a. Data gathering and integration.

The NFL draft data set: 1985 – 2015.

Link: https://www.kaggle.com/datasets/ulrikthygepedersen/nfl-draft-1985-2015

Per our conversation, I have decided instead of working with 32 NFL teams. I will split the teams into two 2 conferences, AFC and NFC. There are 3 extra teams during (1985 – 2015) due to relocation and change of ownership.

- American Football Conference (AFC)
- National Football Conference (NFC)

I will do this step in the main CVS files. Main file: 8435 observations, 34 variables. Therefore, there will be 2 files: NFC_data.csv: 4326 observations, 34 variables; and AFC_data.csv.: 4109 observations, 34 variables.

National Football Conference (NFC):

- PHI – Philadelphia Eagles
- SFO – San Francisco 49ers
- GNB – Green Bay Packers
- MIN – Minnesota Vikings
- NYG – New York Giants
- TAM – Tampa Bay Buccaneers
- SEA – Seattle Seahawks
- DET – Detroit Lions
- NOR – New Orleans Saints
- RAM – Los Angeles Ram
- CHI – Chicago Bears
- WAS – Washington Commanders
- CAR – Carolina Panthers
- ATL – Atlanta Falcons
- ARI – Arizona Cardinals
- PHO – Phoenix Cardinals
- DAL – Dallas Cowboys
- STL – Saint Louis Rams

American Football Conference (AFC):

- KAN – Kansas City Chiefs
- CIN – Cincinnati Bengals
- BUF – Buffalo Bills
- NWE – New England Patriots
- NYJ – New York Jets
- MIA – Miami Dolphins
- BAL – Baltimore Ravens
- OAK – Oakland Raiders
- RAI – Las Vegas Raiders
- JAX – Jacksonville Jaguars
- DEN – Denver Broncos
- IND – Indianapolis Colts
- SDG – San Diego Chargers (now Los Angles Chargers)
- TEN – Tennessee Titans
- CLE – Cleveland Browns
- HOU – Houston Texans
- PIT – Pittsburgh Steelers

```
library(tidyverse)
AFCdata <- AFC_data %>% mutate(Conference = 'AFC')
#Adding 'conference' column indicates type of conference.
#4109 observation, 35 variables.
NFCdata <- NFC_data %>% mutate(Conference = 'NFC')
#Adding 'conference' column indicates type of conference.
#4326 observation, 35 variables.

NFLdata <- full_join(AFCdata, NFCdata)
#Join 2 data frames.
#8435 observations, 35 variables.

summary(NFLdata)
```

```
  column_a         player_id             year           rnd              pick            tm              player             hof           
 Length:8435        Length:8435        Min.   :1985   Min.   : 1.000   Min.   :  1.0   Length:8435        Length:8435        Length:8435       
 Class :character   Class :character   1st Qu.:1991   1st Qu.: 3.000   1st Qu.: 69.0   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Median :1999   Median : 5.000   Median :137.0   Mode  :character   Mode  :character   Mode  :character  
                                       Mean   :1999   Mean   : 4.954   Mean   :139.1                                                        
                                       3rd Qu.:2007   3rd Qu.: 7.000   3rd Qu.:205.0                                                        
                                       Max.   :2015   Max.   :12.000   Max.   :336.0                                                        

    pos            position_standard     first4av           age              to             ap1                pb                st          
 Length:8435        Length:8435        Min.   :-4.000   Min.   :20.00   Min.   :1985   Min.   : 0.00000   Min.   : 0.0000   Min.   : 0.000  
 Class :character   Class :character   1st Qu.: 0.000   1st Qu.:22.00   1st Qu.:1997   1st Qu.: 0.00000   1st Qu.: 0.0000   1st Qu.: 0.000  
 Mode  :character   Mode  :character   Median : 4.000   Median :23.00   Median :2006   Median : 0.00000   Median : 0.0000   Median : 0.000  
                                       Mean   : 9.927   Mean   :22.64   Mean   :2005   Mean   : 0.07362   Mean   : 0.2759   Mean   : 1.831  
                                       3rd Qu.:16.000   3rd Qu.:23.00   3rd Qu.:2013   3rd Qu.: 0.00000   3rd Qu.: 0.0000   3rd Qu.: 3.000  
                                       Max.   :88.000   Max.   :29.00   Max.   :2016   Max.   :10.00000   Max.   :14.0000   Max.   :19.000  
                                                        NA's   :1245    NA's   :1382                                                         
     carav             drav              g               cmp             pass_att        pass_yds          pass_td          pass_int        rush_att      
 Min.   : -4.00   Min.   : -4.00   Min.   :  0.00   Min.   :   0.0   Min.   :    0.0   Min.   :   -8.0   Min.   :  0.00   Min.   :  0.00   Min.   :   0.0  
 1st Qu.:  1.00   1st Qu.:  1.00   1st Qu.: 19.00   1st Qu.:   0.0   1st Qu.:    1.0   1st Qu.:    0.0   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:   3.0  
 Median :  8.00   Median :  6.00   Median : 51.00   Median :   2.0   Median :    5.0   Median :   44.5   Median :  1.00   Median :  1.00   Median :  22.0  
 Mean   : 17.28   Mean   : 13.36   Mean   : 64.52   Mean   : 363.9   Mean   :  615.8   Mean   : 4235.0   Mean   : 25.04   Mean   : 18.92   Mean   : 202.1  
 3rd Qu.: 26.00   3rd Qu.: 18.00   3rd Qu.: 99.00   3rd Qu.: 252.2   3rd Qu.:  475.8   3rd Qu.: 2803.8   3rd Qu.: 12.75   3rd Qu.: 16.75   3rd Qu.: 179.8  
 Max.   :177.00   Max.   :160.00   Max.   :327.00   Max.   :6300.0   Max.   :10169.0   Max.   :71940.0   Max.   :539.00   Max.   :336.00   Max.   :4409.0  
 NA's   :1382     NA's   :2165     NA's   :1415     NA's   :7841     NA's   :7841      NA's   :7841      NA's   :7841     NA's   :7841     NA's   :6789    
    rush_yds         rush_tds          rec             rec_yds          rec_tds           tkl             def_int            sk           
 Min.   : -36.0   Min.   :  0.000   Min.   :  0.0   Min.   : -19   Min.   : 0.00   Min.   :  1.00   Min.   : 0.500   Length:8435       
 1st Qu.:  11.0   1st Qu.:  0.000   1st Qu.:  4.0   1st Qu.:  41   1st Qu.: 0.00   1st Qu.:  4.00   1st Qu.: 1.000   Class :character  
 Median :  94.0   Median :  0.000   Median : 34.0   Median : 315   Median : 1.00   Median : 17.00   Median : 3.500   Mode  :character  
 Mean   : 827.7   Mean   :  5.802   Mean   :105.7   Mean   :1244   Mean   : 7.41   Mean   : 86.75   Mean   : 9.212                     
 3rd Qu.: 655.5   3rd Qu.:  5.000   3rd Qu.:136.5   3rd Qu.:1384   3rd Qu.: 8.00   3rd Qu.:102.00   3rd Qu.:10.000                     
 Max.   :18355.0   Max.   :164.000   Max.   :1549.0   Max.   :22895   Max.   :197.00   Max.   :1562.00   Max.   :200.000                    
 NA's   :6789      NA's   :6789      NA's   :6264     NA's   :6264    NA's   :6264     NA's   :4324      NA's   :6819                       
 college_univ        Conference       
 Length:8435        Length:8435        
 Class :character   Class :character   
 Mode  :character   Mode  :character   
```

- ✍ We can see that there are 15 variables have significant number of missing values Therefore, I will remove those features.

```
> nflData <- NFLdata %>% select(-c("column_a", "player_id", "cmp", "pass_att"
, "pass_yds", "pass_td", "pass_int", "rush_att", "rush_yds", "rush_tds",
"rec","rec_yds", "rec_tds", "tkl", "def_int"))
```

- ✍ Then I notice all players during period of 1985-1993 didn't have University record. Therefore, I will remove observations (rows) from this year frame. Our data set now will be NFL draft during period 1994-2015.
- ✍ The reason I did this as well because of college sport division, where the football player went to college also has high impact on their draft opportunity.

```
nflData = nflData[nflData$year >= "1994" & nflData$year <= "2015", ]
#5538 observations, 20 variables.
```

Variables removal with reasoning:

- ✍ I notice variable ‘sk’ has a lot of missing values, almost half of the data set.
- ✍ Remove the ‘tm’ (team) column since we already have conference type.
- ✍ ‘hof’ variable Hall of Fame has all ‘no’ values.
- ✍ ‘to’ variable says how long they stay with their 1st team which won’t be needed.
- ✍ ‘position_standard’ is the same as ‘position’ variable.
- ✍ ‘player’ variable is name of each player.

