Process Book

Basic Info:

Names: Tj VonWald-Villiard -- u1120747 -- u1120747@utah.edu
James Gibb -- u0915889 – u0915889@utah.edu

Project Title: Exploration of Offensive and Defensive Rankings in Collegiate Football

Github: https://github.com/jamesrgibb/DataVisProject-.git

Overview

Background and Motivation

We both are fans of the sport of football. James has been a long time casual fan of the game, and Tj within the last few years has become increasingly more interested in football as well. We both have done previous statistics projects on sport related topics and found that the data correlations in the sports are fascinating to learn about and explore. We both wanted to learn more about how the statistics in football correlate to wins and ultimately to championships. There is an old adage in the sports, especially in football that says: “Defense wins championships”. At times preconceived notions or old adages are deceptively wrong. Our project looks to visualize and explore the veracity of this statement. Through the visualizations we provide we hope to help the user understand the importance of defense or offense on a team's success in a game and throughout a season. The visual will provide the user the ability to explore correlations between individual statistics, as well as how those statistics change over time.

Project Objectives

The objective of our project is to make a highly interactive tool for exploration of team statistics over the course of 13 seasons. Interactivity is our main goal. We plan to give the user 3 interconnected tools to explore the dataset. The first is a sortable table that shows all of the teams in the dataset, as well as the team’s relevant statistics. The user will be able to switch between viewing offensive and defensive statistics and the correlating rankings for each of the teams. The user will also be able to group these teams into their respective conferences with the correlated offensive and defensive rankings. The second window will allow the user to view the compare correlation between teams, such as scatter plot to show the correlation of different statistics for a given team. The third window will show a histogram of the distribution of the selected

statistic. This will give perspective to the user as to how the team performs, as well as how important a particular statistic may be.

Data

Our dataset was found in the link below:
https://www.kaggle.com/datasets/jeffgallini/college-football-team-stats-2019?select=cfb14.csv
The dataset we are using is from kaggle.com. It contains 146 unique attributes for all 113 NCAA Division I college football teams. The data spans over the course of 13 seasons. We do not plan on using all statistics provided, we plan to identify during our as we process the data which statistics have greater significance. Our main goal is to create a tool that allows the user to explore the importance of defense versus offense through the dataset in an organized fashion.

Data Processing

The dataset in its current state on kaggle is very clean. The dataset provides 146 attributes we will need to determine which of those attributes are the most important to visualize in our project. Our exploration is primarily aimed at team statistics, the dataset is already organized accordingly. We plan on grouping the data further by the existing athletic conferences. An obstacle that we will need to work around is in recent years there have been realignments within conferences. While processing the data we will be vigilant of the realignment and make sure the teams are sorted into the appropriate conference each year. Additionally, we are in the process of looking for additional datasets that break down team statistics by game in at least one season. As of yet, it is not clear how difficult that may be to find and process each game for each team. Fortunately, our current dataset includes per-game statistics for a majority of the statistics, as well as an aggregate total for the year. This will supplement if we are unable to find an adequate dataset for per-game statistics.

Visualization Design

The visualizations will include a table of the year end rankings for every team in the dataset with both offensive and defensive statistics. The table will either display the offensive or defensive statistics at a given time. The user will be able to toggle between the offense and defensive ranking visuals. In addition, the user will be able to group the teams by athletic conference and see the corresponding offensive and defensive statistics and the correlating rankings for each conference. The table will be color coded with a diverging color scale, signaling higher or lower rankings in each individual

statistic. The overall ranking for both defense and offense will be color coded accordingly as well. The visual will also include an interactive scatter-plot to demonstrate how the correlations of individual statistics change over each season. In the scatter plot, the user will be able to explore which variables have the higher correlations for a specific team and conference. This is accomplished by first selecting the team in the table, and subsequently picking the statistics from a drop down menu above the scatter plot. The user's selection will represent the x and y axis in the scatter plot. For additional exploration, the user will be able to select a team and a specific single statistic that will then yield a histogram demonstrating the selected teams placement in the distribution relative to the other teams in college football according to the selected statistic.

Must-Have Features

- Interactive, sortable table that shows all team statistics, categorized by offense or defense depending on the user's selection.
    - Users will be able to toggle between either a defensive or offensive statistics view.
    - Users will be able to further group teams into the athletic conferences to view.
- Additional windows that allow users to explore selected variables.
    - Histogram that shows teams placement in the distribution of a single stat.
    - Scatter plot that allows users to pick teams and chart their statistics.

Optional Features

- Color code individual table cells based on how they compare to the mean for all teams.
- Subgroup table comparisons between conferences or higher-ranking teams.
- Select a single team and show a new table of all team statistics for all 9 seasons. Once a team has been selected the selection will highlight and show a window of team statistics

Project Schedule

We plan to work on the project for at least an hour each day either as a team or as individuals until the project is complete. After each week, we will evaluate our progress. If we haven't made as much progress as we previously planned, then we will increase the time that we work on it. Additionally, if we run into problems that inhibit our progress, we plan to attend office hours with the TA’s or the Professor to ensure we stay on

schedule and successfully complete our project. We both do a good job staying up to date with the homeworks, finishing them in a reasonable time, and we are confident that we will be able to allocate our time wisely.

Sketches

Table Visualization For Individual Teams (Offense and Defense)

Offense												
Teams	Ranks	Wins	Losses	Touchdowns	Total Yards Per Game	Passing Yards	Rushing Yards Per Game	Passing Offensive Rank	Passing Yards Per Game	Rushing Offensive Rank	Passing Touchdowns	Offensive Turnovers
Akron (MAC)	5	5	7	33	373.3	2995	123.7	49	249.6	106	32	26
Alabama (SEC)	12	12	2	68	484.5	3890	206.6	28	277.9	35	67	22
Arizona (Pac-12)	10	10	4	61	463.6	3945	181.9	21	281.8	45	55	18
Arizona St. (Pac-12)	10	10	3	59	442.3	3556	168.8	32	273.5	56	54	13
Arkansas (SEC)	7	7	6	56	406	2444	218	100	188	24	52	17
Arkansas St. (Sun Belt)	7	7	6	65	476.5	3381	216.4	39	260.1	25	60	22
Army West Point (FBS Independent)	4	4	8	40	358.8	747	296.5	125	62.3	5	36	13
Auburn (SEC)	8	8	5	58	485	2984	255.5	66	229.5	13	55	20
Ball St. (MAC)	5	5	7	37	369.3	2590	153.4	78	215.8	76	35	15
Baylor (Big 12)	11	11	2	82	581.5	4757	215.5	4	365.9	27	81	13
Boston College (ACC)	7	7	6	45	384	1681	254.7	122	129.3	14	44	13
Bowling Green (MAC)	8	8	6	51	432.9	3639	173	40	259.9	54	47	23
Buffalo (MAC)	5	5	6	49	438.2	2855	178.6	41	259.5	52	48	18
BYU (FBS Independent)	8	8	5	63	460.5	3623	181.8	26	278.7	47	60	27
California (Pac-12)	5	5	7	61	495.2	4152	149.2	6	346	84	59	20
Central Mich. (MAC)	7	7	6	47	398.1	3157	155.2	52	242.8	74	46	27
Cincinnati (AAC)	9	9	4	57	460.2	3947	156.5	13	303.6	72	53	22
Clemson (ACC)	10	10	3	48	408.3	3404	146.5	38	261.8	88	43	18
Colorado (Pac-12)	2	2	10	42	439.2	3415	154.6	19	284.6	75	42	21
Duke (ACC)	9	9	4	52	398.3	2814	181.8	77	216.5	46	47	15
East Carolina (AAC)	8	8	5	61	533	4835	161.1	3	371.9	64	59	23
Eastern Mich. (MAC)	2	2	10	23	289.6	1821	137.8	115	151.8	96	22	30
FIU (C-USA)	4	4	8	33	293.2	2016	125.2	113	168	104	26	22

