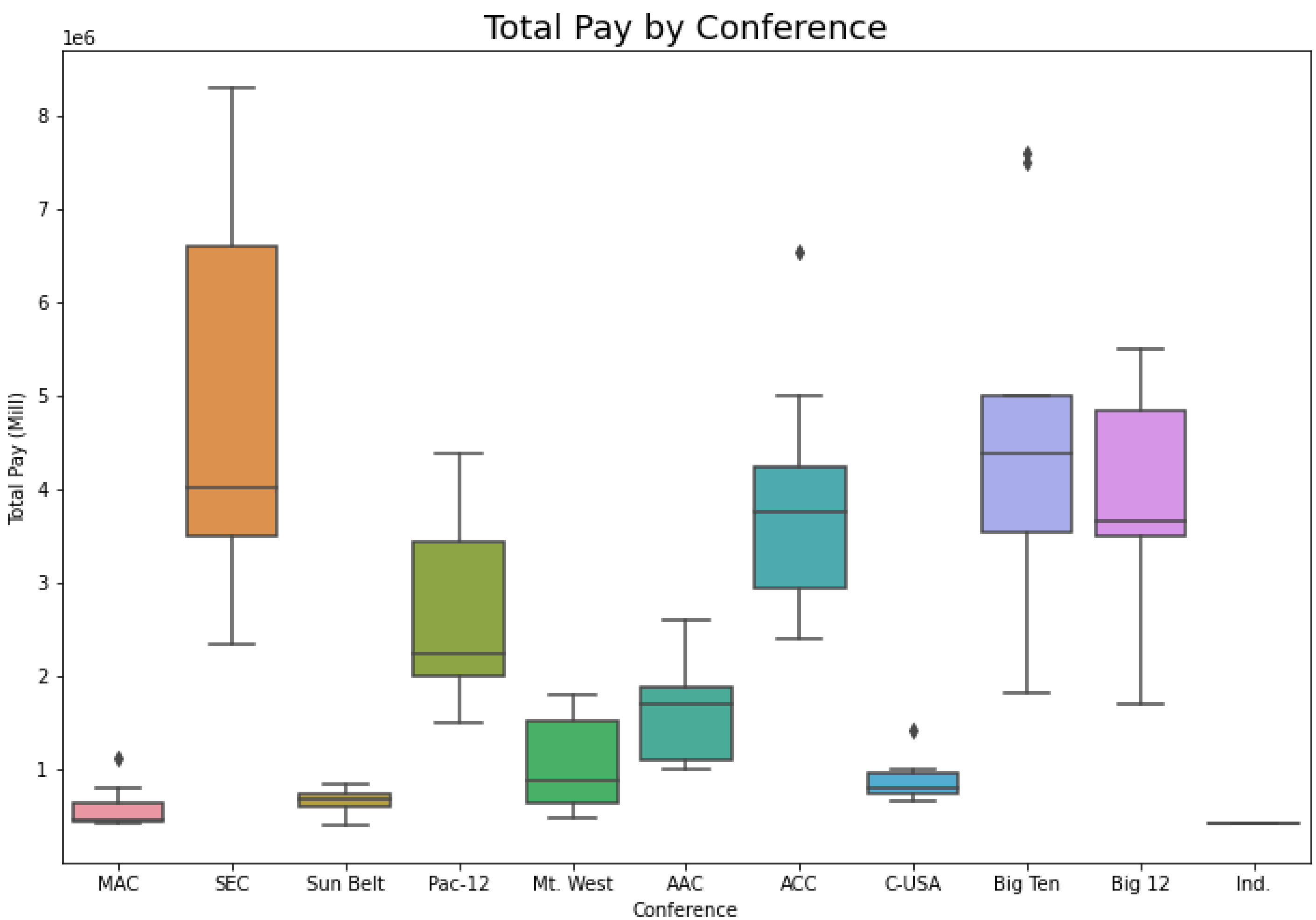
```
temp1.head(10)
#len(temp2)
```

Out[781]:

