Package: staat1cho (via r-universe)

November 3, 2026

Title Study Indicators Based on Dutch Higher Education Data (1CHO)

Version 0.2.0

Description Calculates enrolment, graduation, dropout, and programme-switch indicators from the Dutch higher education registration data (1CHO) supplied by DUO. Includes an interactive 'Shiny' dashboard for exploring results.

License MIT + file LICENSE

URL https://github.com/cedanl/staat-van-onderwijsinstelling

BugReports https://github.com/cedanl/staat-van-onderwijsinstelling/issues

Encoding UTF-8

Language nl

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports cli, dplyr, forcats, readr, rlang, shiny

Suggests bslib, DT, ggplot2, knitr, plotly, readxl, rmarkdown, scales, testthat (>= 3.0.0), tibble, tidyr, writexl

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs cmake make libuv1-dev libx11-dev zlib1g-dev

Repository https://cedanl.r-universe.dev

Date/Publication 2026-11-03 09:48:01 UTC

RemoteUrl https://github.com/cedanl/staat-van-onderwijsinstelling

RemoteRef HEAD

RemoteSha 028e0579a5fdc29e07c0c8ad098cfc27e21b988b

Contents

BEKOSTIGINGSTATUS_CODES . . . 2
bereken_rendement . . . 3
bereken_studiewissel . . . 4
bereken_uitval . . . 5
bereken_wissel_xjr . . . 6
combineer_indicatoren . . . 7
DEFINITIES . . . 9
is_gepseudonimiseerd . . . 10
lees_bekostiging . . . 11
lees_vakhawv . . . 12
maak_analysebestand . . . 12
maak_basisbestand . . . 13
maak_benchmarkrapport . . . 14
maak_diploma_behaald . . . 16
maak_instroom_cohort . . . 17
maak_synthetische_1cho . . . 18
schrijf_analysebestand . . . 19
schrijf_benchmarkrapport . . . 20
start_dashboard . . . 21
verrijk_met_bekostiging . . . 22
verrijk_met_vakhawv . . . 23

Index **25**

`BEKOSTIGINGSTATUS_CODES`

Redencodes bekostigingstatus (VLPBEK/DEFBEK)

Description

Decodeertabel voor de kolom `CodeBekostigingstatus` uit het analysebestand voorlopige/definitieve bekostiging (DUO PvE HO-instelling, bijlage 8, par. 17.7.3/19.7.5). Een inschrijving of graad kan meerdere codes tegelijk hebben (komma-gescheiden in het brondata, bijv. `"na,ti"`).

Usage

```
BEKOSTIGINGSTATUS_CODES
```

Format

Een tibble met kolommen `code`, `omschrijving` en `herstelbaar`

Details

herstelbaar geeft aan of de reden voortkomt uit een registratiefout van de instelling die te herstellen is door tijdig opnieuw aan te leveren (ti/tg, waarde TRUE), een structurele/wettelijke reden is die niet te herstellen is (waarde FALSE), of helemaal geen "niet bekostigd"-reden betreft maar juist aangeeft dat er (deels) bekostigd wordt (waarde NA).

---

bereken_rendement *Bereken rendementsindicatoren per cohort*

Description

Koppelt diplomagegevens aan het instroomcohort en berekent of een student binnen 3, 5 of 8 jaar een diploma heeft behaald.

Usage

```
bereken_rendement(
  cohorten_instroom,
  diploma_behaald,
  niveau = "student",
  laatste_jaar = NULL
)
```

Arguments

cohorten_instroom	Tibble zoals gemaakt door maak_instroom_cohort()
diploma_behaald	Tibble zoals gemaakt door maak_diploma_behaald()
niveau	Analyseniveau: "student" (standaard) of "inschrijving". Moet overeenkomen met het niveau waarop cohorten_instroom en diploma_behaald zijn aangemaakt.
laatste_jaar	Integer, het laatste inschrijvingsjaar in de data. Standaard het laatste instroomjaar in cohorten_instroom. Cohorten waarvoor instroomjaar + x - 1 na dit jaar valt, krijgen voor rendement binnen x jaar "Nog niet waarneembaar": die studenten hebben nog geen x jaar kunnen studeren.

Value

Een tibble met kolommen voor de sleutel(s), eerstejaar_instelling, jaar_eerste_diploma, verblijfsjaar_eerste_diploma, diploma, soort_diploma, opleidingscode_diploma, rendement_xjaar, en factorkolommen rendement_3jr, rendement_5jr, rendement_8jr

Examples

```
cohort <- tibble::tibble(
  persoonsgebonden_nummer = c("S1", "S2"),
  eerstejaar_instelling = 2020L
)
diploma <- tibble::tibble(
  persoonsgebonden_nummer = "S1",
  jaar_eerste_diploma = 2022L,
  verblijfsjaar_eerste_diploma = 3L,
  diploma = "Diploma behaald (excl. propedeuse)"
)
bereken_rendement(cohort, diploma, laatste_jaar = 2025L)
```

---

bereken_studiewissel *Bereken studiewisselindicatoren per cohort*

Description

Bepaalt per student of zij binnen 1 of 3 jaar van opleiding zijn gewisseld. Studenten die al zijn uitgevallen of gediplomeerd in de meetperiode worden buiten beschouwing gelaten.

