White Advantage in Chess and How to Counter It

Jun Won (Lakon) Park (Group Leader) *79453940, Group Leader, data collection, data analysis, report-writing*
Sarah Li *60136959, data collection, data analysis, report-writing*

Research question: Is white at an advantage in chess and if so, what are some optimal strategies for black to increase their winning probability?

Introduction

For several centuries, millions of people worldwide have been playing chess as a recreational and competitive board game at their homes, in clubs, in tournaments, and even online nowadays. In the recent decades, chess has been one of the most popular topic in machine learning and artificial intelligence. The first move advantage has been researched extensively since the end of 19th century, and many studies have been shown that white has an inherent advantage.

Although there is a general set of chess openings, less research has been done on the effects of those openings on the final outcome. This paper intends to confirm white's first move advantage and study the relationship between the openings and the victory status. In particular, we are interested in the openings that are in favour for Black.

This paper's data consists basic player information and game information of over 20000 chess games played on Lichess, a very popular internet chess platform. The data includes game length, number of turns, winner, player elo*, all moves in Standard Chess Notation, Opening Eco*, Opening Name, and Opening Ply*.

Our target population is the data set itself. This paper will perform simple random sampling and stratified random sampling from the data set and compare the results obtained from the two different sampling methods. We will first define a new feature called "average elo" which is a mean of two player's ratings. We will define if a game is played by beginners if the average elo of the game is below 1200. These game records will likely negatively affect our result; if there exists any advantage for certain side, beginners will not likely to be able to use that advantage in their favour.

———————————————————————-

Elo : A numerical measurement to quantify a player's skill level
Eco : Standardised code for any given opening
Ply : Number of moves in the opening phase

———————————————————————-

Sampling methods

For both sampling methods, we will perform sampling twice - once for the White parameter estimate and once for Black. In this way, we do not have to calculate the covariance term which is difficult to calculate as below.

$$Var(X - Y) = Var(X) + Var(Y) - 2Cov(X, Y)$$

By combining two independent samples, the variance will increase but we will choose a sample size that is sufficiently large such that the margin of error is still small. For every sampling method, we will choose the total sample size to be 2000 which is approximately 10% of the population. This sample size is sufficiently large enough such that the width of our confidence intervals are small. For stratified sampling, we will separate the games into 5 different groups based on the average elo of the game from "1200-1400", "1400-1600", "1600-1800", "1800-2000", and "2000+." Players in the same elo range would likely have similar performance which reduces with-in variation of each group. The optimal allocation for stratified sampling yields total of 2003 samples. Hence, we have randomly removed one sample each from the three largest sampled stratas.

The Parameters of Interest

- White and Black's win rate
- White and Black's win rate for specific openings
- White and Black's mean number of turns to win

Estimators and Notations

N = population size
n = sample size
N_h = population size of strata h
n_h = sample size of strata h
S_1 = first SRS taken from population
S_2 = second SRS taken from population
St_1 = first StRS taken from population
St_2 = second StRS taken from population
$S_{1,d}$ = first SRS taken from population with opening d
$S_{2,d}$ = second SRS taken from population with opening d
$St_{1,d}$ = first StRS taken from population with opening d
$St_{2,d}$ = second StRS taken from population with opening d
$p_{i,W} = 1$ if $i \in P_{winner=White}$ (if game i's winner is White)
$p_{i,B} = 1$ if $i \in P_{winner=Black}$ (if game i's winner is Black)
$p_{i,d,W} = 1$ if $i \in P_{winner=White,opening=d}$ (if game i's winner is White with opening d)
$p_{i,d,B} = 1$ if $i \in P_{winner=Black,opening=d}$ (if game i's winner is Black with opening d)
y_i = number of turns took for the winner to win for game i
$\mu_{i,W}$ = y_i$ if $i \in P_{winner=White}$ (if game i's winner is White)
$\mu_{i,B}$ = y_i$ if $i \in P_{winner=Black}$ (if game i's winner is Black)
$\bar{p}_{S_1,W}$ = average of $p_{i,W}$ for $i \in S_1$
$\bar{p}_{S_2,B}$ = average of $p_{i,B}$ for $i \in S_2$
$\bar{p}_{S_1,d,W}$ = average of $p_{i,d,W}$ for $i \in S_{1,d}$

$\bar{p}_{S_2,d,B}$ = average of $p_{i,d,B}$ for $i \in S_{2,d}$
$\bar{p}_{St_1,h,w}$ = average of $p_{i,W}$ for $i \in St_1$ in strata h
$\bar{p}_{St_2,h,B}$ = average of $p_{i,B}$ for $i \in St_2$ in strata h
$\bar{p}_{St_1,h,d,W}$ = average of $p_{i,d,W}$ for $i \in St_{1,d}$ in strata h
$\bar{p}_{St_2,h,d,B}$ = average of $p_{i,d,B}$ for $i \in St_{2,d}$ in strata h
$\bar{\mu}_{S_1,W}$ = average of $\mu_{i,W}$ for $i \in S_{1,W}$
$\bar{\mu}_{S_2,B}$ = average of $\mu_{i,B}$ for $i \in S_{2,B}$
$\bar{\mu}_{St_1,W}$ = average of $\mu_{i,W}$ for $i \in St_{1,W}$
$\bar{\mu}_{St_2,B}$ = average of $\mu_{i,W}$ for $i \in St_{2,B}$
$\bar{\mu}_{St_1,h,W}$ = average of $\mu_{i,W}$ for $i \in St_{1,h,W}$ in strata h
$\bar{\mu}_{St_2,h,B}$ = average of $\mu_{i,W}$ for $i \in St_{2,h,B}$ in strata h

