BDB 2026: Matt Eichenberg, Dean Hall, Zach Rose

2025-12-03

Executive Summary:

Every Sunday, millions of Americans sit down to watch their favorite quarterbacks make throws to receivers who are still coming out of their breaks on routes. Often, these professionals make it look effortless, however when it goes wrong, the ball can go sailing into space or even be thrown for an interception. For our project we decided to dive into this idea, and figure out two main questions. Which quarterbacks throw with anticipation commonly and successfully, and what determines the success of anticipation throws in the NFL.

In order to solve these questions, we decided to focus on plays where the receiver ran ins or outs. Once we located the plays we wanted to use, we used player tracking data to pinpoint when and where the receiver made their break and when the quarterback threw the ball. We used this data to help us define an anticipatory throw as one where the quarterback throws the ball very soon after the receiver makes his break.

Once we gathered our data, we figured out how often each QB attempts anticipation throws, how frequently they complete them, and how much value they generate. We also tested whether certain factors, such as the type of coverage or the time between the break and the throw, determine the success of anticipation throws. In our research, our biggest finding was that anticipatory throws were much more effective against zone coverage than man coverage.

For our last task, we decided to go into further detail on the incomplete passes, and assign blame for to incompletions. To do this, we looked at the positioning of the receivers and defenders at the point in which the quarterback's throw hit the ground. We then split them into 4 different categories depending on our analysis of the play. These categories were bad throw, good defense, all around bad, and drop.

Technical Report

We loaded in the datasets and libraries. Input data is tracking data that happens before the ball is thrown, while output data happens after the ball is thrown. Supplementary data is useful to get results of plays.

```r
setwd("~/Desktop/BigDataBowl2026")
library(tidyverse)
library(knitr)
library(scales)

load('all_inputs.RData')
load('all_outputs.RData')
supp <- read.csv('supplementary_data.csv')
```

The first step once we loaded in the dataset, was to get the plays we wanted to use in our analysis. These are plays where the targeted player was running an in or out route and the play was not nullified by a penalty. We decided to only look at ins and outs for this project.

```r
ins_outs <- supp %>%
  filter(route_of_targeted_receiver %in% c('IN', 'OUT'),
         play_nullified_by_penalty == 'N') %>%
  select(game_id, play_id, play_description, home_team_abbr, visitor_team_abbr,
         quarter, game_clock, pass_result, pass_length, expected_points_added)
```