	School	Conference_x	Coach	TotalPay	Revenue	Expenses	Capacity	conf	gsr	fgr
1	Akron	MAC	Terry Bowden	412500.0	37194485.0	37275978.0	30000.0	Mid-American Conference	75.0	61.0
2	Alabama	SEC	Nick Saban	8307000.0	164090889.0	185317681.0	101821.0	Southeastern Conference	85.0	65.0
4	Appalachian State	Sun Belt	Scott Satterfield	712500.0	37996512.0	37773447.0	24150.0	Sun Belt Conference	75.0	67.0
5	Arizona	Pac-12	Kevin Sumlin	2000000.0	105091389.0	100565835.0	56037.0	Pac-12 Conference	76.0	58.0
6	Arizona State	Pac-12	Herm Edwards	2000000.0	121698840.0	118404377.0	56232.0	Pac-12 Conference	75.0	60.0
7	Arizona State	Pac-12	Herm Edwards	2000000.0	121698840.0	118404377.0	56232.0	Sun Belt Conference	80.0	62.0
8	Arkansas	SEC	Chad Morris	3500000.0	137497788.0	129620361.0	72000.0	Southeastern Conference	67.0	47.0
11	Auburn	SEC	Gus Malzahn	6705656.0	152455416.0	139260711.0	87451.0	Southeastern Conference	76.0	67.0
12	Ball State	MAC	Mike Neu	435689.0	27678480.0	27911602.0	22500.0	Mid-American Conference	73.0	63.0
14	Boise State	Mt. West	Bryan Harsin	1650010.0	50599483.0	49758472.0	37000.0	Mountain West Conference	87.0	63.0

In [782]:

```
#visualize total pay by conference
plt.figure(figsize=(12,8))
coachesbox = sns.boxplot(x="Conference_x",
                         y="TotalPay",
                         data=temp1)
plt.title('Total Pay by Conference', fontsize = 18)
plt.xlabel('Conference')
plt.ylabel('Total Pay (Mill)')
plt.show()
```



In [783]:

```
"""
based on review of the box plot there are conference that will be
meaningless in this analysis. I will be removing the MAC, Conf USA,
Sun Belt and independent schools
"""
cf_data = temp1[temp1.Conference_x != "MAC"]
cf_data = cf_data[cf_data.Conference_x != "C-USA"]
cf_data = cf_data[cf_data.Conference_x != "Sun Belt"]
cf_data = cf_data[cf_data.Conference_x != "Ind."]

cf_data.head(10)
```

Out[783]:

	School	Conference_x	Coach	TotalPay	Revenue	Expenses	Capacity	conf	gsr	fgr
2	Alabama	SEC	Nick Saban	8307000.0	164090889.0	185317681.0	101821.0	Southeastern Conference	85.0	65.0
5	Arizona	Pac-12	Kevin Sumlin	2000000.0	105091389.0	100565835.0	56037.0	Pac-12 Conference	76.0	58.0
6	Arizona State	Pac-12	Herm Edwards	2000000.0	121698840.0	118404377.0	56232.0	Pac-12 Conference	75.0	60.0
7	Arizona State	Pac-12	Herm Edwards	2000000.0	121698840.0	118404377.0	56232.0	Sun Belt Conference	80.0	62.0
8	Arkansas	SEC	Chad Morris	3500000.0	137497788.0	129620361.0	72000.0	Southeastern Conference	67.0	47.0
11	Auburn	SEC	Gus Malzahn	6705656.0	152455416.0	139260711.0	87451.0	Southeastern Conference	76.0	67.0
14	Boise State	Mt. West	Bryan Harsin	1650010.0	50599483.0	49758472.0	37000.0	Mountain West Conference	87.0	63.0
19	California	Pac-12	Justin Wilcox	1500000.0	87500758.0	106676734.0	62717.0	Pac-12 Conference	75.0	62.0
20	Central Florida	AAC	Josh Heupel	1700000.0	69121887.0	67916343.0	45323.0	American Athletic Conference	84.0	64.0
23	Cincinnati	AAC	Luke Fickell	2000000.0	68845672.0	66832326.0	40000.0	American Athletic Conference	85.0	65.0

