

Turning Savings into a Paycheck: AI-Supported Optimization of Pension Income

Mrugendra Rudrappa Madalagi

Java Technical Architect, USA.

Received: 08-12-2026

Revised: 09-02-2026

Accepted: 09-08-2026

Published: 09-11-2026

ABSTRACT

Turning a lifetime of savings into an income that lasts is one of the hardest problems in personal finance, yet most retirement tools still answer it with a single deterministic projection. This article reviews what the research says about decumulation (withdrawal rules, sequence-of-returns risk, and longevity risk) and where Monte Carlo simulation, stochastic control and reinforcement learning have shown measurable gains over rules of thumb. It then examines German utility model DE 20 2025 107 023 U1, which combines these techniques into one AI-supported tool for optimizing pension income, and proposes a Java and cloud reference architecture for building such a system with the auditability that pension administrators need. It closes with governance requirements and the problems that remain open.

KEYWORDS

Retirement Income, Pension Decumulation, Reinforcement Learning, Monte Carlo Simulation, Explainable AI, Java, Cloud Architecture.

1. INTRODUCTION: THE PAYCHECK THAT STOPS

Only 35 percent of non-retired U.S. adults say their retirement savings plan is on track, a figure unchanged from 2024 and down from 40 percent in 2021 (Federal Reserve Board, 2026). In the same survey, 72 percent named retirement savings as a minor or major financial concern. For most people, the anxiety is not abstract. It is the moment the paycheck stops and a pile of savings has to behave like one.

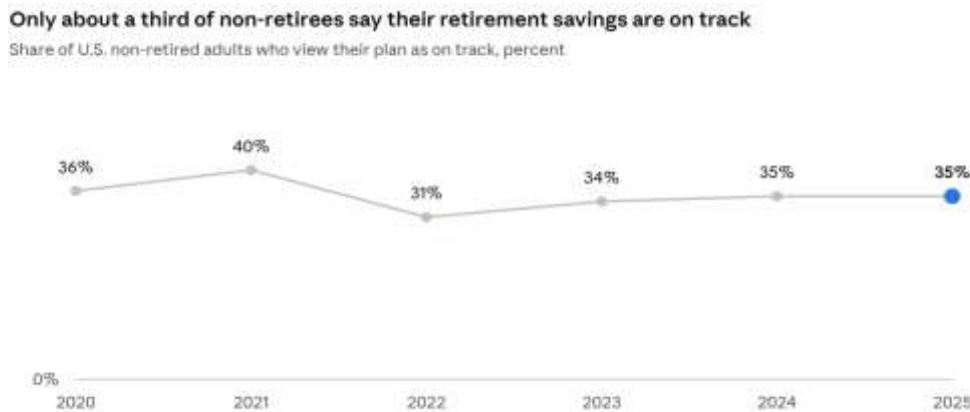


Fig 1 - Percentage of Non-Retired Adults Who Say Their Retirement Savings Are on Track (2020–2025)

Federal Reserve Board, Survey of Household Economics and Decision-making, survey years 2020–2025 The share dipped to 31 percent in 2022 and has crept back since, but it has not returned to its 2021 level.

That moment is harder than it looks. A retiree has to decide how much to spend each year without knowing how long they will live, what markets will return, or what inflation will do to their costs. Economists have known since Yaari (1965) that an uncertain lifespan changes the whole consumption problem. Nobel laureate William Sharpe reportedly called decumulation the "nastiest, hardest problem in finance," a phrase Forsyth (2022) used as the title of his stochastic-control treatment of it.

I have spent my career building enterprise Java systems, and the gap I keep seeing is not a lack of good theory. It is that the theory rarely reaches the person making the decision. Plan portals show one projected balance at age 85, computed once, with fixed returns. This article asks a practical question: what would it take to build a planning system that reflects how retirement actually unfolds, and can AI help without becoming a black box?

2. WHY RULE-OF-THUMB PLANNING FALLS SHORT

The best-known retirement rule came from a historical back-test. Bengen (1994) found that a retiree who withdrew about 4 percent of the starting portfolio, adjusted each year for inflation, would

not have run out of money over any 30-year period in the U.S. record he studied. The rule is simple and memorable, which is why it spread. Its weaknesses are now well documented.

