Project 1

Matthew Graeff mkg2223

NFL Data Analytics - 4th Down Conversion Rate

Introduction

The datasets I have chosen provide data on every passing play in the 2018 regular season of the NFL. The data can be found on Kaggle (https://www.kaggle.com/code/aryashah2k/sports-analytics-visualization/notebook (https://www.kaggle.com/code/aryashah2k/sports-analytics-visualization/notebook)). I chose to use the datasets games.csv and plays.csv.

- Games.csv has data on the week, date, start time, home and away team, and a unique game ID.
- Plays.csv has data on the play such as the game ID, week, quarter, down, game clock, yards to go, yards earned, possession team, and more.

I am interested in this data because I wanted to see for myself how successful teams are at getting a conversion on 4th down because that has become a bigger talking point in NFL media in the last year. *A conversion just means that the team either got the yards they needed in order to gain a new set of downs (attempts) or they scored; if a team fails to convert, then they surrender the ball to the other team.* In this data, field goals are ignored, so scoring is limited to touchdowns. Additionally, I originally thought this data had records for running plays, too, but it is strictly passing plays. Thus, this analysis is incomplete for my original interest in testing the success of any 4th down attempt, but it is still useful for understanding the success of passing on 4th down.

Load Data

This code chunk loads the two datasets in and also loads the libraries needed.

```
# load games dataset from local computer
games <- read.csv("C:/SDS 322E Data Science/Project 1/games.csv")
# load plays dataset from local computer
plays <- read.csv("C:/SDS 322E Data Science/Project 1/plays.csv")
# load libraries
library(tidyverse)
library(ggplot2)
library(RColorBrewer)
```

These datasets are already tidy, and any kind of pivoting would make the data less tidy. So, I will proceed to join them without pivoting them.

Join Data

Games has 253 records and 6 columns. Plays has 19,239 records and 27 columns. Joining the two on game ID yields a dataset called games_plays that has 19,239 records and 32 columns. Joining them in this way added the 5 non-ID columns of games to every record in plays.

```
# count rows, columns of games
dim(games)
```

```
## [1] 253   6
```

```
# count rows, columns of plays
dim(plays)
```

```
## [1] 19239    27
```

```
# join games to plays on gameId
games_plays <- left_join(plays, games, by = "gameId")
# count rows, columns of games_plays
dim(games_plays)
```

```
## [1] 19239    32
```

Wrangling

a.) Exploring dataset

First, I will just preview the first 6 records in games_plays to get a basic idea of the variables. Then, since this dataset only has 253 games records and I know that there were 256 games in the 2018 season, I will investigate which weeks the 3 missing games are from.

```
# look at first 6 records in data
head(games_plays)
```

```
##       gameId playId
## 1 2018090600     75
## 2 2018090600    146
## 3 2018090600    168
##                                                                             playDescription
## 1 (15:00) M.Ryan pass short right to J.Jones pushed ob at ATL 30 for 10 yards (M.Jenkins).
## 2               (13:10) M.Ryan pass incomplete short right to C.Ridley (J.Mills, J.Hicks).
## 3                        (13:05) (Shotgun) M.Ryan pass incomplete short left to D.Freeman.
##   quarter down yardsToGo possessionTeam       playType yardlineSide
## 1       1    1        15            ATL play_type_pass          ATL
## 2       1    1        10            ATL play_type_pass          PHI
## 3       1    2        10            ATL play_type_pass          PHI
##   yardlineNumber offenseFormation       personnelO defendersInTheBox
## 1             20           I_FORM 2 RB, 1 TE, 2 WR                 7
## 2             39       SINGLEBACK 1 RB, 1 TE, 3 WR                 7
## 3             39          SHOTGUN 2 RB, 1 TE, 2 WR                 6
##   numberOfPassRushers       personnelD typeDropback preSnapVisitorScore
## 1                   4 4 DL, 2 LB, 5 DB  TRADITIONAL                   0
## 2                   4 4 DL, 2 LB, 5 DB  TRADITIONAL                   0
## 3                   4 4 DL, 2 LB, 5 DB  TRADITIONAL                   0
##   preSnapHomeScore gameClock absoluteYardlineNumber penaltyCodes
## 1                0  15:00:00                     90
## 2                0  13:10:00                     49
## 3                0  13:05:00                     49
##   penaltyJerseyNumbers passResult offensePlayResult playResult        epa
## 1                               C                10         10  0.2618273
## 2                               I                 0          0 -0.3723598
## 3                               I                 0          0 -0.7027787
##   isDefensivePI   gameDate gameTimeEastern homeTeamAbbr visitorTeamAbbr week
## 1         FALSE 09/06/2018        20:20:00          PHI             ATL    1
## 2         FALSE 09/06/2018        20:20:00          PHI             ATL    1
## 3         FALSE 09/06/2018        20:20:00          PHI             ATL    1
##  [ reached 'max' / getOption("max.print") -- omitted 3 rows ]
```

```
# count number of games/week
games_plays %>%
  group_by(week) %>%
  summarize(number_games = n_distinct(gameId))
```

```
## # A tibble: 17 x 2
##     week number_games
##    <int>        <int>
##  1     1           13
##  2     2           16
##  3     3           16
##  4     4           15
##  5     5           15
##  6     6           15
##  7     7           14
##  8     8           14
##  9     9           13
## 10    10           14
## 11    11           13
## 12    12           15
## 13    13           16
## 14    14           16
## 15    15           16
## 16    16           16
## 17    17           16
```