- ‘college_univ’ variable is unique as well.
- 3 variables: ‘first4av’, ‘carav’ and ‘drav’ all represent a player’s approximate value. Since we’re not doing deep-dive analysis. I will keep one variable ‘carav’
- Thus, remove these variables.

```
nflData <- nflData %>% select(-c("sk", "tm", "hof", "to", "position_standard"
, "player", "college_univ", "first4av", "drav"))
nflData <- na.omit(nflData)
#Remove missing values here and there.
```

FINAL DATASET

```
summary(nflData)
#4895 observation, 11 variables.
```

```
      year           rnd             pick           pos                 age       
 Min.   :1994   Min.   :1.000   Min.   :  1.0   Length:4895        Min.   :20.00  
 1st Qu.:1999   1st Qu.:2.000   1st Qu.: 56.0   Class :character   1st Qu.:22.00  
 Median :2005   Median :4.000   Median :114.0   Mode  :character   Median :23.00  
 Mean   :2005   Mean   :3.999   Mean   :118.3                      Mean   :22.65  
 3rd Qu.:2010   3rd Qu.:6.000   3rd Qu.:179.0                      3rd Qu.:23.00  
 Max.   :2015   Max.   :7.000   Max.   :261.0                      Max.   :29.00  
      ap1                pb               st              carav              g         
 Min.   :0.00000   Min.   : 0.000   Min.   : 0.000   Min.   : -4.00   Min.   :  0.00  
 1st Qu.:0.00000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:  2.00   1st Qu.: 21.00  
 Median :0.00000   Median : 0.000   Median : 1.000   Median :  8.00   Median : 49.00  
 Mean   :0.08356   Mean   : 0.312   Mean   : 2.078   Mean   : 16.71   Mean   : 61.98  
 3rd Qu.:0.00000   3rd Qu.: 0.000   3rd Qu.: 3.000   3rd Qu.: 24.00   3rd Qu.: 95.00  
 Max.   :7.00000   Max.   :14.000   Max.   :17.000   Max.   :177.00   Max.   :270.00  
  Conference       
 Length:4895        
 Class :character   
 Mode  :character   
```

b. Data Exploration

Using data exploration to understand what is happening is important throughout the pipeline and is not limited to this step. However, it is important to use some exploration early on to make sure you understand your data. You must at least consider the distributions of each variable and at least some of the relationships between pairs of variables.

Numerical variable: year, ‘rnd’, pick, age, ‘ap1’, ‘pb’, ‘st’, ‘carav’, ‘g’

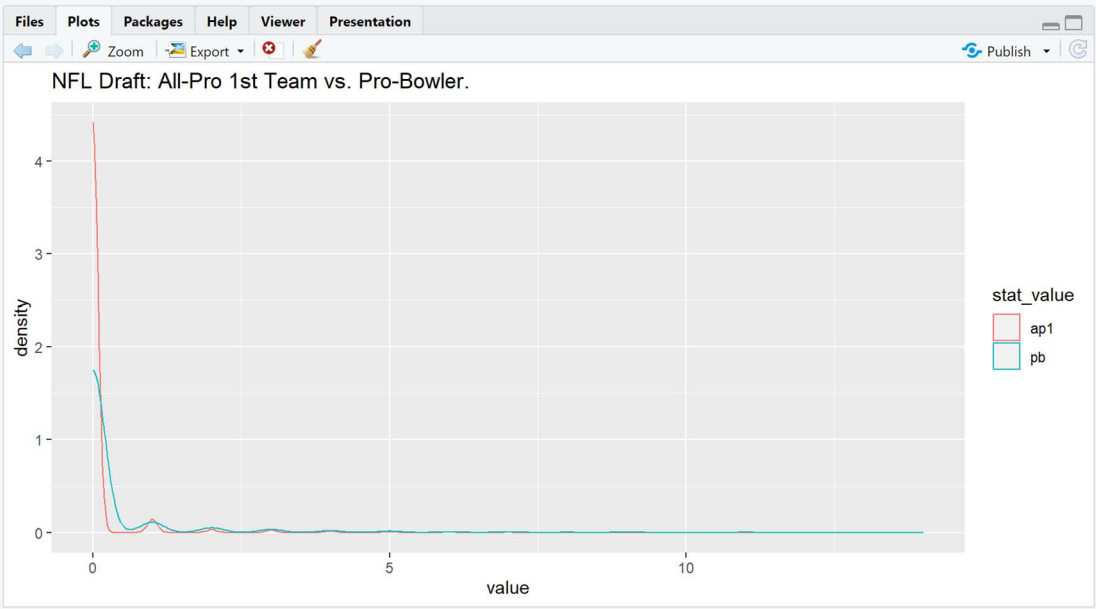
- ‘rnd’ – Round: 7 rounds of NFL draft.
- Pick: Picking order among all the players during the draft season.

- ‘ap1’ – All-Pro 1st: Number of times a player got 1st picked to any teams (top choice /All-Pro). Which means the best player at given position at that given season.
- ‘pb’ – Pro-bowler: Number of times the player was a Pro-bowler. Kind of like All-pro but the player is being chosen based more on popularity and audience preference, rather than stat.
- ‘st’ – Starter: Number of seasons the player was his team's primary starter at his posi tion, rather than bench players.
- ‘carav’ – Career Approximate Value: The seasonal value of a player at given position at that given year.
- ‘g’ – Games played: Number of games played.

- **Using Density plot to visualize the correlation between All-Pro 1 st team and Pro-Bowler.**

- From the graph, we can see that player who got high vote for All-Pro 1st Team. Also get high vote for Pro-Bowler, and vice versa.

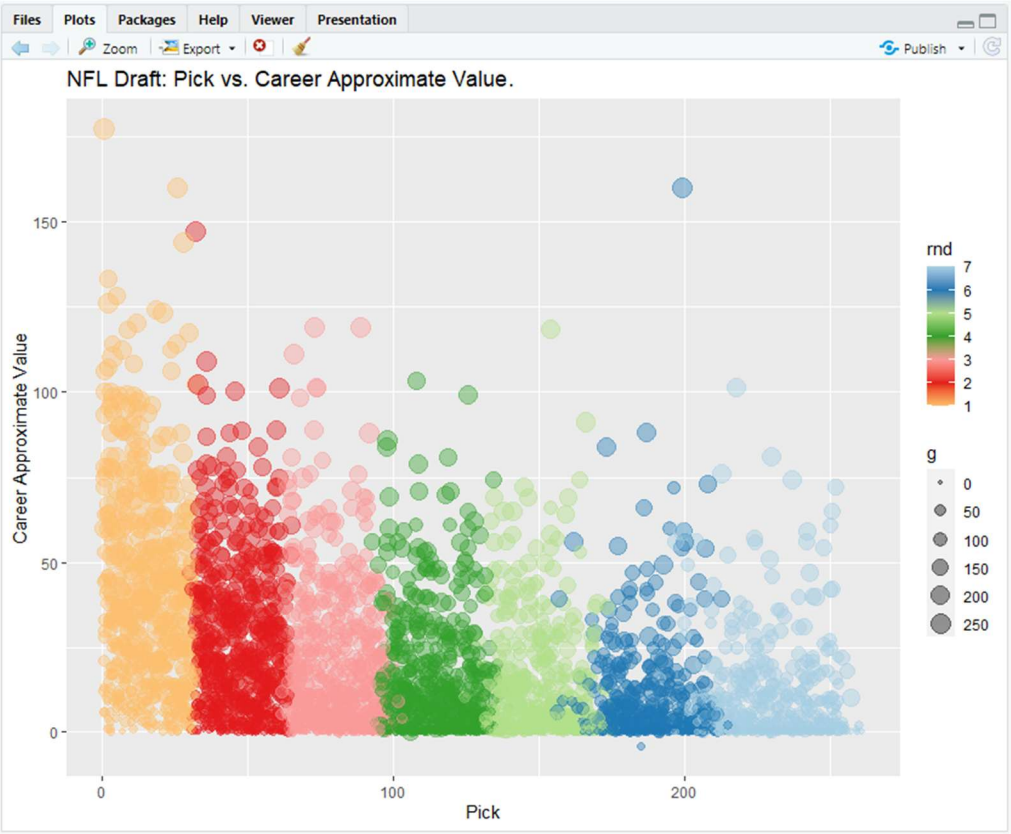
```
stat_value <- nflData %>% pivot_longer(cols = c("ap1", "pb"),
        names_to="stat_value", values_to="value")
ggplot(stat_value, aes(x=value, colour=stat_value))
        + geom_density()
        + ggtitle('NFL Draft: All-Pro 1st Team vs. Pro-Bowler.')
```



- **Using Scatterplot to visualize the correlation between Pick order and Career AV.**

- ✍ From the graph, we can see that higher Career Approximate Value leads to higher chance to be in the 1st draft. Which also means, players got drafted from later ro unds have a smaller number of played games and smaller/ sparsely distributed Career AV.
- ✍ The players are equally distributed among 7 rounds. Higher number of games played corresponds to higher Career AV.

```
ggplot(nflData, aes(x = pick, y = carav, color = rnd))
        + geom_point(aes(size = g), alpha = 0.4) + xlab("Pick")
        + ylab("Career Approximate Value")
        + ggtitle("NFL Draft: Pick vs. Career Approximate Value.")
        + labs(color = "rnd", size = "g")
        + scale_color_distiller(palette = "Paired")
```

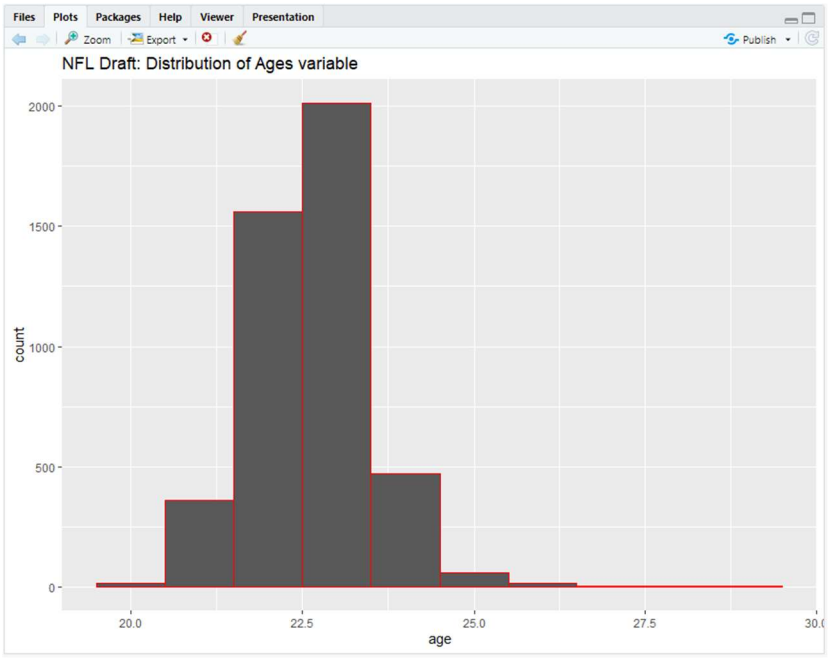


Using Histogram to visualize 'age' variable.