White's win rate in the population will be estimated using the following estimators:

$$\hat{p}_W = \bar{p}_{S_1,W}$$

$$\hat{p}_W = \bar{p}_{St_1,W} = \sum_{h=1}^{h} \frac{N_h}{N} \bar{p}_{St_1,h,w}$$

with Standard errors estimated using

$$\hat{SE}(\hat{p}_W) = \sqrt{(1 - \frac{n}{N}) \frac{\hat{p}_W(1 - \hat{p}_W)}{n}}$$

$$\hat{SE}(\hat{p}_W) = \sqrt{\sum_{h=1}^{5} (\frac{N_h}{N})^2 (1 - \frac{n_h}{N_h}) \frac{\bar{p}_{St_1,h,W}(1 - \bar{p}_{St_1,h,W})}{n_h}}$$

Black's win rate and stand error in the population will be estimated using the same formula as above except $\hat{P}_W$ replaced with $\hat{P}_B$, S_1 replaced with S_2, and St_1 replaced with St_2.

The difference in White and Black's win rate in the population will be estimated using

$$\hat{p}_W - \hat{p}_B = \bar{p}_{S_1,W} - \bar{p}_{S_2,B}$$

$$\hat{p}_W - \hat{p}_B = \bar{p}_{St_1,W} - \bar{p}_{St_2,B}$$

With pooled standard error with equal sample size

$$\begin{aligned} SE &= \sqrt{\frac{(n-1)((Var(\hat{P}_W)^2 + Var(\hat{P}_B)^2)}{2n-2}} \sqrt{\frac{2}{n}} \\ &= \sqrt{\frac{(Var(\hat{P}_W)^2 + Var(\hat{P}_B)^2)}{n}} \\ &= \sqrt{SE(\hat{P}_W)^2 + SE(\hat{P}_B)^2} \end{aligned}$$

White's win rate in the population for a given opening will be estimated using the following estimators:

$$\hat{p}_{d,W} = \bar{p}_{S_1,d,W}$$

$$\hat{p}_{d,W} = \bar{p}_{St_1,d,W} = \sum_{h=1}^{h} \frac{N_h}{N} \bar{p}_{St_1,h,d,w}$$

with Standard errors estimated using

$$\hat{SE}(\hat{p}_{d,W}) = \sqrt{(1 - \frac{n}{N}) \frac{\hat{p}_{d,W}(1 - \hat{p}_{d,W})}{n}}$$

$$\hat{SE}(\hat{p}_{d,W}) = \sqrt{\sum_{h=1}^{5} (\frac{N_h}{N})^2 (1 - \frac{n_h}{N_h}) \frac{\hat{p}_{d,h,W}(1 - \hat{p}_{d,h,W})}{n_h}}$$

Black's win rate for the given opening and stand error in the population will be estimated using the same formula as above except $\hat{P}_{d,W}$ replaced with $\hat{P}_{d,B}$, $S_{1,d}$ replaced with $S_{2,d}$, and $St_{1,d}$ replaced with $St_{2,d}$.

The difference in White and Black's win rate for the given opening in the population will be estimated using

$$\hat{p}_{d,W} - \hat{p}_{d,B} = \bar{p}_{S_1,d,W} - \bar{p}_{S_2,d,B}$$

$$\hat{p}_{d,W} - \hat{p}_{d,B} = \bar{p}_{St_1,d,W} - \bar{p}_{St_2,d,B}$$

With pooled standard error

$$SE = \sqrt{\frac{(n_{d,St_1} - 1)Var(\hat{P}_{d,W})^2 + (n_{d,St_2} - 1)Var(\hat{P}_{d,B})^2}{n_{d,St_1} + n_{d,St_2} - 2}} \sqrt{\frac{1}{n_{d,St_1}} + \frac{1}{n_{d,St_2}}}$$

However, this estimate is very tricky to calculate. Hence, we will assume $n_{d,St_1} = n_{d,St_2}$ such that

$$SE = \sqrt{\hat{SE}(\hat{p}_{d,W})^2 + \hat{SE}(\hat{p}_{d,B})^2}$$

White's mean number of turns to win in the population will be estimated using the following estimators:

$$\hat{\mu}_W = \bar{\mu}_{S_1,W}$$

$$\hat{\mu}_W = \bar{\mu}_{St_1,W} = \sum_{h=1}^{h} \frac{N_h}{N} \bar{\mu}_{St_1,h,w}$$

with Standard errors estimated using

$$\hat{SE}(\hat{\mu}_W) = \sqrt{(1 - \frac{n}{N}) \frac{Var(S_1)}{n}}$$

where $Var(S_1)$ is the variance of turns in S_1

$$\hat{SE}(\hat{\mu}_W) = \sqrt{\sum_{h=1}^{5} (\frac{N_h}{N})^2 (1 - \frac{n_h}{N_h}) \frac{Var(S_{1,h})}{n_h}}$$

where $Var(S_{1,h})$ is the variance of turns in strata h in St_1.