Usage

```
bereken_studiewissel(
  basisbestand,
  cohorten_instroom,
  diploma_behaald,
  uitval_indicatoren
)
```

Arguments

`basisbestand`	Tibble zoals gemaakt door `maak_basisbestand()`
`cohorten_instroom`	Tibble zoals gemaakt door `maak_instroom_cohort()`
`diploma_behaald`	Tibble zoals gemaakt door `maak_diploma_behaald()`
`uitval_indicatoren`	Tibble zoals gemaakt door `bereken_uitval()` Studiewissel is een studentniveau-indicator: `cohorten_instroom` moet één rij per student bevatten.

Value

Een tibble met kolommen `persoonsgebonden_nummer`, `studiewissel_1jr`, `studiewissel_3jr` (factoren; `"Nog niet waarneembaar"` als verblijfsjaar 2 resp. 4 van het cohort nog niet in de data zit) en aanvullende switch-kolommen met de opleiding, opleidingsvorm, niveau en sector na de wissel

Examples

```
basis <- tibble::tibble(
  persoonsgebonden_nummer = c("S1", "S1"),
  verblijfsjaar_actuele_instelling = c(1L, 2L),
  soort_inschrijving_actuele_instelling_label =
    "hoofdinschrijving binnen het domein actuele instelling",
  opleiding_actueel_equivalent = "34401",
  inschrijvingsjaar = c(2020L, 2021L),
  opleidingsvorm = "voltijd",
  type_hoger_onderwijs_binnen_soort_hoger_onderwijs = "bachelor",
  croho_onderdeel_actuele_opleiding = "techniek"
)
cohort <- tibble::tibble(
  persoonsgebonden_nummer = "S1",
  inschrijvingsjaar = 2020L,
  eerstejaar_instelling = 2020L,
  soort_diploma_instelling_label = NA_character_,
  opleidingsvorm_label = "voltijd"
)
diploma <- tibble::tibble(
  persoonsgebonden_nummer = character(0),
  jaar_eerste_diploma = integer(0),
  verblijfsjaar_eerste_diploma = integer(0),
  diploma = character(0)
)
uitval <- tibble::tibble(
  persoonsgebonden_nummer = "S1",
  uitval_xjr = NA_real_
)
bereken_studiewissel(basis, cohort, diploma, uitval)
```

---

bereken_uitval *Bereken uitvalindicatoren per cohort*

Description

Bepaalt voor elke student (of inschrijving) in het instroomcohort of zij zijn uitgevallen, zittend of gediplomeerd. Uitval wordt gemarkeerd als een student niet meer ingeschreven staat en geen diploma heeft.

Usage

```
bereken_uitval(
  basisbestand,
  diploma_behaald,
  cohorten_instroom,
  jaar = NULL,
  niveau = "student"
)
```

Arguments

basisbestand	Tibble zoals gemaakt door maak_basisbestand()
diploma_behaald	Tibble zoals gemaakt door maak_diploma_behaald()
cohorten_instroom	Tibble zoals gemaakt door maak_instroom_cohort()
jaar	Integer, peiljaar van de analyse (bijv. 2025). Studenten die in jaar - 1 nog ingeschreven staan, gelden als zittend. Standaard het jaar na het laatste inschrijvingsjaar in basisbestand. Een jaar waarbij de data inschrijvingen na jaar - 1 bevat geeft een fout, omdat zittende studenten dan als uitgevallen zouden tellen.
niveau	Analyseniveau: "student" (standaard) of "inschrijving". Moet overeenkomen met het niveau waarop de andere invoertibbles zijn aangemaakt.

Value

Een tibble met de sleutelkolom(men), laatste_jaar_inschrijving, diploma, status (factor: Diploma behaald / Zittend / Uitgevallen), uitval_xjr (jaar van uitval t.o.v. instroomjaar), uitval_1jr en uitval_3jr (factoren). Cohorten waarvoor de periode nog niet volledig in de data zit (instroomjaar + 1 resp. + 3 na jaar - 1) krijgen "Nog niet waarneembaar". Attribuut laatste_jaar is jaar - 1, het laatste inschrijvingsjaar in de data. Gooit een fout bij dubbele sleutelcombinaties of ontbrekende statussen.