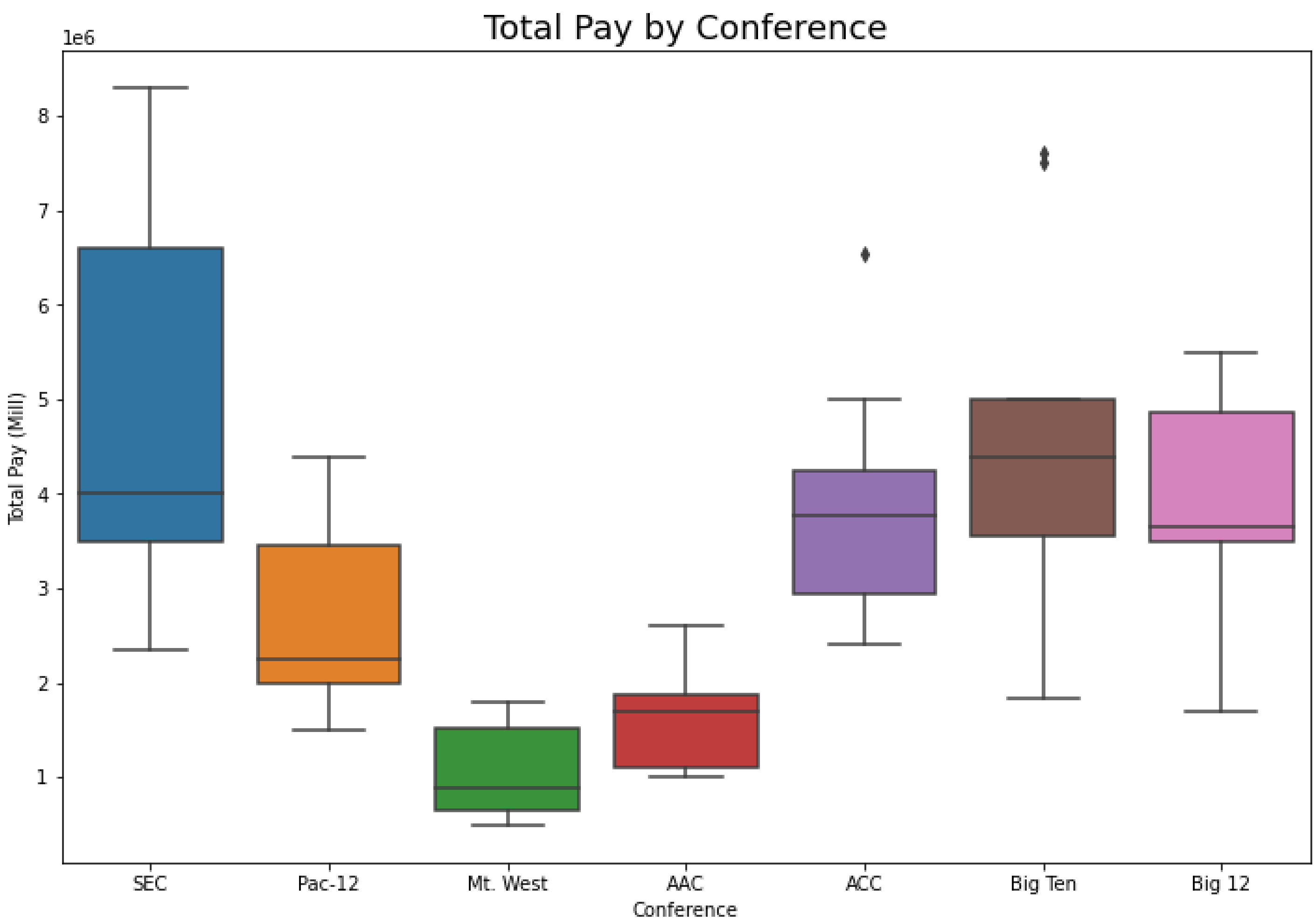
In [784]:
```
#view descriptive stats of the cleaned data
print(cf_data.describe())
```

```
          TotalPay       Revenue      Expenses      Capacity        gsr  \
count  6.900000e+01  6.900000e+01  6.900000e+01      69.000000  69.000000
mean   3.332885e+06  1.113592e+08  1.085395e+08   63800.376812  79.289855
std    1.892218e+06  4.465077e+07  4.194030e+07   21303.282126   8.583674
min    4.865040e+05  3.278738e+07  3.318407e+07   25513.000000  54.000000
25%    1.830000e+06  8.090040e+07  8.081417e+07   50000.000000  75.000000
50%    3.500000e+06  1.084424e+08  1.087859e+08   60862.000000  79.000000
75%    4.377500e+06  1.400109e+08  1.368797e+08   80250.000000  86.000000
max    8.307000e+06  2.238798e+08  2.205730e+08  107601.000000  93.000000

             fgr
count  69.000000
mean   62.086957
std    10.795886
min    31.000000
25%    58.000000
50%    64.000000
75%    70.000000
max    84.000000
```

