College Football (2013-2021) Case Study

by Ryan Eck

Questions to Be Answered

1. What team metrics are most strongly correlated to win percentage?
2. Which offensive styles (if any) have proven to be more successful and/or popular in the last decade?

Data Importation and Cleaning

First, I imported all of the data (College Football Team Stats Seasons 2013 to 2021, uploaded to Kaggle by Jeff Gallini) into Google Sheets. I added a year column to the by-year sheets in preparation for a combined dataset (making sure to format as a text string and Ctrl+Fill to avoid incrementing the years). I then used this function to split the Team column into Team Name and Conference while also removing the parentheses:

```
=SPLIT(LEFT(A2, LEN(A2)-1), "(")
```

This created a minor unforeseen issue with Miami (FL) and Miami (OH), but since the issue was limited to two teams, I decided to fix this manually by removing the parentheses. I also split the Win-Loss column in cfb2021 into Win and Loss to match the rest of the by-year sheets:

```
=SPLIT(F2, “-“)
```

The first benefit of splitting Team into Team Name and Conference was that I could easily see and fill null conference values (Ole Miss in 2013 and 2014 and Pittsburgh in 2014). However, the main reason for doing this is so I can combine Team Name and Year to make a primary key for the combined table. I replaced the Team column with this new Team ID:

```
=CONCATENATE(B2, " (", D2, ")")
```
Example output: Akron (2013)

With the Team ID made, I moved all of the data to the combined sheet. With the spreadsheets combined, I could see college names that needed to be standardized (for example, App State vs Appalachian State). I also discovered that the attributes from 2015 and earlier were mismatched with those from 2016 on, so I fixed the data from 2015 and earlier to match the rest of the dataset.

Dataset Limitations:

- Kickoff Touchbacks are only included from 2016 on, while Total TDs are only included from 2013-2015. When comparing Total TDs to Offensive TDs, I noticed multiple occurrences of the latter being larger than the former, so I decided to remove Total TDs from the dataset.

- Kickoff data and punt data is identical for both returns and kicks. Given that the highest-ranked kickoff teams have a much lower average kickoff return yardage than the lowest-ranked teams, these metrics are likely based off of defensive kick coverage. I decided to remove one of the copies of this data.

With data importation and cleaning complete, I migrated the data into SQL to add calculated attributes that will be useful to me during analysis.

Adding Calculated Attributes in MySQL Workbench

I first made every column in the spreadsheet snake case, since the dataset is being transferred to SQL. I also had to replace the / character with “per” to avoid SQL incorrectly using the division operator. Then, I downloaded the data as a .csv file and used the table data import wizard in MySQL Workbench to migrate the data. After importing the data, I ran a quick query to ensure that the process went smoothly:

```
SELECT *
FROM cfbcombined
LIMIT 10
```

The result revealed that the 2013-2015 data was not successfully imported due to them not having valid integers (instead having null values) in the kickoff_touchbacks column. I ended up filling all the 2013-2015 kickoff_touchbacks data with 0 to facilitate a successful data import. This was done with the understanding that kickoff_touchbacks should only be used from 2016 onward. Only one team since 2016 (Middle Tennessee in 2020) has had zero kickoff_touchbacks in a season, so the new zeros don’t interfere heavily with data exploration. After making this change, full data importation was successful. With the cleaned data in MySQL, I added useful calculated fields, starting with win percent:

```
ALTER TABLE cfbcombined
ADD win_percent double AS (win/(win+loss))
```

Since I’m planning to migrate the data to RStudio for analysis, I decided to make permanent additions to the table instead of creating temp tables.

Creating rush_percent and pass_percent was similar to creating win_percent. To ensure that pass_percent and rush_percent were calculated correctly, I ran a query to check that the sum of the two columns equaled 1:

```
SELECT
    pass_percent,
    rush_percent,
    pass_percent + rush_percent AS total_percent
FROM
       cfbcombined
WHERE
       pass_percent+rush_percent!=1
```

All of the total_percent values not equal to 1 are between 0.999999 and 1, which can be attributed to rounding.

Finally, I decided to make a qualitative variable for rush-heavy vs. pass-heavy teams using this distribution: 10% Rush Heavy, 25% Rush Leaning, 30% Balanced, 25% Pass Leaning, 10% Pass Heavy. I attempted to make a rush_percent_rank column using this query.

```
ALTER TABLE cfbcombined
ADD rush_percent_rank int AS (RANK() OVER(ORDER BY rush_percent DESC))
```

However, since window functions like RANK() cannot be used in an ADD statement, the permanent addition of this column is much easier to implement in a spreadsheet (which I ended up doing). With the successfully implemented rush_percent_rank and the knowledge that there are 1100 rows in the dataset, I was able to create an offense_type column with this query:

```
ALTER TABLE cfbcombined
ADD offense_type text AS
       (CASE
              WHEN
                     rush_percent_rank<=110
              THEN
                     'Rush Heavy'
              WHEN
                     rush_percent_rank>110 AND
```

```
        rush_percent_rank<=385
    THEN
        'Rush Leaning'
    WHEN
        rush_percent_rank>385 AND
        rush_percent_rank<=715
    THEN
        'Balanced'
    WHEN
        rush_percent_rank>715 AND
        rush_percent_rank<=990
    THEN
        'Pass Leaning'
    ELSE
        'Pass Heavy'
    END)
```

I then ran a query to check that offense_type was properly mapped:

```
SELECT
    team_id,
    rush_percent_rank,
    offense_type
FROM
    cfbcombined
ORDER BY
    rush_percent_rank
```

At long last, our dataset is ready to be analyzed.

Data Analysis with RStudio

My analysis documentation was made using R Markdown. The link to that documentation is here.

Presentation

With the analysis completed, I had one other deliverable to create: a presentation to share my findings. The link to that presentation is here.