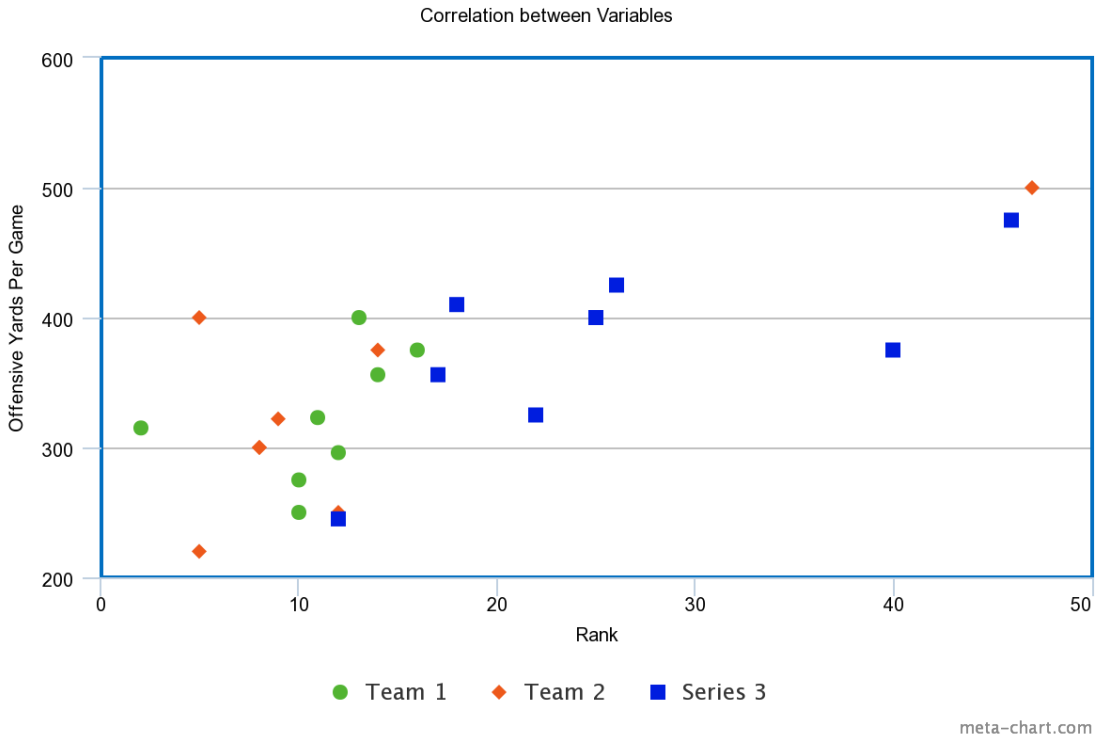
Defense												
Teams	Rank	Wins	Losses	Points Allowed Per Game	Total Yards Allowed Per Game	Toucdowns Allowed	Yards Allowed Per Play	Yards Allowed Per Game	Passing Touchdowns Allowed	Passing Yards Allowed Per Game	Defensive Scoring Rank	Turnovers Forced
Akron (MAC)	44	5	7	34	373.3	34	5.18	371.1	20	222.5	28	24
Alabama (SEC)	12	12	2	24	484.5	24	4.87	328.4	19	226	6	20
Arizona (Pac-12)	103	10	4	47	463.6	47	5.66	451	28	281.2	78	26
Arizona St. (Pac-12)	81	10	3	39	442.3	39	5.62	417.1	22	259.7	74	27
Arkansas (SEC)	10	7	6	30	406	30	5.12	323.4	19	208.8	10	24
Arkansas St. (Sun Belt)	83	7	6	52	476.5	52	5.62	421.2	18	216	90	28
Army West Point (FBS Independent)	90	4	8	50	358.8	50	6.48	430.8	31	237.8	102	15
Auburn (SEC)	64	8	5	44	485	44	5.67	398.8	22	230.1	62	27
Ball St. (MAC)	86	5	7	40	369.3	40	5.87	428.8	16	237.8	67	24
Baylor (Big 12)	50	11	2	41	581.5	41	5.28	381.8	24	264.2	48	26
Boston College (ACC)	11	7	6	34	384	34	5.13	324.2	20	229.8	21	14
Bowling Green (MAC)	115	8	6	62	432.9	62	5.98	493.6	25	291.4	106	33
Buffalo (MAC)	62	5	6	46	438.2	46	5.86	397.7	17	222.5	95	15
BYU (FBS Independent)	56	8	5	44	460.5	44	4.93	391.5	25	269.7	71	26
California (Pac-12)	121	5	7	61	495.2	61	6.27	511.8	42	367.2	120	17
Central Mich. (MAC)	29	7	6	39	398.1	39	5.44	355.5	24	232.5	45	22
Cincinnati (AAC)	96	9	4	40	460.2	40	5.98	439.1	22	255.4	68	24
Clemson (ACC)	1	10	3	23	408.3	23	4.03	260.8	12	157.4	3	24
Colorado (Pac-12)	111	2	10	56	439.2	56	6.55	461	35	256.2	116	11
Duke (ACC)	67	9	4	33	398.3	33	5.28	399.7	14	206.8	24	21
East Carolina (AAC)	37	8	5	36	533	36	5.1	367.3	20	255.5	53	19
Eastern Mich. (MAC)	118	2	10	61	289.6	61	6.75	498.8	27	274.8	122	12
FIU (C-USA)	35	4	8	35	293.2	35	5.38	363.8	17	203.3	40	33

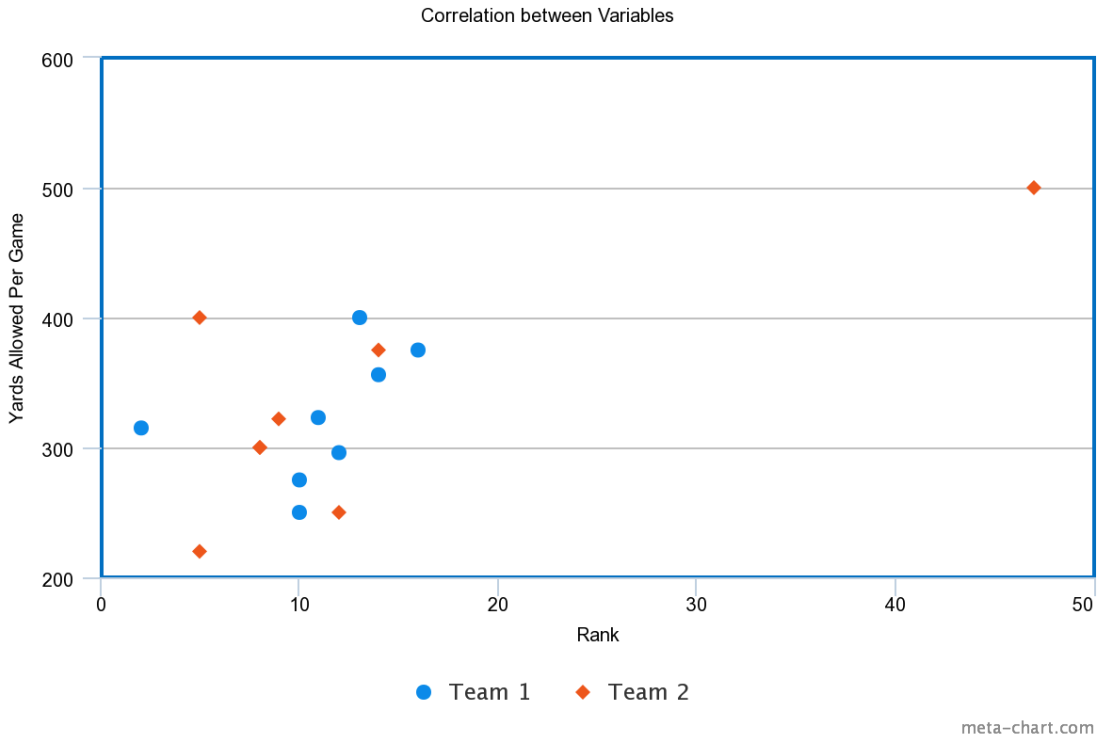
Table Visualization When Grouped By Conference (Offense and Defense)

Offense											
Teams	Ranks	Wins	Losses	Touchdowns	Total Yards Per Game	Passing Yards	Rushing Yards Per Game	Passing Offensive Rank	Passing Yards Per Game	Rushing Offensive Rank	Passing Touchdowns
ACC	5	5	7	33	373.3	2995	123.7	49	249.6	106	32
AAC	12	12	2	68	484.5	3890	206.6	28	277.9	35	67
C-USA	10	10	4	61	463.6	3945	181.9	21	281.8	45	55
Big Twelve	10	10	3	59	442.3	3556	168.8	32	273.5	56	54
BIG Ten	7	7	6	56	406	2444	218	100	188	24	52
FBS Independents	7	7	6	65	476.5	3381	216.4	39	260.1	25	60
Pac Twelve	4	4	8	40	358.8	747	296.5	125	62.3	5	36
MAC	8	8	5	58	485	2984	255.5	66	229.5	13	55
MWC	5	5	7	37	369.3	2590	153.4	78	215.8	76	35
WAC	11	11	2	82	581.5	4757	215.5	4	365.9	27	81
SEC	7	7	6	45	384	1681	254.7	122	129.3	14	44