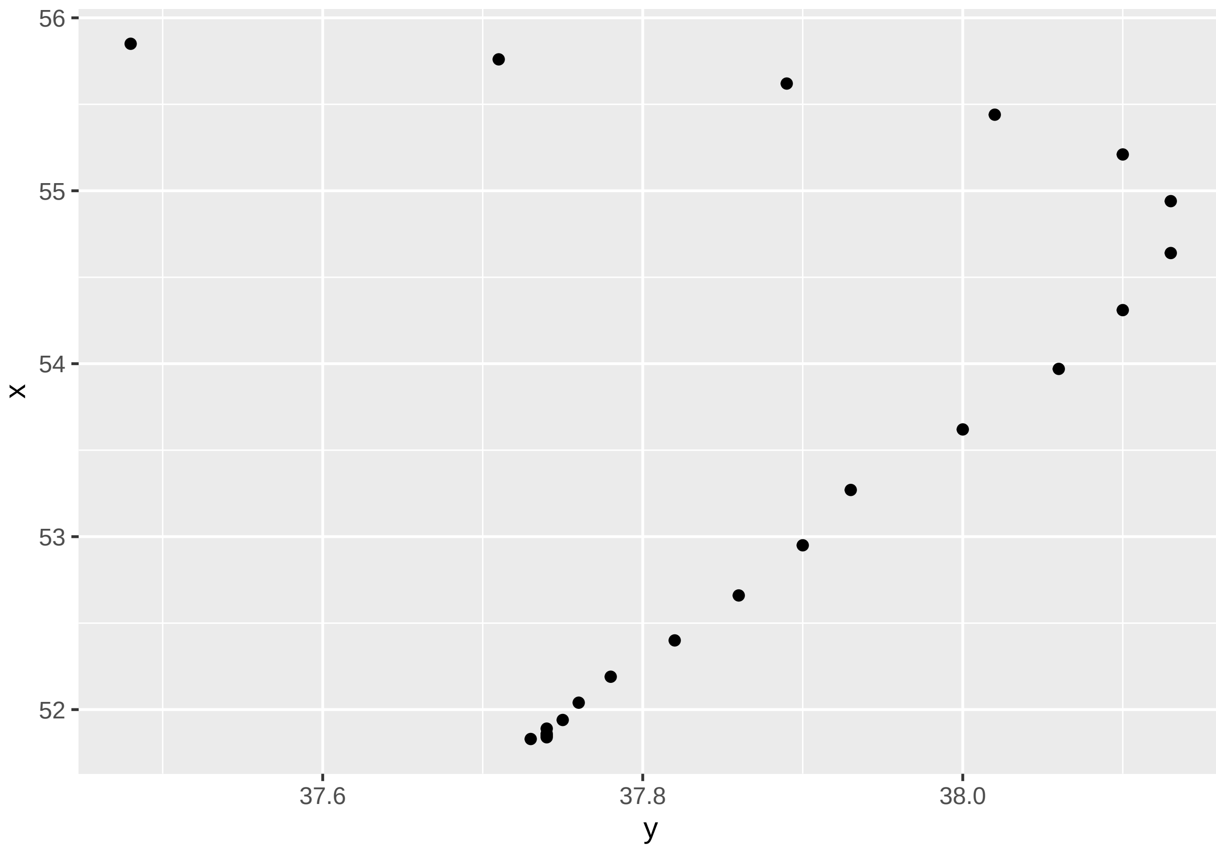
Determining if a throw was anticipatory

First we will examine our process for one particular play, but the following process was then extrapolated to all plays in a for loop. The most important factor when determining if a throw was anticipatory was to find the frame when the break point occurred. To do this, we first determined how far from his original y point the receiver was at every given frame. The graph below shows the outline of the route.

```r
input <- input1 %>%
  filter(game_id == 2023090700, play_id == 1618)



play <- input %>%
    filter(player_to_predict == "True",
           player_position %in% c("WR", 'TE'))%>%
    semi_join(ins_outs, by = c("game_id", "play_id"))%>%
    group_by(game_id, play_id) %>%
    mutate(dist_from_first_y = abs(y - first(y))) %>%
    select(game_id, play_id, x, y, frame_id, dist_from_first_y)

play %>%
  head() %>%
  kable()
```

game_id	play_id	x	y	frame_id	dist_from_first_y
2023090700	1618	51.83	37.73	1	0.00
2023090700	1618	51.84	37.74	2	0.01
2023090700	1618	51.85	37.74	3	0.01
2023090700	1618	51.86	37.74	4	0.01
2023090700	1618	51.89	37.74	5	0.01
2023090700	1618	51.94	37.75	6	0.02

```r
ggplot(play, aes(x = x, y =y))+
  geom_point()+
  coord_flip()
```



Before finding the frame where the route broke, we first found the top of the route which we defined as the point after the route break where the receiver was closest to his original y. We did not look at the first 10 frames for a play for this in order to ensure that the break had already happened.

```
top_of_route <- play %>%
    filter(frame_id > 10) %>%
    group_by(game_id, play_id) %>%
    slice_min(dist_from_first_y, n = 1) %>%
    select(game_id, play_id, top_of_route_frame = frame_id)
```

After finding the top of the route, we were able to determine the route break point. We traced the route back towards the start of the route until we found the most extreme difference in y. On ins and outs routes, the receiver begins by running the opposite direction of his planned route. As you can see, the receiver began by running to the right even though the design was for him to go left. This means that the break, or when the receiver begins to turn, is the most extreme y value that occurred before the top of the route. Starting at the top of the route, we backtracked to find the frame in which the y value was farthest from the receiver's original y.

```
route_break <- play %>%
    left_join(top_of_route, by = c("game_id", "play_id")) %>%
    filter(frame_id < top_of_route_frame) %>%
    group_by(game_id, play_id) %>%
    slice_max(dist_from_first_y, n = 1, with_ties = F) %>%
    select(game_id, play_id, top_of_route_frame, route_break_frame = frame_id)
```

We gathered the important information about each route into one dataframe. To determine if a throw was anticipatory, we looked at the amount of frames between when the receiver made his break and when the quarterback threw the ball. Only throws with less than 8 frames between the break and throw were considered to be thrown with anticipation.