First, the major variables that look useful are quarter, down, yardsToGo, possessionTeam, preSnapVisitorScore, preSnapHomeScore, gameClock, passResult, offensePlayResult, homeTeam, and awayTeam. playDescription is useful for understanding what happened in the play, but it will not be used for any of the data analysis. I will ignore penalties and playResults that are greater than offensePlayResult because I do not want to skew my analysis with penalties that are subjectively called and vary from year to year.

Second, it seems like the 3 missing games are from week 1. I do not know why these games are missing, but they are insignificant in the aggregate.

b.) Dropping 11 columns and adding 2 new columns

In this code chunk I select only the 21 columns that seem interesting and useful to me (aka dropping 11 columns), and I create 2 new columns for a total of 23 columns. The first new column is a Boolean that shows if the home team has possession, and the second is a Boolean that shows if the play resulted in a conversion (1st down or touchdown).

```
# select 21 important columns, create 2 new columns, save new dataframe
games_plays <- games_plays %>%
  select(gameId, playId, playDescription, quarter, down, yardsToGo, possessionTeam, offenseForma
tion, personnelO, defendersInTheBox, numberOfPassRushers, personnelD, typeDropback, preSnapVisit
orScore, preSnapHomeScore, gameClock, passResult, offensePlayResult, homeTeamAbbr, visitorTeamAb
br, week) %>%
  mutate(homeTeamPossession = ifelse(possessionTeam == homeTeamAbbr, TRUE, FALSE),
         converted = ifelse(offensePlayResult >= yardsToGo, TRUE, FALSE))
```

c.) Exploring the data part 2

In this code chunk, I look at the mean value for every column all at once, which allows me to get a quick overview of the numerical data and see what stands out to me.

```
# summarize all columns by mean
games_plays %>%
  summarize_all(mean, na.rm = T)
```

```
##       gameId   playId playDescription  quarter     down yardsToGo
## 1 2018107423 2176.217             NA 2.591039 1.920318  8.923905
##   possessionTeam offenseFormation personnelO defendersInTheBox
## 1             NA               NA         NA          6.036189
##   numberOfPassRushers personnelD typeDropback preSnapVisitorScore
## 1            4.216543         NA           NA            10.75833
##   preSnapHomeScore gameClock passResult offensePlayResult homeTeamAbbr
## 1         12.46548        NA         NA           6.27938           NA
##   visitorTeamAbbr     week homeTeamPossession converted
## 1              NA 8.944592           0.499922 0.3262124
```

A few things that stand out to me:

- *More pass plays happen after halftime than before halftime. This could be because overtime is listed as 5th quarter.*
- *Most passing plays happen in the earlier downs by a small margin*
- *The average yards earned is about 2.6 yards less than the average yards to go*
- *The home team's average pre-snap score is 1.7 points higher than the away team's*
- *The home team's number of plays is exactly split with the away team's*
- *Teams convert on just under a third of all plays. Since a team has 4 downs to try to convert, most of the time they convert before 4th down. But what about their conversion rate on 4th down?*

d.) 3 Summary statistics

I want to dive further into the average yards to go vs average yards gained and the average conversion rate. This code chunk displays 3 summary statistics - 2 for numerical data and 1 for categorical (Boolean).

```
# group by down, show mean of 3 columns
games_plays %>%
  group_by(down) %>%
  summarize(avg_yards_to_go = mean(yardsToGo),
            avg_yards_gained = mean(offensePlayResult),
            avg_conversion_rate = mean(converted),
            occurences = n())
```

```
## # A tibble: 4 x 5
##    down avg_yards_to_go avg_yards_gained avg_conversion_rate occurences
##   <int>           <dbl>            <dbl>               <dbl>      <int>
## 1     1           10.2              6.90               0.288       7405
## 2     2            8.58             6.46               0.345       6315
## 3     3            7.76             5.23               0.349       5166
## 4     4            5.39             5.36               0.453        353
```

