Generating a novel ‘Clutch’ metric for NFL wide receivers in the 2022 season

Matt Sheridan

2023-06-16

Personal Introduction

My name is Matthew Sheridan, I’m a recent college graduate, and this is my quick, rough way of putting who I am on paper.

To begin, I’ll reveal a little about myself - I’m two things: a sports nerd and just the right amount of crazy. My sports analytics origins lie in Ice Hockey, where I became well versed to the point where I could help during Harvard’s Sports Analysis Collective’s discussions, and I’d consider myself to be just as much of a hockey nerd. I can code pretty well, I learn on the fly, and most importantly, I’m a confident, bold individual who knows that their best trait is asking as many questions as humanely possible.

To get to the point, however, I’d like to showcase to you both how I think and my still growing ability to code, and in that case I’ll be using R to give my take on a novel ‘Clutch’ metric for NFL wide receivers from 2022.

Now this is obviously a rough metric, with MANY assumptions made. I’ve sparingly worked with NFL pbp data before, so not only is this a showing of my abilities, it’s also an exemplification of a journey from knowing little to being more comfortable and familiar.

If you have any further questions, I can be reached at 617-529-2271 or matthewsheridan3627@gmail.com

Project Introduction

Wide receivers… are weird. It is quite hard to isolate the impact that an individual receiver has over another because so many factors are different between all of them. Quarterback play, head coach, offensive schemes, opposing corner backs, and many, many more factors are difficult to be isolated with limited resources and computing power.

The first large assumption I’ll be making is that yards after the catch are one of the most important things receivers can affect individually. This is a large jump, because individual skill contributes a lot to actually getting to a spot to make the catch, but the actual throw is arguably just as crucial. Thus, since a receiver defender matchup will define the YAC, using the YAC win probability added metric will be a good proxy for actual receiver outcome (considering WPA is a good baseline for evaluating how ‘clutch’ or important a play is).

Instead of simply aggregating to get a mean YAC WPA, which would show an average level of important play and to an extent defines how clutch a player is, I want to try to adjust for the importance of that players’ games.

To do this, ideally I would be gathering the team’s playoff probability if they win that particular game, and scale the aggregated metric in that game accordingly (a player should be rewarded for having a good WPA in a game that helps their team make the playoffs - these players generally know the stakes). Considering this data is not readily available and a lack of time / resources, I will make it simply and imply that a game when a team is close to .500 is more important that one further away, as around 0.500 is a common threshold for making the playoffs (but is obviously very rough). We will be using the Gaussian probability density function to calculate a relative importance for a team having finished with a certain number of wins and normalizing the values to form a sort of discrete probability density function based roughly on the Gaussian one.

For example, let's say the normalized importance for 9 wins is 0.2. If we're looking at a team like Detroit who finished with 9 wins, for simplicity's sake we'll be assuming that the entire season their players needed to

Data / Tools

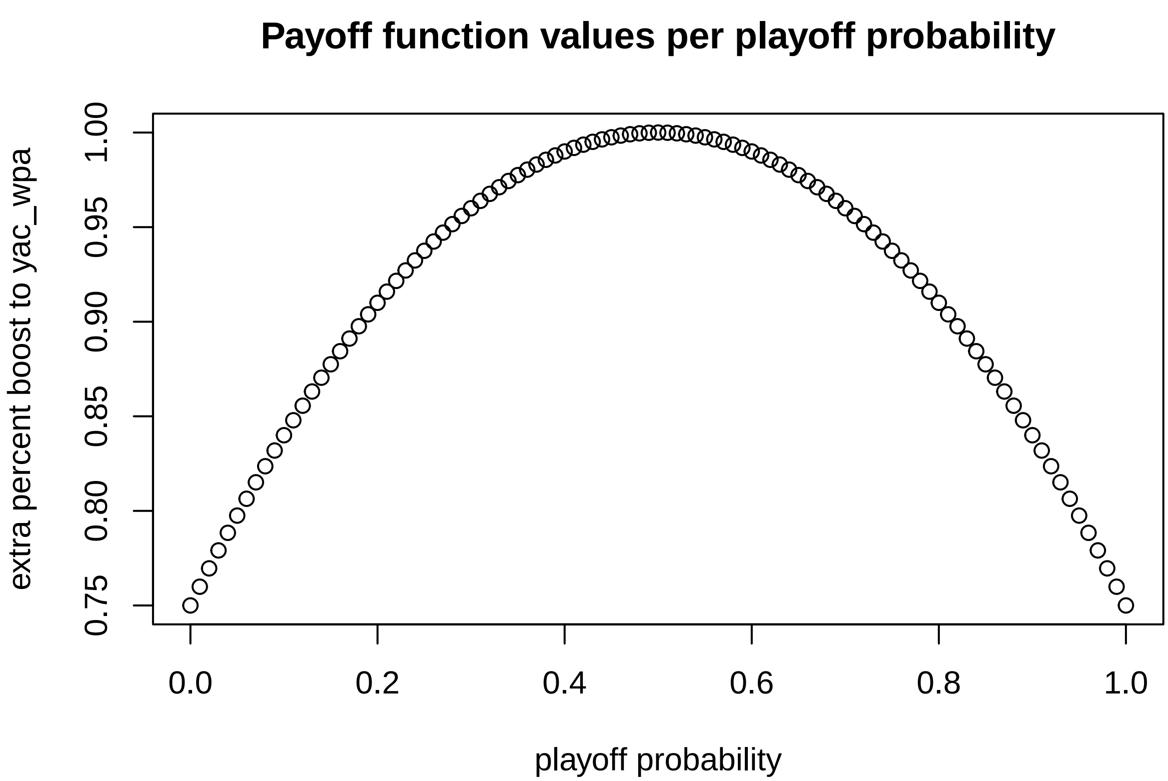
In

```
library(tidyverse)
library(ggrepel)
library(nflreadr)
library(nflplotR)
library(coefplot)
library(dplyr)
library(rvest)
options(scipen=9999)
```

```
playoff_probs <- (matrix(c(0.44, 0.32, 0.22, 0.14, 0.08,0.04,0.02,0.01,0,0,0,0,0,0,0,0,0,0,
                        0.55,0.43,0.32,0.21,0.13,0.07,0.03,0.01,0,0,0,0,0,0,0,0,0,0,
                        0.67,0.55,0.42,0.3,0.2,0.11,0.06,0.02,0.01,0,0,0,0,0,0,0,0,0,
                        0.76,0.67,0.55,0.42,0.29,0.18,0.1,0.04,0.01,0,0,0,0,0,0,0,0,0,
                        0.85,0.77,0.67,0.54,0.41,0.28,0.16,0.08,0.03,0.01,0,0,0,0,0,0,0,0,
                        0.91,0.85,0.77,0.67,0.54,0.39,0.25,0.13,0.05,0.01,0,0,0,0,0,0,0,0,
                        0.95,0.92,0.87,0.78,0.67,0.53,0.37,0.22,0.09,0.03,0,0,0,0,0,0,0,0,
                        0.98,0.96,0.93,0.88,0.8,0.69,0.53,0.34,0.17,0.04,0.01,0,0,0,0,0,0,0,
                        0.99,0.98,0.97,0.95,0.9,0.82,0.7,0.52,0.28,0.11,0,0,0,0,0,0,0,0,
                        1,1,0.99,0.98,0.96,0.93,0.86,0.74,0.5,0,0,0,0,0,0,0,0,0,
                        1,1,1,1,0.99,0.98,0.97,0.91,0,0,0,0,0,0,0,0,0,0,
                        1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,
                        1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
                        1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
                        1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                        1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                        1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                        1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
                        nrow = 18, ncol = 18))

rownames(playoff_probs) <- 0:17
colnames(playoff_probs) <- 0:17

payoff_function <- function(x){
#  ifelse(x > 0.5, (1.5 - x + 0.5)^2, (x + 1)^2)
  -((x-0.5)^2) + 1
}

playoff_probs
```

```
##        0    1    2    3    4    5    6    7    8    9   10 11 12 13 14 15 16 17
## 0  0.44 0.55 0.67 0.76 0.85 0.91 0.95 0.98 0.99 1.00 1.00  1  1  1  1  1  1  1
## 1  0.32 0.43 0.55 0.67 0.77 0.85 0.92 0.96 0.98 1.00 1.00  1  1  1  1  1  1  0
## 2  0.22 0.32 0.42 0.55 0.67 0.77 0.87 0.93 0.97 0.99 1.00  1  1  1  1  1  0  0
## 3  0.14 0.21 0.30 0.42 0.54 0.67 0.78 0.88 0.95 0.98 1.00  1  1  1  1  0  0  0
## 4  0.08 0.13 0.20 0.29 0.41 0.54 0.67 0.80 0.90 0.96 0.99  1  1  1  0  0  0  0
## 5  0.04 0.07 0.11 0.18 0.28 0.39 0.53 0.69 0.82 0.93 0.98  1  1  0  0  0  0  0
## 6  0.02 0.03 0.06 0.10 0.16 0.25 0.37 0.53 0.70 0.86 0.97  1  0  0  0  0  0  0
## 7  0.01 0.01 0.02 0.04 0.08 0.13 0.22 0.34 0.52 0.74 0.91  0  0  0  0  0  0  0
## 8  0.00 0.00 0.01 0.01 0.03 0.05 0.09 0.17 0.28 0.50 0.00  0  0  0  0  0  0  0
## 9  0.00 0.00 0.00 0.00 0.01 0.01 0.03 0.04 0.11 0.00 0.00  0  0  0  0  0  0  0
## 10 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00  0  0  0  0  0  0  0
## 11 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
## 12 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
## 13 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
## 14 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
## 15 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
## 16 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
## 17 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
```

```r
plot(seq(0,1,by = 0.01), payoff_function(seq(0,1,by = 0.01)),
     main = "Payoff function values per playoff probability",
     xlab = 'playoff probability',
     ylab = 'extra percent boost to yac_wpa')
```



```
get_wr_data <- function(YEARS){

  schedule <- filter(load_schedules(seasons = 2022), game_type == "REG")

  rolling_wins <- matrix(nrow = 32, ncol = 18)

  teams <- sort(unique(schedule$away_team))

  for (i in 1:32){
  cur_team <- teams[i]
    for (j in 1:18){
      rolling_wins[i,j] <- ifelse(j == 1, 0, rolling_wins[i,j-1] )
      tmp <- filter(schedule, (home_team == cur_team) | away_team == cur_team, week == j-1)

      if (nrow(tmp) != 0){
        if (tmp$away_team[1] == cur_team){
            rolling_wins[i,j] <- ifelse(j>1,
                                        (1 * (tmp$result[1] < 0)) + rolling_wins[i,(j-1)],
                                        0)
        }
        else if (tmp$home_team[1] == cur_team){
           rolling_wins[i,j] <- ifelse(j>1,
                                        (1 * (tmp$result[1] > 0)) + rolling_wins[i,(j-1)],
                                        0)
        }

      }
    }
  }

  rolling_wins <- as.data.frame(rolling_wins)
  rownames(rolling_wins) <- teams
  colnames(rolling_wins) <- 1:18


  data = load_pbp(YEARS)
  data <- data %>%
    left_join(load_teams(), by = c('posteam' = 'team_abbr'))

  contracts_with_id <- filter(load_contracts(), is_active == T) %>% inner_join(load_players()[,c('gsis_

  nextgen <- filter(load_nextgen_stats(seasons = YEARS,stat_type = c("receiving"),
                    file_type = getOption("nflreadr.prefer", default = "rds")), week == 0)

  #Assessing WPA by receiver

  passes <- filter(data, pass_attempt == 1, season_type == "REG", !is.na(yards_after_catch))

  i <- cbind(match(passes$posteam, rownames(rolling_wins)),
        match(passes$week, colnames(rolling_wins)))

  passes <- cbind(rolling_win_ct = rolling_wins[i], passes)
```

