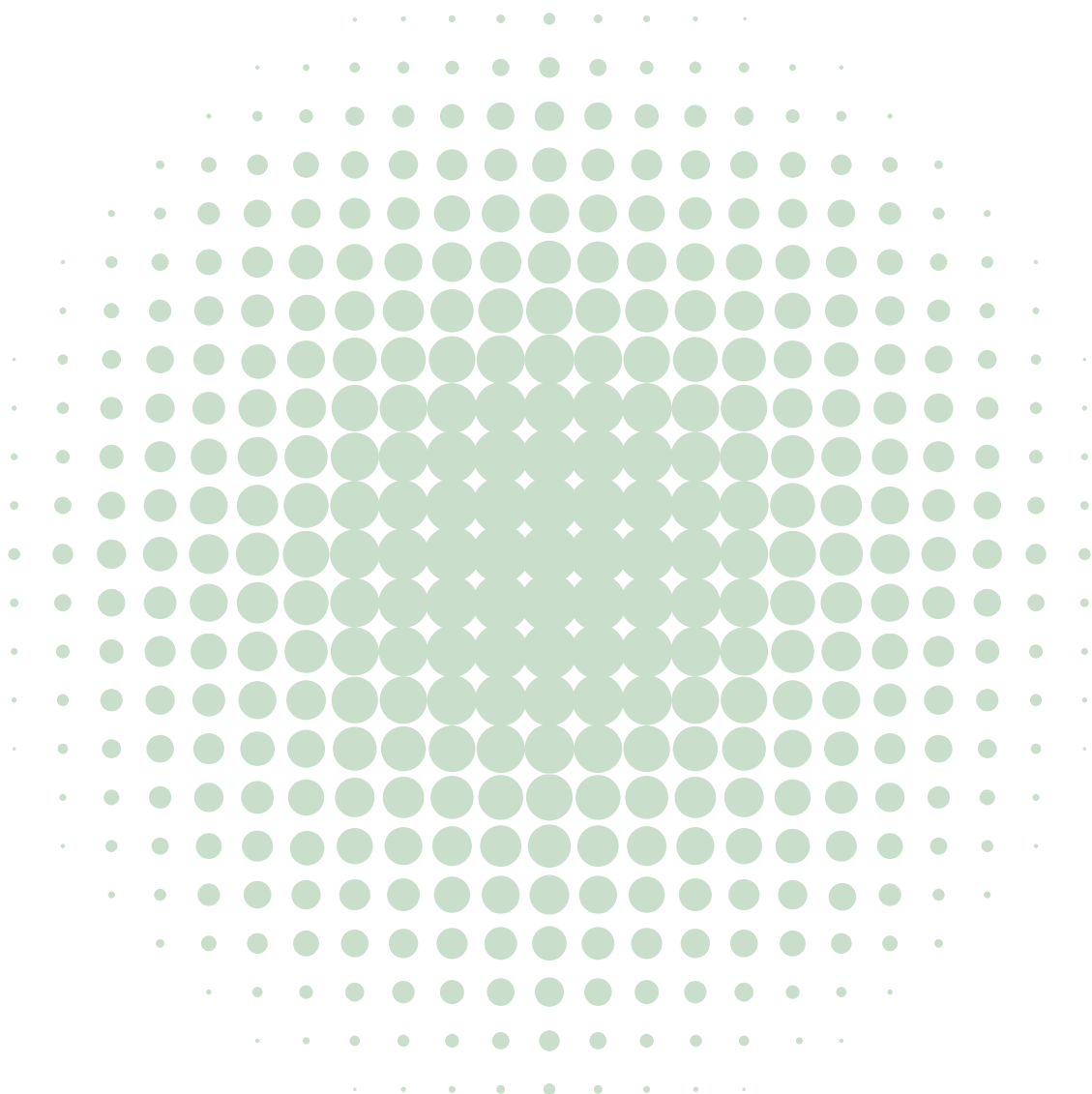




Externe Expertise des revidierten AHV-Gesamtmodells



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement des Innern EDI
Département fédéral de l'intérieur DFI
Bundesamt für Sozialversicherungen BSV
Office fédéral des assurances sociales OFAS

Das Bundesamt für Sozialversicherungen veröffentlicht konzeptionelle Arbeiten sowie Forschungs- und Evaluationsergebnisse zu aktuellen Themen im Bereich der Sozialen Sicherheit, die damit einem breiteren Publikum zugänglich gemacht und zur Diskussion gestellt werden sollen. Die präsentierten Folgerungen und Empfehlungen geben nicht notwendigerweise die Meinung des Bundesamtes für Sozialversicherungen wieder.

Autorinnen/Autoren

Florentin Krämer, Lukas Mergele
BSS Volkswirtschaftliche Beratung AG
Aeschengraben 9, CH-4051 Basel
+41 (0)61 262 05 55, contact@bss-basel.ch
www.bss-basel.ch

Auskünfte

Bundesamt für Sozialversicherungen
Effingerstrasse 20, CH-3003 Bern

Geschäftsfeld Mathematik, Analysen und Statistik
Bereich Mathematik
Jörg Kalbfuss
+41 (0)58 463 25 75, joerg.kalbfuss@bsv.admin.ch

Copyright

Bundesamt für Sozialversicherungen, CH-3003 Bern
Auszugsweiser Abdruck – ausser für kommerzielle Nutzung –
unter Quellenangabe und Zustellung eines Belegexemplares
an das Bundesamt für Sozialversicherungen gestattet.

Publikationsdatum und Auflage

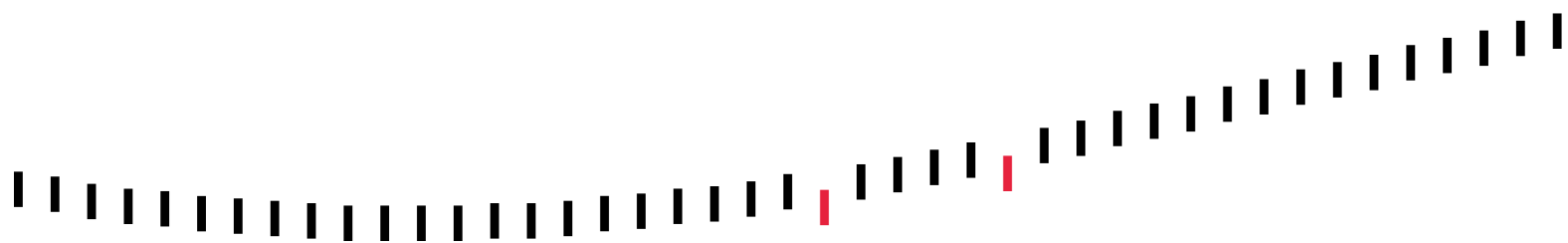
Juli 2026, 1. Auflage

Finalisierter Schlussbericht

Juli 2026

Schlussbericht

Externe Expertise des revidierten AHV-Ge- samtmodells

Basel , 12.06.2026

Impressum

Externe Expertise des revidierten AHV-Gesamtmodells

12.06.2026

Auftraggeberin: Bundesamt für Sozialversicherungen BSV

Autoren: Florentin Krämer, Lukas Mergele

Verantwortlich seitens Auftraggeberin: Jörg Kalbfuss

Projektleitung seitens Auftragnehmerin: Florentin Krämer

Projektbearbeitung: Florentin Krämer, Lukas Mergele

BSS Volkswirtschaftliche Beratung AG

Aeschengraben 9

CH-4051 Basel

T +41 61 262 05 55

contact@bss-basel.ch

www.bss-basel.ch

© 2026 BSS Volkswirtschaftliche Beratung AG

Inhalt

Zusammenfassung.....	iv
Résumé.....	v
Riassunto	vi
Summary	vii
1 Ausgangslage und Ziele	1
2 Vorgehen	1
2.1 Prüfansatz.....	1
2.2 Manuelle Code-Prüfung	2
2.3 KI-gestützter Review	3
2.4 Synthese.....	3
3 Ergebnisse der Prüfung.....	3
3.1 Paket «dinput»	3
3.2 Paket «delfin»	4
3.3 Paket «dinfra»	5
3.4 Modul «Referenzaltererhöhung»	6
3.5 Gesamtarchitektur.....	7
4 Fazit.....	8
A Prüfschema	9
B Relevante Einzelbefunde	12
B.1 Paket «dinput»	13
B.2 Paket «delfin»	16
B.3 Paket «dinfra»	20
B.4 Modul «Referenzaltererhöhung»	22
C Anpassungsvorschläge	24
C.1 Validierungsschicht	24
C.2 Umgang mit Warnungen und Mitteilungen.....	25
C.3 Zentrales Codebuch.....	25

Tabellen

Tabelle 1: Auswahl relevanter Einzelbefunde für das Paket dinput.....	13
Tabelle 2: Auswahl relevanter Einzelbefunde für das Paket delfin	16
Tabelle 3: Auswahl relevanter Einzelbefunde für das Paket dinfra.....	20
Tabelle 4: Auswahl relevanter Einzelbefunde für das Modul Referenzaltererhöhung	22

Zusammenfassung

Das Bundesamt für Sozialversicherungen (BSV) hat sein Modell zur langfristigen Projektion der AHV-Finzen grundlegend überarbeitet. Das revidierte Modell «ATLAS» soll im Sommer 2026 erstmals für die Erstellung der publizierten Finanzperspektiven der AHV eingesetzt werden. Das BSV hat uns beauftragt, dieses Modell als unabhängige Stelle zu prüfen. Im Zentrum der Expertise steht das Umlage-Szenario «Referenz», das die Entwicklung der AHV-Finzen unter der heute geltenden Gesetzeslage abbildet.

Zwei erfahrene Fachpersonen haben das Modell unabhängig voneinander geprüft, ergänzt durch ein KI-gestütztes Prüfverfahren. Geprüft wurde entlang des gesamten Rechenwegs zur Erstellung der Finanzperspektiven unter der geltenden Ordnung – von den Eingangsdaten über die Zwischenberechnungen bis zur Fortschreibung des Fondsvermögens. Die Überführung der Berechnungen in den finalen Output (sprich die Darstellung der Projektionen in Form von publizierbaren Excel-Dateien) war nicht Gegenstand der Expertise. Die Prüfung stützt sich auf den bereitgestellten Programmcode und die gelieferten Daten. Aus Datenschutzgründen lagen nicht alle Rohdaten vor; wo dies der Fall war, wurden uns voraggregierte Daten übermittelt, welche die benötigte Granularität für die weiteren Schritte der AHV-Pipeline aufweisen.

Unser Gesamturteil ist positiv. Das Modell setzt die gesetzlichen Vorgaben der AHV korrekt um und liefert plausible Ergebnisse. Sein Aufbau ist klar gegliedert und es ist durch die konsequente Modularisierung gut wartbar; dies macht das Modell robust gegenüber künftigen Anpassungen. Wir sehen keine grundlegenden Mängel, die einer Erstanwendung im Jahr 2026 entgegenstehen.

