

Elo Calculations

Pricilla

2025-09-25

```
library(knitr)
library(ggplot2)
library(forcats)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(stringr)
```

To compute each player's expected score and the difference vs actual, we use the Elo expected score formula for each game:

The formula I am using comes directly from Arpad Elo's original rating system, which was introduced in the 1960s to improve the older chess rating methods used by the US Chess Federation (USCF). Later, FIDE (the international chess federation) formally adopted it in 1970 as the official system for international chess ratings.

In this chess tournament dataset, the Elo calculation is based on two key columns:

- **PreRating** → the player's rating before the tournament.
- **AvgOpponentRating** → the average rating of all the opponents that player faced.

The Elo expected score formula is:

$$E = \frac{1}{1 + 10^{\frac{(R_{opp} - R_{player})}{400}}}$$

Where:

- R_{player} = the player's **PreRating**

- R_{opp} = the **AvgOpponentRating**

This gives the expected score for each player in the tournament. Returns a value between 0 and 1, representing the expected fraction of points in one game.

Then the new rating is updated after comparing the actual score (from **Points**) vs the expected score E :

$$R_{new} = R_{old} + K \times (S - E)$$

Where:

- R_{old} = PreRating
- S = actual score (from Points)
- E = expected score (from formula above)
- K = development factor (often 32, 24, or 16 depending on federation and rating level).

So in this dataset:

- **PreRating** and **AvgOpponentRating** are the inputs for expected Elo calculation.
- **Points** (tournament score) is compared against the expected score to determine how the player's rating should adjust.

“Gary Hua Elo Calculation Step by Step”

- **Actual Points:** $S = 6.0$
- **PreRating:** $R_{player} = 1794$
- **Avg Opponent Rating:** $R_{opp} = 1605$
- **K-factor:** $K = 32$ (typical)

Compute Expected Score

The Elo expected score formula:

$$E = \frac{1}{1 + 10^{\frac{R_{opp} - R_{player}}{400}}}$$

Substitute Gary's values:

$$E = \frac{1}{1 + 10^{\frac{1605 - 1794}{400}}}$$

Compute the exponent first:

$$1605 - 1794 = -189$$

$$\frac{-189}{400} = -0.4725$$

Then:

$$10^{-0.4725} \approx 0.336$$

$$1 + 0.336 = 1.336$$

$$E = \frac{1}{1.336} \approx 0.748$$

This is the expected score per game against a 1605 rated opponent. Returns a value between 0 and 1, representing the expected fraction of points in one game.

Scale to Tournament Points

Assuming 7 rounds in the tournament:

$$\text{Expected Points} \approx 0.748 \times 7 \approx 5.24$$

Elo expects Gary to score about 5.24 points.

Compute Difference vs Actual

$$\text{Diff} = \text{Actual Points} - \text{Expected Points}$$

$$\text{Diff} = 6.0 - 5.2 = 0.8$$

Gary slightly outperformed

Load Tournament.txt from Github

https://raw.githubusercontent.com/prnakyzazze94/Data_607/refs/heads/main/Class%20Tournament.txt

```
# Read the file
lines <- readLines("https://raw.githubusercontent.com/prnakyzazze94/Data_607/refs/heads/main/Class%20Tournament.txt")
```

Filter only player lines information.

```

# Remove separator lines and blank lines
lines <- lines[!grepl("^--+|^\\s*$", lines)]

# Remove the two header lines
lines <- lines[-c(1, 2)]

# Separate into player + info lines
player_lines <- lines[seq(1, length(lines), by = 2)]
info_lines <- lines[seq(2, length(lines), by = 2)]

```

create df

```

results <- data.frame(
  PairNum = integer(),
  Name = character(),
  State = character(),
  USCF_ID = character(),
  PreRating = numeric(),
  PostRating = numeric(),
  TotalScore = numeric(),
  stringsAsFactors = FALSE
)

