# Agent-based modeling: Persistence in lottery gambling (*Euromillions FDJ France*)

## Table of Contents



---

## Background and motivation

I have been playing the lottery for a long time and I lose more often than I win, but I continue to play. Talking with my friends and family, I noticed that many others do the same. This behavior seemed irrational, even absurd, to me. So I decided to take advantage of the courses on agent-based models (ABM) to try to understand this phenomenon.

---

## Purpose of the model

**Goal:** Understand why players continue to play the lottery despite repeated losses by integrating individual heterogeneity, cognitive biases, and social influence.

### Main hypotheses

1. Each player has:
    - a limited **budget**,
    - a subjective **belief** in their luck,
    - a **risk propensity**,
    - a **stubbornness** (psychological resistance to stopping).
2. Players observe and are influenced by the beliefs/behaviors of neighbors within a social radius.
3. Rare wins, cognitive biases (e.g., gambler's fallacy, hot-hand), and social influence can maintain persistent gambling behavior.

---

## ODD Protocol

### Overview

- Specific objective: quantify how repeated losses, rare wins, and social interactions shape the fraction of active players and collective trajectories such as average belief, percent playing, tickets sold, and winners.
- Time scale: discrete ticks. Default population: N = 1000 (parameter).
- Spatial/social scale: agents may be embedded in a 2D space or a network; social interactions are within a specified social radius (or neighbor set).

## Entities, state variables, and scales

- Entities:
  - Agents (players / turtles)
  - Global environment (lottery parameters and collectors)
- Global parameters:
  - jackpot (monetary), ticket_price (monetary)
  - base_win_prob (per-ticket Bernoulli probability)
  - influence_strength $\beta_{infl} \in [0,1]$
  - social_radius (distance in patches or neighbor count)
  - play_threshold (decision threshold)
  - seed (random seed for reproducibility)
- Per-agent state variables:
  - budget (monetary)
  - belief $\in [0,1]$ (subjective expectation of winning or perceived utility)
  - plays? (boolean; whether the agent continues trying)
  - risk_propensity $\in [0,1]$
  - stubbornness $\in [0,1]$
  - last_win_tick (integer; -1 if never)
  - consecutive_losses (integer)
- Typical scales and dimensions:
  - N ≈ 1000 (parametric)
  - ticks: simulation length (e.g., 10,000)
  - interactions: local (social_radius) or global (mean-field), specified per experiment

## Process overview and scheduling

Per tick:

1. Randomly shuffle agents to avoid ordering bias.
2. For each agent (in shuffled order): a. Observe neighbors_belief = mean(belief_j) for j in social neighborhood.
   b. Compute decision_score (see Submodels).
   c. If decision_score > play_threshold and budget ≥ ticket_price and plays? = true:
   - Budget ← budget − ticket_price (purchase operation).
   - Mark that the agent bought 1 ticket this tick. d. For each ticket bought, perform draw: win ~ Bernoulli(base_win_prob).
   - If win: budget ← budget + jackpot; last_win_tick ← tick; consecutive_losses ← 0.
   - If loss: consecutive_losses ← consecutive_losses + 1. e. Update belief according to the outcome and forgetting/memory rules.
     f. Evaluate stopping rule: if budget < ticket_price → plays? ← false; else evaluate probabilistic

stop rule (prob_stop) modulated by stubbornness. g. Record per-agent metrics (for debugging / optional logs).

3. After all agents processed, aggregate global statistics for the tick: avg_belief, pct_playing, tickets_sold, winners, total_payout, mean_budget.
4. Proceed to next tick until maximum ticks or manual/automated termination condition.

## Design concepts

- Emergence: macro patterns (persistence, waves of play, stationary active fraction) arise from micro decision rules plus stochastic wins.
- Adaptation and learning: belief changes after outcomes and decays over time (memory).
- Interaction: neighbors' beliefs increase/decrease an agent's propensity to play (additive influence or alternative imitation rules).
- Objectives: agents act to maximize perceived utility (implicitly encoded in decision_score), not necessarily monetary optimality.
- Stochasticity: randomness in draws, initial distributions, and execution order; stochastic stopping and cognitive-bias effects.
- Collectives: no formal groups required; clusters of similar belief/behavior can emerge spatially or across the social network.
- Sensing: agents observe neighbors' beliefs (not directly budgets or wins unless modelled), and they know their own outcomes and budget.

## Initialization

Default initialisation:

- Random seed: provided and recorded for every run.
- N = 1000 (parameter).
- budget_i ~ LogNormal(μ=6, σ=1) (monetary units) or set a fixed distribution depending on experiment.
- belief_i ~ Uniform(0.2, 0.8).
- risk_propensity_i ~ Beta(2,5).
- stubbornness_i ~ Uniform(0,1).
- plays? = true, last_win_tick = -1, consecutive_losses = 0.
- Default global parameters:
  - jackpot = 100000
  - ticket_price = 2
  - base_win_prob = 1e-6
  - influence_strength β_infl = 0.3
  - play_threshold = 0.5
  - α_risk = 0.5
  - Δ_win = 0.10
  - Δ_loss = 0.02
  - λ_memory = 0.01

## Submodels and formulas

Explicit mathematical definitions for each submodel below; parameter ranges are indicated in parentheses.

