Scraping and Visualizing College Football Data

Ryan Morgan

April 25, 2017

Introduction

In the past, I've wanted to look at data and statistics for college football teams. Unfortunately, there isn't a downloadable data set online (and some websites, such as cfbstats think it's okay and to charge people $350 or more for their .csv files on college football data). In STAT 585, we learned about web scraping, which gives a way to compile data online into clean, manageable data sets. I used the methods explored in class to scrape college football data from the sports-reference website. I wrote several functions to scrape the various data sets found on the website, and then cleaned and compiled the data into manageable and formatted data sets. After scraping and cleaning the data, I wrote six Shiny apps that can be used to visualize the many aspects of the data. All the functions, data sets, and apps mentioned can be found on the github repository rjmorgan4/585-project. For the sake of space, I will not be including the code used to scrape the data and generate the displayed plots, but all of the code can be found on the github repository.

Data to Scrape

From this page on the sports-reference website, there is a list of all the college football teams that sports-reference has information on. There is a total of 296 schools listed, along with the years that each school's football team was active.

From that page, users can select what team they would like more data on. For example, users can navigate to a page that has information on the 117 seasons of Nebraska football. This page has the Year, Conference, Wins, Losses, Ties, Winning Percentage, SRS (Simple Rating System, which is a way of rating how good the team was that season), SOS (Strength of Schedule, a way of measuring how good a team's opponents were that season), the preseason Associated Press (AP) ranking, the highest AP ranking for that season, the postseason AP ranking, the head coach(es) from that season, the bowl game played in that season, and any notes on the that season.

From the team history page, a user can select a specific season that they would like more data on. For example, a user can navigate to a page on the 2016 Nebraska football season. This page has a table that lists game averages for the offense, game averages for the defense, and the difference between the offensive and defensive averages (differences found by taking offensive average minus defensive average). For example, in the 2016 Nebraska football team averaged 15.5 completions per game on offense, while the defense surrendered an average of 19.5 completions per game to opponents, for a difference of -4.0 completions per game. This table has game averages on 21 statisitcs, including Pass Completions, Rushing Attempts, and Total Turnovers.

A user can also view a page with game by game details for a specific team during a specific season. For example, a user could view the schedule of the 2016 Nebraska football team. This page has information on each game that a team played that season, including the date and location of the game, the opponent of the game, and the result (either Win, Loss, or Tie) of the game.

I will be scraping data from teams' history pages, teams' season specific pages, season average pages, and schedule pages.

"Helper" functions used to assist with Scraping

asNumber

Eventually I realized it would be useful to write a function that converts a text column into numbers. When I was scraping the datasets from the website, I struggled getting R to recognize numeric values as numbers. I eventually found that using as.numeric and paste worked. I thought it would be easier to write a function that I reference multiple times instead of going through this process time after time. I wrote a function called asNumber that will take in a data frame and a list of columns that the user wants converted to a number.

nameFormat

I also realize that I want a function that can take in a team's name and format it properly for the links on the website. For example "Texas A&M" is formatted as "texas-am". Originally, I thought that all I would have to do for this formatting was convert all characters to lowercase, replace spaces with dashes, and remove any rare characters (such as ampersands or apostrophes). When listing information on the Nebraska Cornhuskers, sports-reference refers to the team as "Nebraska" and uses the string "nebraska" in its URLs. When listing information on the Texas A&M Aggies, sports reference refers to the team as "Texas A&M" and uses the string "texas-am" in its URLs. Unfortunately, sports-reference is not consistent in how they refer to several teams. Sports-reference lists the Louisiana State Tigers as "LSU" on its website, but refers to all webpages in relation to this team using the string "louisiana-state". There were 9 such teams who have are referenced to inconsistently. I "fixed" the formatting of these teams' names by changing "LSU" to "Louisiana-state", changing "USC" to "southern-california", etc.

yearsActive

I also wrote a function called "yearsActive" to give a list of all the active seasons for a given team.

With these "helper" functions and knowledge of the layout of the sports-reference website, we are finally ready to start scraping the data.

