Set-Up Distance at Snap

Nate Rowan

10/24/2020

Objective

In this document, my goal is to see how tightly defensive backs line up against recievers at the snap, and whether this affects how well they cover routes. Some things to consider:

- see which defensive backs play tight/soft most frequently, and which teams play tight/soft most frequently
- is there a certain "type" of DB that plays tight/soft more often. Perhaps slower DB's play softer more often so they don't get burned?
- see if there is a relationship between a team's average "softness" and their performance against the pass
- under what situations do teams play softer or tighter coverage more frequently?
- under what situations is it more favorable to play tight/soft coverage?
- relationship between epa/play and tightness
- does relationship between tightness and play success change depending on type of route (go route vs slant)
- does tightness vary for teams that play man vs zone
- does the success of tightness/softness vary for teams that play different coverage schemes

```
### Load in packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gganimate)
```

```
## Warning: package 'gganimate' was built under R version 4.0.3
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(lubridate)
```

```
##
```

```
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
### Read in Big Data Bowl data
plays_data <- read_csv(here::here("datasets/plays.csv"))
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   playDescription = col_character(),
##   possessionTeam = col_character(),
##   playType = col_character(),
##   yardlineSide = col_character(),
##   offenseFormation = col_character(),
##   personnelO = col_character(),
##   personnelD = col_character(),
##   typeDropback = col_character(),
##   gameClock = col_time(format = ""),
##   penaltyCodes = col_character(),
##   penaltyJerseyNumbers = col_character(),
##   passResult = col_character(),
##   isDefensivePI = col_logical()
## )
```

```
## See spec(...) for full column specifications.
```

```
players_data <- read_csv(here::here("datasets/players.csv"))
```

```
## Parsed with column specification:
## cols(
##   nflId = col_double(),
##   height = col_character(),
##   weight = col_double(),
##   birthDate = col_character(),
##   collegeName = col_character(),
##   position = col_character(),
##   displayName = col_character()
## )
```

```
games_data <- read_csv(here::here("datasets/games.csv"))
```

```
## Parsed with column specification:
## cols(
##   gameId = col_double(),
##   gameDate = col_character(),
##   gameTimeEastern = col_time(format = ""),
##   homeTeamAbbr = col_character(),
##   visitorTeamAbbr = col_character(),
##   week = col_double()
## )
```

```
week1_tracking_data <- read_csv(here::here("datasets/week1.csv"))
```

```
## Parsed with column specification:
## cols(
```

```
##   time = col_datetime(format = ""),
##   x = col_double(),
##   y = col_double(),
##   s = col_double(),
##   a = col_double(),
##   dis = col_double(),
##   o = col_double(),
##   dir = col_double(),
##   event = col_character(),
##   nflId = col_double(),
##   displayName = col_character(),
##   jerseyNumber = col_double(),
##   position = col_character(),
##   frameId = col_double(),
##   team = col_character(),
##   gameId = col_double(),
##   playId = col_double(),
##   playDirection = col_character(),
##   route = col_character()
## )
```