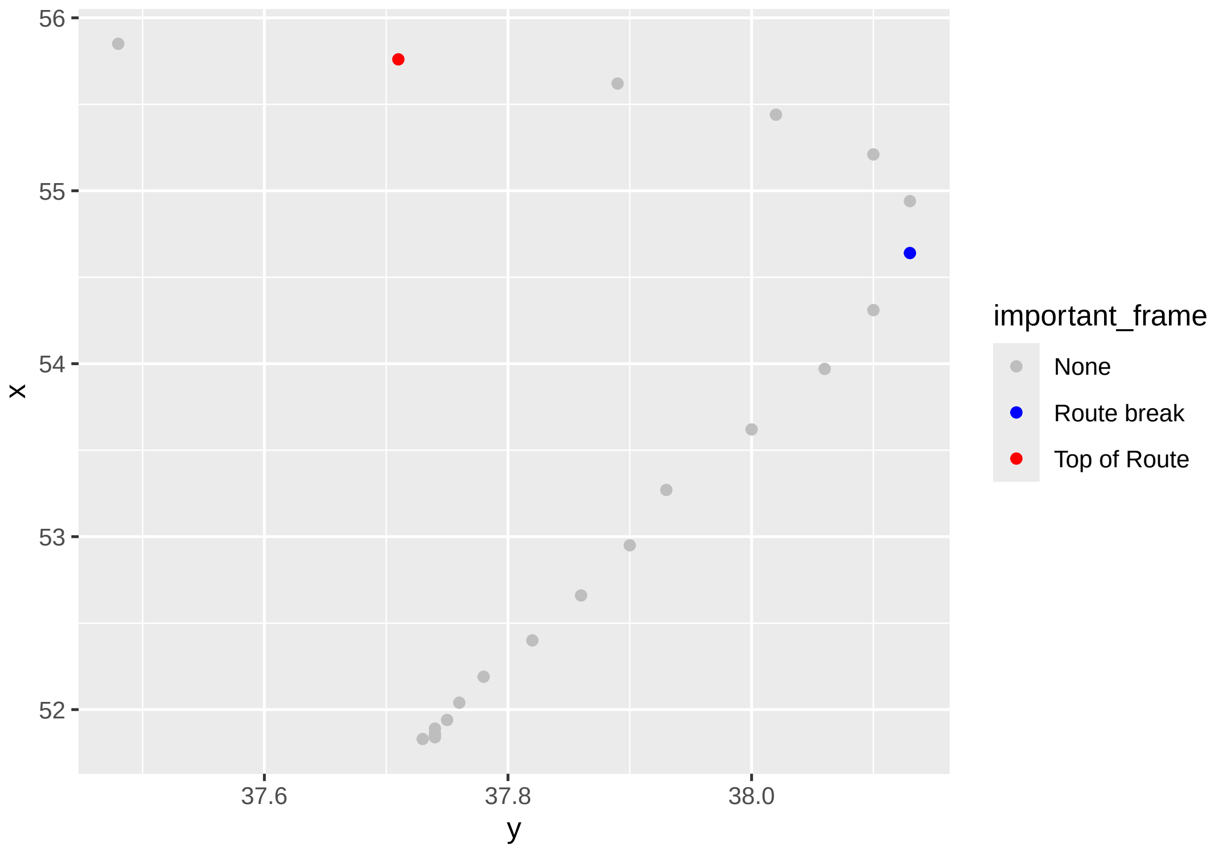
```
route_info <- play %>%
    group_by(game_id, play_id) %>%
```

```r
    summarise(total_frames = max(frame_id)) %>%
    select(game_id, play_id, total_frames) %>%
    left_join(route_break, by = c("game_id", "play_id")) %>%
    mutate(frames_from_break_to_throw = total_frames - route_break_frame)

anticipation_play <- route_info %>%
  filter(frames_from_break_to_throw < 9)

anticipation_play %>%
  kable()
```

game_id	play_id	total_frames	top_of_route_frame	route_break_frame	frames_from_break_to_throw
2023090700	1618	22	21	16	6

This graph shows the route with the important points marked. There are only 6 points (frames) between the break and the throw so it is classified as an anticipation throw.

```r
graph_data <- play %>%
  left_join(route_info, by = c("game_id", "play_id"))%>%
  mutate(
    important_frame = case_when(
    top_of_route_frame == frame_id ~ 'Top of Route',
    route_break_frame == frame_id ~ 'Route break',
    .default = 'None'))

ggplot(graph_data, aes(x = x, y = y, color = important_frame))+
  geom_point()+
  coord_flip()+
  scale_color_manual(
    values = c(
      "Top of Route" = "red",
      "Route break" = "blue",
      "None" = "grey"
    )
  )
```