Team Lists

TeamList

I begin by finding a list of the 296 teams that sports-reference has information on. This list is referred to as "TeamList".

teamsFiltered

I also found a list of teams that were active for at least one season since the Year 2000. I find this list by scraping the table from the main page, and then only including teams whose latest season was later than 1999. I stored this list as teamsFiltered.

Power 5 Conferences

Running some of the scraping functions takes a long time to run, so I compiled lists of teams that are currently in the Pac12, ACC, Big12, Big10, or SEC. These 5 conferences are commonly referred to as "The power 5", as most of the best teams in the country are consistently in one of these 5 conferences. It is useful to have lists of the more important teams, as it allows us to skip over scraping data on the less important teams.

Scraping the Data Sets

All_Schools_History

I then wrote a function to read in the school history page for teams (example: Nebraska Team History). I called this function read_school_history(). Within the function, I scrape, format, and clean the data.

We want to read in the "School History" pages for every team available. I wrote a mult_history function to use the read_school_history function on multiple schools.

I then ran the mult_history function on the TeamList list of 296 teams.

All_Schools_Team_Stats_Since2000

Each team has a page that has more detailed statistics for each year that team was active (example: Nebraska's 2016 season statistics). I wrote a function to read in the first table displayed on these pages. This table has an offensive row, a defensive row, and a row that takes the difference between the two rows.

I called the function "read_team_stats". Within the function, I scrape, clean, and format the data for a given team and season.

Next, we want to use the read_team_stats function on all the teams.

Since there is only detailed team stats for teams since the year 2000, we will use a cut-off of the year 2000.I first write a function to read in the team stats for a single team for every year. I called this function "all_single_team_stats".

I then wrote a function that uses the all_single_team_stats function for multiple teams. I called this function "mult_team_stats". I then run this code on all the teams who have had a season played since the year 2000. We previously stored the list of these teams as "teamsFiltered".

Power5Schedules

Lastly, each team's season also has a "schedule and results" page (ex: http://www.sports-reference.com/cfb/schools/nebraska/2016-schedule.html). I also wrote a function to read in the table on this page. The function scrapes, cleans, and formats the table. I called this function "read_schedule_results".

I then wrote a function that uses the read_schedule_results function to read in the schedules of every year for a single team. I called this function "all_single_team_schedule".

I then wrote a function that uses the "all_single_team_schedule" function on multiple teams. I called this function "mult_team_schedule".

Reading in the schedules takes a really long time to run, so to avoid running the function on every single team, I only run it for the teams that are currently in the Pac12, ACC, Big12, Big10, or SEC. I then store all of these schedules into a single data frame.

To avoid having to do the webscraping every session, I ended up storing these huge data frames as csv files.

Scraping Summary

Now that we have completed scraping and cleaning the data, we have 3 data sets we would like to analyze. The three data sets are:

1. All_Schools_History

The All_Schools_History dataframe has 13227 rows and 16 columns. The columns record:

- The Team
- The Year
- The Team's Conference
- The Team's Wins for that Season
- The Team's Losses for that season
- The Team's Ties for that Season
- The Team's winning percentage for that season
- The Team's SRS (Simple Rating System) for that season
- The Team's SOS (Strength of Schedule) for that season
- The Team's AP ranking at the beginning of that season
- The Team's AP Ranking at the end of that season
- The Highest AP Ranking the team reached during that season
- The Team's head coach(es) for that season
- The Team's Bowl game for that season

- The Result of that Team's Bowl game for that Season
- Notes on the season (usually a record adjustment by the NCAA)

This information was collected for every season available for 296 Teams.

2. All_Schools_Team_Stats_Since2000

The All_Schools_Team_Stats_Since2000 dataframe has 6172 rows and 26 columns. The columns record:

- The Team
- The Season
- The Team's conference
- The Type of statistics recorded in the row (Either offense, defense, or difference, which takes the offense-defense)
- Number of games that team played that season
- Avg. Number of pass completions per game
- Avg. Number of pass attempts per game
- Avg. Completion percentage per game
- Avg. Passing yards per game
- Avg. Passing touchdowns per game
- Avg. Rushing attempts per game
- Avg. Rushing Yards per game
- Avg. Yards per rush per game
- Avg. Rushing touchdowns per game
- Avg. Plays per game
- Avg. Total Yards per game
- Avg. Yards Per play per game
- Avg. First downs via pass per game
- Avg. First downs via rush per game
- Avg. First downs via penalty per game
- Avg. Total First downs per game
- Avg. Number of Penalties per game
- Avg. Penalty Yards per game
- Avg. Fumbles per game
- Avg. Intercpetions per game
- Avg. Turnovers per. game

This data was collected for the 129 teams that have played at least one season since 2000.