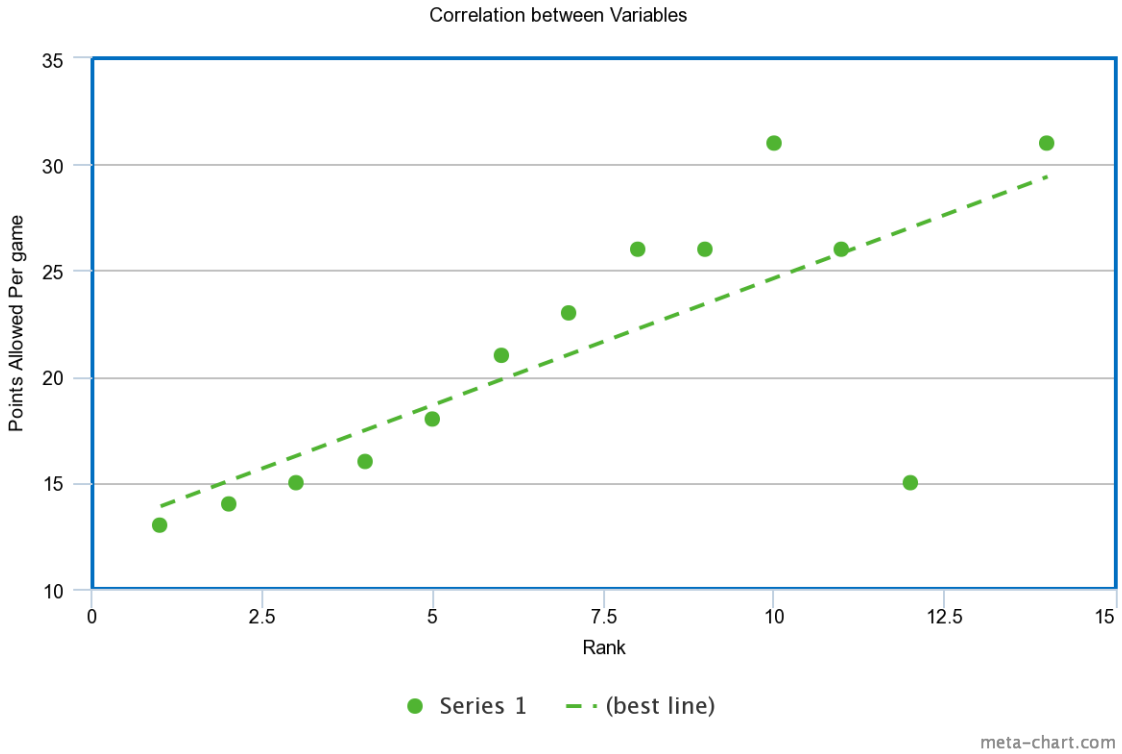
Defense											
Teams	Rank	Wins	Losses	Points Allowed Per Game	Total Yards Allowed Per Game	Toucdowns Allowed	Yards Allowed Per Play	Yards Allowed Per Game	Passing Touchdowns Allowed	Passing Yards Allowed Per Game	Defensive Scoring Rank
ACC	44	5	7	34	373.3	34	5.18	371.1	20	222.5	28
AAC	12	12	2	24	484.5	24	4.87	328.4	19	226	6
C-USA	103	10	4	47	463.6	47	5.66	451	28	281.2	78
Big Twelve	81	10	3	39	442.3	39	5.62	417.1	22	259.7	74
BIG Ten	10	7	6	30	406	30	5.12	323.4	19	208.8	10
FBS Independents	83	7	6	52	476.5	52	5.62	421.2	18	216	90
Pac Twelve	90	4	8	50	358.8	50	6.48	430.8	31	237.8	102
MAC	64	8	5	44	485	44	5.67	398.8	22	230.1	62
MWC	86	5	7	40	369.3	40	5.87	428.8	16	237.8	67
WAC	50	11	2	41	581.5	41	5.28	381.8	24	264.2	48
SEC	11	7	6	34	384	34	5.13	324.2	20	229.8	21