Even though a team always starts 1st down with 10 yards to go, the average yards to go is 10.2 because of penalties. The interesting thing about the average yards to go vs average yards gained to me is that they are nearly identical in 4th down. And yet, the average 4th down conversion rate is only 45%. Why? What situations are more likely to result in a 4th down conversion?

e.) In which situations is a 4th down conversion feasible?

In this code chunk I group by 5 different factors one at a time to explore which factors are more correlated with a higher 4th down conversion rate.

```
# find mean 4th down conversion rate based on different groupings

# yards to go
games_plays %>%
  filter(down == 4) %>%
  group_by(yardsToGo) %>%
  summarize(avg_conversion_rate = mean(converted),
            count = n()) %>%
  arrange(desc(avg_conversion_rate))
```

```
## # A tibble: 21 x 3
##    yardsToGo avg_conversion_rate count
##        <int>               <dbl> <int>
##  1         7               0.684    19
##  2         8               0.667     9
##  3         1               0.621    66
##  4         2               0.569    51
##  5         6               0.52     25
##  6         3               0.489    47
##  7        15               0.429     7
##  8         4               0.382    34
##  9        11               0.286     7
## 10        12               0.286     7
## 11        10               0.25     24
## 12         5               0.24     25
## 13        14               0.2       5
## 14        13               0.167     6
## 15         9               0.111     9
## 16        16               0         2
## 17        18               0         2
## 18        19               0         2
## 19        20               0         1
## 20        22               0         4
## 21        28               0         1
```

```
# quarter
games_plays %>%
  filter(down == 4) %>%
  group_by(quarter) %>%
  summarize(avg_conversion_rate = mean(converted),
            count = n()) %>%
  arrange(desc(avg_conversion_rate))
```

```
## # A tibble: 5 x 3
##   quarter avg_conversion_rate count
##     <int>               <dbl> <int>
## 1       2               0.615    52
## 2       1               0.591    22
## 3       3               0.549    51
## 4       4               0.382   220
## 5       5               0.375     8
```

```
# typeDropback
games_plays %>%
  filter(down == 4) %>%
  group_by(typeDropback) %>%
  summarize(avg_conversion_rate = mean(converted),
            count = n()) %>%
  arrange(desc(avg_conversion_rate))
```

```
## # A tibble: 8 x 3
##   typeDropback             avg_conversion_rate count
##   <chr>                                  <dbl> <int>
## 1 "DESIGNED_ROLLOUT_LEFT"               1          2
## 2 "UNKNOWN"                             0.733     15
## 3 "SCRAMBLE_ROLLOUT_RIGHT"              0.5       24
## 4 "TRADITIONAL"                         0.479    263
## 5 "DESIGNED_ROLLOUT_RIGHT"              0.4        5
## 6 "SCRAMBLE_ROLLOUT_LEFT"               0.211     19
## 7 "SCRAMBLE"                            0.182     11
## 8 ""                                    0.0714    14
```

```
# offensive formation
games_plays %>%
  filter(down == 4) %>%
  group_by(offenseFormation) %>%
  summarize(avg_conversion_rate = mean(converted),
            count = n()) %>%
  filter(count > 1) %>%
  arrange(desc(avg_conversion_rate))
```

```
## # A tibble: 7 x 3
##   offenseFormation avg_conversion_rate count
##   <chr>                          <dbl> <int>
## 1 "SINGLEBACK"                   0.727    22
## 2 ""                             0.688    16
## 3 "PISTOL"                       0.667     3
## 4 "I_FORM"                       0.625     8
## 5 "JUMBO"                        0.6       5
## 6 "SHOTGUN"                      0.431   255
## 7 "EMPTY"                        0.279    43
```

```
# offensive personnel
games_plays %>%
  filter(down == 4) %>%
  group_by(personnelO) %>%
  summarize(avg_conversion_rate = mean(converted),
            count = n()) %>%
  filter(count > 1) %>%
  arrange(desc(avg_conversion_rate))
```

```
## # A tibble: 10 x 3
##    personnelO                                  avg_conversion_rate count
##    <chr>                                                     <dbl> <int>
##  1 2 RB, 2 TE, 1 WR                                          1         2
##  2 1 RB, 2 TE, 2 WR                                          0.75     20
##  3 6 OL, 2 RB, 2 TE, 0 WR                                    0.667     3
##  4 0 RB, 1 TE, 1 WR,1 P,3 LB,1 LS,1 DL,3 DB                  0.5       2
##  5 1 RB, 0 TE, 4 WR                                          0.5      10
##  6 2 RB, 0 TE, 3 WR                                          0.5       2
##  7 2 RB, 1 TE, 2 WR                                          0.5       4
##  8 1 RB, 1 TE, 3 WR                                          0.416   279
##  9 1 RB, 3 TE, 1 WR                                          0.4      10
## 10 0 RB, 1 TE, 4 WR                                          0         5
```

