NFL Betting

Jordyn Raguckas

```r
setwd("~/Documents/STA 536/STA536 Final")
final <- read.csv("spreadspoke_scores.csv")
library(tidyverse)
library(caret)
library(GGally)
NFL<-final %>% #betting data only for 1979 season to 2018 season
  filter(schedule_season >=1979) %>%
  filter(schedule_season <= 2018)

#created over/under/push variable to find the betting result
NFL$over_under_result <-ifelse(NFL$score_home + NFL$score_away ==
NFL$over_under_line, 'P',
                                  ifelse(NFL$score_home + NFL$score_away >
NFL$over_under_line,'O','U'))

nflTeams<- read.csv("nfl_teams.csv")

team_names<- nflTeams$team_name
team_ids<- nflTeams$team_id

# Add id variables to get spread info since favorite was in ID form.

NFL$team_home_id <- NA
NFL$team_away_id <- NA

for (i in 1:nrow(NFL)) {
        for(j in 1:length(team_ids)){
                if(NFL$team_home[i]==team_names[j]){
                        NFL$team_home_id[i]<-team_ids[j]
                }
        }
}

for (i in 1:nrow(NFL)) {
        for(j in 1:length(team_ids)){
                if(NFL$team_away[i]==team_names[j]){
                        NFL$team_away_id[i]<-team_ids[j]
                }
        }
}

divisions <- nflTeams$team_division
```

```r
NFL$home_division <- NA
NFL$away_division <- NA

for (i in 1:nrow(NFL)) {
        for(j in 1:length(divisions)){
                if(NFL$team_home_id[i]==team_ids[j]){
                        NFL$home_division[i]<-divisions[j]
                }
        }
}

for (i in 1:nrow(NFL)) {
        for(j in 1:length(divisions)){
                if(NFL$team_away_id[i]==team_ids[j]){
                        NFL$away_division[i]<-divisions[j]
                }
        }
}

NFL$divisional_game <- ifelse(NFL$home_division==NFL$away_division, 1, 0)

##created underdog id variable

NFL$team_underdog_id <- ifelse(NFL$team_favorite_id ==
NFL$team_home_id,NFL$team_away_id, NFL$team_home_id)

NFL$spread_cover_result <- ifelse(NFL$team_favorite_id == NFL$team_home_id &
NFL$score_home +        NFL$spread_favorite == NFL$score_away, 2,
                                                   ifelse(NFL$team_favorite_id ==
NFL$team_away_id & NFL$score_away +        NFL$spread_favorite ==
NFL$score_home, 2,
                                                          ifelse(NFL$team_favorite_id
== NFL$team_home_id & NFL$score_home +        NFL$spread_favorite >
NFL$score_away,1,

ifelse(NFL$team_favorite_id == NFL$team_away_id & NFL$score_away +
NFL$spread_favorite > NFL$score_home,1 , 0))))

nflStadiums <- read.csv("nfl_stadiums2.csv")
stadiums_name <- nflStadiums$stadium_name
stadiums_type<- nflStadiums$stadium_type
stadiums_surface<-nflStadiums$stadium_surface
stadiums_capacity<- nflStadiums$stadium_capacity
NFL$stadium_type <- NA
NFL$stadium_surface <- NA
NFL$stadium_elevation <- NULL
NFL$stadium_capacity <- NA
```

```
for (i in 1: length(stadiums_surface)){
  if(stadiums_surface[i] == ""){
  stadiums_surface[i] = "Grass"
  }
}

for (i in 1:nrow(NFL)) {
        for(j in 1:length(stadiums_name)){
                if(NFL$stadium[i] == stadiums_name[j]){
                        NFL$stadium_type[i]<-stadiums_type[j]
                        NFL$stadium_capacity[i]<-stadiums_capacity[j]
                        NFL$stadium_surface[i]<-stadiums_surface[j]
                }
        }
}

#How many times from 1979 to 2018 a team has covered the spread. If you bet
on them as the favorite you won your bet.
#Panthers, Jaguars, Ravens, Texans all partial outliers because they were not
teams when the dataset started.
#Panthers (1995), Jaguars(1995), Ravens(1996), Texans (2002)

spread_count<-dplyr::summarize(group_by(filter(NFL, spread_cover_result ==
1), team_favorite_id), count = n())

arrange(spread_count, desc(count))

## # A tibble: 32 × 2
##    team_favorite_id count
##    <chr>            <int>
##  1 PIT                217
##  2 SF                 216
##  3 NE                 215
##  4 DAL                198
##  5 GB                 197
##  6 DEN                193
##  7 PHI                181
##  8 MIN                169
##  9 NYG                161
## 10 MIA                159
## # ℹ 22 more rows

#How many times from 1979 to 2018 a team if you bet on them as an underdog
you would have won your bet
#Panthers, Jaguars, Ravens, Texans all partial outliers because they were not
teams when the dataset started.