- ✍ Majority of players got drafted at 22-23 years old – graduating college.

```
ggplot(nflData, aes(age))
        + geom_histogram(binwidth = 1, color="red")
```

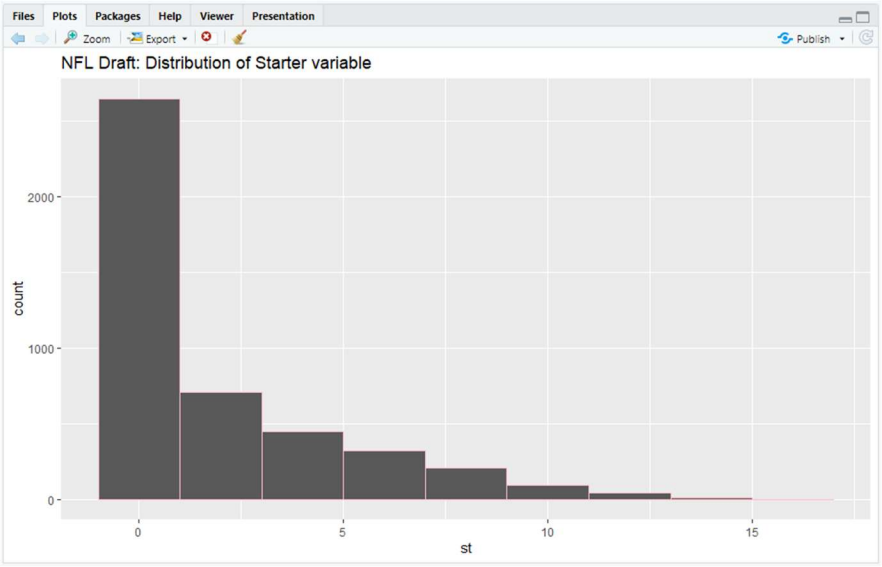
```
        + ggtitle('NFL Draft: Distribution of Ages variable')
```



- **Using Histogram to visualize 'st' Starter variable.**

✍ Majority of players were primary starter at their position 1st year they got draft, and then significantly decrease.

```
ggplot(nflData, aes(st))
        + geom_histogram(binwidth = 2, color="pink")
        + ggtitle('NFL Draft: Distribution of Starter variable')
```



Categorical variable: 'pos' position and conference type.

- ✍ I want to see if each conference draft positions differently. Is there preference for certain positions?
  - **Using Bar graph distribution for each conference with their draft positions.**
- ✍ Based on the graph, we can see that both conferences have almost the same draft ratio.

```
nflData %>% group_by(pos) %>% summarise("count" = n())
```

```
# A tibble: 24 × 2
   pos   count
   <chr> <int>
 1 C       120
 2 CB       24
 3 DB      934
 4 DE      450
 5 DL        1
 6 DT      387
 7 FB       61
 8 FS        3
 9 G       279
10 ILB      10
# … with 14 more rows
# ℹ Use `print(n = ...)` to see more rows
```

```
ggplot(nflData, aes(x=Conference, fill= pos))
        + geom_bar(position="stack")
        + ggtitle('NFL Draft: Position by Conference')
```

c. Data Cleaning

- ✍ Since there were so many missing values from the original dataset. I have cleaned it at the beginning. Just double check.

```
sum(is.na(nflData))
[1] 0
```

- ✍ I used boxplot to check for outliers. There are couple variables have outliers. This is very common because sport performance usually has good indicator at the very first few years then gradually decrease.
- ✍ Overall, I decided to do data normalization on these variables:
  - 'age': bin, and smooth by median.
  - 'ap1', 'pb', 'st': Min-max normalization: [1,10]
  - 'carav': z-score normalization

'age' variable

Chosen v numerical variable: ages → ages_bins

Using equal width: N (bins count) = 2

B (max value of 'ages') = 29 | A (min value of 'ages') = 20

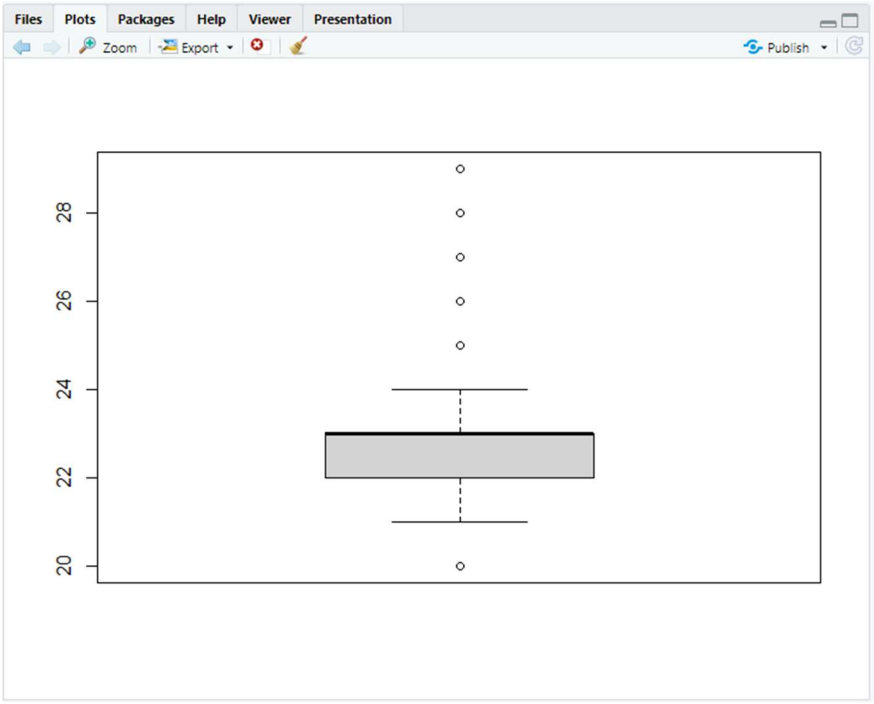
New value: ☞ Less_23_Age range: [20-23]

☞ Greater_23_Age range [23-29]

Reason: Majority of player get drafted fresh out of college between 22-23 years old. After 23 years-old, potentially they were agent-free, and should be accumulated into the same group due to the likelihood of getting drafted low. In fact, 'age' variable median is 23; mean is 22.64.

```
nflData_1 <- nflData
nflData_1 <- nflData_1 %>% mutate(age_bins = cut(age,
        breaks=c(-Inf, 23, Inf), labels=c("<23 Age",">23 Age")))

Less_23_Age <- nflData_1 %>% filter(age_bins == "<23 Age") %>%
        mutate(ages = median(age))
Greater_23_Age <- nflData_1 %>% filter(age_bins == ">23 Age") %>%
        mutate(ages = median(age))
```

```
nflData_1 <- bind_rows(list(Less_23_Age,Greater_23_Age))

boxplot(nflData$age)
```

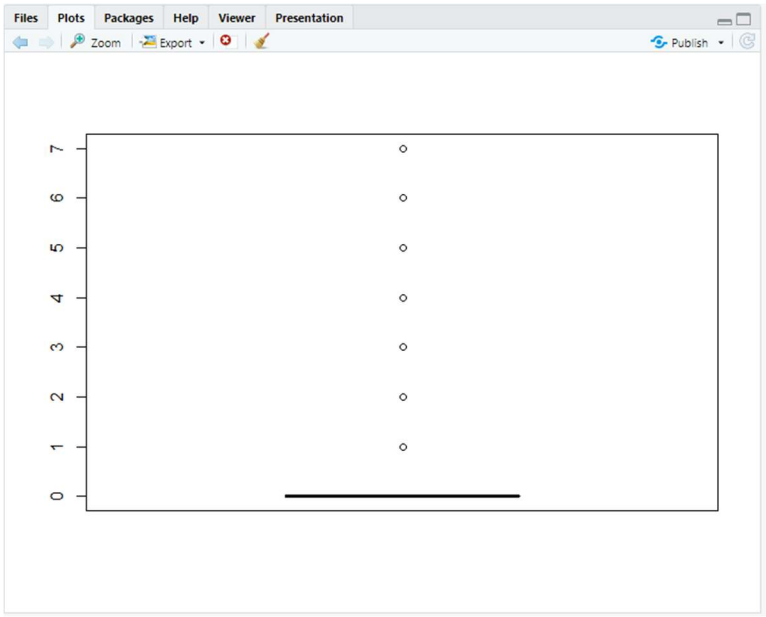



```
summary(nflData_1$ages)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.00   23.00   23.00   23.12   23.00   24.00

nflData_1 <- nflData_1%>% select(-c("age", "age_bins"))
#Remove 'age_bins' and original 'age' values.
#Using nflData_1 data frame moving forward.
```

'ap1' variable

```
norm_minmax <- function(x,new_max=1,new_min=0){(((x-min(x))
       *(new_max-new_min))/(max(x)-min(x)))+new_min}

normalise_ap1 <- as.data.frame(lapply(nflData["ap1"], norm_minmax))
summary(normalise_ap1$ap1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.01194 0.00000 1.00000

nflData_1$ap1 <- normalise_ap1$ap1
```

```
boxplot(nflData$ap1)
```