Black's win rate and stand error in the population will be estimated using the same formula as above except $\hat{P}_W$ replaced with $\hat{P}_B$, S_1 replaced with S_2, and St_1 replaced with St_2.

The difference in White and Black's mean number of turns to win in the population will be estimated using

$$\hat{\mu}_W - \hat{\mu}_B = \bar{\mu}_{S_1,W} - \bar{\mu}_{S_2,B}$$

$$\hat{\mu}_W - \hat{\mu}_B = \bar{\mu}_{St_1,W} - \bar{\mu}_{St_2,B}$$

With pooled standard error with equal sample size

$$SE = \sqrt{SE(\hat{\mu}_W)^2 + SE(\hat{\mu}_B)^2}$$

Does White have higher win rate than Black?

We define win rate to be the proportion of games won by each side. To confirm White's inherent advantage, we will perform two independent sample t-test to determine whether White has a higher win rate than that of Black. Hence, we will test the following hypothesis below

$$H_0 : p_w - p_b = 0 \qquad H_a : p_w - p_b > 0$$

where p_w and p_b represent White and Black's win rate each respectively.

We will assume equal variance among two SRS samples and conduct t-test using pooled variance. The constructed 95% confidence interval is as below:

	Win rate Estimate	SE
White	0.5040	0.0105715
Black	0.4685	0.0105509

Diff	Pooled SE	95.CI.lower	95.CI.upper
0.0355	0.0149358	0.0062263	0.0647737

The constructed 95% confidence interval is $(0.0062, 0.0648)$ which does not contain 0. Hence, we can reject the null hypothesis in favour of the alternative hypothesis that White has a higher winning proportion.

We will assume equal variance among two StRS samples and conduct t-test using pooled variance. The constructed 95% confidence interval is as below:

elo range	winner	win rate est.	strata size	Var. by strata
1200-1400	white	0.5217391	391	0.0006382
1400-1600	white	0.5069552	647	0.0003863
1600-1800	white	0.4898785	494	0.0005059
1800-2000	white	0.5143770	313	0.0007981
2000+	white	0.5032258	155	0.0016128

elo range	winner	win rate est.	strata size	Var. by strata
1200-1400	black	0.4884910	391	0.0006390
1400-1600	black	0.4544049	647	0.0003832
1600-1800	black	0.4534413	494	0.0005017
1800-2000	black	0.4423077	312	0.0007906
2000+	black	0.4358974	156	0.0015762

elo range	strata size prop.
1200-1400	0.1949283
1400-1600	0.3219334
1600-1800	0.2470221
1800-2000	0.1568638
2000+	0.0790407

	Win rate Estimate	SE
White	0.5063808	0.0078503
Black	0.4573546	0.0081990

Diff	Pooled SE	95.CI.lower	95.CI.upper
0.0490262	0.0113512	0.0267782	0.0712742

The constructed 95% confidence interval is $(0.0268, 0.0713)$ which does not contain 0. Hence, we can reject the null hypothesis in favour of the alternative hypothesis that White has a higher winning proportion.

The result from both SRS and StRS show that White has a higher winning proportion. Such phenomenon could be due to white having a first-move advantage.

What is an optimal game opening for Black?

Due to the many possible openings a game can start with, the sample size in each possible domain (split by opening) may be very small. In order to ensure that the confidence interval is of reasonable width, we will only estimate within the domain if its sample size yields a confidence interval including ± 0.2 of our estimate of win rate. In the worst case, the win rate of Black is the same as that of White. Hence, using an initial guess of $p = 0.5$, the minimum sample size is at least 25. Since we know the domain size of each opening, the resulting minimum sample size for each opening will differ and will be less than 25.

opening name	size of sample 1	size of sample 2
French Defense: Knight Variation	28	22
Scandinavian Defense: Mieses-Kotroc Variation	31	24
Scotch Game	35	23
Sicilian Defense	35	39
Sicilian Defense: Bowdler Attack	41	32
Van't Kruijs Opening	24	33

Again, using the openings above, we test the same hypothesis as above. The resulting confidence interval is as below:

opening name	Diff. win rate	SE	95.CI.lower	95.CI.upper
French Defense: Knight Variation	-0.0909091	0.1413246	-0.3750612	0.1932430
Scandinavian Defense: Mieses-Kotroc Variation	0.3346774	0.1269533	0.0800413	0.5893135
Scotch Game	0.1962733	0.1276304	-0.0594011	0.4519477
Sicilian Defense	-0.1355311	0.1152343	-0.3652465	0.0941842
Sicilian Defense: Bowdler Attack	-0.0228659	0.1164512	-0.2550629	0.2093312
Van't Kruijs Opening	-0.1704545	0.1319470	-0.4348822	0.0939731

Using SRS sampling and assuming equal variances, all openings but "Scandinavian Defense: Mieses-Kotroc Variation" contain 0 in their 95% confidence intervals. For these openings, we cannot reject the null hypothesis which states that there is no difference in win rates. More specifically, there is insufficient evidence to suggest that black has a higher win rate when these openings are used. For the intervals that are strictly greater than 0, there is sufficient evidence to reject the null hypothesis in favour of the alternative hypothesis that White has higher win rate. In all, there is no opening that yields higher win rate for Black.