3. Power5Schedules

The Power5Schedules dataframe has 70890 rows and 18 columns. The columns record:

- The Team
- The Season
- The Team's conference
- The Team's Game number for that season
- The Date of the team's game
- The day of the team's game
- The location (home, away, neutral) of the team's game
- The AP rank of the team during the game
- The opponent
- The AP rank of the opponent during the game
- The opponent's conference
- The result of the game (Win, Loss, Tie)

- The points scored by the team in that game
- The points scored by the opponent in that game
- The total number of wins for the season up to that point
- The total number of losses for the season up to that point
- The current streak of the team
- Additional game notes

This data was recorded for every game in the hisotry of the 64 current power five conference teams (teams that are currently in the ACC, Pac12, B1G, Big12, and SEC).

Shiny Apps

Points Scatter plot

For this display, I will allow users to select a conference and select a range of seasons. A scatterplot is then displayed for each team where the x axis is in the number of points given up in a game and the y axis is the number of points scored in a game. The points are colored by result (Win, Loss, Tie) and the shape is selected by location (Home, Away, Neutral). Users can also control "cut-off" lines to split the graph into four quandrants for each team. The scatterplot will display opponent points on the x axis, and team points on the y axis. Users can also choose if they want to include a "win/loss" line, which is a line of y=x. Dots above this line represent a team win, while dots below this line represent a team loss.