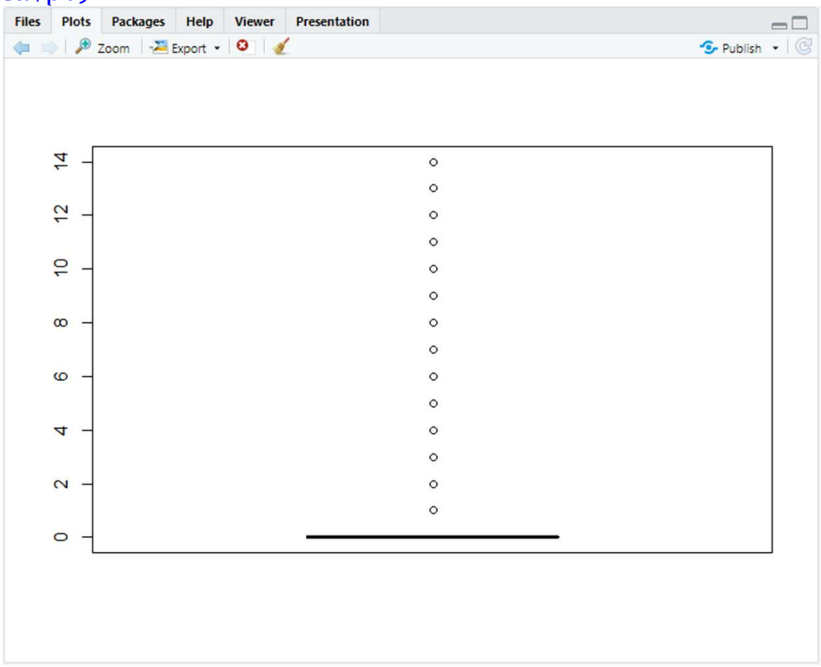
‘pb’ variable

```
normalise_pb <- as.data.frame(lapply(nflData["pb"], norm_minmax))
summary(normalise_pb$pb)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.02228 0.00000 1.00000
```

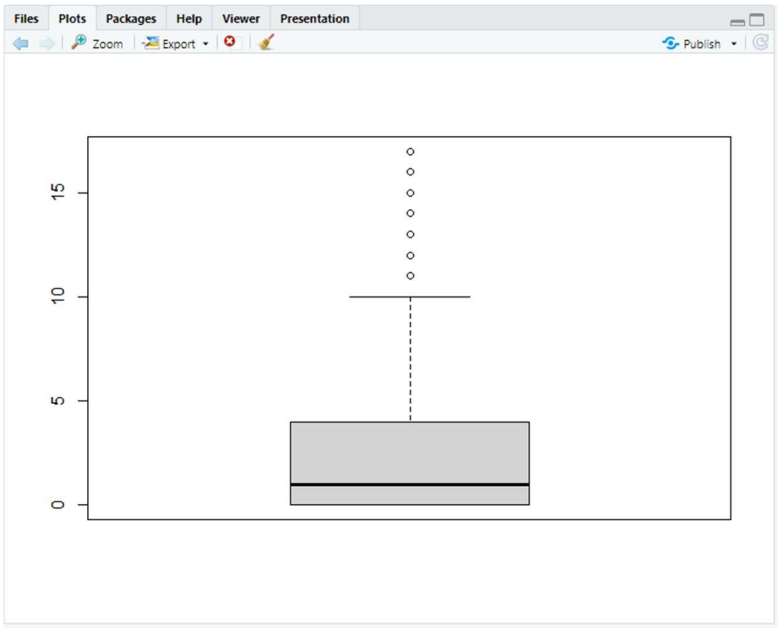
```
nflData_1$pb <- normalise_pb$pb
```

```
boxplot(nflData$pb)
```



‘st’ variable

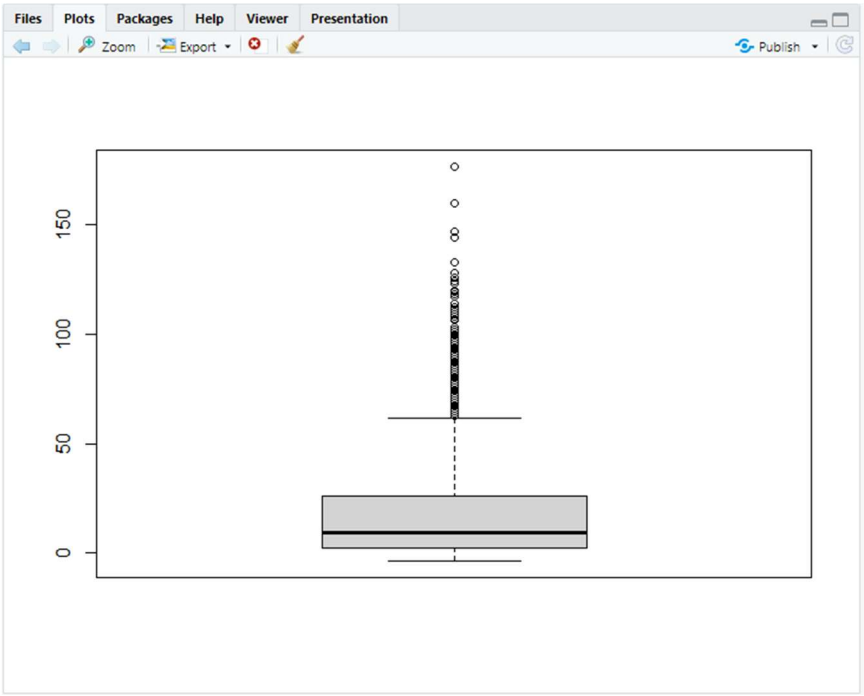
```
normalise_st <- as.data.frame(lapply(nflData["st"], norm_minmax))
summary(normalise_st$st)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.05882 0.12226 0.17647 1.00000

nflData_1$st <- normalise_st$st

boxplot(nflData$st)
```



‘carav’ variable

```
norm_zscore<-function(x){((x-mean(x))/sd(x))}

normalise_carav <- as.data.frame(lapply(nflData["carav"], norm_zscore))
summary(normalise_carav$carav)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.9610 -0.6827 -0.4043  0.0000  0.3380  7.4364

nflData_1$carav <- normalise_carav$carav
```

```
boxplot(nflData$carav)
```




```
head(nflData_1)
```

	year	rnd	pick	pos	ap1	pb	st	carav	g	Conference	ages
1	2015	1	2	QB	0	0.00000000	0.11764706	-0.3990474	23	AFC	23
2	2015	1	3	OLB	0	0.00000000	0.00000000	-0.8099976	10	AFC	23
3	2015	1	4	WR	0	0.07142857	0.05882353	-0.3990474	26	AFC	23
4	2015	1	6	DE	0	0.00000000	0.11764706	-0.3990474	26	AFC	23
5	2015	1	12	NT	0	0.00000000	0.11764706	-0.5360308	27	AFC	23
6	2015	1	14	WR	0	0.00000000	0.00000000	-0.6730142	24	AFC	23

```
summary(nflData_1)
```

```
      year          rnd             pick          pos                ap1                pb          
 Min.   :1994   Min.   :1.000   Min.   :  1.0   Length:4895        Min.   :0.00000   Min.   :0.00000  
 1st Qu.:1999   1st Qu.:2.000   1st Qu.: 56.0   Class :character   1st Qu.:0.00000   1st Qu.:0.00000  
 Median :2005   Median :4.000   Median :114.0   Mode  :character   Median :0.00000   Median :0.00000  
 Mean   :2005   Mean   :3.999   Mean   :118.3                      Mean   :0.01194   Mean   :0.02228  
 3rd Qu.:2010   3rd Qu.:6.000   3rd Qu.:179.0                      3rd Qu.:0.00000   3rd Qu.:0.00000  
 Max.   :2015   Max.   :7.000   Max.   :261.0                      Max.   :1.00000   Max.   :1.00000  
       st               carav                g          Conference            ages      
 Min.   :0.00000   Min.   :-0.9610   Min.   :  0.00   Length:4895        Min.   :23.00  
 1st Qu.:0.00000   1st Qu.:-0.6827   1st Qu.: 21.00   Class :character   1st Qu.:23.00  
 Median :0.05882   Median :-0.4043   Median : 49.00   Mode  :character   Median :23.00  
 Mean   :0.12226   Mean   : 0.0000   Mean   : 61.98                      Mean   :23.13  
 3rd Qu.:0.17647   3rd Qu.: 0.3380   3rd Qu.: 95.00                      3rd Qu.:23.00  
 Max.   :1.00000   Max.   : 7.4364   Max.   :270.00                      Max.   :24.00  
```

d. Data Preprocessing

Making dummy variable for 'pos' variable.

