Team Branding App – Technical Write-Up

# Overview

This application was developed in February 2020 using the [dash libraries](https://dash.plot.ly/) in Python for Dr. Brandus to help her explore the viewership and stadium attendance of college football teams. She works with various schools to build and manage brand recognition.

# Package Dependencies

* Dash
* Dash\_core\_components
* dash\_html\_components
* dash.dependencies import Input, Output
* pandas
* numpy
* datetime
* plotly.graph\_objects

# Dataframes

* TV\_Joined – contains data from historical games including attendance and TV viewership
* Team\_colors – contains color codes (HEX) of each team in the TV\_Joined dataframe
* Team\_logos – contains links for ESPN homepage and logo for each team in the TV\_Joined dataframe

# Key variables from dataframes

TV\_Joined saved into ratings\_df:

* TeamIDsDate: Unique PK for each individual game
* RATING- tv rating given to the game
* VIEWERS – number of viewers
* Home Team
* Visitor Team
* Attend\_pct – percentage of stadium capacity filled. Can exceed 100%

Team\_logos saved into logos:

* Team
* Link – url of ESPN home page for each team
* Logo – url of team’s logo

Team\_colors saved into team\_colors:

* Team
* Color – color code for each team stored in HEX format

# Variables defined within App

* currentGames – used to limit the games after 2010
* annual\_attn – sum of attendance for the entire year for each team
* teamNamesDict – dictionary with all team names from the dataset

# Elements of App Layout

* Html.Div for separating each element into a section
* Html.H2 for dashboard title
* Html.Img used to display NCAA logo in upper right corner and team logos at the bottom
* Html.P for options selections section
* Dcc.Dropdown for team selection. Set to multi = True to allow for multiple teams selection
* Dcc.RangeSlider used for limiting the years being shown on the graphs
* Dcc.RadioItems used to select home games, away games, or both to be shown in graphs
* Dcc.Tabs used to switch between TV Viewership insights and Stadium attendance insights
* Dcc.Graph displays graphs
* Html.A used to provide links to team logos

# Callbacks

First\_GraphA

* Output to TV Viewers over time scatterplot
* Inputs from dropdown, home-away radio buttons, and year selection slider
* Update\_figure1 takes arguments teamX, Radio\_Selection and Year\_Selection and returns a dictionary with two elements: data and layout. Data is a dictionary of elements to be passed to dcc.Graph to specify each team to display. Layout specifies the title and axis titles to display.

First\_GraphB

* Output to TV rating over time scatterplot
* Inputs from dropdown, home-away radio buttons, and year selection slider
* Update\_figure1 takes arguments teamX, Radio\_Selection and Year\_Selection and returns a dictionary with two elements: data and layout. Data is a dictionary of elements to be passed to dcc.Graph to specify each team to display. Layout specifies the title and axis titles to display.

Second\_GraphA

* Output to stadium attendance by team over time barplot
* Inputs from dropdown, home-away radio buttons, and year selection slider
* Update\_figure1 takes arguments teamX, Radio\_Selection and Year\_Selection and returns a dictionary with two elements: data and layout. Data is a dictionary of elements to be passed to dcc.Graph to specify each team to display. Layout specifies the title and axis titles to display.

Second\_GraphA

* Output to percentage of stadium capacity filled over time scatterplot
* Inputs from dropdown, home-away radio buttons, and year selection slider
* Update\_figure1 takes arguments teamX, Radio\_Selection and Year\_Selection and returns a dictionary with two elements: data and layout. Data is a dictionary of elements to be passed to dcc.Graph to specify each team to display. Layout specifies the title and axis titles to display.

Update\_links

* Takes one argument, teamX, and returns the most recently selected team’s logo (logo) and link to their ESPN homepage (href)

A screenshot of a cell phone

Description automatically generated