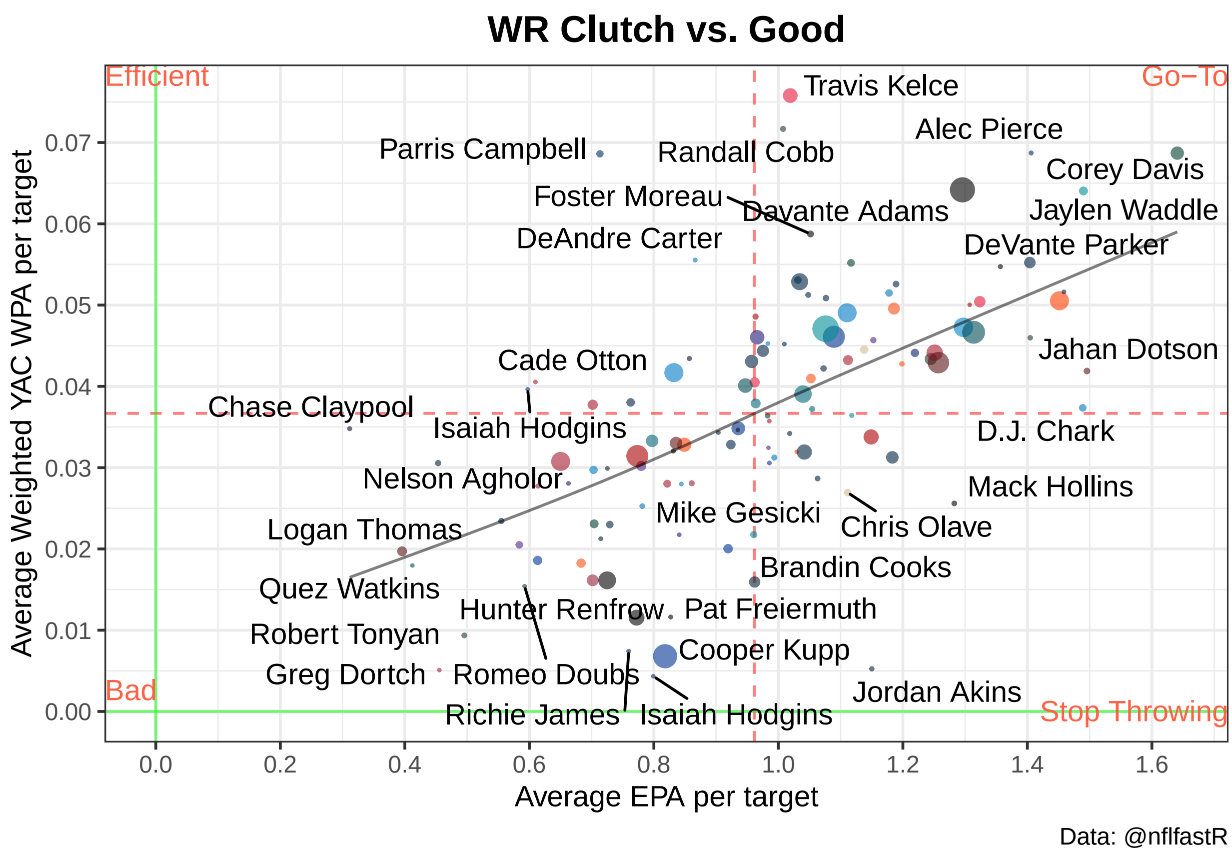
```r
  i <- cbind(match(passes$week - 1 - passes$rolling_win_ct, rownames(playoff_probs)),
        match(passes$rolling_win_ct, colnames(playoff_probs)))

  passes <- cbind(playoff_prob = playoff_probs[i],passes)

  pass_counts <- passes %>% count(receiver_player_name, posteam, receiver_player_id, team_color)

#  avg_wpa_wrs <- setNames(aggregate(wpa ~ receiver_player_name + posteam + receiver_player_id, data = p

  avg_epa_wrs <- aggregate(epa ~ receiver_player_name + posteam + receiver_player_id, data = passes, FUN

#  tot_epa_wrs <- aggregate(epa ~ receiver_player_name + posteam + receiver_player_id, data = passes, FU

#  colnames(tot_epa_wrs)[4] = "tot_epa"

#  avg_cpoe_wrs <- aggregate(cpoe ~ receiver_player_name + posteam + receiver_player_id, data = passes,

  passes$w_yac_wpa <- passes$yac_wpa * (1+payoff_function(passes$playoff_prob))


  avg_w_yacwpa_wrs <- setNames(aggregate(w_yac_wpa ~ receiver_player_name + posteam + receiver_player_id

  avg_uw_yacwpa_wrs <- setNames(aggregate(yac_wpa ~ receiver_player_name + posteam + receiver_player_id

#  avg_yac_wrs <- aggregate(yards_after_catch ~ receiver_player_name + posteam + receiver_player_id, dat

#  wr_data <- avg_wpa_wrs %>% merge(pass_counts, keep=F) %>% merge(avg_epa_wrs, keep=F) %>% merge(avg_cp

    wr_data <- avg_epa_wrs %>% merge(pass_counts, keep=F) %>% merge(avg_w_yacwpa_wrs, keep=F) %>% merge

  team_pa <- aggregate(n~posteam, data = wr_data, FUN=sum)

  wr_data <- wr_data %>% merge(team_pa, by='posteam', suffixes = c("_player", "_team"))

  wr_data <- wr_data %>% inner_join(contracts_with_id, by = join_by(receiver_player_id == gsis_id))

  #wr_data <- filter(wr_data, n_player>50)

  ng_wr_clean <- wr_data %>% inner_join(nextgen, by = join_by(receiver_player_id==player_gsis_id))

  return(ng_wr_clean)
}

data_2022 = get_wr_data(2022)
data_2021 = get_wr_data(2021)
data_2020 = get_wr_data(2020)
```

```r
df <- data.frame(x2=rnorm(100),y2=rnorm(100))

annotations <- data.frame(
        xpos = c(-Inf,-Inf,Inf,Inf),
        ypos =  c(-Inf, Inf,-Inf,Inf),
        annotateText = c("Bad","Efficient"
```

```r
                         ,"Stop Throwing","Go-To"),
        hjustvar = c(0,0,1,1) ,
        vjustvar = c(-2,1,-1,1)) #<- adjust

data_2022 %>%
  ggplot(aes(y = w_yac_wpa, x = epa)) +
  #horizontal line with mean EPA
  geom_hline(yintercept = mean(data_2022$w_yac_wpa), color = "red", linetype = "dashed", alpha=0.5) +
  #vertical line with mean CPOE
  geom_vline(xintercept =  mean(data_2022$epa), color = "red", linetype = "dashed", alpha=0.5) +
  geom_hline(yintercept = 0, color = "green", linetype = "solid", alpha=0.5) +
  #vertical line with mean CPOE
  geom_vline(xintercept =  0, color = "green", linetype = "solid", alpha=0.5) +
  #add points for the QBs with the right colors
  #cex controls point size and alpha the transparency (alpha = 1 is normal)
  geom_point(color = data_2022$team_color, cex=data_2022$apy / mean(data_2022$apy), alpha = .6) +
  #add names using ggrepel, which tries to make them not overlap
  geom_text_repel(aes(label=player), max.overlaps = 12) +
  #add a smooth line fitting wpa + epa
  stat_smooth(geom='line', alpha=0.5, se=FALSE, method='gam')+
  #titles and caption
  labs(x = "Average EPA per target",
       y = "Average Weighted YAC WPA per target",
       title = "WR Clutch vs. Good",
       caption = "Data: @nflfastR") +
  #uses the black and white ggplot theme
  theme_bw() +
  #center title with hjust = 0.5
  theme(
    plot.title = element_text(size = 14, hjust = 0.5, face = "bold")
  ) +
  #make ticks look nice
  #if this doesn't work, `install.packages('scales')`
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) +
  geom_text(data=annotations,aes(x=xpos,y=ypos,hjust=hjustvar,vjust=vjustvar,label=annotateText), color
```

```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```

