DATA ACQUISITION PROJECT

Alex Morris, Felipe Chamma, Felipe Formenti, Ryan Speed

University of San Francisco

Introduction

In this project we will investigate NFL play by play data and try to discover interesting patterns, creating engaging statistical visualizations.

To do that, we gathered data from every play since 1998 season until 2014, as well as other information regarding the teams and the players. We got data from the following websites:

www.nfl.com

www.pro-football-reference.com

Our goal is to connect the game result with the game plays, which brings relevant insight on winning teams characteristics. Here are some examples of questions we expect to answer with that:

- On average, how many yards does a winning team achieve? What is the probability of a team with more than 500 total yards lose a game?
- How many yards did each team score and how many did it allow last season? Did the champion stand out in terms of yardage?
- What is the most common outcome of a rush play? Is it better to run towards the middle of the field or to choose a side?

Collecting data

In this section we will briefly describe the data from each source and what we used to collect and store it.

On the first source (nfl.com), each row of the main table brings information on the game, such as home team, away team, game date and time, home team final score, away team final score, venue, etc.

On the latter (pro-football-reference.com), each row brings play by play information such as: date of the game, offense and defense team, game quarter, number of yards, outcome of the play, expected points before and after the game, among other columns.

There is also a detail column consisting in a string from which we can extract many other important variables using regular expressions. Those variables include (but are not limited to): play type (pass or rush), status (complete or incomplete), location (left, right), whether there was a turnover, fumble, touchdown, and so on.

In order to collect data from both sites we used BeatifulSoup package for python. For both websites we need to loop through hundreds of pages to pull all the information we need. A data cleaning stage was also performed using python.

One of the challenges faced on this step was to conciliate the information between both sources such as name abbreviations.

To have an idea of the sizes of the data sets, we had just over 1 million rows of data, considering all tables.

Once we managed to get the data in the right format we generated tables exported them in a csv format. All csv files were then imported to relational database in Postgres, where we analyzed the data using queries.

Finally, the results of those queries were exported to R, where views and plots were build using the package ggplot2.

Database diagram

After scraping and cleaning the data with python, we created the appropriate tables in PostgreSQL. First, we created the 4 main tables, to store the game schedule, the pass plays, rush plays and field goals:

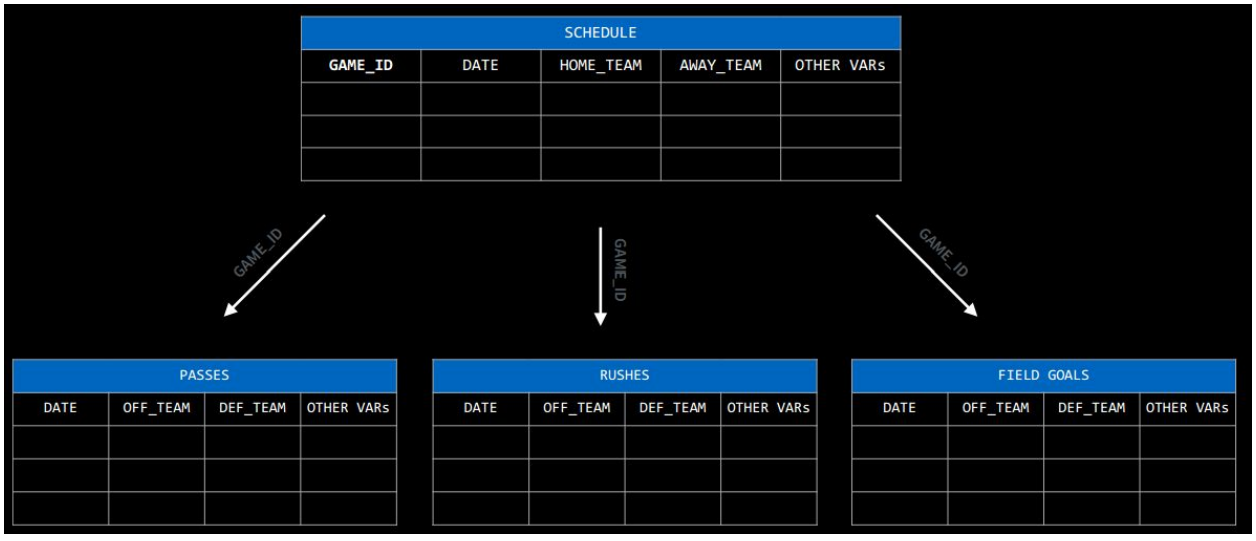
schedule	
game_id	varchar
away_team	varchar
home_team	varchar
date	varchar
time	time
week	integer
venue	varchar
away_score	integer
home_score	integer
away_abv	varchar
home_abv	varchar