Using StRS, we will again determine the valid openings in two different samples and test the same hypothesis.

opening name	size of sample 1	size of sample 2
French Defense: Knight Variation	28	22
Scandinavian Defense: Mieses-Kotroc Variation	31	24
Scotch Game	35	23
Sicilian Defense	35	39
Sicilian Defense: Bowdler Attack	41	32
Van't Kruijs Opening	24	33

The resulting confidence interval for the openings above is as follows:

opening name	Diff. win rate	SE	95.CI.lower	95.CI.upper
French Defense: Knight Variation	0.0426424	0.1230920	-0.2039406	0.2892254
Scandinavian Defense: Mieses-Kotroc Variation	0.2094644	0.1159952	-0.0226415	0.4415702
Scotch Game	-0.0172765	0.1219349	-0.2617412	0.2271882
Sicilian Defense	-0.1609474	0.1026540	-0.3651967	0.0433019
Sicilian Defense: Bowdler Attack	-0.1600468	0.1278376	-0.4168162	0.0967227
Van't Kruijs Opening	-0.3068106	0.0923759	-0.4923531	-0.1212681

The confidence interval for opening "Van't Kruijs Opening" is strictly negative. This suggests that for this opening, Black has a higher win rate than White. For other openings with confidence intervals that contain 0, we cannot reject the null hypothesis which states that there is no difference in win rates. More specifically, there is insufficient evidence to suggest that black has a higher win rate when these openings are used.

Since SRS and StRS sample contain different valid openings, it is not possible to compare the results. However, we have found an opening in which Black has a higher win rate.

Does it take longer for Black to win?

We have already confirmed that White has a first-move advantage over Black. So how does Black actually overcome this advantage? Our hypothesis is that Black will need to spend extra turns to overcome the disadvantage in the beginning. This leads to increase an increase in overall turn spent by Black to win. Hence, we will test the following hypothesis,

$$H_0 : \mu_W - \mu_B = 0 \qquad H_a : \mu_W - \mu_B < 0$$

where μ_W and μ_B represent White and Black's mean number of turns to win each respectively.

We will assume equal variance among two SRS samples and conduct t-test using pooled variance. The constructed 95% confidence interval is as below.

	Est.	SE
White	57.68849	0.9787740
Black	60.52721	0.9840654

Diff	SE	95.CI.lower	95.CI.upper
-2.838722	1.362178	-5.508543	-0.1689018

The constructed confidence interval does not contain 0 which implies there is sufficient evidence to reject the null hypothesis in favour of the alternative hypothesis that Black takes longer to win.

Using StRS and assuming equal variance,

elo range	mean turns for white
1200-1400	51.26471
1400-1600	55.52744
1600-1800	60.51653
1800-2000	65.25466
2000+	55.08974

elo range	mean turns for black
1200-1400	55.90052
1400-1600	57.77551
1600-1800	60.82143
1800-2000	65.34783
2000+	72.05882

elo range	strata size	var. by strata	winner
1200-1400	391	1.725542	white
1400-1600	647	1.235008	white
1600-1800	494	1.563627	white
1800-2000	313	3.240864	white
2000+	155	4.484618	white
elo range	strata size	var. by strata	winner
1200-1400	391	2.331782	black
1400-1600	647	1.338437	black
1600-1800	494	1.553842	black
1800-2000	312	3.169745	black
2000+	156	6.058140	black

Diff	Pooled SE	95.CI.lower	95.CI.upper
-3.058561	1.175147	-5.361806	-0.7553155

The constructed confidence interval does not contain 0 which implies there is sufficient evidence to reject the null hypothesis in favour of the alternative hypothesis that Black takes longer to win.

Conclusion

From our analysis of win rate, we have even more reason to believe that White has a first-move advantage. Both SRS and StRS with optimal allocation produced estimates which suggest a significant difference between Black and White win rate.
Furthermore, we discovered through StRS that there are openings with greater win rates for Black side. Our choice of stratifying by "average elo" may have been effective in lowering variance of our estimate, as players in the same elo range would likely have similar performance. Estimates of win rate per opening suffer from the limitation of sample size, since the abundance of possible openings leads to very few observations of each opening. There may be other openings which are more advantageous for Black side, but are rarely used by players on Lichess.

While the Black side cannot easily change which opening White starts with, they can attempt to play more turns to overcome White's early-game advantage. Our analysis of average number of turns produced results indicating that games won by Black side involve more turns.

Moreover, it is crucial to note that SRS and StRS yielded the same result yet StRS produced smaller standard errors for each estimate. The advantage of SRS is that it eases the difficulty of standard error calculation. However, the disadvantage of SRS is that due to the simplistic nature of the sampling, the standard error calculation yields a larger standard error than that of StRS.

We cannot assume that our sample from Lichess is representative of the entire population of chess players due to differences in the nature of online chess and traditional chess. For example, online elo could be less indicative of actual skill, due to randomness of opponents and lack of formality. However, our results could possibly be extended to other online chess platforms, such as Chess.com and chess24, which likely have similar players. Our new knowledge of openings and game length favouring Black side could be used to try and overcome first-move advantage in online chess.