#Panthers (1995), Jaguars(1995), Ravens(1996), Texans (2002)

spread_underdog_count<-dplyr::summarize(group_by(filter(NFL,
spread_cover_result == 0), team_underdog_id), count = n())
```

```r
arrange(spread_underdog_count, desc(count))
```

```
## # A tibble: 32 × 2
##    team_underdog_id count
##    <chr>            <int>
##  1 DET                205
##  2 TB                 204
##  3 ARI                201
##  4 CIN                185
##  5 CLE                185
##  6 NYJ                184
##  7 KC                 179
##  8 WAS                179
##  9 ATL                178
## 10 IND                178
## # ℹ 22 more rows
```

```r
#How many times from 1979 to 2018 a team if you bet on their game to go over,
the over hit, so you won your bet
#Panthers, Jaguars, Ravens, Texans all partial outliers because they were not
teams when the dataset started.
#Panthers (1995), Jaguars(1995), Ravens(1996), Texans (2002)

over_home_count<-dplyr::summarize(group_by(filter(NFL, over_under_result ==
'O'), team_home_id), count = n())
over_away_count<-dplyr::summarize(group_by(filter(NFL, over_under_result ==
'O'), team_away_id), count = n())

over_count<- over_home_count

for(i in 1 :32){
  if(over_home_count$team_home_id[i]== over_away_count$team_away_id[i]){
    over_count$count[i]<-over_home_count$count[i] + over_away_count$count[i]
  }
}

names(over_count)[1]<- 'team_id'

arrange(over_count, desc(count))
```

```
## # A tibble: 32 × 2
##    team_id count
##    <chr>   <int>
##  1 GB        349
##  2 DEN       338
##  3 NE        335
##  4 TEN       335
##  5 NO        330
##  6 MIN       328
```

```
## 7 SF 328
## 8 LAR 327
## 9 LAC 323
## 10 ARI 322
## # ℹ 22 more rows
```

```r
#How many times from 1979 to 2018 a team if you bet on their game to go over,
the under hit, so you lost your bet
#Panthers, Jaguars, Ravens, Texans all partial outliers because they were not
teams when the dataset started.
#Panthers (1995), Jaguars(1995), Ravens(1996), Texans (2002)

under_home_count<-dplyr::summarize(group_by(filter(NFL, over_under_result ==
'U'), team_home_id), count = n())
under_away_count<-dplyr::summarize(group_by(filter(NFL, over_under_result ==
'U'), team_away_id), count = n())

under_count<- under_home_count

for(i in 1 :32){
  if(under_home_count$team_home_id[i]== under_away_count$team_away_id[i]){
    under_count$count[i]<-under_home_count$count[i] +
under_away_count$count[i]
  }
}

names(under_count)[1]<- 'team_id'

arrange(under_count, desc(count))
```

```
## # A tibble: 32 × 2
##    team_id count
##    <chr>   <int>
##  1 TB        349
##  2 MIA       346
##  3 KC        345
##  4 NYG       343
##  5 PIT       340
##  6 PHI       339
##  7 CHI       338
##  8 BUF       336
##  9 DAL       332
## 10 NE        330
## # ℹ 22 more rows
```

```r
#Proportions of games
#Panthers, Jaguars, Ravens, Texans all partial outliers because they were not
teams when the dataset started.