```

Extract all Player information, PreRating, Points, State, OpponentNums, OpponentPreRatings calculate Average(mean) and Prepare output data frame

```

# Extract all pre-ratings once (used for AvgOpponentRating)
pre_ratings <- sapply(info_lines, function(info) {
  rating_match <- regmatches(info, regexpr("R:\\s*\\d+", info))
  as.integer(gsub("R:\\s*", "", rating_match))
})

# Prepare output data frame
output <- data.frame(
  Player = character(),
  State = character(),
  Points = numeric(),
  PreRating = integer(),
  AvgOpponentRating = numeric(),
  PlayerNum = integer(),
  OpponentNums = character(),
  OpponentPreRatings = character(),
  stringsAsFactors = FALSE
)

# Loop through players
for (i in seq_along(player_lines)) {
  pl <- player_lines[i]
  info <- info_lines[i]

  # Extract clean player name (remove leading number and pipe)
  name <- trimws(gsub("^\\d+\\s*\\|\\s*", "", substr(pl, 5, 36)))
}

```

```

# Extract points
points_match <- regmatches(pl, regexpr("\\|\\s*[0-9]+\\.?[0-9]*\\s*\\|", pl))
points <- as.numeric(gsub("[^0-9.]", "", points_match))

# Extract state (e.g., ON, MI)
state_match <- regmatches(info, regexpr("\\b[A-Z]{2}\\b", info))
state <- if (length(state_match) > 0) state_match else NA

# Extract pre-rating
rating_match <- regmatches(info, regexpr("R:\\s*\\d+", info))
pre_rating <- as.integer(gsub("R:\\s*", "", rating_match))

# Extract opponent indices from line 1 (rounds)
rounds <- unlist(strsplit(pl, "\\|"))
rounds <- rounds[4:length(rounds)] # skip first 3 parts
opp_indices <- as.integer(gsub("[^0-9]", "", rounds))
opp_indices <- opp_indices[!is.na(opp_indices)]

# Calculate average opponent rating
if (length(opp_indices) > 0) {
  opp_ratings <- pre_ratings[opp_indices]
  avg_opp_rating <- round(mean(opp_ratings, na.rm = TRUE), 0)
} else {
  avg_opp_rating <- NA
}

# Add row to output
output <- rbind(output, data.frame(
  Player = name,
  State = state,
  Points = points,
  PreRating = pre_rating,
  AvgOpponentRating = avg_opp_rating,
  stringsAsFactors = FALSE
))
}

p <- 10

# Print only the first p rows
print(head(output, n = p))

```

```

##           Player State Points PreRating AvgOpponentRating
## 1          GARY HUA   ON    6.0      1794             1605
## 2    DAKSHESH DARURI   MI    6.0      1553             1469
## 3        ADITYA BAJAJ   MI    6.0      1384             1564
## 4  PATRICK H SCHILLING   MI    5.5      1716             1574
## 5          HANSHI ZUO   MI    5.5      1655             1501
## 6          HANSEN SONG   OH    5.0      1686             1519
## 7    GARY DEE SWATHELL   MI    5.0      1649             1372
## 8    EZEKIEL HOUGHTON   MI    5.0      1641             1468
## 9        STEFANO LEE   ON    5.0      1411             1523
## 10         ANVIT RAO   MI    5.0      1365             1554