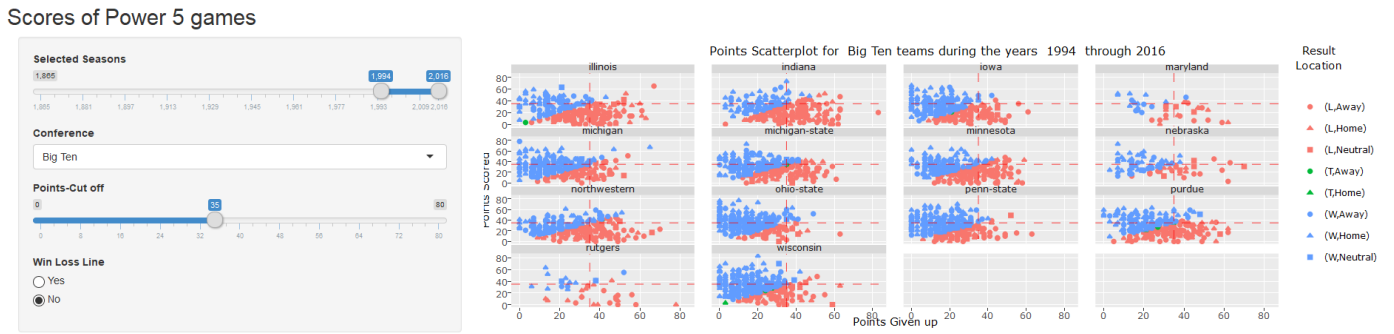

Figure 1: Big Ten Scatterplot for the years 1994 through 2016

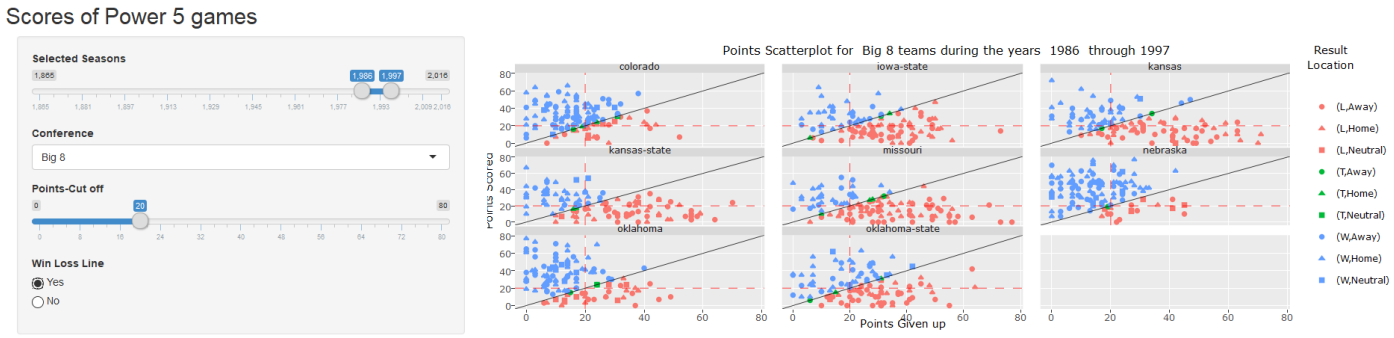

Figure 2: Big 8 Scatterplot for the years 1986 through 1997, with a win-loss line included

Points Scatter plot by Team

This display is nearly identical to the previous display, except users no longer select teams by conference. Instead, users can select individual teams' scatterplots to observe and compare to other teams' scatterplots. In the previous display, users wouldn't be able to compare a Nebraska scatterplot to a West Virginia scatterplot, since Nebraska and West Virginia

were never in the same conference. With this display, users can choose which teams they want to include, allowing users to compare across conferences. Another downside of the previous display is that it only includes a team's games when they were in a certain conference. For example, throughout team history, Nebraska has been in the Big 8 conference, the Big 12 conference, and the Big 10 conference. In the previous display, you wouldn't be able to view a scatterplot of all of Nebraska's games from 1990 to 2000, since Nebraska was in the Big 8 during some of those seasons and the big 12 during the rest of those seasons. With the new display, you select teams whose scatterplots you want to view, not conferences. To avoid making the display cluttered, I only gave the option of including teams either currently in the Big Ten or the Big 12.

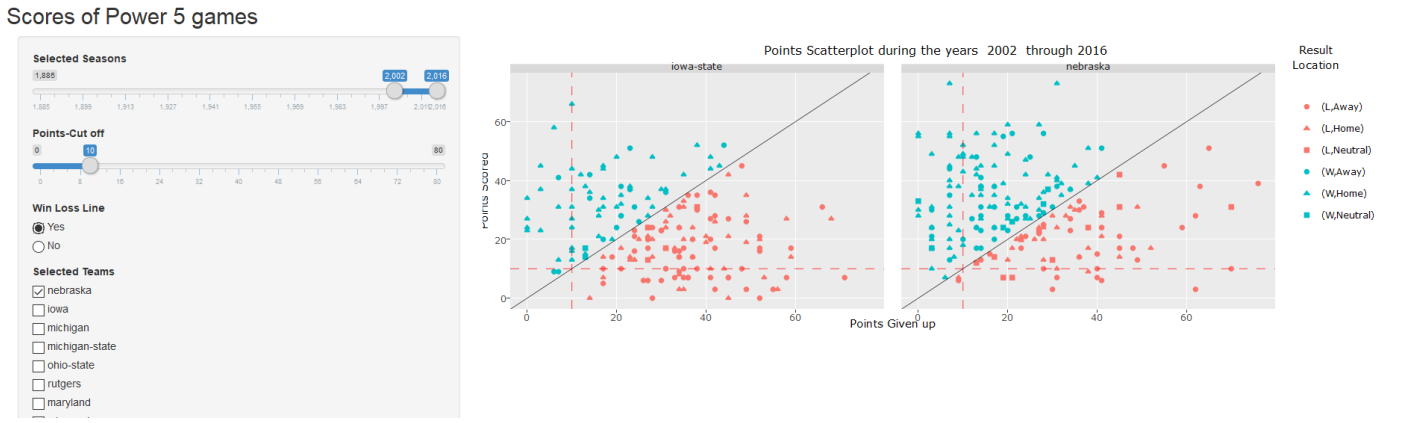

Figure 3: Points scatterplot comparing Iowa State and Nebraska for the years 2002 through 2016

School History

For this display, I will allow users to select a conference, select a range of seasons, and then select a variable recorded in the All_Schools_History data frame. The output will have line graphs (colored by team) displaying the selected statistic over the selected timeline.