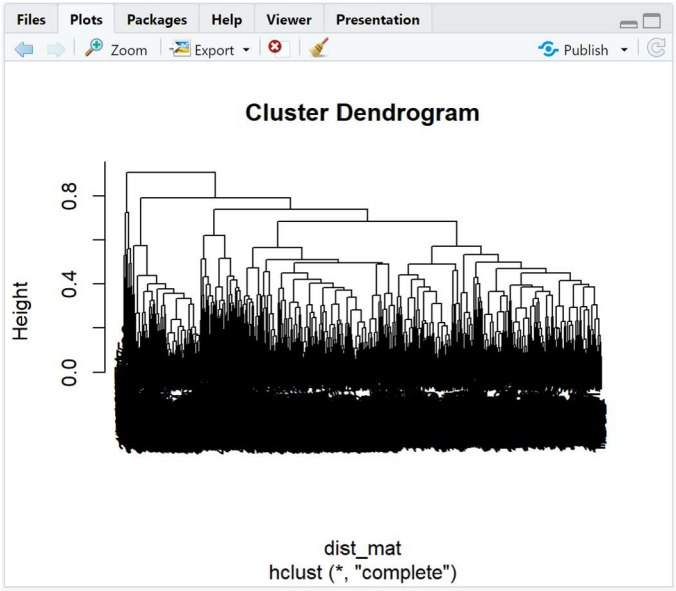
```
nflData_df <- nflData_1
library(lattice)
library(caret)
library(ggplot2)

nflData_df$Conference <- as.factor(nflData_df$Conference)
dummy <- dummyVars(Conference ~ ., data = nflData_df)
nflData_dummies <- as.data.frame(predict(dummy, newdata = nflData_df))
#4895 observations,33 variables.

summary(nflData_dummies)
```

```
      year          rnd             pick             posC              posCB             posDB             posDE             posDL             posDT             posFB        
 Min.   :1994   Min.   :1.000   Min.   :  1.0   Min.   :0.00000   Min.   :0.000000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000000   Min.   :0.00000   Min.   :0.00000  
 1st Qu.:2000   1st Qu.:2.000   1st Qu.: 52.0   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000000   1st Qu.:0.00000   1st Qu.:0.00000  
 Median :2005   Median :4.000   Median :106.0   Median :0.00000   Median :0.000000   Median :0.0000   Median :0.00000   Median :0.0000000   Median :0.00000   Median :0.00000  
 Mean   :2005   Mean   :3.845   Mean   :112.8   Mean   :0.02424   Mean   :0.005116   Mean   :0.1926   Mean   :0.09119   Mean   :0.0002224   Mean   :0.07629   Mean   :0.01179  
 3rd Qu.:2010   3rd Qu.:6.000   3rd Qu.:170.0   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000000   3rd Qu.:0.00000   3rd Qu.:0.00000  
 Max.   :2015   Max.   :7.000   Max.   :261.0   Max.   :1.00000   Max.   :1.000000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000000   Max.   :1.00000   Max.   :1.00000  
     posFS              posG             posILB              posK              posLB             posLS              posNT              posOL              posOLB        
 Min.   :0.0000000   Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Min.   :0.0000   Min.   :0.0000000   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000  
 1st Qu.:0.0000000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.000000  
 Median :0.0000000   Median :0.00000   Median :0.000000   Median :0.000000   Median :0.0000   Median :0.0000000   Median :0.000000   Median :0.000000   Median :0.000000  
 Mean   :0.0006673   Mean   :0.05672   Mean   :0.002224   Mean   :0.007117   Mean   :0.1274   Mean   :0.0002224   Mean   :0.002224   Mean   :0.002447   Mean   :0.004226  
 3rd Qu.:0.0000000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:0.0000000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.000000  
 Max.   :1.0000000   Max.   :1.00000   Max.   :1.000000   Max.   :1.000000   Max.   :1.0000   Max.   :1.0000000   Max.   :1.000000   Max.   :1.000000   Max.   :1.000000  
     posP              posQB              posRB              posS              posSS              posT              posTE              posWR              ap1               pb         
 Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000000   Min.   :0.000000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000   Min.   :0.000   Min.   :0.00000  
 1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.00000  
 Median :0.00000   Median :0.00000   Median :0.00000   Median :0.0000000   Median :0.000000   Median :0.00000   Median :0.00000   Median :0.000   Median :0.000   Median :0.00000  
 Mean   :0.00734   Mean   :0.04159   Mean   :0.08519   Mean   :0.0002224   Mean   :0.001557   Mean   :0.07407   Mean   :0.06228   Mean   :0.123   Mean   :0.013   Mean   :0.02399  
 3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000000   3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.00000  
 Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000000   Max.   :1.000000   Max.   :1.00000   Max.   :1.00000   Max.   :1.000   Max.   :1.000   Max.   :1.00000  
       st               carav                g               ages      
 Min.   :0.00000   Min.   :-0.9926   Min.   :  0.00   Min.   :23.00  
 1st Qu.:0.00000   1st Qu.:-0.7187   1st Qu.: 24.00   1st Qu.:23.00  
 Median :0.05882   Median :-0.3990   Median : 53.00   Median :23.00  
 Mean   :0.13056   Mean   : 0.0000   Mean   : 64.87   Mean   :23.12  
 3rd Qu.:0.23529   3rd Qu.: 0.3772   3rd Qu.: 97.00   3rd Qu.:23.00  
 Max.   :1.00000   Max.   : 7.2720   Max.   :270.00   Max.   :24.00  
```

e. Clustering

```
library(stats)
library(factoextra)
library(ggplot2)
```

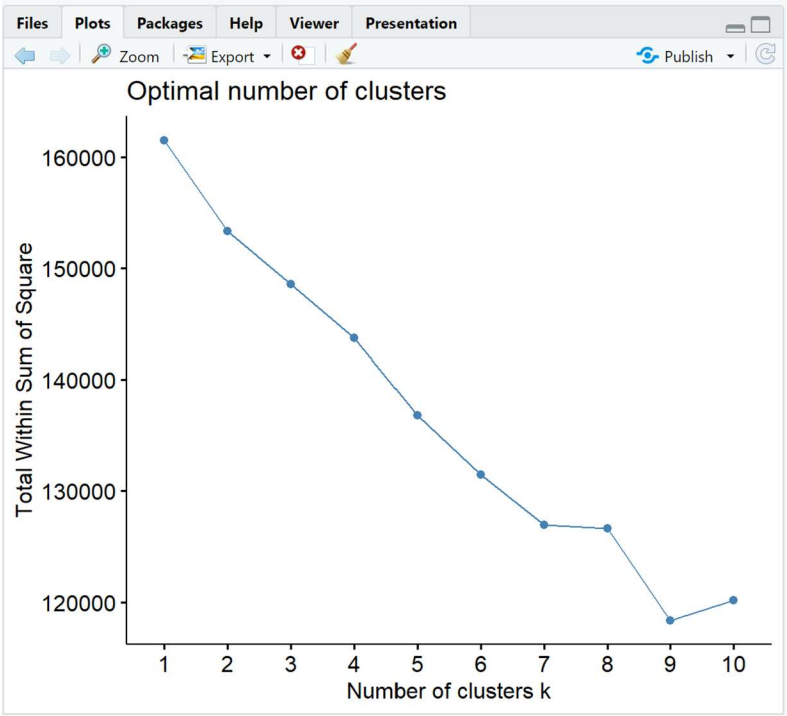
- I did both HAC and k-means methods.
- For HAC method, I used daisy() function and metric = "gower" (can be used with both categorical and numerical data), since I want to keep categorical variable 'pos'. HAC method suggests k =2
- However, I decided to move forward with k-means since the dendrogram from HAC methods is too 'clustered' at the bottom.



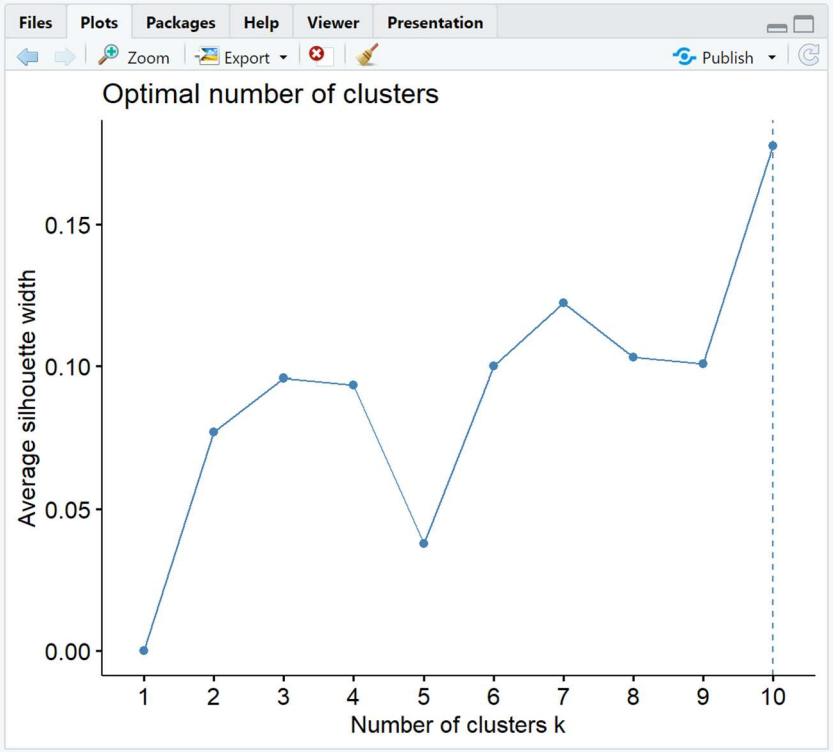
K-means method

- ✍ Moving forward using `nflData_dummies` data set.

```
set.seed(1997)
preproc <- preProcess(nflData_dummies, method=c("center", "scale"))
predictors <- predict(preproc, nflData_dummies)

fviz_nbclust(predictors, kmeans, method = "wss")
#Find k
```



```
fviz_nbclust(predictors, kmeans, method = "silhouette")
```