Die festgestellten Punkte sind klar abgrenzbar, in ihrer Wirkung auf das Gesamtergebnis gering und mit überschaubarem Aufwand zu beheben. Perspektivisch empfehlen wir vor allem eine systematische Eingangskontrolle der Daten, damit künftige Änderungen an den Datenquellen früh erkannt werden. Hervorzuheben ist, dass das BSV Stand heute (12. Juni 2026) gemäss eigener Aussage die in Anhang B aufgeführten relevanten Befunde umgesetzt hat. Die verbleibenden Empfehlungen sind für die Erstanwendung nicht kritisch.

Résumé

L'Office fédéral des assurances sociales (OFAS) a entièrement remanié son modèle de projections à long terme des finances de l'AVS. Son nouveau modèle « ATLAS » sera utilisé pour la première fois pour l'élaboration des perspectives financières publiées de l'AVS en été 2026. L'OFAS a chargé le cabinet de conseil BSS, en tant qu'organisme indépendant, d'évaluer ce modèle. Au cœur de l'expertise se trouve le scénario de répartition de référence, qui reflète l'évolution des finances de l'AVS dans le cadre de la législation en vigueur.

Deux spécialistes expérimentés ont évalué ce modèle indépendamment l'un de l'autre, cette approche étant complétée par une procédure de contrôle basé sur l'IA. L'évaluation a porté sur l'ensemble du processus de calcul nécessaire à l'élaboration des perspectives financières dans le régime en vigueur, depuis les données d'entrée jusqu'à la mise à jour de la fortune du fonds en passant par les calculs intermédiaires. L'expertise ne portait pas sur l'intégration des calculs dans les résultats finaux (c'est-à-dire sur la présentation des projections sous la forme de fichiers Excel publiables). L'évaluation s'appuie sur le code de programmation fourni et les données transmises. Afin de respecter la protection des données, le cabinet de conseil n'a pas disposé de l'ensemble des données brutes ; dans le cas contraire, les données préagrégées qui nous ont été transmises présentaient le niveau de granularité requis pour les étapes suivantes des projections AVS.

Notre jugement global est positif. Le modèle met correctement en œuvre les exigences légales de l'AVS et livre des résultats plausibles. Sa composition est clairement structurée et sa modularisation systématique facilite son entretien, ce qui rend le modèle suffisamment robuste pour résister à de futures adaptations. Aucune lacune fondamentale qui empêcherait une première application de ce modèle en 2026 n'a été relevée.

Les points constatés sont clairement identifiables, leur incidence sur le résultat global est minime et ils peuvent être corrigés avec des moyens raisonnables. Pour la suite, le cabinet recommande avant tout un contrôle systématique à la réception des données entrantes afin que les modifications à venir des sources de données soient rapidement détectées. Il faut souligner qu'à ce jour (12 juin 2026), l'OFAS a déclaré avoir mis en œuvre les conclusions pertinentes figurant dans l'annexe B. Les recommandations restantes ne sont pas pertinentes pour la première application.

Riassunto

L'Ufficio federale delle assicurazioni sociali (UFAS) ha rielaborato in modo approfondito il proprio modello per le proiezioni a lungo termine della situazione finanziaria dell'AVS. Il modello riveduto, denominato «ATLAS», sarà impiegato per la prima volta nell'estate del 2026 per l'allestimento delle prospettive finanziarie dell'AVS destinate alla pubblicazione. L'UFAS ci ha incaricati di verificare questo modello in qualità di ente indipendente. La nostra verifica è incentrata sullo scenario di ripartizione «riferimento», che illustra l'andamento delle finanze dell'AVS in base all'ordinamento vigente.

Due specialisti del settore di grande esperienza hanno esaminato il modello in modo indipendente l'uno dall'altro, avvalendosi a titolo complementare di una procedura di verifica basata sull'intelligenza artificiale. La verifica è stata effettuata lungo l'intero percorso di calcolo utilizzato per l'allestimento delle prospettive finanziarie secondo l'ordinamento vigente, dai dati di partenza ai calcoli intermedi fino alla proiezione del patrimonio del fondo. La trasposizione dei calcoli nel risultato finale (ovvero la presentazione delle proiezioni sotto forma di file Excel pubblicabili) non era oggetto di verifica. La verifica si è basata sul codice di programma e sui dati forniti. Per motivi di protezione dei dati, non ci sono stati messi a disposizione tutti i dati grezzi; nei casi in cui ciò è avvenuto, ci sono stati trasmessi dati preaggregati che presentavano il livello di granularità necessario per le fasi successive del processo dell'AVS.

Il nostro giudizio complessivo è positivo. Il modello applica correttamente le prescrizioni della legislazione AVS e fornisce risultati plausibili. È strutturato in modo chiaro e, grazie alla sua modularità coerente, è facile da gestire; il modello è quindi sufficientemente robusto per far fronte a future modifiche. Non riscontriamo carenze sostanziali che possano ostacolare la sua prima applicazione nel 2026.

I punti constatati sono chiaramente circoscritti, hanno un impatto marginale sul risultato complessivo e possono essere risolti con uno sforzo contenuto. In prospettiva futura, raccomandiamo soprattutto un controllo sistematico dei dati di partenza, affinché eventuali future modifiche alle fonti dei dati possano essere individuate tempestivamente. Va sottolineato che al momento (12.6.2026), secondo le sue dichiarazioni, l'UFAS ha attuato le raccomandazioni riportate nell'allegato B. Quelle rimanenti non sono rilevanti ai fini della prima applicazione del modello.

Summary

The Federal Social Insurance Office (FSIO) has overhauled ATLAS, the model it uses to forecast the long-term financial trajectory of the Old-Age and Survivors' Insurance (OASI) scheme. According to current plans, the new iteration of the model will be used for the first time in summer 2026 to establish the financial perspectives of the OASI scheme, which the FSIO regularly publishes. As an independent body, we were commissioned by the FSIO to review the latest version of ATLAS. Our work focused on the 'reference' pay-as-you-go scenario, which maps the evolution of OASI finances under the current legal framework.

The model was audited separately and independently by two experienced experts and underwent AI-supported testing as well. These reviews covered every step of the process used to generate a projection of the OASI's finances under the current regulatory framework: from the input data and intermediate calculations through to the extrapolation of OASI compensation fund assets. The expert report did not examine the transformation of the calculations into the final output (i.e. the presentation of the financial projections in the form of publishable Excel files). The testing was based on the ATLAS programme code and the data provided. Not all raw data were made available to us for data protection reasons. However, we were provided with pre-aggregated data that met the level of granularity needed for retracing the subsequent steps in the OASI projections.

Our overall assessment is positive. The model correctly implements the statutory requirements of the OASI scheme and delivers plausible results. It is clearly structured and easy to maintain thanks to its consistent modularisation. Consequently, it will be robust to future adjustments. We have not observed any fundamental shortcomings that would prevent its application starting in 2026.

The shortcomings we did identify can be clearly delineated. They have little impact on the overall outcome, and no major effort would be needed to rectify them. Going forward, our primary recommendation is the creation of a systematic data validation process to ensure the early detection of changes to data sources in the future. We would also like to point out that, as of today (12 June 2026), the FSIO has stated that it has implemented the relevant recommendations listed in Appendix B. The remaining recommendations are not critical for the launch of the revised ATLAS model.

1 Ausgangslage und Ziele

Ausgangslage

Das Bundesamt für Sozialversicherungen (BSV) hat sein Projektionsmodell für die Finanzen der Alters- und Hinterlassenenversicherung (AHV) grundlegend überarbeitet. Das revidierte Modell mit dem Arbeitstitel «Analyse der Tragfähigkeit der Langfristfinanzierung aller Sozialversicherungen» (ATLAS) soll im Sommer 2026 erstmals zur Erstellung der publizierten Finanzperspektiven eingesetzt werden. ATLAS ist ein in der Programmiersprache R geschriebenes Modellierungssystem. Der uns übermittelte Code-Ausschnitt besteht aus drei Paketen (dinfra, dinput, delfin), die in einer definierten Reihenfolge zusammenwirken. Der zu prüfende Code umfasst über die drei Pakete rund 5'000 Zeilen in über 75 Skripten.¹

Ziele

Zentraler Gegenstand der Evaluation ist das Umlage-Szenario «Referenz», das den AHV-Finanzhaushalt unter der geltenden gesetzlichen Ordnung modelliert. Die Prüfung klärt, ob der R-Code dieses Szenario technisch korrekt und plausibel implementiert, und beurteilt, ob das Modell einen angemessenen Kompromiss zwischen Detailtreue zum Rentensystem und Pflegbarkeit der Implementierung erzielt. Teil der Prüfung ist auch die konzeptionelle Abbildung einer Erhöhung des Referenzalters, sowohl auf der Ausgaben- wie auch auf der Einnahmenseite. Nicht Gegenstand der Prüfung sind die Unsicherheitsquantifizierung, die Massnahmenrechnungen sowie das Paket doutput, welches die Ergebnisse in die finalen Excel-Dateien überträgt.

Dieser Bericht

Der vorliegende Bericht dokumentiert den finalen Stand der Prüfung per 12. Juni 2026. Er umfasst das Vorgehen, die zentralen Befunde aus den drei geprüften Paketen und dem Modul «Referenzaltererhöhung», sowie ein Gesamturteil. Das verwendete Prüfschema, relevante Einzelbefunde, sowie architektonische Anpassungsvorschläge befinden sich im Anhang.

2 Vorgehen

2.1 Prüfansatz

Die Prüfung basierte auf einer manuellen Prüfung, ergänzt durch einen KI-gestützten Review («Vier-Augen-plus-KI-Prinzip»). Zwei Prüfpersonen prüften den gesamten Code unabhängig voneinander und dokumentierten ihre Befunde separat. Zusätzlich zum manuellen Review wurde ein KI-Sprachmodell als ergänzendes Prüfinstrument eingesetzt. Im Folgenden werden die einzelnen Bestandteile erläutert.

¹ Das Paketensemble, das unter der Bezeichnung «delfinverse» firmiert, enthält noch weitere Pakete und Funktionen – darunter das Paket «doutput» zur Produktion von Excel-Tabellen und «dmeasures» zur Modellierung von politischen Massnahmen und Funktionen zur Erstellung von Szenario-Grafiken.

2.2 Manuelle Code-Prüfung

Zwei Prüfpersonen durchliefen den gesamten R-Code unabhängig voneinander. Die Prüfung erfolgte modulweise entlang des Datenflusses im Modell, also von den Inputdaten über die Zwischenberechnungen bis zur Finanzhaushalt-Aggregation. Diese Reihenfolge stellt sicher, dass kritische Punkte in vorgelagerten Modulen erkannt werden, bevor nachgelagerte Module geprüft werden, die auf deren Ergebnissen aufbauen.