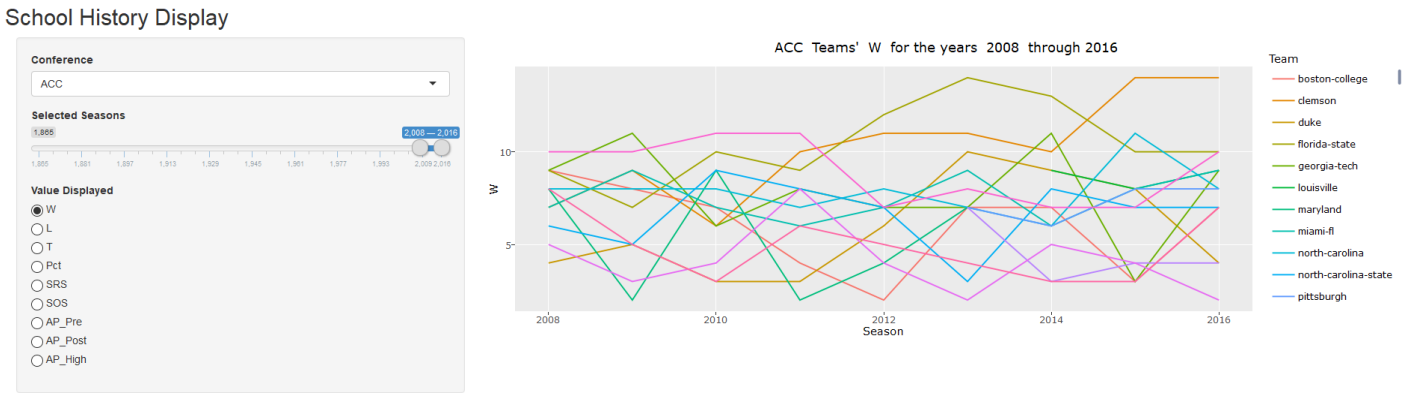

Figure 4: Line Graph of ACC Teams' Wins for the years 2008 through 2016

(

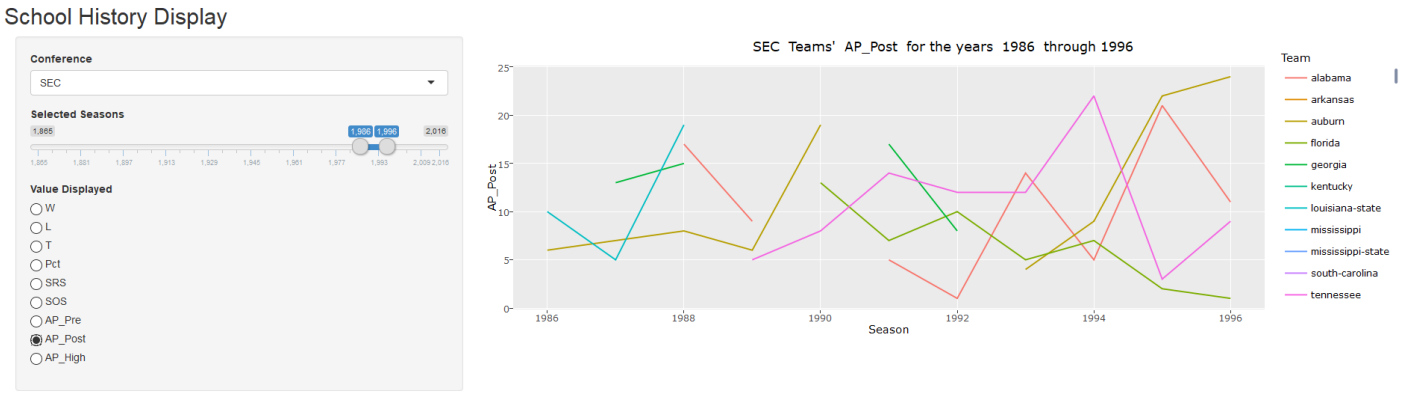

Figure 5: Line Graph of SEC Teams' AP PostSeason Rankings for the years 1986 through 1996

)

School History by Teams

This app is similar to the previous, except instead of selecting a conference, users can now select teams. The motivation for this is the same as why I had both a "Points Scatterplot" app and a "Points Scatterplot by Team" app. To avoid the display being cluttered, I only gave the option of including teams either currently in the Big Ten or the Big 12.

