# Why the Packers struggled offensively in the second half of last season?

CS 170A Final Project Report

**Name:** Jerry Liu

**ID:** 404474229

December 12, 2016

# Contents

# 1 Introduction

As the most popular sports in the United States, NFL not only requires physicality from players on the field, but also prefers finely tuned strategies and schemes to put players on the position to win. In the past, NFL coaches only use their experience and feeling to determine the strategies and schemes against their opponents. As data analysis step in as an important factor, coaches can target opponent's weaknesses and maximize their chances to win. However, data still has not been seen as the most important measurement of performance and the most important factor to determine corresponding strategies and schemes. Analysts and scouts still refer to "eye test" to explain what a phenomenon is.

Since Green Bay Packers' embarrassing performance against Denver Broncos last year, quarterback Aaron Rodgers showed the worst performance since he became the starter for that franchise in 2008. In this report, we will use data to find out what leads to the Green Bay Packers' and Aaron Rodgers's miserable second half of the 2015 - 2016 season. Since no one has ever done any data analysis on Aaron Rodgers' and the Packers' performance, any method is unprecedented. I would like to use Principal Component Analysis to determine the factors that affect Packers' performance most. I would also like to find relevant attributes that may affect Aaron Rodgers' passing game.

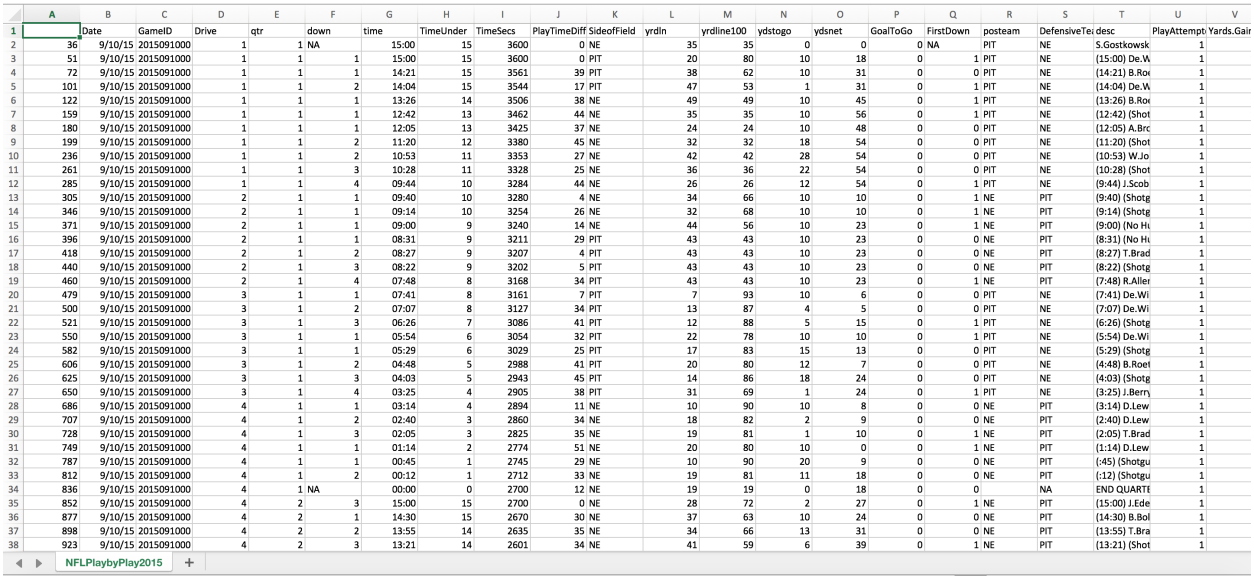
# 2 Possible Factors

There are several possible reasons for their struggles. The most popular reason is: the receivers can't get open. People have the right to believe in this. Denver Broncos started to use press-man coverage against Packers receivers, which means the defensive backs put their hands on Packers receivers to prevent them from gaining separation. According to ESPN [1], the Packers have the League-wide slowest receiving corps. Also, Eddie Lacy, their primary running back, seemed to have overweight issues. There are also criticisms of the coaching staff, but we are unable to verify these claims with data.

# 3 The NFL Play-by-Play Dataset

This is a dataset available on kaggle [2]. This dataset is created by a group of Carnegie Mellon University statistical researchers, led by Maksim Horowitz [2].

The **46,129**-row and **63**-columns dataset, shown in Figure 1, contains all regular season plays from the 2015-2016 NFL season. Each play is broken down into fine details containing players involved, player positions, play results, penalties, etc. With a great amount of detail, we can use it to analyze causes for some phenomena.