```
## Warning: ggrepel: 87 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
data_2022$clutch <- scale(data_2022$w_yac_wpa) +
  scale(data_2022$avg_separation)

data_2021$clutch <- scale(data_2021$w_yac_wpa) +
  scale(data_2021$avg_separation)

data_2020$clutch <- scale(data_2020$w_yac_wpa) +
  scale(data_2020$avg_separation)

cor(data_2022[, colnames(data_2022) %in% c("w_yac_wpa", "avg_separation")])
```

```
##                  w_yac_wpa avg_separation
## w_yac_wpa        1.0000000     -0.2975064
## avg_separation -0.2975064      1.0000000
```

```
dta <- data_2022 %>%
  inner_join(data_2021, by = join_by(receiver_player_id), suffix = c("_22", "_21")) %>%
  inner_join(data_2020, by = join_by(receiver_player_id), suffix = c("", "_20"))
```

```
## Warning in inner_join(., data_2021, by = join_by(receiver_player_id), suffix = c("_22", : Detected a
## i Row 2 of `x` matches multiple rows in `y`.
## i Row 36 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Warning in inner_join(., data_2020, by = join_by(receiver_player_id), suffix = c("", : Detected an u
## i Row 70 of `x` matches multiple rows in `y`.
## i Row 79 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
cor(dta$clutch_22, dta$clutch_21)
```

```
##           [,1]
## [1,] 0.4298733
```

```
cor(dta$clutch_21, dta$clutch)
```

```
##           [,1]
## [1,] 0.3531231
```

```
cor(dta$clutch_22, dta$clutch)
```

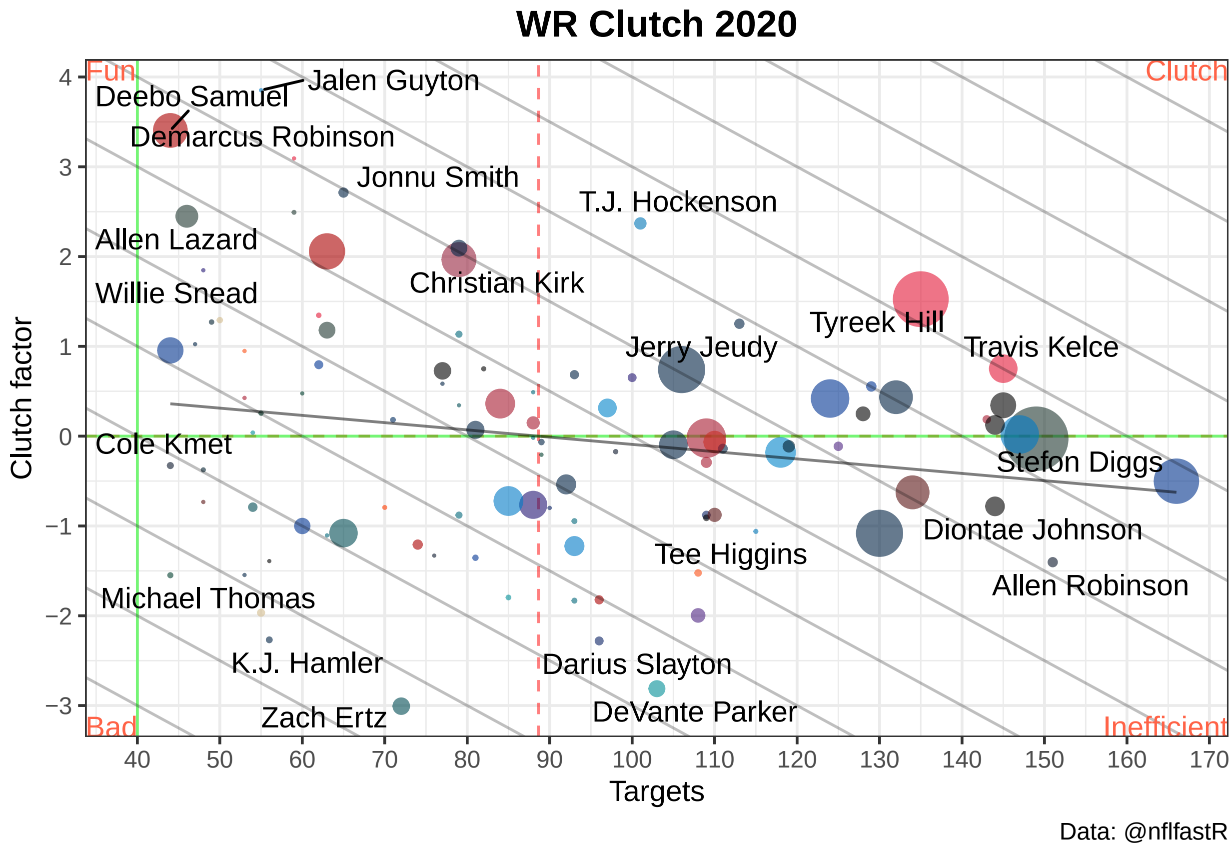
```
##          [,1]
## [1,] 0.326287
```

