Big Data Analytics Homework One

Allison R. Hollingsworth Deming

IST 718: Big Data Analytics

Homework One Report

Introduction

Many things change in the Fall, especially in a college town. First, the population swells from the students returning to town, who are new to their alma mater and excited about everything the college experience brings. Other students have been at the school for weeks already. These are the athletes, particularly the football players, trainers, coaches, and significant support staff. They have spent the last few scorching weeks of Summer honing their craft before the football season itself starts. This is generally around the second week after school begins. This is far from being just a sport for student-athletes; for many, it is religion, and for most Division I schools, it is also big business. Depending on the school, the salary for the head coach can range from just under $100,000 to well over ten million per year. This depends on many things, but how does a school appropriately pick a salary for this vital position? With football bringing in large amounts of cash not just for its own operations but for the entire school, it is essential to get the right person at the right price. Here, multiple models will be produced to estimate what coaches in each conference should be paid, and more specifically, what Syracuse's Coach, Dino Babers, should make at Syracuse and what he could make elsewhere.

Obtaining the Data

More is needed than just the dollars and cents of a wage to develop a model to predict the best salary for a head football coach. Certain schools may have caps to which they can pay their coach. Others with more competitive and extensive programs might negotiate contracts over multiple years with additional bonuses based on performance. One recently published example of this complexity is that of Dabo Swinney at Clemson University. He joined the school as the head football coach in 2008 with a team that had a win percentage of about 70%. Over the next ten-plus years, that percentage went up to 85%. In 2022, Coach Swinney signed a ten-year, $115 million contract to keep him at Clemson. That comes to about $15 million per year plus a bonus structure. The bonus structure includes $50,000 for playing in the ACC Championship Game (their conference), $200,000 for winning this game, and $75,000 each year for winning at least eight games. Playing in a New Year's non-CFP (College Football Playoff) semifinal bowl: $150,000; for a CFP appearance, another $150,000; for a CFP semifinal appearance: $250,000; for a CFP championship appearance: $250,000; for a CFP championship win: $350,000 Single-year academic progress rate (APR) ≥ 950: $75,000; or Single-year APR ≥ 975: $100,000; if awarded ACC Coach of the Year Award: $25,000, and lastly if he receives the National Coach of the Year Award: $50,000. This shows how complex the bonus structure can be for a single coach. If Clemson decides to part ways with Dabo Swinney during this contract, toward the beginning of the ten years, Clemson will pay $64 million, which decreases to a low of $57 million toward the end of the contract. Coaches may also earn money through endorsements, licensing, appearances, and many things not factored into this analysis. In addition to the salary data, our study incorporates three different data sets: college graduation rate, football stadium, and the football teams' wins and losses since 2010. It should be noted that the most recent years are not included in the data, but where that data is to be obtained, it could be filtered into the model for more appropriate (up-to-date) results.

The recent data about Coach Swinney's contract is on the higher end of the pay scale, but that is not why it is included. It is included because it gives a basis for the data sets chosen to put into our model. Each of these data sets can be seen reflected in the 2022 Swinney contract. It shows what is essential to at least one school within the group we are analyzing. Additionally, it gives justification for win percentage and graduation rate being included along with the other monetary variables and how those are weighted within the overall contract.

Data Set One: The professor posted the Coaches' data on the following website: https://github.com/2SUBDA/IST_718. This has 129 entries and eight columns. The columns have variables related to the school's football teams, such as School (The identity of the school), Conference (with which group they compete during the regular season), Coach (designated head coach), SchoolPay (the school pay), TotalPay (total pay of the coach), Bonus (any bonus pay contracted), BonusPaid (actual bonus paid out, generally based on performance that year), AssistantPay (what the assistants are paid), and Buyout (the amount the coach gets if the school decides to buy out their contract).

Data Set Two: The colleges' graduation dataset is available at the following website: https://www.ncaa.org/sports/2016/12/14/shared-ncaa-research-data.aspx. The data set contains columns detailing the colleges' football division, which depends on size and contracts. Additional data contained therein is if the school is private or public. The next piece designates historically black colleges and universities (HBCUs) and the graduation rate of their different sports programs. The model produced will only include that of the football teams.

Data Set Three: The football stadium dataset is also from GitHub: https://raw.githubusercontent.com/gboeing/data-visualization/main/ncaa-footballstadiums/data/stadiums-geocoded.csv. This dataset contains variables that describe stadium features such as the city, state, stadium name, capacity, build, and location.

Data Set Four: The football teams' win and loss percentages are available at: https://www.teamrankings.com/ncf/trends/win_trends/?range=yearly_since_2010. The dataset includes the teams and their win-loss record overall.

Exploratory Data Analyses

The four datasets were imported into Anaconda with Python using the PANDAS package. When each of the individual datasets was processed and explored, changing the strings' columns to integer values. Overall, the data frames contained integers, objects, and floats. Lastly, NaN values were changed to dashes for easier processing. Each data frame requires a lot of filtering of columns to those that are needed for the analysis and those that are not. The exception is the wins and losses data frame, which contains the historical data of wins and losses for each team; all columns were kept and imported. The stadium dataset tells the size and upgrades to the stadiums and multiple columns that indicate the school's investment in the football team facilities. Lastly, the graduation set gives the graduation rate of student-athletes, but the only part that will be used in this analysis is that of the football players.

```
stadium_data = pd.read_csv('https://raw.githubuser
print(stadium_data.info()) #printing
stadium_data.head() #What top 10 rows look like
```