Prüfreihenfolge entlang der Pipeline

Die Reihenfolge der Modulprüfung orientierte sich am Datenfluss:

- Die Prüfung beginnt beim Einlesen der Inputdaten (`mod_read_data` und `mod_input_*`-Funktionen in `dinput/dinfra`), wo geprüft wurde, ob die Originaldaten korrekt in die gemeinsame Datenstruktur überführt werden.
- Darauf folgen die vorbereitenden Berechnungen in `delfin`, welche die demografischen und ökonomischen Grundlagen über den Projektionshorizont fortschreiben. Die Hauptberechnung umfasst auf der Ausgabenseite die Rentensummenprojektion sowie die 13. AHV-Rente, Hilflosenentschädigungen und übrige Ausgaben; auf der Einnahmenseite die Lohnbeiträge, MWST-Erträge, den Bundesbeitrag, die Spielbankenabgabe und übrige Einnahmen. Besonderes Augenmerk galt anschliessend der Bilanz, wo das Fondsvermögen rekursiv fortgeschrieben wird und sich fehlerhafte Berechnungen über den gesamten Projektionshorizont kumulieren können; hier wurden auch der Kapitalertrag, die IV-Schuld-Logik und die einmalige IPSAS-Korrektur 2024 geprüft. Den Abschluss bildete die Nachbearbeitung mit der Diskontierung und der Kennzahlenberechnung, wobei insbesondere die Konsistenz der Referenzgrössen mit den vorherigen Phasen sichergestellt wurde.
- Die Pakete zur Überführung der Berechnungen in den finalen Output (`doutput`) sowie zur Darstellung der Auswirkungen verschiedener politischer Massnahmen (`dmeasures`) waren nicht Bestandteil der Prüfung.

Prüfkriterien

Für jedes Paket/Modul wurden im Wesentlichen fünf Aspekte untersucht:

1. *Datenkonsistenz*: Werden die richtigen Inputdaten verwendet und stimmen die Dimensionen (z. B. Alter, Geschlecht) überall überein? Sind die Einheiten konsistent, ist der Jahresbereich lückenlos?
2. *Logische Korrektheit*: Stimmt der Code mit den rechtlichen Regelungen überein? Werden Formeln, Indizes und Gewichtungen korrekt umgesetzt?
3. *Plausibilität der Zwischenergebnisse*: Liefern die Module sowohl für historische Jahre (Abgleich mit publizierten BSV-Daten) als auch für Projektionsjahre plausible Werte ohne unerwünschte Sprünge oder unrealistische Trends?
4. *Code-Qualität*: Typ-Sicherheit, Vermeidung von Hardcoding und saubere Behandlung von Randfällen wie fehlenden Werten. Auch: Ist der Funktionsumfang angemessen und gegenüber anderen Funktionen klar abgegrenzt?
5. *Dokumentation*: Sind die erwarteten Inputs und Outputs der Funktionen gut dokumentiert? Werden komplexe Berechnungen durch Code-Kommentare verständlich erläutert?

Das detaillierte Prüfschema auf Ebene Skript ist in Anhang A beigelegt.

2.3 KI-gestützter Review

Parallel zum manuellen Review wurden die einzelnen R-Skripte sowie relevantes Kontextmaterial dem KI-Modell Claude Opus 4.7 (in einzelnen Fällen auch Sonnet 4.6) der Firma Anthropic vorgelegt. Das Kontextmaterial umfasste Literatur zur guten Praxis der R-Programmierung, die einschlägigen Gesetzestexte (AHVG, AHVV) sowie die zugehörigen Erläuterungen. Auf dieser Grundlage wurde das KI-Modell mit den jeweiligen Untersuchungsgegenständen des Auftrags abgefragt. Der KI-gestützte Review diente als *ergänzende* Perspektive, nicht als Ersatz für die manuelle Prüfung. Er ermöglichte es, systematisch nach Mustern zu suchen, die bei einem manuellen Durchgang leicht übersehen werden können (z. B. inkonsistente Namenskonventionen, potenziell fehleranfällige Konstrukte oder Abweichungen von den im Code selbst angelegten Konventionen). Die endgültige Bewertung aller Befunde oblag stets den Prüfpersonen.

2.4 Synthese

Nach Abschluss der unabhängigen Prüfungen für ein Skript wurden die Befunde zusammengeführt. Punkte, bei denen die Prüfpersonen zu unterschiedlichen Bewertungen gelangten, wurden konsentiert. Bei verbleibenden Unklarheiten wurden Rückfragen an die entsprechenden BSV-Kontaktpersonen gerichtet. Ergebnis dieser Phase war ein konsolidiertes Prüfprotokoll, das die Grundlage für den vorliegenden Bericht bildet.

3 Ergebnisse der Prüfung

Wir stellen nachfolgend die Ergebnisse unserer Prüfung für die drei Pakete und das zusätzliche Modul «Referenzaltererhöhung» vor, wobei wir einem einheitlichen Abschnittsaufbau folgen. Zunächst erläutern wir kurz die Rolle, die das Paket innerhalb der Code-Pipeline einnimmt. Anschliessend geben wir eine zusammenfassende Einschätzung ab, ohne auf Einzelbefunde detailliert einzugehen – diese Rolle kommt den tabellarischen Übersichten in Anhang B zu. Dort stellen wir auch dar, welche Konsequenzen das BSV nach eigenen Angaben aus den Befunden gezogen hat.

3.1 Paket «dinput»

3.1.1 Rolle des Pakets

Das Paket dinput verantwortet das Einlesen der Originaldaten aus den unterschiedlichen Quellen (z. B. Schweizerische Arbeitskräfteerhebung des Bundesamts für Statistik) und deren Überführung in die gemeinsame Datenstruktur des Modells.

3.1.2 Übergeordnete Einschätzung

Das Paket erfüllt seine Aufgabe – das Einlesen heterogener Quelldaten und deren Überführung in die gemeinsame Modellstruktur – zuverlässig und nachvollziehbar. Pro Datenquelle existiert eine klar abgegrenzte `mod_input_*`-Funktion mit konsistentem Rückgabetyt und durchgängiger Roxygen-Dokumentation; die Originaldaten werden nicht in-place verändert. Die Prüfung der Einzelskripte ergab keine fachlichen Berechnungsfehler. Wo wir Auffälligkeiten (vgl. Anhang B.1) feststellten, betrafen diese überwiegend die Rohdaten selbst (etwa unplausibel hochaltrige Personen in den Registerauszügen) und nicht deren Verarbeitung; ihre Wirkung auf die Aggregate ist durchweg gering.

Aktuell werden die Originaldaten weitgehend ohne explizite Validierung in die Modelldatenstruktur überführt. Soweit uns eine Beurteilung möglich ist, ergeben sich dadurch, bezogen auf den aktuellen Datenstand, im weiteren Verlauf keine Probleme. Kritisch erscheint uns aber die Frage, ob die Einleselogik robust gegenüber Änderungen am Aufbau der Rohdaten ist. Ohne einen anschließenden Abgleich zwischen erwarteten Werten und tatsächlichen Werten können strukturelle oder inhaltliche Anomalien in den Inputdaten unbemerkt in nachgelagerte Berechnungen propagieren und sich über den Projektionshorizont kumulieren. Wir präzisieren diesen Punkt in Abschnitt 3.5.

3.2 Paket «delfin»

3.2.1 Rolle des Pakets

Das Paket `delfin` enthält die eigentliche Finanzhaushaltsrechnung der AHV. Auf Basis der von `dinput` bereitgestellten Daten schreibt es die demografischen und ökonomischen Grundlagen über den Projektionshorizont fort und berechnet daraus die Komponenten der Ausgaben- und Einnahmenseite (insbesondere Rentensummen, 13. AHV-Rente, Hilflosenentschädigungen, Lohnbeiträge, MWST-Erträge, Bundesbeitrag und Spielbankenabgabe). Anschliessend führt `delfin` diese Komponenten in der Bilanz zusammen, schreibt das Fondsvermögen rekursiv fort und ermittelt Kapitalertrag, IV-Schuld sowie die zentralen Kennzahlen des Umlageergebnisses.

3.2.2 Übergeordnete Einschätzung

`delfin` bildet den fachlichen Kern des Modells und setzt die institutionellen Rahmenbedingungen der AHV-Finanzhaushaltsrechnung insgesamt korrekt um. Die Komponenten der Ausgaben- und Einnahmenseite, die rekursive Fondsfortschreibung und die Kennzahlenberechnung sind sauber strukturiert, defensiv programmiert und berücksichtigen die geltenden gesetzlichen Vorgaben (AHVG/AHV, STAF, AHV 21).² Die identifizierten Befunde (vgl. Anhang B.2) sind lokal eingrenzbar und in ihrer numerischen Wirkung – soweit überhaupt vorhanden – klein. Aus unserer Sicht besteht der grösste Hebel für eine Verbesserung der Code-Qualität in diesem Paket darin,

² Eine defensive Programmierung impliziert, dass Code so entworfen wird, dass er auch unter unerwarteten Bedingungen oder fehlerhaften Eingaben stabil bleibt und Schäden verhindert.

Pfadabhängigkeiten explizit zu adressieren und implizite Reihenfolgeannahmen zu härten sowie konsequent auf eine einheitliche Typisierung der Variablen zu achten.³

Inhaltlich merken wir an, dass die Effekte der AHV 21-Reform gezwungenermassen auf vielen Annahmen über das Bezugsverhalten beruhen: Wann gehen Kohorten mit vorteilhaften Vorbezugssätzen in Rente? Wie stark reagiert das Renteneintrittsalter auf eine Erhöhung des Referenzalters? Das BSV weist diese Annahmen sowohl im Code als auch im begleitenden ATLAS-Handbuch transparent aus, was wir positiv würdigen. Unsere Empfehlung besteht darin, das tatsächliche Bezugsverhalten im Laufe der Übergangsphase eng zu monitoren und die Modellannahmen ggf. nachzujustieren.

3.3 Paket «dinfra»

3.3.1 Rolle des Pakets

Das Paket dinfra stellt die technische Infrastruktur des Modells bereit. Es definiert die gemeinsamen Datenstrukturen (insbesondere die Liste `tl_inp` und das Parameter-Tibble `PARAM_GLOBAL`), bündelt zentrale Hilfsfunktionen (z. B. Diskontierung, Zeitreihenfortschreibung) und stellt einheitliche Schnittstellen für das Einlesen und Verarbeiten der Daten bereit. dinfra enthält keine fachlichen Berechnungen, sondern bildet ein technisches Fundament, auf dem dinput, delfin und doutput aufbauen.

3.3.2 Übergeordnete Einschätzung

dinfra erfüllt seine Rolle als technisches Fundament überzeugend: Die Trennung von technischer Infrastruktur und fachlicher Berechnung ist konsequent durchgehalten, die zentralen Datenstrukturen (`tl_inp`, `PARAM_GLOBAL`) und Hilfsfunktionen sind klar verantwortet und gut dokumentiert. Wesentliche Fehler haben wir in diesem Paket keine festgestellt.