```
df <- data.frame(x2=rnorm(100),y2=rnorm(100))

annotations <- data.frame(
        xpos = c(-Inf,-Inf,Inf,Inf),
        ypos =  c(-Inf, Inf,-Inf,Inf),
        annotateText = c("Bad","Fun"
                        ,"Inefficient","Clutch"),
        hjustvar = c(0,0,1,1) ,
        vjustvar = c(0,1,0,1)) #<- adjust

plot_clutch <- function(dataframe, yr){
  dataframe %>%
    ggplot(aes(y = clutch, x = targets)) +
    #horizontal line with mean EPA
    geom_hline(yintercept = mean(dataframe$clutch), color = "red", linetype = "dashed", alpha=0.5) +
    #vertical line with mean CPOE
    geom_vline(xintercept =  mean(dataframe$targets), color = "red", linetype = "dashed", alpha=0.5) +
    geom_hline(yintercept = 0, color = "green", linetype = "solid", alpha=0.5) +
    #vertical line with mean CPOE
    geom_vline(xintercept =  40, color = "green", linetype = "solid", alpha=0.5) +
    #add points for the QBs with the right colors
    #cex controls point size and alpha the transparency (alpha = 1 is normal)
    geom_point(color = dataframe$team_color, cex=dataframe$value / median(dataframe$value), alpha = .6)
    #add names using ggrepel, which tries to make them not overlap
    geom_text_repel(aes(label=player), max.overlaps = 5) +
    #add a smooth line fitting wpa + epa
    stat_smooth(geom='line', alpha=0.5, se=FALSE, method='lm')+
    geom_abline(slope = -.05, intercept = (15:-5), alpha = .25)+
    #titles and caption
    labs(x = "Targets",
```

```
          y = "Clutch factor",
          title = paste("WR Clutch", yr),
          caption = "Data: @nflfastR") +
     #uses the black and white ggplot theme
     theme_bw() +
     #center title with hjust = 0.5
     theme(
       plot.title = element_text(size = 14, hjust = 0.5, face = "bold")
     ) +
     #make ticks look nice
     #if this doesn't work, `install.packages('scales')`
     scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
     scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) +
     geom_text(data=annotations,aes(x=xpos,y=ypos,hjust=hjustvar,vjust=vjustvar,label=annotateText), col
}

plot_clutch(data_2020, "2020")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

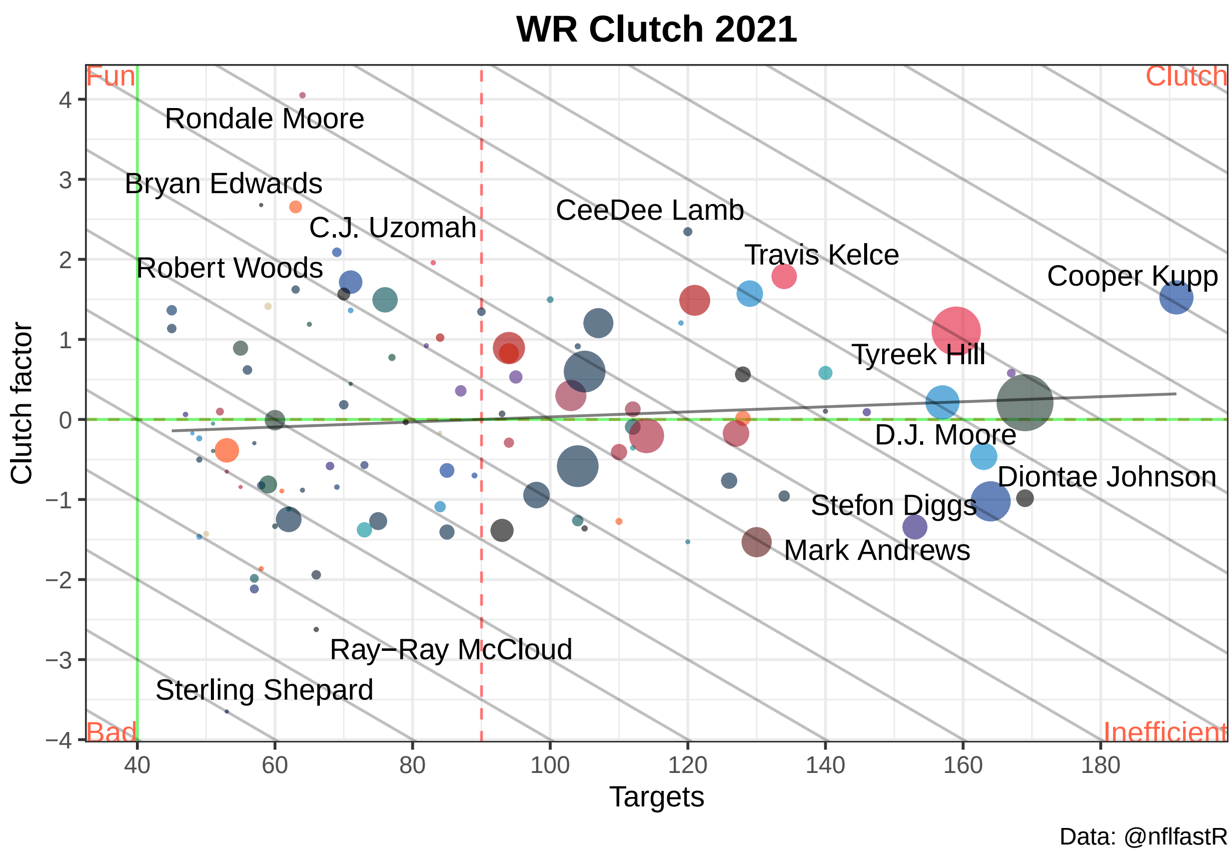
```
## Warning: ggrepel: 80 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```r
plot_clutch(data_2021, "2021")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

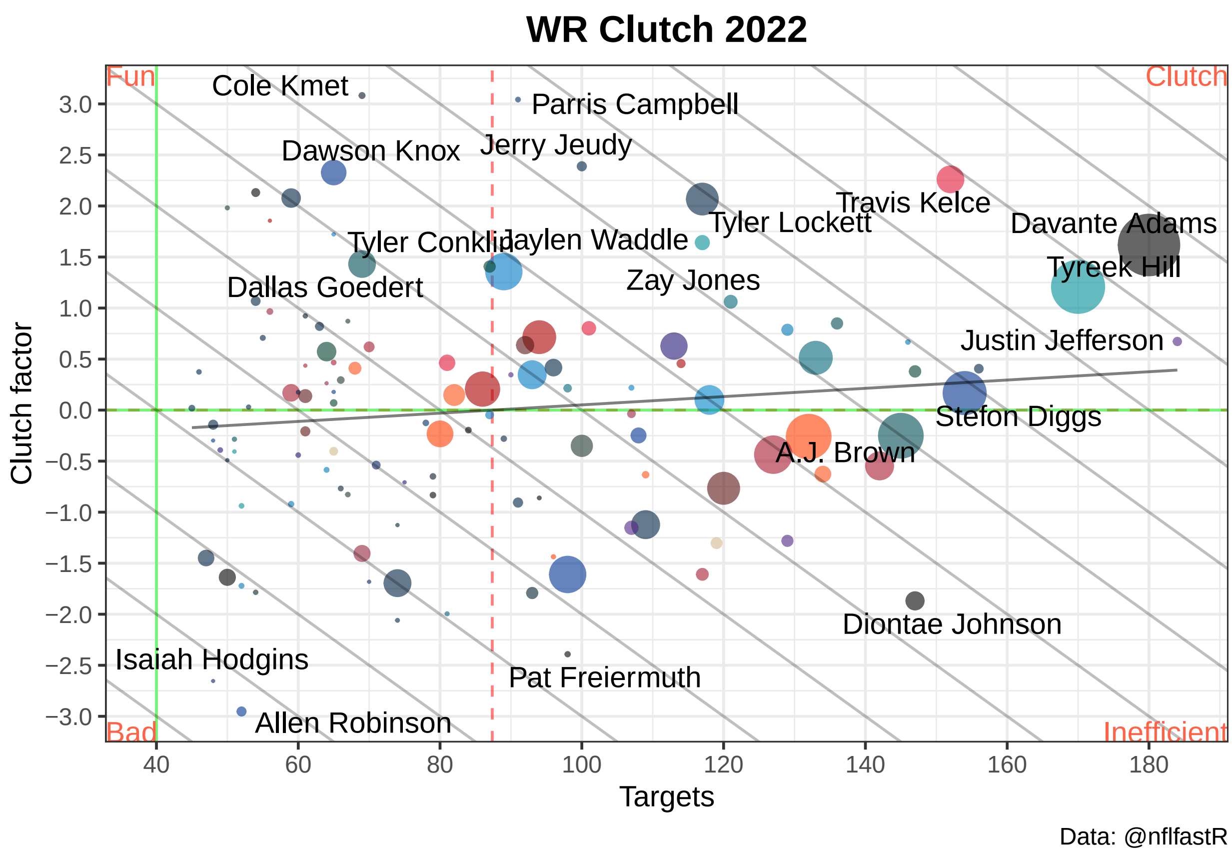
```
## Warning: ggrepel: 92 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