Appendix

Dataset: https://www.kaggle.com/datasnaek/chess

```
# Load data
df <- read.csv("games.csv")

# Calculate the average elo of the game
df <- mutate(df %>% rowwise(),
       average_elo = rowMeans(cbind(black_rating, white_rating)))

# Filter games by average elo
df <- filter(df, average_elo >= 1200)

# Select only necessary columns for analysis
df <- subset(df,
             select = c(id, turns, white_rating, black_rating, victory_status,
                        winner, moves, opening_eco, opening_name, opening_ply,
                        average_elo ))
```

```
# Simple Random Sampling
N <- nrow(df)
n <- 2000
set.seed(1234)
sample.index.s1 <- sample(1:N, size=n, replace = FALSE)
srs.sample.s1 <- df[sample.index.s1,]

set.seed(4321)
sample.index.s2 <- sample(1:N, size=n, replace = FALSE)
srs.sample.s2 <- df[sample.index.s2,]
knitr::kable(table1)

# Determine minimum and maximum before stratifying
min(df$average_elo)
max(df$average_elo)

df$elo_range <- cut(df$average_elo,
                    c(1200, 1400, 1600, 1800, 2000, 2600))
levels(df$elo_range) <- c("1200-1400", "1400-1600", "1600-1800", "1800-2000",
                          "2000+")
df$winner <- as.factor(df$winner)

# Check if standard deviations of the strata are identical
se.by.strata <- aggregate(as.numeric(df$winner), by=list(df$elo_range), FUN=sd)
se.by.strata

# Standard deviations within strata are not identical, \
# so find optimal sample sizes
```

```r
pop.size.by.strata <- aggregate(df$winner, by=list(df$elo_range), FUN=length)
denom <- sum(pop.size.by.strata[2] * se.by.strata[2])
sample.size.by.strata <- (pop.size.by.strata[2] * se.by.strata[2]) / denom

# Sample from each strata
strsample <- function(df, sample.size.by.strata, n, seed) {
  str.sample <- df[FALSE,]
  colnames(str.sample) <- names(df)
  for (i in 1:length(levels(df$elo_range))) {
    strata <- which(df$elo_range == levels(df$elo_range)[i])
    set.seed(seed)
    sample.idx <- sample(strata,
                         size = ceiling(sample.size.by.strata$x[i] * n),
                         replace = FALSE)
    sample <- df[sample.idx,]
    str.sample <- rbind(str.sample, sample)
  }

  # Stratified sample contains 1003 samples due to rounding of the proportions,
  # so we randomly remove three from random strata
  strata.for.removal <- sample(1:5, 2)
  for (s in strata.for.removal) {
    set.seed(1234)
    to.remove <- sample(which(str.sample$elo_range == levels(df$elo_range)[s]), 1)
    str.sample <- str.sample[-to.remove,]
  }

  return(str.sample)
}

white.str.sample <- strsample(df, sample.size.by.strata, n, 1234) %>%
  group_by(elo_range)
black.str.sample <- strsample(df, sample.size.by.strata, n, 4321) %>%
  group_by(elo_range)
z.95 <- qnorm(0.975)
# Returns the sample variance of a given proportion
var.est <- function(p) {
  p * (1 - p)
}
# Calculate white's win rate
white.prop <- srs.sample.s1 %>%
  count(winner) %>%
  group_by(winner) %>%
  mutate(win.prop = n / 2000)

white.p <- as.numeric(white.prop[3,3])
```

```r
black.prop <- srs.sample.s2 %>%
  count(winner) %>%
  group_by(winner) %>%
  mutate(win.prop = n / 2000)

black.p <- as.numeric(black.prop[1,3])
white.var <- (1-n/N)*(var.est(white.p)/n)
black.var <- (1-n/N)*(var.est(black.p)/n)
srs.se <- sqrt(white.var + black.var)
```

```r
# Determine minimum and maximum before stratifying
min(df$average_elo)
max(df$average_elo)

df$elo_range <- cut(df$average_elo,
                    c(1200, 1400, 1600, 1800, 2000, 2600))
levels(df$elo_range) <- c("1200-1400", "1400-1600", "1600-1800", "1800-2000",
                          "2000+")
df$winner <- as.factor(df$winner)

# Check if standard deviations of the strata are identical
se.by.strata <- aggregate(as.numeric(df$winner), by=list(df$elo_range), FUN=sd)
se.by.strata

# Standard deviations within strata are not identical, \
# so find optimal sample sizes
pop.size.by.strata <- aggregate(df$winner, by=list(df$elo_range), FUN=length)
denom <- sum(pop.size.by.strata[2] * se.by.strata[2])
sample.size.by.strata <- (pop.size.by.strata[2] * se.by.strata[2]) / denom

# Sample from each strata
strsample <- function(df, sample.size.by.strata, n, seed) {
  str.sample <- df[FALSE,]
  colnames(str.sample) <- names(df)