passes	
pass_id	bigint
date	date
attacking	varchar
defending	varchar
quarter	integer
time	time
down	integer
togo	integer
field_location	varchar
yard_line	integer
attacking_score	integer
defending_score	integer
passer	varchar
receiver	varchar
completed	boolean
yards	integer
epb	numeric
epa	numeric

runs	
run_id	bigint
date	date
attacking	varchar
defending	varchar
quarter	integer
time	time
down	integer
togo	integer
field_location	varchar
yard_line	integer
attacking_score	integer
defending_score	integer
rusher	varchar
tackler	varchar
direction	varchar
yards	integer
epb	numeric
epa	numeric

goals	
goal_id	bigint
date	date
attacking	varchar
defending	varchar
quarter	integer
time	time
down	integer
togo	integer
field_location	varchar
yard_line	integer
attacking_score	integer
defending_score	integer
kicker	varchar
good	boolean
yards	integer
epb	numeric
epa	numeric

That's the relational model used to join the data when using queries. Basically, we had to join the games from the schedule table to their plays (pass, rush or field goal) using a game id:



We had to customize the game id to use across tables that came from different sources. To identify a game as unique, we used its date and the name of the teams - as teams never play twice on the same day.

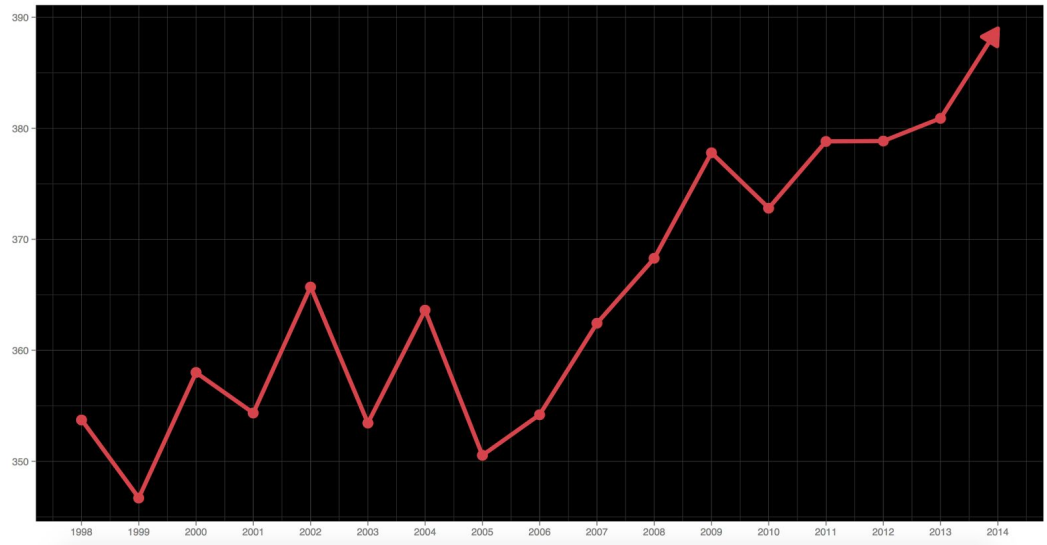
A union table was also created to stack pass and rush plays, in order to calculate total yards per game. We used the following query for that:

CREATE TABLE nfl.plays AS
SELECT "date", season, week, attacking, defending, yards, epb, epa, away_team, away_score, home_team, home_score, winning_team

```
 FROM nfl.passes_new
 UNION
 SELECT "date", season, week, attacking, defending, yards, epb, epa, away_team, away_score, home_team, home_score,
winning_team
 FROM nfl.runs_new;
```

Analyzing the results

1. **How the average yards scored by the winning team changed over seasons**



Despite a natural up and down variation, the average yards scored by the winning team on each game has presented an upward trend.

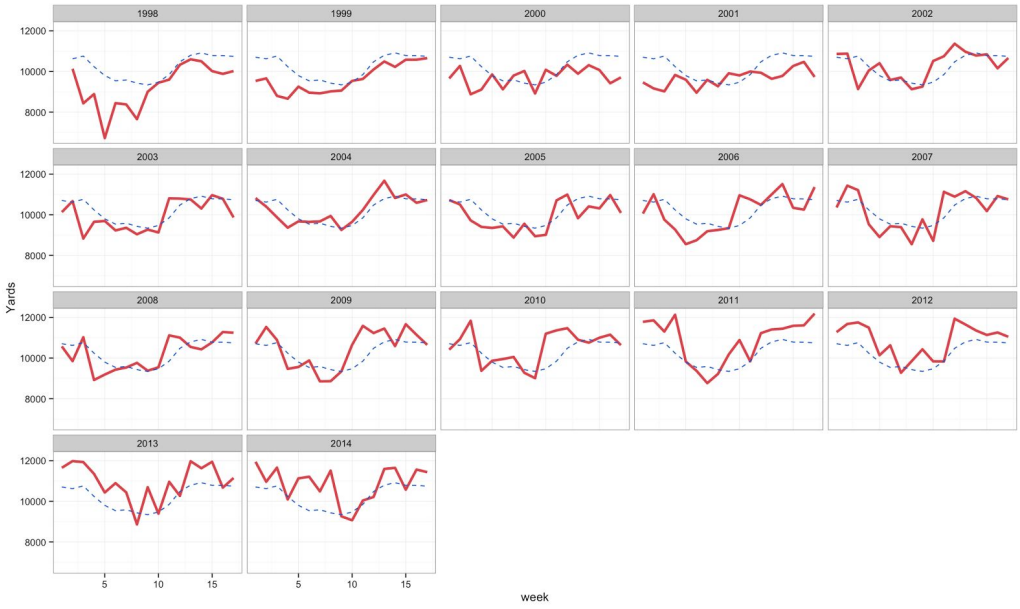
We may attribute that in part to rule changes, which usually aim to protect the players' safety. Since defense teams are much more likely to be infringe those rules when defending their end zone, they give away more yards on the penalties, which ultimately increases the total yards on the game.

The query for such graph is the following:

```
SELECT  year, avg(sum_yards) AS avg_yards
FROM
 (SELECT date_part('year', game_date) AS year, sum(RHS.yards) AS sum_yards
  FROM
    nfl.schedule AS LHS
    LEFT JOIN
    nfl.plays AS RHS
      ON LHS.game_date = RHS.date AND LHS.winning_team = RHS.attacking
```

```
  GROUP BY LHS.game_id, year) AS self
GROUP BY year
ORDER BY year DESC;
```

2. **Average yards run all teams per week, per year. Is there a trend throughout the years?**