```

```
# View output
#print(output)
```

calculate each player's expected score (e.g. 4.3) and the difference from their actual score (e.g 4.0)

```
# --- Function to compute expected score for one game ---
elo_expect <- function(r_player, r_opp) {
  1 / (1 + 10 ^ ((r_opp - r_player) / 400))
}

# --- Prepare an output frame with consistent columns ---
output <- data.frame(
  Player = character(),
  Points = numeric(),
  PreRating = integer(),
  AvgOpponentRating = numeric(),
  ExpectedPoints = numeric(),
  Diff = numeric(),
  NewRating = numeric(),
  stringsAsFactors = FALSE
)

n_players <- length(pre_ratings) # assumes pre_ratings exists
K <- 32 # You can adjust this value depending on federation rules

# --- Loop through each player ---
for (i in seq_along(player_lines)) {
  pl <- player_lines[i]
  info <- info_lines[i]

  # Extract player name
  name <- trimws(gsub("^\\d+\\s*\\|\\s*", "", substr(pl, 5, 36)))

  # Extract points
  points_match <- regmatches(pl, regexpr("\\|\\s*[0-9]+\\.?[0-9]*\\s*\\|", pl))
  points <- as.numeric(gsub("[^0-9.]", "", points_match))

  # Extract pre-rating
  rating_match <- regmatches(info, regexpr("R:\\s*\\d+", info))
  pre_rating <- if (length(rating_match) > 0) as.integer(gsub("R:\\s*", "", rating_match)) else NA

  # Extract opponent indices from round columns
  rounds <- unlist(strsplit(pl, "\\|"))
  if (length(rounds) >= 4) {
    rounds <- rounds[4:length(rounds)]
    opp_indices <- as.integer(str_extract(rounds, "\\d+"))
    opp_indices <- opp_indices[!is.na(opp_indices)]
    opp_indices <- opp_indices[opp_indices >= 1 & opp_indices <= n_players]
  } else {
    opp_indices <- integer(0)
  }

  # Compute average opponent rating and expected score
```

```

if (length(opp_indices) > 0) {
  opp_ratings <- pre_ratings[opp_indices]
  avg_opp_rating <- round(mean(opp_ratings, na.rm = TRUE), 0)
  exp_per_game <- elo_expect(pre_rating, opp_ratings)
  expected_points <- sum(exp_per_game, na.rm = TRUE)
} else {
  avg_opp_rating <- NA
  expected_points <- NA
}

#Difference vs Actual Score
diff_points <- if (!is.na(expected_points)) points - expected_points else NA

# --- New Elo Rating calculation
new_rating <- if (!is.na(expected_points)) {
  pre_rating + K * (points - expected_points)
} else {
  NA
}

# Append to output dataframe
output <- rbind(output, data.frame(
  Player = name,
  Points = points,
  PreRating = pre_rating,
  AvgOpponentRating = avg_opp_rating,
  ExpectedPoints = expected_points,
  Diff = diff_points,
  NewRating = new_rating,
  stringsAsFactors = FALSE
))
}

# --- Rounding & formatting ---
output <- output %>%
  mutate(
    ExpectedPoints = round(ExpectedPoints, 1),
    Diff = round(Diff, 2),
    NewRating = round(NewRating, 0)
  )

final_output <- output %>%
  select(Player, Points, ExpectedPoints, Diff, PreRating, AvgOpponentRating, NewRating)

#print(final_output)
kable(
  final_output,
  caption = "Tournament Results with Elo Expected Points and Updated Ratings",
  align = "lcccccc"
)