It ignores the individual. A fixed real withdrawal treats a 62-year-old with a defined-benefit pension and a 70-year-old with only a 401(k) the same way. Scott, Sharpe and Watson (2009) showed that pairing a fixed spending rule with a volatile portfolio is internally inconsistent: the retiree either overspends in bad markets or leaves large surpluses in good ones, and pays for that mismatch.

Sequence of returns matters as much as average returns. Two retirees with the same average return can have very different outcomes if one meets a crash in the first five years. Withdrawals made after a loss lock it in. Guyton and Klinger (2006) responded with decision rules that cut or raise spending when the portfolio crosses guardrails, a step toward adaptive policy.

Longevity is a distribution, not a number. Planning to an average life expectancy means roughly half of retirees outlive the plan. Milevsky and Robinson (2000) framed this as a probability of lifetime ruin that depends jointly on spending, portfolio risk and mortality.

People are not optimizers. Low financial literacy is widespread and predicts poorer retirement planning (Lusardi & Mitchell, 2014), and default effects shape behavior more than disclosures do (Benartzi & Thaler, 2013). A tool that produces the right answer but cannot explain it, or cannot nudge the person toward acting on it, fails in practice.

MacDonald et al. (2013) reviewed the drawdown literature and found a persistent gap between academic recommendations and what retirees actually do. That gap is a systems problem as much as a finance problem.

3. WHAT AI AND QUANTITATIVE METHODS ADD

The research points to three layers that together outperform static rules: simulation to describe uncertainty, optimization to choose a policy under it, and learning to keep that policy current.

Simulation. Monte Carlo methods generate thousands of possible futures for returns, inflation and lifespan, and report outcomes as distributions rather than a single balance (Glasserman, 2003). For a retiree, the useful outputs are a probability of meeting an income floor, the expected size of any shortfall, and how long the money lasts in the bad tail.

Optimization. Once outcomes are distributions, the plan becomes a control problem: choose withdrawals and asset allocation each year to balance income against risk. Forsyth (2022) solved this for defined-contribution decumulation, maximizing expected total withdrawals while limiting expected shortfall at the end of the horizon. Compared with a constant withdrawal strategy, the optimal policy delivered noticeably higher average withdrawals for a very small increase in tail risk, and held up on bootstrapped historical data. Forsyth and Li (2025) extend this by comparing tail-risk

measures for running out of money, which matters because the choice of risk metric shapes the policy.

The general objective can be written as a trade-off between income and shortfall risk, with a weight that reflects the retiree's tolerance:

$$\max_{\{q_t, p_t\}} \mathbb{E} \left[\sum_{t=1}^T q_t \right] + \kappa \mathbb{E} \left[\text{ES}_{\alpha}(W_T) \right]$$

Here q is the withdrawal in year t , p is the equity share, W is terminal wealth, ES is expected shortfall at level α , and κ sets how much the retiree cares about the worst outcomes.

Learning. Exact control solutions become intractable once taxes, annuities, Social Security timing and realistic market models are added. Reinforcement teaching (Sutton & Barto, 2018) handles that complexity by learning a policy from simulated experience. Irlam (2020) applied it directly to the retirement portfolio problem. Where the optimal answer was known, the learned policy came within a few percent of it. Where it was not, the learned policy beat common alternatives, while handling income taxes, time-varying yield curves and uncertain lifespans.

Evidence from deployed robo-advice. Field evidence supports the promise and the caveats. D'Acunतो, Prabhala and Rossi (2019) studied a real robo-adviser and found that previously undiversified investors gained diversification and better risk-adjusted returns, while already-diversified investors gained little and traded more. Personalization helps most where people start furthest from a good plan, which describes many pension participants.

Table 1 sets the four approaches side by side. Moving down the table, each method adapts more and handles more real-world detail, but becomes harder to explain without extra tooling.