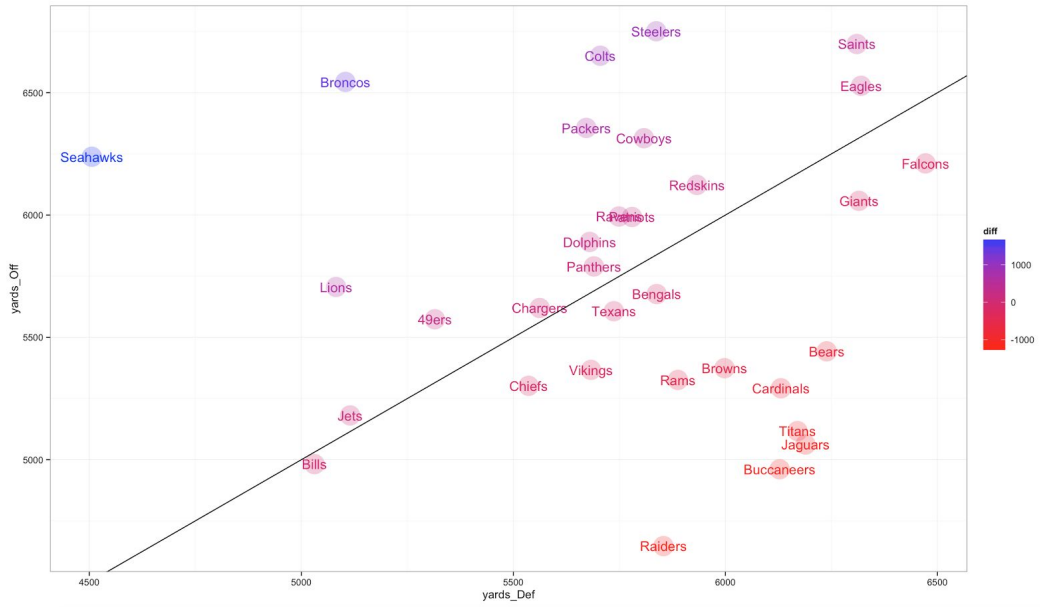
In almost every season we can see a significant drop of total yards after the second week. We are guessing that in the beginning of the season teams tend to surprise each other (since they have no information from previous games), and after two weeks the defense starts to respond better.

The late rise is probably because the teams are taking more risks as the end of the season approaches.

The query for such graph is the following:

```
SELECT sum(yards), season, week
FROM nfl.plays
GROUP BY season, week
ORDER BY season, week;
```

3. **Looking at last season stats, how was the difference of offensive yards to defensive yards and how this translates into the team's success?**



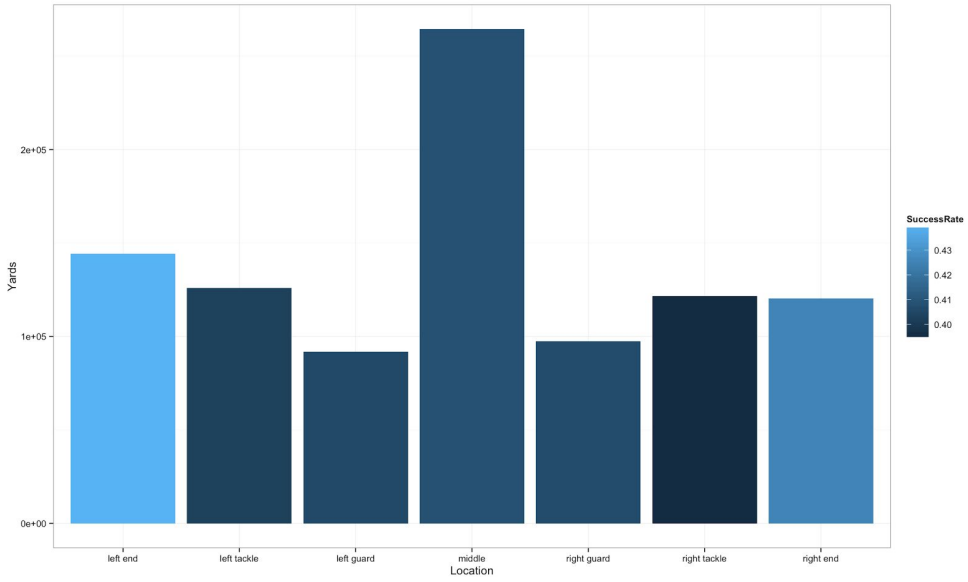
On this plot the total yards gained are on the y-axis and the number of yards suffered are on x-axis, this way we can tell whether a team is better at offense or at defense, and by how much that difference is.