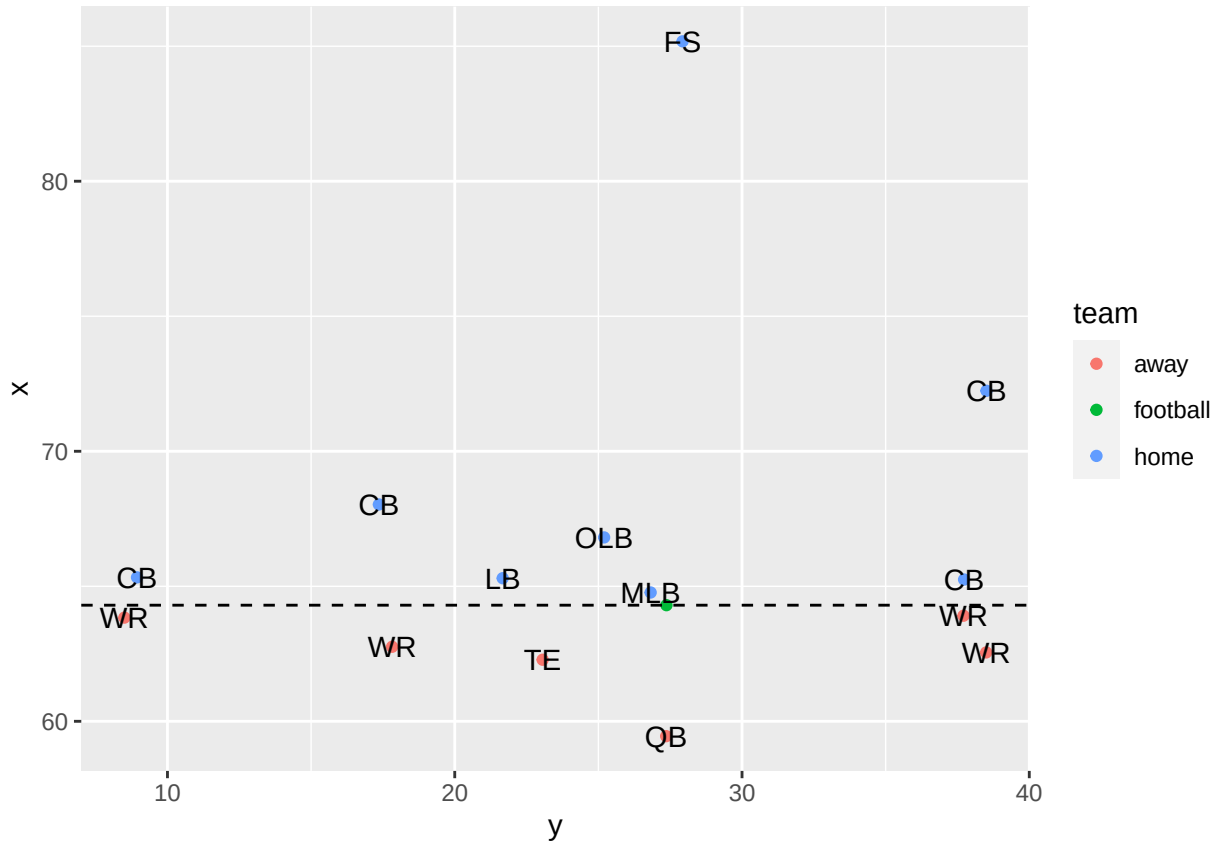
```r
### Clean data

### Standardize so that offense always goes in same direction
week1_tracking_data <- week1_tracking_data %>%
  mutate(x = ifelse(playDirection == "left", 120-x, x),
         y = ifelse(playDirection == "left", 160/3 - y, y))

### Changes names from ugly camelCase to snake_case
plays_data <- janitor::clean_names(plays_data)
players_data <- janitor::clean_names(players_data)
games_data <- janitor::clean_names(games_data)
week1_tracking_data <- janitor::clean_names(week1_tracking_data)

### Change birthdates to a datetime object
players_data <- players_data %>%
  mutate(birth_date = lubridate::parse_date_time(birth_date,
                                                 orders = c("y-m-d", "m/d/y")))
```

```r
### Joining data together

### Join plays and games data
plays_data <- left_join(plays_data, games_data, by = "game_id")

### Filter so I just have week 1 plays data
week1_plays_data <- plays_data %>%
  filter(week == 1)

### Join plays and tracking data
plays_tracking_data <- left_join(week1_plays_data,
                                 week1_tracking_data,
                                 by = c("game_id", "play_id"))

### Nest the tracking data into a list of dataframes
plays_tracking_data <- plays_tracking_data %>%
```

```
  nest(tracking_data = x:route)
```

```
### Calculate distance from line of scrimmage/reciever at snap
set.seed(123)

ex_ids <-  plays_tracking_data %>%
  filter(pass_result == "C") %>%
  slice_sample(n = 1) %>%
  select(game_id, play_id)

example_game_id <- ex_ids %>%
  select(game_id) %>%
  flatten_dbl()

example_play_id <- ex_ids %>%
  select(play_id) %>%
  flatten_dbl()

example_play <- plays_tracking_data %>%
  filter(play_id == example_play_id, game_id == example_game_id) %>%
  unnest(cols = tracking_data)

snap_data <- example_play %>%
  filter(event == "ball_snap")

snap_data %>%
  ggplot(aes(x = y, y = x)) +
    geom_point(aes(color = team)) +
    geom_hline(linetype = "dashed", yintercept = filter(snap_data, display_name == "Football") %>%
                                                   .$x) +
    geom_text(aes(label = position))
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```


FS
CB
CB
OLB
CB
LB
MLB
CB
WR
WR
WR
TE
QB
80
70
60
10
20
30
40
x
y
team
away
football
home