Examples

```
basis <- tibble::tibble(
  persoonsgebonden_nummer = c("S1", "S1", "S2"),
  inschrijvingsjaar = c(2020L, 2021L, 2020L),
  soort_inschrijving_actuele_instelling = "hoofdinschrijving"
)
diploma <- tibble::tibble(
  persoonsgebonden_nummer = "S1",
  jaar_eerste_diploma = 2022L,
  verblijfsjaar_eerste_diploma = 3L,
  diploma = "Diploma behaald (excl. propedeuse)"
)
cohort <- tibble::tibble(
  persoonsgebonden_nummer = c("S1", "S2"),
  eerstejaar_instelling = 2020L
)
bereken_uitval(basis, diploma, cohort, jaar = 2023L)
```

---

bereken_wissel_xjr *Bereken studiewissel voor een specifiek verblijfsjaar-paar*

Description

Hulpfunctie die de switchlogica hergebruikt voor zowel de 1jr- als 3jr-meting. Een wissel wordt vastgesteld als een student in jaar `doeljaar` een andere opleidingscode heeft dan in jaar 1, en het verschil in kalenderjaren gelijk is aan `doeljaar - 1`.

Usage

```
bereken_wissel_xjr(
  basisbestand,
  zittend,
  verblijfsjaren,
  doeljaar,
  label_gewisseld,
  label_niet,
  suffix
)
```

Arguments

`basisbestand`	Tibble zoals gemaakt door `maak_basisbestand()`
`zittend`	Tibble met de populatie waarvoor de wissel bepaald wordt
`verblijfsjaren`	Integer vector van lengte 2: begin- en eindjaar van het meetvenster, bijv. `c(1, 2)` of `c(1, 4)`
`doeljaar`	Integer, het verblijfsjaar waarop de wissel wordt gemeten
`label_gewisseld`	Character, label als student gewisseld is
`label_niet`	Character, label als student niet gewisseld is
`suffix`	Character, achtervoegsel voor de kolomnamen (`"1jr"` of `"3jr"`)

Value

Een tibble met alleen de studenten die gewisseld zijn, met kolommen voor wissel, nieuwe opleidingscode, opleidingsvorm, niveau en sector

---

`combineer_indicatoren` *Combineer alle indicatoren tot een analysebestand*

Description

Voegt rendement-, uitval- en (optioneel) studiewisselindicatoren samen met het instroomcohort. Past kolomnamen en factorniveaus aan voor gebruik in rapportages.

Usage

```
combineer_indicatoren(
  cohorten_instroom,
  rendement_indicatoren,
  uitval_indicatoren,
  studiewissel_indicatoren = NULL,
  niveau = "student"
)
```

Arguments

`cohorten_instroom`	Tibble zoals gemaakt door `maak_instroom_cohort()`
`rendement_indicatoren`	Tibble zoals gemaakt door `bereken_rendement()`
`uitval_indicatoren`	Tibble zoals gemaakt door `bereken_uitval()`
`studiewissel_indicatoren`	Tibble zoals gemaakt door `bereken_studiewissel()`, of `NULL`. Studiewissel is een studentniveau-concept en alleen van toepassing bij `niveau = "student"`. Geef `NULL` door bij inschrijvingsniveau.
`niveau`	Analyseniveau: `"student"` (standaard) of `"inschrijving"`. Moet overeenkomen met het niveau waarop de andere invoertibbles zijn aangemaakt.

Value

Een tibble met gecombineerde indicatorkolommen, klaar voor rapportage. Bevat o.a. `status`, `rendement`, `uitval` en alle onderliggende deelscores. Bij `niveau = "student"` zijn ook studiewisselkolommen aanwezig als `studiewissel_indicatoren` is meegegeven.

`vooropleiding` vat de hoogste vooropleiding voor het HO samen (havo, vwo, mbo, ho, buitenlands, overig of onbekend). `eerstejaars_ho` geeft aan of het instroomjaar ook het eerste jaar in het hoger onderwijs is (`"eerstejaars HO"`) of dat de student al eerder in het HO stond (`"eerder in HO"`). Beide zijn `"onbekend"` als de 1CHO-kolommen `hoogste_vooropleiding_voor_het_ho_omschrijving_vooropleiding` resp. `eerste_jaar_in_het_hoger_onderwijs` ontbreken.

Attributen: `niveau` en `peildatum` (1 oktober van het laatste inschrijvingsjaar in de data, zie `maak_benchmarkrapport()`).