The colors of each point depicts the difference between both axis. All teams above the line identity line obviously finished in a good position at the end of the season.

The query for such graph is the following:

SELECT attacking as team, off_yards, def_yards, off_yards - def_yards as difference FROM
(SELECT attacking, avg(yards) as off_yards from nfl.plays where season = '2014' GROUP BY attacking) as LHS
INNER JOIN
(SELECT defending, avg(yards) as def_yards from nfl.plays where season = '2014' GROUP BY defending) as RHS
ON LHS.attacking = RHS.defending;

4. **What is the success rate of each line for rushing yards?**



On this graph we used to x-axis to illustrate the different lanes inside the field. The y-axis represent the number of yards runned (which can be seen as a measure of attempts) and the color of each bar is colored by the success rate. The success rate was defined as the the following:

- For the first down, a drive is classified as successful in case the runner achieved 50% of the total yards need for the first down.
- For the second down, a drive is classified as successful in case the runner achieved 75% of the total yards need for the first down.
- For the third down, a drive is classified as successful in case the runner achieved 100% of the total yards need for the first down.
- For the fourth down, a drive is classified as successful in case the runner achieved 100% of the total yards need for the first down.

This plot suggests that it is better to run through the ends rather than going through the middle of the field.

The query for such graph is the following:

```
SELECT direction, sum(yards) as total_yards, avg(case
  WHEN down = 1 AND yards >= togo/2 THEN 1
  WHEN down = 2 AND yards >= togo*3/4 THEN 1
  WHEN down = 3 AND yards >= togo THEN 1
  WHEN down = 4 AND yards >= togo THEN 1
  ELSE 0
```

END) AS success_rate
from nfl.runs
GROUP BY direction;

Other interesting stats

5. Who are the most prolific passers since 1998?

SELECT passer, COUNT(passer) AS completed_attempts, sum(yards) AS total_yards
FROM nfl.passes
WHERE completed = TRUE
GROUP BY passer ORDER BY completed_attempts DESC;

Passer	Attempts	Yards
Peyton Manning	5994	70240
Drew Brees	5029	56603
Tom Brady	4623	54018
Brett Favre	4431	49999
Eli Manning	3377	40314
Ben Roethlisberger	3275	39947
Donovan McNabb	3254	38039
Carson Palmer	3137	36076
Matt Hasselbeck	3103	35259
Philip Rivers	3087	37289

6. Who are the most prolific rushers since 1998?

SELECT rusher, COUNT(rusher) AS attempts, sum(yards) AS total_yards
FROM nfl.runs
WHERE yards > 0
GROUP BY rusher ORDER BY total_yards DESC;

Runner	Attempts	Yards
LaDainian Tomlinson	2653	14776
Edgerrin James	2514	13005
Fred Taylor	2051	12271
Steven Jackson	2190	12058
Adrian Peterson	1873	12044
Frank Gore	1995	11601
Thomas Jones	2255	11311
Jamal Lewis	2174	11230
Curtis Martin	2094	10957
Corey Dillon	1888	10839

7. **What is probability of team with over 500 yards losing a game?**

```
SELECT passer, COUNT(passer) AS completed_attempts, sum(yards) AS total_yards
FROM nfl.passes
WHERE completed = TRUE
GROUP BY passer ORDER BY completed_attempts DESC;
```

The probability of that happening is only 1.785 %.