#Panthers (1995), Jaguars(1995), Ravens(1996), Texans (2002)
```

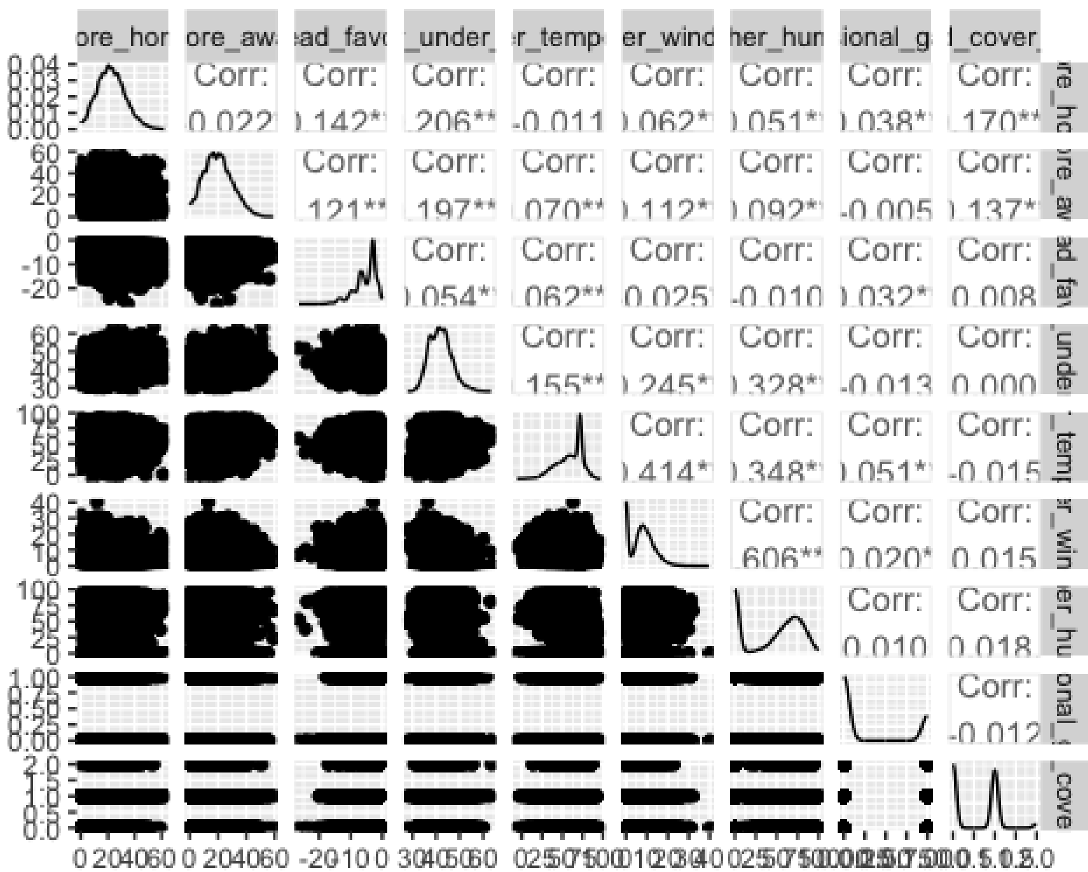
```r
#Arrange all data frames by alphabet first then do the for loop
spread_count_loss<-dplyr::summarize(group_by(filter(NFL, spread_cover_result
== 0), team_favorite_id), count = n())
spread_count_win<-dplyr::summarize(group_by(filter(NFL, spread_cover_result
== 1), team_favorite_id), count = n())

s_count_loss<-arrange(spread_count_loss, desc(team_favorite_id))
s_count_win<-arrange(spread_count_win, desc(team_favorite_id))

games_count<- s_count_win

games_count$count<-NA
games_count$spread_count<-NA
games_count$cover_percentage <- NA

for(i in 1 :32){
  if(games_count$team_favorite_id[i] == s_count_win$team_favorite_id[i]){
    games_count$count[i]<- s_count_win$count[i] + s_count_loss$count[i]
    games_count$spread_count[i]<-s_count_win$count[i]
    games_count$cover_percentage[i] <- s_count_win$count[i] /
games_count$count[i]
  }
}

names(games_count)[1]<- 'team_id'

arrange(games_count, desc(cover_percentage))

## # A tibble: 32 × 4
##    team_id count spread_count cover_percentage
##    <chr>   <int>        <int>            <dbl>
##  1 GB        252          197            0.782
##  2 IND       226          143            0.633
##  3 BUF       242          150            0.620
##  4 DEN       319          193            0.605
##  5 ATL       231          133            0.576
##  6 NE        381          215            0.564
##  7 PHI       324          181            0.559
##  8 CIN       225          122            0.542
##  9 CHI       281          150            0.534
## 10 SF        405          216            0.533
## # ℹ 22 more rows

#Proportions of games
#Panthers, Jaguars, Ravens, Texans all partial outliers because they were not
teams when the dataset started.