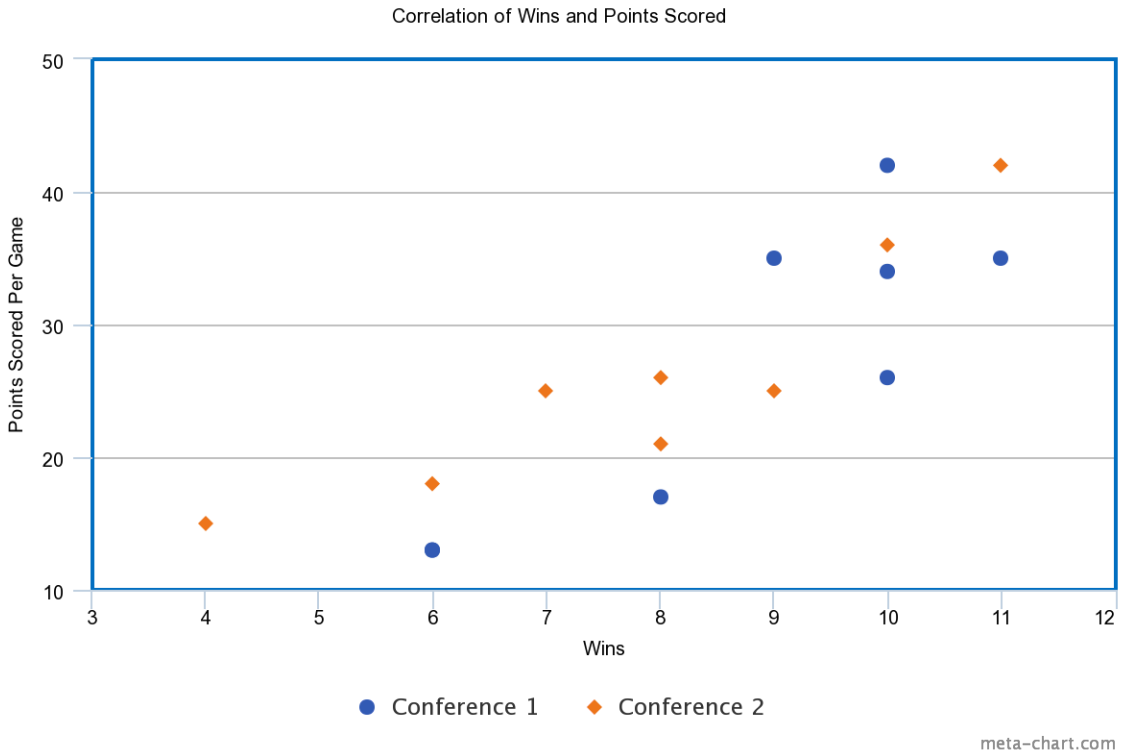
Scatter Plot with Teams



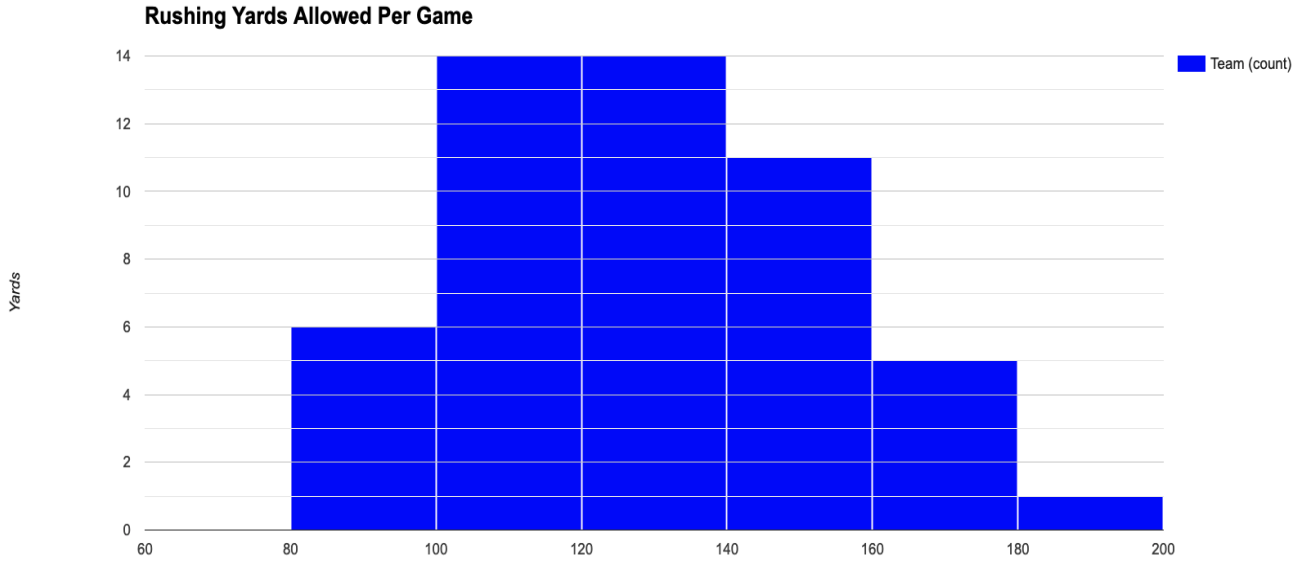


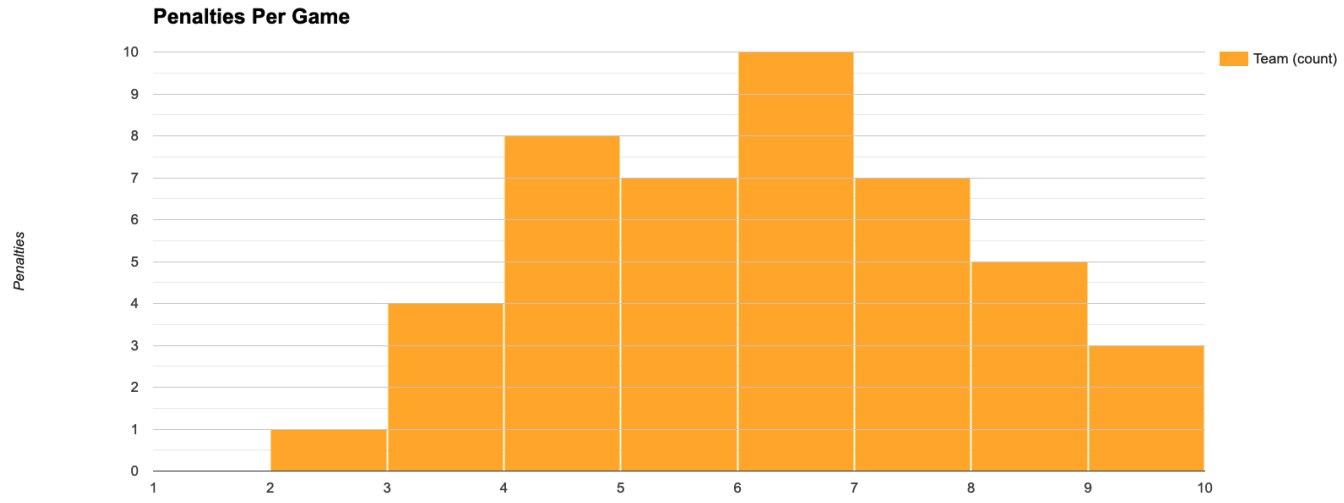
Scatter Plot using Conferences





Histograms Feature





Notes from the peer review:

Lane:

- Find a way to tell a story in the visualization.
- Focus on interactivity.
- Add Icons for each of the teams to create more intrigue with the visual.
- Find ways to correlate the different offensive and defensive statistics in the dataset.
- Focus on making the data and the visual not intimidating to those who may not be interested in sports.
- Find a ways to make the visual easy to approach and easy to explore.
- This can be done with a lot of variety in features and in the interactivity

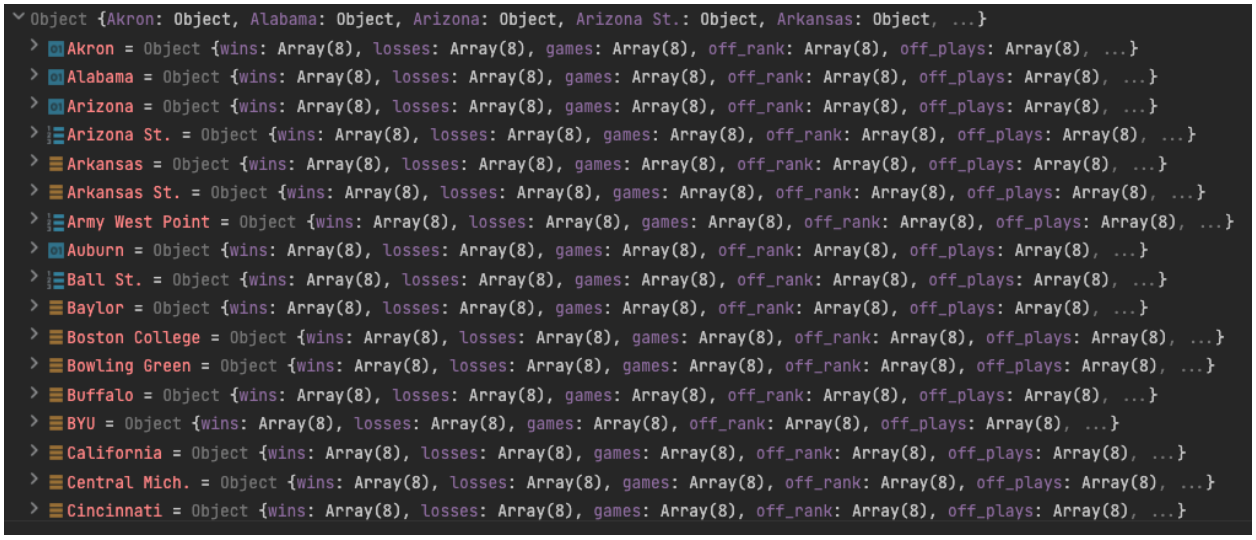
Adam:

- Asked about how we are going to wrangle our columns.
- What ways will be the best to store them? Find a way that makes them easy to access in the future.
- How are we presenting the columns to the user? We will show all of the columns in the dataset or choose the ones we find applicable?
- How will we will narrow down which columns will be applicable and important for the user?
- How are we showing correlation to the users? Not just regurgitating the information in the dataset but telling the story of the data with our visualization.
- What possible ways are there to shows the correlations between defensive and offensive statistics?
- Find a way to implement the school colors in the graphs in order to make the visual stick out more and viewers have an easier time with picking out which school correlates with which statistic.
- Plan to have checkmarks along the way that ensure you are implementing enough user interactivity for the project.