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | Date | GameID | Drive | qtr | down | time | TimeUnder | TimeSecs | PlayTimeDiff | SideofField | yrdln | ydsnet | ydstogo | ydsnet | GoalToGo | FirstDown | posteam | DefensiveTea | desc | PlayAttempt | Yards.Gair |
| 2 | 36 | 9/10/15 | 2015091000 | 1 | 1 | NA | 15:00 | 15 | 3600 | 0 | NE | 35 | 35 | 0 | 0 | 0 | NA | PIT | NE | S.Gostkowsk | 1 | |
| 3 | 51 | 9/10/15 | 2015091000 | 1 | 1 | 1 | 15:00 | 15 | 3600 | 0 | PIT | 20 | 80 | 10 | 18 | 0 | 1 | PIT | NE | (15:00) De.W | 1 | |
| 4 | 72 | 9/10/15 | 2015091000 | 1 | 1 | 1 | 14:21 | 15 | 3561 | 39 | PIT | 38 | 62 | 10 | 31 | 0 | 0 | PIT | NE | (14:21) B.Roe | 1 | |
| 5 | 101 | 9/10/15 | 2015091000 | 1 | 1 | 2 | 14:04 | 15 | 3544 | 17 | PIT | 47 | 53 | 1 | 31 | 0 | 1 | PIT | NE | (14:04) De.W | 1 | |
| 6 | 122 | 9/10/15 | 2015091000 | 1 | 1 | 1 | 13:26 | 14 | 3506 | 38 | NE | 49 | 49 | 10 | 45 | 0 | 1 | PIT | NE | (13:26) B.Roe | 1 | |
| 7 | 159 | 9/10/15 | 2015091000 | 1 | 1 | 1 | 12:42 | 13 | 3462 | 44 | NE | 35 | 35 | 10 | 56 | 0 | 1 | PIT | NE | (12:42) (Shot | 1 | |
| 8 | 180 | 9/10/15 | 2015091000 | 1 | 1 | 1 | 12:05 | 13 | 3425 | 37 | NE | 24 | 24 | 10 | 48 | 0 | 0 | PIT | NE | (12:05) A.Brc | 1 | |
| 9 | 199 | 9/10/15 | 2015091000 | 1 | 1 | 2 | 11:20 | 12 | 3380 | 45 | NE | 32 | 32 | 18 | 54 | 0 | 0 | PIT | NE | (11:20) (Shot | 1 | |
| 10 | 236 | 9/10/15 | 2015091000 | 1 | 1 | 2 | 10:53 | 11 | 3353 | 27 | NE | 42 | 42 | 28 | 54 | 0 | 0 | PIT | NE | (10:53) W.Jo | 1 | |
| 11 | 261 | 9/10/15 | 2015091000 | 1 | 1 | 3 | 10:28 | 11 | 3328 | 25 | NE | 36 | 36 | 22 | 54 | 0 | 0 | PIT | NE | (10:28) (Shot | 1 | |
| 12 | 285 | 9/10/15 | 2015091000 | 1 | 1 | 4 | 09:44 | 10 | 3284 | 44 | NE | 26 | 26 | 12 | 54 | 0 | 1 | PIT | NE | (9:44) J.Scob | 1 | |
| 13 | 305 | 9/10/15 | 2015091000 | 2 | 1 | 1 | 09:40 | 10 | 3280 | 4 | NE | 34 | 66 | 10 | 10 | 0 | 1 | NE | PIT | (9:40) (Shotg | 1 | |
| 14 | 346 | 9/10/15 | 2015091000 | 2 | 1 | 1 | 09:14 | 10 | 3254 | 26 | NE | 32 | 68 | 10 | 10 | 0 | 1 | NE | PIT | (9:14) (Shotg | 1 | |
| 15 | 371 | 9/10/15 | 2015091000 | 2 | 1 | 1 | 09:00 | 9 | 3240 | 14 | NE | 44 | 56 | 10 | 23 | 0 | 1 | NE | PIT | (9:00) (No Hu | 1 | |
| 16 | 396 | 9/10/15 | 2015091000 | 2 | 1 | 1 | 08:31 | 9 | 3211 | 29 | PIT | 43 | 43 | 10 | 23 | 0 | 0 | NE | PIT | (8:31) (No Hu | 1 | |
| 17 | 418 | 9/10/15 | 2015091000 | 2 | 1 | 2 | 08:27 | 9 | 3207 | 4 | PIT | 43 | 43 | 10 | 23 | 0 | 0 | NE | PIT | (8:27) T.Brad | 1 | |
| 18 | 440 | 9/10/15 | 2015091000 | 2 | 1 | 3 | 08:22 | 9 | 3202 | 5 | PIT | 43 | 43 | 10 | 23 | 0 | 0 | NE | PIT | (8:22) (Shotg | 1 | |
| 19 | 460 | 9/10/15 | 2015091000 | 2 | 1 | 4 | 07:48 | 8 | 3168 | 34 | PIT | 43 | 43 | 10 | 23 | 0 | 1 | NE | PIT | (7:48) R.Aller | 1 | |
| 20 | 479 | 9/10/15 | 2015091000 | 3 | 1 | 1 | 07:41 | 8 | 3161 | 7 | PIT | 7 | 93 | 10 | 6 | 0 | 0 | PIT | NE | (7:41) De.Wi | 1 | |
| 21 | 500 | 9/10/15 | 2015091000 | 3 | 1 | 2 | 07:07 | 8 | 3127 | 34 | PIT | 13 | 87 | 4 | 5 | 0 | 0 | PIT | NE | (7:07) De.Wi | 1 | |
| 22 | 521 | 9/10/15 | 2015091000 | 3 | 1 | 3 | 06:26 | 7 | 3086 | 41 | PIT | 12 | 88 | 5 | 15 | 0 | 1 | PIT | NE | (6:26) (Shotg | 1 | |
| 23 | 550 | 9/10/15 | 2015091000 | 3 | 1 | 1 | 05:54 | 6 | 3054 | 32 | PIT | 22 | 78 | 10 | 10 | 0 | 1 | PIT | NE | (5:54) De.Wi | 1 | |
| 24 | 582 | 9/10/15 | 2015091000 | 3 | 1 | 1 | 05:29 | 6 | 3029 | 25 | PIT | 17 | 83 | 15 | 13 | 0 | 0 | PIT | NE | (5:29) (Shotg | 1 | |
| 25 | 606 | 9/10/15 | 2015091000 | 3 | 1 | 2 | 04:48 | 5 | 2988 | 41 | PIT | 20 | 80 | 12 | 7 | 0 | 0 | PIT | NE | (4:48) B.Roet | 1 | |
| 26 | 625 | 9/10/15 | 2015091000 | 3 | 1 | 3 | 04:03 | 5 | 2943 | 45 | PIT | 14 | 86 | 18 | 24 | 0 | 0 | PIT | NE | (4:03) (Shotg | 1 | |
| 27 | 650 | 9/10/15 | 2015091000 | 3 | 1 | 4 | 03:25 | 4 | 2905 | 38 | PIT | 31 | 69 | 1 | 24 | 0 | 1 | PIT | NE | (3:25) J.Berry | 1 | |
| 28 | 686 | 9/10/15 | 2015091000 | 4 | 1 | 1 | 03:14 | 4 | 2894 | 11 | NE | 10 | 90 | 10 | 8 | 0 | 0 | NE | PIT | (3:14) D.Lew | 1 | |
| 29 | 707 | 9/10/15 | 2015091000 | 4 | 1 | 2 | 02:40 | 3 | 2860 | 34 | NE | 18 | 82 | 2 | 9 | 0 | 0 | NE | PIT | (2:40) D.Lew | 1 | |
| 30 | 728 | 9/10/15 | 2015091000 | 4 | 1 | 3 | 02:05 | 3 | 2825 | 35 | NE | 19 | 81 | 1 | 10 | 0 | 1 | NE | PIT | (2:05) T.Brad | 1 | |
| 31 | 749 | 9/10/15 | 2015091000 | 4 | 1 | 1 | 01:14 | 2 | 2774 | 51 | NE | 20 | 80 | 10 | 0 | 0 | 1 | NE | PIT | (1:14) D.Lew | 1 | |
| 32 | 787 | 9/10/15 | 2015091000 | 4 | 1 | 1 | 00:45 | 1 | 2745 | 29 | NE | 10 | 90 | 20 | 9 | 0 | 0 | NE | PIT | (:45) (Shotgu | 1 | |
| 33 | 812 | 9/10/15 | 2015091000 | 4 | 1 | 2 | 00:12 | 1 | 2712 | 33 | NE | 19 | 81 | 11 | 18 | 0 | 0 | NE | PIT | (:12) (Shotgu | 1 | |
| 34 | 836 | 9/10/15 | 2015091000 | 4 | 1 | NA | 00:00 | 0 | 2700 | 12 | NE | 19 | 19 | 0 | 18 | 0 | 0 | | NA | END QUARTE | 1 | |
| 35 | 852 | 9/10/15 | 2015091000 | 4 | 2 | 3 | 15:00 | 15 | 2700 | 0 | NE | 28 | 72 | 2 | 27 | 0 | 1 | NE | PIT | (15:00) J.Ede | 1 | |
| 36 | 877 | 9/10/15 | 2015091000 | 4 | 2 | 1 | 14:30 | 15 | 2670 | 30 | NE | 37 | 63 | 10 | 24 | 0 | 0 | NE | PIT | (14:30) B.Bol | 1 | |
| 37 | 898 | 9/10/15 | 2015091000 | 4 | 2 | 2 | 13:55 | 14 | 2635 | 35 | NE | 34 | 66 | 13 | 31 | 0 | 0 | NE | PIT | (13:55) T.Bra | 1 | |
| 38 | 923 | 9/10/15 | 2015091000 | 4 | 2 | 3 | 13:21 | 14 | 2601 | 34 | NE | 41 | 59 | 6 | 39 | 0 | 1 | NE | PIT | (13:21) (Shot | 1 | |

NFLPlaybyPlay2015

Figure 1: A screenshot of the CMU *NFL Play-by-Play Dataset*. This dataset is not numeric oriented and not very friendly to data analysis.

The data can be processed as follows:

```
[NUM, TXT, RAW] = xlsread('NFLPlaybyPlay2015.xlsx');
% Turn the cell into a string matrix
RAWStr = string(RAW);

% Offensive / Defensive Team information is in column 18, 19 respectively
Off = RAWStr(:, 18);
Def = RAWStr(:, 19);
PackersOffIndex = find(Off == 'GB') - 1;
PackersDefIndex = find(Def == 'GB') - 1;

```

```
Attr = RAWStr(2, :);
PackersOffNum = NUM(PackersOffIndex, 2 : 66);
PackersDeffNum = NUM(PackersDefIndex, 2 : 66);
```

Here the `NUM` matrix contains all numeric data; `TXT` cell contains all non-numeric data and the whole dataset is represented in the cell `RAW`.