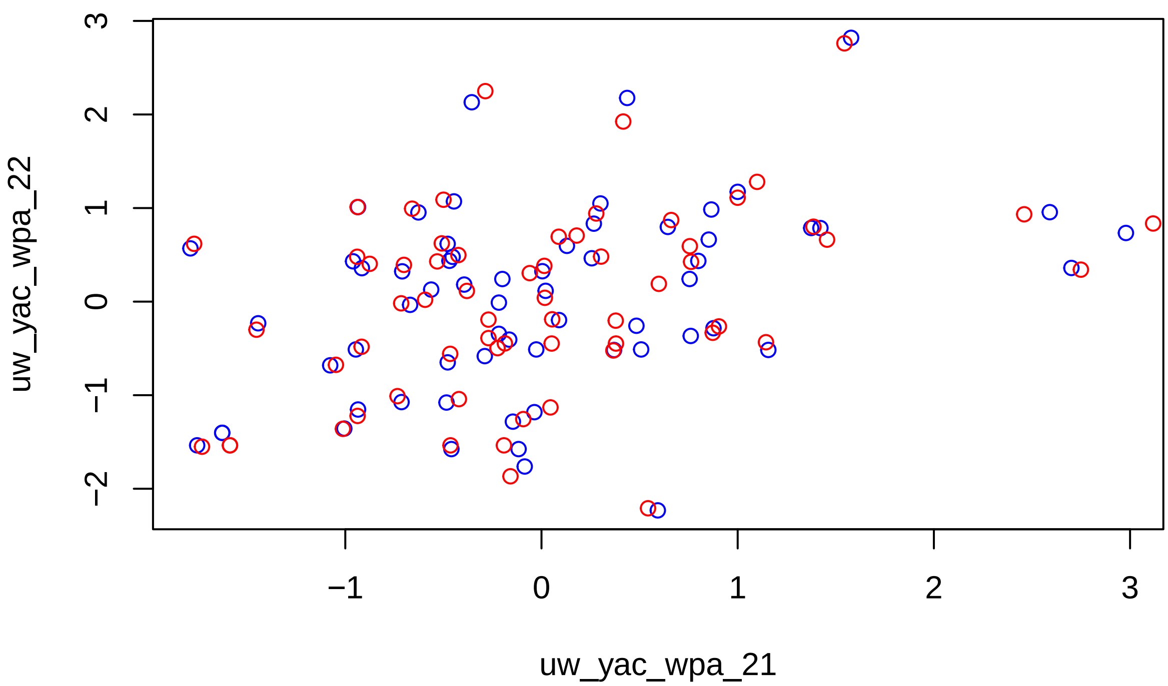

```r
plot_clutch(data_2022, "2022")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: ggrepel: 99 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
dta[,colnames(dta) %in% c("w_yac_wpa_22", "uw_yac_wpa_22",
                          "uw_yac_wpa_21", "w_yac_wpa_21",
                          "uw_yac_wpa", "w_yac_wpa")] <-scale(dta[,colnames(dta) %in% c("w_yac_wpa_
                          "uw_yac_wpa_21", "w_yac_wpa_21",
                          "uw_yac_wpa", "w_yac_wpa")])
plot(uw_yac_wpa_22~uw_yac_wpa_21, dta, col = 'blue')
points(w_yac_wpa_22~w_yac_wpa_21, dta, col = 'red')
```



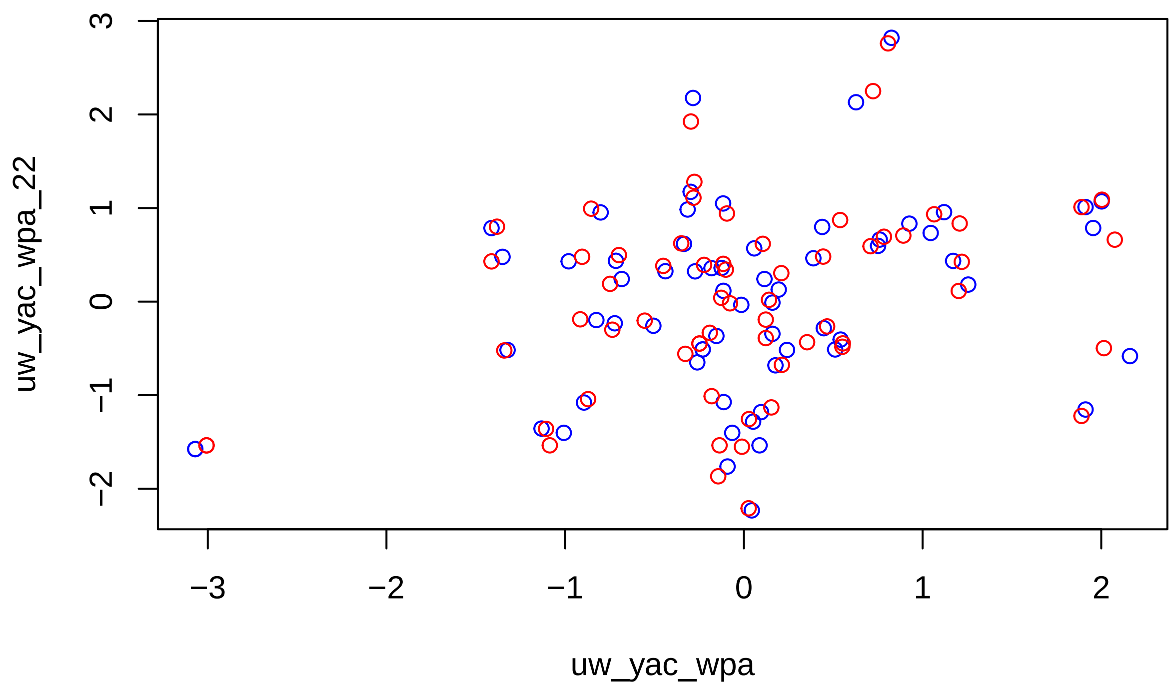

```
cor(dta$uw_yac_wpa_22,dta$uw_yac_wpa_21)
```

```
## [1] 0.377042
```

```
cor(dta$w_yac_wpa_22,dta$w_yac_wpa_21)
```

```
## [1] 0.3856426
```

```
plot(uw_yac_wpa_22~uw_yac_wpa, dta, col = 'blue')
points(w_yac_wpa_22~w_yac_wpa, dta, col = 'red')
```