Process Book

Data Processing

The first step we took in putting together this project was organizing the data so that we could work with it. This has been a challenge to do in JavaScript, especially when we have strong backgrounds in Python. We are working with 9 different csv files that each cover a year of college football statistics. Before beginning work on our visualizations, we organized the data first by season and then by team across all 9 years. This allowed us to do an analysis for each individual year or for a single team over time.
The CSV file proved to be challenging to use in the code. It required several different refactorings to be able to manipulate the data across both the base visualizations. The table visual was easier to implement from the csv because of the similarity in layout. However, with the line-graph it proved to be difficult to display the correct data for each across the time period without having keys set up in the data structure. This caused us to refactor the existing data to be organized first with a key for each of the teams in the table. Then the corresponding statistics each have their own key and store a list of the values over the nine seasons of the data we gathered. This made retrieving the data for the line-graph much simpler and easier to keep track of. We believe that going forward the keys will help us access the data much quicker than how our data was originally laid out. Below is an example of how the data is laid out with the corresponding keys. Also displayed is how the keys were assigned to the various values in the script.js file.



```
        sack_total.push(parseInt(year[0]['Sacks']))
        sack_rank.push(parseInt(year[0]['Sack.Rank']))
        opp_pass_tds_allowed.push(parseInt(year[0]['Opp.Pass.TDs.Allowed']))

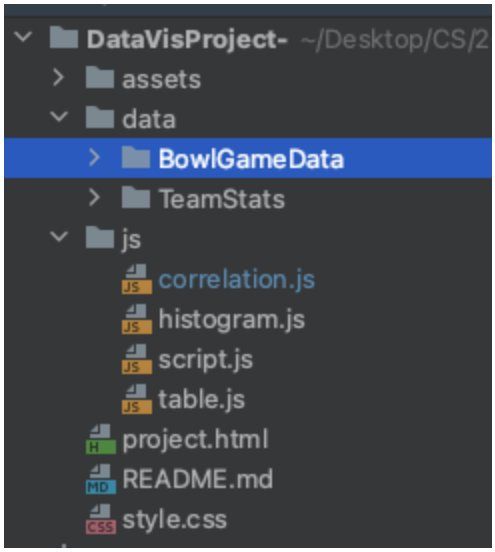
    })
    teamStats[team[0][0].Team].wins = wins
    teamStats[team[0][0].Team].losses = losses
    teamStats[team[0][0].Team].games = games
    teamStats[team[0][0].Team].off_rank = off_rank
```

```
Akron = Object {wins: Array(8), losses: Array(8), games: Array(8), off_rank: Array(8), off_plays: Array(8), ...}
  wins = Array(8) [5, 5, 8, 5, 7, 4, 0, 1]
  losses = Array(8) [7, 7, 5, 7, 7, 8, 12, 5]
  games = Array(8) [12, 12, 13, 12, 14, 12, 12, 6]
  off_rank = Array(8) [106, 88, 104, 84, 121, 126, 130, 121]
  off_plays = Array(8) [837, 891, 905, 776, 906, 764, 725, 360]
  off_yards = Array(8) [4104, 4479, 4642, 4649, 4424, 3533, 2918, 1687]
  off_yards_play = Array(8) [4, 5, 5, 5, 4, 4, 4, 4]
  off_yards_game = Array(8) [342, 373, 357, 387, 316, 294, 243, 281]
  off_TD = Array(8) [30, 32, 38, 38, 38, 21, 14, 12]
  def_rank = Array(8) [59, 44, 15, 114, 103, 61, 83, 110]
  def_yards_allowed = Array(8) [4764, 4453, 4310, 5592, 6200, 4657, 4967, 2759]
  def_yards_allowed_play = Array(8) [5, 5, 4, 6, 6, 5, 5, 7]
  def_yards_allowed_game = Array(8) [397, 371, 331, 466, 442, 388, 413, 459]
  def_TD_allowed_Total = Array(8) [44, 35, 37, 50, 49, 41, 56, 34]
  first_dwn_rank = Array(8) [54, 51, 45, 109, 129, 60, 58, 24]
  pass_off_rank = Array(8) [75, 49, 97, 45, 81, 95, 100, 119]
  pass_def_rank = Array(8) [67, 53, 83, 79, 84, 27, 29, 41]
```

The data inside each of the categories is laid out sequentially, keeping it organized and easily accessed. We believe that in the future we will make further adjustments to the overall structure of our data as problems present themselves. For now, the current structure is working sufficiently.

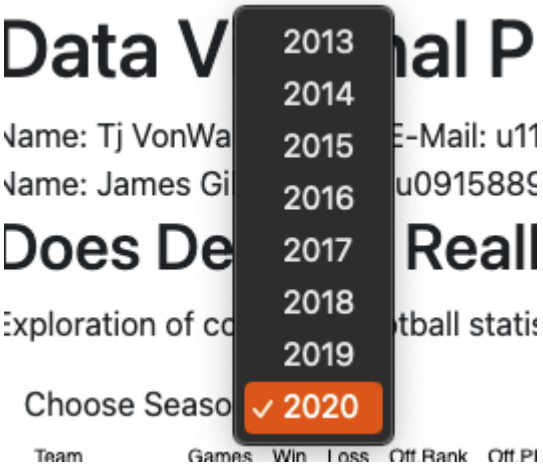
File Organization

After gathering and processing our data. We decided to separate the work on our combined visualization, with one of us constructing the table and the other working on a correlation. The histogram will be implemented as we continue to progress with the project, it requires different interaction from the table that will be implemented in the near future. Our project organization has each visualization organized in its own js file. Then, in the script.js file we have this file organized in an object-oriented way that calls the various different visuals in a manner as we have seen in class. Below is a layout of the file system described above.



Visualizations and Features:

The table was constructed from a basic html skeleton within its own div, we placed a table element with a head and body. The table is then dynamically populated with what yearly dataset the user has selected. This is selected by a dropdown down menu at the top of the page as shown below.

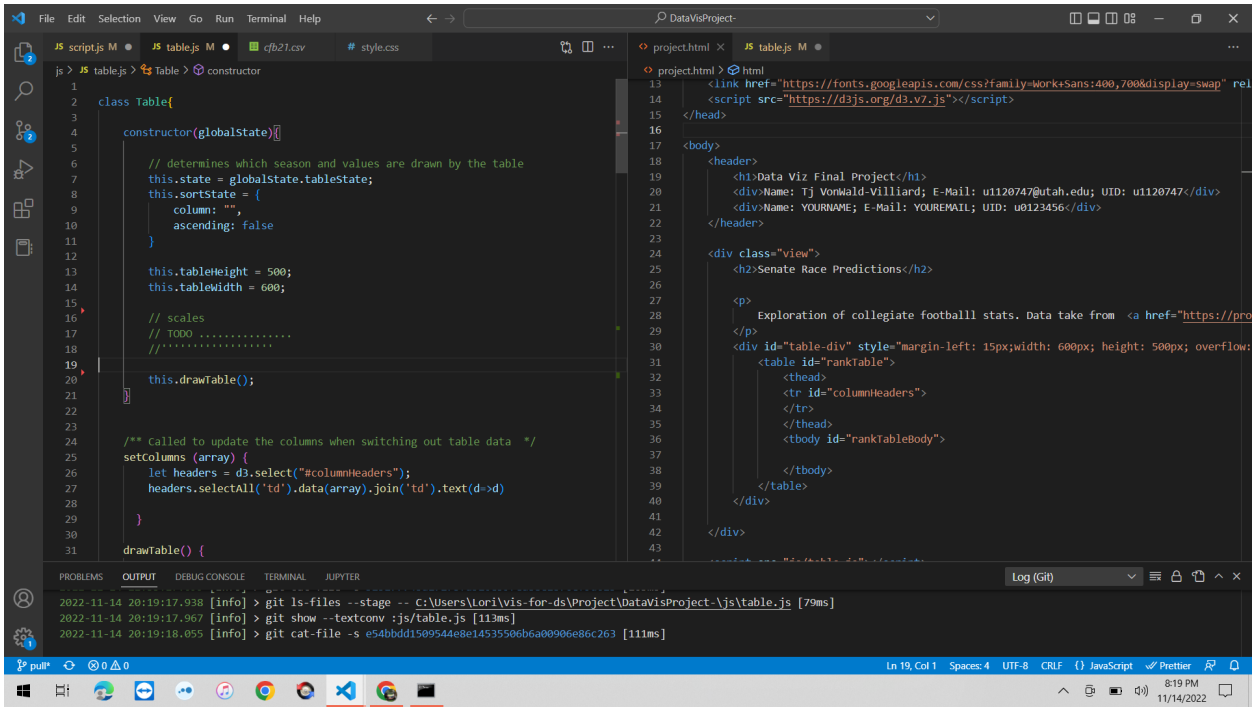


Within each column of the table, the data is color coded according to how it compares to the column mean. The greater values are hued green while the lower values are hued red. We have also implemented basic sorting of the columns when clicking on table headers. When a repeated click occurs, the column alternates between a descending and ascending order. Below shows the difference in the ascending and descending order of wins for a given year.