Title 1: Comparison of Retirement Spending Approaches and Their Key Characteristics

Approach	How spending adapts	Handles taxes, annuities, plan rules	Ease of explanation	Key source
Fixed real withdrawal ("4% rule")	Never; fixed in real terms from day one	No	Very easy	Bengen (1994)
Guardrail decision rules	Cuts or raises spending when the portfolio crosses preset bounds	Limited	Easy	Guyton & Klinger (2006)
Stochastic optimal control	Withdrawals and allocation reset each year from wealth and time	Hard beyond simple market models	Moderate; the policy can be plotted	Forsyth (2022)

left				
Reinforcement learning	Learned policy responds to the full financial state	Yes, including taxes and uncertain lifespan	Hard without explanation tools	Irlam (2020)

4. THE DE 20 2025 107 023 U1 UTILITY MODEL

Utility model DE 20 2025 107 023 U1, "AI-Supported Financial Planning Tool for Personalized Optimization of Pension Income," brings the three layers above into a single claimed system (Kabade et al., 2025). It was filed with the German Patent and Trade Mark Office (DPMA) on 15 November 2025, and its utility model specification was published on 8 January 2026. It is classified under G06Q 40/06 (asset management and financial planning).

The main claim describes four parts working together:

- A data acquisition interface that collects user-specific and wider financial data, such as income, assets, liabilities, spending, demographics and macroeconomic indicators.
- A knowledge modeling unit that finds relationships between financial, demographic and macroeconomic variables.
- A predictive simulation module that produces probabilistic retirement-income paths, by Monte Carlo or an equivalent sampling method (claim 2).
- A decision optimization unit based on reinforcement learning that computes individual investment strategies and withdrawal policies to maximize income adequacy while minimizing financial risk.

The dependent claims add dynamic re-allocation as market data and user behavior change (claim 3), explainable models that give interpretable recommendations with confidence levels (claim 4), and a secure computing core with encrypted data handling and checks against pension and data-protection rules (claim 5). The description scores each simulated scenario on income stability, sufficiency and liquidity, and uses that score as the learning agent's reward.

For an architect, the interesting part is that explainability and compliance sit inside the claims instead of being bolted on later. The literature in Section 3 mostly stops at the optimal policy. A pension administrator also needs to show a participant, an auditor and a regulator why that policy was recommended.

Two points of context matter for readers. First, a German utility model is registered after a formal check only; the DPMA does not examine novelty or inventive step before registration, and validity is tested only if someone challenges it (DPMA, n.d.). Second, the specification describes the reward function and learning agent at a functional level and does not publish validation results. The architecture below is therefore my own implementation proposal consistent with the claims, not a description of a deployed product.

5. A REFERENCE ARCHITECTURE IN JAVA AND THE CLOUD

The design maps each claimed unit to its own service and ties every run to a versioned, append-only record, so any recommendation can be reproduced for an auditor years later.

