Analysis of tennis players' playstyle and suggestion of countermeasures for each type

Minyong Lee

11/20/2022

Phase 1 : Problem Definition

Area Tennis

Project Title Analysis of tennis players' playstyle and suggestion of countermeasures for each type

Data sources

- tennisabstract.com
- atptour.com

Issues addressed Professional tennis players play the game in different ways. Each has different advantages and disadvantages, and they must adjust their play to defeat the opponent's play style. The purpose of this project is by using the game data of tennis players from the past to the present, to classify players into several types based on similarities and differences and analyze their play styles and pros and cons for each type. If applied to the actual industry, the project can be used in part to propose competition strategies according to the type of matchup opponent and predict the points and the results of the game.

Phase 2 : Data Collection

I got several data sets about 3,600 tennis matches from 1960 to 2022 and detailed stats of those form tennisabstract.com. These data sets are results of Match Charting Project (MCP). MCP match records contain shot-by-shot data for every point of a match and some overview information. Among many data about tennis matches, I have selected several resources for this term project, and those listed below are the ones. All these data are for men's tennis matches and will be selected and utilized according to the detailed analysis model during the project.

matches This data set is about match basic information. There are features like date, tournament, player and surface of court. It has 3,608 observations from 1960 to 2022 tennis matches.

points This data is about every point in each game. Information about Set score, game score, who the server was, rally count, how the rally was processed are included. The rally process is recorded as some letter symbols like "5r28f1x@" or "4b2n#". Each letter represents a behavior in tennis like forehand, backhand or slice. This set has 619,647 observations.

overview This data has observations about each set's overview. How many serves were in specific set? How many aces, return or unforced shot were there? You can answer these questions through this data set.

shot types This data is about how many times a player hit each type of shot (forehand, backhand, slice, drop, volley etc.) throughout the game and some detail results of the shots.

others (serve basics, return depth, net points, serve & volley) There are some data sets for specific tennis plays like serve, return, net play and serve & volley. Each data set has observations for specific numerical information of each type of play. Through these data sets, I will analyze the tendency of each player's play type and classify it into several groups.

Phase 3 : Data Cleaning & Preprocessing

At the beginning, there were enough data that that I can find, but those were scattered in too many files. I decided to reorganize given data sets into four main data sets like below. So I reorganized the raw data to the form that I needed, and data cleaning were done during the process.

- Dataset for matches
- Dataset for rallies, or each point
- Dataset for serves
- Dataset for shots taken by each player

Data 1 - data_matches

This dataset has the basic information about the tennis matches. It consists of columns of match_id, date, player1, player2, tournament, court, surface

```
# check summary and if there are NA values in the data
summary(data_matches)
```

```
##    match_id              date             player1            player2
##  Length:3608        Min.   :19600529   Length:3608        Length:3608
##  Class :character   1st Qu.:20050412   Class :character   Class :character
##  Mode  :character   Median :20150528   Mode  :character   Mode  :character
##                     Mean   :20109571
##                     3rd Qu.:20190510
##                     Max.   :20220713
##   tournament           court             surface
##  Length:3608        Length:3608        Length:3608
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
names(which(colSums(is.na(data_matches)) > 0))
```

```
## character(0)
```

Preprocessing 1 The column "date" was given as number type in the raw data set. I'm planning to use the date data in my project, so I convert the number type value (yyyyMMdd) to the date value(yyyy-MM-dd) by using some character methods and as.Date function.

```r
### Preprocessing 1
date_num <- data_matches$date
date_str <- str_glue("{substr(date_num, 1, 4)}-{substr(date_num, 5, 6)}-{substr(date_num, 7, 8)}")
data_matches$date <- as.Date(date_str)

head(data_matches)
```

```
##                                                        match_id       date
## 1                20220713-M-Newport-R16-Andy_Murray-Max_Purcell 2022-07-13
## 2                20220712-M-Newport-R32-Andy_Murray-Sam_Querrey 2022-07-12
## 3           20220712-M-Bastad-R32-Emil_Ruusuvuori-Dominic_Thiem 2022-07-12
## 4           20220706-M-Wimbledon-QF-Nick_Kyrgios-Cristian_Garin 2022-07-06
## 5 20220704-M-Wimbledon-R16-Rafael_Nadal-Botic_Van_De_Zandschulp 2022-07-04
## 6     20220703-M-Wimbledon-R16-Novak_Djokovic-Tim_Van_Rijthoven 2022-07-03
##           player1                player2 tournament        court surface
## 1     Andy Murray            Max Purcell    Newport       Center   Grass
## 2     Andy Murray            Sam Querrey    Newport      Stadium   Grass
## 3 Emil Ruusuvuori          Dominic Thiem     Bastad       Center    Clay
## 4    Nick Kyrgios         Cristian Garin  Wimbledon            1   Grass
## 5    Rafael Nadal Botic Van De Zandschulp  Wimbledon Centre Court   Grass
## 6  Novak Djokovic       Tim Van Rijthoven  Wimbledon Centre Court   Grass
```

Data 2 - data_points

This dataset has the basic information about the tennis matches. It consists of columns of match_id, seq, set1, set2, gm1, gm2, points, gm, svr, x1st, x2nd, winner, isSvrWinner, rallyCount

```r
# check summary and if there are NA values in the data
summary(data_points)
```

```
##    match_id              seq               set1            set2       
##  Length:619647      Min.   :  1.00   Min.   :0.000   Min.   :0.0000  
##  Class :character   1st Qu.: 43.00   1st Qu.:0.000   1st Qu.:0.0000  
##  Mode  :character   Median : 87.00   Median :0.000   Median :0.0000  
##                     Mean   : 99.93   Mean   :0.542   Mean   :0.5124  
##                     3rd Qu.:141.00   3rd Qu.:1.000   3rd Qu.:1.0000  
##                     Max.   :980.00   Max.   :2.000   Max.   :2.0000  
##                                                                       
##       gm1              gm2              points               gm           
##  Min.   : 0.000   Min.   : 0.000   Length:619647      Length:619647     
##  1st Qu.: 1.000   1st Qu.: 1.000   Class :character   Class :character  
##  Median : 2.000   Median : 2.000   Mode  :character   Mode  :character  
##  Mean   : 2.489   Mean   : 2.384                                         
##  3rd Qu.: 4.000   3rd Qu.: 4.000                                         
##  Max.   :68.000   Max.   :69.000                                         
##  NA's   :1        NA's   :1                                              
##       svr             x1st               x2nd               winner     
##  Min.   :1.000   Length:619647      Length:619647      Min.   :1.000  
##  1st Qu.:1.000   Class :character   Class :character   1st Qu.:1.000  
##  Median :1.000   Mode  :character   Mode  :character   Median :1.000  
##  Mean   :1.495                                         Mean   :1.497  
##  3rd Qu.:2.000                                         3rd Qu.:2.000  
```

```
##  Max.   :2.000                                          Max.   :2.000
##
##   isSvrWinner      rallyCount
##  Min.   :0.0000   Length:619647
##  1st Qu.:0.0000   Class :character
##  Median :1.0000   Mode  :character
##  Mean   :0.6403
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

```
names(which(colSums(is.na(data_points)) > 0))
```

```
## [1] "gm1" "gm2"
```

Preprocessing 2 There were some observations that has NA values in column gm1 or gm2. In this dataset, those observations are not that necessary so I just removed the observations with NA.

```
### Preprocessing 2
data_points <- data_points %>%
  filter(!is.na(gm1) & !is.na(gm2))

names(which(colSums(is.na(data_points)) > 0))
```

```
## character(0)
```

Preprocessing 3 At first, column "gm" was containing two information (game # and seq in game) as a string value using parenthesis. (like "2 (3)" which means it is third shot in second game) So I divided column "gm" into two number type columns "gm" and "seqInGm" using character methods.

```
### Preprocessing 3
gm_split <- str_split_fixed(data_points$gm, " ", 2)

data_points <- data_points %>%
  mutate(
    seqInGm = substr(gm_split[, 2], 2, nchar(gm_split[, 2]) - 1),
    gm = gm_split[, 1]
    )
```

Preprocessing 4 The column "rallyCount" was character value, so I had to convert it to integer value.

```
### Preprocessing 4
data_points <- data_points %>%
  mutate(rallyCount = as.integer(rallyCount)) %>%
  filter(!is.na(rallyCount))
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

Data 3 - data_serves

This dataset has informations about serves like the number of serve, aces and won points, the direction of serves and Serve & Volley. It consists of columns of match_id, player, pts, ptsWon, aces, ptsWonLte3Shots, wide, body, t, snvPts, snvRatio, snvPtsWnRatio.