Key takeaways:

- ***Group by Yards to Go:*** *As expected, most 4th down attempts occur when there are only 1 or 2 yards to go, but interestingly, attempts with 7 or 8 yards to go have the highest conversion rate. 7, 8, 1, 2, and 6 yards to go are all converted more than 50% of the time.*

- ***Group by Quarter:*** *Also as expected, most 4th down attempts occur in the 4th quarter when teams care slightly more about running out of time than about giving up good field position to the other team. However, conversion rates drop dramatically in the 4th quarter. This could just be a correlation between the less successful teams taking more risks later in the game vs more successful teams being aggressive and dominant early. It does not necessarily prove that there’s something about the 4th quarter that hinders a team’s ability to convert on 4th down.*

- ***Group by Type of Dropback:*** *It’s amazing to me that only two plays were called with a designed quarterback rollout to the left since they were both successful. I wonder if quarterbacks are secretly better at left rollouts than we thought, or if maybe the rarity of a left rollout surprised the defense and that is the* *reason for the 100% conversion rate. The element of surprise seems to be a factor because the unknown dropback was successful 73% of 15 attempts. If it was called unknown, it was probably because it was an unconventional dropback that defenses have not prepared for.*

- ***Group by Offensive Formation:*** *This grouping shows me again that offenses really need to do something unconventional on 4th down to be successful because the more common formations appear to have the worse conversion rates.*
- ***Group by Offensive Personnel:*** *At this point, I am beginning to wonder if offensive coordinators ever look at the stats because I cannot understand why they called a “1 RB, 1 TE, 3 WR” personnel so many times when just about every other personnel was more successful.*

f.) Double grouping

Next, I look at grouping by yards to go and quarter to see if getting more detailed can paint a better picture of what is going on during 4th down attempts.

```
# quarter + yards
games_plays %>%
  filter(down == 4) %>%
  group_by(quarter, yardsToGo) %>%
  summarize(avg_conversion_rate = mean(converted),
            count = n()) %>%
  arrange(desc(avg_conversion_rate))
```

```
## # A tibble: 51 x 4
## # Groups:   quarter [5]
##    quarter yardsToGo avg_conversion_rate count
##      <int>     <int>               <dbl> <int>
##  1       1         4               1         1
##  2       1         6               1         2
##  3       3         7               1         1
##  4       3        11               1         1
##  5       5         2               1         1
##  6       5         7               1         1
##  7       5        15               1         1
##  8       2         7               0.875     8
##  9       1         2               0.833     6
## 10       4         8               0.75      8
## 11       3         2               0.692    13
## 12       3         1               0.667    15
## 13       4         1               0.64     25
## 14       3         3               0.625     8
## 15       2         1               0.611    18
## 16       2         3               0.6      10
## 17       2         4               0.6       5
## 18       1         1               0.5       8
## 19       1         5               0.5       2
## 20       2         2               0.5       8
## 21       2         5               0.5       2
## 22       3         4               0.5       4
## 23       4         6               0.478    23
## 24       4         3               0.462    26
## 25       4         7               0.444     9
## 26       4         2               0.435    23
## 27       4        15               0.333     6
## 28       4         4               0.318    22
## 29       4        12               0.286     7
## 30       4        10               0.273    22
## 31       4         5               0.267    15
## 32       4        13               0.2       5
## 33       4        14               0.2       5
## 34       4        11               0.167     6
## 35       4         9               0.143     7
## 36       1         3               0         3
## 37       2         9               0         1
## 38       3         5               0         5
## 39       3         9               0         1
## 40       3        10               0         2
## 41       3        13               0         1
## 42       4        16               0         2
## 43       4        18               0         2
## 44       4        19               0         2
## 45       4        20               0         1
## 46       4        22               0         3
## 47       4        28               0         1
## 48       5         4               0         2
## 49       5         5               0         1
## 50       5         8               0         1
## 51       5        22               0         1
```