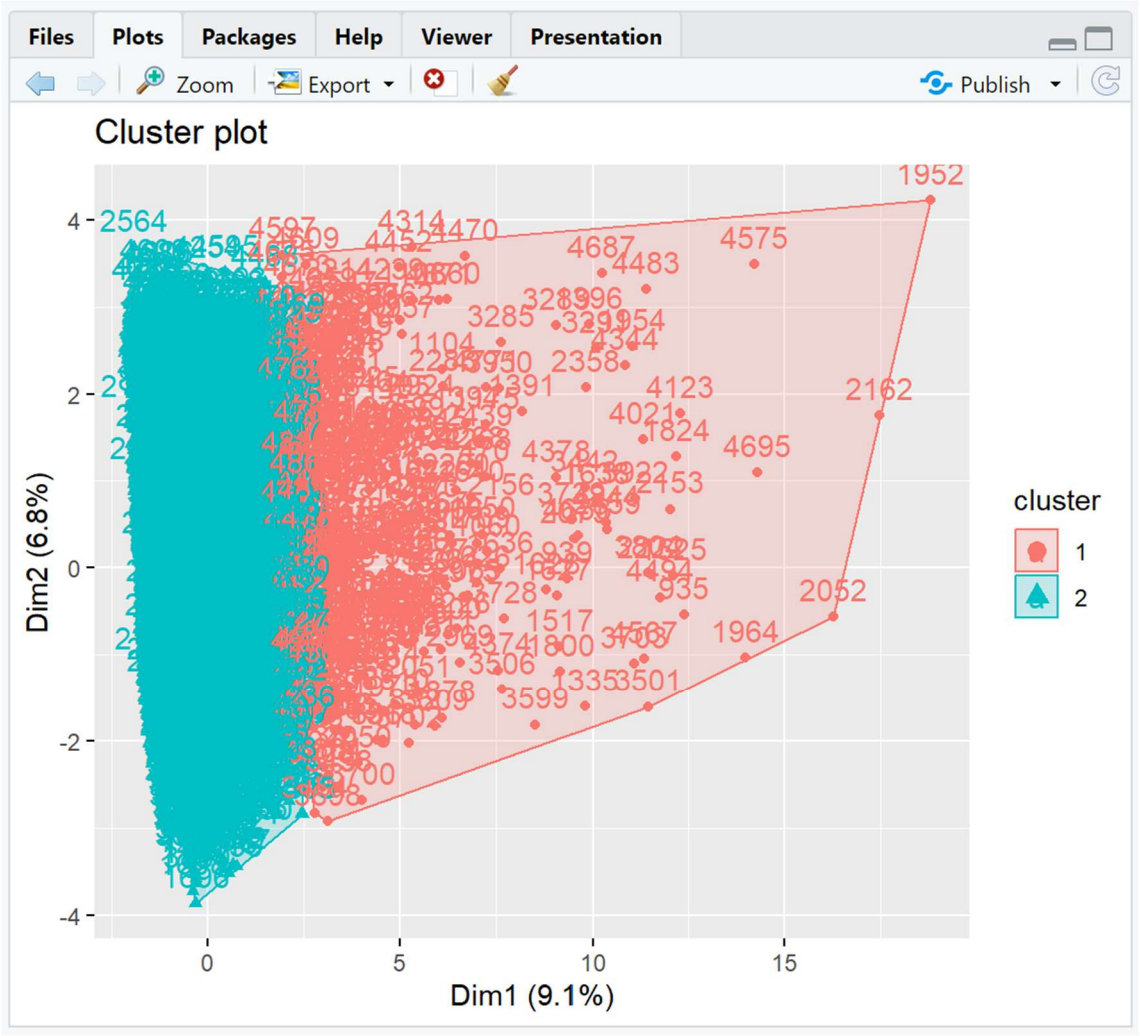


- ✍ Since 'wss' method suggests k=2 as well. Even though 'silhouette' method suggests k=9, the fact that we have 2 class labels and HAC method also gives k=2. I will move forward with k=2

```
fit <- kmeans(predictors, centers = 2, nstart = 25)
fit
```

```
K-means clustering with 2 clusters of sizes 360, 4535

Cluster means:
         year          rnd         pick         posC        posCB        posDB
1 -0.23524582 -0.081268898 -0.083850091 -0.014823719  0.009341923 -0.026086466
2  0.01867442  0.006451335  0.006656237  0.001176745 -0.000741586  0.002070811
         posDE        posDL        posDT        posFB        posFS         posG
1  0.075991645 -0.014293008 -0.015045802 -0.037210704 -0.024761277  0.017744188
2 -0.006032413  0.001134616  0.001194375  0.002953882  0.001965614 -0.001408579
        posILB         posK        posLB        posLS        posNT        posOL
1  0.077787802  0.0041166240 -0.018686047 -0.014293008 -0.049568171 -0.05159747
2 -0.006174996 -0.0003267882  0.001483347  0.001134616  0.003934849  0.00409594
        posOLB         posP        posQB        posRB         posS        posSS
1  0.026919470 -0.053559043 -0.016888885  0.0062570937 -0.014293008  0.035661707
2 -0.002136937  0.004251655  0.001340683 -0.0004967042  0.001134616 -0.002830918
         posT         posTE        posWR        ap1         pb          st       carav
1  0.077416034 -0.0123131872 -0.034950009  2.0509848  2.7339636  2.2450077  2.5340242
2 -0.006145484  0.0009774526  0.002774422 -0.1628125 -0.2170291 -0.1782145 -0.2011574
           g         ages
1  0.117095324  0.080845087
2 -0.009295329 -0.006417692
```

```
fviz_cluster(fit, data = predictors)
```



f. Classification

- I moved forward with the `nflData_dummies` dataset, but the accuracy result is very low (around 50%), and SVM was not generate result. My assumption is the dummy 'pos' variable creates this issue, so I removed it.
- Then I still receive 50% accuracy result, testing both normalized and original dataset (remove "pos" categorical variable). **Therefore, I will show the step using the original dataset.**
- I will use kNN (tune k) and SVM (tune C) for classification.
- First, I will run PCA.

Run PCA

```
nflData_2 <- nflData %>% select(-c("pos"))
#4895 observations, 10 variables.
```

- ✍ Now the data set has all numerical variables, thus no need to convert to dummies.
- ✍ 'pos' – position variable was resulting 25 numerical dummy variables.

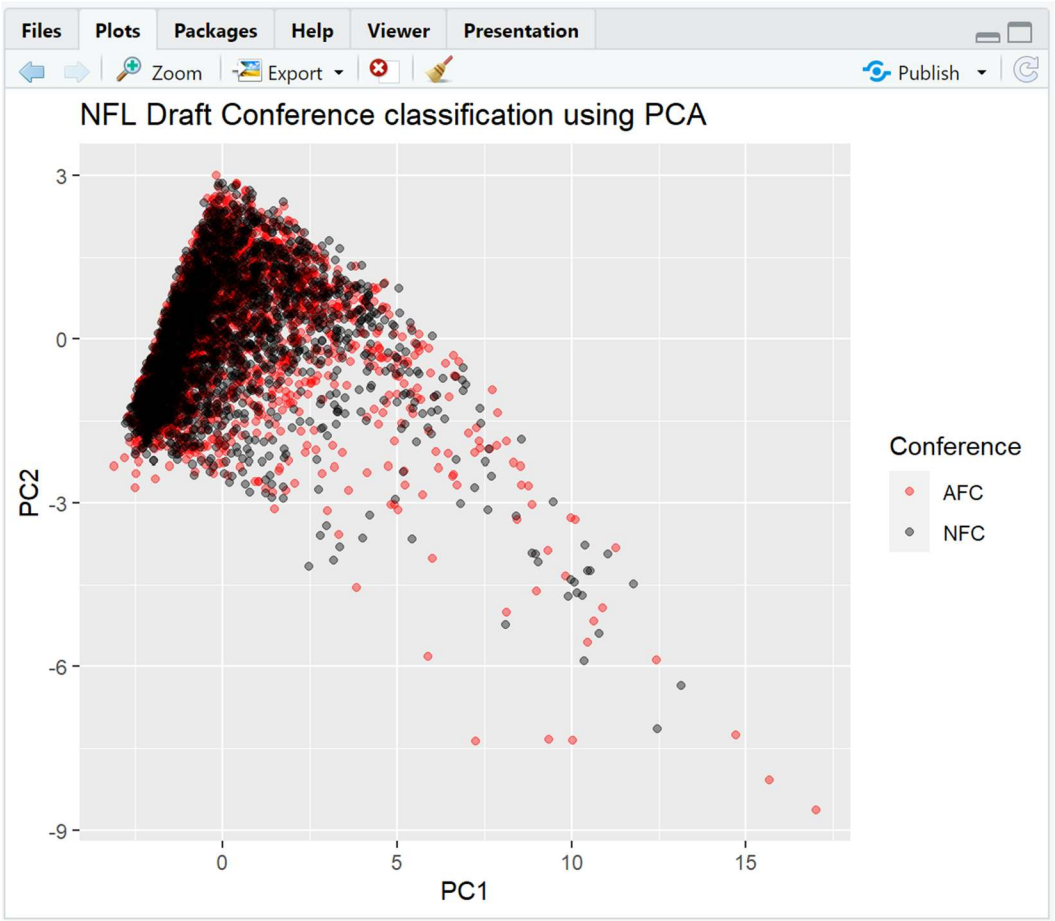
```
nflData_2 <- nflData_2 %>% select(-c("Conference"))
#Remove class label 'Conference'
predictors <- nflData_2

set.seed(1234)
preproc <- preProcess(predictors, method=c("center", "scale"))
predictors <- predict(preproc, predictors)
# Normalizing, scaling data. And fit 'predictors' data frame.


head(predictors)
# A tibble: 6 × 9
   year   rnd  pick    age    ap1     pb      st  carav      g
  <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>
1  1.64 -1.50 -1.62 -1.88  -0.174 -0.273 -0.0263 -0.358 -0.767
2  1.64 -1.50 -1.61 -1.88  -0.174 -0.273 -0.698  -0.775 -1.02
3  1.64 -1.50 -1.59 -1.88  -0.174  0.602 -0.362  -0.358 -0.708
4  1.64 -1.50 -1.57 -1.88  -0.174 -0.273 -0.0263 -0.358 -0.708
5  1.64 -1.50 -1.48 -0.743 -0.174 -0.273 -0.0263 -0.497 -0.689
6  1.64 -1.50 -1.45 -0.743 -0.174 -0.273 -0.698  -0.636 -0.748


pca = prcomp(predictors)
summary(pca)
```

- ✍ From the results shown below, the variance is captured by almost all principal components at +99% variance (8 PCs). Thus, I will use the original dataset itself for classification.