  for (i in 1:length(levels(df$elo_range))) {
    strata <- which(df$elo_range == levels(df$elo_range)[i])
    set.seed(seed)
    sample.idx <- sample(strata,
                         size = ceiling(sample.size.by.strata$x[i] * n),
                         replace = FALSE)
    sample <- df[sample.idx,]
    str.sample <- rbind(str.sample, sample)
  }

  # Stratified sample contains 1003 samples due to rounding of the proportions,
  # so we randomly remove three from random strata
  strata.for.removal <- sample(1:5, 2)
```

```
  for (s in strata.for.removal) {
    set.seed(1234)
    to.remove <- sample(which(str.sample$elo_range == levels(df$elo_range)[s]), 1)
    str.sample <- str.sample[-to.remove,]
  }

  return(str.sample)
}

white.str.sample <- strsample(df, sample.size.by.strata, n, 1234) %>%
  group_by(elo_range)
black.str.sample <- strsample(df, sample.size.by.strata, n, 4321) %>%
  group_by(elo_range)
```

```
z.95 <- qnorm(0.975)
# Returns the sample variance of a given proportion
var.est <- function(p) {
  p * (1 - p)
}
# Calculate white's win rate
white.prop <- srs.sample.s1 %>%
  count(winner) %>%
  group_by(winner) %>%
  mutate(win.prop = n / 2000)

white.p <- as.numeric(white.prop[3,3])

black.prop <- srs.sample.s2 %>%
  count(winner) %>%
  group_by(winner) %>%
  mutate(win.prop = n / 2000)

black.p <- as.numeric(black.prop[1,3])

srs.se <- sqrt((1-n/N)*(var.est(white.p) + var.est(black.p))/n)
```

```
# Calculate Nh/N, the strata proportion
Nh <- df %>% count(elo_range, .drop=FALSE)
Nh <- Nh[complete.cases(Nh),]

nh.white <- white.str.sample %>% count(elo_range, .drop=FALSE)
nh.black <- black.str.sample %>% count(elo_range, .drop=FALSE)
strata.size.prop <- Nh[2] / N

# Calculate white's win proportion by each strata
white.win.prop <- white.str.sample %>%
  count(winner) %>%
```

```
  group_by(elo_range) %>%
  mutate(win.prop = n / sum(n))

# The estimated aggregated win proportion for white
white.prop <- white.win.prop[white.win.prop$winner == "white", ]
white.p.str.est <- sum(white.prop$win.prop * strata.size.prop)

# The estimated aggregated variance of win proportion for white
white.se.by.strata <- bind_cols(white.prop, nh = nh.white$n)
white.se.by.strata <- white.se.by.strata %>% mutate(var.by.strata = win.prop * (1-win.prop)/nh)
white.se.by.strata <- bind_cols(white.se.by.strata, strata.prop.sq = strata.size.prop$n^2)
white.se.by.strata <- white.se.by.strata %>% mutate(strata.prop.sq*(1-n/nh)*var.by.strata)
white.str.se <- sqrt(sum(white.se.by.strata$`strata.prop.sq * (1 - n/nh) * var.by.strata`))

# Calculate black's win proportion by each strata
black.win.prop <- black.str.sample %>%
  count(winner) %>%
  group_by(elo_range) %>%
  mutate(win.prop = n / sum(n))

# The estimated aggregated win proportion for white
black.prop <- black.win.prop[black.win.prop$winner == "black", ]
black.p.str.est <- sum(black.prop$win.prop * strata.size.prop)

# The estimated aggregated variance of win proportion for black
black.se.by.strata <- bind_cols(black.prop, nh = nh.black$n)
black.se.by.strata <- black.se.by.strata %>% mutate(var.by.strata = win.prop * (1-win.prop)/nh)
black.se.by.strata <- bind_cols(black.se.by.strata, strata.prop.sq = strata.size.prop$n^2)
black.se.by.strata <- black.se.by.strata %>% mutate(strata.prop.sq*(1-n/nh)*var.by.strata)
black.str.se <- sqrt(sum(black.se.by.strata$`strata.prop.sq * (1 - n/nh) * var.by.strata`))

# Their difference

diff.p <- white.p.str.est - black.p.str.est
pooled.se <- sqrt(white.str.se^2 + black.str.se^2)
(diff.p) + z.95 * pooled.se * c(-1, 1)
```

```
# Guess the most conservative variance
# Find minimum domain sample size for desired CI width
var.guess <- 0.25
ci.width <- 0.2
n0 <- z.95^2 * var.guess / ci.width^2
```

```
openings.df.s1 <- data.frame(table(white.str.sample$opening_name))
openings.df.s2 <- data.frame(table(black.str.sample$opening_name))
names(openings.df.s1) <- c("name", "frequency")
names(openings.df.s2) <- c("name", "frequency")
```

```r
# Include openings with sample size large enough for usable CI
openings.freq.s1 <- openings.df.s1[openings.df.s1$frequency > 15,]
openings.freq.s2 <- openings.df.s2[openings.df.s2$frequency > 15,]

openings.df.p <- data.frame(table(df$opening_name))