Examples

```
cohort <- tibble::tibble(
  persoonsgebonden_nummer = "S1",
  inschrijvingsjaar = 2020L,
  eerstejaar_instelling = 2020L,
  geslacht_label = "man",
  locatie_label = "Breda",
  opleiding_actueel_equivalent = "34401",
  opleidingsvorm_label = "voltijd",
  type_hoger_onderwijs_binnen_soort_hoger_onderwijs = "ba",
  indicatie_internationale_student_label = "geen internationale student",
```

```
  indicatie_eer_actueel_label = "geen EER-student",
  croho_onderdeel_actuele_opleiding_label = "techniek",
  leeftijd_per_peildatum_1_oktober = 19L,
  postcodecijfers_student_op_1_oktober = "4818",
  postcodecijfers_van_de_hoogste_vooropl_voor_het_ho = "4818",
  soort_diploma_instelling_label = NA_character_
)
rendement <- tibble::tibble(
  persoonsgebonden_nummer = "S1",
  eerstejaar_instelling = 2020L,
  jaar_eerste_diploma = NA_real_,
  verblijfsjaar_eerste_diploma = NA_integer_,
  diploma = NA_character_,
  rendement_xjaar = factor(NA_character_),
  rendement_3jr = factor("Geen diploma"),
  rendement_5jr = factor("Geen diploma"),
  rendement_8jr = factor("Geen diploma")
)
uitval <- tibble::tibble(
  persoonsgebonden_nummer = "S1",
  laatste_jaar_inschrijving = NA_real_,
  diploma = NA_character_,
  status = factor("Zittend"),
  uitval_xjr = NA_real_,
  uitval_1jr = factor("Na 1 jaar nog ingeschreven of diploma behaald"),
  uitval_3jr = factor("Na 3 jaar nog ingeschreven of diploma behaald")
)
wissel <- tibble::tibble(
  persoonsgebonden_nummer = "S1",
  studiewissel_1jr = factor("Niet gewisseld binnen 1 jaar"),
  studiewissel_3jr = factor("Niet gewisseld binnen 3 jaar"),
  opleidingscode_na_switch1jr = factor(NA_character_),
  opleidingsvorm_na_switch1jr = factor(NA_character_),
  opleidingsniveau_na_switch1jr = factor(NA_character_),
  sector_na_switch1jr = factor(NA_character_),
  opleidingscode_na_switch3jr = factor(NA_character_),
  opleidingsvorm_na_switch3jr = factor(NA_character_),
  opleidingsniveau_na_switch3jr = factor(NA_character_),
  sector_na_switch3jr = factor(NA_character_)
)
suppressWarnings(combineer_indicatoren(cohort, rendement, uitval, wissel))
```

DEFINITIES *Beschrijvingen van de studie-indicatoren*

Description

Korte definities van alle indicatoren die het package berekent, uitgesplitst naar analyseniveau. Afgeleid uit de berekeningslogica in `maak_instroom_cohort()`, `bereken_rendement()`, `bereken_uitval()` en `bereken_studiewissel()`. Bedoeld voor gebruik als tooltip-tekst in dashboards.

Usage

```
DEFINITIES
```

Format

Een nested named list met sub-lijsten `student` en `inschrijving`

Details

Gebruik `DEFINITIES[["student"]]$instroom` of `DEFINITIES[["inschrijving"]]$instroom` om de juiste definitie op te halen.

---

is_gepseudonimiseerd *Is een ID-kolom gepseudonimiseerd door 1cijferho?*

Description

Herkent de pseudoniemen die 1cijferho kan maken: 64 hexadecimale tekens. Een gepseudonimiseerd 1CHO-bestand kan niet gekoppeld worden aan een niet-gepseudonimiseerd VAKHAVW- of VLPBEK-bestand.

Usage

```
is_gepseudonimiseerd(x)
```

Arguments

`x`	Vector met persoonsnummers, bijv. `basis$persoonsgebonden_nummer`

Value

`TRUE` als alle niet-lege waarden pseudoniemen zijn, anders `FALSE`

Examples

```
is_gepseudonimiseerd(c("123456789", "987654321"))
is_gepseudonimiseerd(strrep("a1", 32))
```

`lees_bekostiging` *Lees een VLPBEK-bestand in*

Description

Leest een pipegescheiden VLPBEK-bestand (DUO-formaat, Voorlopige Bekostiging) in en extraheert de relevante kolommen per inschrijving. Alleen BRD-regels worden verwerkt; de koptekstregel (VLP), totaalregel (BLB) en sluitregel (SLR) worden genegeerd. Het peiljaar wordt uit de VLP-koptekstregel gelezen.

Usage

```
lees_bekostiging(pad)
```

Arguments

`pad`	Pad naar het VLPBEK-bestand (latin-1 gecodeerd, pipegescheiden)

Details

Koppelt nog niet aan het EV-bestand:

Het VLPBEK-bestand bevat het BSN of onderwijsnummer. Het persoonsgebonden nummer in het 1CHO-bestand (EV) is een eigen DUO-nummer en geen BSN, dus `verrijk_met_bekostiging()` vindt bij echte leveringen (vrijwel) geen inschrijvingen terug en waarschuwt. De koppeling wordt nog aangepast.