1. Decision score (purchase propensity)

- neighbors_belief = mean(belief_j) for j in neighborhood
- decision_score = belief · (1 + α_risk · risk_propensity) + β_infl · neighbors_belief
- α_risk ∈ [0,1]; β_infl ∈ [0,1]
- Purchase condition: decision_score > play_threshold and budget ≥ ticket_price and plays? = true

2. Ticket purchase and draw

- When purchase occurs: budget ← budget − ticket_price
- Win outcome: win ~ Bernoulli(base_win_prob). If win: budget ← budget + jackpot

3. Belief update and memory

- Forgetting / memory decay (applied each tick or after update):
  - belief ← belief · exp(−λ_memory) where λ_memory ≥ 0
- Outcome-based update:
  - if win: belief ← min(1, belief + Δ_win)
  - if loss: belief ← max(0, belief − Δ_loss)
- Optional cognitive-bias adjustments:
  - Gambler's fallacy (parameterized): after k consecutive losses, belief ← belief + p_gf · gf_delta (p_gf ∈ [0,1])
  - Hot-hand: recent wins increase belief proportionally to recent_win_count · hot_delta

4. Stopping rule

- Deterministic stop: if budget < ticket_price → plays? ← false
- Probabilistic stop: prob_stop = max(0, γ · (stop_belief_threshold − belief)) · (1 − stubbornness)
  - γ ≥ 0; stubbornness ∈ [0,1]
  - If U(0,1) < prob_stop then plays? ← false

5. Aggregation statistics

- avg_belief(tick) = mean_i belief_i
- pct_playing(tick) = 100 × (agents with plays? = true) / N
- tickets_sold(tick) = sum of tickets bought this tick
- winners(tick) = number of winning draws this tick
- total_payout(tick) = sum of payouts paid to winners
- mean_budget(tick) and belief distribution histogram

## Outputs, data format, and reproducibility

- Per-run CSV output (one line per tick) with standardized columns:
  - run_id, seed, tick, avg_belief, pct_playing, tickets_sold, winners, total_payout, mean_budget, params_hash
- Save an accompanying metadata file (runs_metadata.json) with:
  - model_version, git_commit_hash, date, author, paramset_name, RNG seed list
- Save per-run parameter snapshot (PARAMS file) to ensure exact replication.
- All experiments must record the seed and any system-specific details required to reproduce results.

## Experimental protocol and tests

- Repetitions: perform at least 30–50 independent runs per parameter configuration to estimate variance (report mean ± 95% CI).
- Default sensitivity grid:
  - β_infl ∈ {0, 0.2, 0.5, 0.8}
  - α_risk ∈ {0, 0.5, 1}
  - Δ_loss ∈ {0.01, 0.05, 0.1}
  - stubbornness ∈ {0.1, 0.5, 0.9}
- Stationarity test: define ε and window W (e.g., ε = 0.01, W = 100 ticks); consider pct_playing stable if relative variation < ε over W ticks.
- Unit and sanity tests (to automate in headless mode):
  - Reproducibility: same seed → identical time series
  - Edge-case logic: ticket_price = 0 and decision_score > play_threshold → all eligible agents buy; verify
  - No social influence (β_infl = 0) → spatial/neighborhood correlation in beliefs should vanish (statistical test)
  - Conservation checks where applicable (e.g., budget bookkeeping)
- Provide headless run scripts and an experiments folder with paramsets (.toml or .csv) and a run_batch.sh (or PowerShell) wrapper.

---

## Note:

- document exact choices in PARAMETERS.md of initialisation variables
- all must be documented with units/ranges in PARAMETERS.md about global params
- CI:Confidence interval
- Observation before action models agents acting on the basis of currently available local information; mixing avoids deterministic biases, so caution is required.

---

## Conclusion

This ODD protocol formalizes the model structure, equations, initialization, results, and experimental protocol necessary to study persistence in games of chance. Implementing this ODD in NetLogo with parameters stored in a documented file such as PARAMETERS.md and automated batch scripts will enable reproducible experiments, sensitivity analyses, and policy testing.

Next steps (the latter will be updated regularly on git but will not be released):

- Implement the NetLogo model following the ODD.
- Create PARAMETERS.md with parameter descriptions and default ranges.
- Add experiments with predefined parameter grids and a run_batch script.
- Provide analysis notebooks that aggregate CSV results and produce key figures (avg-belief, pct-playing, traces and CI bands, distributions, spatial maps).

---

# Author

- Peyanan TRAORE

# Repository

- Agent-Based Model

# License

- GitHub Free