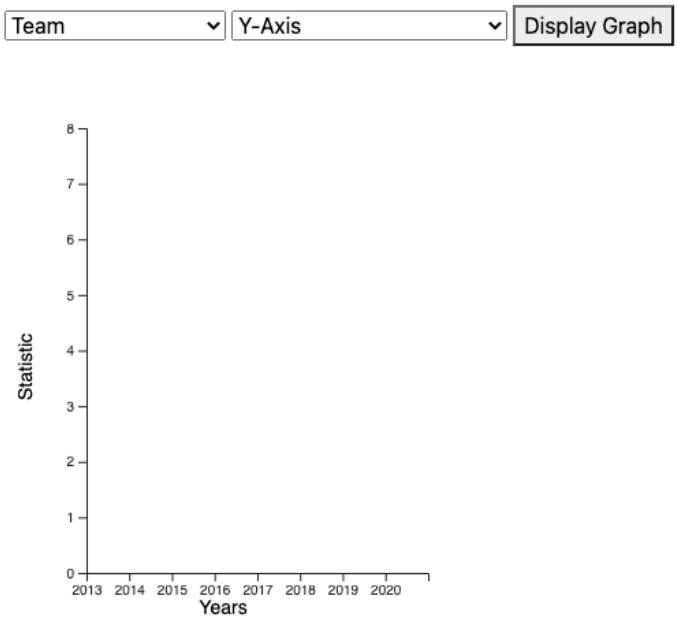
Team	Games	Win	Loss	Off.Rank	Off.Plays	Off.Yards	Off.Yards.Play	Off.TDs	Off.Yards.per.Game	Def.Rank
LSU	15	15	0	1	1080	8526	7.89	93	568.4	31
Clemson	15	14	1	5	1075	7931	7.38	85	528.7	6
Appalachian St.	14	13	1	39	969	6064	6.26	65	433.1	26
Ohio St.	14	13	1	4	1069	7419	6.94	87	529.9	1
Boise St.	14	12	2	48	993	6006	6.05	58	429	33
Georgia	14	12	2	61	940	5713	6.08	47	408.1	3
Memphis	14	12	2	10	992	6791	6.85	65	485.1	61
Oklahoma	14	12	2	3	946	7527	7.96	74	537.6	38
Oregon	14	12	2	40	956	6063	6.34	61	433.1	22
Air Force	13	11	2	51	881	5483	6.22	55	421.8	17
Alabama	13	11	2	6	842	6640	7.89	76	510.8	20
Baylor	14	11	3	52	948	5901	6.22	59	421.5	39
Cincinnati	14	11	3	80	993	5464	5.5	49	390.3	40
Fla. Atlantic	14	11	3	23	1053	6280	5.96	62	448.6	50
Florida	13	11	2	45	865	5597	6.47	54	430.5	9
Louisiana	14	11	3	8	987	6918	7.01	69	494.1	47
Minnesota	13	11	2	42	883	5616	6.36	56	432	10
Navy	13	11	2	19	877	5925	6.76	62	455.8	16
Notre Dame	13	11	2	43	886	5608	6.33	57	431.4	18
Penn St.	13	11	2	57	885	5355	6.05	59	411.9	34
Utah	14	11	3	48	912	6006	6.59	54	429	2
Hawaii	15	10	5	13	1056	7065	6.69	63	471	96
Iowa	13	10	3	99	864	4765	5.52	33	366.5	12
Louisiana Tech	13	10	3	35	933	5679	6.09	48	436.8	53
San Diego St.	13	10	3	110	933	4461	4.78	27	343.2	5
SMU	13	10	3	9	1037	6368	6.14	70	489.8	107
UCF	13	10	3	2	1016	7026	6.92	70	540.5	32
Wisconsin	14	10	4	38	963	6065	6.3	56	433.2	4
Auburn	13	9	4	64	947	5285	5.58	51	406.5	28
Michigan	13	9	4	68	905	5220	5.77	51	401.5	11
UAB	14	9	5	105	896	4926	5.5	36	351.9	8
Virginia	14	9	5	81	962	5445	5.66	51	388.9	48
Western Ky.	13	9	4	84	900	5041	5.6	38	387.8	24
Arizona St.	13	8	5	94	842	4837	5.74	34	372.1	67
Arkansas St.	13	8	5	33	925	5711	6.17	50	439.3	124
Buffalo	13	8	5	82	937	5051	5.39	50	388.5	7
California	13	8	5	117	841	4271	5.08	35	328.5	65

Team	Games	Win	Loss	Off.Rank	Off.Plays	Off.Yards	Off.Yards.Play	Off.TDs	Off.Yards.per.Game	Def.Rank
Akron	12	0	12	130	725	2918	4.02	14	243.2	83
Massachusetts	12	1	11	126	822	3517	4.28	29	293.1	130
Old Dominion	12	1	11	128	810	3285	4.06	21	273.8	46
UTEP	12	1	11	116	768	3950	5.14	27	329.2	95
Arkansas	12	2	10	111	783	4081	5.21	28	340.1	110
New Mexico	12	2	10	69	832	4803	5.77	33	400.3	129
New Mexico St.	12	2	10	102	832	4333	5.21	31	361.1	126
Rutgers	12	2	10	129	735	3277	4.46	18	273.1	98
South Alabama	12	2	10	113	791	3963	5.01	29	330.3	79
UConn	12	2	10	108	810	4136	5.11	27	344.7	116
Bowling Green	12	3	9	120	839	3829	4.56	24	319.1	117
Georgia Tech	12	3	9	127	717	3436	4.79	26	286.3	89
Kansas	12	3	9	90	773	4527	5.86	35	377.3	122
Maryland	12	3	9	109	767	4121	5.37	39	343.4	109
Northwestern	12	3	9	124	842	3565	4.23	23	297.1	25
Rice	12	3	9	125	753	3530	4.69	27	294.2	58
Texas St.	12	3	9	121	768	3814	4.97	25	317.8	86
Vanderbilt	12	3	9	123	744	3592	4.83	21	299.3	101
Arizona	12	4	8	30	873	5281	6.05	42	440.1	120
Colorado St.	12	4	8	34	841	5249	6.24	39	437.4	52
East Carolina	12	4	8	46	900	5162	5.74	34	430.2	119
Fresno St.	12	4	8	85	767	4632	6.04	44	386	88
Houston	12	4	8	78	780	4698	6.02	39	391.5	118
Middle Tenn.	12	4	8	59	771	4900	6.36	40	408.3	114
NC State	12	4	8	88	881	4563	5.18	28	380.3	72
North Texas	12	4	8	54	850	5023	5.91	44	418.6	81
Ole Miss	12	4	8	26	891	5343	6	40	445.3	85
Purdue	12	4	8	75	862	4718	5.47	38	393.2	100
South Carolina	12	4	8	96	862	4463	5.18	30	371.9	66
South Fla.	12	4	8	112	773	3970	5.14	33	330.8	71
Stanford	12	4	8	97	790	4415	5.59	26	367.9	93
Texas Tech	12	4	8	11	935	5691	6.09	44	474.3	127
Tulsa	12	4	8	53	947	5036	5.32	38	419.7	70
UCLA	12	4	8	65	916	4868	5.31	40	405.7	113
UNLV	12	4	8	92	831	4492	5.41	37	374.3	105
UTSA	12	4	8	107	819	4139	5.05	28	344.9	92

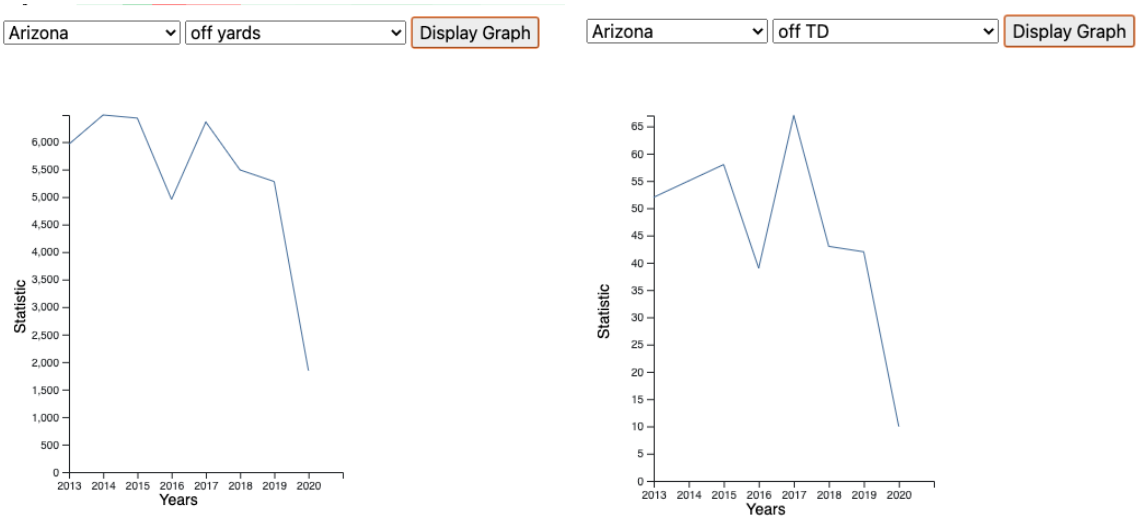
The next picture shows the skeleton code we started with for our html file, and what was referenced in the table.js file.