Preprocessing 5 I wanted to bind the information about Serve & Volley to the basic data set about serve. So I reorganized the dataset about Serve & Volley and using left join function combined them into one data frame about serve. After doing left join, some NA values occured, so I handled them in a proper way. In this case, NA was meaning that SnV was not tried. Thus, replacing them with 0 was the most proper way.

```
snv <- s_n_v %>%
  select(match_id, player, row, snv_pts, pts_won) %>%
  filter(row == "SnV")

non_snv <- s_n_v %>%
  select(match_id, player, row, snv_pts, pts_won) %>%
  filter(row == "nonSnV")

data_snv <- left_join(snv, non_snv, by = c("match_id", "player")) %>%
  select(match_id, player, snv_pts.x, pts_won.x, snv_pts.y, pts_won.y) %>%
  rename(snvPts = snv_pts.x, snvPtsWon = pts_won.x, nonSnvPts = snv_pts.y, nonSnvPtsWon = pts_won.y)

data_serves <- data_serves %>%
  left_join(data_snv, by = c("match_id", "player"))

# check summary and if there are NA values in the data
summary(data_serves)
```

```
##    match_id            player           pts             ptsWon
##  Length:5236        Min.   :1.000   Min.   :  5.00   Min.   :  0.0
##  Class :character   1st Qu.:1.000   1st Qu.: 59.00   1st Qu.: 37.0
##  Mode  :character   Median :2.000   Median : 78.00   Median : 50.0
##                     Mean   :1.501   Mean   : 85.04   Mean   : 54.3
##                     3rd Qu.:2.000   3rd Qu.:103.00   3rd Qu.: 66.0
##                     Max.   :2.000   Max.   :491.00   Max.   :385.0
##
##       aces         ptsWonLte3Shots       wide            body
##  Min.   :  0.000   Min.   :  0.00   Min.   :  2.0   Min.   :  0.00
##  1st Qu.:  3.000   1st Qu.: 19.00   1st Qu.: 23.0   1st Qu.: 10.00
##  Median :  5.000   Median : 27.00   Median : 32.0   Median : 16.00
##  Mean   :  6.842   Mean   : 29.95   Mean   : 36.3   Mean   : 18.22
##  3rd Qu.:  9.000   3rd Qu.: 38.00   3rd Qu.: 46.0   3rd Qu.: 24.00
##  Max.   :115.000   Max.   :301.00   Max.   :230.0   Max.   :105.00
##
##        t              snvPts          snvPtsWon         nonSnvPts
##  Min.   :  1.00   Min.   :  1.00   Min.   :  0.00   Min.   :  1.0
##  1st Qu.: 19.00   1st Qu.:  2.00   1st Qu.:  1.00   1st Qu.: 48.0
##  Median : 27.00   Median :  4.00   Median :  3.00   Median : 67.0
##  Mean   : 30.45   Mean   : 15.18   Mean   : 10.25   Mean   : 72.4
##  3rd Qu.: 38.00   3rd Qu.: 13.00   3rd Qu.:  8.00   3rd Qu.: 93.0
##  Max.   :217.00   Max.   :170.00   Max.   :115.00   Max.   :436.0
##                   NA's   :2135     NA's   :2135     NA's   :2153
```

```
##   nonSnvPtsWon
##  Min.   :  0.00
##  1st Qu.: 32.00
##  Median : 45.00
##  Mean   : 47.87
##  3rd Qu.: 61.00
##  Max.   :345.00
##  NA's   :2153
```

```
names(which(colSums(is.na(data_serves)) > 0))
```

```
## [1] "snvPts"       "snvPtsWon"    "nonSnvPts"    "nonSnvPtsWon"
```

```
# There are NAs in column related "SnV", which mean not tried, so replace them with 0
data_serves[is.na(data_serves)] <- 0
```

Data 4 - data_shots

This dataset has information about how each player hit the ball. (type, count, was it winning shot and etc) It consists of columns of match_id, player, type, shots, winners, serveRet, shotsInPtsWon, shotsInPtsLost

Preprocessing 6 According to the planned analysis, I didn't need some rows in the dataset, because they were unnecessary observations in my study. Thus, I removed them from the original dataset.

```
### Preprocessing 6
data_shots <- data_shots %>%
  filter(type != "Fside" & type != "Bside" & type != "F" & type != "B" & type != "R" &type != "S" &
         type != "U" & type != "Y" & type != "L" & type != "M" & type != "V" & type != "Z" & type != "O

# check summary and if there are NA values in the data
summary(data_shots)
```

```
##    match_id             player          type               shots       
##  Length:63187       Min.   :1.000   Length:63187       Min.   :   1.0  
##  Class :character   1st Qu.:1.000   Class :character   1st Qu.:   3.0  
##  Mode  :character   Median :1.000   Mode  :character   Median :  16.0  
##                     Mean   :1.498                      Mean   :  93.7  
##                     3rd Qu.:2.000                      3rd Qu.: 151.0  
##                     Max.   :2.000                      Max.   :1181.0  
##     winners           serveRet      shotsInPtsWon    shotsInPtsLost  
##  Min.   :  0.000   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00  
##  1st Qu.:  0.000   1st Qu.:  0.00   1st Qu.:  2.00   1st Qu.:  1.00  
##  Median :  3.000   Median :  0.00   Median :  9.00   Median :  7.00  
##  Mean   :  6.129   Mean   : 19.92   Mean   : 46.97   Mean   : 46.74  
##  3rd Qu.:  9.000   3rd Qu.: 36.00   3rd Qu.: 74.00   3rd Qu.: 75.00  
##  Max.   :131.000   Max.   :238.00   Max.   :623.00   Max.   :658.00
```

```
names(which(colSums(is.na(data_shots)) > 0))
```

```
## character(0)
```

Result of Data Cleaning

```
head(data_matches)
```

```
##                                                    match_id       date
## 1               20220713-M-Newport-R16-Andy_Murray-Max_Purcell 2022-07-13
## 2               20220712-M-Newport-R32-Andy_Murray-Sam_Querrey 2022-07-12
## 3          20220712-M-Bastad-R32-Emil_Ruusuvuori-Dominic_Thiem 2022-07-12
## 4          20220706-M-Wimbledon-QF-Nick_Kyrgios-Cristian_Garin 2022-07-06
## 5 20220704-M-Wimbledon-R16-Rafael_Nadal-Botic_Van_De_Zandschulp 2022-07-04
## 6     20220703-M-Wimbledon-R16-Novak_Djokovic-Tim_Van_Rijthoven 2022-07-03
##           player1                 player2 tournament        court surface
## 1     Andy Murray             Max Purcell    Newport       Center   Grass
## 2     Andy Murray             Sam Querrey    Newport      Stadium   Grass
## 3 Emil Ruusuvuori           Dominic Thiem     Bastad       Center    Clay
## 4    Nick Kyrgios          Cristian Garin  Wimbledon            1   Grass
## 5    Rafael Nadal Botic Van De Zandschulp  Wimbledon Centre Court   Grass
## 6  Novak Djokovic       Tim Van Rijthoven  Wimbledon Centre Court   Grass
```

```
head(data_points)
```

```
##                                         match_id seq set1 set2 gm1 gm2 points
## 1 20220713-M-Newport-R16-Andy_Murray-Max_Purcell   1    0    0   0   0    0-0
## 2 20220713-M-Newport-R16-Andy_Murray-Max_Purcell   2    0    0   0   0   15-0
## 3 20220713-M-Newport-R16-Andy_Murray-Max_Purcell   3    0    0   0   0   30-0
## 4 20220713-M-Newport-R16-Andy_Murray-Max_Purcell   4    0    0   0   0   40-0
## 5 20220713-M-Newport-R16-Andy_Murray-Max_Purcell   5    0    0   1   0    0-0
## 6 20220713-M-Newport-R16-Andy_Murray-Max_Purcell   6    0    0   1   0   0-15
##   gm svr     x1st     x2nd winner isSvrWinner rallyCount seqInGm
## 1  1   1        S               1           1          1       1
## 2  1   1        S               1           1          1       2
## 3  1   1       6n    5b1d#      1           1          1       3
## 4  1   1       4*               1           1          1       4
## 5  2   2       6d 5b28b3n@      1           0          2       1
## 6  2   2 4r18f3n@               1           0          2       2
```

```
head(data_serves)
```

```
##                                                   match_id player pts
## 1 19751219-M-Davis_Cup_World_Group_F-RR-Bjorn_Borg-Jiri_Hrebec      1  69
## 2 19751219-M-Davis_Cup_World_Group_F-RR-Bjorn_Borg-Jiri_Hrebec      2  63
## 3    19780125-M-Pepsi_Grand_Slam-SF-Brian_Gottfried-Bjorn_Borg      1  55
## 4    19780125-M-Pepsi_Grand_Slam-SF-Brian_Gottfried-Bjorn_Borg      2  49
## 5                19800705-M-Wimbledon-F-John_Mcenroe-Bjorn_Borg      1 180
## 6                19800705-M-Wimbledon-F-John_Mcenroe-Bjorn_Borg      2 196
##   ptsWon aces ptsWonLte3Shots wide body  t snvPts snvPtsWon nonSnvPts
## 1     46    2              18   29   28 12     30        23        38
## 2     27    2              13   16   20 27     10         6        51
## 3     30    0              12   19   27  9     23        17        31
## 4     35    0              11   12   23 14      0         0         0
## 5    119   12              93   60   59 61    165       107        12
```

```
## 6    131    8              68    48    82 65     116         81         73
##   nonSnvPtsWon
## 1           23
## 2           21
## 3           13
## 4            0
## 5           12
## 6           50
```

```
head(data_shots)
```