```

Table 1: Tournament Results with Elo Expected Points and Updated Ratings

Player	Points	ExpectedPoints	Diff	PreRating	AvgOpponentRating	NewRating
GARY HUA	6.0	5.2	0.84	1794	1605	1821
DAKSHESH DARURI	6.0	3.8	2.22	1553	1469	1624
ADITYA BAJAJ	6.0	1.9	4.05	1384	1564	1514
PATRICK H SCHILLING	5.5	4.7	0.76	1716	1574	1740
HANSHI ZUO	5.5	4.4	1.12	1655	1501	1691
HANSEN SONG	5.0	4.9	0.06	1686	1519	1688
GARY DEE	5.0	4.6	0.42	1649	1372	1662
SWATHELL						
EZEKIEL HOUGHTON	5.0	5.0	-0.03	1641	1468	1640
STEFANO LEE	5.0	2.3	2.71	1411	1523	1498
ANVIT RAO	5.0	1.9	3.06	1365	1554	1463
CAMERON WILLIAM MC LEMAN	4.5	5.3	-0.84	1712	1468	1685
KENNETH J TACK	4.5	4.1	0.39	1663	1506	1676
TORRANCE HENRY JR	4.5	5.0	-0.45	1666	1498	1652
BRADLEY SHAW	4.5	4.2	0.32	1610	1515	1620
ZACHARY JAMES HOUGHTON	4.5	1.4	3.13	1220	1484	1320
MIKE NIKITIN	4.0	3.8	0.20	1604	1386	1610
RONALD	4.0	4.7	-0.66	1629	1499	1608
GRZEGORCZYK						
DAVID SUNDEEN	4.0	4.6	-0.59	1600	1480	1581
DIPANKAR ROY	4.0	4.3	-0.33	1564	1426	1554
JASON ZHENG	4.0	5.1	-1.13	1595	1411	1559
DINH DANG BUI	4.0	4.3	-0.32	1563	1470	1553
EUGENE L MCCLURE	4.0	4.5	-0.48	1555	1300	1540
ALAN BUI	4.0	3.9	0.06	1363	1214	1365
MICHAEL R ALDRICH	4.0	2.6	1.45	1229	1357	1275
LOREN SCHWIEBERT	3.5	6.3	-2.78	1745	1363	1656
MAX ZHU	3.5	4.1	-0.60	1579	1507	1560
GAURAV GIDWANI	3.5	4.0	-0.50	1552	1222	1536
SOFIA ADINA	3.5	3.3	0.19	1507	1522	1513
STANESCU-BELLU						
CHIEDOZIE OKORIE	3.5	4.6	-1.10	1602	1314	1567
GEORGE AVERY	3.5	6.0	-2.52	1522	1144	1441
JONES						
RISHI SHETTY	3.5	5.1	-1.59	1494	1260	1443
JOSHUA PHILIP	3.5	3.7	-0.22	1441	1379	1434
MATHEWS						
JADE GE	3.5	4.6	-1.14	1449	1277	1412
MICHAEL JEFFERY	3.5	3.4	0.06	1399	1375	1401
THOMAS						
JOSHUA DAVID LEE	3.5	5.0	-1.46	1438	1150	1391
SIDDHARTH JHA	3.5	2.7	0.80	1355	1388	1381
AMIYATOSH	3.5	0.8	2.73	980	1385	1067
PWNNANANDAM						
BRIAN LIU	3.0	2.1	0.87	1423	1539	1451

Player	Points	ExpectedPoints	Diff	PreRating	AvgOpponentRating	NewRating
JOEL R HENDON	3.0	3.6	-0.62	1436	1430	1416
FOREST ZHANG	3.0	2.9	0.06	1348	1391	1350
KYLE WILLIAM MURPHY	3.0	2.4	0.64	1403	1248	1423
JARED GE	3.0	5.0	-2.01	1332	1150	1268
ROBERT GLEN VASEY	3.0	4.3	-1.33	1283	1107	1240
JUSTIN D SCHILLING	3.0	2.1	0.93	1199	1327	1229
DEREK YAN	3.0	4.4	-1.37	1242	1152	1198
JACOB ALEXANDER LAVALLEY	3.0	0.0	2.96	377	1358	472
ERIC WRIGHT	2.5	3.2	-0.69	1362	1392	1340
DANIEL KHAIN	2.5	2.5	-0.03	1382	1356	1381
MICHAEL J MARTIN	2.5	2.5	-0.04	1291	1286	1290
SHIVAM JHA	2.5	1.4	1.08	1056	1296	1090
TEJAS AYYAGARI	2.5	1.0	1.47	1011	1356	1058
ETHAN GUO	2.5	0.3	2.20	935	1495	1006
JOSE C YBARRA	2.0	1.7	0.28	1393	1345	1402
LARRY HODGE	2.0	3.4	-1.40	1270	1206	1225
ALEX KONG	2.0	1.4	0.56	1186	1406	1204
MARISA RICCI	2.0	1.1	0.92	1153	1414	1182
MICHAEL LU	2.0	1.3	0.70	1092	1363	1114
VIRAJ MOHILE	2.0	0.4	1.57	917	1391	967
SEAN M MC CORMICK	2.0	0.4	1.59	853	1319	904
JULIA SHEN	1.5	0.6	0.90	967	1330	996
JEZZEL FARKAS	1.5	1.0	0.53	955	1327	972
ASHWIN BALAJI	1.0	0.9	0.12	1530	1186	1534
THOMAS JOSEPH HOSMER	1.0	1.4	-0.43	1175	1350	1161
BEN LI	1.0	2.3	-1.27	1163	1263	1122