✓ 0.5s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253 entries, 0 to 252
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   stadium     253 non-null    object
 1   city        253 non-null    object
 2   state       253 non-null    object
 3   team        253 non-null    object
 4   conference  253 non-null    object
 5   capacity    253 non-null    int64
 6   built       253 non-null    int64
 7   expanded    146 non-null    object
 8   div         253 non-null    object
 9   latitude    253 non-null    float64
 10  longitude   253 non-null    float64
dtypes: float64(2), int64(2), object(7)
memory usage: 21.9+ KB
None
```

Figure 1: Stadium Data Column Names and Types

	stadium	city	state	team	conference	capacity	built	expanded	div	latitude	longitude
0	Michigan Stadium	Ann Arbor	MI	Michigan	Big Ten	107601	1927	2015	fbs	42.265869	-83.748726
1	Beaver Stadium	University Park	PA	Penn State	Big Ten	106572	1960	2001	fbs	40.812153	-77.856202
2	Ohio Stadium	Columbus	OH	Ohio State	Big Ten	104944	1922	2014	fbs	40.001686	-83.019728
3	Kyle Field	College Station	TX	Texas A&M	SEC	102733	1927	2015	fbs	30.610098	-96.340729
4	Neyland Stadium	Knoxville	TN	Tennessee	SEC	102455	1921	2010	fbs	35.954734	-83.925333

Figure 1.1: First Four Rows of Stadium Data

```
coach_data = pd.read_csv('https://raw.git
print(coach_data.info()) #printing

coach_data = coach_data.rename(columns={'
coach_data.head() #First 10 rows of df
```

✓ 0.3s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129 entries, 0 to 128
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   School        129 non-null    object
 1   Conference    129 non-null    object
 2   Coach         129 non-null    object
 3   SchoolPay     129 non-null    object
 4   TotalPay      129 non-null    object
 5   Bonus         129 non-null    object
 6   BonusPaid     129 non-null    object
 7   AssistantPay  129 non-null    object
 8   Buyout        129 non-null    object
dtypes: object(9)
memory usage: 9.2+ KB
None
```

Figure 1.2: Columns and Rows of the Coach Data

	team	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

Figure 1.3: First Five Rows of the Coach Data Frame

```
wins_losses = wins_losses.rename(columns={'Team':'t
wins_losses.head() #First rows
```

✓ 0.8s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 131 entries, 0 to 130
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Team             131 non-null    object
 1   Win-Loss Record  131 non-null    object
 2   Win %            131 non-null    object
 3   MOV              131 non-null    float64
 4   ATS +/-          131 non-null    float64
dtypes: float64(2), object(3)
memory usage: 5.2+ KB
None
```

	team	Win-Loss Record	Win %	MOV	ATS +/-
0	Alabama	161-19-0	89.4%	24.4	2.2
1	Ohio State	146-23-0	86.4%	20.8	2.8
2	Clemson	147-31-0	82.6%	17.3	1.8
3	Oklahoma	135-36-0	79.0%	14.5	-0.6
4	Georgia	137-39-0	77.8%	15.9	1.2

Figure 1.4: Wins and Losses Data Frame Information

```
grad_data = pd.read_csv('https://ncaaorg.s3.amazonaws.
print(grad_data.info()) #Printing
grad_data.head() #Taking a look at the first few rows
```

✓ 1.6s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5403 entries, 0 to 5402
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   SCL_UNITID          5403 non-null   int64
 1   SCL_NAME            5403 non-null   object
 2   SCL_DIVISION        5403 non-null   int64
 3   SCL_SUBDIVISION     5403 non-null   int64
 4   SCL_CONFERENCE      5403 non-null   object
 5   DIV1_FB_CONFERENCE  4005 non-null   object
 6   SCL_HBCU            5403 non-null   int64
 7   SCL_PRIVATE         5403 non-null   int64
 8   SPORT               5403 non-null   object
 9   SPONSORED           5403 non-null   int64
 10  FED_RATE            5093 non-null   float64
 11  GSR                 5264 non-null   float64
dtypes: float64(2), int64(6), object(4)
memory usage: 506.7+ KB
None
```

Figure 1.5: Graduation Date Columns and Type

SCL_UNITID	SCL_NAME	SCL_DIVISION	SCL_SUBDIVISION	SCL_CONFERENCE	DIV1_FB_CONFERENCE	SCL_HBCU	SCL_PRIVATE	SPORT	SPONSORED	FED_RATE	GSR
100654	Alabama A&M University	1	2	Southwestern Athletic Conf.	Southwestern Athletic Conf.	1	0	WSB	1	57.0	61.0
100654	Alabama A&M University	1	2	Southwestern Athletic Conf.	Southwestern Athletic Conf.	1	0	MFB	1	47.0	62.0
100654	Alabama A&M University	1	2	Southwestern Athletic Conf.	Southwestern Athletic Conf.	1	0	WSO	1	50.0	67.0
100654	Alabama A&M University	1	2	Southwestern Athletic Conf.	Southwestern Athletic Conf.	1	0	MSO	0	29.0	NaN
100654	Alabama A&M University	1	2	Southwestern Athletic Conf.	Southwestern Athletic Conf.	1	0	WBW	1	86.0	86.0