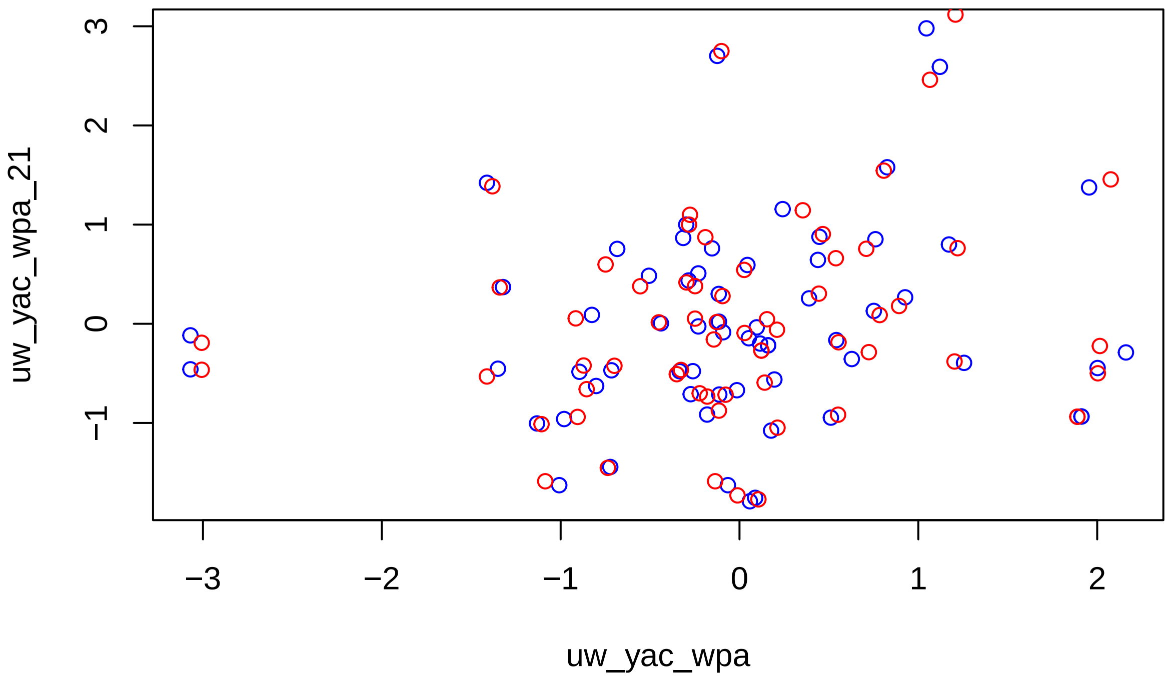

```
cor(dta$uw_yac_wpa_22,dta$uw_yac_wpa)
```

```
## [1] 0.3161489
```

```
cor(dta$w_yac_wpa_22,dta$w_yac_wpa)
```

```
## [1] 0.3241923
```

```
plot(uw_yac_wpa_21~uw_yac_wpa, dta, col = 'blue')
points(w_yac_wpa_21~w_yac_wpa, dta, col = 'red')
```




```
cor(dta$uw_yac_wpa_21,dta$uw_yac_wpa)
```

```
## [1] 0.1710394
```

```
cor(dta$w_yac_wpa_21,dta$w_yac_wpa)
```

```
## [1] 0.1964427
```

```
cor(dta[,colnames(dta) %in% c("w_yac_wpa_22", "uw_yac_wpa_22",
                              "uw_yac_wpa_21", "w_yac_wpa_21",
                              "uw_yac_wpa", "w_yac_wpa")])
```

```
##                w_yac_wpa_22 uw_yac_wpa_22 w_yac_wpa_21 uw_yac_wpa_21 w_yac_wpa
## w_yac_wpa_22      1.0000000     0.9969648    0.3856426     0.3836154 0.3241923
## uw_yac_wpa_22     0.9969648     1.0000000    0.3773010     0.3770420 0.3300367
## w_yac_wpa_21      0.3856426     0.3773010    1.0000000     0.9979166 0.1964427
## uw_yac_wpa_21     0.3836154     0.3770420    0.9979166     1.0000000 0.1876859
## w_yac_wpa         0.3241923     0.3300367    0.1964427     0.1876859 1.0000000
## uw_yac_wpa        0.3086915     0.3161489    0.1781817     0.1710394 0.9982040
##                uw_yac_wpa
## w_yac_wpa_22    0.3086915
## uw_yac_wpa_22   0.3161489
## w_yac_wpa_21    0.1781817
## uw_yac_wpa_21   0.1710394
```

```
## w_yac_wpa      0.9982040
## uw_yac_wpa     1.0000000
```

```r
data_2022[order(data_2022$w_yac_wpa, decreasing = T),
          colnames(data_2022) %in% c("posteam", "receiver_player_name", "uw_yac_wpa", "w_yac_wpa")]
```

```
##     posteam receiver_player_name   w_yac_wpa  uw_yac_wpa
## 54       KC              T.Kelce 0.075780341 0.040050282
## 43       GB               R.Cobb 0.071675410 0.036481713
## 48      IND             A.Pierce 0.068713076 0.035842773
## 90      NYJ              C.Davis 0.068693029 0.035077597
## 49      IND           P.Campbell 0.068608784 0.035619391
## 65       LV              D.Adams 0.064180126 0.035410665
## 72      MIA             J.Waddle 0.064043712 0.032626957
## 68       LV             F.Moreau 0.058747402 0.032921442
## 62      LAC             D.Carter 0.055538361 0.028067544
## 79       NE             D.Parker 0.055230710 0.028164340
## 89      NYJ             G.Wilson 0.055191033 0.028155392
## 97      PIT            G.Pickens 0.054729311 0.030013808
## 92      PHI              D.Smith 0.053076336 0.030024325
## 101     SEA            T.Lockett 0.052876227 0.026801363
## 31      DEN              J.Jeudy 0.052580060 0.027425813
## 111     TEN   N.Westbrook-Ikhine 0.051625569 0.026229139
## 38      DET          T.Hockenson 0.051490903 0.026990587
## 110     TEN             A.Hooper 0.051250161 0.026575795
## 114     TEN              T.Burks 0.050861609 0.026128026
## 24      CLE             A.Cooper 0.050534401 0.027268115
## 55       KC   M.Valdes-Scantling 0.050418799 0.026595196
## 103      SF           J.Jennings 0.050038514 0.025876737
## 22      CIN               T.Boyd 0.049566389 0.025455889
## 61      LAC           M.Williams 0.049052320 0.024990852
## 105      SF              B.Aiyuk 0.048580675 0.025367818
## 16      CAR              D.Moore 0.047263817 0.025708813
## 69      MIA               T.Hill 0.047086778 0.023997488
## 93      PHI              A.Brown 0.046655038 0.025374144
## 13      BUF              S.Diggs 0.046099960 0.024150311
## 8       BAL            M.Andrews 0.046038302 0.023801863
## 40       GB             C.Watson 0.045976037 0.024901541
## 75      MIN          J.Jefferson 0.045685626 0.024480873
## 35      DET          A.St. Brown 0.045265594 0.024584443
## 112     TEN            C.Okonkwo 0.045204465 0.023142733
## 83       NO            J.Johnson 0.044504905 0.024029082
## 81       NE             J.Meyers 0.044367435 0.022842736
## 109      TB              M.Evans 0.044143764 0.023039330
## 85      NYG            D.Slayton 0.044115854 0.022803108
## 18      CHI               C.Kmet 0.043427049 0.023147582
## 39       GB             A.Lazard 0.043367261 0.022834673
## 6       ATL              K.Pitts 0.043232856 0.022168120
## 78       NE              H.Henry 0.043066830 0.022276414
## 116     WAS           T.McLaurin 0.042912446 0.022029069
## 23      CIN            T.Higgins 0.042767070 0.022038030
## 30      DAL               C.Lamb 0.042204291 0.022290152
## 117     WAS             J.Dotson 0.041886259 0.021655034
## 60      LAC              K.Allen 0.041688681 0.021436719
```

```
## 21      CIN            J.Chase 0.040966123 0.021012310
## 108      TB            C.Otton 0.040547354 0.021207888
## 56       KC  J.Smith-Schuster 0.040495379 0.020567149
## 94      PHI          D.Goedert 0.040097718 0.021438262
## 15      BUF          I.Hodgins 0.039640781 0.020110995
## 51      JAX             C.Kirk 0.039046103 0.021008404
## 28      DAL          D.Schultz 0.038026654 0.020525731
## 52      JAX            Z.Jones 0.037903931 0.020385955
## 107      TB             R.Gage 0.037734619 0.020623086
## 34      DET            D.Chark 0.037357755 0.020313722
## 50      JAX            M.Jones 0.037207731 0.019443427
## 70      MIA        T.Sherfield 0.036427698 0.019021437
## 91      NYJ            E.Moore 0.036383076 0.018333645
## 5       ATL        O.Zaccheaus 0.035726703 0.018417278
## 12      BUF             D.Knox 0.034834889 0.018269436
## 19      CHI         C.Claypool 0.034794024 0.019615779
## 17      CHI           D.Mooney 0.034638139 0.017829096
## 32      DEN          G.Dulcich 0.034319627 0.018746131
## 80       NE         T.Thornton 0.034203365 0.018060800
## 104      SF           G.Kittle 0.033780531 0.017640199
## 53      JAX           E.Engram 0.033293554 0.018021794
## 118     WAS           C.Samuel 0.033022867 0.016849864
## 113     TEN            R.Woods 0.032836080 0.017044266
## 25      CLE            D.Njoku 0.032814924 0.017053223
## 74      MIN           K.Osborn 0.032441360 0.016950911
## 96      PIT         C.Claypool 0.032049404 0.017207257
## 26      CLE    D.Peoples-Jones 0.031951776 0.016936253
## 33      DEN           C.Sutton 0.031921587 0.016578492
## 102      SF           D.Samuel 0.031434396 0.015963379
## 29      DAL           M.Gallup 0.031262144 0.016762225
## 36      DET         J.Reynolds 0.031240765 0.016000475
## 106      TB           C.Godwin 0.030765595 0.016001683
## 14      BUF         I.McKenzie 0.030575401 0.016025769
## 77       NE          N.Agholor 0.030558292 0.015488126
## 73      MIN          A.Thielen 0.030214195 0.015957976
## 46      HOU          N.Collins 0.029901702 0.015977544
## 64      LAC          G.Everett 0.029713876 0.015004992
## 27      DAL            N.Brown 0.028663300 0.014837158
## 3       ARI            M.Brown 0.028084694 0.014775465
## 11      BAL         D.Robinson 0.028060709 0.014238558
## 7       ATL           D.London 0.028020912 0.015118666
## 63      LAC           J.Palmer 0.027980122 0.014123889
## 1       ARI            R.Moore 0.027730889 0.014334658
## 10      BAL           I.Likely 0.027038717 0.013740274
## 84       NO            C.Olave 0.026935262 0.014402680
## 66       LV          M.Hollins 0.025599198 0.013776494
## 37      DET          K.Raymond 0.025266716 0.014193874
## 100     SEA             N.Fant 0.023435746 0.011940699
## 88      NYJ          T.Conklin 0.023091515 0.012003207
## 82       NE           K.Bourne 0.022993756 0.011904751
## 71      MIA          M.Gesicki 0.021764179 0.011163257
## 9       BAL         D.Duvernay 0.021739621 0.010951339
## 44      HOU            C.Moore 0.021258489 0.011964464
## 76      MIN        T.Hockenson 0.020501005 0.011367046
```

```
## 58       LA          Al.Robinson 0.020030671 0.010429893
## 115     WAS             L.Thomas 0.019710850 0.010227158
## 57       LA             T.Higbee 0.018584422 0.009895789
## 20      CIN              H.Hurst 0.018249069 0.009336949
## 95      PHI            Q.Watkins 0.017964954 0.009794641
## 98      PIT            D.Johnson 0.016138316 0.009566317
## 2       ARI               Z.Ertz 0.016136353 0.008313572
## 47      HOU              B.Cooks 0.015932894 0.008601829
## 41       GB              R.Doubs 0.015402348 0.008091228
## 99      PIT         P.Freiermuth 0.011621105 0.006929647
## 67       LV            H.Renfrow 0.011543992 0.006967857
## 42       GB             R.Tonyan 0.009358933 0.004892449
## 86      NYG              R.James 0.007414540 0.003688882
## 59       LA               C.Kupp 0.006799223 0.003583528
## 45      HOU              J.Akins 0.005243501 0.003172899
## 4       ARI             G.Dortch 0.005091509 0.002498867
## 87      NYG            I.Hodgins 0.004332973 0.002137981
```