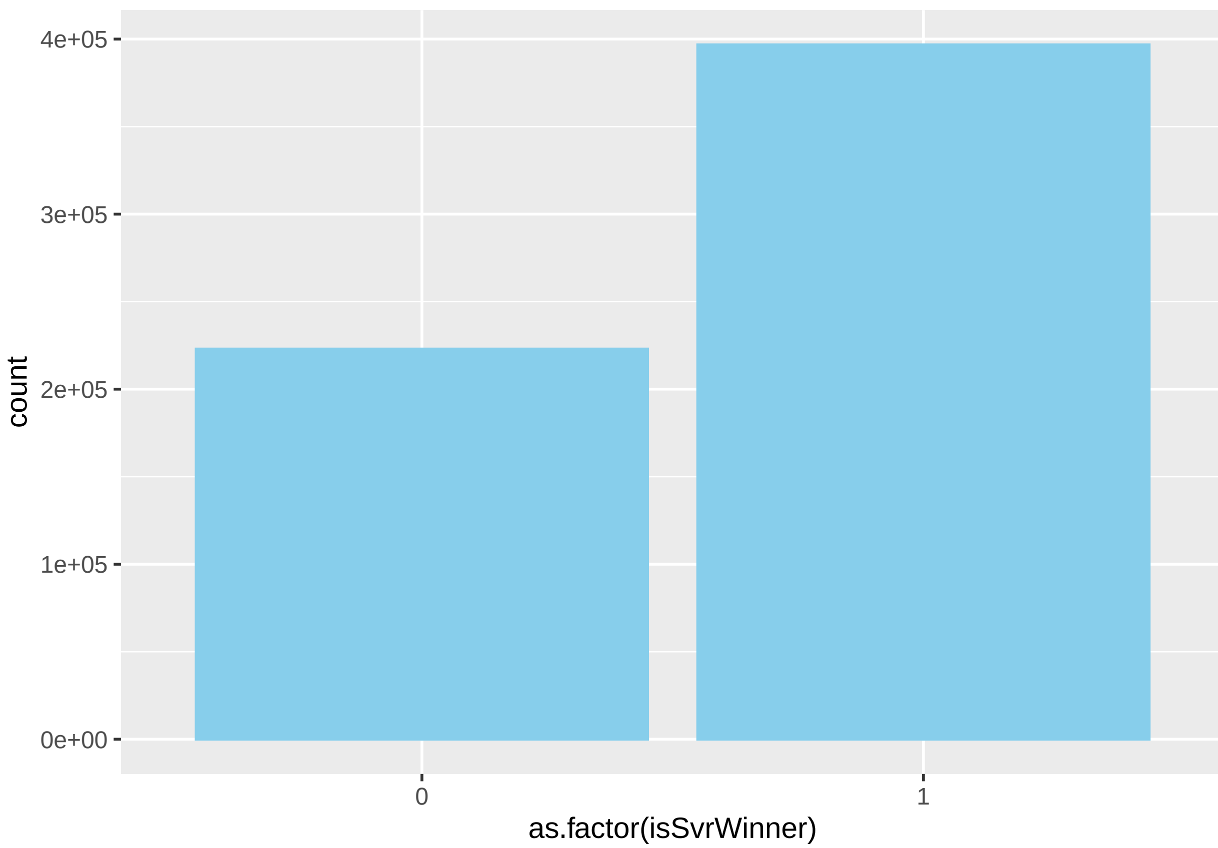
```
##                                                        match_id player  type
## 1 19751219-M-Davis_Cup_World_Group_F-RR-Bjorn_Borg-Jiri_Hrebec      1 Total
## 2 19751219-M-Davis_Cup_World_Group_F-RR-Bjorn_Borg-Jiri_Hrebec      1   Fgs
## 3 19751219-M-Davis_Cup_World_Group_F-RR-Bjorn_Borg-Jiri_Hrebec      1   Bgs
## 4 19751219-M-Davis_Cup_World_Group_F-RR-Bjorn_Borg-Jiri_Hrebec      1  Base
## 5 19751219-M-Davis_Cup_World_Group_F-RR-Bjorn_Borg-Jiri_Hrebec      1   Net
## 6 19751219-M-Davis_Cup_World_Group_F-RR-Bjorn_Borg-Jiri_Hrebec      1    Gs
##   shots winners serveRet shotsInPtsWon shotsInPtsLost
## 1   269      26       53           170             99
## 2   113       3       33            72             41
## 3   112       7       20            65             47
## 4   228      10       53           138             90
## 5    41      16        0            32              9
## 6   213       9       49           131             82
```

Phase 4 : EDA

EDA 1

Basically, tennis is a game that is more advantageous to score if you serve. In fact, the data and plot show that the player who serves is twice as likely to score that point. Therefore, when analyzing actual data, more meaningful analysis will be possible if the difference between advantages and disadvantages according to these serve-receive.

```
ggplot(data_points) + geom_bar(aes(x = as.factor(isSvrWinner)), color = "skyblue", fill = "skyblue")
```




```
sum(data_points$isSvrWinner) / nrow(data_points) * 100
```

```
## [1] 64.02631
```

EDA 2

In general, players serve the ball by targeting the inside or outside course to make it difficult for their opponents to receive the serve. According to actual data, players averaged 18.22 body(middle) serves, 36.3 wide(outside) serves, and 30.45 t-zone(inside) serves in each game. When analyzing a player, the higher the ratio of wide and t compared to the body, the more sophisticated the player can be judged as a big server.

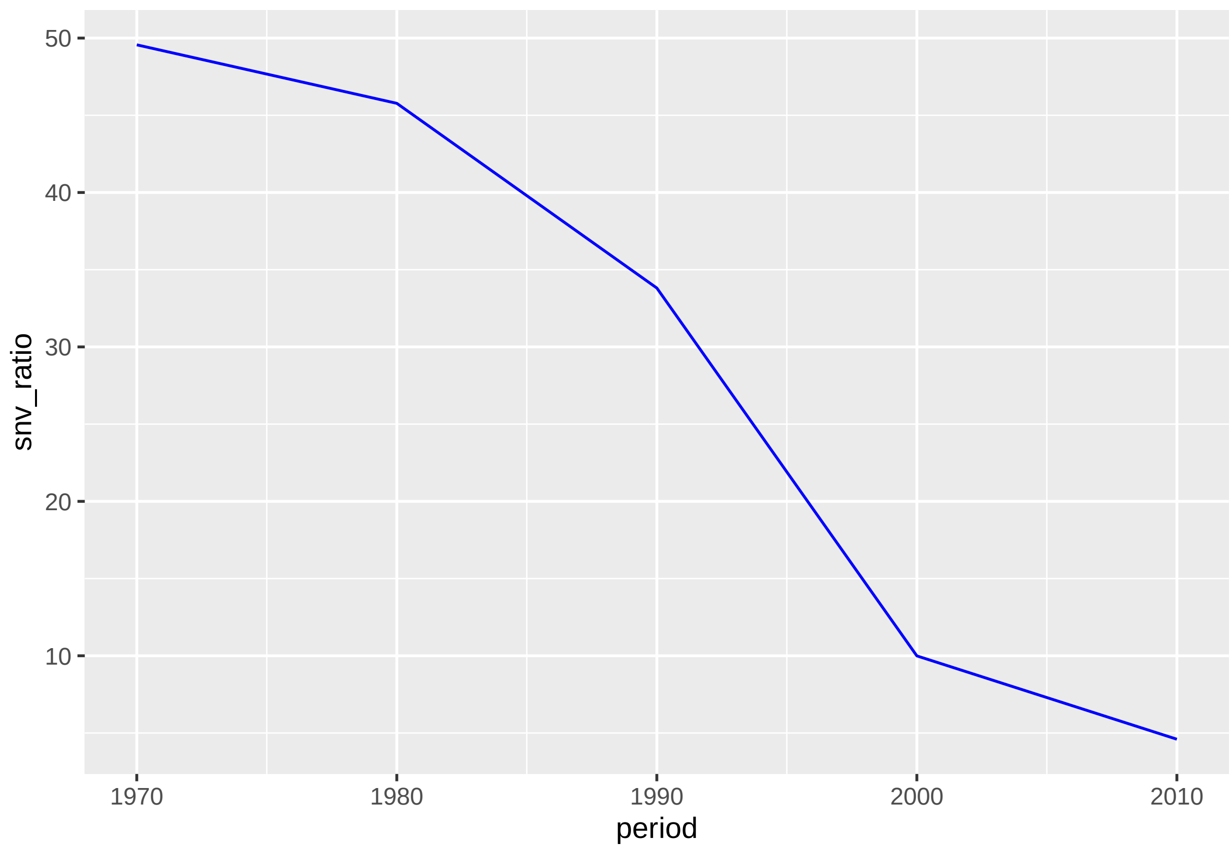
```
summary(data_serves %>% select(wide, body, t))
```

```
##       wide            body               t
##  Min.   :  2.0   Min.   :  0.00   Min.   :  1.00
##  1st Qu.: 23.0   1st Qu.: 10.00   1st Qu.: 19.00
##  Median : 32.0   Median : 16.00   Median : 27.00
##  Mean   : 36.3   Mean   : 18.22   Mean   : 30.45
##  3rd Qu.: 46.0   3rd Qu.: 24.00   3rd Qu.: 38.00
##  Max.   :230.0   Max.   :105.00   Max.   :217.00
```

EDA 3

The data set includes the 1970s to 2022 games, and I thought there would be differences not only between players but also between trends of the period. The data showed that the ratio of serve & volley, which used to be close to 50%, decreased over time. In modern tennis, serve & volley is not as powerful a tactic as before, so the percentage of attempts has decreased to less than 5%.

