SAL 608 Assignment 4

Andrew Fish

2025-11-23

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(randomForest)
```

```
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.5.2
```

```
library(performance)
library(ggplot2)
```

#1.

```r
set.seed(01042024)
su_dat <- read_csv('data/wnba-team-elo-ratings.csv') %>%
  mutate(score_diff = score1 - score2)
```

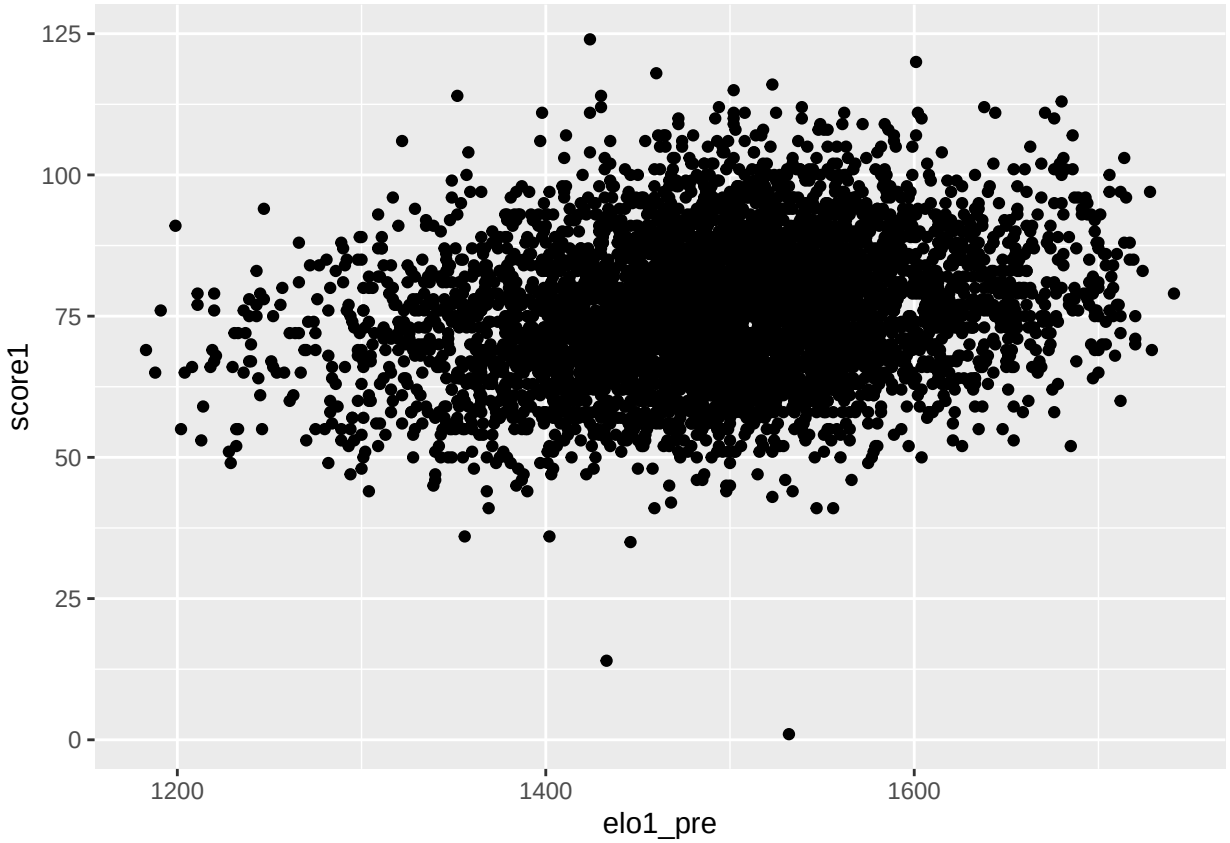
```
## Rows: 5244 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): date, team1, team2, name1, name2
## dbl (10): season, neutral, playoff, score1, score2, elo1_pre, elo2_pre, elo1...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
summary(su_dat)
```

