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 phasenewline

———————————————————

1

For every sampling method, we will perform sampling twice. 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 interval is small.

The parameters of Interest

The parameters of Interest are

- 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

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	0.0105612	0.0148004	0.0561996

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	0.0105612	0.0332945	0.0647579

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. There is no other plausible explanation for such phenomenon other than that White has a first-move advantage. (I think the argument is a bit weak here)

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_name) may be very small. In order to ensure that the confidence interval is of reasonable width, we will only estimate if sample size in the domain 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 $p = 0.5$, the initial 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	n in sample 1	n in 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

```
knitr::kable(openings)
```

opening name	Diff. win rate	SE	95.CI.lower	95.CI.upper
French Defense: Knight Variation	-0.0909091	0.0942050	-0.2803210	0.0985028
Scandinavian Defense: Mieses-Kotroc Variation	0.3346774	0.0842921	0.1656088	0.5037460
Scotch Game	0.1962733	0.0820415	0.0319243	0.3606223
Sicilian Defense	-0.1355311	0.0768343	-0.2886975	0.0176352
Sicilian Defense: Bowdler Attack	-0.0228659	0.0764140	-0.1752310	0.1294993
Van't Kruijs Opening	-0.1704545	0.0874960	-0.3458004	0.0048913

Using SRS sampling and assuming equal variances, all openings but "Scandinavian Defense: Mieses-Kotroc Variation" and "Scotch Game" 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	n in sample 1	n in 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 opening above is as follows

opening name	Diff. win rate	SE	95.CI.lower	95.CI.upper
French Defense: Knight Variation	0.0426424	0.0651798	-0.0879285	0.1732133
Scandinavian Defense: Mieses-Kotroc Variation	0.2094644	0.0495076	0.1103999	0.3085289
Scotch Game	-0.0172765	0.0705189	-0.1586583	0.1241054
Sicilian Defense	-0.1609474	0.0549505	-0.2702818	-0.0516131
Sicilian Defense: Bowdler Attack	-0.1600468	0.0686168	-0.2978677	-0.0222258
Van't Kruijs Opening	-0.3068106	0.0539687	-0.4152100	-0.1984112

The confidence intervals for openings "Sicilian Defense", "Sicilian Defense: Bowdler Attack", and " Van't Kruijs Opening" are strictly negative. For these openings, Black has a higher win rate than that of White. For "Scandinavian Defense: Mieses-Kotroc Variation", the constructed confidence interval is strictly positive so White has a higher win rate than that of Black. For the 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 several openings 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		
57.68849	0.9787740		
60.52721	0.9840654		
-2.838722	0.9814232	-4.762277	-0.9151682

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	0.0490262	-4.687201	-1.42992

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

something

Appendix

```r
# 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 ))
```

```r
# 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,]
```

```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
```

```r
# 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)
```

```r
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)
```

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

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


table1 <- t(c(white.p, black.p, srs.se, (white.p-black.p) + z.95 * srs.se * c(-1, 1)))
table1 <- data.frame(table1)
names(table1) <- c("White's win rate", "Black's win rate", "SE of the difference", "Lower Conf
#knitr::kable(table1)

# 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)
```

```r
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
diff.se <- sqrt(white.str.se^2 + black.str.se^2)
(diff.p) + z.95 * diff.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")

# 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")
```

```r
estimates <- 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,]
  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$n)
  white.se.by.strata <- white.se.by.strata %>% mutate(var.by.strata = win.prop * (1-win.prop)/
  white.se.by.strata <- bind_cols(white.se.by.strata, strata.prop.sq = strata.size.prop$n^2)
  white.se.by.strata <- white.se.by.strata %>% mutate(strata.prop.sq*(1-n/nh)*var.by.strata)
  white.se.by.strata[is.na(white.se.by.strata)] <- 0
  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 <- 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
```

```r
    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$n)
    black.se.by.strata <- black.se.by.strata %>% mutate(var.by.strata = win.prop * (1-win.prop)/n
    black.se.by.strata <- bind_cols(black.se.by.strata, strata.prop.sq = strata.size.prop$n^2)
    black.se.by.strata <- black.se.by.strata %>% mutate(strata.prop.sq*(1-n/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 - n/nh) * var.by.strata`))

    # Their difference
    estimates[i] <- white.p.str.est - black.p.str.est
    # Using pooled variance
    diff.se <- sqrt(((n.d.s1-1)*white.str.se^2 + (n.d.s2-1)*black.str.se^2)/(n.d.s1+n.d.s2-2))
    intervals[i,] <- (white.p.str.est - black.p.str.est) + qt(0.975, n.d.s1+n.d.s2- 2) * diff.se
}

openings <- data.frame(openings.valid.str.sample$name, intervals)
colnames(openings) <- c("name", "95.CI.lower", "95.CI.upper")
white.higher <- openings[openings$`95.CI.lower` > 0,]
white.lower <- openings[openings$`95.CI.upper` < 0,]
openings
```

```r
# mean number of turns for white wins vs black wins?
white.win <- srs.sample.s1[srs.sample.s1$winner == "white",]
black.win <- srs.sample.s2[srs.sample.s2$winner == "black",]

white.win.turns.avg <- mean(white.win$turns)
black.win.turns.avg <- mean(black.win$turns)

n.w <- nrow(white.win)
n.b <- nrow(black.win)

srs.se <- sqrt((1-n.w/N)*var(white.win$turns)/n.w + (1-n.b/N)*var(black.win$turns)/n.b)
(white.win.turns.avg - black.win.turns.avg) + z.95 * srs.se * c(-1, 1)
```

```r
white.win <- white.str.sample[white.str.sample$winner == "white",]
black.win <- black.str.sample[black.str.sample$winner == "black",]

# Calculate average number of turns for white win by each strata
avg.turns.w <- white.win %>%
  group_by(elo_range) %>%
  summarise(mean_turns = mean(turns))

# and for black win
avg.turns.b <- black.win %>%
  group_by(elo_range) %>%
```

```r
    summarise(mean_turns = mean(turns))

# The estimated average number of turns
white.avg <- sum(Nh[2]/N * avg.turns.w$mean_turns)
black.avg <- sum(Nh[2]/N * avg.turns.b$mean_turns)

# The estimated se by strata
white.turn.strata  <- white.win %>% group_by(elo_range) %>% summarise(var = var(turns))
white.turn.strata <- bind_cols(white.turn.strata, nh = nh.white$n)
white.turn.strata <- bind_cols(white.turn.strata, Nh = Nh[2])
white.se.by.strata <-white.turn.strata %>% mutate(var.by.str =  (1-nh/n)*var/nh)
white.se.by.strata <- bind_cols(white.se.by.strata, strata.size.prop)
white.se.by.strata <- sqrt(sum(white.se.by.strata$n...6^2 * white.se.by.strata$var.by.str))

black.turn.strata  <- black.win %>% group_by(elo_range) %>% summarise(var = var(turns))
black.turn.strata <- bind_cols(black.turn.strata, nh = nh.black$n)
black.turn.strata <- bind_cols(black.turn.strata, Nh = Nh[2])
black.se.by.strata <-black.turn.strata %>% mutate(var.by.str =  (1-nh/n)*var/nh)
black.se.by.strata <- bind_cols(black.se.by.strata, strata.size.prop)
black.se.by.strata <- sqrt(sum(black.se.by.strata$n...6^2 * black.se.by.strata$var.by.str))

diff.se <- sqrt(white.se.by.strata^2 + black.se.by.strata^2)
(white.avg - black.avg) + z.95 * diff.se * c(-1, 1)
```