Four claimed units in one pipeline, with a reproducible record of every run

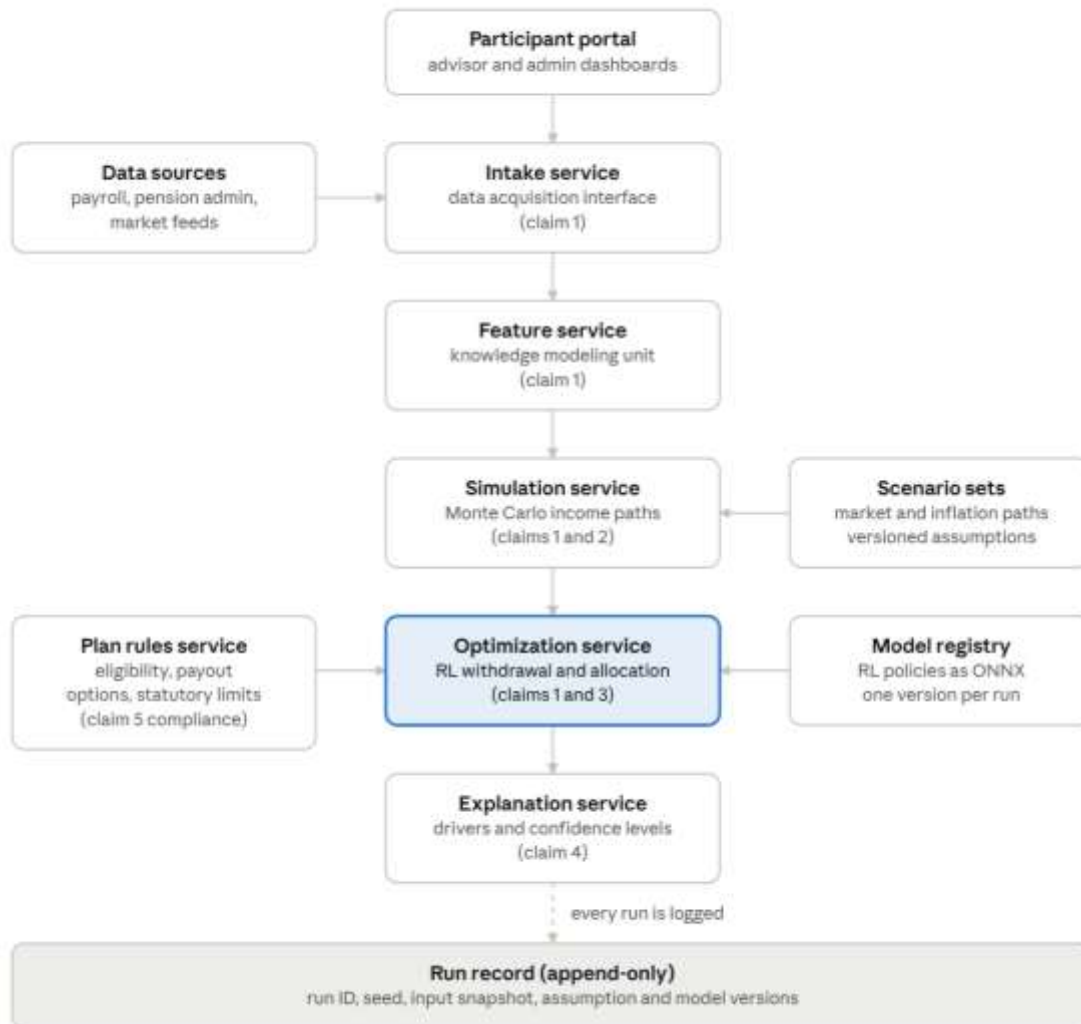


Fig 2 - Integrated Pipeline Architecture for Data Intake, Simulation, Optimization, Explanation, and Reproducible Run Recording

Reference architecture claimed units mapped to services the four claimed units run as a pipeline; plan rules, scenario sets and versioned policies feed in from the sides, and every run lands in one record that can be replayed.

Most pension administration platforms already run on the JVM, with established security, logging and compliance tooling. Building the planning engine in the same ecosystem lowers the cost of getting it approved and keeps it maintainable by the teams who own the core records. The choices below follow from that.

Reproducible simulation. Monte Carlo paths are independent, so they parallelize well on a Fork Join pool or a parallel stream. Reproducibility comes from recording a single root seed per run and deriving each path's generator from it with `SplittableRandom` or the Random Generator API introduced in JDK 17 (JEP 356). The same seed, inputs and assumption versions then produce the same distribution every time. A simplified sketch:

```
public record SimulationRun(String runId, long seed,
                           String assumptionsVersion, String modelVersion) {}

public double probabilityOfMeetingFloor(Participant p, Assumptions a,
                                       SimulationRun run, int paths) {
    var root = new SplittableRandom(run.seed());
    long[] pathSeeds = LongStream.range(0, paths)
        .map(i -> root.split().nextLong()) // sequential: same seeds every run
        .toArray();
    long successes = Arrays.stream(pathSeeds).parallel()
        .filter(s -> simulatePath(p, a, new SplittableRandom(s)).meetsFloorEveryYear())
        .count();
    return (double) successes / paths;
}
```

Seeds are drawn sequentially so they never depend on thread scheduling; only the path simulation runs in parallel. Java 21 virtual threads (JEP 444) are a better fit for the I/O side, such as pulling payroll, pension and market data concurrently during intake, than for this CPU-bound work.

Learned policy, served in Java Reinforcement-learning policies are usually trained offline in Python. Exporting the trained policy to ONNX and serving it through the Deep Java Library or ONNX Runtime's Java API keeps training flexible while production inference stays inside the JVM, under the same deployment and monitoring controls as everything else. Each policy version lives in a model registry, and the run record stores which version produced each recommendation.

Rules as data Pension plan rules (eligibility, payout options, timing limits, and statutory caps) change on their own schedule. Holding them as versioned decision tables in a rules service, rather than in code, lets the optimizer treat them as hard constraints and lets compliance staff review changes without reading Java.