names(openings.df.p) <- c("name", "frequency")

# Include openings with sample sizes yielding the desired CI width
domain.sizes.s1 <- c()
domain.sizes.s2 <- c()

for (name in openings.freq.s1$name) {
  size <- n0 / (1 + n0 / openings.df.p[openings.df.p$name == name,]$frequency)
  domain.sizes.s1 <- c(domain.sizes.s1, size)
}

for (name in openings.freq.s2$name) {
  size <- n0 / (1 + n0 / openings.df.p[openings.df.p$name == name,]$frequency)
  domain.sizes.s2 <- c(domain.sizes.s2, size)
}

openings.valid.s1 <-
  openings.freq.s1[openings.freq.s1$frequency > domain.sizes.s1,]
openings.valid.s2 <-
  openings.freq.s2[openings.freq.s2$frequency > domain.sizes.s2,]

openings.valid.str.sample <- merge(openings.valid.s1,
                                   openings.valid.s2, by = "name")
```

```r
# Guess the most conservative variance
# Find minimum domain sample size for desired CI width
var.guess <- 0.25
ci.width <- 0.2
n0 <- z.95^2 * var.guess / ci.width^2
openings.df.s1 <- data.frame(table(srs.sample.s1$opening_name))
openings.df.s2 <- data.frame(table(srs.sample.s2$opening_name))
names(openings.df.s1) <- c("name", "frequency")
names(openings.df.s2) <- c("name", "frequency")

# Include openings with sample size large enough for usable CI
openings.freq.s1 <- openings.df.s1[openings.df.s1$frequency > 15,]
openings.freq.s2 <- openings.df.s2[openings.df.s2$frequency > 15,]

openings.df.p <- data.frame(table(df$opening_name))
names(openings.df.p) <- c("name", "frequency")
```

```r
# Include openings with sample sizes yielding the desired CI width
domain.sizes.s1 <- c()
domain.sizes.s2 <- c()

for (name in openings.freq.s1$name) {
  domain.sizes.s1 <- c(domain.sizes.s1, n0 / (1 + n0 / openings.df.p[openings.df.p$name == name
}

for (name in openings.freq.s2$name) {
  domain.sizes.s2 <- c(domain.sizes.s2, n0 / (1 + n0 / openings.df.p[openings.df.p$name == name
}

openings.valid.s1 <- openings.freq.s1[openings.freq.s1$frequency > domain.sizes.s1,]
openings.valid.s2 <- openings.freq.s2[openings.freq.s2$frequency > domain.sizes.s2,]

openings.valid.srs.sample <- merge(openings.valid.s1, openings.valid.s2, by = "name")

estimates <- rep(0, nrow(openings.valid.srs.sample))
diff.ses <- rep(0, nrow(openings.valid.srs.sample))
intervals <- matrix(0, nrow(openings.valid.srs.sample), 2)

for (i in 1:nrow(openings.valid.srs.sample)) {
  # Find estimate and CI for difference in win rate for white/black
  # for one opening
  domain.name <- openings.valid.srs.sample[i, 1]
  domain.s1 <- srs.sample.s1[srs.sample.s1$opening_name == domain.name,]
  domain.s2 <- srs.sample.s2[srs.sample.s2$opening_name == domain.name,]

  n.d.1 <- openings.valid.srs.sample[i, 2]
  n.d.2 <- openings.valid.srs.sample[i, 3]

  domain.p <- df[df$opening_name == domain.name,]
  N.d <- nrow(domain.p)

  white.win.count <- nrow(domain.s1[domain.s1$winner == "white",])
  black.win.count <- nrow(domain.s2[domain.s2$winner == "black",])

  # Vanilla estimates
  white.p <- white.win.count / n.d.1
  black.p <- black.win.count / n.d.2

  estimates[i] <- white.p - black.p
  # Using pooled variance
  pooled.var <- sqrt(((n.d.1-1)*var.est(white.p) + (n.d.2-1)*var.est(black.p))/(n.d.1+n.d.2-2))
  diff.ses[i] <-   pooled.var * sqrt((1-n.d.1/N)*1/n.d.1 + (1-n.d.2/N)*1/n.d.2)
  intervals[i,] <- (white.p - black.p) + qt(0.975, n.d.1+n.d.2-2) * diff.ses[i] * c(-1, 1)
}
```

```r
openings <- data.frame(openings.valid.srs.sample$name, estimates, diff.ses, intervals)
names(openings) <- c("opening name", "Diff. win rate", "SE", "95.CI.lower", "95.CI.upper")
white.higher <- openings[openings$`95.CI.lower` > 0,]
white.lower <- openings[openings$`95.CI.upper` < 0,]
```

```r
openings.df.s1 <- data.frame(table(white.str.sample$opening_name))
openings.df.s2 <- data.frame(table(black.str.sample$opening_name))
names(openings.df.s1) <- c("name", "frequency")
names(openings.df.s2) <- c("name", "frequency")

# Include openings with sample size large enough for usable CI
openings.freq.s1 <- openings.df.s1[openings.df.s1$frequency > 15,]
openings.freq.s2 <- openings.df.s2[openings.df.s2$frequency > 15,]

openings.df.p <- data.frame(table(df$opening_name))
names(openings.df.p) <- c("name", "frequency")