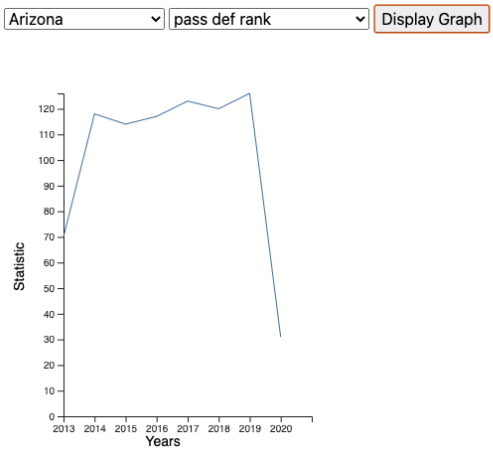


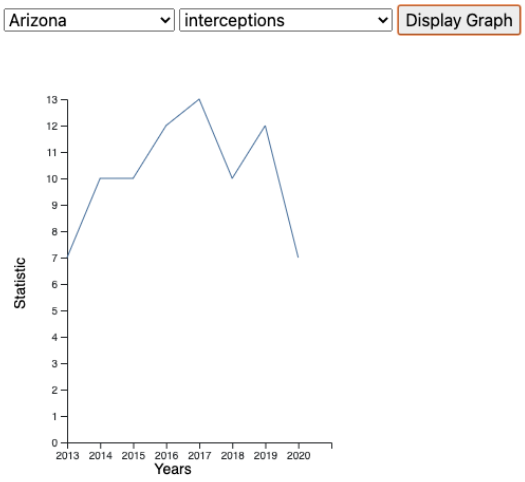
In the correlation.js file, we decided to first explore displaying how the statistical trends change over time and vary from team to team. The graph first shows a blank graph with the years displayed on the x-axis. The user has the option to choose what statistic is displayed on the y-axis and which team the user would like to see displayed. Once the user has made both of these decisions the user may click “Display Graph” and the graph will populate with the selected data. This process is shown below.



Once the user has clicked “Display Graph” the graph as mentioned before will populate with the selected data. The y-axis will also adjust accordingly to scale the data values appropriately, the ticks on the y-axis are also updated so the user knows what the line means comparatively. Below are a few different visuals of different statistics displayed on the graph.







Project Checkpoint Meeting

The Teachers Assistant we met with encouraged us to apply more interactivity in our visualization to create more interest with users. She told us that we had a good start on the development with the table and the interactivity of the line graph as well as other features could be improved.

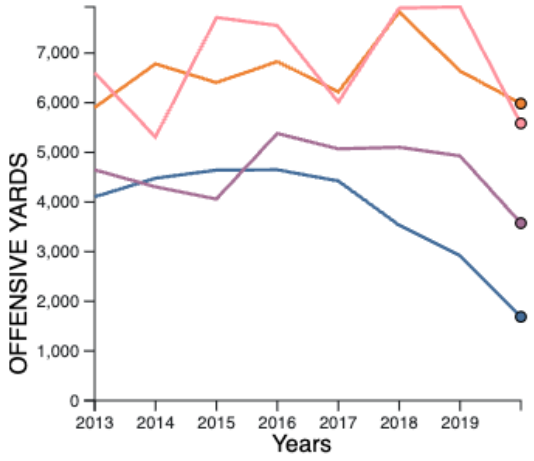
Table

The table is the centerpiece of our project. The table allows us to view all our data, filtering by offensive or defensive statistics and the year of play. We decided to encode the table data using color. The cells are on a diverging scale: a deeper color of red indicates that the selected statistic is farther below the mean of the columns, with green indicating the opposite. We implemented this by storing the entirety of the data in the Table object, then applying our selected filters, storing the result in a seperate object property that is bound to the table data. Furthermore, the table is sortable. The user can click on any column header, continued clicking will alternate between ascending and descending sorting.

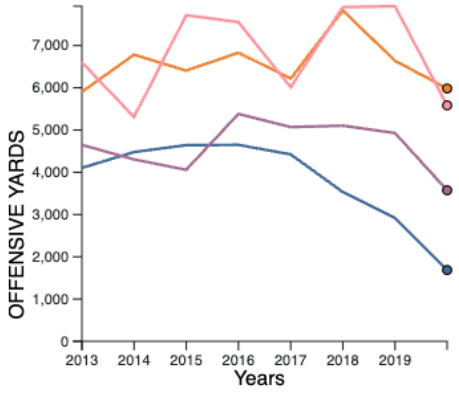
The table also serves as the point of interaction for our histogram and line chart. When a <td> is clicked, the histogram changes to measure the data column. The bin containing the selected data is then highlighted. The line chart is also updated, showing a graph of the selectec team's stat over all 8 years of data. If the user clicks on different teams in the same column, they too will be graphed.If the user selects any data that is not in the currently selectec column, the linechart is reset.

Line Graph

The idea of the line graph as mentioned before was to show how the statistics change over time for a certain team. This was a good starting point for the line graph but we decided to implement additional features to promote more interactivity as the TA suggested. We then decided to implement a feature that allows the user to click on a particular cell in the table and the data for that team will populate the line graph. This feature can be seen below. We also decided that just showing one team was not enough perspective for a user. We chose to have the line graph show multiple lines for each team that had been selected. The chart will show the teams that have been selected for a particular statistic. If a new statistic is selected the line chart will show the updated version of the line chart with just the newly selected team. The line chart also dynamically populates the y-axis according to what the user clicks. The range of values in the y-axis change as new values are taken in to always show all the data points in the graph. The graph also updates the y-axis label for the corresponding statistic the user has selected. The update for the line chart showing multiple lines can be seen in the screenshot below.



This feature gave the user greater perspective of how much individual team statistics can change year to year. This was a good step in the right direction for helping the user see the different trends over time. We also decided to add a legend with corresponding teams name and the average, we decided to color code the table legend to help the user discern which teams corresponded with which line. The full and final view of the line chart can be seen below.

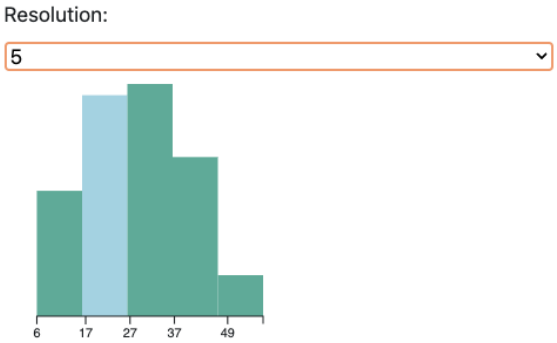


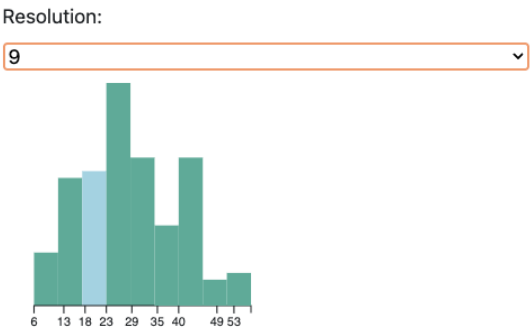
Clear Selection

Team	Average
Akron	3804.5
Alabama	6573.875
Army West Point	4633.75
Clemson	6827.25

Histogram

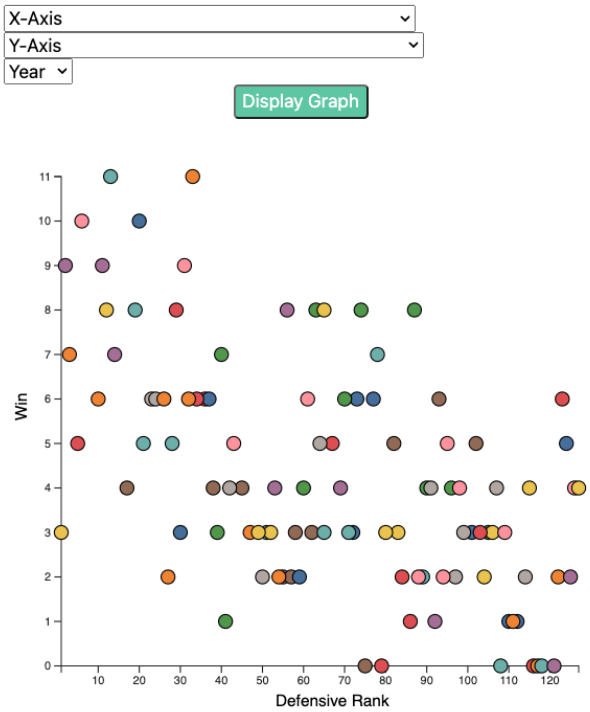
The plan for the histogram was to be able to select a cell and show the distribution of that variable while highlighting the bin that contains the team of the specified cell. Work began by using d3.histogram() to construct the visualization but this caused a number of problems. For reasons we could not decipher, the bins were being constructed and shown with varying widths and placed in nonsensical locations on the x-axis. This led to us scrapping the prebaked implementation and making the whole thing from scratch. This ended up working perfectly with the exception of a bug related to the y-axis. Though the scale was working perfectly, the ticks on the y-axis would just alternate between 0 and the number of bins. We decided to omit the y-axis in the end, as it wasnt ultimately necessary. The final addition to the histogram was the option to change the resolution (number of bins) which allows the user to get a better picture of the distribution should they desire. The screenshot below shows the histogram before the user selects a different number of bins than the default and after as well.