We added information about the QB who threw the pass and where the ball landed to the dataset to help determine player performance later on.

```
qb_mapping <- input %>%
    filter(player_position == 'QB') %>%
    select(game_id, play_id, nfl_id, player_name) %>%
    unique()

anticipation_play <- anticipation_play %>%
  left_join(qb_mapping, by = c("game_id", "play_id")) %>%
  select(game_id, play_id, nfl_id, player_name,
         total_frames, top_of_route_frame, route_break_frame,
         frames_from_break_to_throw)

ball_land_mapping <- input %>%
  select(game_id, play_id, ball_land_x, ball_land_y) %>%
  distinct()

anticipation_play <- anticipation_play %>%
  left_join(ball_land_mapping, by = c("game_id", "play_id"))

route_break_frames <- anticipation_play %>% select(game_id, play_id,
                                                   route_break_frame)

player_mapping <- route_break_frames %>%
  inner_join(play, by = c("game_id", "play_id",
                          'route_break_frame' = 'frame_id'))

anticipation_play <- anticipation_play %>%
  left_join(player_mapping, by = c("game_id", "play_id"))
```

We replicated the above process for all relevant plays across the 18 week season.

```r
anticipation_ins_outs_list <- list()



for (i in 1:18){
  input <- get(paste0("input", i))


  plays <- input %>%
    filter(player_to_predict == "True",
           player_position %in% c("WR", 'TE')) %>%
    semi_join(ins_outs, by = c("game_id", "play_id"))%>%
    group_by(game_id, play_id) %>%
    mutate(dist_from_first_y = abs(y - first(y))) %>%
    select(game_id, play_id, x, y, frame_id, dist_from_first_y)

  top_of_routes <- plays %>%
    filter(frame_id > 10) %>%
    group_by(game_id, play_id) %>%
    slice_min(dist_from_first_y, n = 1) %>%
    select(game_id, play_id, top_of_route_frame = frame_id)

  route_breaks <- plays %>%
    left_join(top_of_routes, by = c("game_id", "play_id")) %>%
    filter(frame_id < top_of_route_frame) %>%
    group_by(game_id, play_id) %>%
    slice_max(dist_from_first_y, n = 1, with_ties = F) %>%
    select(game_id, play_id, top_of_route_frame, route_break_frame = frame_id)

  route_info <- plays %>%
    group_by(game_id, play_id) %>%
    summarise(total_frames = max(frame_id)) %>%
    select(game_id, play_id, total_frames) %>%
    left_join(route_breaks, by = c("game_id", "play_id")) %>%
    mutate(frames_from_break_to_throw = total_frames - route_break_frame)

  qb_mapping <- input %>%
    filter(player_position == 'QB') %>%
    select(game_id, play_id, nfl_id, player_name) %>%
    unique()

  anticipation_ins_outs <- route_info %>%
    filter(frames_from_break_to_throw < 9) %>%
    left_join(qb_mapping, by = c("game_id", "play_id")) %>%
    select(game_id, play_id, nfl_id, player_name,
           total_frames, top_of_route_frame, route_break_frame,
           frames_from_break_to_throw)


  ball_land_mapping <- input %>%
    select(game_id, play_id, ball_land_x, ball_land_y) %>%
    distinct()

  anticipation_ins_outs <- anticipation_ins_outs %>%
    left_join(ball_land_mapping, by = c("game_id", "play_id"))

  route_break_frames <- anticipation_ins_outs %>% select(game_id, play_id,
                                                         route_break_frame)

  player_mapping <- route_break_frames %>%
    inner_join(plays, by = c("game_id", "play_id",
```

```r
                              'route_break_frame' = 'frame_id'))

  anticipation_ins_outs <- anticipation_ins_outs %>%
    left_join(player_mapping, by = c("game_id", "play_id"))


  anticipation_ins_outs_list[[i]] = anticipation_ins_outs

}


anticipation_ins_outs <- bind_rows(anticipation_ins_outs_list)

anticipation_ins_outs %>%
  ungroup() %>%
  select(player_name, total_frames, top_of_route_frame, route_break_frame.x,
         ball_land_x, ball_land_y, x, y) %>%
  rename(route_break_frame = route_break_frame.x) %>%
  head() %>%
  kable()
```

player_name	total_frames	top_of_route_frame	route_break_frame	ball_land_x	ball_land_y	x	y
Patrick Mahomes	22	21	16	54.16	33.34	54.64	38.13
Jared Goff	22	19	15	82.24	39.21	83.00	44.05
Bryce Young	23	23	17	67.98	31.13	66.78	36.80
Deshaun Watson	22	21	14	90.82	41.88	90.03	35.57
Trevor Lawrence	16	11	10	46.59	44.65	52.68	34.83
Trevor Lawrence	17	11	10	80.60	7.23	77.17	14.35