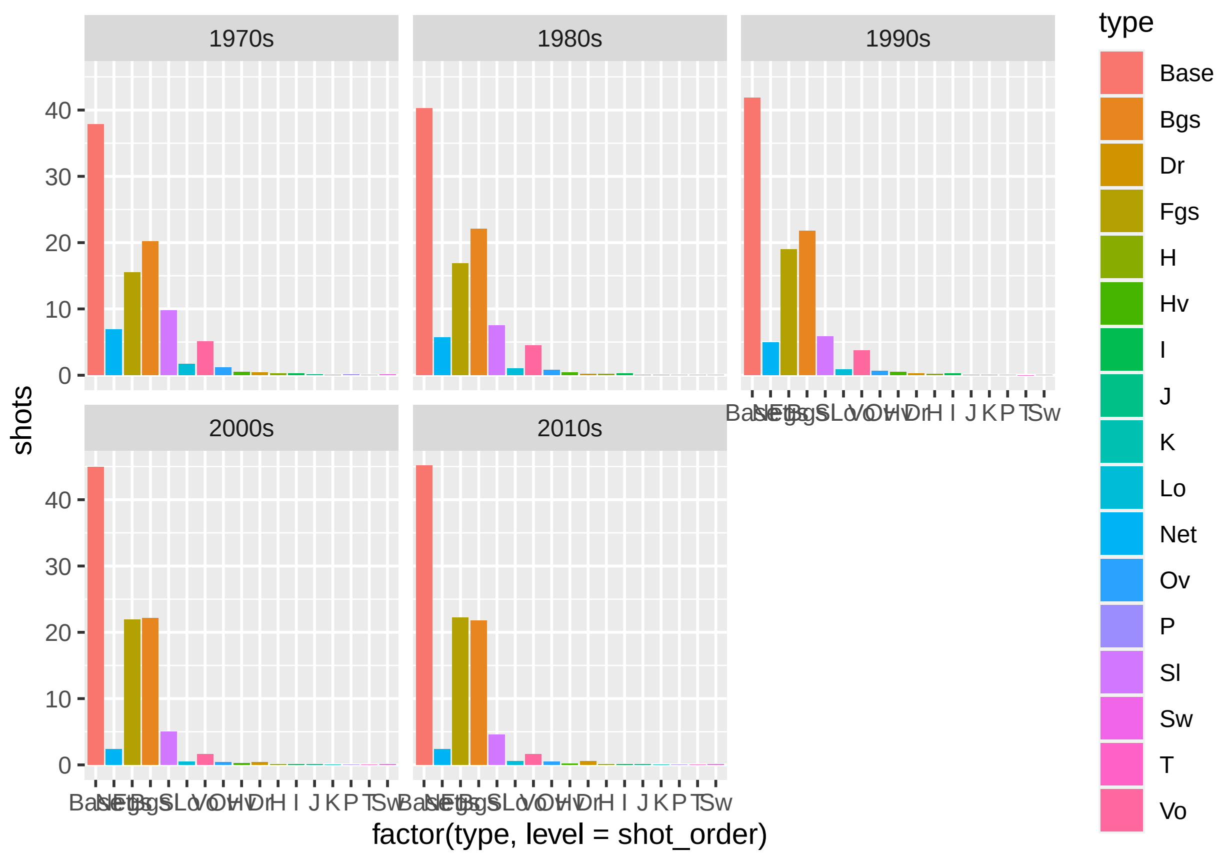
```
match_date <- data_matches %>%
  select(match_id, date)

serve_1970s <- data_serves %>%
  left_join(match_date, by= "match_id") %>%
  filter(date < as.Date("1980-01-01"))

serve_1980s <- data_serves %>%
  left_join(match_date, by= "match_id") %>%
  filter(date >= as.Date("1980-01-01") & date < as.Date("1990-01-01"))

serve_1990s <- data_serves %>%
  left_join(match_date, by= "match_id") %>%
  filter(date >= as.Date("1990-01-01") & date < as.Date("2000-01-01"))

serve_2000s <- data_serves %>%
  left_join(match_date, by= "match_id") %>%
  filter(date >= as.Date("2000-01-01") & date < as.Date("2010-01-01"))

serve_2010s <- data_serves %>%
  left_join(match_date, by= "match_id") %>%
  filter(date >= as.Date("2010-01-01"))

ratio1970 <- sum(serve_1970s$snvPts) / sum(serve_1970s$pts) * 100
ratio1980 <- sum(serve_1980s$snvPts) / sum(serve_1980s$pts) * 100
ratio1990 <- sum(serve_1990s$snvPts) / sum(serve_1990s$pts) * 100
ratio2000 <- sum(serve_2000s$snvPts) / sum(serve_2000s$pts) * 100
ratio2010 <- sum(serve_2010s$snvPts) / sum(serve_2010s$pts) * 100

snv_ratio <- data.frame(period = c(1970, 1980, 1990, 2000, 2010), snv_ratio = c(ratio1970, ratio1980, ra
ggplot(data = snv_ratio) + geom_line(mapping = aes(x = period, y = snv_ratio), color = "blue")
```



EDA 4

We looked at the types and proportions of shots that players actually play during the game, and whether there are differences in their patterns depending on the times through a plot by era.

```
shot_order <- c("Base", "Net", "Gs", "Fgs", "Bgs", "Sl", "Lo", "Vo", "Ov", "Hv", "Dr",  "H", "I", "J",

ggplot(shots, aes(x = factor(type, level = shot_order), y = shots, fill = type)) + geom_bar(stat = "iden
```



Phase 5 : Modeling and Analysis

In this phase, various modeling such as player clustering, logistic regression analysis, and linear regression analysis are done through tennis data.

Modeling 1 : Clustering by Playing Type

There are various ways of classifing tennis players, and the classification of player types is sometimes not clearly applied to all players. Nevertheless, it is often divided into four types, Aggressive Baseliner, Counter Puncher, Serve & Volleyer, and All-rounder, so I used these four types for clustering model.

To briefly explain each type, first, **Aggressive Baseliner** is the most common type in modern tennis. They are the bassliner that leads the game with a powerful ground stroke from the baseline. Along with the above, **Counter Puncher**, which is equally common in modern tennis, is a baseliner that seeks an opponent's mistake or a decisive winner shot in a long rally through persistent defense, rather than focusing on strong attacks. **Serve & Volleyer** is the type that approaches the net after a strong serve or receive and seeks to score through volley. This type is relatively rare type in modern tennis. **All-rounder** is a balanced type of player that performs well not only on serve, but also on both the baseline and the net. This type of player is very rare.

In this modeling, representative players of each type were first labeled and the remaining players were clustered by K-NN based on the stat similarity of the players. There are four stats used for classification. - Ratio of Wide & T serve among all serve - Percentage of sub and volley attempts - Average rally count - Winner shot ratio between base and net