```
##      season         date              team1              team2
##  Min.   :1997   Length:5244        Length:5244        Length:5244
##  1st Qu.:2003   Class :character   Class :character   Class :character
##  Median :2008   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2008
##  3rd Qu.:2014
##  Max.   :2019
##     name1              name2              neutral     playoff
##  Length:5244        Length:5244        Min.   :0   Min.   :0.00000
##  Class :character   Class :character   1st Qu.:0   1st Qu.:0.00000
##  Mode  :character   Mode  :character   Median :0   Median :0.00000
##                                        Mean   :0   Mean   :0.07323
##                                        3rd Qu.:0   3rd Qu.:0.00000
##                                        Max.   :0   Max.   :1.00000
##      score1           score2          elo1_pre       elo2_pre      elo1_post
##  Min.   :  1.00   Min.   :  0.00   Min.   :1183   Min.   :1190   Min.   :1168
##  1st Qu.: 67.00   1st Qu.: 64.00   1st Qu.:1443   1st Qu.:1440   1st Qu.:1441
##  Median : 75.00   Median : 72.00   Median :1498   Median :1497   Median :1498
##  Mean   : 75.79   Mean   : 72.55   Mean   :1495   Mean   :1493   Mean   :1495
##  3rd Qu.: 84.00   3rd Qu.: 81.00   3rd Qu.:1550   3rd Qu.:1547   3rd Qu.:1550
##  Max.   :124.00   Max.   :127.00   Max.   :1741   Max.   :1735   Max.   :1743
##    elo2_post        prob1          score_diff
##  Min.   :1188   Min.   :0.0790   Min.   :-45.000
##  1st Qu.:1440   1st Qu.:0.5090   1st Qu.: -6.000
##  Median :1497   Median :0.6180   Median :  4.000
##  Mean   :1493   Mean   :0.6066   Mean   :  3.244
##  3rd Qu.:1549   3rd Qu.:0.7160   3rd Qu.: 12.000
##  Max.   :1741   Max.   :0.9510   Max.   : 59.000
```

Wanted to look at summary statistics to get a sense of the range for elo_pre for team 1 and 2 (measure of range of team skills) and the range of score_diff

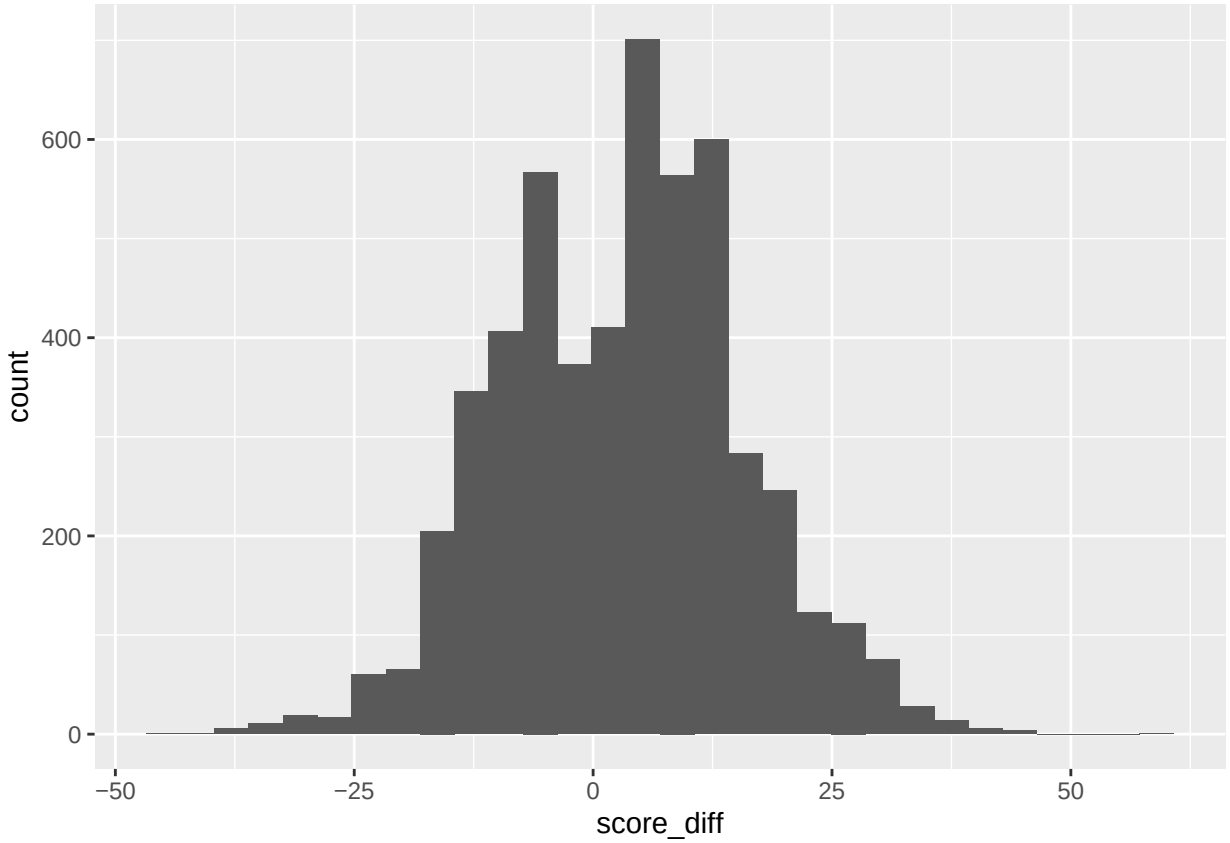
```r
ggplot(data = su_dat, aes(elo1_pre, score1)) +
  geom_point()
```



Visualizing the relationship between score_diff and elo