# 4 NFL Savant Play-by-Play Dataset

However, after some arduous digging of the unfriendly kaggle dataset [2], I decided to move on to the more friendly *NFL Savant* dataset [3]. This dataset is available on nfl savant [3]. This dataset consists of **46278**-row and **45**-columns of data. Most of the values are numeric, and unlike the kaggle data in Section 3, logical values are expressed in 0 / 1, which is much easier to analyze. The dataset is shown in Figure 2.

| GameId | GameDate | Quarter | Minute | Second | OffenseTeam | DefenseTeam | Down | ToGo | YardLine | SeriesFirstDown | NextScore | Description | TeamWin | SeasonYear | Yards | Formation | PlayType |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ########## | 9/10/15 | 2 | 2 | 0 | | PIT | 0 | 0 | 0 | 1 | 0 | TWO-MINUT | 0 | 2015 | 0 | UNDER CENTER | |
| 2015091300 | 9/13/15 | 3 | 4 | 50 | | GB | 0 | 0 | 0 | 1 | 0 | TIMEOUT AT | 0 | 2015 | 0 | UNDER CENTER | |
| 2015091300 | 9/13/15 | 4 | 0 | 0 | | GB | 0 | 0 | 0 | 1 | 0 | END GAME | 0 | 2015 | 0 | UNDER CENTER | |
| 2015091301 | 9/13/15 | 2 | 2 | 0 | | SEA | 0 | 0 | 0 | 1 | 0 | TWO-MINUT | 0 | 2015 | 0 | UNDER CENTER | |
| 2015091302 | 9/13/15 | 1 | 0 | 0 | | CAR | 0 | 0 | 0 | 1 | 0 | END QUARTE | 0 | 2015 | 0 | | |
| 2015091303 | 9/13/15 | 2 | 9 | 48 | WAS | MIA | 2 | 2 | 20 | 1 | 0 | (9:48) 46-A.M | 0 | 2015 | 4 | UNDER CENT | RUSH |
| 2015091303 | 9/13/15 | 2 | 9 | 13 | WAS | MIA | 1 | 10 | 24 | 1 | 0 | (9:13) 8-K.CC | 0 | 2015 | 22 | UNDER CENT | PASS |
| 2015091303 | 9/13/15 | 4 | 2 | 33 | WAS | MIA | 2 | 11 | 76 | 0 | 0 | (2:33) (SHOT | 0 | 2015 | 4 | SHOTGUN | PASS |
| 2015091304 | 9/13/15 | 1 | 4 | 57 | BUF | IND | 2 | 20 | 37 | 0 | 0 | (4:57) 5-T.TA | 0 | 2015 | 0 | UNDER CENT | PASS |
| 2015091304 | 9/13/15 | 2 | 12 | 24 | | IND | 0 | 0 | 0 | 1 | 0 | TIMEOUT #1 | 0 | 2015 | 0 | UNDER CENT | TIMEOUT |
| 2015091304 | 9/13/15 | 3 | 14 | 4 | BUF | IND | 1 | 10 | 68 | 0 | 0 | (14:04) 5-T.T | 0 | 2015 | 3 | UNDER CENT | PASS |
| 2015091304 | 9/13/15 | 3 | 13 | 27 | BUF | IND | 2 | 7 | 71 | 0 | 0 | (13:27) 25-L. | 0 | 2015 | -1 | UNDER CENT | RUSH |
| 2015091304 | 9/13/15 | 3 | 0 | 0 | | IND | 0 | 0 | 0 | 1 | 0 | END QUARTE | 0 | 2015 | 0 | | |
| 2015091304 | 9/13/15 | 4 | 13 | 33 | BUF | IND | 4 | 17 | 13 | 0 | 0 | (13:33) 6-C.S | 0 | 2015 | 0 | PUNT | PUNT |
| 2015091304 | 9/13/15 | 4 | 2 | 4 | | IND | 0 | 0 | 0 | 1 | 0 | TIMEOUT #2 | 0 | 2015 | 0 | UNDER CENT | TIMEOUT |
| 2015091305 | 9/13/15 | 1 | 8 | 20 | CLE | NYJ | 3 | 1 | 28 | 1 | 0 | (8:20) (SHOT | 0 | 2015 | 11 | SHOTGUN | PASS |
| 2015091305 | 9/13/15 | 1 | 7 | 26 | CLE | NYJ | 2 | 9 | 40 | 0 | 0 | (7:26) 34-I.CI | 0 | 2015 | 8 | UNDER CENT | RUSH |
| 2015091305 | 9/13/15 | 2 | 2 | 0 | | CLE | 0 | 0 | 0 | 1 | 0 | TWO-MINUT | 0 | 2015 | 0 | UNDER CENTER | |
| 2015091305 | 9/13/15 | 2 | 2 | 0 | NYJ | CLE | 2 | 2 | 47 | 1 | 0 | (2:00) (SHOT | 0 | 2015 | 11 | SHOTGUN | PASS |
| 2015091305 | 9/13/15 | 2 | 0 | 0 | | CLE | 0 | 0 | 0 | 1 | 0 | END QUARTE | 0 | 2015 | 0 | | |
| 2015091305 | 9/13/15 | 3 | 10 | 52 | CLE | NYJ | 3 | 7 | 37 | 1 | 0 | (10:52) (SHO | 0 | 2015 | 18 | SHOTGUN | PASS |
| 2015091306 | 9/13/15 | 1 | 0 | 0 | | KC | 0 | 0 | 0 | 1 | 0 | END QUARTE | 0 | 2015 | 0 | | |
| 2015091306 | 9/13/15 | 3 | 4 | 49 | KC | HOU | 1 | 10 | 35 | 0 | 0 | (4:49) 11-A.S | 0 | 2015 | 0 | UNDER CENT | PASS |
| 2015091306 | 9/13/15 | 3 | 4 | 42 | KC | HOU | 2 | 10 | 35 | 0 | 0 | (4:42) (SHOT | 0 | 2015 | 2 | SHOTGUN | PASS |
| 2015091306 | 9/13/15 | 4 | 3 | 26 | KC | HOU | 2 | 6 | 24 | 0 | 0 | (3:26) 11-A.S | 0 | 2015 | 0 | UNDER CENT | PASS |
| 2015091306 | 9/13/15 | 4 | 1 | 22 | KC | HOU | 1 | 10 | 50 | 0 | 0 | (1:22) 11-A.S | 0 | 2015 | 0 | UNDER CENT | QB KNEEL |
| 2015091307 | 9/13/15 | 1 | 7 | 20 | NO | ARI | 2 | 1 | 41 | 0 | 0 | (7:20) 29-K.R | 0 | 2015 | 0 | UNDER CENT | RUSH |
| 2015091307 | 9/13/15 | 3 | 9 | 2 | NO | ARI | 1 | 10 | 43 | 0 | 0 | (9:02) 29-K.R | 0 | 2015 | 8 | UNDER CENT | RUSH |
| 2015091308 | 9/13/15 | 1 | 0 | 0 | | DET | 0 | 0 | 0 | 1 | 0 | END QUARTE | 0 | 2015 | 0 | | |
| 2015091308 | 9/13/15 | 2 | 5 | 2 | SD | DET | 0 | 0 | 35 | 1 | 0 | 2-J.LAMBO K | 0 | 2015 | 0 | UNDER CENT | KICK OFF |
| 2015091308 | 9/13/15 | 4 | 14 | 28 | SD | DET | 2 | 7 | 50 | 0 | 0 | (14:28) (SHO | 0 | 2015 | -3 | SHOTGUN | PASS |
| 2015091309 | 9/13/15 | 2 | 2 | 0 | | BAL | 0 | 0 | 0 | 1 | 0 | TWO-MINUT | 0 | 2015 | 0 | UNDER CENTER | |
| 2015091309 | 9/13/15 | 4 | 14 | 12 | BAL | DEN | 2 | 10 | 35 | 0 | 0 | (14:12) 5-J.FI | 0 | 2015 | 0 | UNDER CENT | PASS |
| 2015091309 | 9/13/15 | 4 | 3 | 0 | | BAL | 0 | 0 | 0 | 1 | 0 | TIMEOUT #1 | 0 | 2015 | 0 | UNDER CENT | TIMEOUT |
| 2015091310 | 9/13/15 | 2 | 13 | 41 | OAK | CIN | 4 | 6 | 47 | 0 | 0 | (13:41) 7-M.I | 0 | 2015 | 0 | PUNT | PUNT |
| 2015091310 | 9/13/15 | 3 | 7 | 50 | CIN | OAK | 1 | 10 | 15 | 0 | 0 | (7:50) 14-A.D | 0 | 2015 | 8 | UNDER CENT | PASS |
| 2015091310 | 9/13/15 | 3 | 0 | 47 | OAK | CIN | 2 | 6 | 32 | 1 | 0 | (:47) (NO HU | 0 | 2015 | 8 | NO HUDDLE | PASS |

pbp-2015

Figure 2: A screenshot of the *Play-by-Play 2015 Dataset*. This dataset is much more numeric oriented and more friendly to data analysis.