Figure 1.6: Graduation Rate Data Frame Information

Building a Uniquely Combined and Clean Data Frame

Building this data frame was complex. It turned out that merging all four original datasets into a new data frame was the best idea. Several columns needed to be renamed to merge with the other data frames' columns properly. For example, the “School” column needed to be changed to “team” in the Coaches dataset to match the other data frames. Each had a column labeled “team” with the same information. Next, the SPORT column was filtered to the MFB column to capture male football players. The column SCL_NAME needed to be scrubbed of common phrases such as “University” and “of” along with punctuation, then renamed as “team” like the other data frames. Everything was merged, and the data frame needed for this analysis was completed. Some schools ended up dropped from the data frame following the merge. The reason for this is unknown, but the number was small and likely due to inconsistencies between the different data frames. In the end, there were 50 rows and 25 columns with the following named columns: TEAM, DESCRIPTION OF TEAM, STADIUM, CITY, STATE, WIN-LOSS RECORD, and WIN PERCENTAGE.

```
Data columns (total 26 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   team             50 non-null     object
 1   Conference       50 non-null     object
 2   Coach            50 non-null     object
 3   SchoolPay        50 non-null     object
 4   TotalPay         50 non-null     object
 5   Bonus            50 non-null     object
 6   BonusPaid        50 non-null     object
 7   AssistantPay     50 non-null     object
 8   Buyout           50 non-null     object
 9   SCL_PRIVATE      50 non-null     int64
 10  FED_RATE         50 non-null     float64
 11  GSR              50 non-null     float64
 12  stadium          50 non-null     object
 13  city             50 non-null     object
 14  state            50 non-null     object
 15  conference       50 non-null     object
 16  capacity         50 non-null     int64
 17  built            50 non-null     int64
 18  expanded         39 non-null     object
 19  div              50 non-null     object
```

Figure 2: Final Data Frame Columns and Type (Partial)

Scrubbing the Data Frame

Additional scrubbing needs to be performed before the final data frame is ready to use. For example, the columns which should be numbers such as ‘Totalpay’, ‘Schoolpay’, “Bonus’, ‘BonusPaid’, ‘AssistantPay’, ‘Buyout’, and “Win %’all need to be numeric columns, Also ‘Win %’ needs to be divided by ten to get what is needed for the analysis. Three functions were made for scrubbing. The first will call items column names that contain NaN or empty row values.

```
def index_find_isnull(dataframe, col1, col2): # This function will show find the index of a unavailable records in the first specified column.
    d2 = dataframe.loc[pd.isnull(dataframe[col1]),[col1,col2]]
    return d2
```

Figure 3: Function 1 for Scrubbing

```
index_find_isnull(cleanMerge,'TotalPay','Conference') #Looking for the index of the missing data in TotalPay
```

	TotalPay	Conference
5	NaN	Big 12
35	NaN	C-USA

Figure 3.1: Columns with NaN Rows

The second function that was made takes those NaN values from the numeric data frames and replaces them with the mean of the conference. This is true except for the Independent Conference, which is alone in its category. In this situation, it was determined that a combined mean of all teams would be best for NaN values.

```
def totalpay_clean(dataframe, TotalPay, Conference, index): #Basically doing the same thing with TotalPay Here
    mean2 = dataframe[TotalPay].loc[(dataframe['Conference'] == Conference) & (dataframe[TotalPay].notna())].mean()

    dataframe.loc[index,TotalPay]=mean2
```

Figure 4: Function 3 Fill s NaN Rows

The third and final function that was made for data scrubbing out new and improved combined data frame uses our second function and does that final filling of NaN values in the columns with the appropriate mean discussed above. As can be seen from the code, 'mean2'is calculated from the TotalPay and Conference variables. Once complete, all three functions have been run through those money variables that need to be numeric and cleaned, such as TotalPay and Bonus.

```
def bonus_clean(dataframe, Bonus, Conference, index):
    mean2 = dataframe[Bonus].loc[(dataframe['Conference'] == Conference) & (dataframe[Bonus].notna())].mean()

    dataframe.loc[index,Bonus]=mean2
```

Figure 5: Function That Cleans and Replaces the Bonus Using 'mean2' and Conference

```
cleanMerge
cleanMerge.info() #Taking a Look
```

Figure 6: Using Cleanmerge to Take a Look at our Columns, Type, and NaNs in the Data Frame