Analyzing player performance

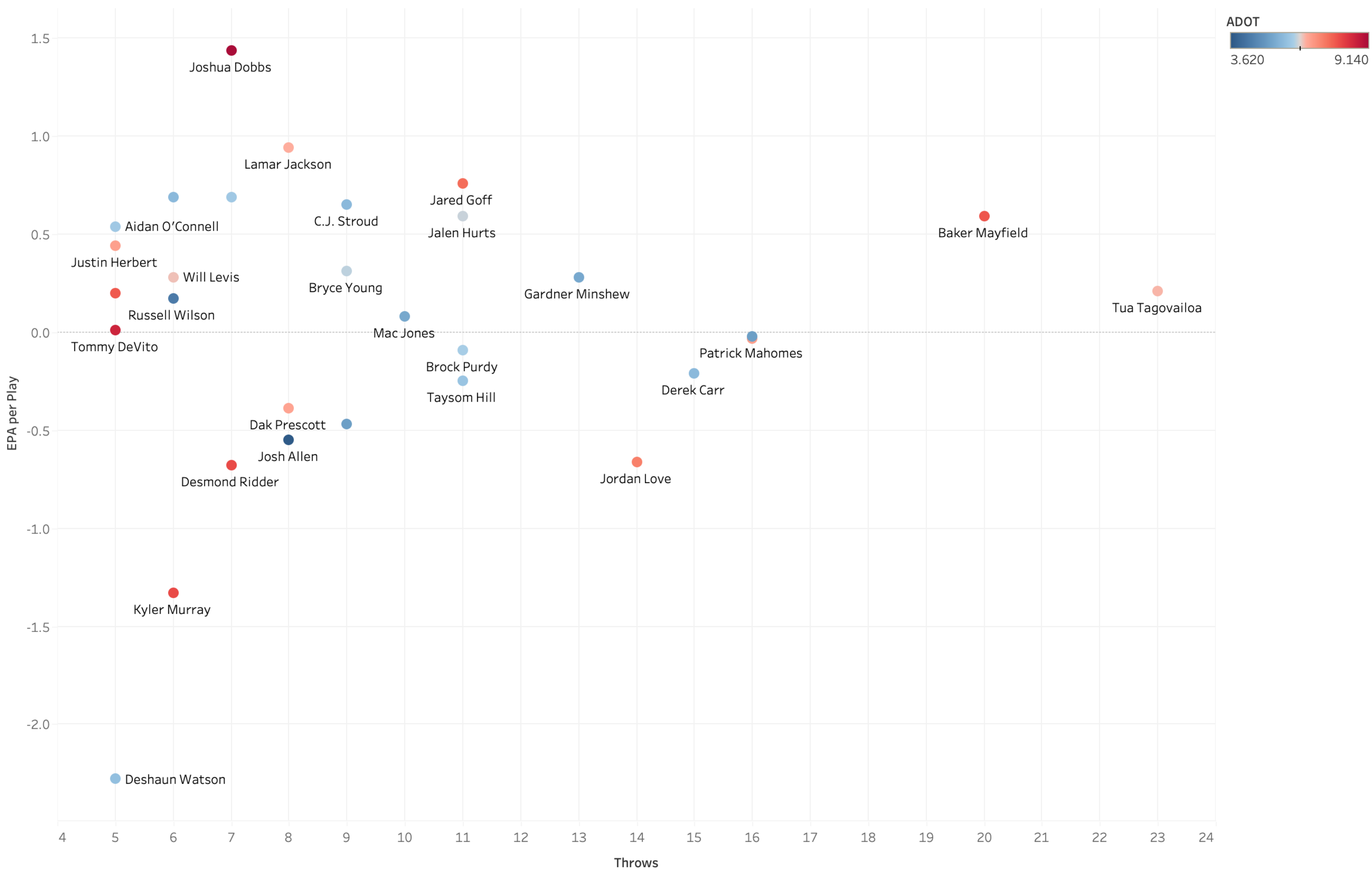
To determine player performance we used the supplementary dataset. For each quarterback that attempted an anticipation throw, we calculated important metrics such as completion percentage, EPA (expected points added) and ADOT (average depth of target).

```r
anticipation_ins_outs_stats <- anticipation_ins_outs %>%
  left_join(supp, by = c('game_id', 'play_id')) %>%
  select(game_id, play_id, nfl_id, player_name,
         week, home_team_abbr, visitor_team_abbr, quarter,
         play_description, pass_result, pass_length, expected_points_added)


anticipation_ins_outs_stats <- anticipation_ins_outs_stats %>%
  group_by(nfl_id, player_name) %>%
  summarise(throws = n(),
            completions = sum(pass_result == 'C'),
            total_EPA = sum(expected_points_added),
            ADOT = round(mean(pass_length), 2)) %>%
  mutate(completion_percentage = percent(completions / throws),
         EPAperPlay = round(total_EPA/throws, 2)) %>%
  select(-total_EPA) %>%
  arrange(desc(throws))

anticipation_ins_outs_stats %>%
  head() %>%
  kable()
```