Verbesserungspotenzial sehen wir aus übergeordneter Perspektive an zwei Stellen (vgl. Anhang B.3 für relevante Einzelbefunde). Erstens sollten die generischen Hilfsfunktionen robuster gegen unerwartete Eingaben gemacht werden: `diskontierung()` etwa multipliziert derzeit alle Nicht-Jahr-Spalten und sollte über `where(is.numeric)` auf numerische Spalten beschränkt oder im Roxygen-Header klar auf den zulässigen Spalten-Scope festgelegt werden. Zweitens ist die in Abschnitt 3.5 beschriebene pauschale Unterdrückung von Warnungen und Mitteilungen (`suppressMessages/suppressWarnings`, `showWarnings = FALSE`) hier an mehreren Stellen verankert und sollte gezielter gestaltet werden.

³ Auch hier der Hinweis, dass uns aktuell keine Stellen im Code bekannt sind, an denen der Verzicht auf das Erzwingen der Sortierreihenfolge negative Auswirkungen hat. Die Empfehlung ist daher unter dem Gesichtspunkt zu lesen, den Code robust gegenüber künftigen Anpassungen zu machen.

3.4 Modul «Referenzaltererhöhung»

3.4.1 Rolle des Moduls

Das Modul berechnet die finanziellen Folgen einer (hypothetischen) Referenzaltererhöhung gegenüber einem bereits gerechneten AHV-Baseline-Lauf. Institutionell werden drei Zahlungswirkungen abgebildet: Eingesparte Altersrentenzahlungen durch späteren Renteneintritt, zusätzliche Einnahmen durch längere Erwerbsphase, niedrigerer Bundesbeitrag als Folge der Kopplung an AHV-Ausgaben. Die Berechnung liegt (in der uns übermittelten Version) ausserhalb der Standard-ATLAS-Pipeline, verwendet aber deren Outputs.

3.4.2 Übergeordnete Einschätzung

Das Modul ist methodisch sorgfältig aufgebaut: Die drei Zahlungswirkungen sind sauber getrennt, die Kohorten-Verschiebungslogik ist mathematisch korrekt umgesetzt und die zentralen Annahmen (AHV-21-Baseline, $\text{cum_shift} \leq 1$) sind dokumentiert.

Auf übergeordneter Ebene ergeben sich fünf Befunde

- Erstens war die intertemporale Sterblichkeitsgewichtung (wf_mort) ursprünglich am Ziel statt am Ursprungszustand der Kohortenfortschreibung definiert, was die verschobenen Renten- und Beitragssummen systematisch unterschätzte; dies ist der methodisch gewichtigste Befund und wurde inzwischen korrigiert (siehe Anhang B.4).
- Zweitens behandelten die Ausgaben- und Einnahmenseite die Verschiebung asymmetrisch (rekursive Teilverschiebung gegenüber einmaliger Verschiebung des aggregierten cum_shift); dies ist nur bei unterjährigen Rampen relevant und ohne Auswirkung auf die AHV-Pipeline (siehe Anhang B.4). Die rekursive Teilverschiebung wurde bereits behoben.
- Drittens beruht das Modul auf der Ausgabenseite auf der starken Annahme vollständiger Passivität ($p = 1$, vollständiges Mitwandern des Rentenbezugs mit dem Referenzalter); diese ist als Vereinfachung vertretbar und kann mindestens teilweise aus empirischen Ergebnissen abgeleitet werden. Das BSV hat uns zugesichert, die Annahmen, unter denen $p = 1$ gilt, in der Begleitdokumentation transparent darzulegen und klar zu differenzieren, was aus dem zitierten Forschungspapier⁴ abgeleitet werden kann und an welchen Stellen komplementäre Annahmen erforderlich sind (siehe Anhang B.4).
- Viertens empfehlen wir, hartcodierte Konstanten wo nötig zu parametrisieren und die Eingangsvalidierung über die blosse Existenzprüfung von coh hinaus auf $\text{alt}/\text{ref_alt}/\text{wf}$ auszuweiten (vgl. auch Kapitel 3.5).
- Fünftens verwendet das Modul zwar die Outputs der Standard-ATLAS-Pipeline, war jedoch zum Zeitpunkt der Prüfung noch nicht in deren Paketstruktur integriert. Die empfohlene Integration als Modul innerhalb der delfinverse-Pakete ist nach Angaben des BSV mittlerweile erfolgt.

⁴ Lalive, R., Magesan, A., & Staubli, S. (2023). How social security reform affects retirement and pension claiming. *American Economic Journal: Economic Policy*, 15(3), 115-150.

3.5 Gesamtarchitektur

Auf Ebene der Gesamtarchitektur ergeben sich aus unserer Prüfung vier übergeordnete Beobachtungen.

Modularisierung

Die Phasen Datenaufbereitung (`dinput`), Vorberechnung und Hauptberechnung (`delfin`) und übergeordnete Funktionen (`dinfra`) sind sauber modularisiert und auf eigenständige Pakete verteilt. Diese Aufteilung erachten wir als vorbildlich. Sie trägt massgeblich dazu bei, dass die einzelnen Module mit vertretbarem Aufwand gepflegt werden können und allfällige kritische Punkte frühzeitig erkannt und behoben werden.

Datenvalidierung

Wie in Abschnitt 3.1 ausgeführt, empfehlen wir perspektivisch die Einführung einer eigenen Validierungsschicht zwischen dem Einlesen der Originaldaten und den nachgelagerten Berechnungen. Diese Schicht hätte über die reine Plausibilitätsprüfung hinaus den Nebennutzen, die typischen Variablen (`jahr`, `alt`, `sex` und weitere häufig für Joins oder Filter genutzte Spalten) in einen einheitlichen Typ zu konvertieren. Dies ist besonders relevant bei `jahr`, das im aktuellen Code teils als `double`, teils als `integer` vorliegt.⁵ Architektonisch wäre eine solche Schicht entweder als eigenes Paket («`dvalidate`») oder als ausgebauter Teilbereich von `dinfra` umsetzbar. Wir skizzieren eine mögliche Umsetzung in Anhang C.1.

Umgang mit Warnungen und Mitteilungen

Ein weiterer Befund mit Querschnittscharakter betrifft die pauschale Unterdrückung von Warnungen und Mitteilungen mittels `suppressMessages()` und `suppressWarnings()`. Sie findet sich an mehreren Stellen, vorwiegend beim Einlesen von Excel- und Textdateien (`mod_input_bip_szenario`, `mod_input_arbeitslosenquote`, `mod_input_lohnindex`, `mod_input_freiwillig_versicherte`, `mod_input_preisindex`, `mod_read_data`), zudem im Imputationsschritt in `mod_ahv_hinterlassenrenten` und bei der globalen Typkonvertierung in `run_prepare_input`. Die Praxis birgt drei Risiken:

1. Erwartete und unerwartete Hinweise werden gleichermassen unterdrückt, so dass Code-Änderungen oder Paket-Updates stille Verhaltensänderungen erzeugen können;
2. Deprecation-Hinweise als einzige Vorwarnung vor Breaking Changes gehen verloren; und
3. Inhaltlich relevante Diagnostik (etwa von `readr` zu Parsing-Problemen oder von `fable` zu Modellkonvergenz) wird im Modelllauf nicht mehr sichtbar.

Konkrete Vorschläge zum Umgang mit dieser Thematik sind in Anhang C.2 ausgeführt.

⁵ Schlüsselvariablen wie `jahr` werden im Code teils als Fließkommazahlen (`double`) geführt. Fließkommazahlen werden binär nur näherungsweise gespeichert, sodass ein als 2024 gemeinter Wert intern minimal abweichen kann (z. B. 2023.9999999998). Exakte Vergleiche (`jahr == 2024`) und Joins über `jahr` können dadurch unbemerkt fehlschlagen oder Zeilen verlieren. Eine durchgängig ganzzahlige Kodierung (`integer`) schliesst dieses Risiko aus.

Hartkodierte Werte

Ein Kritikpunkt, der sich durch alle drei geprüften Pakete zieht, sind institutionell relevante Werte, die direkt im Code fixiert statt parametrisiert sind. Diese Praxis ist aus unserer Sicht dann unschädlich, wenn zwei Bedingungen kumulativ erfüllt sind:

1. Die Werte beziehen sich auf historische Ereignisse oder gesetzlich festgeschriebene Parameter;
2. Die Werte werden nur einmal benötigt und in der Code-Pipeline nicht dupliziert.

Beispiele für Werte, die *nicht* hartkodiert werden sollten, sind Fenstergrössen für Vor- und Nachbezug sowie die Verschiebung des Frauen-Referenzalters im Zuge der AHV-21-Reform. Wir regen an, diese Werte in einem zentralen Codebuch abzulegen. Daraus ergeben sich zwei Vorteile: (i) Anpassungen bei künftigen Reformen oder Datenrevisionen müssen nur an einer Stelle vorgenommen werden, und (ii) das Codebuch ist auch für nicht-technische Stakeholder zugänglich, da es sich mit überschaubarem Aufwand in eine Excel-Tabelle oder ein PDF-Dokument überführen lässt. Ein Vorschlag zur Umsetzung dieser Anregung ist in Anhang C.3 ausgeführt.

4 Fazit

Auf Basis der manuellen Prüfung, ergänzt durch den KI-gestützten Review, gelangen wir zu folgender Gesamteinschätzung des revidierten AHV-Modells ATLAS im Referenzszenario:

Der R-Code implementiert das Referenzszenario in den geprüften Paketen `dinput`, `delfin` und `dinfra` angemessen und gibt die institutionellen Rahmenbedingungen korrekt wieder. Die identifizierten wesentlichen Befunde sind benennbar, lokal eingrenzbar und mit überschaubarem Aufwand zu beheben. Die gewählte Modularisierung ist aus unserer Sicht vorbildlich und trägt entscheidend dazu bei, das Modell robust gegenüber künftigen Anpassungen zu machen. Die Struktur ist nachvollziehbar, die Verantwortlichkeiten der Pakete sind klar getrennt. Die Lesbarkeit des Codes ist sehr gut.

Wir sehen aufgrund der Prüfung keine grundlegenden Mängel, die einer Erstanwendung 2026 entgegenstehen würden. Einschränkend ist zu erwähnen, dass die Prüfung ausschliesslich auf Basis des bereitgestellten Codes und der mitgelieferten Daten erfolgte, welche aus Datenschutzgründen nicht alle Rohdaten umfassten. Auch vorab definierte Teile der Programmierung wie z.B. das Paket `doutput` waren nicht Teil der Überprüfung.