#Panthers (1995), Jaguars(1995), Ravens(1996), Texans (2002)

#Arrange all data frames by alphabet first then do the for loop
```

```r
h_count2<-dplyr::summarize(group_by(NFL, team_home_id), count = n())
a_count2<-dplyr::summarize(group_by(NFL, team_away_id), count = n())

home_count2<-arrange(h_count2, desc(team_home_id))
away_count2<-arrange(a_count2, desc(team_away_id))
spread_counting2<- arrange(spread_underdog_count, desc(team_underdog_id))

games_count2<-home_count2

games_count2$spread_count<-NA
games_count2$underdog_win_percentage <- NA

for(i in 1 :32){
  if(games_count2$team_home_id[i]== away_count2$team_away_id[i]){
    games_count2$count[i]<-home_count2$count[i] + away_count2$count[i]
    games_count2$spread_count[i]<-spread_counting2$count[i]
    games_count2$underdog_win_percentage[i] <- spread_counting2$count[i] /
games_count2$count[i]
  }
}

names(games_count2)[1]<- 'team_id'

arrange(games_count2, desc(underdog_win_percentage))

## # A tibble: 32 × 4
##    team_id count spread_count underdog_win_percentage
##    <chr>   <int>        <int>                   <dbl>
##  1 DET       641          205                   0.320
##  2 TB        644          204                   0.317
##  3 ARI       641          201                   0.314
##  4 CLE       594          185                   0.311
##  5 CAR       401          118                   0.294
##  6 CIN       645          185                   0.287
##  7 NYJ       651          184                   0.283
##  8 JAX       398          111                   0.279
##  9 KC        650          179                   0.275
## 10 ATL       651          178                   0.273
## # ℹ 22 more rows

# got rid of missing value for weather detail

for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == '')