nfl_id	player_name	throws	completions	ADOT	completion_percentage	EPAperPlay
52413	Tua Tagovailoa	23	14	6.57	61%	0.21
46070	Baker Mayfield	20	16	7.85	80%	0.59
44822	Patrick Mahomes	16	14	5.12	88%	-0.02
53430	Trevor Lawrence	16	10	6.81	62%	-0.03
41265	Derek Carr	15	7	5.67	47%	-0.21
52434	Jordan Love	14	7	7.29	50%	-0.66

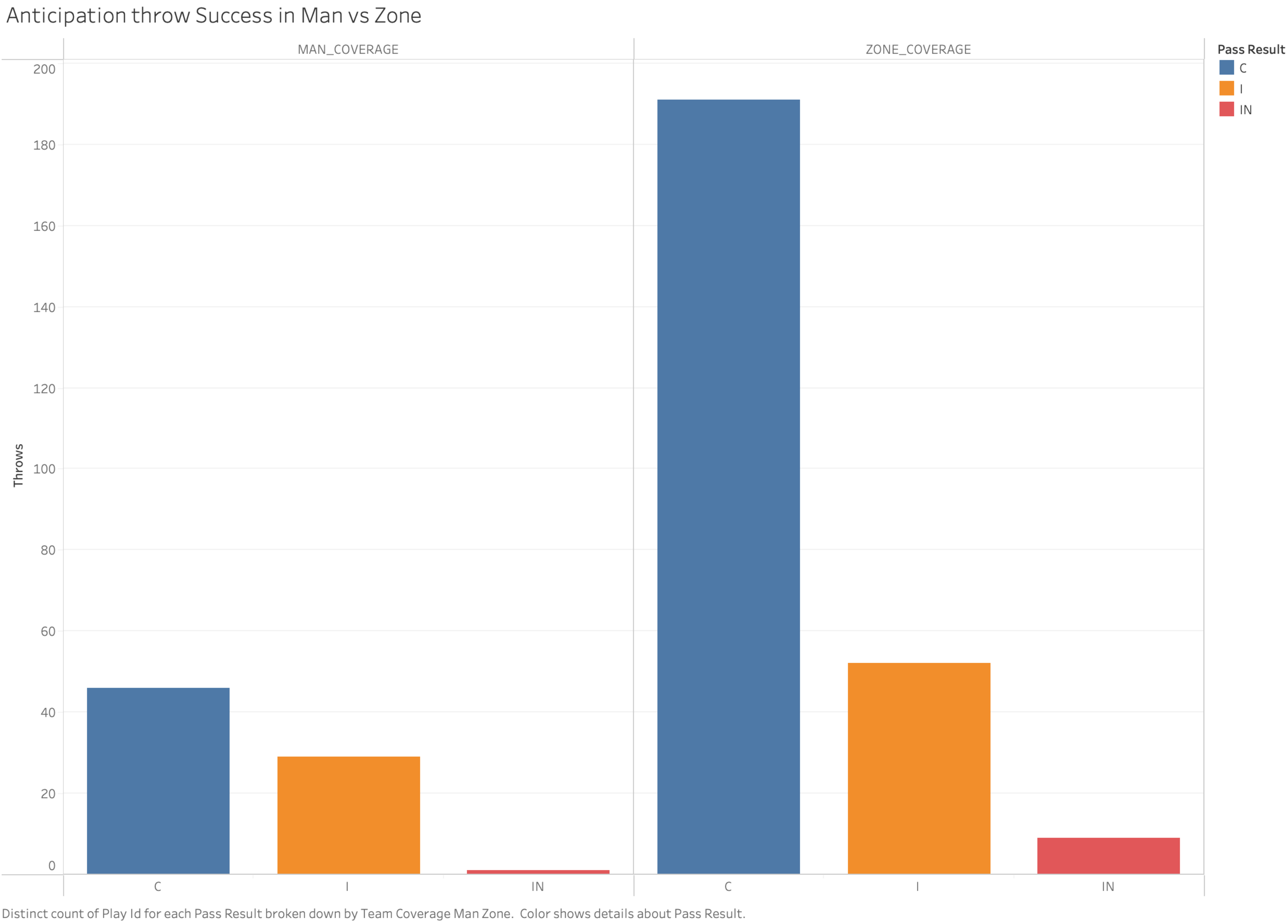
Below is a helpful visualization to better understand the success of anticipation throws. As you can see, Tua Tagovailoa and Baker Mayfield threw the most anticipation throws, and it worked for them. Both of them had positive EPA numbers and above average ADOTs, so they were making difficult passes. Josh Dobbs is also an interesting case. He did not throw a lot of anticipation throws, but when he did he threw them far and they were highly successful.