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 26 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   team           50 non-null     object
 1   Conference     50 non-null     object
 2   Coach          50 non-null     object
 3   SchoolPay      48 non-null     float64
 4   TotalPay       50 non-null     float64
 5   Bonus          50 non-null     float64
 6   BonusPaid      34 non-null     float64
 7   AssistantPay   50 non-null     int64
 8   Buyout         39 non-null     float64
 9   SCL_PRIVATE    50 non-null     int64
 10  FED_RATE       50 non-null     float64
 11  GSR            50 non-null     float64
 12  stadium        50 non-null     object
 13  city           50 non-null     object
 14  state          50 non-null     object
 15  conference     50 non-null     object
 16  capacity       50 non-null     int64
 17  built          50 non-null     int64
 18  expanded       39 non-null     object
 19  div            50 non-null     object
 20  latitude       50 non-null     float64
 21  longitude      50 non-null     float64
```

Visual and Numeric Exploratory Data Analysis

Sometimes the best way to see how your data is distributed is to plot the data quickly and eloquently. Early in this project, variables were looked at in one or two dimensions only to see the distributions and variability of a coach's salary within or between the different conferences. By looking at the information, those conferences, such as the SEC with large franchises, spend tons of money and value their football program. These programs include but are not limited to, Alabama, Ole Miss', Florida State, and Florida. While other powerhouse franchises (such as the example discussed above at Clemson above) exist in other conferences, there seems to be a culture of more significance being better in all areas of the SEC. The ACC has a lower salary average than the SEC. That is interesting to know, but what is more interesting is if the full coach's salary affects the winning percentage and vice versa. Below is a stacked bar chart that is colored by conference, has an x-axis of the coach's total pay (divided by $100,000), and a y-axis of schools count.