Among the 700 players, 590 players, excluding 110 players who lacked stat data, were classified as follows.

```r
############### Modeling 1 ###################
# K-NN Clusturing
agr_baseliner <- c("Kei Nishikori", "Alexander Zverev", "Rafael Nadal", "Novak Djokovic")
snvolleyer <- c("Patrick Rafter", "Pete Sampras", "Brian Teacher", "Kim Warwick", "Chris Lewis", "Andrea
cnt_pnchr <- c("Michael Chang", "Lleyton Hewitt", "Andy Murray", "Daniil Medvedev")
all_rounder <- c("Roger Federer", "Grigor Dimitrov", "Stefanos Tsitsipas", "Grigor Dimitrov")

data_players$playing_type <- if_else(data_players$player %in% agr_baseliner, "aggressive baseliner", dat
data_players$playing_type <- if_else(data_players$player %in% snvolleyer, "serve & volleyer", data_playe
data_players$playing_type <- if_else(data_players$player %in% cnt_pnchr, "counter puncher", data_players
data_players$playing_type <- if_else(data_players$player %in% all_rounder, "all-round player", data_pla

train_set <- data_players %>%
  filter(playing_type != "")

predict_set <- data_players %>%
  filter(playing_type == "")

# normalization function
nor <- function(x) { (x - min(x)) / (max(x) - min(x)) }

# run nomalization on dataset
# because they are the predictors
data_players_norm <- as.data.frame(lapply(data_players[,c(2,3,4,5)], nor))
data_players_norm <- cbind(player = data_players$player, data_players_norm, playing_type = data_players$

train_set_norm <- data_players_norm %>%
  filter(playing_type != "")

predict_set_norm <- data_players_norm %>%
  filter(playing_type == "")

set.seed(400)
pr <- knn(train_set_norm[,c(2:5)], predict_set_norm[,c(2:5)], cl = train_set[,6], k = 10)
predict_set <- cbind(predict_set[,c(1:5)], playing_type = pr)
clustered_data_players <- rbind(train_set, predict_set)
data_players <- clustered_data_players
```

```r
summary(pr)
```

```
## aggressive baseliner     all-round player      counter puncher
##                  238                    8                  281
##     serve & volleyer
##                   45
```

```r
head(clustered_data_players)
```

```
##                player wide_t_ratio snv_ratio avg_rally_count
## 1        Andy Murray     77.29912 2.0349815        4.784482
## 2       Rafael Nadal     71.15789 1.0138002        4.705317
## 3     Novak Djokovic     80.70546 1.7051326        4.810109
## 4 Stefanos Tsitsipas     76.26883 2.7108434        4.048335
```

```
## 5    Daniil Medvedev     85.36274 1.2932605        4.660476
## 6    Grigor Dimitrov     77.94711 0.9854423        4.154044
##   winner_base_net_ratio         playing_type
## 1              3.058552      counter puncher
## 2              3.788285 aggressive baseliner
## 3              3.051153 aggressive baseliner
## 4              1.879880     all-round player
## 5              3.503937      counter puncher
## 6              2.932075     all-round player
```

```
head(data_players)
```

```
##               player wide_t_ratio snv_ratio avg_rally_count
## 1        Andy Murray     77.29912 2.0349815        4.784482
## 2        Rafael Nadal     71.15789 1.0138002        4.705317
## 3      Novak Djokovic     80.70546 1.7051326        4.810109
## 4 Stefanos Tsitsipas     76.26883 2.7108434        4.048335
## 5    Daniil Medvedev     85.36274 1.2932605        4.660476
## 6    Grigor Dimitrov     77.94711 0.9854423        4.154044
##   winner_base_net_ratio         playing_type
## 1              3.058552      counter puncher
## 2              3.788285 aggressive baseliner
## 3              3.051153 aggressive baseliner
## 4              1.879880     all-round player
## 5              3.503937      counter puncher
## 6              2.932075     all-round player
```

Modeling 2, 3, 4 : Analysis of winning strategies by type - Logistic Regression Modeling

These three models are an analysis of winning strategies by type. There are conventional winning strategies for each type. Logistic regression analysis between indicators representing the strategy and winning, analyzed how meaningful the strategy is in the real data.

Modeling 2 (Aggressive Baseliner) Since this type generally has strength in strokes near the baseline, it is important to make them hit many shots near the net to defeat them. To verify this, analysis between the ratio of net shot and winning was conducted.

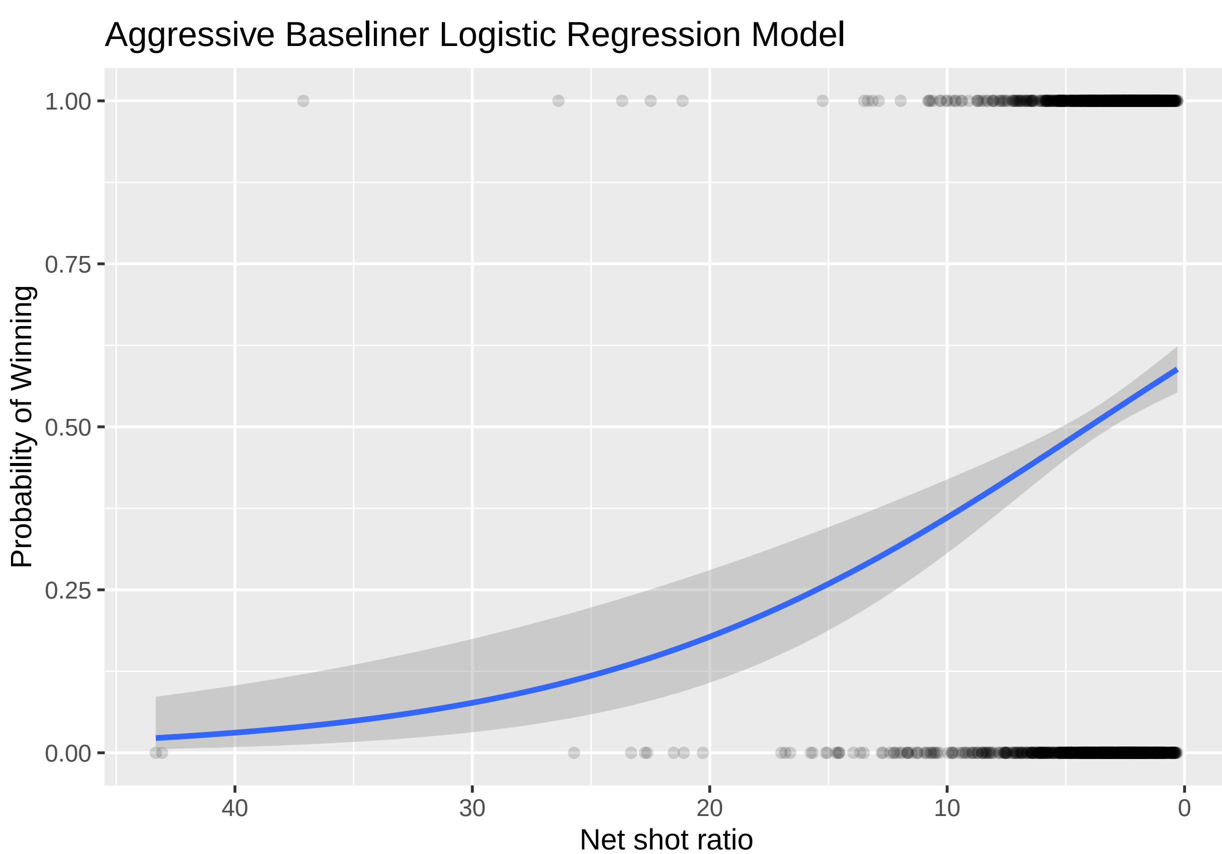
```
############### Modeling 2 ###################
# Logistic Regression Analysis
model <- glm(is_AB_winner ~ net_ratio, family = "binomial", data = matches_AB)
summary(model)
```

```
##
## Call:
## glm(formula = is_AB_winner ~ net_ratio, family = "binomial",
##     data = matches_AB)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.332  -1.208   1.037   1.124   2.534
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  0.38653    0.07886   4.902 9.50e-07 ***
## net_ratio   -0.09581    0.01792  -5.346 8.97e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2507.0  on 1808  degrees of freedom
## Residual deviance: 2472.8  on 1807  degrees of freedom
## AIC: 2476.8
##
## Number of Fisher Scoring iterations: 3
```

```
matches_AB %>%
  ggplot(aes(net_ratio, is_AB_winner)) +
  geom_point(alpha = .15) +
  scale_x_reverse() +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  ggtitle("Aggressive Baseliner Logistic Regression Model") +
  xlab("Net shot ratio") +
  ylab("Probability of Winning")
```