It consists of not only numeric data, but also textual data (i.e. strings). I first converted the `csv` file into an `xlsx` file, and then used `xlsread` to get the raw data. The code is shown on the next page:

```
[NUM, TXT, RAW] = xlsread('pbp-2015.xlsx');
RAW = string(RAW);
TXT = string(TXT);

% Attributes, ["GameId" "GameDate" "Quarter" "Minute" ...]
Attr = RAW(1, :);

% Get the col numbers that represent teams
OffCol = Attr == 'OffenseTeam';
DefCol = Attr == 'DefenseTeam';
OffIndex = find(RAW(:, OffCol) == 'GB') - 1;
DefIndex = find(RAW(:, DefCol) == 'GB') - 1;

% Filter Data
Off = NUM(OffIndex, :);
OffRaw = RAW(OffIndex + 1, :);
DenverGameIndex = find(OffRaw(:, DefCol) == 'DEN', 1);
offLength = length(Off);
Def = NUM(DefIndex, :);

% Get the games, in sequence
Games = process_game(RAW(OffIndex + 1, DefCol));
```

The source code of the function `process_game` is below:

```
% The source code of function process_game:
function Games = process_game(GameLog)
    % Allocate Storage
    Games = GameLog(1 : 16);
    prevGame = GameLog(1);
    j = 2;
    for i = 2 : length(GameLog)
        if GameLog(i) ~= prevGame
            prevGame = GameLog(i);
            Games(j) = prevGame;
            j = j + 1;
        end
    end
end
```

# 5 Analysis

## 5.1 Principal Component Analysis

Since there are a lot of attributes for this dataset, we want to use *Principal Component Analysis* (PCA) to find out which attribute(s) have a significant impact on the whole data. First select relevant numeric attributes according to my football knowledge, shown in Table 1.

| **Attribute** | **Type** | **Meaning** |
|---|---|---|
| Yards | Numeric | Yards Per Play. |
| Down | Numeric | The down count in the current drive. |
| ToGo | Numeric | Yards needed to get a first down / score. |
| YardLine | Numeric | The ball's snap position (at $n$ yard line) on the field. |
| IsRush | Logical | Indicates whether the current play is a rush. |
| IsPass | Logical | Indicates whether the current play is a pass. |
| IsIncomplete | Logical | Indicates whether the pass is incomplete. |
| IsTouchdown | Logical | Indicates whether the current play is a TD. |
| IsSack | Logical | Indicates whether the passer is sacked. |
| IsInterception | Logical | Indicates whether the passer is intercepted. |
| IsFumble | Logical | Indicates whether the offense fumbles the ball. |
| IsPenalty | Logical | Indicates whether a penalty flag is on the field. |
| IsPenaltyAccepted | Logical | Indicates whether the penalty is accepted. |
| IsNoPlay | Logical | Indicates whether the current play is a no play (offsetting penalty). |

Table 1: Table of Projected Attributes and Corresponding Meanings

This forms a $1402 \times 14$ matrix and we want to use PCA to project it onto a 3D plane. We make the following distinction:

- If $YPP$(yards per play) $< 5$, then we categorize it as *low YPP*.
- If $5 \geq YPP < 15$, then we categorize it as *medium YPP*.
- If $YPP \geq 15$, then we categorize it as *high YPP*.

Then we use PCA (SVD algorithm) to extract the first three principal components. Figure 3 is a nice 3D plot from our projection and Figure 4 is an overview of the 3D plot.

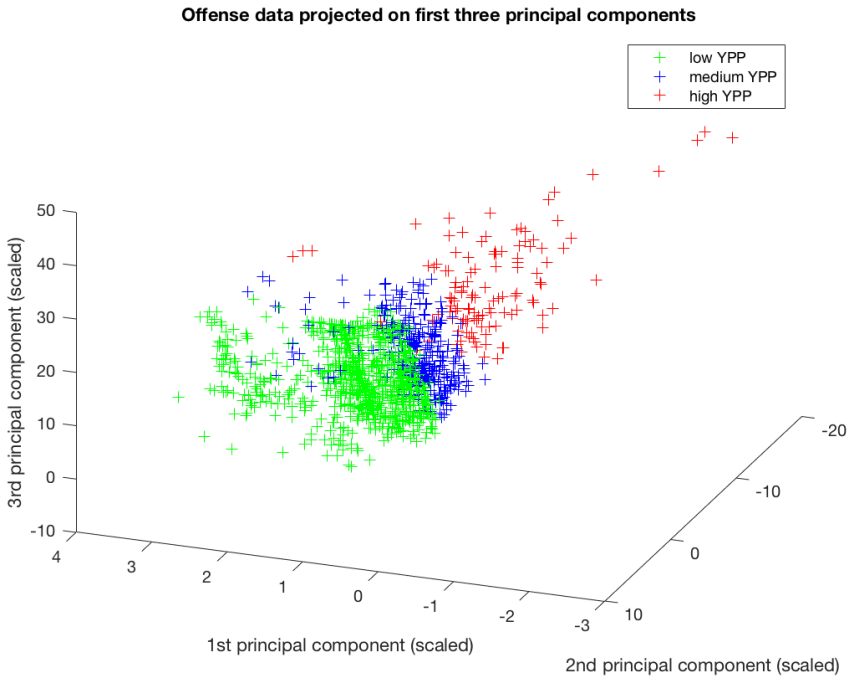



Figure 3: The PCA 3D Projection of our offense data

The code for projection is shown below:

```
% Select Relevant Attribute
IndexToProj = [20, 8 : 10, 23 : 26, 28, 33 : 35, 41, 43];
% Yards, Down, ToGo, YardLine, IsRush, IsPass, IsIncomplete, IsTouchdown, IsSack, ...
    IsInterception, IsFumble, IsPenalty, IsPenaltyAccepted, IsNoPlay
AttrToProj = Attr(:, IndexToProj);
OffToProj = double(OffRaw(:, IndexToProj));

% Sort the data according to "Yards"
[Val, I] = sort(OffToProj, 1, 'ascend');
OffToProj = OffToProj(I(:, 1), :);

% Divide Line between low, medium and high YPP
LowYards = find(OffToProj(:, 1) == 5, 1) - 1;
GoodYards = find(OffToProj(:, 1) == 15, 1) - 1;

```

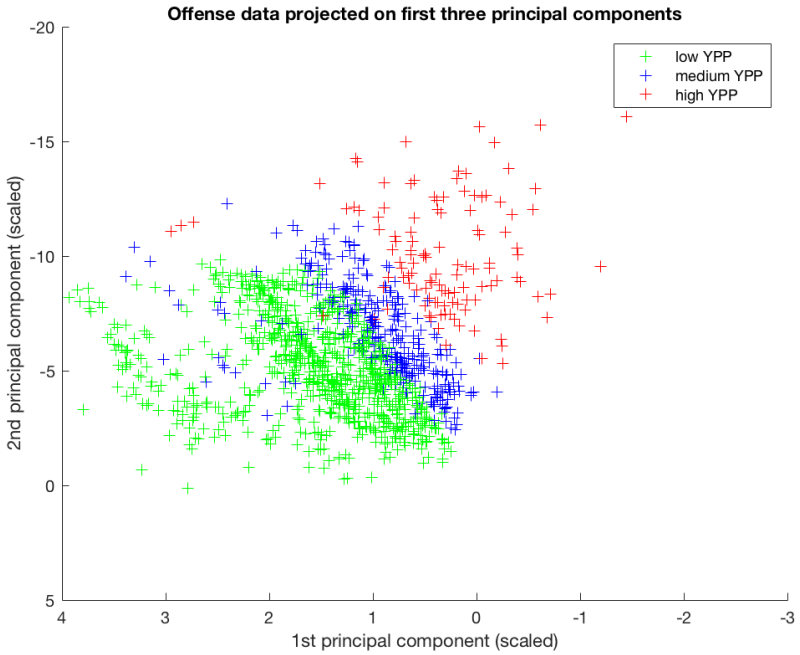



Figure 4: Overview of the PCA 3D Projection of our offense data

```
% Normalize the data using correlation matrix
OffToProjNorm = corr(OffToProj);

% PCA
[U,S,V] = svd(OffToProjNorm);
n = size(OffToProj, 1);

PrincipalComponent1 = U(:,1);
PrincipalComponent2 = U(:,2);
PrincipalComponent3 = U(:,3);
SingularValues = diag(S(1:3,1:3));

% Projection to 3D
X = OffToProj * PrincipalComponent1;
Y = OffToProj * PrincipalComponent2;
Z = OffToProj * PrincipalComponent3;

figure
hold on
plot3( X(1 : LowYards), Y(1 : LowYards), Z(1 : LowYards), 'g+' )
```

```
plot3( X(LowYards : GoodYards), Y(LowYards : GoodYards), Z(LowYards : GoodYards), ...
    'b+' )
plot3( X(GoodYards : n),   Y(GoodYards : n),   Z(GoodYards : n),   'r+' )
xlabel('1st principal component (scaled)')
ylabel('2nd principal component (scaled)')
zlabel('3rd principal component (scaled)')
title('Offense data projected on first three principal components')
legend('low YPP','medium YPP','high YPP')
rotate3d on
hold off
```