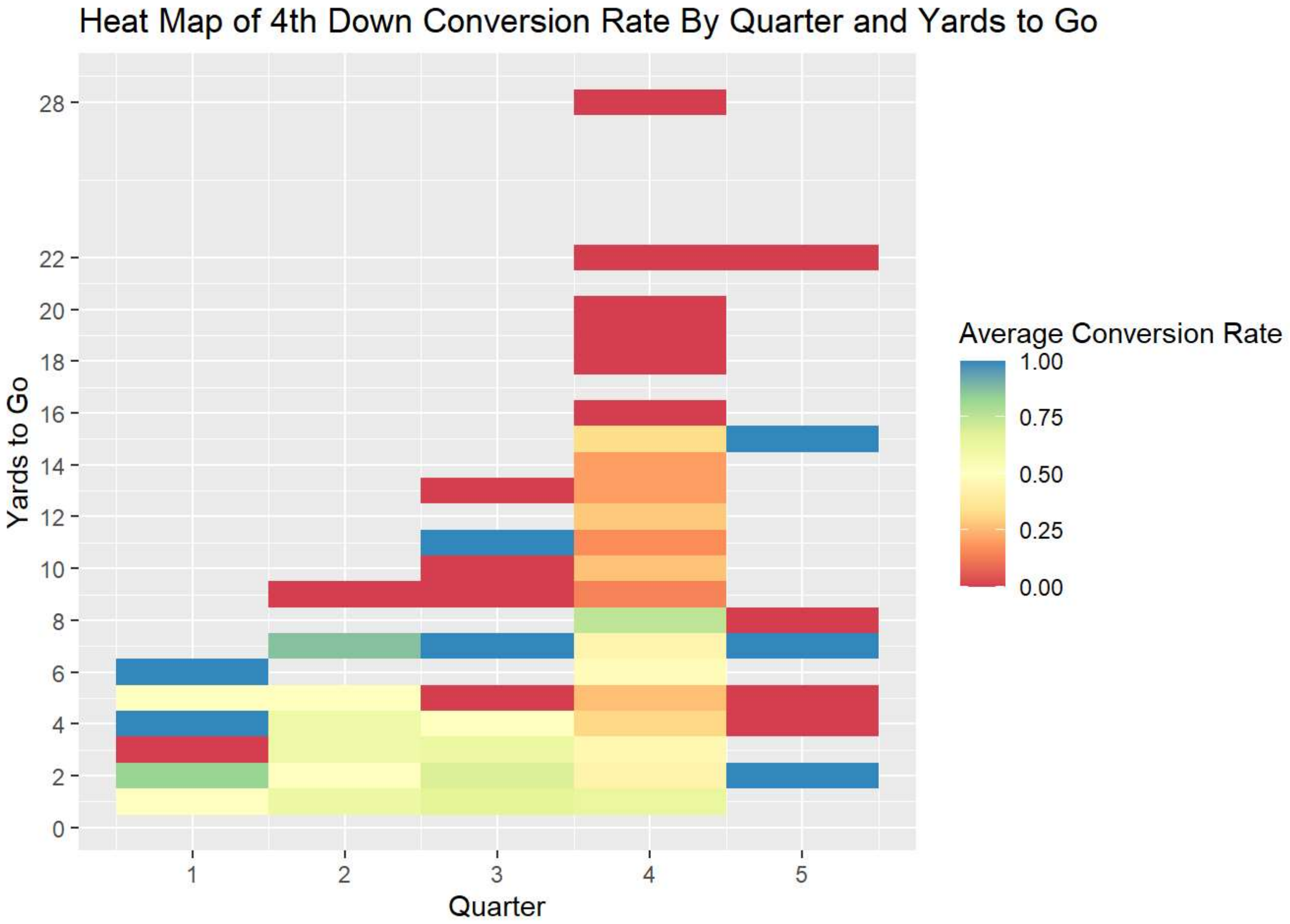
This data looks really interesting. Already I can see that the most successful conversion attempts are not the ones attempted most often. However, with 51 rows of data, it’s hard to really understand it all just by looking at the numbers, so I will plot it in the next section.