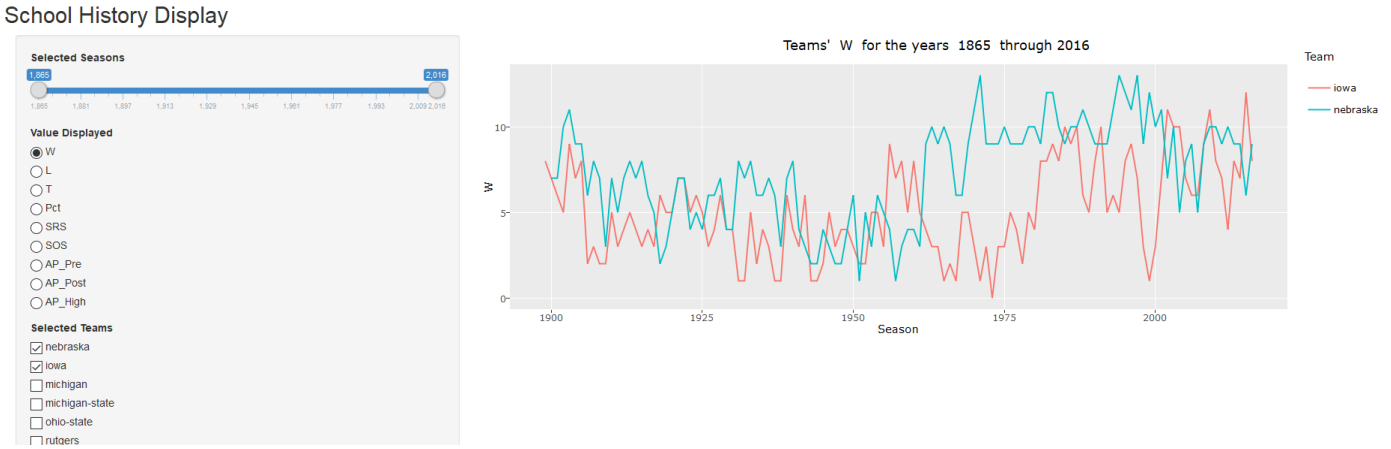

Figure 6: Comparing Iowa and Nebraska's wins from the years 1865 through 2016

(

)

Team Stats

For this display, I will allow users to select a conference, select a range of seasons, and then select both an x-varialbe and a y-variable in the "All_School_Team_Stats" dataframe. There will also be a radio button to select either the offense stats, defense stats, or difference stats. The display will then be a scatterplot.