Hervorzuheben ist, dass das BSV im Rahmen der Klärungsschleife bereits sämtliche in Anhang B dokumentierten Anpassungen umgesetzt hat oder nachvollziehbar begründen konnte, dass eine Anpassung keinen Mehrwert bietet; die einzelnen Reaktionen sind in den Tabellen in Anhang B dokumentiert. Zu den umgesetzten Vorschlägen zählen sowohl die methodisch gewichtigeren Korrekturen (etwa die Sterblichkeitsindexierung im Modul Referenzaltererhöhung oder die Umstellung auf die exakte Fisher-Formel bei den Zinssätzen) als auch zahlreiche Wartbarkeits- und Robustheitsverbesserungen. Die wesentlichen angemerkten Punkte sind damit bereits behoben oder durch nachvollziehbare Begründungen entkräftet; die verbleibenden Kritikpunkte verlieren dadurch zusätzlich an Gewicht und betreffen überwiegend perspektivische, nicht erstanwendungskritische Verbesserungen.

A Prüfschema

Zur Beurteilung der Skripte wurde das nachfolgende Prüfschema für jedes Skript jeweils von beiden Prüfpersonen sowie dem KI-Modell unabhängig voneinander ausgefüllt.

Kopf

Feld	Eintrag
Skript	(z. B. R/mod_ahv_bilanz.R)
Paket	(dinfra dinput delfin)
Pipeline-Phase	(1-Input 2-Vorbereitung 3-Ausgaben 4-Einnahmen 5-Bilanz 6-Nachbearbeitung)
Reviewer	
Datum	

Bewertungsschema

Kürzel	Bedeutung
✓	Geprüft – korrekt
⚠	Geprüft – Anmerkung oder Unklarheit
✗	Fehler identifiziert
□	Nicht anwendbar für dieses Skript

1. Datenkonsistenz

#	Prüfpunkt	Status	Kommentar
1.1	Inputdaten stimmen mit der Dokumentation überein (Quelle, Dimensionen, Bestands- vs. Stromgrössen)		
1.2	Filterungen und Selektionen sind nachvollziehbar und korrekt		
1.3	Einheiten sind konsistent innerhalb des Skripts (CHF/Mio. CHF, Prozent/Dezimal, monatlich/jährlich)		
1.4	Zeitbezüge sind korrekt (Basisjahr, Abrechnungsjahr, Projektionshorizont); Jahresbereich lückenlos		
1.5	Join-Operationen sind korrekt (Join-Typ passt; keine stille Zeilen-Vervielfachung oder -Eliminierung)		

2. Logische Korrektheit

#	Prüfpunkt	Status	Kommentar
2.1	Formeln, Indizes, Summen, Gewichtungen und Rundungen entsprechen der Spezifikation		
2.2	Schwellenwerte, Beitragssätze und gesetzliche Parameter stimmen mit AHVG/AHVV überein		
2.3	Fortschreibungslogik: Rekursionen und Lag-Strukturen sind korrekt (t-1 vs. t, a-1 vs. a)		
2.4	Kontrollfluss (if/else, case_when) deckt alle relevanten Fälle ab; kein Fall fällt stillschweigend durch		
2.5	Übergang historisch → Projektion: Kein Doppelzählen, keine Lücke am Übergangsjahr		

3. Plausibilität

#	Prüfpunkt	Status	Kommentar
3.1	Größenordnungen und Vorzeichen sind plausibel		
3.2	Keine unerwarteten Sprünge oder Trendbrüche in den Ergebnissen		
3.3	Grenzfälle: Verhalten bei leeren Kohorten, Nullwachstum oder Extremwerten geprüft		

4. Code-Qualität

#	Prüfpunkt	Status	Kommentar
4.1	Kein stilles Vektor-Recycling; Indexierung korrekt (kein Off-by-one)		
4.2	NA-Handling: Fehlende Werte werden explizit behandelt (na.rm bewusst gesetzt, keine stillen NAs)		
4.3	Typ-Sicherheit: Keine unbeabsichtigten Typ-/Faktor-Konvertierungen; kein Fliessschma-Vergleich mit ==		
4.4	Keine unbeabsichtigten globalen Variablen; Funktionen greifen nicht auf das Global Environment zu		
4.5	Funktionsdesign: Klare Einzelverantwortung, alle Inputs als Parameter übergeben, konsistenter Rückgabetyt		
4.6	Keine hartcodierten Werte (Magic Numbers), die konfigurierbare Parameter sein sollten		

#	Prüfpunkt	Status	Kommentar
4.7	Kein toter Code (auskommentierte Blöcke, unerreichbare Pfade); Originaldaten werden nicht in-place verändert		
4.8	Spaltenzugriffe über Bezeichnungen, nicht Indizes		

5. Dokumentation (Skriptebene)

#	Prüfpunkt	Status	Kommentar
5.1	Funktionen haben Roxygen-Header (@param, @return, @export)		
5.2	Komplexe Berechnungsschritte verweisen auf die Spezifikation oder den Gesetzesartikel		
5.3	Annahmen und Einschränkungen sind dokumentiert (z. B. «gilt nur für Referenzszenario», «ab Jahrgang X»)		

Befunde

#	Zeile(n)	Schweregrad	Befund
1			
2			
3			
4			
5			

Schweregrade:  Wesentlich |  Beachtenswert |  Nachrangig

Zusammenfassung

Gesamteinschätzung: (2–3 Sätze)

B Relevante Einzelbefunde

Die nachfolgenden Tabellen fassen (i) relevante Einzelbefunde der Skriptprüfungen zusammen ($\triangle$ = beachtenswert, $\times$ = wesentlich); (ii) beschreiben, welche Massnahmen seitens BSS daraufhin vorgeschlagen wurden; und (iii) führen aus, wie das BSV mit dem Vorschlag umgeht («Rückmeldung BSV»). Die vollständige Liste von Befunden befindet sich in den konsolidierten Protokollen der Skriptprüfungen und ist nicht Teil dieses Berichts. Die BSV-Spalte verwendet folgende Symbole:

- ✓ Massnahme wird übernommen (allenfalls in präzisierter oder alternativer Form, mit kurzer Begründung).
- $\triangle$ Massnahme wird nur teilweise übernommen.
- $\times$ Befund oder Massnahme wird nicht übernommen (mit Begründung).
- diagnostischer Hinweis, kein Handlungsbedarf bzw. Operationalisierung an anderer Stelle.

B.1 Paket «dinput»

Tabelle 1: Auswahl relevanter Einzelbefunde für das Paket dinput

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
mod_input_emigration_beitragsjahre	⚠ Bei der Berechnung der durchschnittlichen Beitragsjahre wird die AHV21-Reform nicht berücksichtigt	Bei der Berechnung der durchschnittlichen Beitragsjahre die schrittweise Erhöhung des Frauen-Referenzalters gemäss AHV 21 berücksichtigen.	— Beobachtung berechtigt. Die schrittweise Anhebung des Frauen-Referenzalters ist in der Berechnung der durchschnittlichen Beitragsjahre noch nicht abgebildet. Eine saubere Trennung des Reform-Effekts von der Baseline-Projektion ist methodisch wünschenswert, aber nicht trivial. Wir nehmen den Punkt zur weiteren Diskussion auf.
wrap_input_pop_bestand / wrap_input_pop_szenario	⚠ Quoten (z. B. Erwerbsquote erwq) werden analog zu Beständen bei Aggregation der Sammelaltersgruppe 99+ aufsummiert. Für relationale Grössen führt die Summenbildung jedoch zu einer mathematisch invaliden Kennzahl. Die Altersgruppe ist für das Gesamtergebnis wenig relevant, daher stark begrenzte Wirkung.	Quoten werden bei Aggregation neu berechnet (nach der Aggregation der absoluten Zähler- und Nenner-Werte, nicht durch einfache Summation).	✓ Vorschlag übernommen: Quoten werden aus den aggregierten Zähler- und Nenner-Werten neu berechnet statt durch Summation der Anteile. Die numerische Auswirkung ist klein — im SAKE-Pfad sind nur wenige hundert Zeilen mit Alter ≥ 99 betroffen, der Szenario-Pfad ist mangels Quoten in diesem Altersbereich nicht tangiert.
wrap_input_pop_bestand / wrap_input_pop_szenario	⚠ Die beiden wrap_input_pop*-Funktionen bedürfen einer Anpassung; ensure_alt_range() wird zweimal lokal definiert.	Funktion ensure_alt_range() auslagern.	✓ Vorschlag übernommen: ensure_alt_range() wird in einen geteilten Helper ausgelagert.

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
mod_input_bfs_erwerbsszenario	X Sofern jahr als Faktor eingelesen wird, erfolgt eine Konversion in Faktorlevels (1, 2, ...) statt in Jahre.	Zunächst in Text konvertieren (as.character())	X Nicht reproduzierbar: ein vorgelagerter Schritt zur Behebung von Zeichensatz-Problemen wandelt die Jahresspalte bereits in Text um, sodass die anschließende Umwandlung in Ganzzahlen die korrekten Jahreszahlen liefert. Eine zusätzliche Konversion ist daher nicht nötig.
mod_input_freiwilling_versicherte	X Strukturbrüche in der Output-Datei in den Jahren 1990 und 2002	Hinweis auf korrupte Rohdaten. Wenn möglich, Fehler in den Rohdaten beheben. Ansonsten linear zwischen den angrenzenden Jahren interpolieren.	⚠ Wir folgen dem Vorschlag teilweise. 1990 (44'731 → 26'459 → 45'466) ist eine V-Form mit Wiederanstieg über das Vorjahr und plausibel ein administrativer Stichtag-Effekt — wir nehmen das Jahr in die lineare Interpolation in wrap_input_pop_bestand.R: 101-113 auf. 2002 ist hingegen ein realer Regimewechsel im Zuge der Personenfreizügigkeit: der Bestand vor 2001 (~45k) und ab 2007 (~41k) sind nicht direkt vergleichbar. Für 2002 imputieren wir daher auf das Niveau 2003 (~41'318) statt zu interpolieren.
wrap_input_pop_bestand	X Strukturbrüche in der Output-Datei in den Jahren 1990 und 2002; nur 2002 wird korrigiert	Hinweis auf korrupte Rohdaten. Wenn möglich, Fehler in den Rohdaten beheben. Ansonsten linear zwischen den angrenzenden Jahren interpolieren.	⚠ Analog zur Anpassung bei mod_input_freiwilling_versicherte; Begründung und Code-Änderung dort.
mod_input_ikregister	X Hochaltrige Personen enthalten (Maximalalter: 129 Jahre)	Rohdaten überprüfen	✓ Beobachtung sachlich richtig: Alter reicht von -1 bis 129, mit 348 Personen ab 110 Jahren und 3 mit Alter -1 — vermutlich fehlerhaft codierte Geburtsjahre

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
			realer Beitragszahler. Der Anteil bleibt im Sub-Promille-Bereich der Personen-Jahre und der Beitragssumme. Die Auffälligkeit melden wir den Rohdatenlieferanten; eine code-seitige Filterung erscheint angesichts der minimalen Aggregat-Wirkung nicht verhältnismässig.
mod_input_rentenregister_ahv	X Hochaltrige Personen enthalten	Rohdaten überprüfen	✓ Hinweis bestätigt: RR_AHV_ALLE_ALTER enthält Einträge bis alt = 123 (95 mit alt ≥ 110). Die pmin(99, alt)-Konsolidierung verschiebt diese in den alt=99-Bucket (dort 0.12 % der Personen bzw. 0.07 % der monatlichen Rentensumme = CHF 99'332 von 150.4 Mio.) — plausibel reale Auszahlungen mit fehlerhaften Geburtsjahr-Codes; wir leiten die Auffälligkeit an die Rohdatenlieferanten weiter und verzichten wegen der geringen Aggregat-Materialität auf eine code-seitige Filterung.
mod_input_saisonniers	X Output-File enthält für einige Personen (mutmasslich Frauen) im Alter 70-99 Jahre keine Angaben zu Geschlecht und Anzahl.	Plausibilisieren, ob Bestände statistisch auf 0 gesetzt werden dürfen oder imputiert werden müssen.	✓ Beobachtung berechtigt. Historische Saisonniers-Daten zeigen für Frauen ab 70 nahezu keine Fälle (sub-promille). Saisonniers waren ein überwiegend männliches Gastarbeiter-Phänomen, daher ist die Imputation = 0 für die fehlenden Jahrgänge empirisch gestützt. Wir machen die Annahme im Producer explizit (mod_input_saisonniers Z. 121-134), damit sie am Definitionsort sichtbar ist.