    NFL$weather_detail[i]<-'C'
}
```

```r
for(i in 1: nrow(NFL)){
  if(is.na(NFL$weather_humidity[i]) == TRUE)
    NFL$weather_humidity[i]<- 0
}

for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == 'Rain | Fog')
    NFL$weather_detail[i]<-'R'
}

for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == 'Snow | Fog')
    NFL$weather_detail[i]<-'S'
}

for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == 'Snow | Freezing Rain')
    NFL$weather_detail[i]<-'S'
}

for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == 'DOME (Open Roof)')
    NFL$weather_detail[i]<-'D'
}


for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == 'DOME')
    NFL$weather_detail[i]<-'D'
}


for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == 'Snow')
    NFL$weather_detail[i]<-'S'
}

for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == 'Fog')
    NFL$weather_detail[i]<-'F'
}

for(i in 1: nrow(NFL)){
  if(NFL$weather_detail[i] == 'Rain')
    NFL$weather_detail[i]<-'R'
}

dplyr::summarize(group_by(NFL, weather_detail),count = n())
```

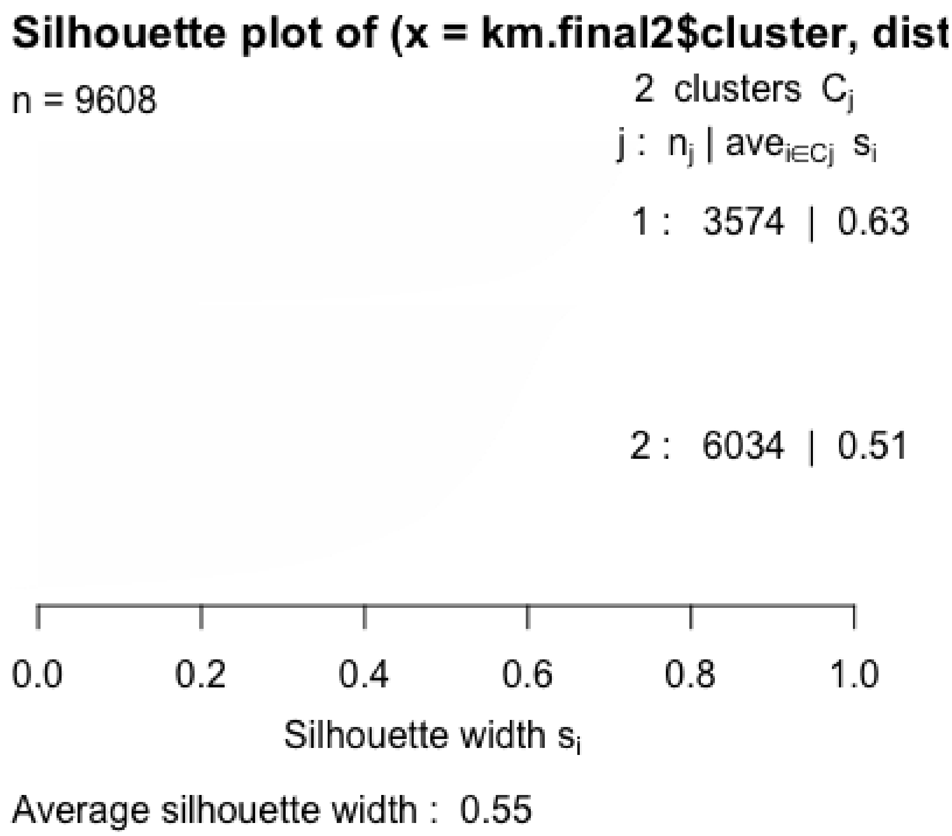
```
## # A tibble: 5 × 2
##   weather_detail count
##   <chr>          <int>
## 1 C               7486
## 2 D               2247
## 3 F                 28
## 4 R                129
## 5 S                 20
```

```
#ggpairs
un.NFL <- NFL[,-c(1,2,3,4,5,8,12,13,17,21,22,24,26,27,28)]
ggpairs(na.omit(un.NFL[,-c(3,9,10,11)]))
```

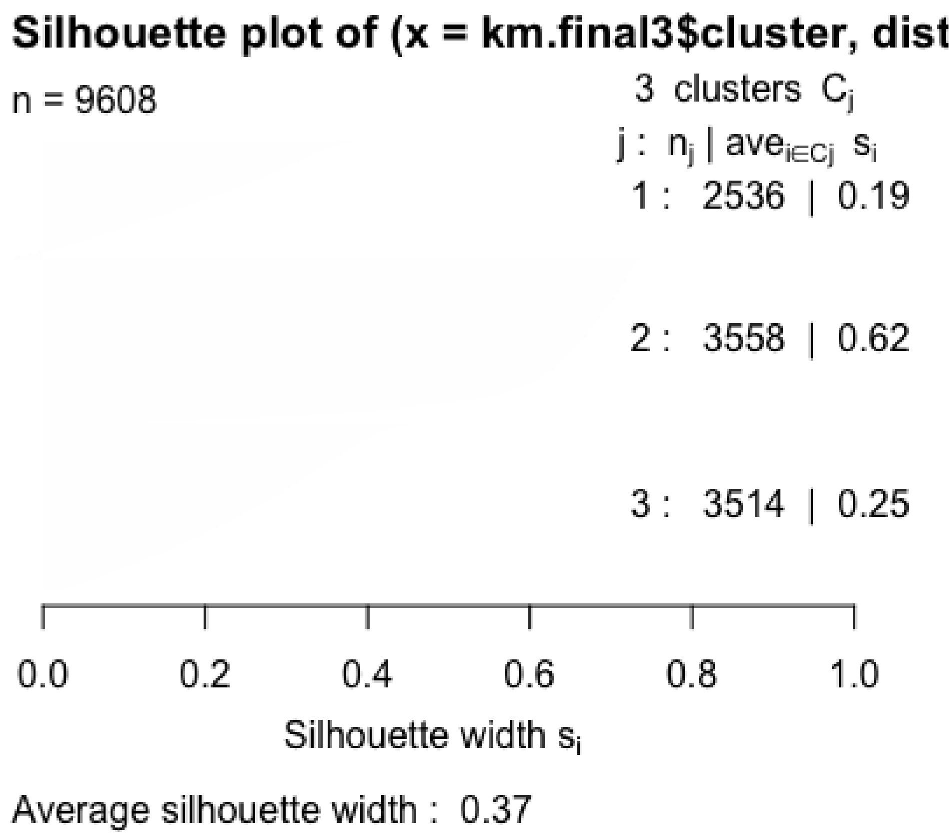


```
#clustering
#2 clusters seems best
new.NFL<-na.omit(un.NFL[,-c(3,9,10,11)])

newob.final<-new.NFL