```
ggplot(data = su_dat, aes(score_diff)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



Visualizing the distribution of score_diff

#2.

```
set.seed(456)
##didn't use is_home1 since not a variable in the data set
(su.rf <- randomForest(score_diff ~ elo1_pre + elo2_pre + playoff,
                       data = su_dat))
```

```
##
## Call:
##  randomForest(formula = score_diff ~ elo1_pre + elo2_pre + playoff,      data = su_dat)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 141.9169
##                     % Var explained: 10.55
```

```
##rmse using the performance package
rmse(su.rf)
```

```
## [1] 11.91289
```

The RSMSE for this random forest model is 11.9289

#3.

```
set.seed(1234)
##using 70/30 train test split
train <- sample(nrow(su_dat), nrow(su_dat) * .7)

train_data <- su_dat[train, ]
test_data <- su_dat[-train, ]

##tuning the node size
find_term_val <- function(term_val) {
  min_nod <- floor(term_val)
  mod <- randomForest(score_diff ~ elo1_pre + elo2_pre + playoff,
                      data = train_data,
                      nodesize = min_nod)
  rmse(mod)
}
```

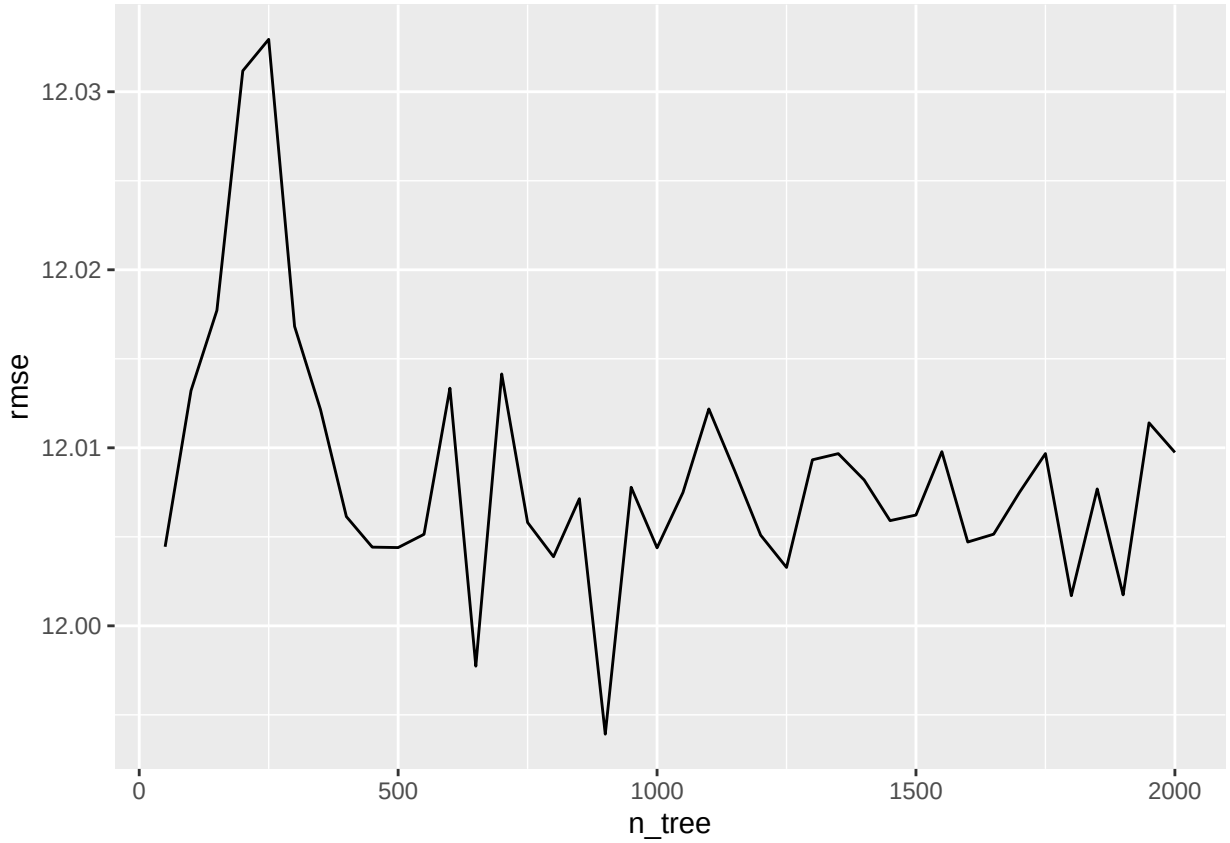
```
set.seed(1234)
ideal <- optimize(find_term_val, c(5, 500))
floor(ideal$minimum)
```

```
## [1] 194
```

```
##tuning number of trees using same training and test data
find_n_tree <- function(n_tree) {