Plain-language explanations, grounded in numbers. The explanation service computes driver attributions, for example SHAP values (Lundberg & Lee, 2017), and sensitivity results such as "what changes if inflation is 2 points higher." A language model, called through Spring AI, can then turn those computed figures into a short explanation for the participant. The model should only rephrase numbers the system has already calculated; it must never produce figures of its own.

Security Participant data is encrypted at rest with keys held in a cloud key management service, identifiers such as Social Security numbers get field-level encryption, and access follows role-based policies with retention limits, matching claim 5.

6. EXPLAINABILITY, GOVERNANCE AND REGULATION

A pension-income recommendation affects someone's standard of living for decades, so the system has to be governed like advice, not like a calculator.

In the United States, automated advisers that are registered investment advisers carry the same fiduciary duties as human ones. The SEC's staff guidance on robot-advisers asks firms to explain clearly how the algorithm works and what it does not consider, to collect enough information to make suitable recommendations, and to test and monitor the code (SEC Division of Investment Management, 2017). Those three expectations map directly onto the explanation service, the intake service and the run record above.

Table 2: Shows How Each Expectation Lands in the Architecture from Section 5

Expectation	Source	Where the system meets it
Explain how the algorithm works and what it leaves out	SEC (2017)	Explanation service and participant disclosures
Collect enough information for a suitable recommendation	SEC (2017)	Intake service with validation and completeness checks
Test and monitor the code	SEC (2017); NIST (2023), "measure"	Back-tests, drift monitoring, versioned run record
Clear accountability for model changes	NIST (2023), "govern"	Approvals in the model registry, with rollback
Record-keeping and transparency	EU AI Act (2024)	Append-only run record

The NIST AI Risk Management Framework gives a practical structure for the rest: govern, map, measure and manage (NIST, 2023). For this kind of system, that means writing down who approves new policy versions, documenting the populations the model was trained and tested on, measuring calibration of the "probability of meeting your income floor" figure against back-tests, and having a rollback path when monitoring finds drift.

In Europe, the AI Act (Regulation (EU) 2024/1689) lists certain financial uses, such as creditworthiness assessment and risk pricing for life and health insurance, as high-risk. Retirement-planning tools are not named in that list, but systems that touch annuity pricing or are deployed by insurers may fall near it, and the Act's transparency and record-keeping expectations are a sensible baseline regardless.

One point in the utility model's description deserves care in practice: it describes the process as able to run without outside financial advisers. Technically that is achievable. From a governance

standpoint, I would recommend keeping a human review step for high-impact actions, such as annuity purchases, lump-sum elections and changes to pension payout options, because those decisions are often irreversible.

7. LIMITATIONS AND OPEN PROBLEMS

The biggest risk in AI-supported retirement planning is false precision: a policy that looks optimal on simulated markets can fail on the one future that actually happens:

- Model risk. Reinforcement-learning policies are only as good as the market generator they learn from. Forsyth (2022) tested on bootstrapped historical data precisely because parametric models can be wrong. Any production system needs out-of-sample and stress-scenario validation, not just in-simulation reward.
- The choice of risk measure. Probability of ruin, expected shortfall and linear shortfall lead to different policies (Forsyth & Li, 2025). Picking one is a value judgment that should be made with the plan sponsor and shown to the participant, not hidden in the reward function.
- Data quality and fairness. Pension records often have gaps for people with interrupted careers, part-time work or caregiving periods. A model trained on clean records may serve them worst. Testing outcomes across these groups belongs in the measurement step of the governance process.
- Behavior. D'Acunto et al. (2019) found that a robo-adviser did not fully remove behavioral biases. A correct recommendation still has to be understood and acted on, which is a design problem as much as a modeling one.
- Independent evidence. Published results for integrated, end-to-end systems like the one claimed in DE 20 2025 107 023 U1 are still scarce. Open benchmarks, with shared scenario sets and participant profiles, would let researchers and vendors compare approaches on equal terms.

8. CONCLUSION

On the finance side, the evidence points one way: adaptive, simulation-based withdrawal policies do better than fixed rules, and reinforcement learning copes with real-world detail that exact methods struggle with. What has been missing is engineering that brings those results to the people making the decision, with explanations they can follow and records that regulators can check.