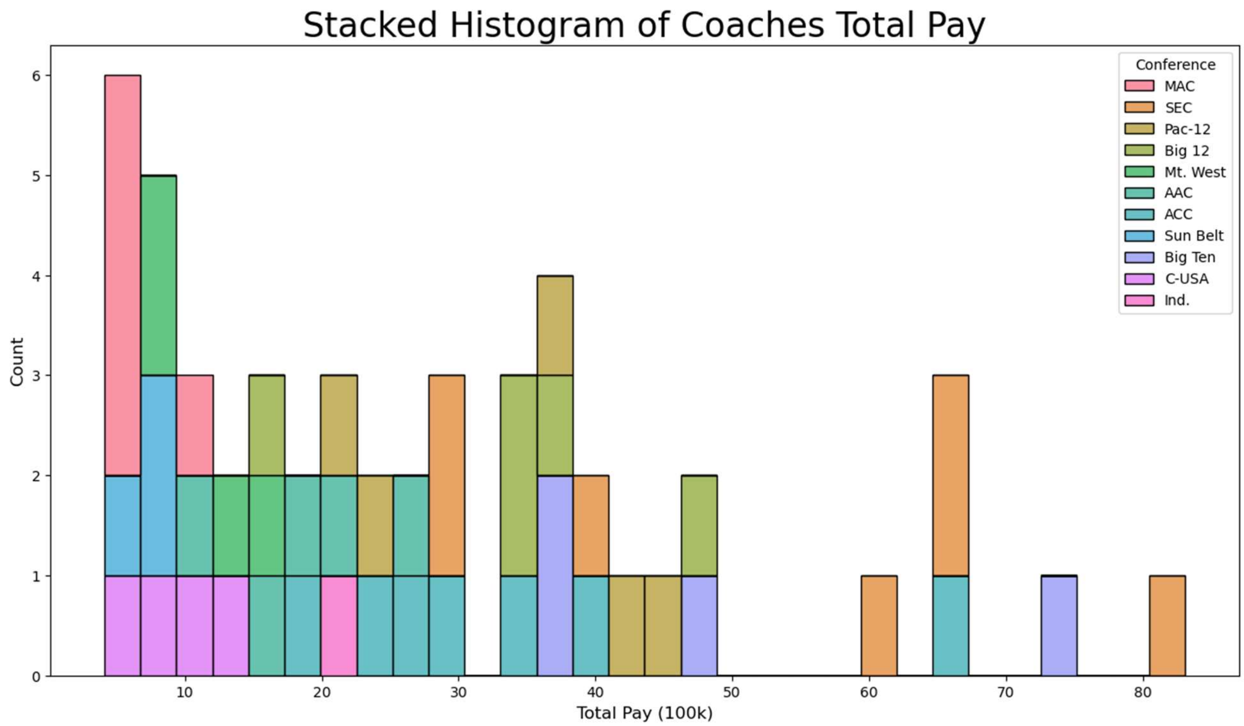

Figure 7: Stacked Histogram of Coach's Salaries by Conference

The first and most obvious observation is that there is a significant right skew of the data, with several outliers on the right. The four most right-hand columns are the highest-paid coaching jobs, and the burned orange color filling makes up most of these high salary bars.

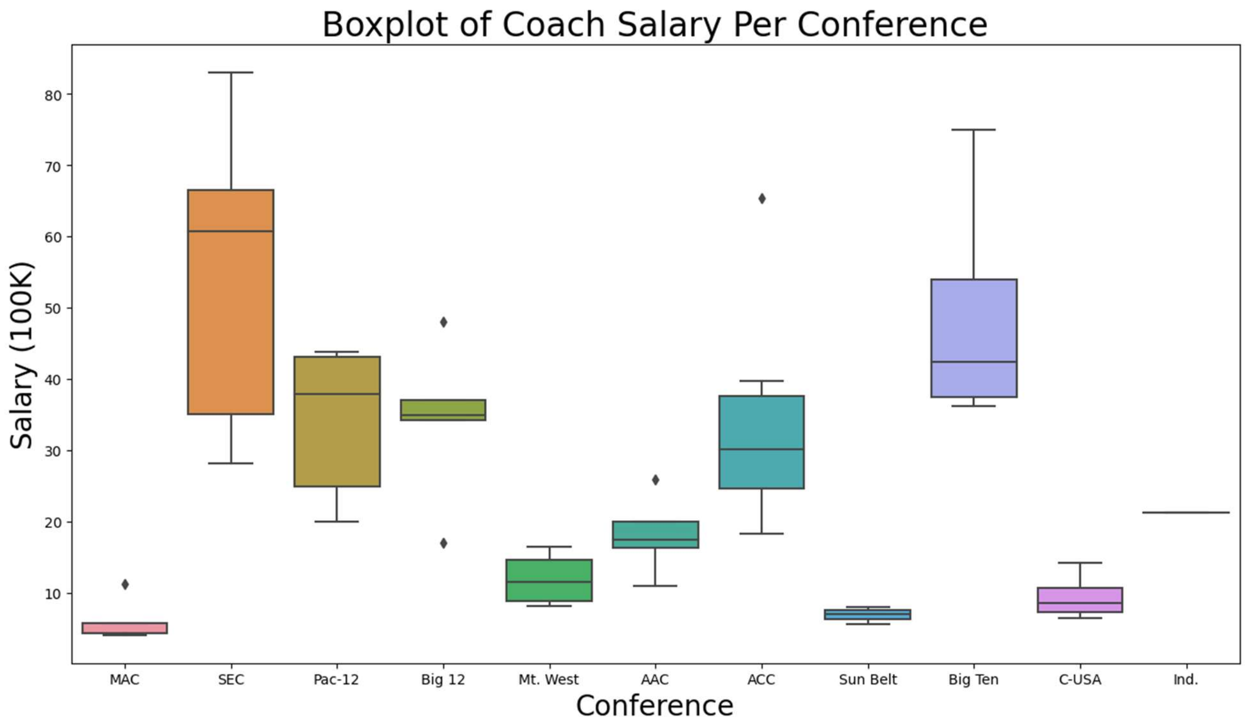

Figure 7.1: Boxplot of Coach's Salary By Conference

Above, in Figure 7.1 is a Boxplot of each conference on the x-axis and salary divided by 100K on the left-hand or y-axis. The thick lines in the middle of each rectangular box represent the median of the data rather than the mean; This makes for easy comparisons across the conferences for the middle salary. This is less susceptible to outliers when compared to the mean. The ACC, Big Ten, and Pac-12 have the highest three medians in this representation; It is simple to visualize what each conference is paying with this type of graph.

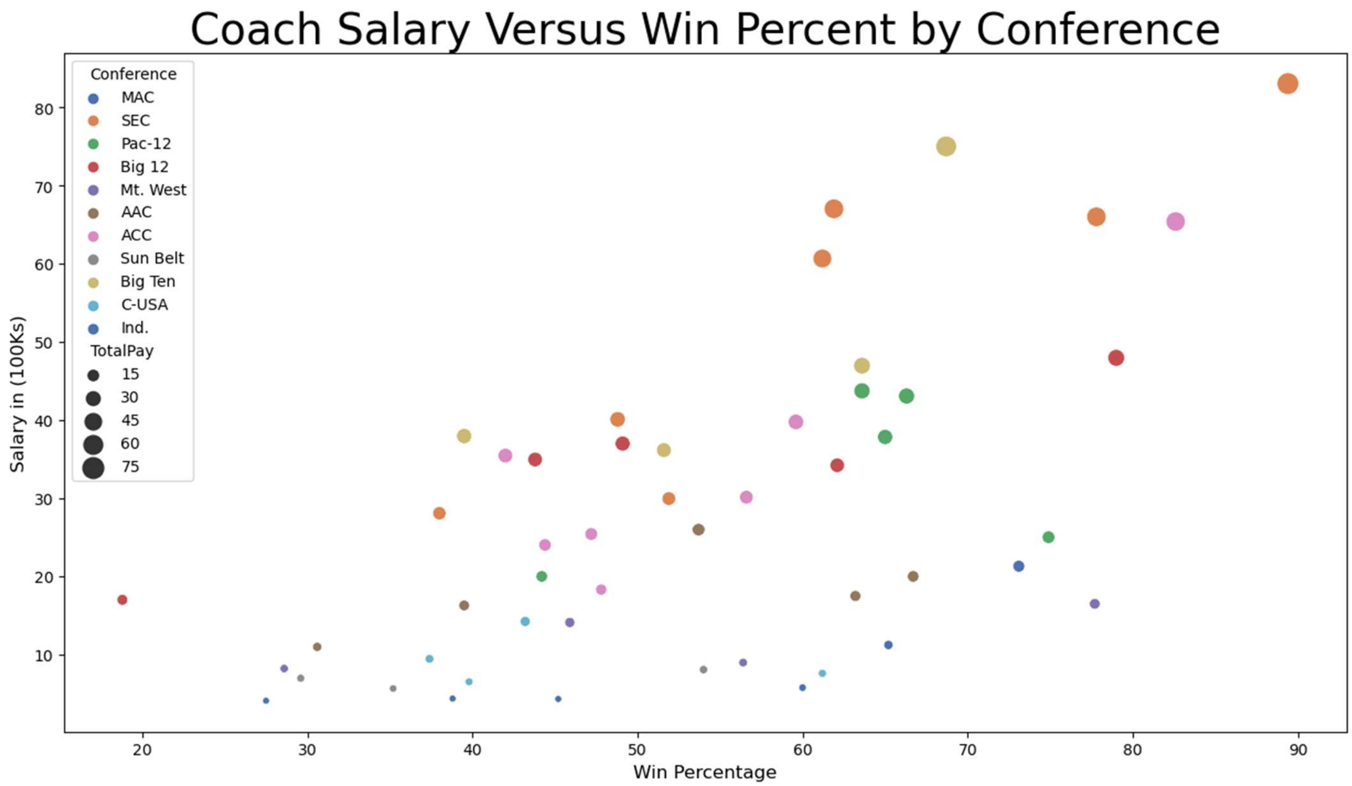

Figure 7.2: Coach Salary Versus Win Percentage by Conference

This plot is a scatterplot of the win percentage on the x-axis and the coach's salary on the y-axis. Encoded within the color of the data is the conference the school belongs to, and the size of the dot is the Total Pay variable for the coaches. The first thing seen is a linear relationship that looks moderately positive, with a low win percentage predicting a lower salary. The higher the win percentage, the more one sees a higher salary. This plot provides a good start for a linear model down the road.

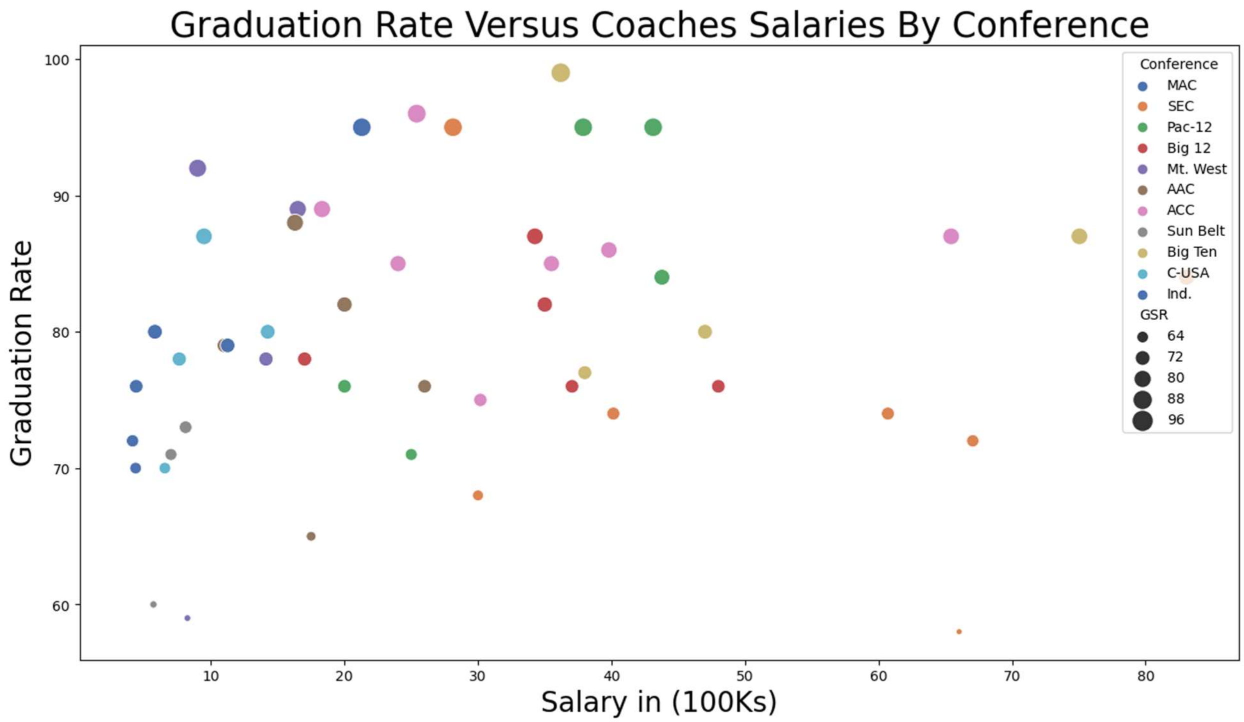

Figure 8: Graduation Rate Versus Salaries By Conference