  min_nod <- floor(ideal$minimum)
  mod <- randomForest(score_diff ~ elo1_pre + elo2_pre + playoff,
                      data = train_data,
                      nodesize = min_nod,
                      ntree = n_tree)
  rmse(mod)
}
```

```
set.seed(321)
##tibble of tree number and rmse
perform_by_tree <- tibble(
  n_tree = 1:40 * 50,
  rmse = map_dbl(1:40 * 50, find_n_tree)
)
```

```
##mapping the n_trees to find optimal number
ggplot(perform_by_tree, aes(n_tree, rmse)) +
  geom_line()
```



From this graph it looks like the RMSE starts leveling out around 1300-1400 trees. I will use 1300 for this problem. So overall the optimal number of trees is 1300 and the min nod size is 194

#4.

```
set.seed(789)
(final.rf <- randomForest(score_diff ~ elo1_pre + elo2_pre + playoff,
                          data = su_dat,
                          nodesize = floor(ideal$minimum),
                          ntree = 1300,
                          importance = TRUE))
```

```
##
## Call:
##  randomForest(formula = score_diff ~ elo1_pre + elo2_pre + playoff,      data = su_dat, nodesize = fl
##                Type of random forest: regression
##                      Number of trees: 1300
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 141.9173
##                     % Var explained: 10.55
```

```
rmse(final.rf)
```

```
## [1] 11.9129
```