Visualizations

a.) Heat map: quarter + yards to go

This visualization is a heat map that shows the 4th down conversion rate as a color based on two numerical factors: quarter on the x axis and yards to go as the y axis. The more blue the tile is, the better the conversion rate, and the more red the tile is, the worse the conversion rate.

```
# visualization of quarter + yards
games_plays %>%
  filter(down == 4) %>%
  group_by(quarter, yardsToGo) %>%
  summarize(avg_conversion_rate = mean(converted),
            count = n()) %>%
  arrange(desc(avg_conversion_rate)) %>%
  ggplot(aes(x = quarter, y = yardsToGo, fill = avg_conversion_rate)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral", direction = 1, name = "Average Conversion Rate") +
  scale_y_continuous(breaks = c(0,2,4,6,8,10,12,14,16,18,20,22,28)) +
  labs(title = "Heat Map of 4th Down Conversion Rate By Quarter and Yards to Go")+
  xlab("Quarter") +
  ylab("Yards to Go")
```



Key Takeaways:

- *No matter what quarter it is, going for it with only 1 or 2 yards to go is always about 50/50 (except 4th and 2 in the 1st quarter, which has greater odds).*
- *In the 1st quarter, every attempt except 3 yards to go is more likely to succeed than fail.*
- *In the 2nd quarter, every attempt except 9 yards to go is not likely to fail.*
- *In the 3rd quarter, attempts with 1-3 yards to go are likely to be successful.*
- *The 4th quarter has the worst average conversion rate.*
- *There does not seem to be enough 4th down attempts in overtime to make any definitive conclusions.*

So why does the average conversion rate decrease as the game goes on? One possible reason is that the defense has figured out what kind of plays the offense likes and how to stop them. Another possible reason is that defenses are caught off guard by 4th down attempts early in the game because they are so rare. However, I think the biggest reason for this trend is simply that the teams that are less likely to perform well and successfully convert on 4th downs are the teams that are losing by the end of the game and the ones that are getting more desperate to make a big play, despite their inability to do so the rest of the game. Thus, it's likely less a matter of cause-and-effect and more a matter of correlation between the teams that attempt 4th down and the teams that are less likely to convert on 4th down.

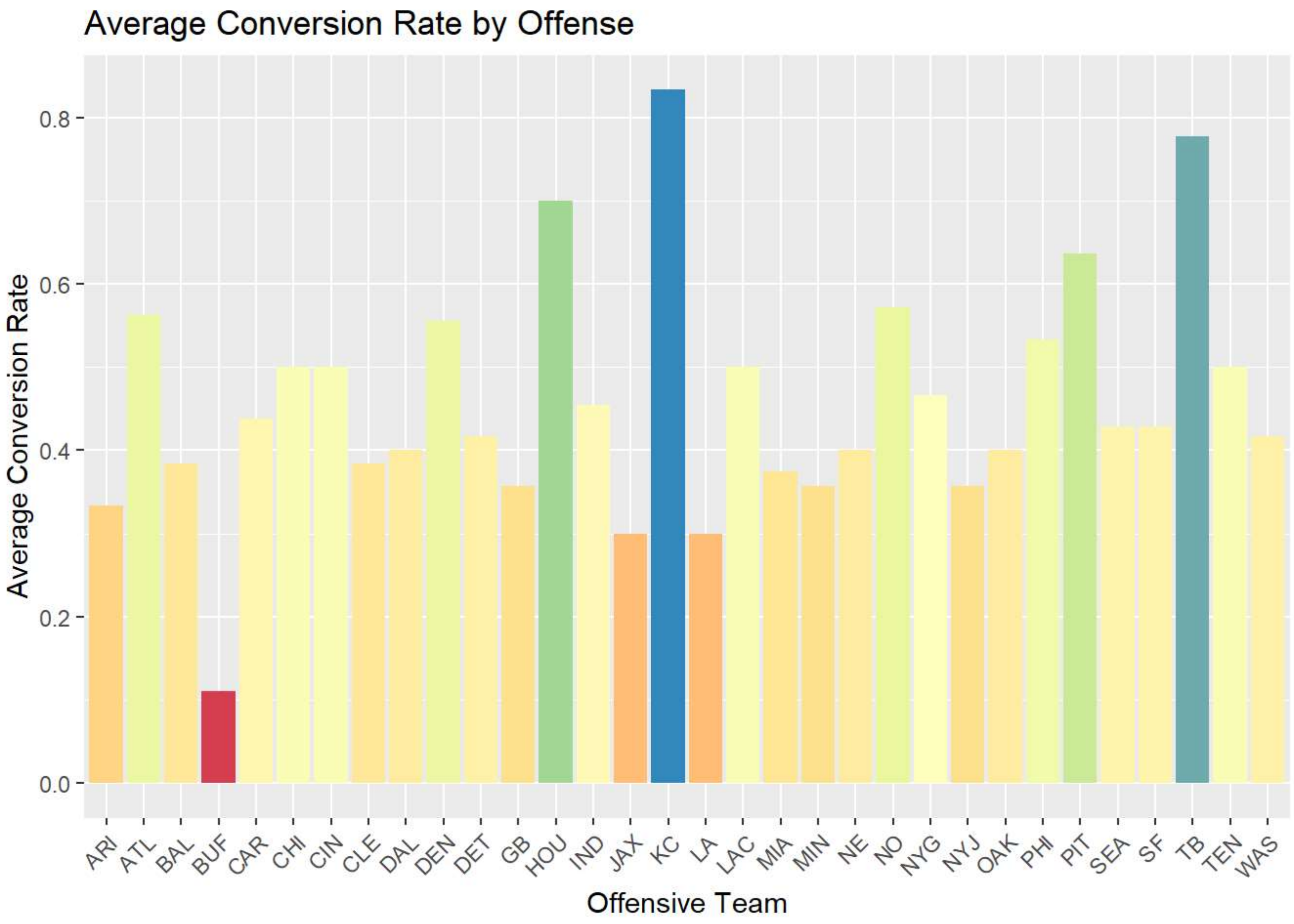
b.) What about momentum factors?