Top 5 overperformers

Top 5 overperformers

```
kable((output %>%
  arrange(desc(Diff)) %>%
  slice_head(n = 5) %>%
  select(Player, Points, ExpectedPoints, Diff, PreRating, NewRating, AvgOpponentRating,)))
```

Player	Points	ExpectedPoints	Diff	PreRating	NewRating	AvgOpponentRating
ADITYA BAJAJ	6.0	1.9	4.05	1384	1514	1564
ZACHARY JAMES HOUGHTON	4.5	1.4	3.13	1220	1320	1484
ANVIT RAO	5.0	1.9	3.06	1365	1463	1554
JACOB ALEXANDER LAVALLEY	3.0	0.0	2.96	377	472	1358
AMIYATOSH PWNANANDAM	3.5	0.8	2.73	980	1067	1385

Aditya's PreRating is 1384.

His Opponents averaged 1564, which is 180 points higher.

Elo predicts Aditya should lose most games, his expected score is 1.9 points. But he actually scored 6 points beating much stronger players.

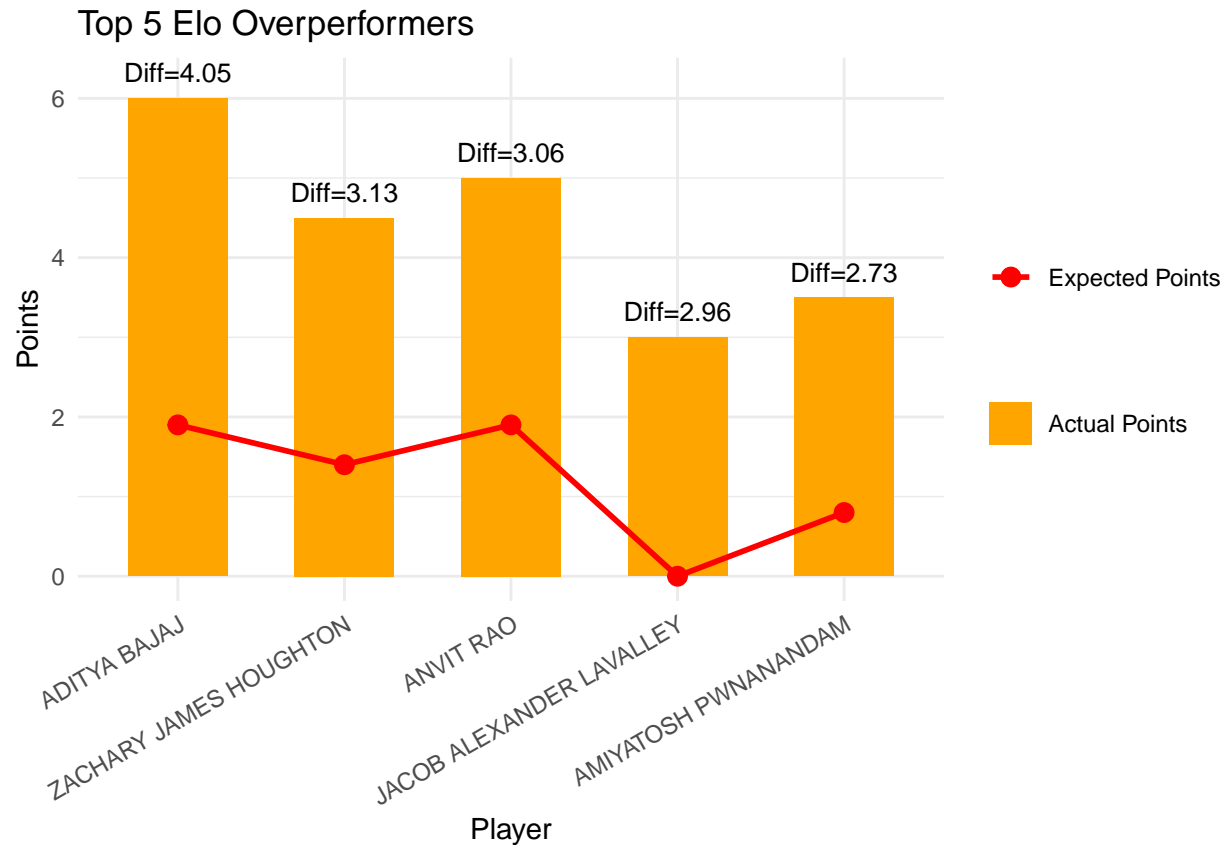
As a result, Elo rewarded him with +130 rating points, bringing his new prerating to 1514.