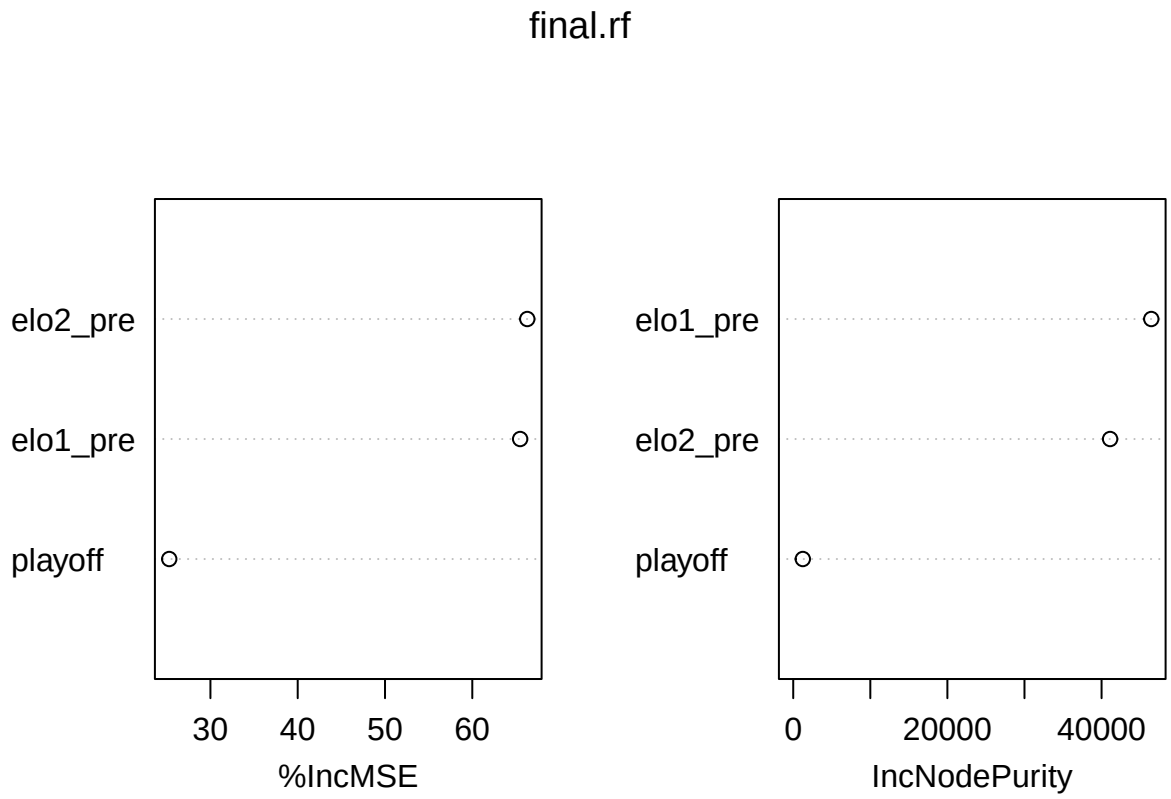
```
importance(final.rf)
```

```
##           %IncMSE IncNodePurity
## elo1_pre 65.48182     46449.814
## elo2_pre 66.29769     41108.697
## playoff  25.28202      1242.519
```

```
importance(final.rf)
```

```
##           %IncMSE IncNodePurity
## elo1_pre 65.48182     46449.814
## elo2_pre 66.29769     41108.697
## playoff  25.28202      1242.519
```

```
varImpPlot(final.rf)
```



elo2_pre and elo1_pre have the greatest importance to this random forest model when predicting score_diff. This makes sense as elo is essentially a metric that determines relative skill. So it is saying on paper which team is more skilled or better. Analytically this makes sense that these two would be very important to the model as they encompass every skill metric and are a great measure of how good a team really is. This metric would be similar to a March Madness seed for a viewer, anyone can see it and despite their basketball knowledge have a good idea of who is going to win the game.