Value

Een tibble met de kolommen `persoonsgebonden_nummer`, `opleidingscode`, `inschrijvingsjaar`, `indicatie_hoofdinschrijving`, `opleidingsvorm`, `sector`, `bekostigingsstatus` (`"bekostigd"`, `"deels bekostigd"` (redencode `pd`) of `"niet bekostigd"`), `code_bekostigingstatus` (ruwe DUO-redencode(s)), `reden_niet_bekostigd` (leesbare toelichting, zie BEKOSTIGINGSTATUS_CODES) en `indicatie_herstelbaar` (`TRUE` als de redenen allemaal een te late aanlevering door de instelling zijn en dus hersteld kunnen worden, `FALSE` bij een structurele reden, `NA` als de inschrijving wel bekostigd is), plus het attribuut `peiljaar` (integer) dat aangeeft voor welk bekostigingsjaar het bestand geldt. Gereed voor gebruik in `verrijk_met_bekostiging()`

Examples

```
pad <- system.file("extdata/voorbeeld_vlpbek.csv", package = "staat1cho")
lees_bekostiging(pad)
```

lees_vakhawv *Lees een VAKHAVW-bestand in*

Description

Leest een semicolongescheiden VAKHAVW-bestand (DUO 1CHO-formaat) in, controleert de aanwezigheid van de vereiste kolommen en converteert cijferkolommen naar decimale waarden. DUO slaat cijfers op als gehele getallen vermenigvuldigd met 10 (bijv. 69 = 6.9).

Usage

```
lees_vakhawv(pad)
```

Arguments

pad	Pad naar het semicolongescheiden VAKHAVW-bestand (UTF-8)

Value

Een tibble met de ruwe vakcijferdata per student per vak, gereed voor gebruik in `verrijk_met_vakhawv()`

Examples

```
pad <- system.file("extdata/voorbeeld_vakhawv.csv", package = "staat1cho")
vakhawv <- lees_vakhawv(pad)
```

maak_analysebestand *Maak in één stap het analysebestand uit een 1CHO-bestand*

Description

Voert de hele rekenketen uit: `maak_basisbestand()`, `maak_instroom_cohort()`, `maak_diploma_behaald()`, `bereken_rendement()`, `bereken_uitval()`, `bereken_studiewissel()` (alleen op studentniveau) en `combineer_indicatoren()`. Het dashboard rekent met deze functie, dus de uitkomst is gelijk aan de CSV-download van het dashboard. Gebruik deze functie om staat1cho in een eigen pipeline op te nemen; de losse functies zijn er voor wie een tussenstap wil inzien of aanpassen.

Usage

```
maak_analysebestand(invoer, niveau = c("student", "inschrijving"))
```

Arguments

invoer	Pad naar de `_enriched.csv` van 1cijferho, of een data frame met dezelfde kolommen (bijvoorbeeld al ingelezen uit parquet of een database).
niveau	Analyseniveau: `"student"` (standaard) of `"inschrijving"`. Op studentniveau telt elke student één keer, bij de opleiding waarin die begon. Op inschrijvingsniveau is elke opleiding een eigen cohort.

Value

Een tibble met één rij per student (of per student-opleiding bij `niveau = "inschrijving"`), met attributen `niveau` en `peildatum`. Kolommen:

persoonsgebonden_nummer Persoonsnummer uit het 1CHO-bestand.

inschrijvingsjaar Instroomjaar (het cohort).

geslacht, leeftijd_bij_instroom, int_student, indicatie_EER Persoonskenmerken in het instroomjaar.

opleidingscode, opleidingsnaam, opleidingsvorm, opleidingsniveau, sector, locatie Opleiding van instroom. `sector` is het CROHO-onderdeel.

vooropleiding havo, vwo, mbo, ho, buitenlands, overig of onbekend.

eerstejaars_ho "eerstejaars HO", "eerder in HO" of "onbekend".

postcode4_student_1okt, postcode4_vooropleiding_voorHO Postcode (4 cijfers) van de student en van de vooropleiding.

status Diploma behaald, Zittend of Uitgevallen op de peildatum.

soortdiploma, opleidingscode_diploma, diploma_in_instroomopleiding Eerste diploma aan de instelling en of dat in de instroomopleiding is behaald.

rendement_3jr, rendement_5jr, rendement_8jr Diploma binnen x jaar, of "Nog niet waarneembaar".

uitval_xjr, uitval_1jr, uitval_3jr Jaar van uitval na instroom, en uitval binnen 1 en 3 jaar.

studiewissel_1jr, studiewissel_3jr, ..._na_switch1jr, ..._na_switch3jr Alleen op studentniveau: wissel van opleiding en de nieuwe opleiding.

rendement, uitval, studiewissel Samenvattingen in één kolom.

De precieze definities staan in DEFINITIES.

Examples

```
synth <- maak_synthetische_1cho(n_per_jaar = 20, jaren = 2018:2023)
analyse <- maak_analysebestand(synth)
table(analyse$inschrijvingsjaar, analyse$rendement_3jr)
```

---

maak_basisbestand *Lees het 1CHO-bestand in en voeg label-kolommen toe*

Description

Leest een semicolongescheiden CSV-bestand (UTF-8) in en voegt extra `_label`-kolommen toe voor gebruik in rapportages. Het pakket bevat een klein synthetisch voorbeeldbestand zonder echte persoonsgegevens (`inst/extdata/voorbeeld_1cho.csv`).

Usage

```
maak_basisbestand(pad_invoer)
```

Arguments

pad_invoer	Pad naar het semicolongescheiden CSV-bestand (UTF-8), of een data frame met dezelfde kolommen (bijvoorbeeld al ingelezen uit parquet of een database).

