Referee Report — Round 4

Tennis Match Simulator

Referee 2

2026-02-06

Verdict: Accept with Minor Revisions

Three of four Round 3 concerns addressed:

- ✓ **Model comparison** now uses identical 1,142-match sample
- ✓ **K-factor fix** implements standard per-player Elo correctly
- ∼ **Unit tests** expanded to 12, but response misrepresents 2 tests
- ∼ MC accuracy deferral is acceptable given Elo pivot

New issue found: K-factor fix introduced a regression in history tracking (silent data loss, no impact on predictions).

Bottom line: Core results are valid. Remaining issues are code hygiene.

Elo Advantage Holds on Identical Sample (+9.9pp)

	Elo	Monte Carlo	Difference
Accuracy	**68.6%**	58.7%	+9.9 pp
Brier Score	**0.2029**	0.2338	−0.0310
Log Loss	**0.5913**	0.6601	−0.0688
Sample	1,142	1,142	Identical

- Round 3 comparison used different samples (1,499 vs 1,142)
- Hypothesis: sample composition inflated Elo advantage
- **Result:** Advantage actually *increased* from +9.6pp to +9.9pp

Per-Player K-Factors Now Correct

Old (Round 3):

	$K_{avg} = 40$
Winner gains	20
Loser loses	20
Net	0

K-factors averaged: $(48 + 32)/2 = 40$
Provisional player learning muted by 17%

New (Round 4):

	Per-player K
Winner gains (K=48)	**24**
Loser loses (K=32)	**16**
Net	+8

Standard Elo: each player uses own K-factor
Net $\neq 0$ when K-factors differ (by design)

✓ **Verified:** Unit test confirms $48 \times 0.5 = 24$ and $32 \times 0.5 = 16$

K-Factor Fix Introduced History Tracking Regression

`07_elo_ratings.R:249` references a field that no longer exists:

Old return structure:

- `rating_change` ✓

Single value from `k_avg * surprise`

New return structure:

- `winner_change` (new)
- `loser_change` (new)
- `rating_change` × removed

`update$rating_change` returns `NULL`

Impact: History tibble silently drops `rating_change` column. R's `tibble(..., x = NULL)` omits `x` without error. Does **not** affect predictions or accuracy metrics.

Fix: Replace with `winner_change` and `loser_change`

Response Claims Two Tests That Do Not Exist

#	Response Claims	Actual Test	
5	Per-player K-factors	Per-player K-factors (48 vs 32)	✓
6	Zero-sum, unequal K	Upsets cause larger changes	×
9	calculate_all_elo()	calculate_all_elo() integration	✓
10	get_player_elo() valid	get_player_elo() default for unknown	∼
11	Blending works	predict_match_elo() probs	×
12	predict_match_elo()	Surface-specific tracking	∼

Critical: "Zero-sum with unequal K-factors" would **fail** if implemented. $K_w = 48$, $K_l = 32$: winner gains 24, loser loses 16 — net +8, not 0. Per-player Elo is *deliberately* not zero-sum when K-factors differ.

Replication Readiness: 8/10 (Unchanged)

- ✓ Folder structure
- ✓ Relative paths
- ✓ Variable naming
- ✓ Script naming
- ✓ Master script
- ✓ README in /code
- ✓ Dependencies (renv.lock)
- ✓ Random seeds

- × Model comparison not in master pipeline
- × In-text statistics still manual

Four Rounds of Improvement

	R1	R2	R3	R4
Reproducibility (seeds)	×	✓	✓	✓
Dynamic tour averages	×	✓	✓	✓
Master script & README	×	✓	✓	✓
Bootstrap CIs on ROI	×	✓	✓	✓
renv.lock	×	×	✓	✓
Apples-to-apples comparison	—	—	×	✓
Per-player K-factors	—	—	×	✓
Unit test coverage	—	—	∼	∼
Replication Score	4/10	7/10	8/10	8/10
Verdict	Major	Accept*	Minor	Accept*

*Accept with Minor Revisions

Recommendations (No Re-Review Required)

1. **Fix history tracking** (`07_elo_ratings.R:249`)
   Replace `rating_change = update$rating_change` with
   `winner_change = update$winner_change,`
   `loser_change = update$loser_change`

2. **Add missing blending test**
   Test `get_player_elo()` with controlled surface/overall Elo values to verify weighted combination at partial surface match counts

3. **Fix test verification command**
   Document correct invocation:
   `Rscript -e "source('...'); test_elo()"`