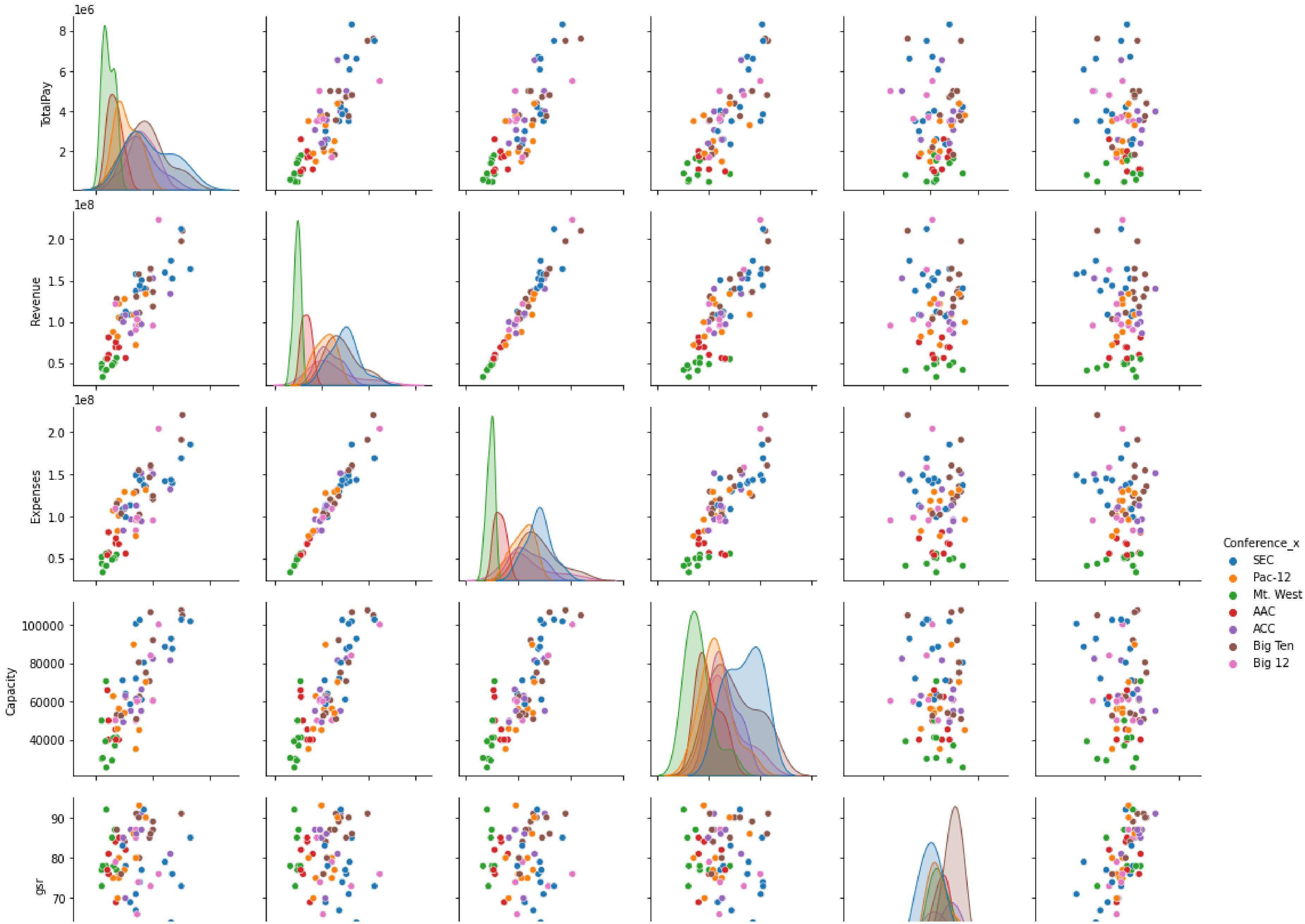
In [785]:

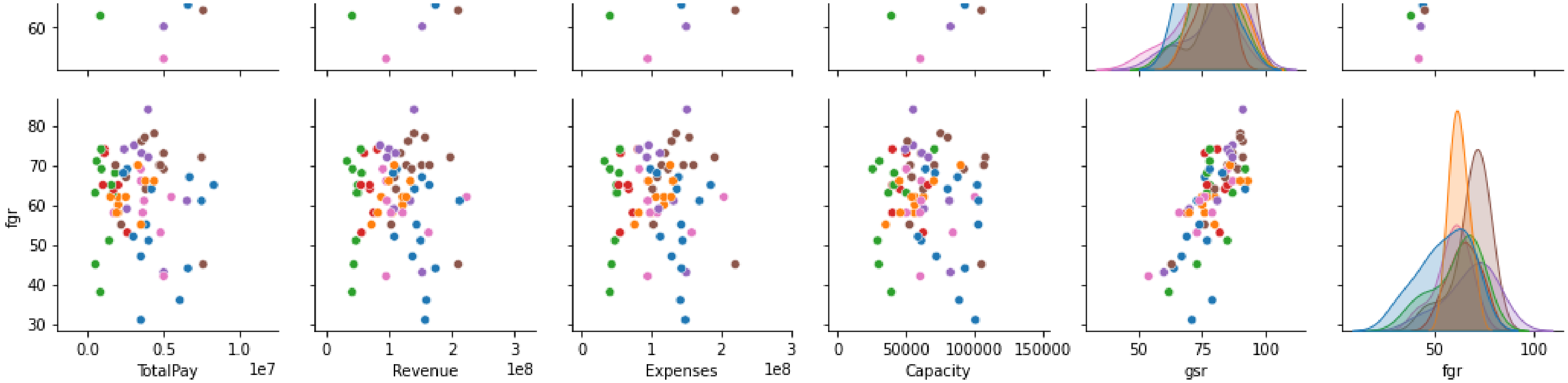
```
#visualize total pay by conference
plt.figure(figsize=(12,8))
coachesbox = sns.boxplot(x="Conference_x",
                         y="TotalPay",
                         data=cf_data)
plt.title('Total Pay by Conference', fontsize = 18)
plt.xlabel('Conference')
plt.ylabel('Total Pay (Mill)')
plt.show()
```



In [786]:
```
#visualize all numeric relationships to understand by conference
#how inputs are related (this seemed easier than doing it 1 by 1)
sns.pairplot(cf_data, hue="Conference_x")
```

Out[786]: <seaborn.axisgrid.PairGrid at 0x1309e5b80>





In [787]:
```python
for col in cf_data.columns:
    print(col)

len(cf_data)
```

```
School
Conference_x
Coach
TotalPay
Revenue
Expenses
Capacity
conf
gsr
fgr
```

Out[787]:
```
69
```

In [788]:

```
#understand the structure of the dataframes
cf_data.info()
temp2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 69 entries, 2 to 129
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   School        69 non-null     object 
 1   Conference_x  69 non-null     object 
 2   Coach         69 non-null     object 
 3   TotalPay      69 non-null     float64
 4   Revenue       69 non-null     float64
 5   Expenses      69 non-null     float64
 6   Capacity      69 non-null     float64
 7   conf          69 non-null     object 
 8   gsr           69 non-null     float64
 9   fgr           69 non-null     float64
dtypes: float64(6), object(4)
memory usage: 8.0+ KB
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60 entries, 1 to 129
Data columns (total 16 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   School        60 non-null     object 
 1   Conference_x  60 non-null     object 
 2   Coach         60 non-null     object 
 3   TotalPay      60 non-null     float64
 4   Revenue       60 non-null     float64
 5   Expenses      60 non-null     float64
 6   Capacity      60 non-null     float64
 7   conf_x        60 non-null     object 
 8   currwin%      60 non-null     object 
 9   win_percent   60 non-null     float64
 10  oc            60 non-null     object 
 11  dc            60 non-null     object 
 12  stc           60 non-null     object 
 13  conf_y        60 non-null     object 
 14  gsr           60 non-null     float64
 15  fgr           60 non-null     float64
dtypes: float64(7), object(9)
memory usage: 8.0+ KB
```