This plot is like the one above, except the win percentage is removed, and the Graduation Rate is substituted. In the Clemson contract reviewed above, Swinney would receive bonuses for improving academic performance amongst his players. This is relatively rare regarding the importance of most big-name coaches and schools. It should be noted that the program at Clemson works with many NFL-bound students graduating in three years and another large group that completes a master's degree in addition to their bachelor's all in four years before eligibility runs out. Our plot does not show a linear relationship, but with many more variables, such as Public or Private, something could still be found hidden in this data. Further explanation is needed.

Correlation and Regression Models

As mentioned above, some things may be hidden in our dataset. It is known that there is a linear relationship between the primary variable we are trying to predict (the coach's salary) and at least one of the other variables (win percentage). Now, a correlation matrix can be run to show how all variables relate to each other in terms of correlation strength and direction.

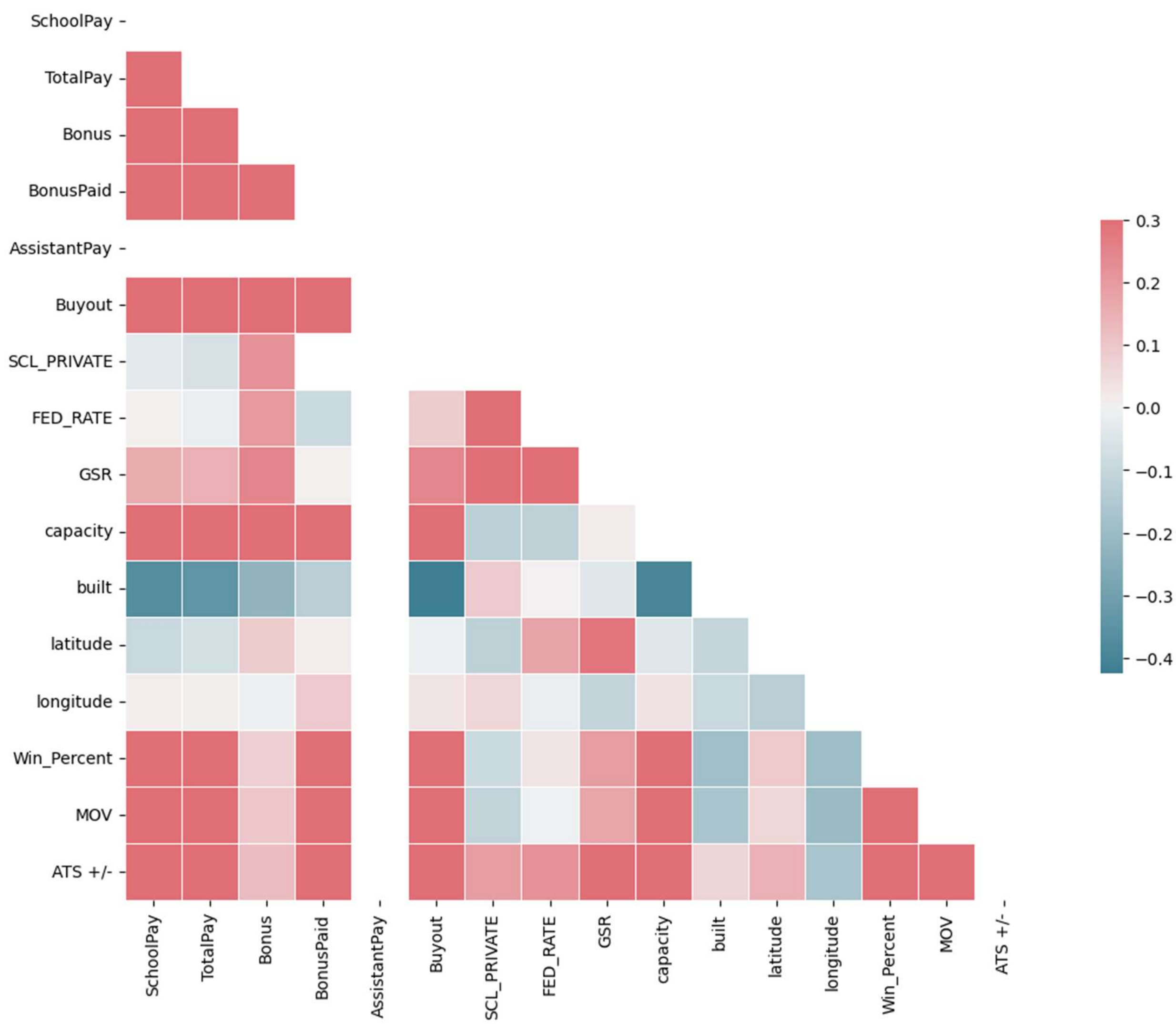

Figure 9: Correlation Matrix of the Data Frame

Modeling the Data

Three models were used to investigate the compiled data frame of coaches' salaries in more depth. This was done with multivariate regression, with one output variable and multiple input variables. The first one used TotalPay as the main variable we are trying to predict. The predictor variables were Win Percentage, SCL_PRIVATE (whether a school is public or private), GSR (graduation rate of football players), and Bonus. The results can be seen in Figure 10.0. Overall, the Adjusted R Squared, or variance accounted for in TotalPay by the predictor variables, was 58.4%. When fit to the testing data compiled earlier in the analysis, the accounted-for variance was shown to be .329.