We can see a rough 3-cluster distribution from the graph, corresponding to low, medium and high YPP. With the first three Principal Components (PC), we want to find which attributes have significant weights in each of the three PC. The limit is quite arbitrary; for example, we can choose the attributes that exceed the mean of the absolute values of one PC vector. The values for all three vectors are shown in Table 2.

| **Vector Index** | $1^{\text{st}}$ **PC** | $2^{\text{nd}}$ **PC** | $3^{\text{rd}}$ **PC** |
|---|---|---|---|
| 1 | −0.051190666 | −0.211690262 | **0.603892049** |
| 2 | 0.143645804 | −0.023413132 | −0.117699317 |
| 3 | 0.051154443 | −0.149140799 | 0.02702564 |
| 4 | 0.012284638 | −0.0742752 | 0.114178674 |
| 5 | −0.216100654 | **0.445428616** | 0.268642516 |
| 6 | 0.1825421 | **−0.634452804** | 0.011132822 |
| 7 | 0.220952173 | **−0.395082098** | −0.302385616 |
| 8 | 0.010351095 | −0.104089723 | 0.29517581 |
| 9 | −0.000825734 | 0.17579242 | **−0.500473668** |
| 10 | 0.024150857 | −0.073811184 | 0.048718148 |
| 11 | −0.029826596 | 0.104427344 | −0.273225556 |
| 12 | **0.530853587** | 0.183283386 | 0.115779429 |
| 13 | **0.536912381** | 0.21031385 | 0.102165359 |
| 14 | **0.52294893** | 0.184219641 | 0.074117185 |

Table 2: Table of the First Three PC Vector Values

The above manually chosen significant absolute values are typed in boldface, and their correspond-

ing emphasized attributes are shown in Table 3. This table also contains the emphasized values if we choose the mean of the sum of absolute values in the principal component vector as the criterion. For the first component, we can see that the first PC emphasizes values relevant to penalties, that

| **Method of Selection** | **1st PC** | **2nd PC** | **3rd PC** |
|---|---|---|---|
| Manual | IsPenalty<br>IsPenaltyAccepted<br>IsNoPlay | IsRush<br>IsPass<br>IsIncomplete | Yards<br>IsSack |
| Mean of Absolute Values of the PC Vectors | IsRush<br>IsPass<br>IsIncomplete<br>IsPenalty<br>IsPenaltyAccepted<br>IsNoPlay | IsRush<br>IsPass<br>IsIncomplete | Yards<br>IsRush<br>IsIncomplete<br>IsTouchdown<br>IsSack<br>IsFumble |

Table 3: Table of the Emphases of First Three PC Vector Values

the second PC emphasizes values relevant to play type and pass result (data related to passing will not be discussed in this report since they have already been analyzed by media), and that the third PC emphasizes yards per play and sack. This gives us some clue on analyzing the play-by-play data. The code of selecting values above the mean of absolute values of the PC vectors is shown below.

```
FirstEmpAttr = calc_attr(PrincipalComponent1, AttrToProj)
SecondEmpAttr = calc_attr(PrincipalComponent2, AttrToProj)
ThirdEmpAttr = calc_attr(PrincipalComponent3, AttrToProj)

% calc_attr function implementation
function EmphasizedAttrs = calc_attr(PC, Attr)
    PC = abs(PC);
    divVal = mean(PC);
    EmphasizedAttrs = Attr(PC > divVal);
end
```

## 5.2 Penalty

It is natural to take a look into the the emphasized category in the first PC: *penalty*. We first need to get the plays with *offensive penalties*, which is accomplished by the following code:

```
% Filter Data
PenaltyIndex = find(OffRaw(:, Attr == 'PenaltyTeam') == 'GB');
PenaltyRaw = OffRaw(PenaltyIndex, :);
PenaltyYards = double(PenaltyRaw(:, Attr == 'PenaltyYards'));

% Find out the Denver Game Index
DenverGameIndex = find(PenaltyRaw(:, DefCol) == 'DEN', 1);
penLength = length(PenaltyYards);

gameLength = length(Games);
preDenLength = find(Games == 'DEN') - 1;
postDenLength = gameLength - preDenLength;
```

### 5.2.1 Number of Penalties

The number of penalties will definitely set the offense back a lot and it can "change the momentum". Then we need to find out whether our data suggests that there is a significant change in the number of penalties per game, which is accomplished by the code below.

```
PenaltyPreAvg = length(1 : DenverGameIndex - 1) / preDenLength
PenaltyPostAvg = length(DenverGameIndex : penLength) / postDenLength
```

It turns out that the number of offensive penalties per game is very consistent: 5 before the Denver game and 4.6 after the Denver game.

### 5.2.2 Penalties Yardage

Since the number of offensive penalties is very consistent, we need to take a look at the penalty yards, in that a 5-yard penalty is much easier to overcome than a serious penalties. The resulting plot is shown in Figure 5. The code is shown on the next page.

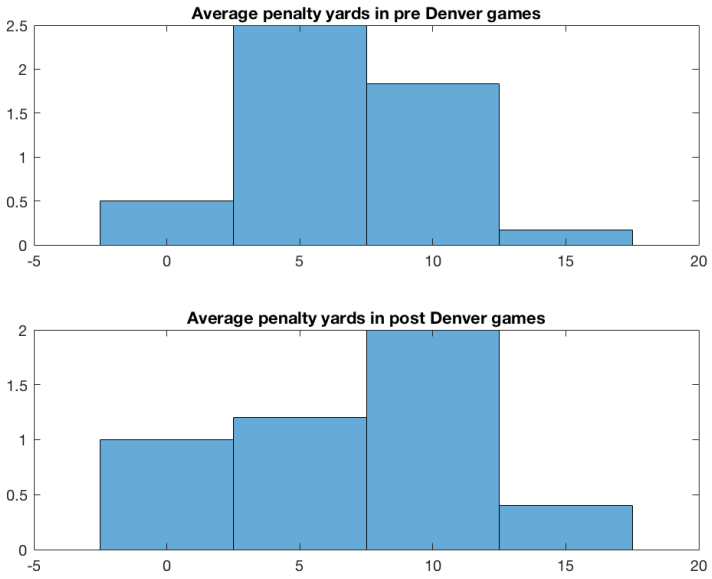



Figure 5: Average Penalty yards per play histograms.

```
% Penalty Yards Pre and Post Denver Game
PenaltyYardsPre = PenaltyYards(1 : DenverGameIndex - 1, :);
PenaltyYardsPost = PenaltyYards(DenverGameIndex : penLength, :);

% Average Penalty Yards Per Game
numberOfBins = 5;
df = numberOfBins - 1;
Edges = linspace(-2.5, 17.5, numberOfBins);
[prePenCounts, Edges] = histcounts(PenaltyYardsPre, Edges);
[postPenCounts, Edges] = histcounts(PenaltyYardsPost, Edges);

% Get the average
prePenCounts = prePenCounts / preDenLength;
postPenCounts = postPenCounts / postDenLength;

figure
subplot(2, 1, 1)
histogram('BinEdges', Edges, 'BinCounts', prePenCounts)
title('Average penalty yards in pre Denver games');
subplot(2, 1, 2)
histogram('BinEdges', Edges, 'BinCounts', postPenCounts)
title('Average penalty yards in post Denver games');
```

From the plot we can see an increase in the number of serious penalties: penalties of 10 and 15 yards. To see the difference better, please take a look at Figure 6. Serious penalties are much harder to overcome. For example, if on 3$^{rd}$ down and 2, the Packers commit a 5-yard penalty, it will be 3$^{rd}$ down and 8; 8 yards are not hard for Aaron Rodgers and company to overcome. However, if it is a 10 or even 15 yard penalty, considering how slow Packers' receivers are last season, one can easily imagine how those penalties can greatly affect the Packers' offense.