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
Standard deviation     2.055 1.2493 1.0724 0.94050 0.86448 0.45245 0.3946 0.25042
Proportion of Variance 0.469 0.1734 0.1278 0.09828 0.08304 0.02275 0.0173 0.00697
Cumulative Proportion  0.469 0.6425 0.7702 0.86853 0.95156 0.97431 0.9916 0.99858
                          PC9
Standard deviation     0.11309
Proportion of Variance 0.00142
Cumulative Proportion  1.00000
```

```
nfl.pca = as.data.frame(pca$x)
nfl.pca$Conference <- nflData$Conference
```

```
ggplot(data = nfl.pca, aes(x = PC1, y = PC2, color = Conference))
        + geom_point(alpha= 0.4)
        + ggtitle("NFL Draft Conference classification using PCA")
        + scale_color_manual(values=c('red','black'))
```



SVM (tune C) classification method

```
grid <- expand.grid(C = seq(1,2,0.1))
#Set grid search
ctrl <- trainControl(method="cv", number = 10)

nflData_2$Conference <- nflData$Conference
#Put back 'Conference' class label

svm_grid <- train(Conference ~., data = nflData_2, method = "svmLinear",
       trControl = train_control, tuneGrid = grid)
svm_grid
```

```
Support Vector Machines with Linear Kernel

4895 samples
   9 predictor
   2 classes: 'AFC', 'NFC'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 4405, 4406, 4405, 4406, 4405, 4406, ...
Resampling results across tuning parameters:

  C    Accuracy   Kappa
  1.0  0.5056125  0.011210809
  1.1  0.5052043  0.010391240
  1.2  0.5049998  0.009981588
  1.3  0.5056129  0.011206041
  1.4  0.5045925  0.009165122
  1.5  0.5056145  0.011212733
  1.6  0.5056145  0.011212733
  1.7  0.5060227  0.012029060
  1.8  0.5060227  0.012029060
  1.9  0.5064317  0.012847250
  2.0  0.5062276  0.012438999

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was C = 1.9.
```

✍ C = 1.9 give the best accuracy result of approximately 50%

kNN (tune k) classification method

```
set.seed(9876)
ctrl <- trainControl(method="cv", number = 10)


knnFit <- train(Conference ~ ., data = nflData_2, method = "knn",
        trControl = ctrl, preProcess = c("center","scale"), tuneLength = 15)
knnFit
```

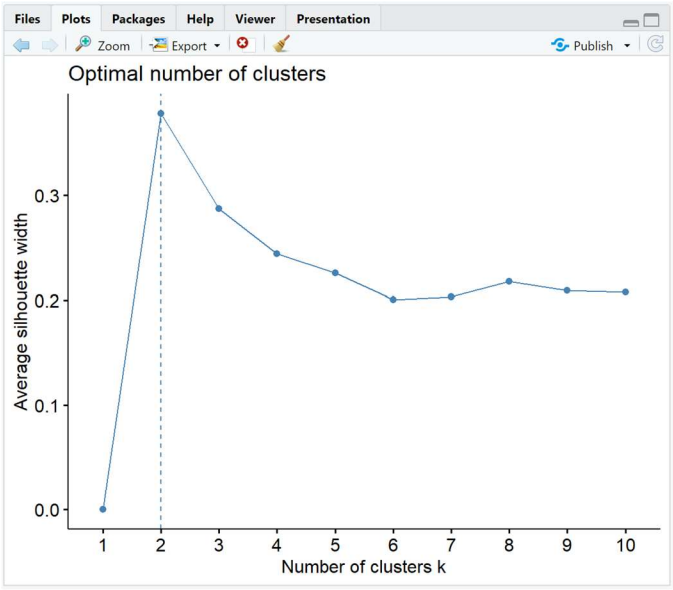
```
k-Nearest Neighbors

4895 samples
   9 predictor
   2 classes: 'AFC', 'NFC'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 4406, 4405, 4405, 4406, 4405, 4406, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.4935792  -0.0128240732
   7  0.5033843   0.0067679185
   9  0.4976625  -0.0046521810
  11  0.4980631  -0.0038561644
  13  0.5005146   0.0010483328
  15  0.4958207  -0.0083332043
  17  0.5001089   0.0002254017
  19  0.5033792   0.0067860248
  21  0.5027699   0.0055537464
  23  0.5039944   0.0079940512
  25  0.5082826   0.0165578816
  27  0.5117575   0.0235144617
  29  0.5054247   0.0108434888
  31  0.5052206   0.0104432066
  33  0.4998969  -0.0002063383

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 27.
```

✍ k = 27 gives the best accuracy result approximately 50%

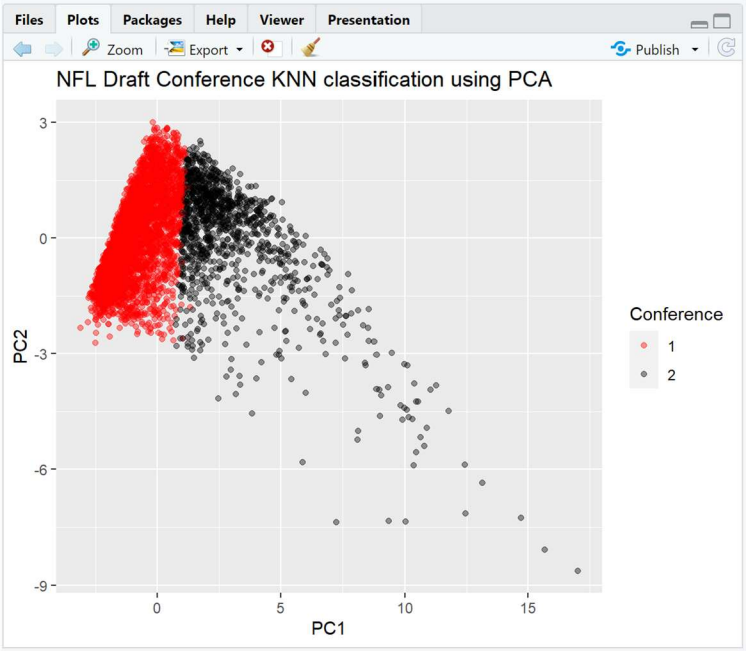
Use PCA again to visualize the labels for kNN and SVM.

kNN

```
fviz_nbclust(predictors, kmeans, method = "silhouette")
```

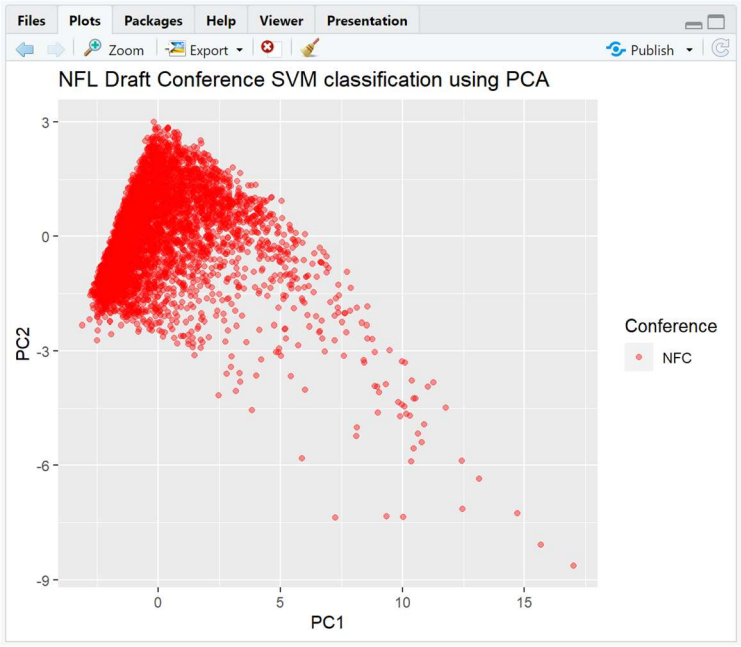



```
fit <- kmeans(predictors, centers = 2, nstart = 25)
nfl.pca$Conference = as.factor(fit$cluster)

ggplot(data = nfl.pca, aes(x = PC1, y = PC2, color = Conference))
        + geom_point(alpha= 0.4)
        + ggtitle("NFL Draft Conference KNN classification using PCA")
        + scale_color_manual(values=c('red','black'))
```



SVM

```
svm_predictors <- predict(svm_grid, predictors)
svm_predictor <- as.data.frame(svm_predictors)

nfl.pca$Conference <- svm_predictor$svm_predictors

ggplot(data = nfl.pca, aes(x = PC1, y = PC2, color = Conference))
        + geom_point(alpha= 0.4)
        + ggtitle("NFL Draft Conference SVM classification using PCA")
        + scale_color_manual(values=c('red','black'))
```



g. Evaluation

```
library(tibble)
library(bitops)
library(rattle)
library(pROC)
```

Confusion matrix (60/40)

```
set.seed(4000)
nflData$Conference <- as.factor(nflData$Conference)
nflData$pos <- as.factor(nflData$pos)
```