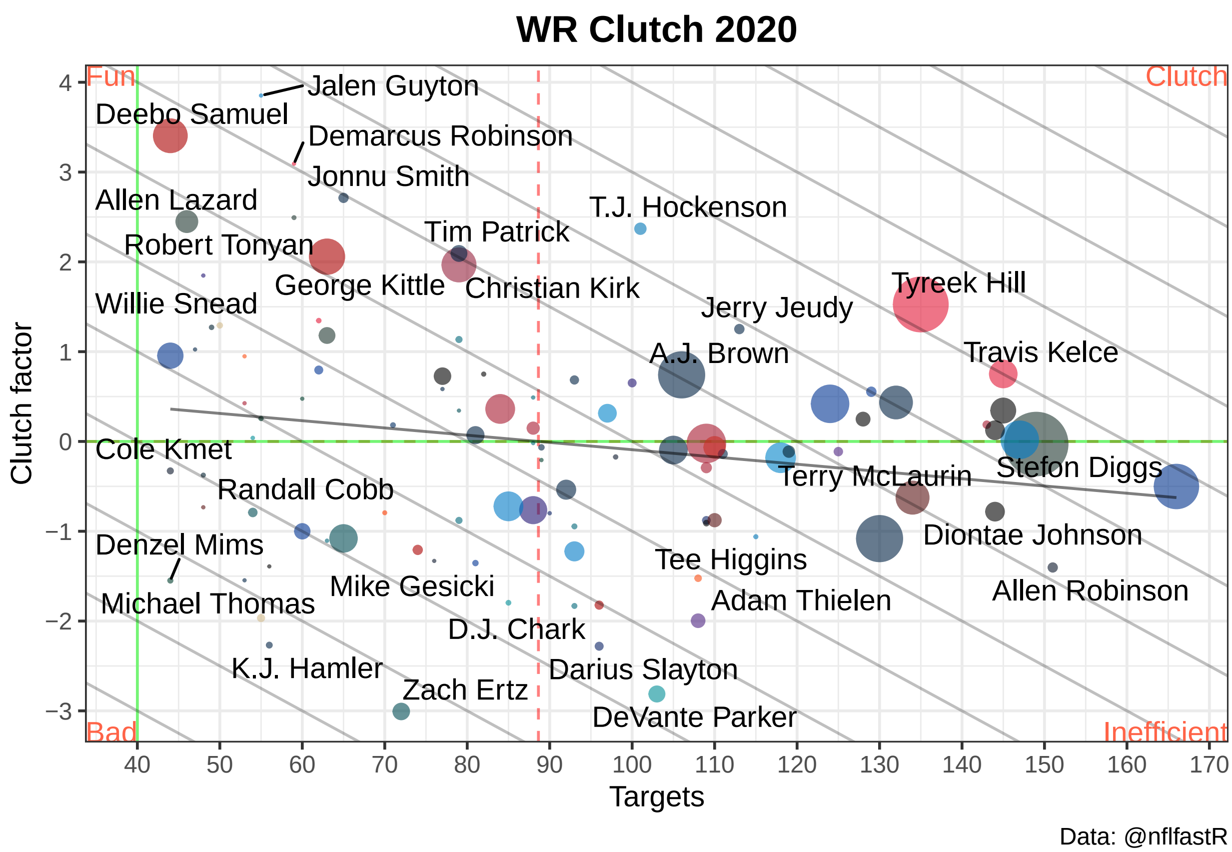
```
dta <- clutch_2022 %>%
  inner_join(clutch_2021, by = join_by(receiver_player_id), suffix = c("_22", "_21")) %>%
  inner_join(clutch_2020, by = join_by(receiver_player_id), suffix = c("", "_20"))

plot(clutch_22~clutch_21, dta)

cor(dta$clutch_22, dta$clutch_21)

plot(clutch_21~clutch, dta)

cor(dta$clutch_21, dta$clutch)

plot(clutch_22~clutch, dta)

cor(dta$clutch_22, dta$clutch)
```

```
df <- data.frame(x2=rnorm(100),y2=rnorm(100))

annotations <- data.frame(
        xpos = c(-Inf,-Inf,Inf,Inf),
        ypos =  c(-Inf, Inf,-Inf,Inf),
        annotateText = c("Bad","Fun"
                        ,"Inefficient","Clutch"),
        hjustvar = c(0,0,1,1) ,
        vjustvar = c(0,1,0,1)) #<- adjust

plot_clutch <- function(dataframe, yr){
  dataframe %>%
    ggplot(aes(y = clutch, x = targets)) +
    #horizontal line with mean EPA
    geom_hline(yintercept = mean(dataframe$clutch), color = "red", linetype = "dashed", alpha=0.5) +
    #vertical line with mean CPOE
    geom_vline(xintercept =  mean(dataframe$targets), color = "red", linetype = "dashed", alpha=0.5) +
    geom_hline(yintercept = 0, color = "green", linetype = "solid", alpha=0.5) +
    #vertical line with mean CPOE
    geom_vline(xintercept =  40, color = "green", linetype = "solid", alpha=0.5) +
```

```r
    #add points for the QBs with the right colors
    #cex controls point size and alpha the transparency (alpha = 1 is normal)
    geom_point(color = dataframe$team_color, cex=dataframe$value / median(dataframe$value), alpha = .6)
    #add names using ggrepel, which tries to make them not overlap
    geom_text_repel(aes(label=player), max.overlaps = 7) +
    #add a smooth line fitting wpa + epa
    stat_smooth(geom='line', alpha=0.5, se=FALSE, method='lm')+
    geom_abline(slope = -.05, intercept = (15:-5), alpha = .25)+
    #titles and caption
    labs(x = "Targets",
         y = "Clutch factor",
         title = paste("WR Clutch", yr),
         caption = "Data: @nflfastR") +
    #uses the black and white ggplot theme
    theme_bw() +
    #center title with hjust = 0.5
    theme(
      plot.title = element_text(size = 14, hjust = 0.5, face = "bold")
    ) +
    #make ticks look nice
    #if this doesn't work, `install.packages('scales')`
    scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) +
    geom_text(data=annotations,aes(x=xpos,y=ypos,hjust=hjustvar,vjust=vjustvar,label=annotateText), col
}

plot_clutch(data_2020, "2020")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

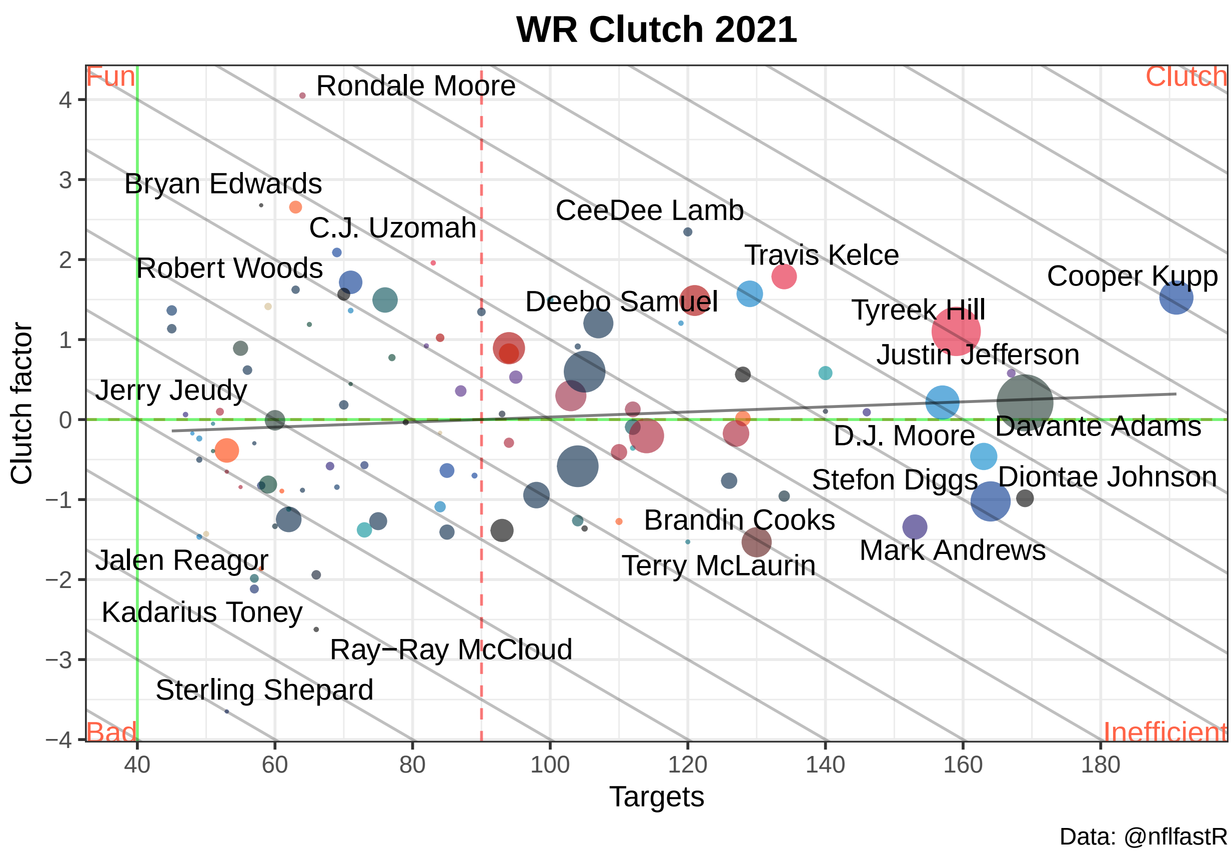
```
## Warning: ggrepel: 70 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
plot_clutch(data_2021, "2021")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