B.2 Paket «delfin»

Tabelle 2: Auswahl relevanter Einzelbefunde für das Paket delfin

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
mod_zinssaetze	⚠ Berechnung des nominalen Zinssatzes erfolgt mit vereinfachter Formel aus der Addition von Realzinssatz und der Inflationsrate.	Es wird vorgeschlagen, die exakte Berechnung («Fisher-Formel») zu verwenden: $(1 + r_{\text{real}}) * (1 + \pi) - 1$	✓ Vorschlag übernommen: der nominale Zinssatz in mod_zinssaetze. R:39-40 wird auf die exakte Fisher-Formel umgestellt. Der Effekt ist klein (Zinssatz-Differenz im Sub-Promille-Bereich, stationär rund 0.02 %), kumuliert sich aber über den Projektionshorizont auf einen mittleren zweistelligen Millionenbetrag im End-Fondskapital 2040.
mod_ahv_altersrentenzyklus	⚠ Endjahr (2024) hartkodiert. Wenn sich das Bezugsjahr ändert, muss auch Code angepasst werden.	Endjahr als Parameter übergeben.	✓ Übernommen: die hartkodierte Endjahr-Zahl im Erstrentenniveau-Korrekturschritt ist durch das parametrisierte Bezugsjahr ersetzt (mod_ahv_altersrentenzyklus. R:68). Reine Wartbarkeitsverbesserung, keine quantitativen Auswirkungen.
mod_ahv_altersrentenzyklus	⚠ Implizit wird bei der Berechnung des Rentenzyklus für die 65- bis 85-jährigen über Domizil und Nationalität aggregiert. Empirisch zeigt sich, dass zwischen Domizil Ausland und Domizil Schweiz relevante Unterschiede zwischen der Entwicklung des Rentenniveaus über die Zeit bestehen (Durchschnitt über alle Kohorten, differenziert nach Geschlecht).	Prüfen, ob eine Differenzierung nach Domizil sinnvoll erscheint. Wenn nicht nach Domizil differenziert wird, sollte das begründet werden.	✗ Die Beobachtung ist sachlich richtig, eine Differenzierung nehmen wir trotzdem nicht vor: pro Schätzzeile (Kohorte × Alter × Domizil × Geschlecht) würde die Stichprobe spürbar kleiner, besonders am hochaltrigen Rand mit wenig Auslands-Beobachtungen. Die Pooling-Verzerrung ist empirisch klein (Differenz CH vs. Ausland höchstens wenige Prozent am oberen Rand, in der Grössenordnung der Zellstreuung). Das Renteneintritts-Niveau wird zudem bereits nach Domizil differenziert — die gepoolte Zyklusform steuert nur den weiteren Verlauf.

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
mod_minimalrentenentwicklung	X Werte für den Lohn- und Preisindex sind als Datenreihen vorhanden. Hier werden sie aber neu berechnet, mit hartkodierten Startwerten.	Besser: Die Datenreihen mit den gesetzlich vorgeschriebenen Startwerten abgleichen.	⚠ Teilweise: die Beobachtung ist berechtigt, die quantitative Auswirkung ist aber exakt null — die Rekonstruktion reproduziert die publizierte Preisindex-Reihe über alle historischen Jahre identisch. Den Hinweis greifen wir bei der Code-Klarheit auf: die beiden hartkodierten Anschlusswerte für Lohn- und Preisindex werden als benannte Konstanten mit Quellenkommentar ausgelagert, ohne den Mechanismus zu ändern.
mod_population	X Bei Saisonniers, freiwillig Versicherten und Grenzgängern werden Zellen aufgefüllt, wenn keine Daten vorhanden sind. Das Jahr 2000 stellt einen Schwellenwert dar; Zellen in früheren Jahren werden mit dem Anteil des frühesten verfügbaren Jahrs hochgerechnet, Zellen in späteren Jahren mit spätestem verfügbarem Jahr.	Erläutern, warum das Jahr 2000 eine sinnvolle Schwelle darstellt. Vorgehen zur Imputation der fehlenden Werte erläutern.	✓ Vorschlag übernommen (reine Dokumentation): die Schwelle 2000 markiert die regulatorischen Stichtage rund um Personenfreizügigkeit, Revision der Freiwilligen-Versicherung und Aufhebung des Saisonnier-Statuts Anfang der 2000er. Wir ergänzen die Imputationslogik in <code>mod_population.R:252-272</code> mit erläuternden Kommentaren; keine Code-Änderung.
mod_wf_eprc	⚠ In den ESPOP-Rohdaten fehlen für die Jahre vor 1980 Angaben zu Einwanderung und Auswanderung. Die Zellen sind aber nicht NA, sondern 0. Das führt bei der <code>bj_umbuchung</code> dazu, dass Jahre vor 1980 fälschlicherweise auf 0 gesetzt werden (Zeile 89 ff.).	Fehlende Werte explizit als NA kodieren	X Beobachtung sachlich richtig, aber ohne praktische Wirkung: die nachgelagerte Umbuchungsformel koalesziert 0 und NA identisch zu 0. Ein Imputations-Test der fehlenden Wanderungssalden (historische Daten gesplittet nach Emi- versus Immigration in dieser Zeitspanne nicht verfügbar) ergab ein Bilanz-Delta von 0 CHF über alle Projektionsjahre, da die Projektion am Rentenregister-Stand verankert ist. Wir ergänzen einen Code-Kommentar, der diese Abschirmung dokumentiert.

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
mod_ahv_hinterlassenrente	⚠ Waisenrenten (üblicherweise unter dem Sujet Hinterlassenenrente mitgeführt) werden nicht behandelt	Dokumentieren, dass hier ausschliesslich Witwen-/Witwerrenten behandelt werden	⚠ Wir präzisieren in der Roxygen-Beschreibung des Hinterlassenenrenten-Moduls, dass hier ausschliesslich Witwen- und Witwerrenten behandelt werden. Ergänzend ein Hinweis am Output der Hinterlassenen-Projektion sowie am Zusammenführungs-Schritt in wrap_ahv_rentensumme_go.R: 146/183, dass Waisenrenten über den generischen Rentensummen-Pfad projiziert werden und damit im Aggregat enthalten sind.
mod_ahv_hilflosenentschädigung	⚠ Hilflosenentschädigung (HE) wird proportional zur aggregierten Rentensumme projiziert, was empirisch plausibel ist. Laut AHVG (Art. 43bis, Abs. 3) ist die HE jedoch an die Minimalrente gekoppelt.	Projektion anhand der Minimalrentenentwicklung würde gesetzliche Bestimmung genauer wiedergeben.	⚠ Die Hilflosenentschädigung wird als konstantes Vielfaches der aggregierten Altersrentensumme fortgeschrieben — der Quotient wird im Abrechnungsjahr fixiert. Der Hinweis auf die gesetzliche Kopplung an die Minimalrente ist sachlich richtig. Eine reine Minimalrenten-Projektion bildet aber die mit der Babyboomer-Alterung steigende Altersmix-Verschiebung der Bezügerstruktur nicht ab und wäre nicht zwingend genauer. Wir behalten den derzeitigen Proxy bei und prüfen die Form neu, sollte sich die Stabilität des Ankers in künftigen Datenrevisionen verschlechtern.
mod_ahv_13rente	⚠ Dezember-Sterbefälle werden bei der Berechnung des «Liechtenstein-Effekts» als Proxy für unterjährige Sterbefälle genutzt. Saisonalitätseffekte («Winterübersterblichkeit») bleiben daher aussen vor.	Verwendung eines empirischen Saisonalitätsvektors würde zusätzliche Präzision ermöglichen, Auswirkung auf Gesamtprognose jedoch gering.	✗ Die Beobachtung ist sachlich richtig: der Liechtenstein-Effekt unterstellt eine gleichmässig übers Jahr verteilte Sterblichkeit und blendet die Winter-Übersterblichkeit aus. Quantitativ greift die Korrektur jedoch nur an der 13. Rente (rund ein Zwölftel der Altersrentensumme) und bleibt auf die Gesamtausgaben im Sub-Promille-Bereich — BSS selbst stuft die Wirkung als gering ein. Der Aufwand einer empirischen Saisonalitätsmodellierung steht damit nicht im

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
			Verhältnis zum Präzisionsgewinn; wir verbleiben bei der Approximation.
mod_beitrags- summe	⚠ Die Bestimmung des Referenzalters für Frauen erfolgt als Stufenfunktion: <code>if_else(sex == "f" & jahr <= 2024, 64, 65)</code> . Die AHV 21 sieht jedoch eine schrittweise Erhöhung ab 2025 (um jeweils 3 Monate pro Jahrgang) vor.	Rentenaltergewichtung für AHV 21 monats-/jahresanteilig modellieren.	✓ Berechtigter Hinweis. Auf der Rentenseite ist die AHV-21-Reform als separater Ex-post-Aufschlag modelliert; beitragsseitig fehlt eine solche Korrektur-Schicht, weshalb die Rampe direkt in die Baseline-Projektion gehört. Die Vierteljahres-Staffelung des Frauen-Referenzalters (64 in 2024, dann 64.25 / 64.50 / 64.75 bis 65 ab 2028) wird in die Beitragssummen-Modellierung übernommen. Die quantitative Wirkung ist klein (Sub-Promille-Bereich im Übergangsfenster, zusätzlich durch den Anker im Abrechnungsjahr gedämpft); die Anpassung dient vor allem der konsistenten AHV-21-Behandlung über beide Seiten.
mod_diskontfaktor	⚠ Positionsbasierte Zeilenzugriffe (z. B. <code>preis[1] <- 0</code>).	Eliminierung positionsbasierter Zeilenzugriffe: <code>preis[1] <- 0 → mutate(preis = if_else(jahr == min(jahr), 0, preis))</code> .	✓ Übernommen. Der positionsbasierte Zugriff in <code>mod_diskontfaktor.R:32</code> entfällt ersatzlos — ein Filter weiter oben in <code>mod_eckwerte</code> entfernt die betroffene 1979-NA-Zeile bereits am Quellort. Keine quantitative Auswirkung; die Diskontfaktoren ab 1980 bleiben identisch.
mod_eckwerte / mod_diskontfaktor	⚠ NA für 1979 entsteht durch <code>mutate(lag(...))</code> ohne nachgelagerten Filter in <code>mod_eckwerte</code> ; erzwingt Workaround <code>preis[1] <- 0</code> in <code>mod_diskontfaktor</code> .	Filter nach <code>mutate(lag(...))</code> in <code>mod_eckwerte</code> (vermeidet das NA für 1979 und macht den Workaround <code>preis[1] <- 0</code> in <code>mod_diskontfaktor</code> überflüssig).	✓ Vorschlag übernommen: in <code>mod_eckwerte.R:84-90</code> wird nach dem <code>mutate(lag(...))</code> ein <code>filter(!is.na(lohn))</code> ergänzt. Damit entfällt die NA-Zeile für 1979 am Quellort und der Workaround in <code>mod_diskontfaktor</code> wird obsolet. Diskontfaktoren ab 1980 bleiben identisch.