```
[5 rows x 27 columns]
| | | | | | |     OLS Regression Results
==============================================================================
Dep. Variable:               TotalPay   R-squared:                       0.584
Model:                            OLS   Adj. R-squared:                  0.531
Method:                 Least Squares   F-statistic:                     11.21
Date:                Sun, 29 Jan 2023   Prob (F-statistic):           8.46e-06
Time:                        19:06:30   Log-Likelihood:                -150.34
No. Observations:                  37   AIC:                             310.7
Df Residuals:                      32   BIC:                             318.7
Df Model:                           4
Covariance Type:            nonrobust
===============================================================================
| | | | |     coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept     -30.3692     23.262     -1.306      0.201     -77.753      17.014
Win_Percent     0.8803      0.184      4.784      0.000       0.505       1.255
SCL_PRIVATE    -2.4238      9.104     -0.266      0.792     -20.967      16.120
GSR             0.0307      0.318      0.097      0.924      -0.617       0.679
Bonus           1.0169      0.382      2.664      0.012       0.239       1.794
==============================================================================
Omnibus:                        0.083   Durbin-Watson:                   2.424
Prob(Omnibus):                  0.959   Jarque-Bera (JB):                0.241
Skew:                           0.094   Prob(JB):                        0.886
Kurtosis:                       2.652   Cond. No.                         918.
==============================================================================
```

Figure 10: First Model Using Multivariate Regression

The second model, which output can be seen entirely in Figure 10.1 below, was similar in the result but was an entirely different model. TotalPay was still the primary variable we were trying to predict, but this time using Win Percentage, Capacity, and Graduation Rate. The Adjusted R Squared for the model was 93%, with the testing data accounting for 48% of the variance. Unfortunately, there was an error in the additional output that the model had high multicollinearity, which means that at least two of the variables are far too much alike numerically via correlation to both be in the analysis. Some technical errors can cause this, but it is likely because of variable correlation.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               TotalPay   R-squared:                       0.930
Model:                            OLS   Adj. R-squared:                  0.924
Method:                 Least Squares   F-statistic:                     147.1
Date:                Sun, 29 Jan 2023   Prob (F-statistic):           3.61e-19
Time:                        19:06:30   Log-Likelihood:                -117.24
No. Observations:                  37   AIC:                             242.5
Df Residuals:                      33   BIC:                             248.9
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -45.6157      8.404     -5.428      0.000     -62.714     -28.518
Win_Percent     0.1455      0.088      1.651      0.108      -0.034       0.325
capacity        0.0008   5.72e-05     14.448      0.000       0.001       0.001
GSR             0.3161      0.101      3.131      0.004       0.111       0.521
==============================================================================
Omnibus:                        5.642   Durbin-Watson:                   2.427
Prob(Omnibus):                  0.060   Jarque-Bera (JB):                4.561
Skew:                          -0.846   Prob(JB):                        0.102
Kurtosis:                       3.305   Cond. No.                     4.74e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.74e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Figure 10.1: Second Model Using Multivariate Regression

A third model was created; the full results can be seen in Figure 10.2 below. This model tried to predict TotalPay for a third time, with Win Percentage, capacity, Graduation Rate, and built. The results from this analysis can be seen below in Figure 10.2. The Adjusted R Squared Value was on par with the other two analyses, at 92.2%, and the amount of variance accounted for in the testing set was 47.2%. Unfortunately, there was an error in the additional output that the model had high multicollinearity, which means that at least two of the variables are far too much alike numerically via correlation to both be in the analysis. Some technical errors can cause this, but it is likely because of variable correlation.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               TotalPay   R-squared:                       0.931
Model:                            OLS   Adj. R-squared:                  0.922
Method:                 Least Squares   F-statistic:                     107.3
Date:                Sun, 29 Jan 2023   Prob (F-statistic):           4.58e-18
Time:                        19:06:30   Log-Likelihood:                -117.19
No. Observations:                  37   AIC:                             244.4
Df Residuals:                      32   BIC:                             252.4
Df Model:                           4
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept     -21.4378     79.045     -0.271      0.788    -182.447     139.571
Win_Percent     0.1461      0.089      1.635      0.112      -0.036       0.328
capacity        0.0008   6.19e-05     13.255      0.000       0.001       0.001
GSR             0.3117      0.103      3.016      0.005       0.101       0.522
built          -0.0120      0.039     -0.308      0.760      -0.092       0.068
==============================================================================
Omnibus:                        5.713   Durbin-Watson:                   2.407
Prob(Omnibus):                  0.057   Jarque-Bera (JB):                4.642
Skew:                          -0.854   Prob(JB):                       0.0982
Kurtosis:                       3.301   Cond. No.                     4.40e+06
==============================================================================
```