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

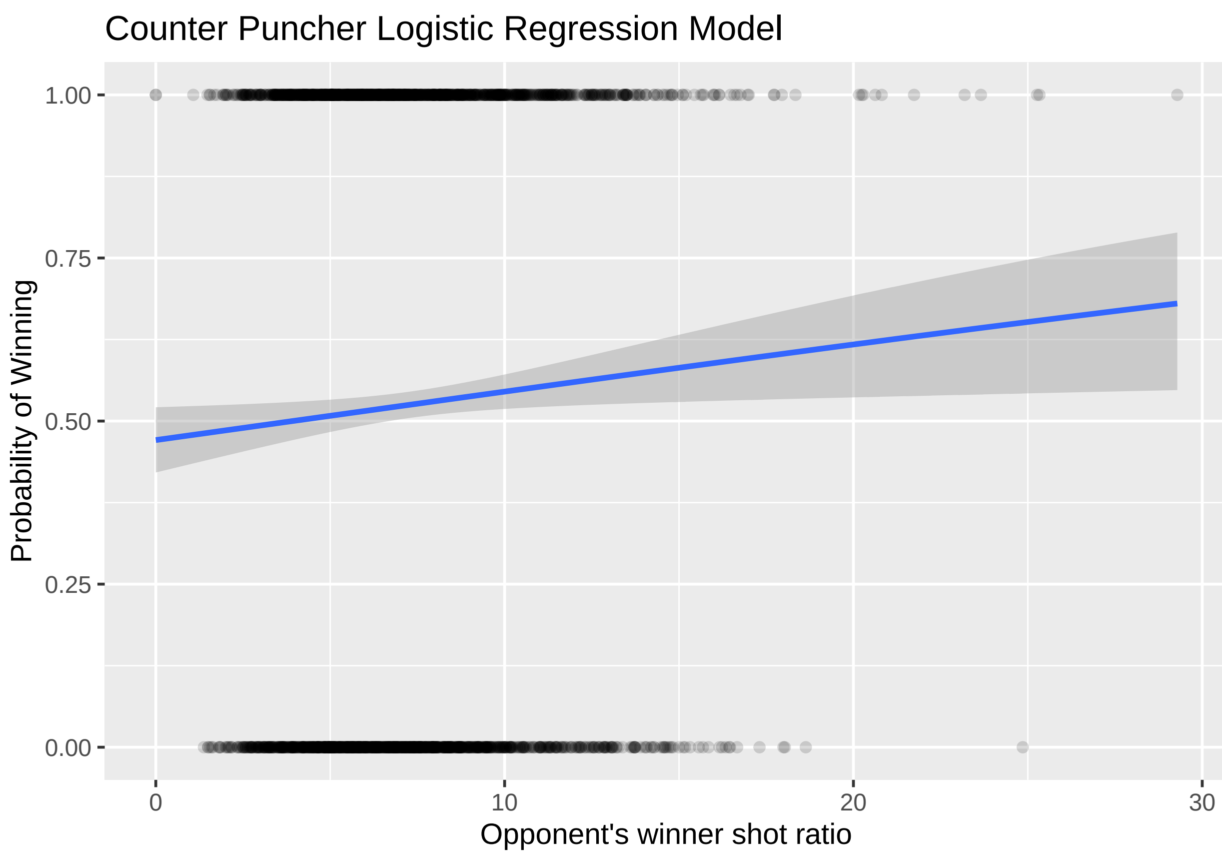


Evaluation As a result of analysis, there was a somewhat significant relationship and P-value. It is not a very decisive independent variable, but it can be seen as a relatively significant strategy in tennis games where many factors are involved. So I can say that there is sufficient evidence to support the hypothesis.

Modeling 3 (Counter Puncher) Counter puncher is a type capable of persistent defense. Therefore, when playing against them, it is important to continue the rally more patiently than a hasty attack to finish the point quickly. To verify this, an analysis between the proportion of opponent's winner shot among total shot and the victory was conducted.

```
##
## Call:
## glm(formula = is_CP_winner ~ winner_shot_ratio, family = "binomial",
##     data = matches_CP)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.451  -1.211   1.059   1.143   1.227
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -0.11658    0.10269  -1.135   0.2563
## winner_shot_ratio  0.02976    0.01294   2.299   0.0215 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3268.5  on 2361  degrees of freedom
## Residual deviance: 3263.2  on 2360  degrees of freedom
## AIC: 3267.2
##
## Number of Fisher Scoring iterations: 3
```

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



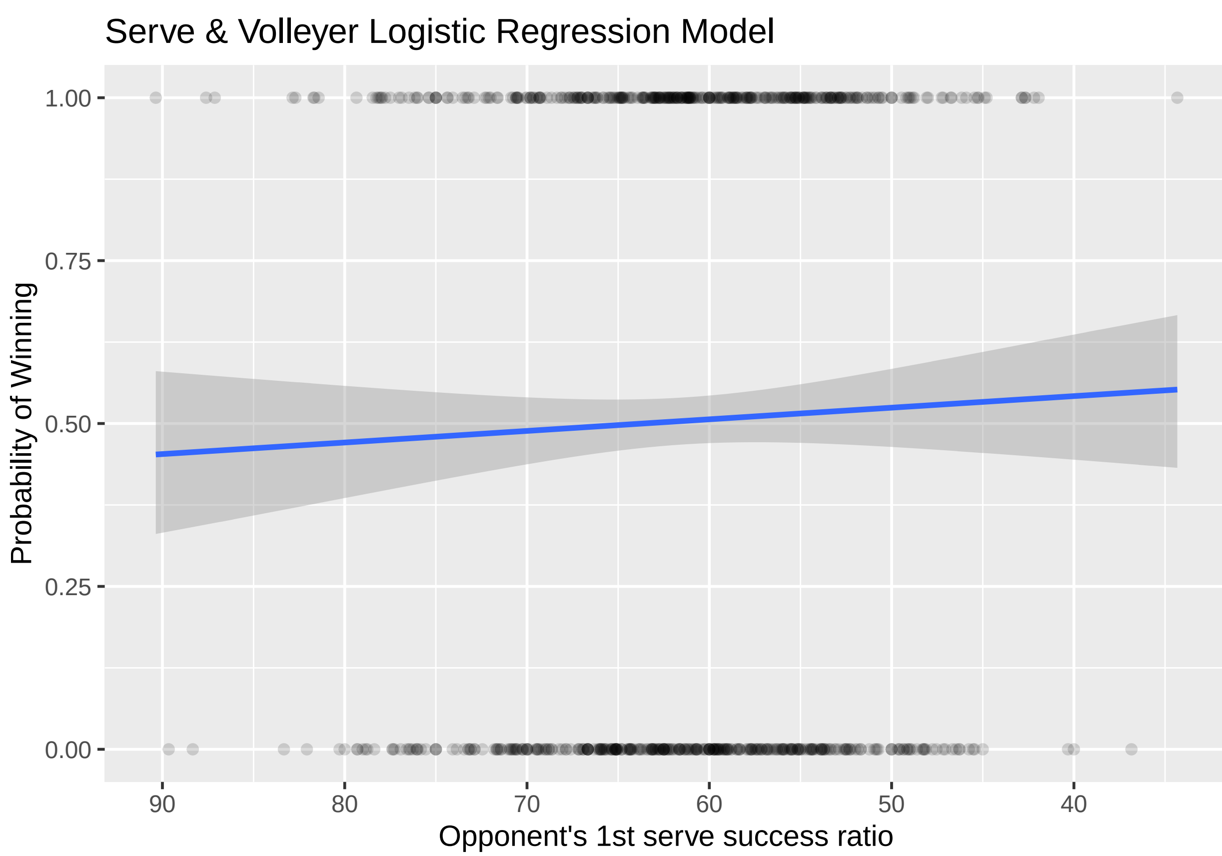
Evaluation As a result of the analysis, the independent variable was not significant for Counter Puncher's victory. The relationship was not clear and the p-value was not low enough. Conventional ideas have not been clearly demonstrated in the data. So there is not sufficient evidence to support the claim.

Modeling 4 (Serve & Volleyer) Serve & Volleyer prefer to receive a weak shot and approach to the net. Therefore, when playing with them, it is important to succeed the first serve and not give the opponent a chance with a weak second serve. To verify this, an analysis was conducted between the opponent's first serve success rate and victory.

```
############### Modeling 4 ###################
model <- glm(is_SV_winner ~ fst_srv_suc_ratio, family = "binomial", data = matches_SV)
summary(model)
```

```
##
## Call:
## glm(formula = is_SV_winner ~ fst_srv_suc_ratio, family = "binomial",
##     data = matches_SV)
##
## Deviance Residuals:
##    Min       1Q   Median       3Q      Max
## -1.260   -1.184   1.114    1.170   1.259
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)        0.453887   0.538548   0.843    0.399
## fst_srv_suc_ratio -0.007133   0.008695  -0.820    0.412
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1023.0  on 737  degrees of freedom
## Residual deviance: 1022.4  on 736  degrees of freedom
## AIC: 1026.4
##
## Number of Fisher Scoring iterations: 3

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



Evaluation As a result of the analysis, the independent variable was not significant for Serve & Volleyer's victory. The relationship was not clear and the p-value was not low enough. Conventional ideas have not been clearly demonstrated in the data. So there is not sufficient evidence to support the claim.

Modeling 5 : Analysis of the relationship between serve area and serve point scoring - Linear Regression Modeling

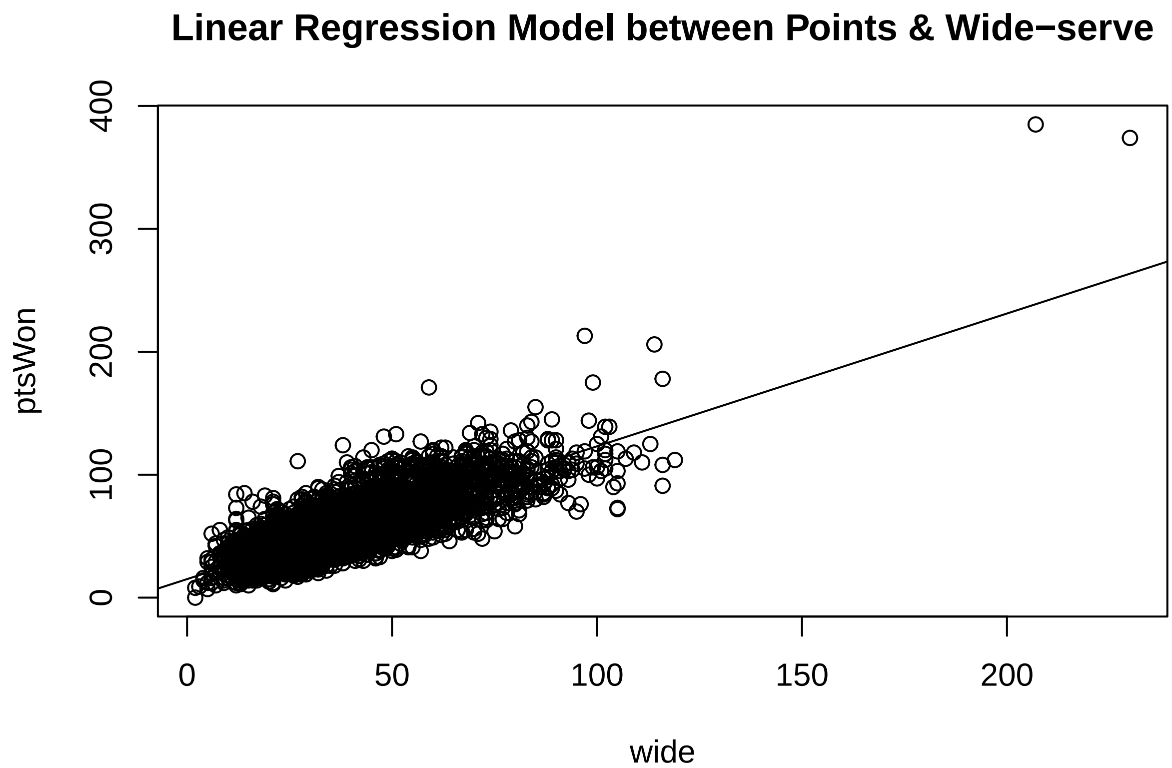
Serve is very important in tennis. Rather than an ordinary body serve, especially a wide serve that goes outward and a T serve that goes in the opponent's backhand direction are more powerful. Thus, I conducted linear regression analysis to see the relationship between each serve area and the serve-point scoring.