ADOT and EPA for NFL QBs on Anticipation Throws



Throws vs. EP Aper Play. Color shows sum of Adot. The marks are labeled by Player Name. The data is filtered on sum of Throws, which includes values greater than or equal to 5.

This graph shows that leaguewide, anticipation throws are used more commonly against zone coverage than man coverage. This makes sense, as they are often used to find the soft spots in a zone. When facing zone coverage, the completion percentage is much higher.



Assigning blame

For this section, we attempted to determine who was at fault for an incomplete pass. We first filtered the data to only look at incompletions. We first started by only looking at week 1 plays, but later extrapolated the process to the whole season.

```
incompletions <- anticipation_ins_outs %>%
  left_join(supp %>% select(game_id, play_id, pass_result), by = c('game_id', 'play_id')) %>%
  filter(pass_result %in% c('I', 'IN')) %>%
  select(game_id, play_id, player_name, ball_land_x, ball_land_y)
```

We first gathered the positions of all players on the field for a given throw as well as where the ball ended up.

```
output <- output1
input <- input1

position_mapping <- input %>%
    select(nfl_id, player_position) %>%
    distinct()
```

```r
landings <- input %>%
  select(game_id, play_id, ball_land_x, ball_land_y) %>%
  distinct()

qb_mapping <- input %>%
  filter(player_position == 'QB') %>%
  select(game_id, play_id, nfl_id, player_name) %>%
  unique()

separation <- output %>%
  semi_join(incompletions, by = c('game_id', 'play_id')) %>%
  left_join(position_mapping, by = 'nfl_id') %>%
  left_join(landings, by = c('game_id', 'play_id'))



separation %>%
  head() %>%
  kable()
```

game_id	play_id	nfl_id	frame_id	x	y	player_position	ball_land_x	ball_land_y
2023091008	840	48516	1	37.00	39.13	OLB	49.01	37.5
2023091008	840	48516	2	37.21	39.48	OLB	49.01	37.5
2023091008	840	48516	3	37.45	39.83	OLB	49.01	37.5
2023091008	840	48516	4	37.72	40.16	OLB	49.01	37.5
2023091008	840	48516	5	38.03	40.46	OLB	49.01	37.5
2023091008	840	48516	6	38.36	40.73	OLB	49.01	37.5

Next we determined how far a player was from the ball at a given frame. This gives us a good idea of if the receiver was open.

```r
separation <- separation %>%
    group_by(game_id, play_id) %>%
    slice_max(frame_id, n = 1) %>%
    mutate(x_dist = ball_land_x - x,
           y_dist = ball_land_y - y,
           dist_to_ball = sqrt(x_dist^2 + y_dist^2)) %>%
    select(-x_dist, -y_dist)

separation %>%
  head() %>%
  kable()
```

game_id	play_id	nfl_id	frame_id	x	y	player_position	ball_land_x	ball_land_y	dist_to_ball
2023091008	840	48516	11	40.40	41.31	OLB	49.01	37.50	9.415316
2023091008	840	55921	11	46.42	34.48	CB	49.01	37.50	3.978504
2023091008	840	56220	11	45.98	34.23	WR	49.01	37.50	4.458003
2023091010	1249	46086	9	8.75	12.18	SS	10.31	12.51	1.594522
2023091010	1249	46775	9	7.98	5.95	CB	10.31	12.51	6.961502
2023091010	1249	52431	9	10.23	17.43	MLB	10.31	12.51	4.920650