Many critics of data analytics in football have said that the decision to go for it on 4th down should not be based on numbers but rather on the momentum and feel of the game. So, in these last two code chunks I visualize 4th down conversion rates based on what I would call momentum factors. The first chart shows breaks it down by the offensive team, and the second breaks it down by quarter, home team advantage, and point-lead advantage.

b.1.) Bar chart: conversion rate by team

Like the heat map above, the average conversion rate is shown as a color, but this time, it is also the height of the bars.

```
# visualization by team
games_plays %>%
  filter(down == 4) %>%
  group_by(possessionTeam) %>%
  summarize(avg_conversion_rate = mean(converted)) %>%
  arrange(desc(avg_conversion_rate)) %>%
  ggplot(aes(x=possessionTeam, y=avg_conversion_rate, fill = avg_conversion_rate)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  scale_fill_distiller(palette = "Spectral", direction = 1) +
  labs(title = "Average Conversion Rate by Offense") +
  xlab("Offensive Team") +
  ylab("Average Conversion Rate")
```

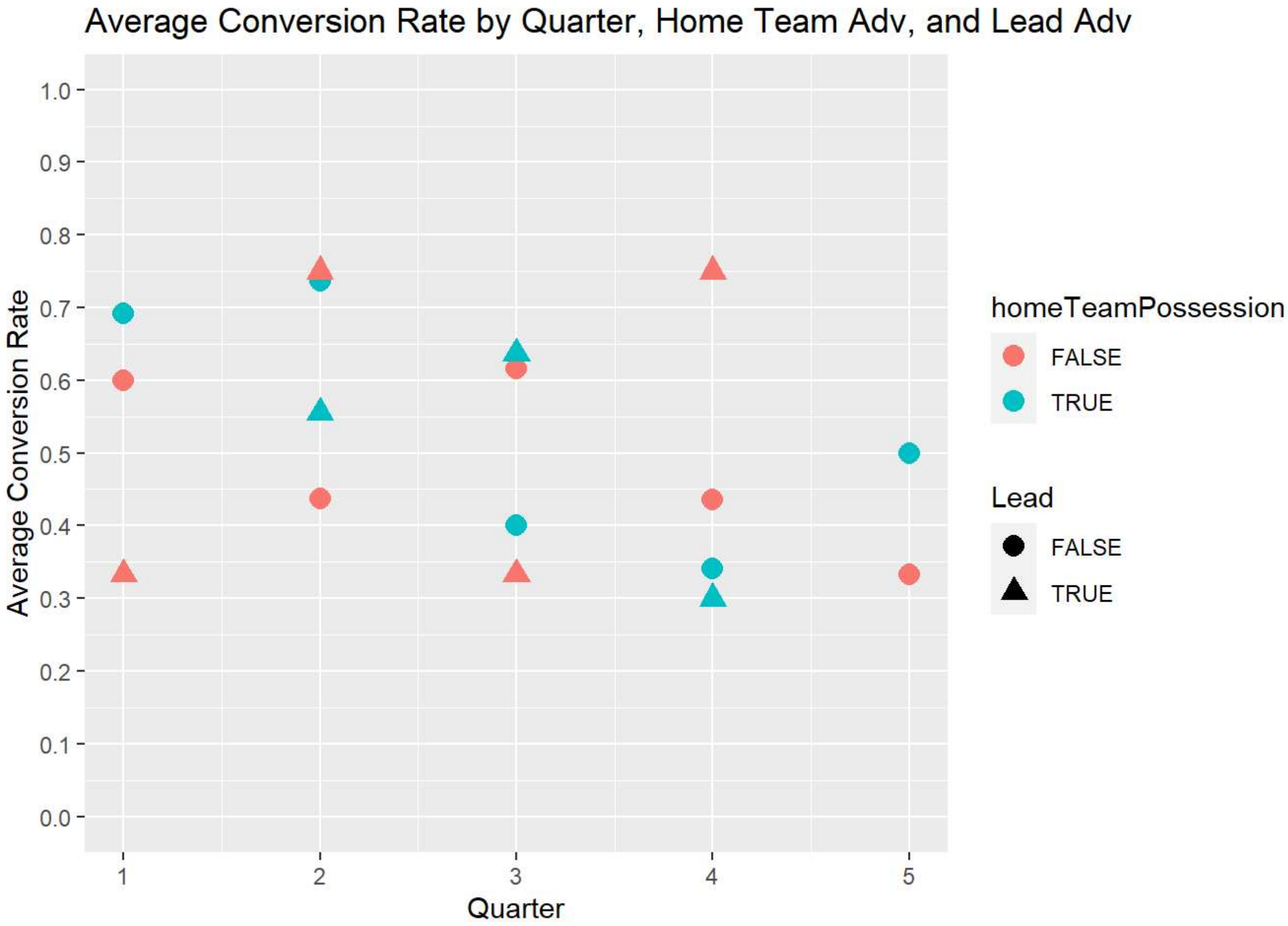


With a few exceptions like the Kansas City Chiefs, Tampa Bay Buccaneers, Houston Texans, and Buffalo Bills, almost every team is in the middle of the pack in terms of average 4th down conversion rate.

b.2.) Scatter plot: conversion rate by quarter, home team advantage, and lead advantage

This chart shows the quarter in the x axis, the average conversion rate in the y axis, home team advantage as the color, and having a point lead as the shape. This chart has a lot going on, but I think it is the only fair way to address the critic’s views about there being too many things happening in a game for the data to perfectly tell you what to do. Each situation is unique because there are so many factors to juggle, and this chart does not even show half of them.