```r
############### Modeling 5 ###################
wide.model <- lm(ptsWon ~ wide, data = data_serves)
summary(wide.model)
```

Wide Serve

```
##
## Call:
## lm(formula = ptsWon ~ wide, data = data_serves)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.522  -8.525  -0.856   6.966 146.272
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  15.0743     0.4257   35.41   <2e-16 ***
## wide          1.0804     0.0105  102.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.73 on 5234 degrees of freedom
## Multiple R-squared:  0.6693, Adjusted R-squared:  0.6693
## F-statistic: 1.06e+04 on 1 and 5234 DF,  p-value: < 2.2e-16
```

```r
plot(ptsWon ~ wide, data = data_serves)
title(main = "Linear Regression Model between Points & Wide-serve")
abline(wide.model)
```

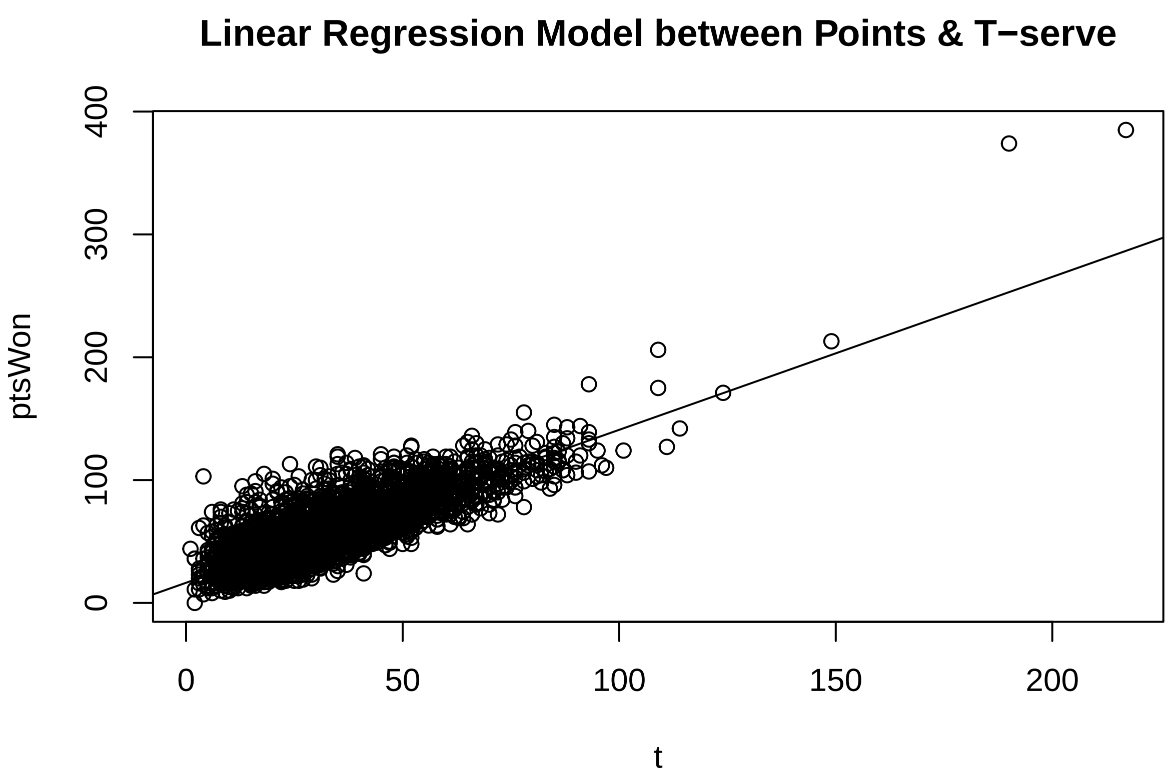



```
t.model <- lm(ptsWon ~ t, data = data_serves)
summary(t.model)
```

T Serve

```
##
## Call:
## lm(formula = ptsWon ~ t, data = data_serves)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.441  -8.783  -1.538   6.785 121.027
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.38884    0.40426   40.54   <2e-16 ***
## t            1.24518    0.01178  105.68   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.49 on 5234 degrees of freedom
## Multiple R-squared:  0.6809, Adjusted R-squared:  0.6808
## F-statistic: 1.117e+04 on 1 and 5234 DF,  p-value: < 2.2e-16
```

```
plot(ptsWon ~ t, data = data_serves)
title(main = "Linear Regression Model between Points & T-serve")
abline(t.model)
```



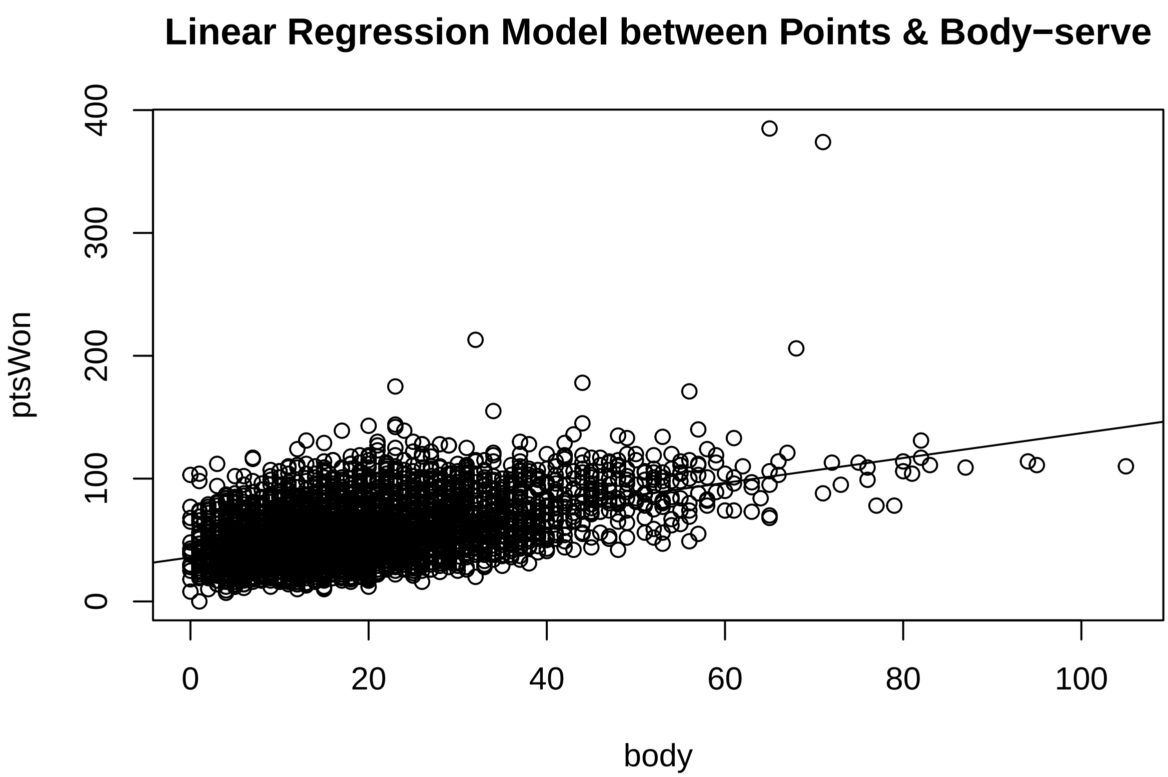