Utility model DE 20 2025 107 023 U1 sets out one blueprint for that integration: data intake, modeling, probabilistic simulation and reinforcement-learning optimization, with explainability and compliance written into the claims. The Java and cloud architecture proposed here shows that the blueprint can be built with mainstream enterprise tools, reproducibly and securely.

The next step is evidence. Pension administrators, researchers and technology teams should pilot these systems against transparent benchmarks and publish what they find. A retiree deserves to know not just what the plan says, but how confident the plan is and why.

REFERENCES

- [1] Benartzi, S., & Thaler, R. H. (2013). Behavioral economics and the retirement savings crisis. *Science*, 339(6124), 1152–1153.
- [2] Bengen, W. P. (1994). Determining withdrawal rates using historical data. *Journal of Financial Planning*, 7(4), 171–180.
- [3] D'Acunto, F., Prabhala, N., & Rossi, A. G. (2019). The promises and pitfalls of robo-advising. *The Review of Financial Studies*, 32(5), 1983–2020. <https://doi.org/10.1093/rfs/hhz014>
- [4] Deutsches Patent- und Markenamt [DPMA]. (n.d.). *Utility model protection*. https://dpma.de/english/utility_models/utility_model_protection/index.html
- [5] European Parliament & Council of the European Union. (2024). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union*.
- [6] Federal Reserve Board. (2026). *Economic well-being of U.S. households in 2025: Savings and investments*. <https://www.federalreserve.gov/publications/2026-economic-well-being-of-us-households-in-2025-savings-investments.htm>
- [7] Forsyth, P. A. (2022). A stochastic control approach to defined contribution plan decumulation: "The nastiest, hardest problem in finance." *North American Actuarial Journal*, 26(2), 227–251. <https://doi.org/10.1080/10920277.2021.1878043>
- [8] Forsyth, P. A., & Li, Y. (2025). *Risk measures for DC pension plan decumulation* (arXiv:2502.16364). <https://arxiv.org/abs/2502.16364>
- [9] Glasserman, P. (2003). *Monte Carlo methods in financial engineering*. Springer.
- [10] Guyton, J. T., & Klinger, W. J. (2006). Decision rules and maximum initial withdrawal rates. *Journal of Financial Planning*, 19(3), 48–58.
- [11] Irlam, G. (2020). Machine learning for retirement planning. *The Journal of Retirement*, 8(1), 32–39. <https://doi.org/10.3905/jor.2020.1.067>
- [12] Kagalkar, A., Kabade, S., Chaudhari, B. B., Sharma, A., & Maurya, S. (2025). Artificial intelligence-supported financial planning tool for personalized optimization of pension income (German Utility Model No. DE202025107023U1). Deutsches Patent- und Markenamt. <https://patents.google.com/patent/DE202025107023U1/en>
- [13] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30* (NeurIPS 2017).
- [14] Lusardi, A., & Mitchell, O. S. (2014). The economic importance of financial literacy: Theory and evidence. *Journal of Economic Literature*, 52(1), 5–44.
- [15] MacDonald, B.-J., Jones, B., Morrison, R. J., Brown, R. L., & Hardy, M. (2013). Research and reality: A literature review on drawing down retirement financial savings. *North American Actuarial Journal*, 17(3), 181–215.
- [16] Milevsky, M. A., & Robinson, C. (2000). Self-annuitization and ruin in retirement. *North American Actuarial Journal*, 4(4), 112–124.
- [17] National Institute of Standards and Technology [NIST]. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)* (NIST AI 100-1). <https://doi.org/10.6028/NIST.AI.100-1>
- [18] Scott, J. S., Sharpe, W. F., & Watson, J. G. (2009). The 4% rule – at what price? *Journal of Investment Management*, 7(3), 31–48.
- [19] Securities and Exchange Commission [SEC], Division of Investment Management. (2017). *Robo-advisers* (IM Guidance Update No. 2017-02).
- [20] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd Ed.). MIT Press.
- [21] Yaari, M. E. (1965). Uncertain lifetime, life insurance, and the theory of the consumer. *The Review of Economic Studies*, 32(2), 137–150.