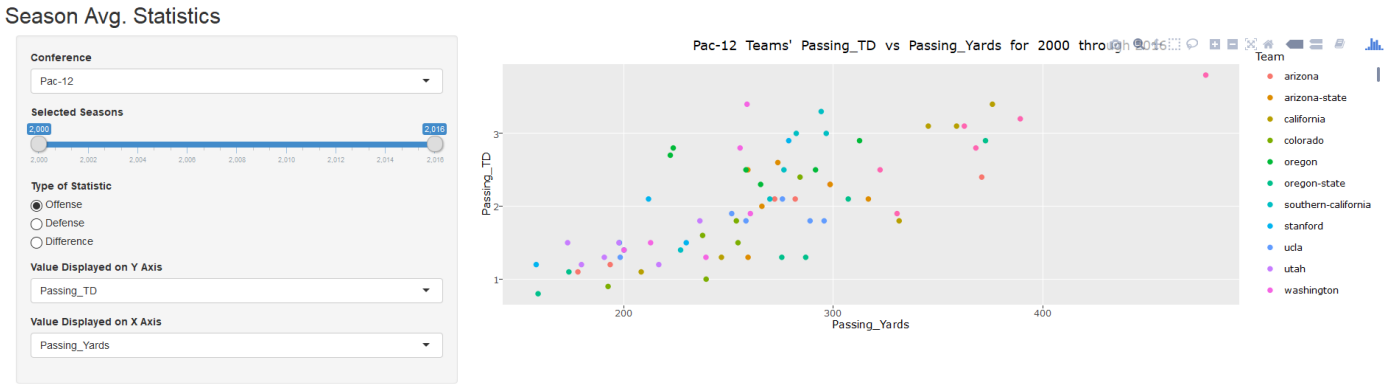

Figure 7: Scatterplot comparing Passing Touchdowns and Passing Yards for Pac 12 teams' offenses for the years 2000 through 2016

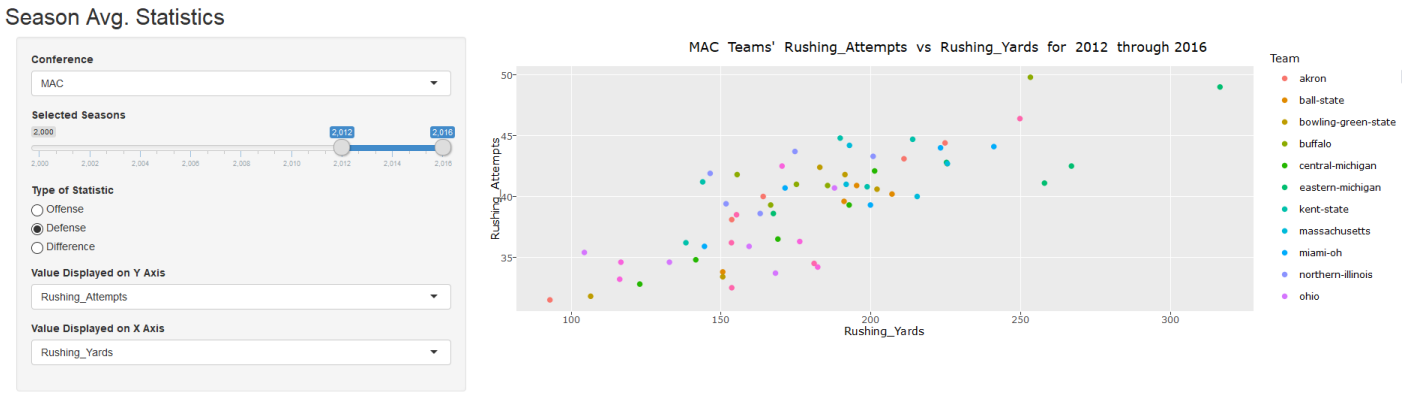

Figure 8: Scatterplot comparing Rushing Yards and Rushing Attempts for MAC teams' defenses for the years 2012 through 2016

Total School History

For this display, I will allow users to select or deselect a checkbox for teams currently in the Big Ten or Big 12. I restricted selection to only these 24 teams to keep the display from becoming cluttered. From there, users can select a radio button to choose which all time statistic the user wants to compare between teams: Total Wins, Total Losses, Total Bowl Appearances, Total Bowl Wins, or all time win percentage up to that point. The user can also select up to what year to view. This display would provide different information then the School History displays since it makes it easier to see how teams' all time statistics compare over time. For instance, you can view how Nebraska's all time wins compares to Minnesota's all time wins and see around when Nebraska passed Minnesota in all time wins. This display also makes it easier to view when a team had a run of many good seasons (ex: Nebraska in the 90's) or when a team had a run of many bad seasons (ex: Iowa State during the 2000's).

To get to this data, I will need to clean and change the data set used. I store this new data frame as totalSchoolHistory.

