#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

#load baseline coaches file

coaches = pd.read\_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/coaches\_baseline.csv')

coaches.head(10)

#load ancilary revenue data

revenue = pd.read\_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/revenue.csv')

revenue.head(10)

#load ancilary stadium size data

size = pd.read\_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/Stadium\_Size.csv')

size.head(10)

#load ancilary coach win data

coach = pd.read\_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/coach\_win.csv')

coach.head(10)

#load ancilary grad rate win data

grad = pd.read\_csv('https://raw.githubusercontent.com/barrettfranks/ist718/master/grad\_rates.csv')

grad.head(10)

#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)

for col in temp1.columns:

print(col)

#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(',','')).astype(float)

temp1['TotalPay'] = temp1['TotalPay'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype(float)

temp2['Revenue'] = temp2['Revenue'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype(float)

temp1['Revenue'] = temp1['Revenue'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype(float)

temp2['Expenses'] = temp2['Expenses'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype(float)

temp1['Expenses'] = temp1['Expenses'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype(float)

temp1['Capacity'] = temp1['Capacity'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype(float)

temp2['Capacity'] = temp2['Capacity'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype(float)

temp2['win\_percent'] = temp2['win\_percent'].apply(lambda x: x.replace('$', '')).apply(lambda x: x.replace(',','')).astype(float)

for col in temp1.columns:

print(col)

len(temp1)

temp1.head(10)

#len(temp2)

#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()

"""

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)

#view descriptive stats of the cleaned data

print(cf\_data.describe())

#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()

#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")

for col in cf\_data.columns:

print(col)

len(cf\_data)

#understand the structure of the dataframes

cf\_data.info()

temp2.info()

#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()

#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()

"""

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

"""

#syr projected salary

syr = temp2.loc[temp2['School'] == 'Syracuse']

model.predict(syr)

#view of the big 10

syr = temp2.loc[temp2['Conference\_x'] == 'Big Ten']

model.predict(syr)

temp2.loc[temp2['School'] == 'Syracuse', 'Conference\_x'] = "Big Ten"

temp2[temp2['School'] == 'Syracuse']

# 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.

"""

# run regression model with dummy big ten input

syr\_proxy = temp2.loc[temp2['School'] == 'Indiana']

model.predict(syr\_proxy)

#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