d<- dist(newob.final)

km.final2 <- kmeans(newob.final, 2, nstart = 20)
```

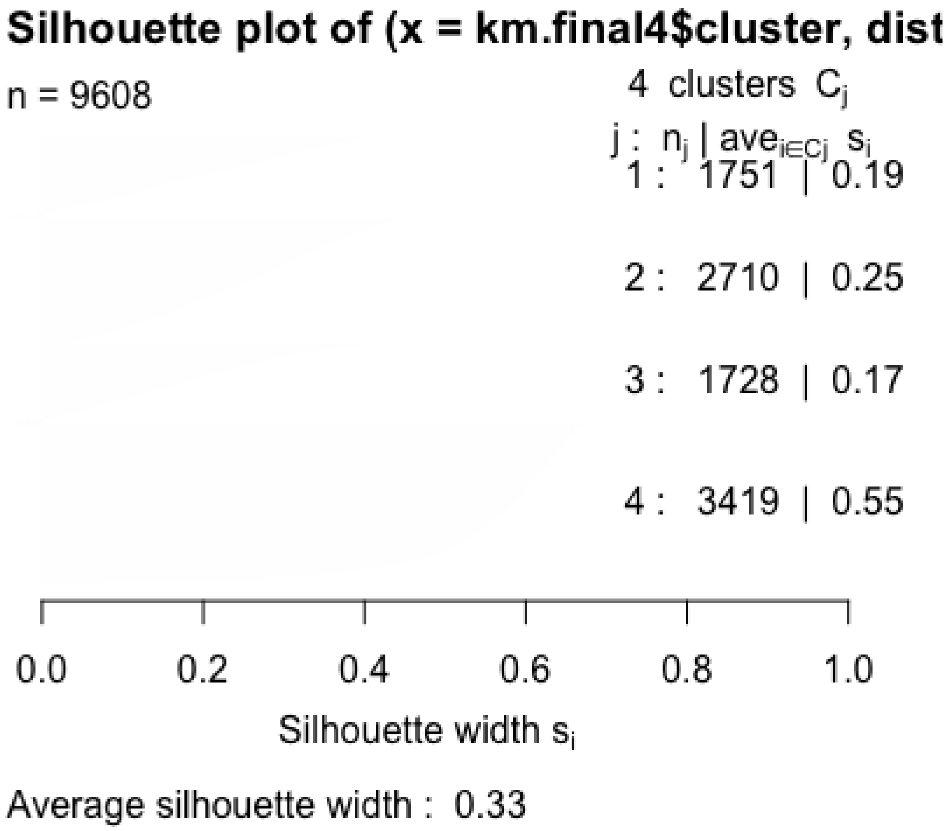
```
  clusters <- as.character(km.final2$cluster)
  table(clusters)

## clusters
##    1    2
## 3574 6034

km.final3 <- kmeans(newob.final, 3, nstart = 20)
  clusters <- as.character(km.final3$cluster)
  table(clusters)

## clusters
##    1    2    3
## 2536 3558 3514

km.final4<- kmeans(newob.final, 4, nstart = 20)
  clusters <- as.character(km.final4$cluster)
  table(clusters)

## clusters
##    1    2    3    4
## 1751 2710 1728 3419

km.final5 <- kmeans(newob.final, 5, nstart = 20)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 480400)

  clusters <- as.character(km.final5$cluster)
  table(clusters)

## clusters
##    1    2    3    4    5
##  563 1733 2880 1743 2689

km.final6 <- kmeans(newob.final, 6, nstart = 20)
  clusters <- as.character(km.final6$cluster)
  table(clusters)

## clusters
##    1    2    3    4    5    6
##  561 1271 1341 1337 2867 2231

km.final7 <- kmeans(newob.final, 7, nstart = 20)
  clusters <- as.character(km.final7$cluster)
  table(clusters)

## clusters
##    1    2    3    4    5    6    7
## 1339 2232 1650 1274  511 1271 1331

library(cluster)

plot(silhouette(km.final2$cluster, d))
```



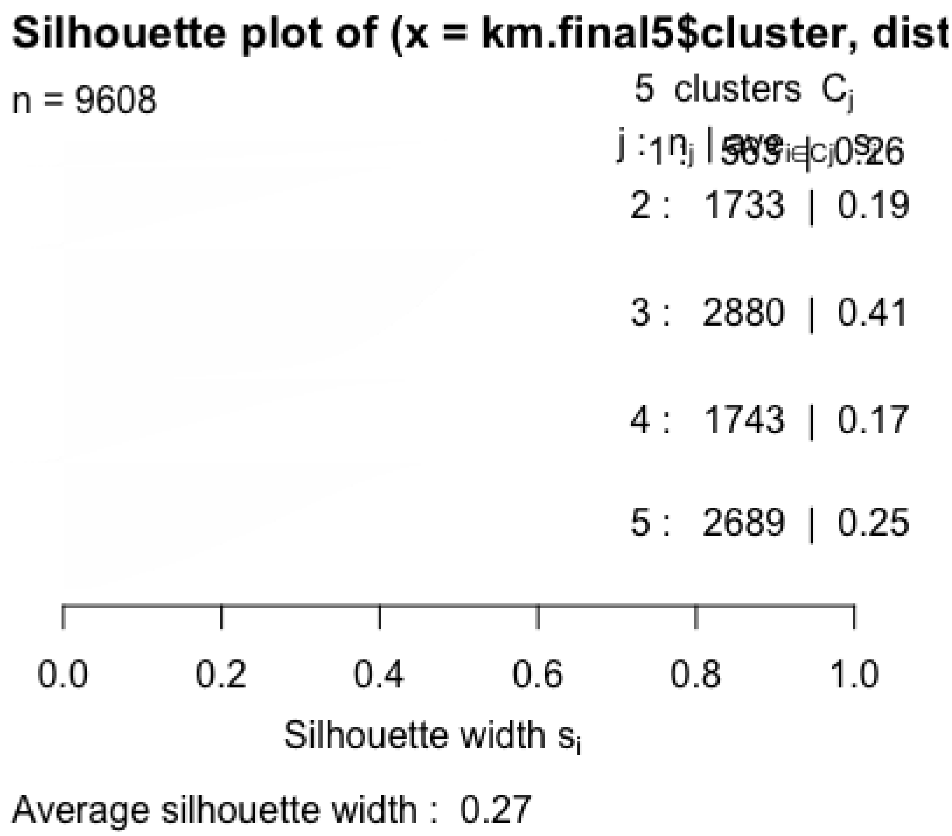

```
plot(silhouette(km.final3$cluster, d))
```



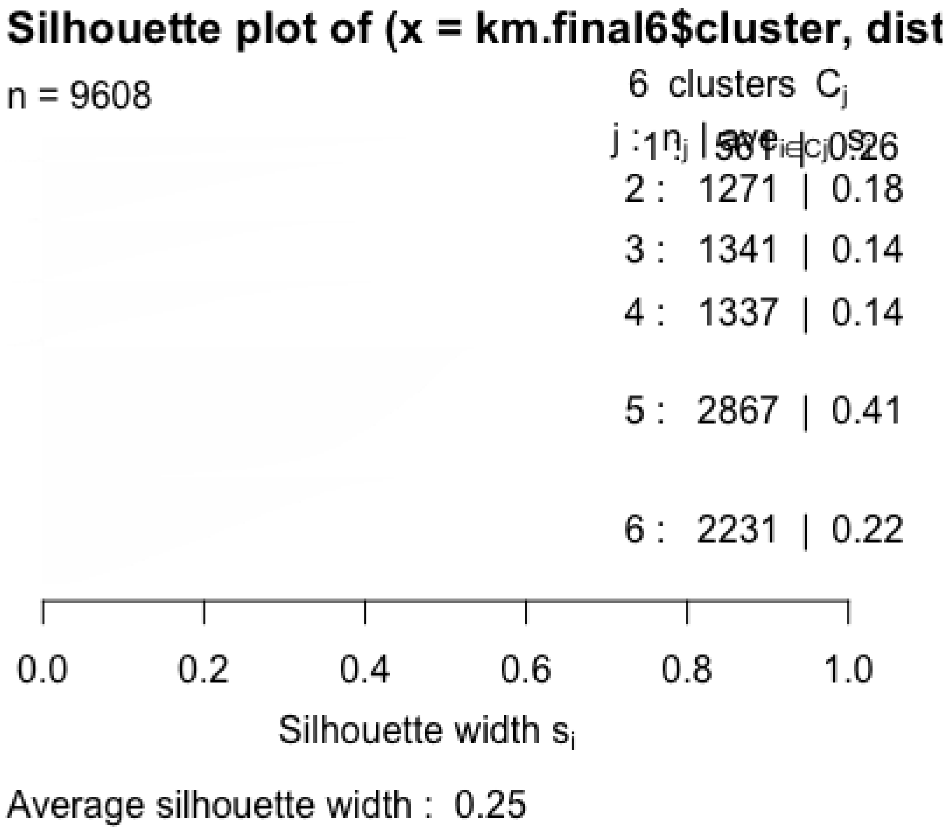

```
plot(silhouette(km.final4$cluster, d))
```



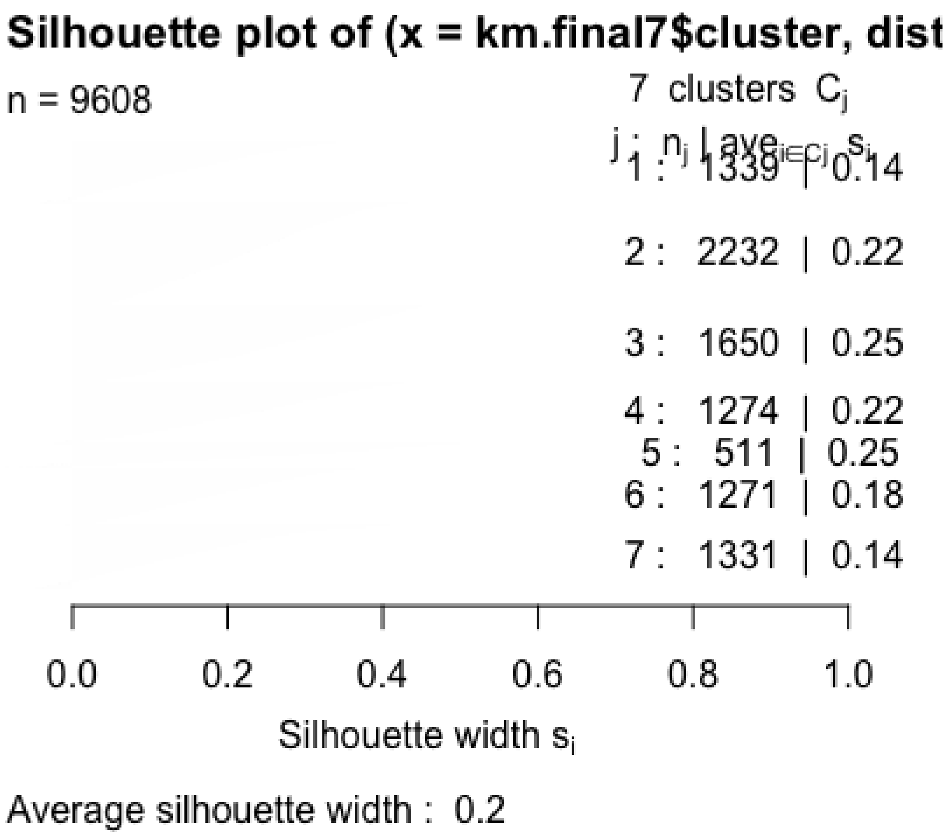

```
plot(silhouette(km.final5$cluster, d))
```




```
plot(silhouette(km.final6$cluster, d))
```




```
plot(silhouette(km.final7$cluster, d))
```




```
#Supervised LDA