Value

Een tibble met alle 1CHO-regels plus extra _label-kolommen die de originele categorische waarden bewaren voor gebruik in rapportages

Examples

```
# voorbeeld_1cho.csv is a small synthetic dataset bundled with the package
pad <- system.file("extdata/voorbeeld_1cho.csv", package = "staat1cho")
basis <- suppressMessages(maak_basisbestand(pad))
```

---

maak_benchmarkrapport *Maak een geaggregeerd benchmarkrapport*

Description

Aggregeert het indicatorenbestand per sector, opleidingsvorm, opleidingsniveau en instroomjaar. Berekent percentages voor de kernuitkomsten en voegt een totaalrij per jaar toe.

Usage

```
maak_benchmarkrapport(
  indicatoren,
  drempel = 30L,
  min_cel = 5L,
  niveau = attr(indicatoren, "niveau"),
  peildatum = attr(indicatoren, "peildatum")
)
```

Arguments

indicatoren	Tibble zoals gemaakt door combineer_indicatoren(), eventueel aangevuld via verrijk_met_vakhawv() of verrijk_met_bekostiging().
drempel	Minimale groepsgrootte. Groepen kleiner dan deze waarde krijgen NA voor alle kolommen inclusief n. Standaard 30.
min_cel	Minimale celgrootte voor percentages (zie hierboven). Standaard 5; 0 zet celonderdrukking uit.
niveau	Analyseniveau waarop indicatoren is gemaakt ("student" of "inschrijving"). Standaard het attribuut niveau dat combineer_indicatoren() zet. Wordt vastgelegd in de metadata, zodat rapporten van verschillende instellingen vergelijkbaar blijven.
peildatum	Peildatum van de data (Date of "JJJJ-MM-DD"). Standaard het attribuut peildatum dat combineer_indicatoren() zet; ontbreekt dat, dan 1 oktober van het laatste instroomjaar in indicatoren.

Details

Onvolledige cohorten:

Percentages worden berekend over de waarneembare rijen: studenten met "Nog niet waarneembaar" (cohort te recent voor het meetvenster) tellen niet mee in teller of noemer. Is niemand in een groep waarneembaar, dan is het percentage NA.

Privacyonderdrukking:

- **Primair:** groepen met minder dan drempel studenten (standaard 30) krijgen NA voor alle uitkomsten, inclusief n.
- **Secundair:** is in een instroomjaar precies één groep primair onderdrukt, dan is die groep terug te rekenen uit de totaalrij min de zichtbare groepen. Daarom wordt dan ook de kleinste zichtbare groep in dat jaar onderdrukt.
- **Cellen:** een percentage wordt NA als de teller of het complement (noemer - teller) kleiner is dan min_cel (standaard 5), zodat bijvoorbeeld "0% uitval" in een groep niets over iedereen in die groep verraadt. Zet min_cel = 0 om dit uit te zetten.
- **Secundair per cel:** is in een instroomjaar precies één groep leeg voor een percentage, dan is dat percentage terug te rekenen uit de totaalrij. Dan wordt hetzelfde percentage ook leeggemaakt in de kleinste groep in dat jaar waar het nog zichtbaar is.
- Percentages worden afgerond op hele procenten, gemiddelden op één decimaal.

De drempel van 30 is een eigen keuze van het project, geen voorgeschreven norm. Laat een privacy officer of FG van de deelnemende instellingen drempel en min_cel toetsen voordat rapporten gedeeld worden.

De kolom onderdrukt maakt zichtbaar welke rijen zijn onderdrukt (primair of secundair) zonder iets te onthullen over de werkelijke omvang.

Peildatum:

Elke rij krijgt de peildatum van de data: standaard 1 oktober van het laatste inschrijvingsjaar (het attribuut peildatum dat combineer_indicatoren() zet). Cijfers van eerdere cohorten kunnen bij een latere peildatum veranderen, bijvoorbeeld als een student na een tussenjaar terugkeert en dan niet meer als uitgevallen telt. Vergelijk daarom alleen rapporten met dezelfde peildatum, en vermeld de peildatum bij elk gedeeld cijfer.

Samenstelling instroom:

Als het analysebestand vooropleiding en eerstejaars_ho bevat (zie combineer_indicatoren()), geeft het rapport per groep het aandeel havo-, vwo- en mbo-instromers en het aandeel eerstejaars HO. Onbekende waarden tellen niet mee in de noemer. Verschillen in rendement en uitval tussen instellingen hangen sterk samen met deze samenstelling.

Gebruik als validatiemiddel:

CEDA kan met dit rapport controleren of de tool correct werkt:

- **Totalen vs. DUO-publicaties**: n in de totaalrij per instroomjaar moet overeenkomen met het aantal studenten dat DUO voor die instelling en dat jaar publiceert. Een grote afwijking wijst op een verwerkingsfout.
- **Plausibele marges**: uitvalpercentages buiten circa 5-40 % of rendement van 0 % zijn een signaal om de pipelinestap te controleren.
- **Reproduceerbaarheid**: hetzelfde invoerbestand moet op elk moment identieke cijfers opleveren.