PLOT TO VISUALLY SHOW ELO TOP PERFORMERS

```
top5 <- output %>%
  arrange(desc(Diff)) %>%
  slice_head(n = 5) %>%
  select(Player, Points, ExpectedPoints, Diff, PreRating, NewRating, AvgOpponentRating)

# Plot: Actual vs Expected Points with Diff labels
ggplot(top5, aes(x = reorder(Player, -Diff))) +
  geom_col(aes(y = Points, fill = "Actual Points"), width = 0.6) +
  geom_line(aes(y = ExpectedPoints, group = 1, color = "Expected Points"), size = 1) +
  geom_point(aes(y = ExpectedPoints, color = "Expected Points"), size = 3) +
  geom_text(aes(y = Points + 0.2, label = paste0("Diff=", round(Diff, 2))),
    vjust = 0, size = 3.5) +
  scale_fill_manual("", values = c("Actual Points" = "orange")) +
  scale_color_manual("", values = c("Expected Points" = "red")) +
  labs(title = "Top 5 Elo Overperformers",
    x = "Player", y = "Points") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



You can see that all these players scored well above Elo expectations, which explains why each of them gained rating points.

All five players in this plot overperformed relative to Elo.

The red “expected” line sits much lower than the orange bars, this visual gap highlights how much each exceeded expectations.

These players see big Elo rating gains after the tournament.

The biggest standout is Aditya Bajaj, with over 4 points more than expected.

5 TOP UNDER PERFORMERS

Top 5 underperformers