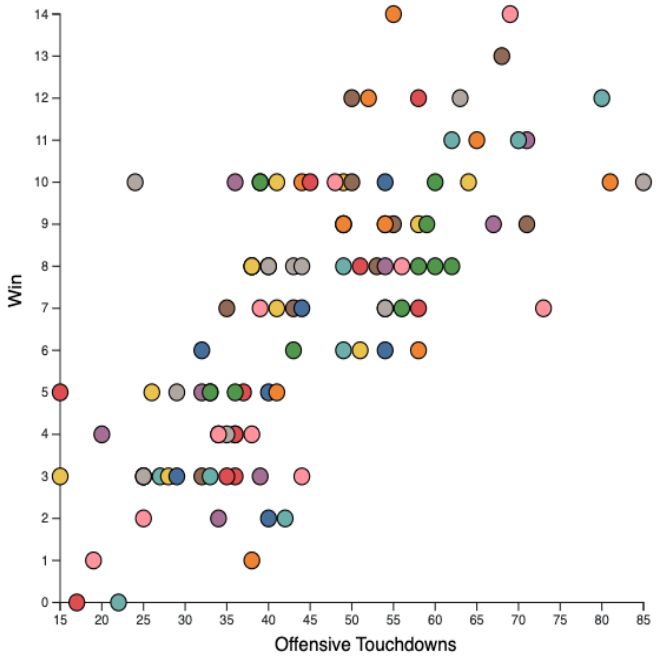


Scatter Plot

One of our main goals was to show the overall correlation between different statistics and what statistics really affect a teams overall performance. We thought tha
t the implementation of a scatter plot would provide the user with additional insight into what statistics correlated with which statistics. Initially we began we the traditional scatter plot that just showed the Defensive Ranking of the teams and how that correlated with their wins. This process proved to be more difficult than anticipated for accessing the appropriate values for the scatter plot. We decided to use the same type of data structure that we used for displaying the data. The d3.rollup function proved to be very useful in establishing the corresponding keys with the correct data values. Below is a layout of the scatter plot in its default state.



The Scatter Plot is configured with two drop down menus that allow the user to decide which data value to put on the x and y axis. Then the user can also choose which year to display. Once these selections have been made the user clicks the “Display Graph” button and the scatter plot will display a different set of data points.

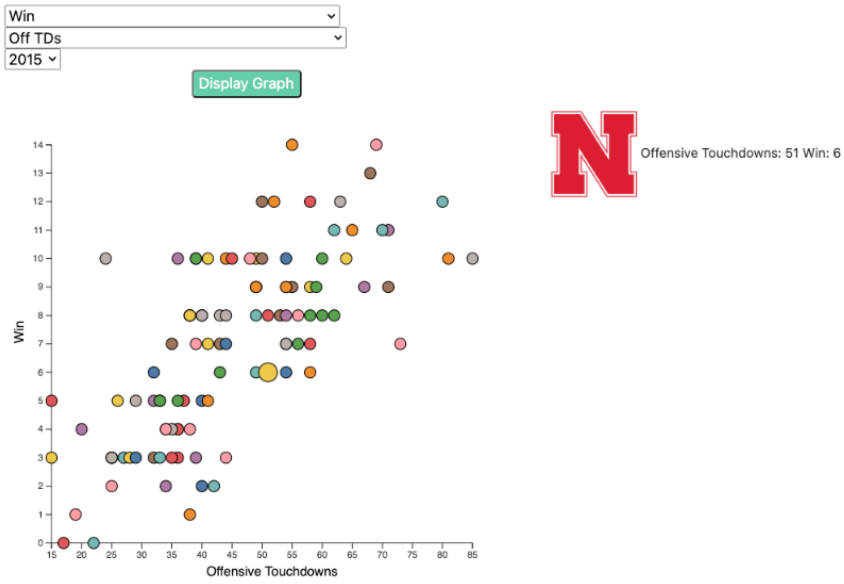


The following code then converted this to a json file.

```
```{r setup, include=FALSE}
install.packages("rjson")
jsonData<-toJSON(logo_ref)
write(jsonData,'cfb_logos')
```

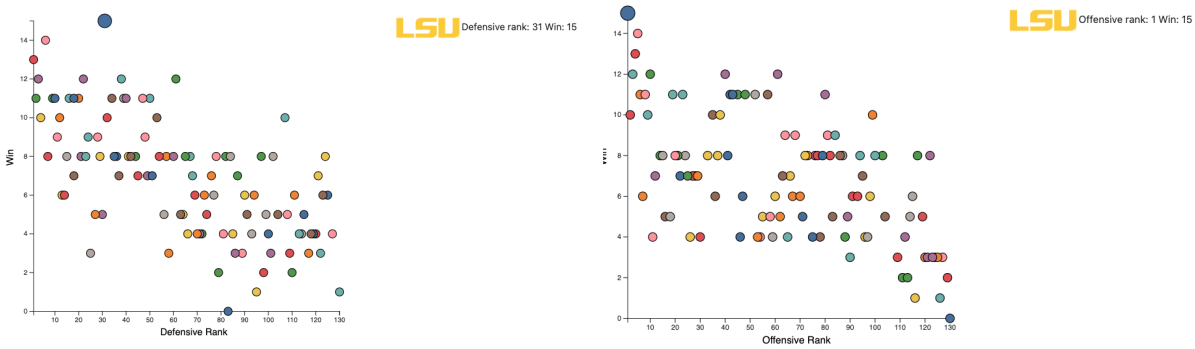
This file was then stored in the data folder of our project. The file was then imported to our code as a dictionary with the corresponding name of the school and the corresponding logo. We had to sift through some of the school names that were not formatted the same as our other dataset, which proved to be tedious. Once, this was loaded in we initially began by starting by creating a rect in the svg and displaying it on top of the graph like we have in previous assignments. This worked in displaying the box but passing the image to the box we encountered some errors created by cookies from accessing the photos from other external sites. We were able to work around this by creating a div that would be shown on the side since it can not be added to an svg element. We did try working with the .foreignObject call in d3. However, this returned the same error we encountered before. We ultimately opted to show the logo of the school that was being hovered over immediately to the right of the scatter plot. We added the corresponding statistics for the team as well so the user could quantify what
```

they were looking at. We also added an animation to dots to show where the user was hovering over. The display of the scatter plot can be shown below.



Evaluation

The visualization taught us a lot about football that were prior misconceptions. We went into the project thinking that the data was going to tell us that defense is utterly more important than offense or that offensive turnovers are a direct correlation with losses. All of these things are important but we noticed that the teams that were the most successful, in terms of winning, were the teams that were ranked on average the highest in both offensive and defensive rankings. For example, in 2019 the LSU Tigers had one of the best teams probably ever in college football, the team had the number one ranking offense in the country and the fifteenth ranked defense. They were not a lopsided team, this allowed them to win the National Championship that year. The graphic of their ranking is shown below.



The opposite is also true for teams that are very successful on offense or defense but not as much as on the other tend to still struggle to win games. The Texas Tech Red Raiders were an offensive powerhouse the same year ranking as the fourth best offensive team in the nation, however, they were ranked as 127th defensively that year. In 2019, Texas Tech won 4 games total. This shows that you need to be consistently ranked highly in both offense and defense to win a lot of games. It doesn't necessarily matter how great you are on one side of the ball, you need to be great on both sides to win a championship.