```
index = createDataPartition(y=nflData$Conference, p=0.6, list=FALSE)
train_nfl = nflData[index,]
test_nfl = nflData[-index,]

train_control = trainControl(method = "cv", number = 10)
tree <- train(Conference ~., data = train_nfl, method = "rpart",
        trControl = train_control)
pred_nfl <- predict(tree, test_nfl)

cm <- confusionMatrix(test_nfl$Conference, pred_nfl)

cm
```

```
Confusion Matrix and Statistics

          Reference
Prediction AFC NFC
       AFC 806 173
       NFC 820 158

               Accuracy : 0.4926
                 95% CI : (0.4702, 0.515)
    No Information Rate : 0.8309
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0152

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.4957
            Specificity : 0.4773
         Pos Pred Value : 0.8233
         Neg Pred Value : 0.1616
             Prevalence : 0.8309
         Detection Rate : 0.4119
   Detection Prevalence : 0.5003
      Balanced Accuracy : 0.4865

       'Positive' Class : AFC
```

Precision and Recall

```
metrics <- as.data.frame(cm$byClass)
metrics
```

```
                     cm$byClass
Sensitivity           0.4956950
Specificity           0.4773414
Pos Pred Value        0.8232891
Neg Pred Value        0.1615542
Precision             0.8232891
Recall                0.4956950
F1                    0.6188100
Prevalence            0.8308636
Detection Rate        0.4118549
Detection Prevalence  0.5002555
Balanced Accuracy     0.4865182
```

```
metrics[c("Precision"),]
[1] 0.8232891

> metrics[c("Recall"),]
[1] 0.495695

metrics[c("Specificity"),]
[1] 0.4773414

> metrics[c("F1"),]
[1] 0.61881

> metrics[c("Balanced Accuracy"),]
[1] 0.4865182
```

ROC plot

```
library(mlbench)

train_control = trainControl(method = "cv", number = 10)
dtree <- train(Conference ~., data = train_nfl, method = "rpart", trControl =
train_control)

dtree
CART

2938 samples
   9 predictor
   2 classes: 'AFC', 'NFC'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 2644, 2644, 2644, 2645, 2645, 2644, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.01089176  0.5078347   0.01533232
  0.01270706  0.5081760   0.01602697
  0.04288632  0.4931752  -0.01317591

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01270706.
```

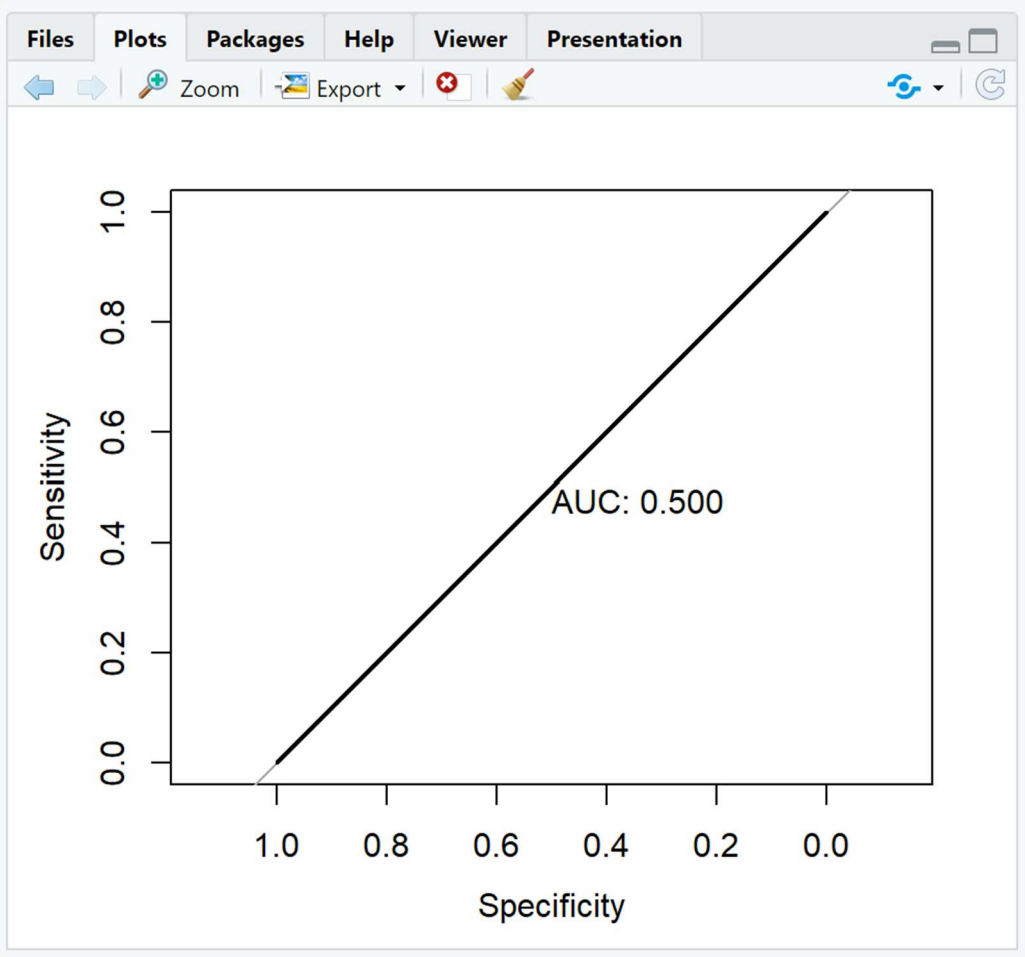
```
pred_nfl2 <- predict(dtree, test_nfl)
confusionMatrix(test_nfl$Conference, pred_nfl2)
```

```
Confusion Matrix and Statistics

          Reference
Prediction AFC NFC
       AFC 347 387
       NFC 364 370

               Accuracy : 0.4884
                 95% CI : (0.4626, 0.5143)
    No Information Rate : 0.5157
    P-Value [Acc > NIR] : 0.9828

                  Kappa : -0.0232

 Mcnemar's Test P-Value : 0.4221

            Sensitivity : 0.4880
            Specificity : 0.4888
         Pos Pred Value : 0.4728
         Neg Pred Value : 0.5041
             Prevalence : 0.4843
         Detection Rate : 0.2364
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.4884

       'Positive' Class : AFC
```

```
pred_prob <- predict(dtree, test_nfl, type = "prob")
head(pred_prob)
```

```
        AFC       NFC
1 0.4715026 0.5284974
2 0.5161290 0.4838710
3 0.5161290 0.4838710
4 0.5161290 0.4838710
5 0.5161290 0.4838710
6 0.4715026 0.5284974
```

```
roc_obj <- roc((test_nfl$Conference), pred_prob[,1])
plot(roc_obj, print.auc=TRUE)
```



Explain how these performance measures makes your classifier look compared to accuracy.

- ✍ After doing classification, it is clearly show that my model is not doing great with 'classification'. The ROC curve has AUC value exactly at 0.5 which indicates I could have made errors in building training algorithm.

- ✍ However, looking overall, my objective in analyzing this dataset is to find if there is a specific pattern, indicator, or performance stat; that can classify a football player drafted for a particular NFL Conference. This doesn't seem so.

h. Report

For part a. I merge two datasets; each dataset represents NFL teams for each Conference: National Football Conference (NFC) and American Football Conference (AFC). Then I did some research on football statistics to understand more about the data I am working on. Then I remove some data due to missing values and use it as final dataset. Move on to looking at data distribution, part b. I have plotted several graphs and decided to separate by categorical and numerical values. Then I transform 'age' value by binning (smooth by median), number of games played by z-scores, and 3 stats that represent votes by min-max normalization.

Move on clustering and classification, I did a lot of trials and errors and decided to show the best result. Overall, I got 50% accuracy result. Initially, I thought it was causing by the ‘pos’ position variable, this categorical variable produces 25 dummy variables. By this time, I start to think that maybe it should be 50% since my class label are the NFL conferences (AFC and NFC). The possibility of a player goes to either conference should be 50-50. These players pretty much have good/similar performance stats and both conferences recruit the same rate of players/positions per round.

Overall, I have learned most from doing this assignment. By using my own dataset, I have learned to read and understand these concepts thoroughly. Maybe I should have classified manually player's position using Approximate Value (AV). For example, quarter back position has higher AV than tight-ends position. By doing this, I could reduce number of dummy variables potentially. Another take away I learned from analyzing this dataset and doing this homework too, is that I cannot expect to have good result like the 'homework'. Raw data has so many characteristics which requires trials and errors, knowledge, experience, and intuition to reduce the expected result to make better decision.

Finally, I have learned how to use R properly. Maybe this homework result is not what I expected, but I completely understand the code and the algorithm. I can confidently explain every step and line of code.

i. Reflection

Data science in general is the studies of data by using a variety of methods to make better decision. There are two main branches; first, utilizing and analyzing data to make and predict better decisions (business management related purposes). Second branch is more on machine learning (algorithm improvement for data processing). Even though data can be spoken/analyzed by algorithm, it is still necessary to have human intuition involved to get the data 'makes sense'. Throughout the course, I have learned about the data mining process, what needed to be done when encountering a large dataset; by looking at variable's distribution, correlation, and missing values. I also learned about 3 data classification methods: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree. All are supervised learning machine algorithms. In addition, I also learned about clustering (unsupervised learning machine algorithm) with 2 methods: HAC and K-means. I also learned how to use R, and frankly say, it is very difficult. There are many ways to make inputs for the same purpose (with different packages and tools). By this time, I have known some basic R libraries (tidyverse, caret, dyplr, etc.). To verify the accuracy of a model, I can use test and train datasets in many different combinations. And specifically, from this Homework 5, my model was not resulting high accuracy value on neither classification methods. I realized that the datasets that I have been learning/working on throughout the class makes 'data science' seems very achievable, but it does not seem so. Lots of practicing, trials and errors to gain more knowledge and experience.