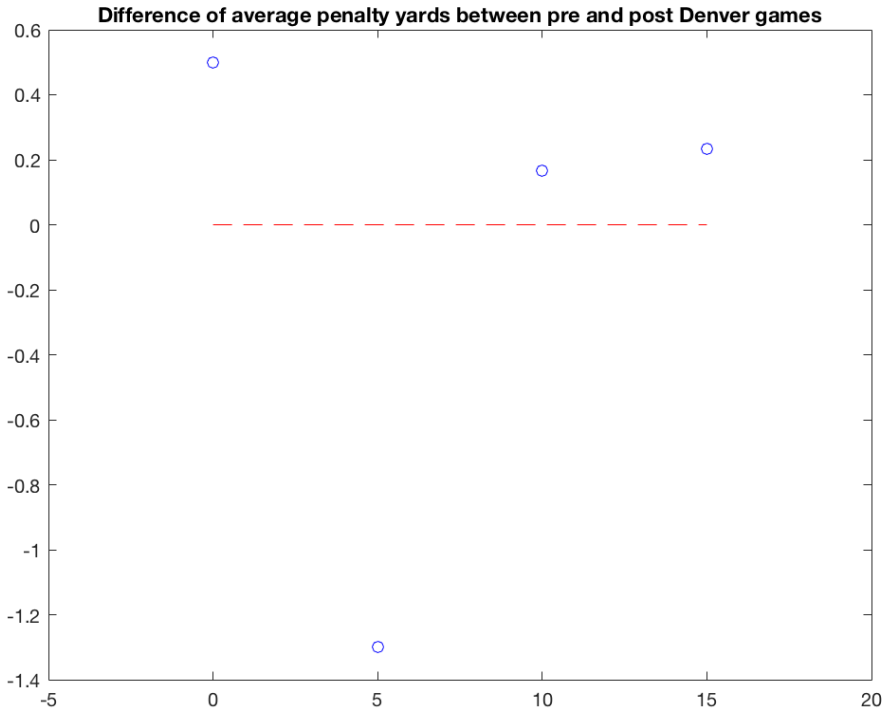



Figure 6: Difference between penalty yards between post and pre Denver game

The code for the difference is below:

```
% Difference
diffX = linspace(0, 15, df);
figure
plot(diffX, postPenCounts - prePenCounts, 'bo')
hold on
plot(diffX, zeros(df, 1), 'r--')
title('Difference of average penalty yards between pre and post Denver games');
axis([-5, 20, -1.4, 0.6]);
```

### 5.3 Yards Per Play

As the dominant value in the third PC, we have to look into the value: yards per play. First, we want to take a look at histograms of yards per play in a game. We split the data into two parts: pre-Denver games and post-Denver games. Since there are a lot more games after the Denver game, we need to use the average of yards per play.

```
OffYards = Off(:, Attr == 'Yards');
gameLength = length(Games);
preDenLength = find(Games == 'DEN') - 1;
postDenLength = gameLength - preDenLength;
OffPre = OffYards(1 : DenverGameIndex - 1, :);
OffPost = OffYards(DenverGameIndex : offLength, :);
Edges = linspace(-12, 66, 27);

% Use histcount function to process counts
[preCounts, Edges] = histcounts(OffPre, Edges);
[postCounts, Edges] = histcounts(OffPost, Edges);

% Get the average
preCountsAvg = preCounts / preDenLength;
postCountsAvg = postCounts / postDenLength;
```

From the above code, we get two histograms of yards per play in pre Denver games and post Denver games, shown in Figure 7. From the plot we can see that there are more negative plays, and more plays with little gain.

To confirm our feeling, we plotted the difference between the post and pre Denver games in Figure 8: `postCounts - preCounts`. From Figure 8, we can see that there is a sharp increase in plays with small gains and significant increase in negative plays. There are also noticeable decreases for plays beyond 10 yards.

We see that the offense is not as productive after the Denver game than before the Denver game.

```
% Difference
start = mean(Edges(1 : 2));
ending = mean(Edges(26 : 27));
diffX = linspace(start, ending, 26);
```

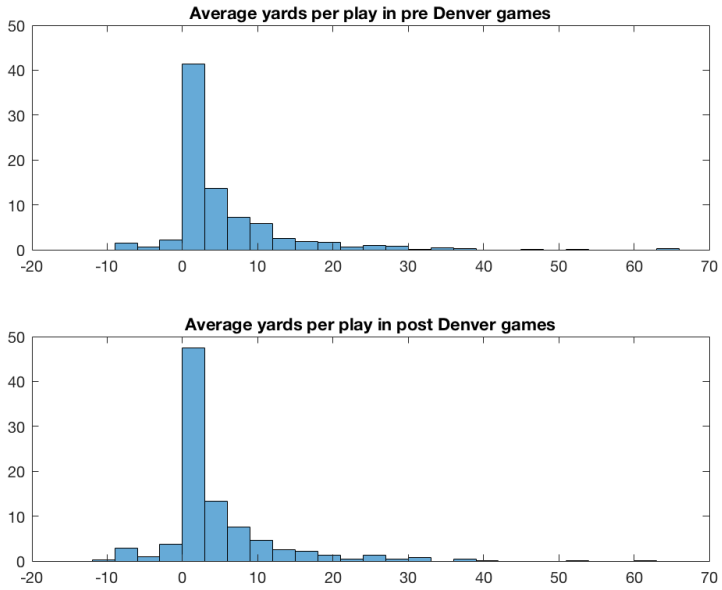



Figure 7: Average yards per play histograms.

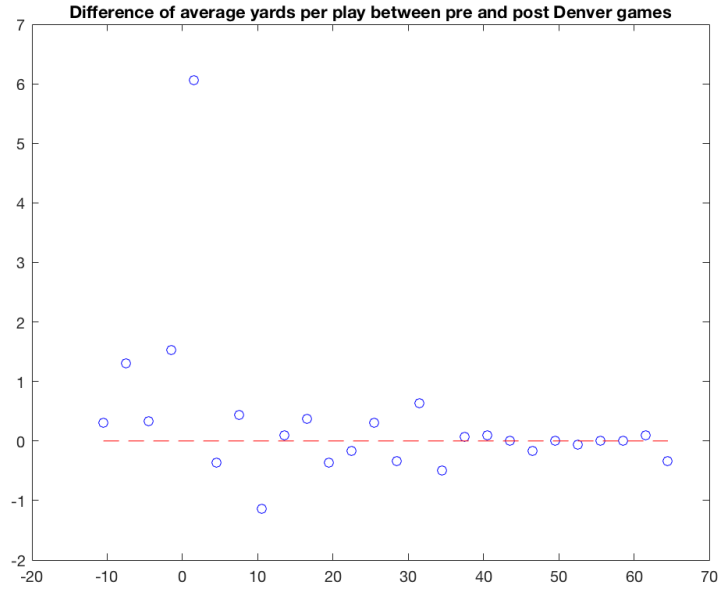



Figure 8: Difference between yards per play between post and pre Denver game

```
plot(diffX, postCounts - preCounts, 'bo')
hold on
plot(diffX, zeros(26, 1), 'r--')
title('Difference of average yards per play between pre and post Denver games');
```

### 5.3.1 Hypothesis Test ($t$ Test)

To see if there is enough evidence for worse yards per play in post Denver games, we want to setup a hypothesis test with significance level $\alpha = 0.05$. Let subscript 1 denote yards per play in pre Denver games and let subscript 2 denote yards per play in post Denver games.

$$H_0 : \mu_{\text{pre}} = \mu_{\text{post}}$$
$$H_a : \mu_{\text{pre}} > \mu_{\text{post}}$$

$$t = \frac{\overline{x_1} - \overline{x_2} - 0}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$$
$$p\text{-value} = \mathbf{P}(t > t_{df,\alpha}), \qquad df = \min(n_1, n_2) - 1,\ \alpha = 0.05$$

And the result is that we reject the null hypothesis, which means we have enough evidence that the offense before the Denver game is better than that after the Denver game. This is not surprising as we see Aaron Rodgers and company struggle on offense. Receivers are pressed at the line of scrimmage and they cannot gain separation from average-best corners, making Rodgers scramble for himself or dumping a short check-pass a running back. The data are consistent with our eye-test, but are there any other possible cause for the debacle of their offense? Can we find, if any, from the dataset?