B.3 Paket «dinfra»

Tabelle 3: Auswahl relevanter Einzelbefunde für das Paket dinfra

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
mod_read_data / path_out_resolve	⚠ Pauschales suppressMessages(suppressWarnings(...)) beim Datenimport bzw. Argument showWarnings = FALSE in dir.create() kann wertvolle Hinweise (Parsing-/Typ-/Pivot-Warnungen, fehlende Schreibrechte) verdecken.	mod_read_data: An der Quelle (read_delim) beheben oder gesammelt via purrr::quietly() / readr::problems() abfangen. path_out_resolve: Rückgabewert von dir.create() prüfen und bei FALSE abbrechen.	⚠ Bei mod_read_data berechtigter Hinweis: die pauschale Unterdrückung kann Typ-Konvertierungen verdecken (etwa Dezimalkomma oder "N/A" in numerischen Spalten). Wir entfernen die Warnungs-Unterdrückung bzw. sammeln die Meldungen pro Datei; die reine Hinweis-Unterdrückung in der CSV-Schleife bleibt, da dort nur Rauschen anfällt. Bei path_out_resolve übernehmen wir den Abbruch nicht: ein bereits existierendes Zielverzeichnis ist bei uns der Normalfall und würde unnötig auslösen; echte Anlage-Fehler fängt die nachgelagerte Pfad-Validierung im aufrufenden Modul ab.
diskontierung	⚠ across(c(-jahr, -diskontfaktor), ~ . * diskontfaktor) multipliziert alle übrigen Spalten ohne Typprüfung; nicht-numerische Spalten (bspw. sex, nat) führen zu Abbruch oder falschen Werten.	Auf numerische Spalten beschränken (where(is.numeric)) bzw. Scope/Spaltenanforderungen im Roxygen-Header dokumentieren.	⚠ Beobachtung sachlich richtig. Ein where(is.numeric)-Filter ersetzt aber das heutige laute Scheitern durch stilles Überspringen und kann künftige Struktur-Änderungen verdecken. Wir präzisieren stattdessen die Eingangs-Erwartung im Roxygen-Header (Jahres-Spalte plus ausschliesslich numerische Wertspalten) und behalten den Abbruch bei Verletzung bei.
shift_rae	X Die Annahme vollständiger Passivität (p = 1, Rentenbezug folgt der Referenzaltererhöhung 1:1) ist stark und nicht vollständig durch die zitierte Empirie (Lalive	Passivitätsparameter p global (PARAM_GLOBAL) definieren; bei Beibehaltung von p = 1 die	✓ Beobachtung berechtigt: Wir dokumentieren klar und transparent, unter welchen Annahmen p = 1 gilt, welche Schlussfolgerungen aus der zitierten Empirie

Skript	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
	et al. 2023) gedeckt; je nach Annahme über das Verhalten der aktiven Typen und je nach angenommenem Regime sind Werte zwischen 0.89 und 1 (Vorbezug nicht vorteilhaft) resp. 0.67 und 1 (Vorbezug vorteilhaft) plausibel.	zugrundeliegenden Annahmen klar benennen.	abgeleitet werden können und welche komplementären Argumente für unsere Annahme erforderlich sind.
shift_rae	⚠ Eingangvalidierung prüft nur die Existenz von coh, nicht von alt/ref_alt/wf; bei fehlenden Spalten entsteht ein wenig aussagekräftiger Tidyverse-Fehler tief im Pipe.	Eingangvalidierung um alt/ref_alt/wf erweitern.	⚠ Beobachtung berechtigt — geprüft wird nur coh. Die Funktion ist ein interner Helfer mit zwei Aufrufstellen, die alt, ref_alt und wf unmittelbar davor konstruieren; eine zusätzliche Existenzprüfung würde den Fehler nur eine Ebene tiefer verschieben, da die bestehende Meldung die fehlende Spalte ohnehin benennt. Wir ergänzen die erwarteten Spalten in der Funktions-Dokumentation.

B.4 Modul «Referenzaltererhöhung»

Tabelle 4: Auswahl relevanter Einzelbefunde für das Modul Referenzaltererhöhung

Skript/Funktion	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
mod_rae_ausgaben	X Die Ausgabenseite wendet Teilverschiebungen rekursiv auf bereits verschobene Daten an. Bei unterjährigen Rampen ($\text{rampe} < 12$) ist dies nicht äquivalent zu einem einmaligen kumulierten Shift, sondern erzeugt eine binomialartige Streuung über mehrere Alters-/Jahres-Lags.	Die Einnahmenseite aggregiert dagegen zuerst zu <code>cum_shift</code> und verschiebt dann einmal pro Kohortengruppe. Die Ausgabenseite könnte analog implementiert werden oder $\text{rampe} = 12$ als einzige zulässige Einstellung erzwingen.	✓ Beobachtung sachlich richtig: die iterative Anwendung erzeugt bei unterjährigen Rampen eine binomiale Lag-Streuung, die einmalige Verschiebung nicht. In der produktiven AHV-Pipeline wirkt sich das nicht aus, da nur ganzjährige Rampen eingesetzt werden. Wir vereinheitlichen die Ausgabenseite trotzdem auf den Ansatz der Einnahmenseite (Kohorten-Aggregation vor einer einzigen Verschiebung), damit beide Pfade konsistent bleiben.
mod_rae_ausgaben	△ Die Ausgabenseite verwendet implizit $p = 1$, d.h. damit wird angenommen, dass 100% der betroffenen Kohorten den Rentenbezug physisch um die Referenzaltererhöhung aufschieben. Der gesetzlich mögliche monatliche und teilweise Rentenvorbezug wird nicht als Verhaltensreaktion modelliert. Dadurch kann der kurz- bis mittelfristige Cash-Flow-Effekt des AHV-Fonds überschätzt werden, weil reduzierte Vorbezugsrenten früher fliessen könnten.	Das Vorgehen ist als Vereinfachung vertretbar, sollte aber als Limitation dokumentiert werden. Die Komplexität genauer abbilden würde eine Modellierung der Rentenvorbezugsreaktion anhand einer Annahme zum Anteil der Kohorte, der zum alten Referenzalter oder früher reduziert vorbezieht sowie dem dafür anzusetzenden Kürzungsfaktor.	△ Teilweise: empirische Evidenz für die Schweiz (Lalive et al., 2023) in Verbindung mit plausiblen Annahmen über das Verhalten aktiver Typen und die Attraktivität des Vorbezugs stützt die $p = 1$ -Approximation. Wir ergänzen den Code-Kommentar in <code>mod_rae_ausgaben</code> mit dieser Rechtfertigung sowie der Erklärung der Asymmetrie zur Beitragsseite ($p = p_{\text{beitraege}}$); eine darüber hinausgehende detaillierte Vorbezugs-Modellierung würde die programmatische Einfachheit des einheitlichen <code>verschiebung_rae</code> -Operators aufgeben und ist daher nicht geplant.

Skript/Funktion	Befund	Vorgeschlagene Massnahme	Rückmeldung BSV
mod_rae_ausgaben	X Der intertemporale Gewichtungsfaktor für das Überleben (wf_mort) wird aktuell über die Sterbewahrscheinlichkeit des Zieljahres und Zielalters definiert. Für die kohortendiagonale Fortschreibung einer Bestandsgrösse von t-1 nach t ist methodisch zwingend die Sterbewahrscheinlichkeit des Ursprungszustands anzuwenden. Der Vorgriff auf die zukünftige Sterblichkeit führt (bedingt durch den Anstieg der Mortalität im Alter) zu einer Unterschätzung der verschobenen Rentensummen.	Die Definition ist analog zur korrekten EPRC-Rekursion (Verwendung von lag(wr_sterblichkeit) innerhalb der Kohortenzeitreihe) anzupassen.	✓ Übernommen: Die Überlebenswahrscheinlichkeit wird neu am Ursprungszustand (t-1, alt-1) ausgewertet, konsistent mit der EPRC-Rekursion. Effekt klein: jährliche Verschiebung der verschobenen Rentensumme höchstens rund ein halbes Prozent, kumuliert über die Projektionsperiode im tiefen zweistelligen Mio.-Bereich (geringere Einsparung); Beitragsseite praktisch unverändert.

C Anpassungsvorschläge

C.1 Validierungsschicht

Die in Abschnitt 3.1 angesprochene Lücke besteht darin, dass die Originaldaten weitgehend ohne explizite Validierung in die gemeinsame Modellstruktur überführt werden. Wir schlagen ein zweistufiges Vorgehen mit zwei komplementären R-Paketen vor: *pointblank* als zentraler Validierungsrahmen an der Systemgrenze zwischen Originaldaten und Modellstruktur, ergänzt durch *assertr* für kleinere, lokale Plausibilitätsprüfungen innerhalb einzelner Berechnungsskripte.

pointblank für die Systemgrenze. Das Paket *pointblank* (Iannone, CRAN) erlaubt es, für jede Inputdatei einen deklarativen Validierungsplan zu definieren, der über `create_agent()` und `interrogate()` ausgeführt wird. Verletzungen werden je nach konfigurierter Schwelle als Hinweis gemeldet oder erzwingen einen Abbruch. Pro Inputdatei würde der Plan typischerweise die in Abschnitt 3.1 genannten Dimensionen abdecken:

- Strukturelle Konformität (z. B. `col_exists()`, `col_is_integer("jahr")`);
- Vollständigkeit in Schlüsselspalten (`col_vals_not_null()`);
- Zulässige Wertebereiche und Code-Mengen (`col_vals_between("alt", 0, 110)`, `col_vals_in_set("cgpr", c(10L, 20L))`);
- Eindeutigkeit von Schlüsseltupeln (`rows_distinct()`);
- Konsistenz zwischen Aggregaten und Einzelwerten.