- **Optionele bestanden**: als VLPBEK of VAKHAVW is geladen, moeten de bijbehorende kolommen (`pct_bekostigd`, `gem_eindcijfer`) voor de meeste rijen gevuld zijn. Kolommen die volledig leeg zijn duiden op een koppelfout.

Value

Een tibble met kolommen `peildatum`, `sector`, `opleidingsvorm`, `opleidingsniveau`, `inschrijvingsjaar`, `onderdrukt`, `n`, `pct_uitval_1jr`, `pct_uitval_3jr`, `pct_rendement_3jr`, `pct_rendement_5jr`, `pct_rendement_8jr`, `pct_studiewissel_1jr`, `pct_studiewissel_3jr`, `pct_int_student`, `gem_leeftijd_instroom`, plus optioneel `pct_eerstejaars_ho`, `pct_vooropl_havo`, `pct_vooropl_vwo`, `pct_vooropl_mbo`, `gem_eindcijfer`, `gem_wiskundecijfer`, `gem_aantal_vakken`, `pct_bekostigd`, `pct_hoofdinschrijving` en `pct_herstelbaar` (van de niet-bekostigde rijen: percentage waarvan de reden een te late aanlevering is en dus hersteld kan worden, zie BEKOSTIGINGSTATUS_CODES). Attributen: `gegenereerd_op` (POSIXct) en `metadata` (lijst met peildatum, niveau, drempel, min_cel, laatste inschrijvingsjaar, geladen optionele bestanden, packageversie en tijdstip).

See Also

`schrijf_benchmarkrapport()` om het rapport met toelichting als Excel-bestand op te slaan.

Examples

```
df <- tibble::tibble(
  sector          = "gezondheidszorg",
  opleidingsvorm  = "voltijd",
  opleidingsniveau = "bachelor",
  inschrijvingsjaar = 2022L,
  uitval_1jr      = factor("Na 1 jaar nog ingeschreven of diploma behaald"),
  uitval_3jr      = factor("Na 3 jaar nog ingeschreven of diploma behaald"),
  rendement_3jr   = factor("Geen diploma"),
  rendement_5jr   = factor("Geen diploma"),
  rendement_8jr   = factor("Geen diploma"),
  int_student     = "geen internationale student",
  leeftijd_bij_instroom = 19L
)
maak_benchmarkrapport(df, drempel = 1L, niveau = "student")
```

---

maak_diploma_behaald *Maak een bestand met het vroegst behaalde diploma per student (of inschrijving)*

Description

Filtert het basisbestand op diplomasoorten die gelden als afgeronde opleiding en behoudt per sleutel alleen het eerste diploma op basis van diplomajaar.

Usage

```
maak_diploma_behaald(basisbestand, niveau = "student")
```

Arguments

basisbestand	Tibble zoals gemaakt door maak_basisbestand()
niveau	Analyseniveau: "student" (standaard) of "inschrijving". Bepaalt de sleutel waarop gededupliceerd wordt.

Value

Een tibble met één rij per student (bij niveau = "student") of per student-opleidingcombinatie (bij niveau = "inschrijving"), met kolommen voor de sleutel(s), jaar_eerste_diploma, verblijfsjaar_eerste_diploma (verblijfsjaar aan de instelling resp. in de opleiding), diploma, soort_diploma en opleidingscode_diploma (de opleiding waarin het diploma behaald is). Gooit een fout bij dubbele sleutelcombinaties.

Examples

```
basis <- tibble::tibble(
  persoonsgebonden_nummer = c("S1", "S2"),
  soort_diploma_instelling = c(
    "Hoofd-bachelor-diploma binnen de actuele instelling",
    NA_character_
  ),
  diplomajaar = c(2022L, NA_integer_),
  verblijfsjaar_actuele_instelling = c(3L, 1L)
)
maak_diploma_behaald(basis)
```

---

maak_instroom_cohort *Maak een eerstejaarscohort per instelling*

Description

Filtert het basisbestand op het opgegeven soort hoger onderwijs, hoofdinschrijvingen en eerste verblijfsjaar.

Usage

```
maak_instroom_cohort(basisbestand, soort_ho, niveau = "student")
```

Arguments

basisbestand	Tibble zoals gemaakt door maak_basisbestand()
soort_ho	Character vector met toegestane waarden van soort_hoger_onderwijs, bijv. c("hoger beroepsonderwijs", "hbo")
niveau	Analyseniveau: "student" (standaard) of "inschrijving". Bij "student" is de sleutel persoonsgebonden_nummer en wordt gefilterd op het eerste jaar aan de instelling (verblijfsjaar_actuele_instelling). Bij "inschrijving" is de sleutel persoonsgebonden_nummer + opleiding_actueel_equivalent en wordt gefilterd op het eerste jaar in de specifieke opleiding (verblijfsjaar_actuele_opleiding_inste zodat wisselaars als eerstejaars in hun nieuwe opleiding worden meegenomen.