Since we know and see that their passing game struggled, we would like to know what else contributed to the avalanche of their offense. First we want to see whether the running game has affected the outcome. There is no significant difference between the average rushing attempts per game, as the value before the Denver game is 28.3333 and that after the Denver game is 26.4000. A 2-attempt difference is completely unnoticeable in a game. Thus, we want to see whether the efficiency of rushing has declined. The average rushing yards per attempt is 4.7471 before the Denver game and 4.2197 after the Denver game. The average rushing yards per attempt dropped for about 0.53 yards, which is 11%.

The code for the hypothesis test is shown below:

```
% Hypothesis Test
[mu1, s1, n1] = calc_t_attr(OffPre);
[mu2, s2, n2] = calc_t_attr(OffPost);
df = min(n1, n2) - 1;

tValProb = cdf('T', (mu1 - mu2) / sqrt(s1^2 / n1 + s2^2 / n2), df, 'upper');
Significance = 0.05;

if tValProb < Significance
    fprintf('Reject Null Hypothesis.\n');
else
    fprintf('Do not reject Null Hypothesis.\n');
end
```

The `calc_t_attr` function implementation is shown below:

```
function [mu, s, n] = calc_t_attr(X)
    mu = mean(X); s = std(X); n = length(X);
end
```

### 5.3.2 $\chi^2$ Test

I originally planned to only use $\chi^2$ *Test*, but in some bins the number of occurrences is 0. If we simply plug these ill-formatted histogram counts into the `cdf` function, it will only return `NaN`. Also, since we are comparing the means, $t$ test (another form of hypothesis test) seems more natural. However, we can reduce the number of bins to avoid 0s in the histogram counts. Note: reducing the number of bins can lead to a poorly represented data. For yardages where a 5-yard play differs from a 10-yard play significantly, I do think it is better to use Hypothesis Test. The code is below:

```
% Chi-Square Test
numberOfBins = 5;
df = numberOfBins - 1;
Edges = linspace(-20, 80, numberOfBins);

% Use histcount function to process counts
[postCounts, Edges] = histcounts(OffPost, Edges);
[seasonCounts, Edges] = histcounts(OffYards, Edges);

% Get the average
postCounts = postCounts / preDenLength;
seasonCounts = seasonCounts / gameLength;

ChiSquareStatistic = sum((postCounts - seasonCounts) .^ 2 ./ seasonCounts)
ChiSquareProbability = cdf('Chisquare', ChiSquareStatistic, df)
```

The resulting $\chi^2$ statistic is really large: $\approx 46.5266$, and the corresponding $\chi^2$ probability is $\approx 1$, which means this situation is almost impossible. Thus, the $\chi^2$ Test agrees with our Hypothesis Test result: the offense performed worse in post-Denver games.

## 5.4 Rushing Yards Per Play

Like what we did in Section 5.3, we use histograms to find out what happened to the ground game. The following code plots histograms of average rushing yards per play from both pre and post Denver games, shown in Figure 9. From the graph, we actually see that there are more small yards

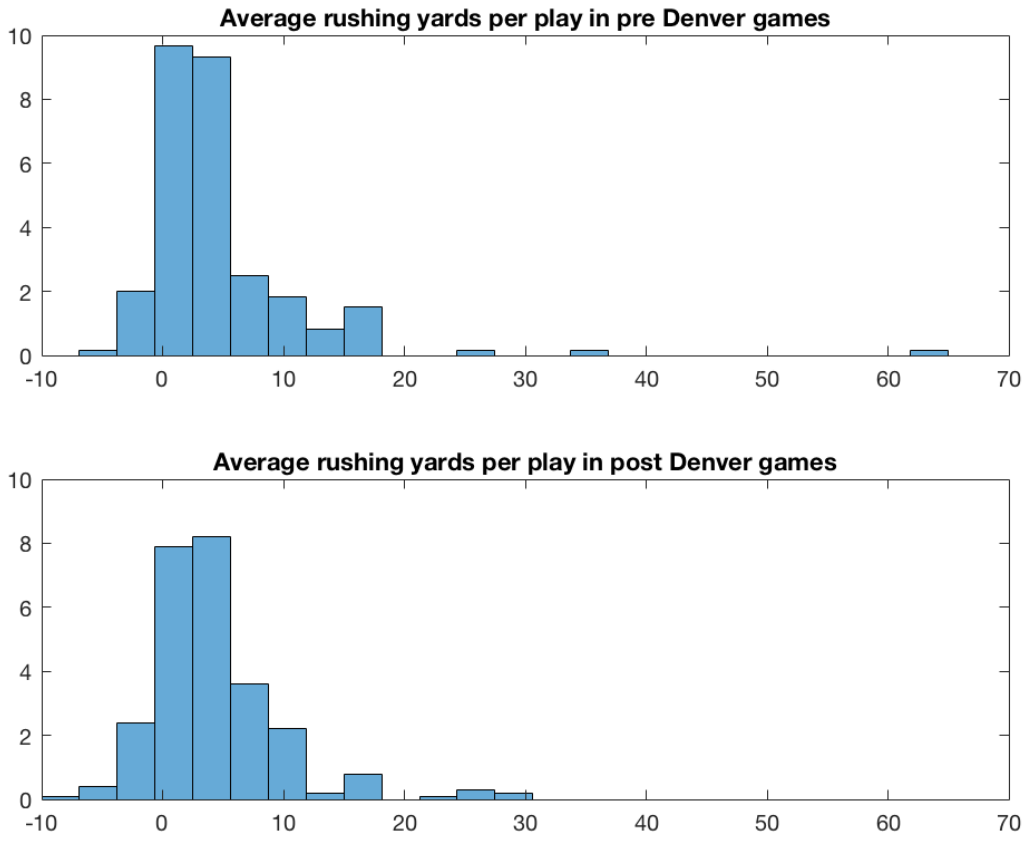



Figure 9: Average rushing yards per play histograms.

and less good gains before the Denver games. The average value before the Denver game might very well be boosted by some large yardage runs at the right hand side of the histogram. The plot of the difference in Figure 10. The code is shown below.

```
% Filter Rushing Data
OffYards = Off(:, Attr == 'Yards');
RushIndex = double(OffRaw(:, Attr == 'IsRush')) == 1;
RushRaw = OffRaw(RushIndex, :);
```