```
# visualization of home team adv + lead adv
games_plays %>%
  filter(down == 4) %>%
  mutate(preSnapPossessionScore = ifelse(homeTeamPossession, preSnapHomeScore, preSnapVisitorSco
re),
         preSnapDefenderScore = ifelse(homeTeamPossession, preSnapVisitorScore, preSnapHomeScor
e),
         Lead = ifelse(preSnapDefenderScore < preSnapPossessionScore, TRUE, FALSE)) %>%
  group_by(Lead, quarter, homeTeamPossession) %>%
  summarize(avg_conversion_rate = mean(converted)) %>%
  filter(!is.na(Lead)) %>%
  ggplot(aes(x=quarter, y=avg_conversion_rate, shape=Lead, color=homeTeamPossession))+
  geom_point(size = 3.5) +
  scale_y_continuous(n.breaks = 12, limits = c(0,1)) +
  labs(title = "Average Conversion Rate by Quarter, Home Team Adv, and Lead Adv") +
  xlab("Quarter") +
  ylab("Average Conversion Rate")
```



The most important thing about this chart to me is what is happening in the 2nd and 3rd quarters. In the 2nd quarter, both the away team with the lead and the home team without the lead have about a 75% chance of converting on 4th down. However, in the 3rd quarter, it is exactly reversed; the home team with the lead and the away team without the lead have nearly identical, positive chances of converting. While someone could possibly use this chart to better understand their odds of going for it on 4th down, I think these two examples show that there is more to football than just a few factors that can be easily represented in a chart or two. Overall, I learned a lot about the chances of going for it in certain situations, but I also better understand the viewpoint that analytics do not tell the full story.