Figure 10.2: Third Model Using Multivariate Regression

Conclusion and Final Thoughts

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               TotalPay   R-squared:                       0.933
Model:                            OLS   Adj. R-squared:                  0.799
Method:                 Least Squares   F-statistic:                     6.974
Date:                Sun, 29 Jan 2023   Prob (F-statistic):              0.129
Time:                        19:06:30   Log-Likelihood:                -19.153
No. Observations:                   7   AIC:                             48.31
Df Residuals:                       2   BIC:                             48.04
Df Model:                           4
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept    -137.3928     93.786     -1.465      0.281    -540.923     266.138
Win_Percent     1.5103      0.893      1.691      0.233      -2.333       5.354
SCL_PRIVATE    -2.6053     15.858     -0.164      0.885     -70.835      65.625
GSR             0.4737      0.650      0.729      0.542      -2.323       3.271
Bonus           3.1343      3.388      0.925      0.453     -11.444      17.712
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   2.114
Prob(Omnibus):                    nan   Jarque-Bera (JB):                0.148
Skew:                           0.298   Prob(JB):                        0.929
Kurtosis:                       2.610   Cond. No.                     3.70e+03
==============================================================================
```
Figure 11: ACC Multiple Regression Prediction

Based on the ACC data only. The recommended Coaches salary for DINO BABERS at Syracuse, based on his Win Percent of 43.5%, the school is Private, the school Graduation Rate of 85%, and a Bonus of $1587639.0, their salary will be $1804855 +/- 1558906.

```
                              OLS Regression Results
==============================================================================
Dep. Variable:               TotalPay   R-squared:                       0.851
Model:                            OLS   Adj. R-squared:                  0.834
Method:                 Least Squares   F-statistic:                     50.30
Date:                Tue, 31 Jan 2023   Prob (F-statistic):           4.26e-17
Time:                        18:16:03   Log-Likelihood:                -173.06
No. Observations:                  50   AIC:                             358.1
Df Residuals:                      44   BIC:                             369.6
Df Model:                           5
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept    -64.9362     81.287     -0.799      0.429    -228.760      98.887
Win_Percent    0.1949      0.095      2.061      0.045       0.004       0.386
capacity       0.0007   7.57e-05      9.356      0.000       0.001       0.001
GSR            0.1634      0.131      1.252      0.217      -0.100       0.427
built          0.0152      0.040      0.375      0.710      -0.066       0.097
Bonus          0.3440      0.203      1.697      0.097      -0.065       0.753
==============================================================================
Omnibus:                       29.659   Durbin-Watson:                   2.455
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               75.816
Skew:                          -1.600   Prob(JB):                     3.44e-17
Kurtosis:                       8.113   Cond. No.                     3.88e+06
==============================================================================
```

Figure 12: SEC Multiple Regression Prediction

If an opening in the SEC has significantly higher salaries, as we have seen throughout this analysis, his predicted salary with the same win percentage and parameters above his salary range would be from $3,233,348+/-$1,795,332.

This modeling and regression in a new programming language have been challenging, but something that was luckily just doable. Most of the time was spent on data preparation and scrubbing, getting the database into a form from the original four separate files down to a single file that could answer the established questions. This took many different steps, outlined in this report and the Jupyter Notebook containing the code. One should find notations about decisions and problems at the coding time. Many different limitations should be noted. First, the final database was less extensive and representative than possible for a multivariate analysis like this and any testing and training scenario. Many more schools might tell a different story than the one we found in the data. Aside from the sample size, there were problems with multicollinearity in our database when the regression models were run. This means that one should go back and analyze the variables until it is figured out which two or more are correlated enough to cause the problem. The variables could be eliminated down to one or combined in some way where both are still represented in the data set. Lastly, upon pondering the variables in the analysis, it was realized that all of the numeric and dollar-based variables come from one place: the school's overall budget for the football program each year. That may be an exorbitant amount of money in many situations, but overall, it limits our variables and how much they can change. The ability to change represents the ability to have relationships and is the basis of the statistics we use to examine the data.