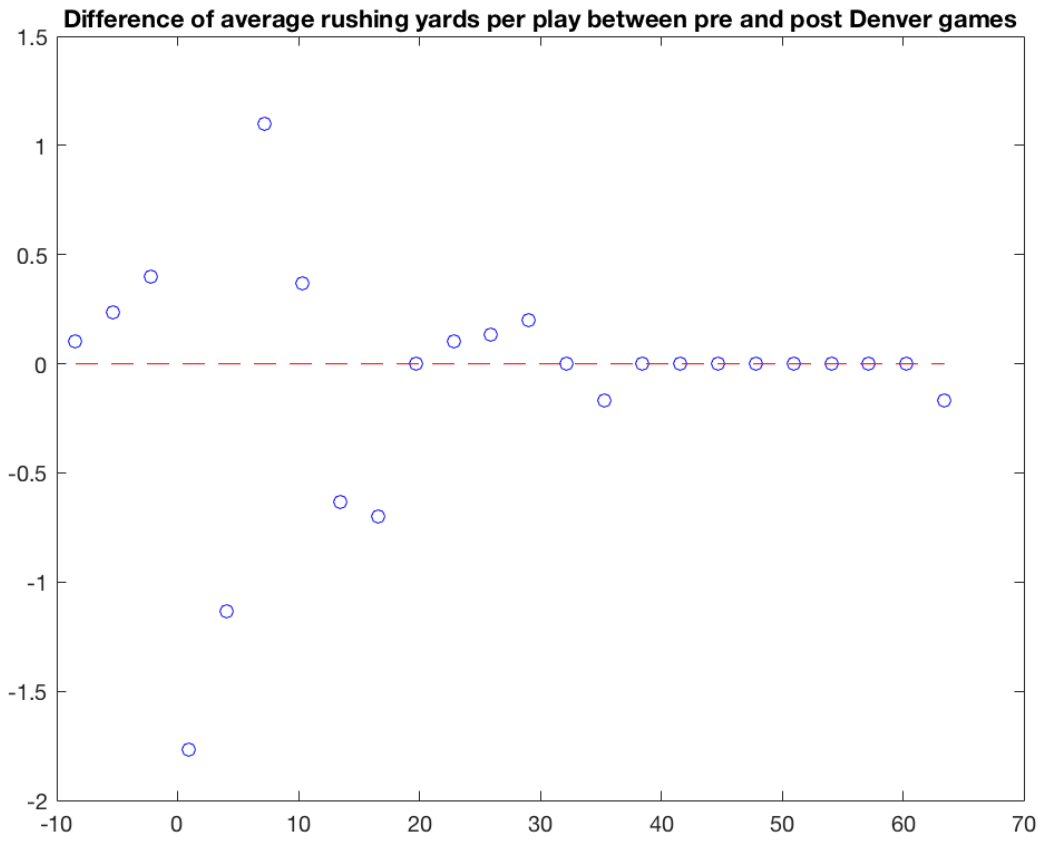



Figure 10: Difference between rushing yards per play between post, pre Denver game

```
RushLength = length(RushRaw);
RushYards = OffYards(RushIndex, :);
DenverGameIndex = find(RushRaw(:, DefCol) == 'DEN', 1);

% Find the first Denver game index
gameLength = length(Games);
preDenLength = find(Games == 'DEN') - 1;
postDenLength = gameLength - preDenLength;

% Divide Data into two parts
RushPre = RushYards(1 : DenverGameIndex - 1, :);
RushPost = RushYards(DenverGameIndex : RushLength, :);

Edges = linspace(-10, 65, 25);

```

```
% Use histcount function to process counts
[preCounts, Edges] = histcounts(RushPre, Edges);
[postCounts, Edges] = histcounts(RushPost, Edges);


% Get Average
preCountsAvg = preCounts / preDenLength;
postCountsAvg = postCounts / postDenLength;

figure
subplot(2, 1, 1)
histogram('BinEdges', Edges, 'BinCounts', preCountsAvg)
title('Average rushing yards per play in pre Denver games');
subplot(2, 1, 2)
histogram('BinEdges', Edges, 'BinCounts', postCountsAvg)
title('Average rushing yards per play in post Denver games');

% Difference
start = mean(Edges(1 : 2));
ending = mean(Edges(24 : 25));
diffX = linspace(start, ending, 24);
figure
plot(diffX, postCountsAvg - preCountsAvg, 'bo')
hold on
plot(diffX, zeros(24, 1), 'r--')
title('Difference of average rushing yards per play between pre, post Denver games');
```

### 5.4.1 Hypothesis Test ($t$ Test)

To see if there is enough evidence for worse rushing attacks in post Denver games, we want to setup a hypothesis test with significance level $\alpha = 0.05$. Let subscript 1 denote rush yards in pre Denver games and let subscript 2 denote rush yards in post Denver games.

$$H_0 : \mu_{\text{pre}} = \mu_{\text{post}}$$
$$H_a : \mu_{\text{pre}} > \mu_{\text{post}}$$

$$t = \frac{\overline{x_1} - \overline{x_2} - 0}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$$
$$p\text{-value} = \mathbf{P}(t > t_{df,\alpha}), \qquad df = \min(n_1, n_2) - 1,\ \alpha = 0.05$$

Thus we use the following code to do the test:

```
% Hypothesis Test
[mu1, s1, n1] = calc_t_attr(RushPre);
[mu2, s2, n2] = calc_t_attr(RushPost);
df = min(n1, n2) - 1;

tValProb = cdf('T', (mu1 - mu2) / sqrt(s1^2 / n1 + s2^2 / n2), df, 'upper');
Significance = 0.05;

if tValProb < Significance
    fprintf('Reject Null Hypothesis.\n');
else
    fprintf('Do not reject Null Hypothesis.\n');
end
```

And the result is that we do not reject the null hypothesis, which means we do not have enough evidence to show that the ground game before the Denver game is better than that after the Denver game. Thus we can put much less faith on that "improving the ground game will boost the offense".

## 5.5 Sack

Here we will look into another dominant value in the third PC: *Sack*. Sack is somewhat represented in negative plays, but in critical game situations, a sack can be a game changer. Like what we did in Section 5.2, we filter the data first and calculate the number of sacks per game.

```
% Filter out sack data
SackIndex = find(OffRaw(:, Attr == 'IsSack') == '1');
SackRaw = OffRaw(SackIndex, :);
SackYards = OffYards(SackIndex);
DenverGameIndex = find(SackRaw(:, DefCol) == 'DEN', 1);
sackLength = length(SackYards);

gameLength = length(Games);
preDenLength = find(Games == 'DEN') - 1;
postDenLength = gameLength - preDenLength;

% The mean of the number of sacks in a game
SackPreAvg = length(2 : DenverGameIndex) / preDenLength
SackPostAvg = length(DenverGameIndex : sackLength) / postDenLength
```

We see that before the Denver game, the average number of sacks is only 2.1667, while after the Denver game, this number bumps $\approx 75.38\%$ to 3.8. An increase in sacks can indicate the following:

- Lack of ground game caused too many passing attempts, which naturally generated more sacks. This is already declined in Section 5.4.1.
- Lack of protection from offensive line. The Packers' offensive line was plagued by injuries in the second half of the season, with which the data are consistent.
- Aaron Rodgers held the ball too long. The protection will always break down as time-after-snap increases, which will result in more sacks. NFL Analysts have already found the cause: receives can't get open.

Since there is nothing novel coming out of the *sack* data, I will not do more data analysis on this part.

# 6 Conclusion

Because of my familiarity with American football and NFL, I decided to analyze NFL play-by-play data. In my humble opinion, except *ProFootballFocus*, no other media really pay attention to data mining, and analysts either only refer to simple statistics or rely solely on their "eye-test". For example, for Green Bay Packers' struggle in the second half of last season, I heard the following rhetoric for quite a while:

1. "Aaron Rodgers is not the same."
2. "There is no running game."
3. "There is no protection from the offensive line."
4. "The receivers can't get open."

For 1. and 2., often analysts do not refer to statistics at all. They use their *eye-test* to say what they feel, which is not very reliable.

In my analysis, for example, I found that there is no sufficient evidence that the ground game was worse in the second half of last season. NFL Analysts also did not notice an increase in serious penalties in the second half of the season, as shown in Section section::penalty, which created difficult tasks for the Packers' offense.

In this project, converting csv files / cells into *string matrices* costs me some time. I looked through several MATLAB documents to properly import the play-by-play data [3] into MATLAB. After my Principal Component Analysis, I spent quite some time getting histograms work. Since I need to use the average of yards per play (YPP), or YPP per game, I need to modify the counts in the histogram counts. In this course, we only saw `hist`, and I could not find a way to manipulate this function to get the average of counts. I looked several documents and decided to use a combination of `histcounts` and `histogram`. Then I used $\chi^2$ Test and Hypothesis Test (mainly the latter) to confirm my intuitions and found out that one of them is false. The hardest part of this project will be converting cells / char vectors into string matrices, and then manipulating and extracting data from the raw data string matrix. For the report, the hardest part is to find a proper arrangement so that the graphs and relevant paragraphs are not very far apart.

In conclusion, I found out that

- The ground game was not worse in the second half of the season.
- Packers' had noticeably more "serious" penalties, penalties with 10 or 15 yards.
- The number of sacks go up quite a bit, consistent with Analysts' "eye-test".

Although the dataset used in this project is already very large, it also lacks some attributes required by more advanced analyses. For example, if the receivers really cannot get open in press coverage, can we confirm that from the data; e.g. is the yards per play / completion percentage significantly down from non-press coverages? If I have this kinds of data, I would assign numeric labels to replace their string representations and use correlation matrix to do PCA again to confirm whether they play a huge role in offense.

## References


[1] Demovsky, R. Passing problem: Aaron rodgers, packers have nfl's slowest weapons. `http://www.espn.com/`, November 2015.

[2] Horowitz, M. Detailed nfl play-by-play data 2015. `https://www.kaggle.com`, October 2016.

[3] NFL-Savant. Nfl play-by-play data 2015. `http://nflsavant.com/about.php`, 2016.