```
kable((output %>%
  arrange(Diff) %>%
  slice_head(n = 5) %>%
  select(Player, Points, ExpectedPoints, Diff, PreRating, NewRating, AvgOpponentRating)))
```

Player	Points	ExpectedPoints	Diff	PreRating	NewRating	AvgOpponentRating
LOREN SCHWIEBERT	3.5	6.3	-2.78	1745	1656	1363
GEORGE AVERY JONES	3.5	6.0	-2.52	1522	1441	1144
JARED GE	3.0	5.0	-2.01	1332	1268	1150
RISHI SHETTY	3.5	5.1	-1.59	1494	1443	1260

Player	Points	ExpectedPoints	Diff	PreRating	NewRating	AvgOpponentRating
JOSHUA DAVID LEE	3.5	5.0	-1.46	1438	1391	1150

LOREN SCHWIEBERT's PreRating is 1745 which suggests Loren was much stronger than his opponents with a 1363 averageopponents preRating.

Elo expected him to dominate 6.3 points.

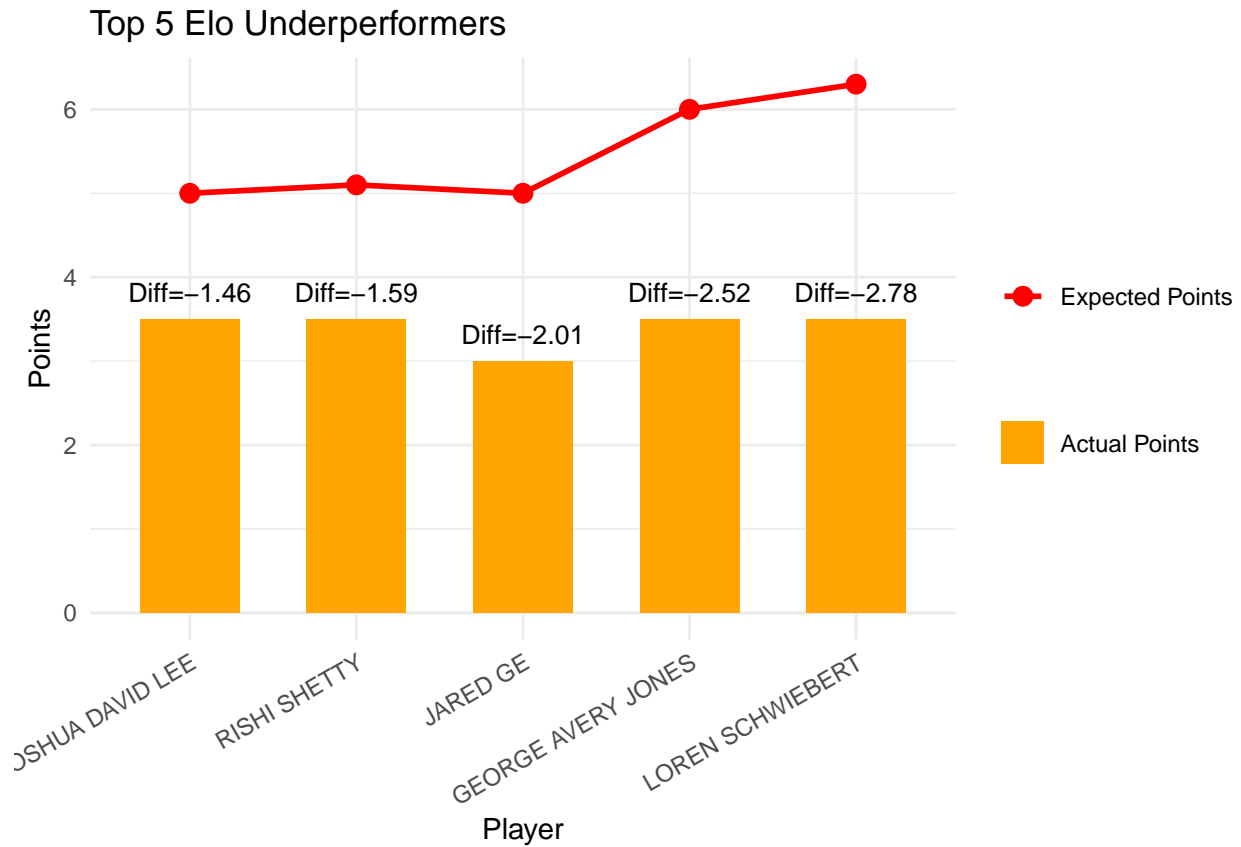
But he only scored 3.5.

Therefore, Elo deducted about 89 rating points, giving him a NewRating of 1656 which better reflects his actual tournament strength.

PLOT TO VISUALLY SHOW ELO TOP UNDER PERFORMERS

```
lower5 <- output %>%
  arrange(Diff) %>%
  slice_head(n = 5) %>%
  select(Player, Points, ExpectedPoints, Diff, PreRating, NewRating, AvgOpponentRating)

# Plot: Actual vs Expected Points with Diff labels
ggplot(lower5, aes(x = reorder(Player, -Diff))) +
  geom_col(aes(y = Points, fill = "Actual Points"), width = 0.6) +
  geom_line(aes(y = ExpectedPoints, group = 1, color = "Expected Points"), size = 1) +
  geom_point(aes(y = ExpectedPoints, color = "Expected Points"), size = 3) +
  geom_text(aes(y = Points + 0.2, label = paste0("Diff=", round(Diff, 2))),
    vjust = 0, size = 3.5) +
  scale_fill_manual("", values = c("Actual Points" = "orange")) +
  scale_color_manual("", values = c("Expected Points" = "red")) +
  labs(title = "Top 5 Elo Underperformers",
    x = "Player", y = "Points") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



The red “expected” line sits much higher above the orange bars, this visual gap highlights how much each player did not meet expectations.

These players see big Elo rating gains after the tournament.

The biggest standout is LOREN SCHWIEBERT, with 2.78 points less than expected.