#Accuracy of 51.207%

Sup.NFL<- NFL[,-c(1,2,3,4,5,6,7,8,9,12,13,19,20,21,22,24,25,26,27,28)]

set.seed(1)
fitControl <- trainControl(method = "cv", number = 5)

final.lda<- train(na.omit(over_under_result) ~ .,
                  data = na.omit(Sup.NFL),
                  method = "lda",
                  trControl = fitControl)
final.lda

## Linear Discriminant Analysis
##
## 9608 samples
##    7 predictor
##    3 classes: 'O', 'P', 'U'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 7687, 7688, 7685, 7686, 7686
```

```
## Resampling results:
##
##   Accuracy  Kappa
##   0.51207   0.04153287

pred.class<- predict(final.lda, Sup.NFL)
pred.prob<- predict(final.lda, Sup.NFL, type= "prob")


final.lda$final

## Call:
## lda(x, grouping = y)
##
## Prior probabilities of groups:
##          O          P          U
## 0.48428393 0.01894255 0.49677352
##
## Group means:
##   spread_favorite over_under_line weather_temperature weather_wind_mph
## O       -5.366430        41.73374            59.65442         7.117559
## P       -5.326923        41.54945            59.21978         7.307692
## U       -5.341819        41.97415            59.84140         7.613241
##   weather_humidity weather_detailD weather_detailF weather_detailR
## O         43.08016       0.2379110     0.003653557      0.01289491
## P         45.03846       0.2142857     0.000000000      0.00000000
## U         43.82631       0.2298345     0.002304630      0.01424680
##   weather_detailS divisional_game
## O     0.002578981       0.2654202
## P     0.000000000       0.2472527
## U     0.001676095       0.2918500
##
## Coefficients of linear discriminants:
##                              LD1          