In [789]:
```
#total pay run against revenue and capacity. note, expenses added no additional value to the model
#model_str = ('TotalPay ~ Revenue + Capacity + gsr + fgr')
#^this added no value to the model
model_str = ('TotalPay ~ Revenue + Capacity')
model = smf.ols(model_str, data=cf_data).fit()
model.summary()
```

Out[789]:

OLS Regression Results

Dep. Variable:	TotalPay	**R-squared:**	0.722
Model:	OLS	**Adj. R-squared:**	0.713
Method:	Least Squares	**F-statistic:**	85.57
Date:	Sun, 25 Jul 2021	**Prob (F-statistic):**	4.68e-19
Time:	21:29:29	**Log-Likelihood:**	-1050.6
No. Observations:	69	**AIC:**	2107.
Df Residuals:	66	**BIC:**	2114.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.122e+06	3.88e+05	-2.895	0.005	-1.9e+06	-3.48e+05
Revenue	0.0244	0.005	4.857	0.000	0.014	0.034
Capacity	27.1635	10.549	2.575	0.012	6.103	48.224

Omnibus:	1.635	**Durbin-Watson:**	1.953
Prob(Omnibus):	0.442	**Jarque-Bera (JB):**	1.529
Skew:	0.354	**Prob(JB):**	0.466
Kurtosis:	2.826	**Cond. No.**	3.81e+08

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.81e+08. This might indicate that there are
strong multicollinearity or other numerical problems.

In [790]:
```python
#looping in win percentage
#limitation is a reduced data set
#this is being looked at by all conferences
#model_str = ('TotalPay ~ Revenue + Capacity + win_percent + gsr + fgr')
#^this added no value to the model
model_str = ('TotalPay ~ Revenue + Capacity + win_percent')
model = smf.ols(model_str, data=temp2).fit()
model.summary()
```

Out[790]:

OLS Regression Results

Dep. Variable:	TotalPay	**R-squared:**	0.853
Model:	OLS	**Adj. R-squared:**	0.845
Method:	Least Squares	**F-statistic:**	108.3
Date:	Sun, 25 Jul 2021	**Prob (F-statistic):**	2.69e-23
Time:	21:29:29	**Log-Likelihood:**	-902.59
No. Observations:	60	**AIC:**	1813.
Df Residuals:	56	**BIC:**	1822.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-2.494e+06	5.38e+05	-4.638	0.000	-3.57e+06	-1.42e+06
Revenue	0.0190	0.006	3.311	0.002	0.008	0.031
Capacity	31.5103	11.948	2.637	0.011	7.575	55.445
win_percent	3.128e+06	1.02e+06	3.054	0.003	1.08e+06	5.18e+06

Omnibus:	0.301	**Durbin-Watson:**	2.015
Prob(Omnibus):	0.860	**Jarque-Bera (JB):**	0.079
Skew:	-0.087	**Prob(JB):**	0.961
Kurtosis:	3.037	**Cond. No.**	1.05e+09

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.05e+09. This might indicate that there are
strong multicollinearity or other numerical problems.

In [791]:
```
"""
Given a coach could be hired from outside power 5
minor conferences, from the NFL, etc.
I am going to use all conferences but leverage
the win percentage field as it seems to have added
more to the model than isolating major conferences
"""
```

Out[791]:
```
'\nGiven a coach could be hired from outside power 5\nminor conferences, from the NFL, etc. \nI am going to us
e all conferences but leverage\nthe win percentage field as it seems to have added \nmore to the model than is
olating major conferences\n'
```