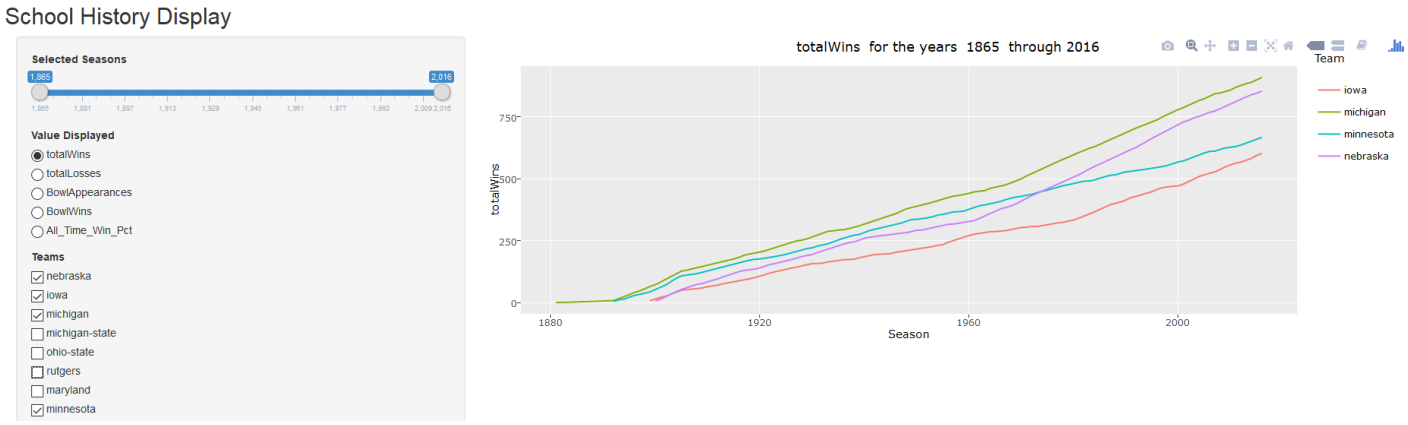

Figure 9: Line graph of Iowa, Michigan, Minnesota, and Nebraska's all time wins from the years 1865 through 2016

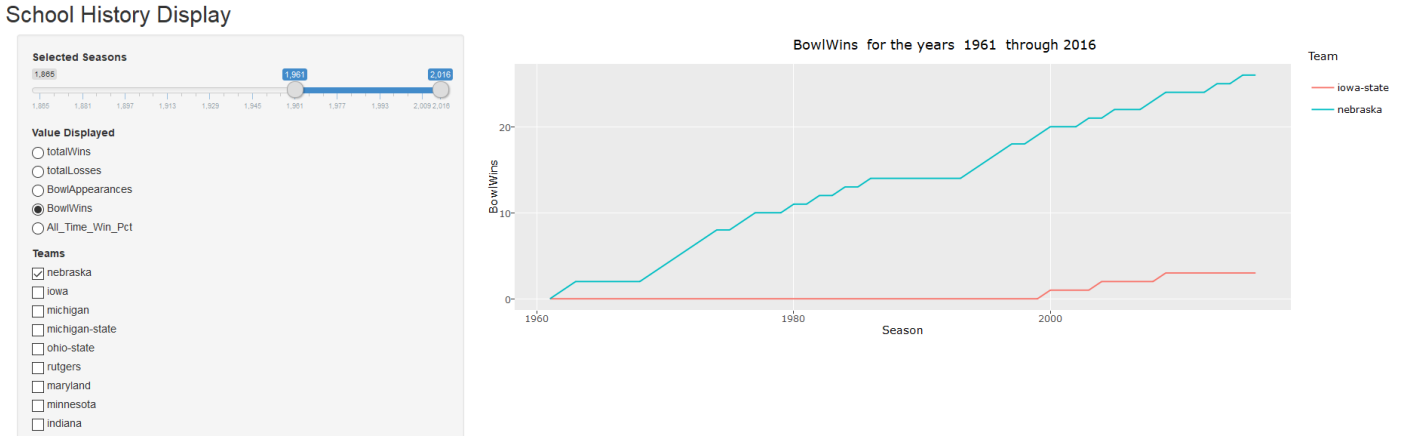

Figure 10: A sad graph comparing Nebraska and Iowa State's all time Bowl Wins for the years 1961 through 2016