```
body.model <- lm(ptsWon ~ body, data = data_serves)
summary(body.model)
```

Body Serve

```
##
## Call:
## lm(formula = ptsWon ~ body, data = data_serves)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.227 -13.923  -3.227  10.903 283.413
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.87927    0.52403   68.47   <2e-16 ***
## body         1.01088    0.02412   41.92   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.66 on 5234 degrees of freedom
## Multiple R-squared:  0.2513, Adjusted R-squared:  0.2512
## F-statistic:  1757 on 1 and 5234 DF,  p-value: < 2.2e-16
```

```
plot(ptsWon ~ body, data = data_serves)
title(main = "Linear Regression Model between Points & Body-serve")
abline(body.model)
```



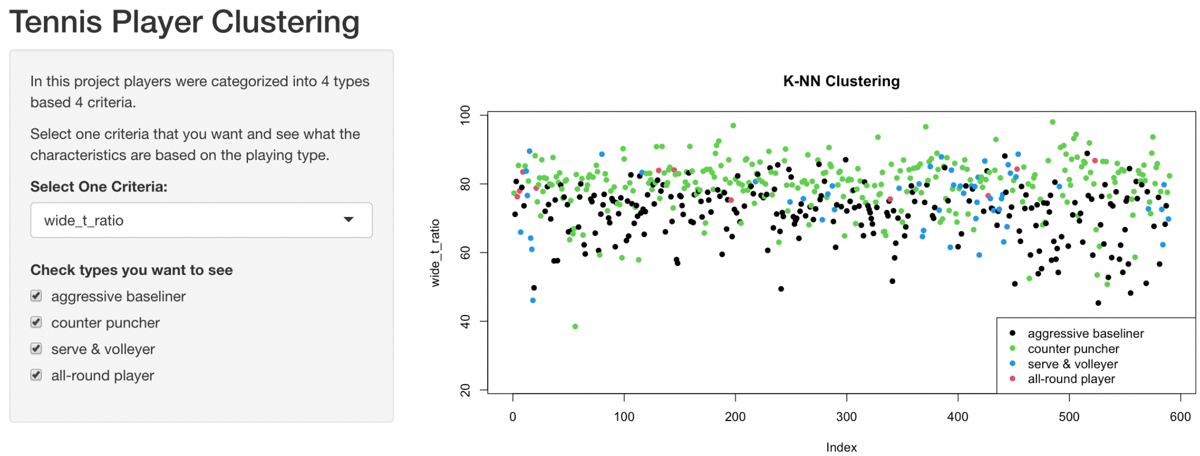
Evaluation As a result of analysis, in the case of Wide and T serve, significant results were shown with a low p-value and a high R square value. On the other hand, in the case of the body serve, there was no significant result, confirming that the hypothesis was correct. The more wide or T serve was succeeded than body serve, the higher the frequency of serve-point scored. There is sufficient evidence to support the hypothesis.

Phase 6: Data Product

In this phase, I made a simple R Shiny Application to show the result of my analysis. The dashboard consist of two part. One is "Player Clustering" and another is "Match Analysis of Each Player".

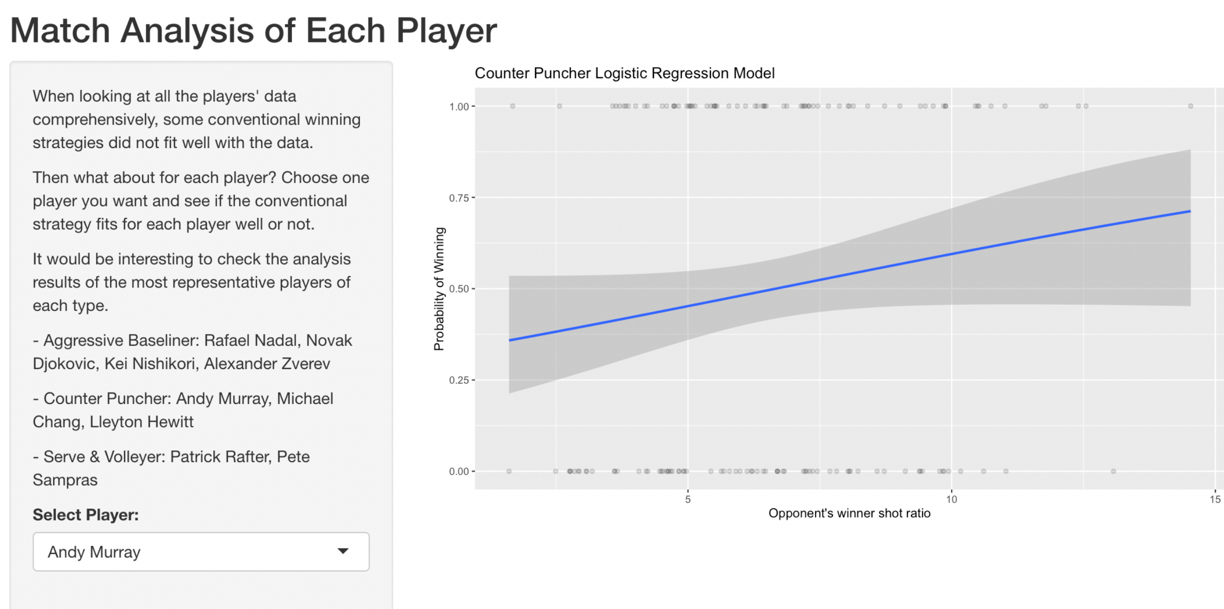
Part 1

In first part, it shows a plot representing the result of K-NN Clustering Model. All Players were categorized into 4 types based on 4 criteria. So you can choose one criteria and see what the characteristics are based on the playing type. You can also filter some types by checking the options.



Part 2

In second part, it shows a plot about Logistic Regression Analysis. In my project, when looking at all the players' data comprehensively, some conventional winning strategies did not fit well with the data. However, I thought those strategies still can fit for some players. You can choose one player and check if he fits for the conventional winning strategy. Unlike the results of the overall data confirmed in the analysis phase, the most representative players of each type showed some significant correlation.



Conclusion

In this project, there were two hypothesis that I wanted to figure out.

1) When we classify the types of tennis players through data, the baseliner will be the most, the serve & volleyer will be the relatively few, and the all-round player will be the least as it is generally known.

2) The traditional winning strategies for each type (aggressive baseliner, counter pucher, serve & volleyer) will also be valid based on real match data.

Analysis Through K-NN clustering, I found sufficient evidence to support the first hypothesis.The baseliner including aggressive baseliner and counter puncher was the most and the serve & volleyer and all-round player were few.

According to the logistic regression analysis, I found sufficient evidence to support that the strategy for the aggressive baseliner is valid. On the other hand, there was not sufficient evidence to support the other two strategies. However, when it comes to the most representative players for each type, the strategies were more suitable. So I can say there was more clear evidence to support the hypothesis about the representative players for each type.

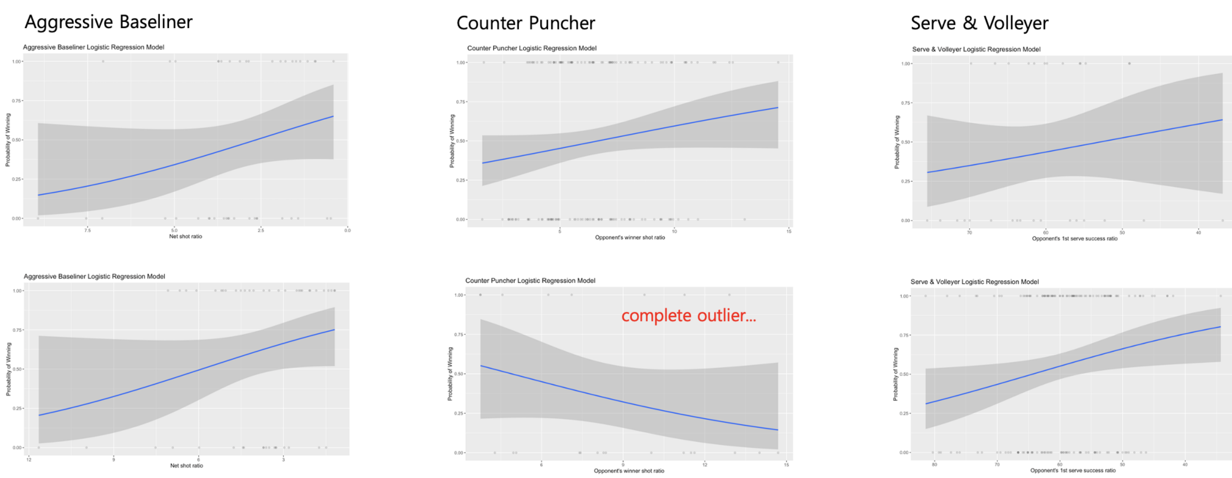


Figure 1: Regression Analysis for Representative Players