We looked at the last frame of the output data (when the ball hits the ground) and determined how far the closest defender and the offensive player were to the ball. We added a column, diff, to which shows the difference between the receiver and closest defender. If it is positive, the receiver is closer and if it is negative the defender is closer. We also added in information about which quarterback threw the pass.

```r
separation <- separation %>%
    mutate(player_type = case_when(
      player_position %in% c('WR', 'TE') ~ 'offense',
      .default = 'defense'
    )) %>%
    group_by(game_id, play_id, player_type) %>%
    summarise(closest_player = min(dist_to_ball)) %>%
    pivot_wider(names_from = player_type,
                values_from = closest_player) %>%
    mutate(diff = defense - offense) %>%
    rename(def_dist_to_ball = defense,
           off_dist_to_ball = offense)

separation <- separation %>%
  left_join(qb_mapping, by = c('game_id', 'play_id'))

separation %>%
  head() %>%
  kable()
```

game_id	play_id	def_dist_to_ball	off_dist_to_ball	diff	nfl_id	player_name
2023091008	840	3.9785037	4.4580029	-0.4794992	52434	Jordan Love
2023091010	1249	1.5945223	1.8796281	-0.2851058	52413	Tua Tagovailoa
2023091010	3803	1.6025623	1.6229006	-0.0203383	52413	Tua Tagovailoa
2023091011	952	0.8523493	0.7349841	0.1173652	53444	Mac Jones
2023091013	1015	2.9751161	1.3961750	1.5789412	43424	Dak Prescott

The above process was repeated for all weeks.

```r
separation_list <-  list()

for (i in 1:18){
  output <- get(paste0("output", i))
  input <- get(paste0("input", i))


  position_mapping <- input %>%
    select(nfl_id, player_position) %>%
    distinct()

  landings <- input %>%
    select(game_id, play_id, ball_land_x, ball_land_y) %>%
    distinct()

  separation <- output %>%
    semi_join(incompletions, by = c('game_id', 'play_id')) %>%
    left_join(position_mapping, by = 'nfl_id') %>%
    left_join(landings, by = c('game_id', 'play_id')) %>%
    group_by(game_id, play_id) %>%
    slice_max(frame_id, n = 1) %>%
    mutate(x_dist = ball_land_x - x,
           y_dist = ball_land_y - y,
           dist_to_ball = sqrt(x_dist^2 + y_dist^2)) %>%
    select(-x_dist, -y_dist) %>%
    mutate(player_type = case_when(
      player_position %in% c('WR', 'TE') ~ 'offense',
      .default = 'defense'
    )) %>%
    group_by(game_id, play_id, player_type) %>%
```

```r
    summarise(closest_player = min(dist_to_ball)) %>%
    pivot_wider(names_from = player_type,
                values_from = closest_player) %>%
    mutate(diff = defense - offense) %>%
    rename(def_dist_to_ball = defense,
           off_dist_to_ball = offense)

  qb_mapping <- input %>%
  filter(player_position == 'QB') %>%
  select(game_id, play_id, nfl_id, player_name) %>%
  unique()

  separation <- separation %>%
    left_join(qb_mapping, by = c("game_id", "play_id"))


  separation_list[[i]] = separation
}

separation <- bind_rows(separation_list)


separation %>%
  head() %>%
  kable()
```

game_id	play_id	def_dist_to_ball	off_dist_to_ball	diff	nfl_id	player_name
2023091008	840	3.9785037	4.4580029	-0.4794992	52434	Jordan Love
2023091010	1249	1.5945223	1.8796281	-0.2851058	52413	Tua Tagovailoa
2023091010	3803	1.6025623	1.6229006	-0.0203383	52413	Tua Tagovailoa
2023091011	952	0.8523493	0.7349841	0.1173652	53444	Mac Jones
2023091013	1015	2.9751161	1.3961750	1.5789412	43424	Dak Prescott
2023091700	989	2.6759846	5.0820651	-2.4060805	52434	Jordan Love

Conclusion and next steps

We are happy with the process we used to determine which plays were anticipatory throws. After watching film on some of these plays, it is clear that the ones we identified are anticipatory and the ones we considered identifying (increasing the amount of frames to throw) were not anticipatory. A next logical step would be to extend this research to hook, slant, and post routes. The process would likely look different, and the break point would have to be calculated in another way, but once this was figured out, adding the extra data would vastly help our research, especially in the blame portion. After adding in the other routes, we would continue our analysis of assigning blame onto players by using statistics such as catch probability to truly know who is at fault for an incomplete pass.