LD2
## spread_favorite      0.020191141 -0.030864050
## over_under_line      0.111744169  0.004099433
## weather_temperature  0.018219217  0.002398865
## weather_wind_mph     0.194210679  0.001032284
## weather_humidity     0.002309624 -0.003744072
## weather_detailD      1.209140773  0.776495310
## weather_detailF     -2.075165082  8.489461098
## weather_detailR      1.063278448  5.945592999
## weather_detailS     -1.994662302  8.407300109
## divisional_game      0.933480202  0.381053869
##
## Proportion of trace:
##  LD1  LD2
## 0.92 0.08
```

```r
#Supervised Random Forest
set.seed(1)
mtryGrid <- expand.grid(mtry = 1:3)
fitControl <- trainControl(method = "cv", number = 5)

final.rf<- train(na.omit(over_under_result) ~ .,
                  data = na.omit(Sup.NFL),
                  method = "rf",
                  trControl = fitControl,
                  tuneGrid=mtryGrid)
final.rf

## Random Forest
##
## 9608 samples
##    7 predictor
##    3 classes: 'O', 'P', 'U'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 7687, 7688, 7685, 7686, 7686
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   1     0.5074933  0.03148695
##   2     0.5093680  0.03741764
##   3     0.5039550  0.02605329
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.

varImp(final.rf)

## rf variable importance
##
##                      Overall
## weather_temperature 100.0000
## over_under_line      91.7725
## weather_humidity     85.3023
## spread_favorite      80.7293
## weather_wind_mph     74.0531
## divisional_game       9.8636
## weather_detailR       2.8375
## weather_detailD       2.6402
## weather_detailF       0.4864
## weather_detailS       0.0000
```