Architektonisch lässt sich dies als eigenständiges Paket «dvalidate» realisieren, das je Inputmodul eine `interrogate_<modul>()`-Funktion exportiert; diese wird im jeweiligen `mod_input_*`-Skript unmittelbar nach dem Einlesen aufgerufen, vor jeder weiteren Typkonvertierung oder Umstrukturierung. Als Nebenprodukt entsteht über `get_agent_report()` ein versionierbarer Bericht je Modelllauf.

assertr für Plausibilität in Einzelskripten. Für Invarianten, die erst *innerhalb* einer Berechnungskette aussagekräftig formuliert werden können (etwa nach einem Join, einer Pivotierung oder einer rekursiven Fortschreibung) ist *assertr* (Fischetti, CRAN) ein passenderes Werkzeug. Es fügt sich mit `verify()` und `assert()` direkt in tidyverse-Pipelines ein und bricht den Skriptlauf bei Verletzung mit aussagekräftiger Meldung ab. So liesse sich etwa nach der Sammelaltergruppierung in `wrap_input_pop_bestand` ein `verify(all(erwq >= 0 & erwq <= 1))` einfügen, um die in Tabelle 2 dokumentierte (und mittlerweile behobene) invalide Aufsummierung von Quoten frühzeitig zu erkennen.

Abgrenzung der beiden Werkzeuge. *pointblank* adressiert die Validierung an der Systemgrenze mit Reporting und konfigurierbaren Aktionsstufen; *assertr* adressiert lokale Invarianten innerhalb der Berechnungslogik, bei denen ein sofortiger Abbruch die einzig sinnvolle Reaktion ist. Beide Pakete sind aktiv gepflegt und auf CRAN verfügbar.

C.2 Umgang mit Warnungen und Mitteilungen

Die in Abschnitt 3.5 angesprochene pauschale Unterdrückung lässt sich an den meisten Fundstellen durch gezieltere Mittel ersetzen. Wir empfehlen ein gestuftes Vorgehen entlang folgender Prinzipien.

- *Zunächst erwartete Hinweise an der Quelle abschalten:* Viele Mitteilungen entstehen aus konfigurierbarem Verhalten der eingesetzten Pakete. Die «New names»-Mitteilung von `readxl` lässt sich durch explizite Spaltennamen (`col_names = c(...)`) oder das Argument `.name_repair` vermeiden. In `read_delim()` unterdrückt `col_types = cols(...)` mit voll spezifizierten Spaltentypen sowohl die Spec-Mitteilung als auch viele Parsing-Warnungen, da die Konvertierung deterministisch festgelegt ist.
- *Gezielt unterdrücken statt pauschal:* Wo eine bestimmte erwartete Warnung wirklich entfallen soll, ist `withCallingHandlers()` mit einem Regex auf die Warnung vorzuziehen. Dadurch passiert nur die gewünschte Warnung den Filter, alle übrigen bleiben sichtbar und können bei Code- oder Paket-Updates wahrgenommen werden.
- *Diagnostische Hinweise abfangen:* In `mod_ahv_hinterlassenenrente` werden mit `suppressWarnings(impute_fable(...))` Warnungen aus der Modellanpassung unterdrückt. Diese können auf Konvergenzprobleme oder Datenlücken hinweisen. Wir empfehlen, sie strukturiert mit `purrr::quietly()` einzusammeln und in einem Lauf-Protokoll auszugeben, statt sie zu verwerfen.

In Fällen, in denen kein gezielterer Mechanismus praktikabel ist, sollte der Aufruf möglichst eng auf den einzelnen Funktionsaufruf bezogen und mit einem kurzen Kommentar versehen werden, der festhält, welche Warnung erwartet wird und warum sie ignoriert werden darf. Damit bleibt die Stelle für spätere Leser nachvollziehbar, und bei Paket-Updates kann gezielt geprüft werden, ob die Begründung noch trägt.

C.3 Zentrales Codebuch

Das in Kapitel 3.5 skizzierte Problem besteht darin, dass in den Skripten zwei Arten von hartkodierten Variablen auftauchen:

1. *Externe Codes aus Quelldateien:* `cgpr` $\in \{10, 20\}$ (Altersrente ord./aussord.), `csortie` $\in \{4, 9\}$ (Sterbefall-Klassifikation), `gpr` $\in \{43, 44\}$, `dom` < 100 (CH/Ausland), Nationalitäten-Mapping (`nat`), und weitere.
2. *Fachliche Schwellen / Konstanten:* 1987 als Startjahr der konsolidierten Rentenordnung, Fenstergrösse 5/6 Jahre, «2024» als RAM-Filterjahr.

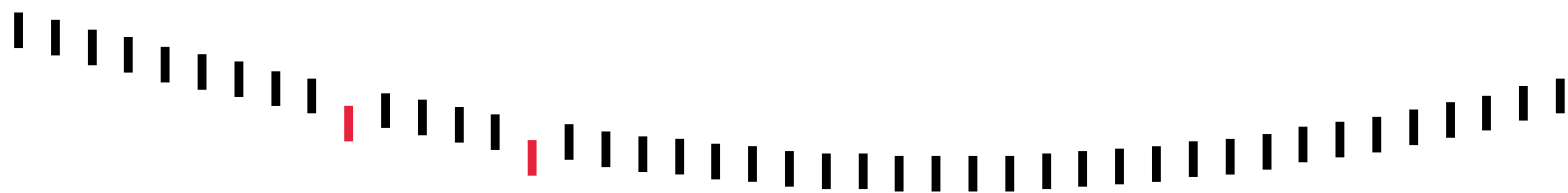
Der konzeptionelle Lösungsansatz besteht darin, im Paket eine einzige Stelle zu definieren, die alle externen Codes und fachlichen Konstanten kennt, dokumentiert (mit Quellenverweis) und für den Code maschinell zugänglich macht.

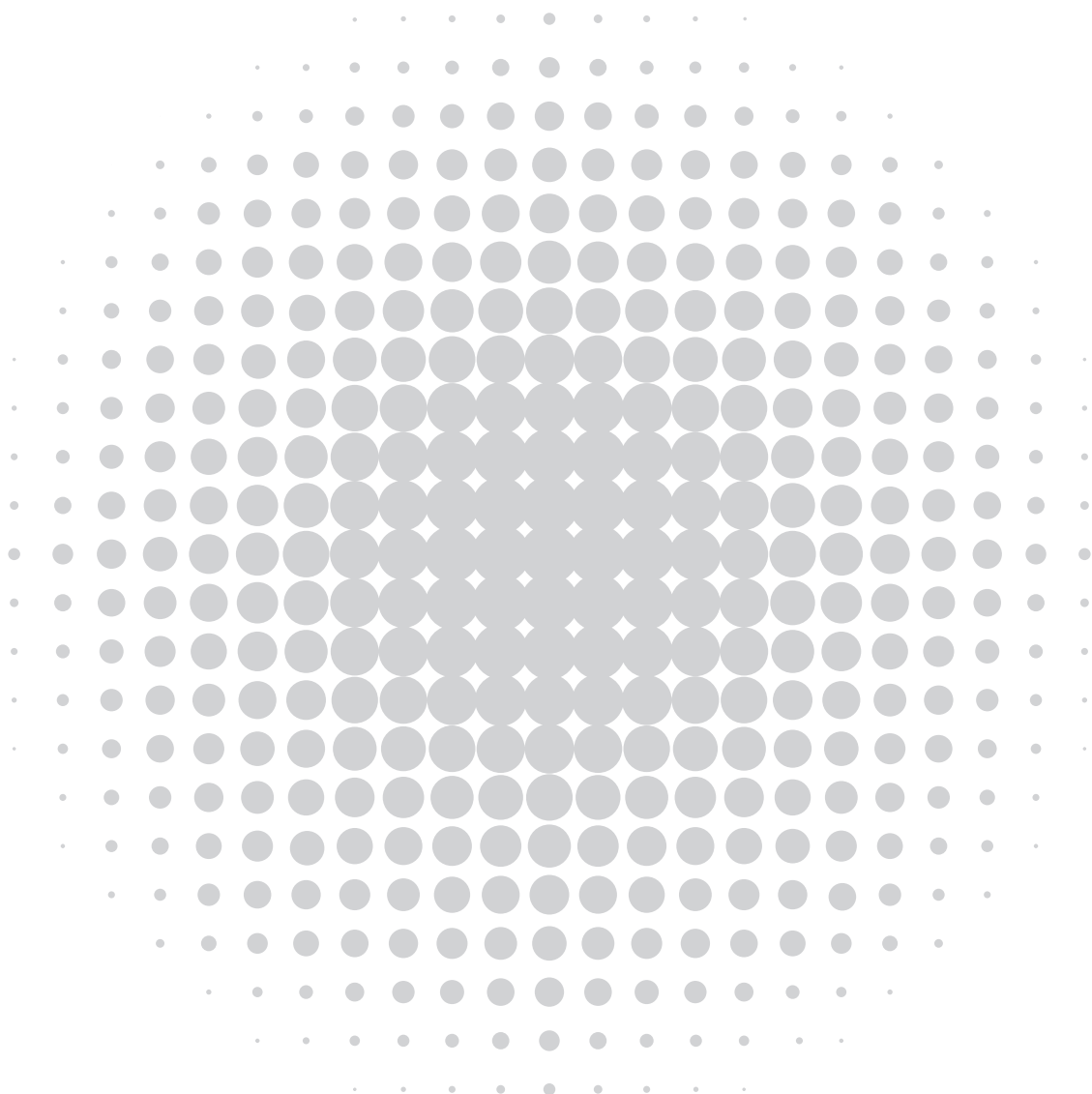
Das lässt sich beispielsweise wie folgt erreichen: Man übergibt dem Paket (im Beispiel `dinput`) eine Mapping-Tabelle als Datenobjekt, die unter `inst/codebook/` abgelegt und versioniert wird.

Beispiel:

```
codebook_rentenregister <- tribble(
  ~variable, ~code, ~label, ~bedeutung, ~quelle,
  "cgpr", 10L, "altersr_ord", "Altersrente ordentlich", "Codebuch [Bezeichnung,
  Stelle]",
  "cgpr", 20L, "altersr_ao", "Altersrente ausserordentlich", "Codebuch [Bezeich-
  nung, Stelle]",
  ...
)
```

Beim Paket-Build wird das Codebook via `usethis::use_data()` zu einem internen Datenobjekt, und ein kleines Helfer-Skript exportiert daraus die benannten Konstanten und Lookup-Helfer. Zum Beispiel: `codes("cgpr", labels = c("altersr_ord", "altersr_ao"))` gibt `c(10L, 20L)` zurück; `decode("cgpr")` gibt umgekehrt Labels aus.





bsv.admin.ch



Forschungspublikationen – «Beiträge zur Sozialen Sicherheit»