In [792]:
```
temp2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60 entries, 1 to 129
Data columns (total 16 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   School        60 non-null     object 
 1   Conference_x  60 non-null     object 
 2   Coach         60 non-null     object 
 3   TotalPay      60 non-null     float64
 4   Revenue       60 non-null     float64
 5   Expenses      60 non-null     float64
 6   Capacity      60 non-null     float64
 7   conf_x        60 non-null     object 
 8   currwin%      60 non-null     object 
 9   win_percent   60 non-null     float64
 10  oc            60 non-null     object 
 11  dc            60 non-null     object 
 12  stc           60 non-null     object 
 13  conf_y        60 non-null     object 
 14  gsr           60 non-null     float64
 15  fgr           60 non-null     float64
dtypes: float64(7), object(9)
memory usage: 8.0+ KB
```

In [793]:
```
#syr projected salary
syr = temp2.loc[temp2['School'] == 'Syracuse']
model.predict(syr)
```

Out[793]:
```
103    2.468638e+06
dtype: float64
```

In [794]:
```
#view of the big 10
syr = temp2.loc[temp2['Conference_x'] == 'Big Ten']
model.predict(syr)
```

Out[794]:
```
44     3.241235e+06
45     4.540959e+06
63     6.680307e+06
71     4.668913e+06
89     6.064071e+06
91     3.393591e+06
128    5.115489e+06
dtype: float64
```

In [802]:
```
temp2.loc[temp2['School'] == 'Syracuse', 'Conference_x'] = "Big Ten"
```

In [803]:
```
temp2[temp2['School'] == 'Syracuse']
```

Out[803]:

	School	Conference_x	Coach	TotalPay	Revenue	Expenses	Capacity	conf_x	currwin%	win_percent	oc	dc	stc	conf_y	
103	Syracuse	Big Ten	Dino Babers	2401206.0	99800000.0	82900000.0	49250.0	ACC	0.4	0.483	Sterlin Gilbert	Tony White	Vacant	Atlantic Coast Conference	8

In [804]:

```
# run regression model with dummy big ten input
syr = temp2.loc[temp2['School'] == 'Syracuse']
model.predict(syr)
"""
I realized after running this model I needed to have created
several new columns and mark conference with a 1 or 0
depending on the membership then run that as a variable in a
logit regression. I have run out of time to pull that off, however...
So I decided to proxy syracuse against Indiana University as they
were very similar in terms of stadium size, revenue, win% and
graduation rates.
"""
```

Out[804]:

```
'\nI realized after running this model I needed to have created \nseveral new columns and mark conference with
a 1 or 0 \ndepending on the membership then run that as a variable in a \nlogit regression. I have run out of
time to pull that off, however...\nSo I decided to proxy syracuse against Indiana University as they \nwere ve
ry similar in terms of stadium size, revenue, win% and\ngraduation rates.\n'
```

In [805]:

```
# run regression model with dummy big ten input
syr_proxy = temp2.loc[temp2['School'] == 'Indiana']
model.predict(syr_proxy)
```

Out[805]:

```
44    3.241235e+06
dtype: float64
```

In [806]:

```
#What is the recommended salary for the Syracuse football coach?
    #The recommended salary for the Syracuse football coach is:
        #$2.46 Million
        #$2.25 Million looks to be actual salary...
#What would his salary be if we were still in the Big East? What if we went to the Big Ten?
    #Using Indiana University as a proxy recommended salary for the Syracuse football coach is:
        #$3.24 Million
#What schools did we drop from our data and why?
    #I didn't just drop schools I dropped entire conferences to try interpret the data...
    #after viewing some of the visuals some of the conferences looked to be meaningless
    #in comparison to Syracuse's conference (ACC). To be specific Baylor, BYU and SMU
    #were always dropped across all models as total pay was missing
#What effect does graduation rate have on the projected salary?
    #Based on the mix of inclusion and exclusion of graduation rates it proved to be a
    #largely insignificant variable and did not add much to any model judging by its
    #impact on the R^2 and adj. R^2
#How good is our model?
    #Reasonable - the R^2 says that ~85% of the variance is explained in my better model
#What is the single biggest impact on salary size?
    #Stadium size and athletic budget seemed to be the two largest impacts on salary size
    #I mention two because in isolation they were virtually equal in impact. Intuitively,
    #these make sense as bigger budget schools with big fan bases may have a larger
    #propensity to spend money on a coach's salary
```

In [ ]:

```
```

In [ ]:

```
```