# openings.size.p1 <- openings.df.p[openings.df.p$name %in% openings.freq.s1$name,]
# openings.size.p2 <- openings.df.p[openings.df.p$name %in% openings.freq.s2$name,]

# Include openings with sample sizes yielding the desired CI width
domain.sizes.s1 <- c()
domain.sizes.s2 <- c()

for (name in openings.freq.s1$name) {
  domain.sizes.s1 <- c(domain.sizes.s1, n0 / (1 + n0 / openings.df.p[openings.df.p$name == name
}

for (name in openings.freq.s2$name) {
  domain.sizes.s2 <- c(domain.sizes.s2, n0 / (1 + n0 / openings.df.p[openings.df.p$name == name
}

openings.valid.s1 <- openings.freq.s1[openings.freq.s1$frequency > domain.sizes.s1,]
openings.valid.s2 <- openings.freq.s2[openings.freq.s2$frequency > domain.sizes.s2,]

openings.valid.str.sample <- merge(openings.valid.s1, openings.valid.s2, by = "name")

estimates <- rep(0, nrow(openings.valid.str.sample))
diff.ses <- rep(0, nrow(openings.valid.str.sample))
intervals <- matrix(0, nrow(openings.valid.str.sample), 2)
for (i in 1:nrow(openings.valid.str.sample)) {
  # Find estimate and CI for difference in win rate for white/black
  # for one domain
  domain.name <- openings.valid.str.sample[i, 1]
  domain.s1 <- white.str.sample[white.str.sample$opening_name == domain.name,]
  domain.s2 <- black.str.sample[black.str.sample$opening_name == domain.name,]
```

```r
domain.p <- df[df$opening_name == domain.name,]

n.d.s1 <- openings.valid.str.sample[i, 2]
n.d.s2 <- openings.valid.str.sample[i, 3]

N.d <- nrow(domain.p)
nh.d1 <- domain.s1 %>% count(elo_range, .drop=FALSE)
nh.d2 <- domain.s2 %>% count(elo_range, .drop=FALSE)
Nh.d <- domain.p %>% count(elo_range, .drop=FALSE)
strata.size.prop <- Nh.d[2]/N.d

# Calculate white's win proportion by each strata
white.win.prop <- domain.s1 %>%
  count(winner, .drop=FALSE) %>%
  group_by(elo_range) %>%
  mutate(win.prop = n / sum(n))

# The estimated aggregated win proportion for white
white.prop <- white.win.prop[white.win.prop$winner == "white", ]
white.prop[is.na(white.prop)] <- 0
white.p.str.est <- sum(white.prop$win.prop * strata.size.prop)

# The estimated aggregated variance of win proportion for white
white.se.by.strata <- bind_cols(white.prop, nh = nh.d1$n)
white.se.by.strata <- white.se.by.strata %>% mutate(var.by.strata = win.prop * (1-win.prop)/r
white.se.by.strata <- bind_cols(white.se.by.strata, Nh = Nh.d$n)
white.se.by.strata <- bind_cols(white.se.by.strata, strata.prop.sq = strata.size.prop$n^2)
white.se.by.strata <- white.se.by.strata %>% mutate(strata.prop.sq*(1-nh/Nh)*var.by.strata)
white.se.by.strata[is.na(white.se.by.strata)] <- 0
white.str.se <- sqrt(sum(white.se.by.strata$`strata.prop.sq * (1 - nh/Nh) * var.by.strata`))

# Calculate black's win proportion by each strata
black.win.prop <- domain.s2 %>%
  count(winner, .drop=FALSE) %>%
  group_by(elo_range) %>%
  mutate(win.prop = n / sum(n))

# The estimated aggregated win proportion for white
black.prop <- black.win.prop[black.win.prop$winner == "black", ]
black.prop[is.na(black.prop)] <- 0
black.p.str.est <- sum(black.prop$win.prop * strata.size.prop)

# The estimated aggregated variance of win proportion for black
black.se.by.strata <- bind_cols(black.prop, nh = nh.d2$n)
black.se.by.strata <- black.se.by.strata %>% mutate(var.by.strata = win.prop * (1-win.prop)/r
black.se.by.strata <- bind_cols(black.se.by.strata, Nh = Nh.d$n)
black.se.by.strata <- bind_cols(black.se.by.strata, strata.prop.sq = strata.size.prop$n^2)
```

```
  black.se.by.strata <- black.se.by.strata %>% mutate(strata.prop.sq*(1-nh/Nh)*var.by.strata)
  black.se.by.strata[is.na(black.se.by.strata)] <- 0
  black.str.se <- sqrt(sum(black.se.by.strata$`strata.prop.sq * (1 - nh/Nh) * var.by.strata`))


  # Their difference
  estimates[i] <- white.p.str.est - black.p.str.est
  # Using pooled variance
  diff.ses[i] <- sqrt(white.str.se^2 + black.str.se^2)
  intervals[i,] <- (white.p.str.est - black.p.str.est) + qt(0.975, n.d.s1+n.d.s2- 2) * diff.ses
}



openings <- data.frame(openings.valid.str.sample$name, estimates, diff.ses, intervals)
names(openings) <- c("opening name", "Diff. win rate", "SE", "95.CI.lower", "95.CI.upper")
white.higher <- openings[openings$`95.CI.lower` > 0,]
white.lower <- openings[openings$`95.CI.upper` < 0,]
openings
```