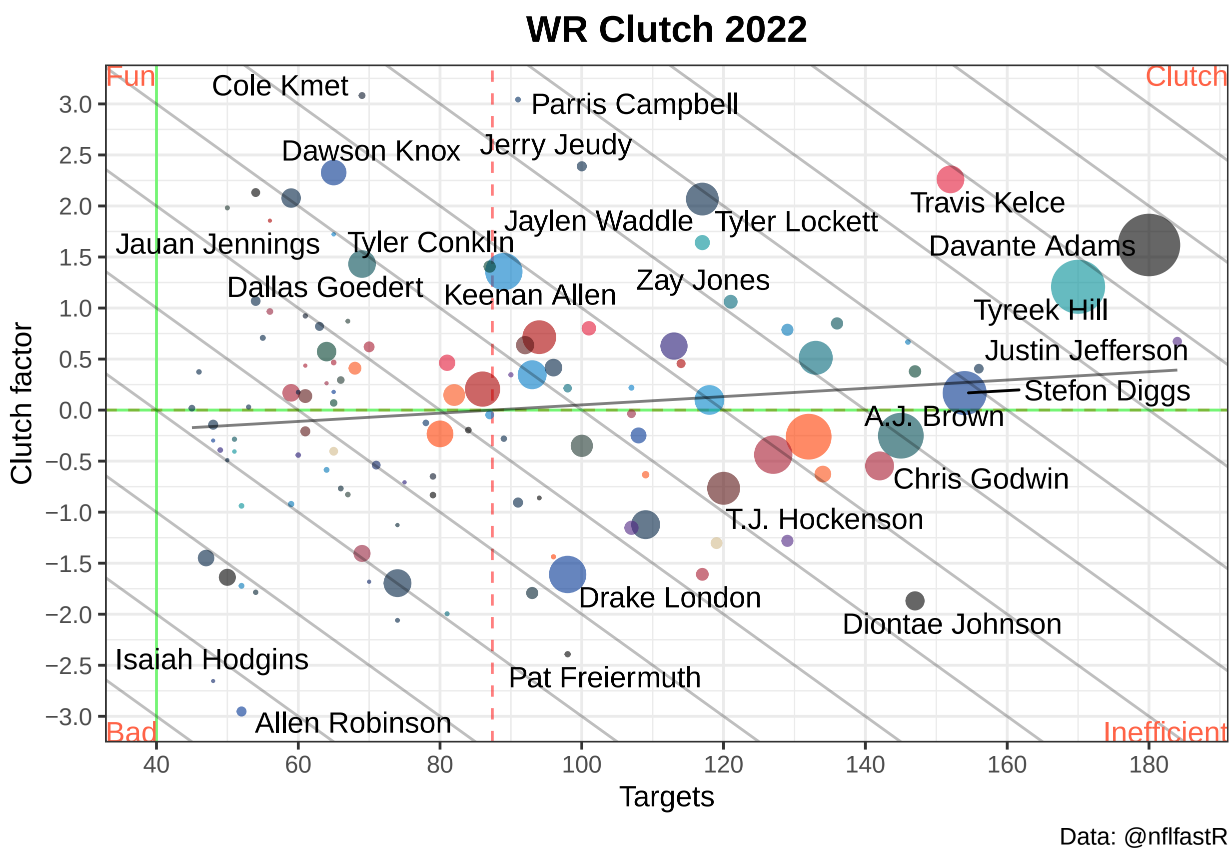
```
## Warning: ggrepel: 84 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```r
plot_clutch(data_2022, "2022")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: ggrepel: 94 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



WR Clutch 2022
Fun
Clutch
Bad
Inefficient
Cole Kmet
Parris Campbell
Dawson Knox
Jerry Jeudy
Travis Kelce
Jaylen Waddle
Tyler Lockett
Jauan Jennings
Tyler Conklin
Davante Adams
Dallas Goedert
Keenan Allen
Zay Jones
Tyreek Hill
Justin Jefferson
Stefon Diggs
A.J. Brown
Chris Godwin
T.J. Hockenson
Drake London
Diontae Johnson
Isaiah Hodgins
Pat Freiermuth
Allen Robinson
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
40
60
80
100
120
140
160
180
Clutch factor
Targets
Data: @nflfastR