Value

Een tibble met een rij per student (bij `niveau = "student"`) of per student-opleidingcombinatie (bij `niveau = "inschrijving"`), aangevuld met kolom `eerstejaar_instelling` (= inschrijvingsjaar). Gooit een fout als er dubbele sleutelcombinaties zijn.

Examples

```
basis <- tibble::tibble(
  persoonsgebonden_nummer = c("S1", "S2", "S3"),
  soort_hoger_onderwijs = c("hbo", "wo", "hbo"),
  soort_inschrijving_actuele_instelling_label =
    "hoofdinschrijving binnen het domein actuele instelling",
  verblijfsjaar_actuele_instelling = 1L,
  verblijfsjaar_actuele_opleiding_instelling = 1L,
  inschrijvingsjaar = 2020L,
  soort_diploma_instelling_label = NA_character_
)
maak_instroom_cohort(basis, "hbo")
```

---

maak_synthetische_1cho

Maak een synthetisch 1CHO-bestand met complete studieloopbanen

---

Description

Simuleert instroomcohorten van een fictieve hbo-instelling met vaste kansen op uitval, studiewissel en diplomering, in het kolomformaat van de 1cijferho-enriched output. Bedoeld voor demo's, de showcase en als referentie bij validatie: omdat per student bekend is wat er gebeurde, moet de pipeline die uitkomsten exact terugvinden.

Usage

```
maak_synthetische_1cho(
  n_per_jaar = 200L,
  jaren = 2012:2023,
  p_uitval_1jr = 0.15,
  p_uitval_later = 0.1,
  p_wissel = 0.1,
  seed = 1L
)
```

Arguments

`n_per_jaar`	Aantal instromers per cohort
`jaren`	Integer vector met instroomjaren; het laatste jaar is ook het laatste inschrijvingsjaar in de data
`p_uitval_1jr`, `p_uitval_later`, `p_wissel`	Kansen, zie hierboven
`seed`	Seed voor reproduceerbaarheid

Details

Per student wordt één loopbaan getrokken:

- met kans `p_uitval_1jr` uitval na het eerste jaar;
- anders met kans `p_uitval_later` uitval na jaar 2 of 3;
- anders met kans `p_wissel` een wissel naar een andere opleiding in jaar 2, gevolgd door een diploma;
- anders een diploma na 4, 5 of 6 jaar (kansen 0,5 / 0,3 / 0,2). Een wisselaar doet er een jaar langer over.

Het diploma staat op de inschrijvingsrij van het laatste studiejaar, met `diplomajaar` gelijk aan dat `inschrijvingsjaar` (dezelfde conventie als de DUO-data). Loopbanen worden afgekapt na het laatste jaar in `jaren`.

Value

Een tibble met één rij per student per inschrijvingsjaar, klaar om met `readr::write_delim(x, pad, delim = ";", na = "")` weg te schrijven en in te lezen met `maak_basisbestand()`. Het attribuut `waarheid` bevat per student `persoonsgebonden_nummer`, `instroomjaar`, `uitval_na` (jaar van uitval of `NA`), `diploma_na` (studiejaar van diplomering of `NA`), `gewisseld` (`TRUE` bij een wissel in jaar 2), `vooropleiding` (havo, vwo, mbo, buitenlands of onbekend) en `eerder_in_ho` (`TRUE` als de student twee jaar voor instroom al in het HO stond), zonder afkapping.

Examples

```
synth <- maak_synthetische_1cho(n_per_jaar = 20, jaren = 2018:2023)
head(attr(synth, "waarheid"))

pad <- tempfile(fileext = ".csv")
readr::write_delim(synth, pad, delim = ";", na = "")
basis <- maak_basisbestand(pad)
```

---

schrijf_analysebestand

Sla het analysebestand op als CSV voor Excel

Description

Schrijft met puntkomma als scheidingsteken en komma als decimaalteken, zodat een Nederlandse Excel het bestand direct in kolommen en met de juiste letters (UTF-8 met BOM) opent. Ook in Python (`sep=";"`), SPSS of Power BI is het zo in te lezen.

Usage

```
schrijf_analysebestand(analysebestand, pad)
```

Arguments

analysebestand	Tibble zoals gemaakt door maak_analysebestand().
pad	Pad van het CSV-bestand.

Details

Het bestand bevat persoonsnummers en is alleen voor gebruik binnen de instelling. Deel met andere instellingen alleen het benchmarkrapport.

Value

pad, onzichtbaar.

Examples

```
synth <- maak_synthetische_1cho(n_per_jaar = 20, jaren = 2018:2023)
pad <- tempfile(fileext = ".csv")
schrijf_analysebestand(maak_analysebestand(synth), pad)
```

---

schrijf_benchmarkrapport

Sla een benchmarkrapport op als Excel-bestand

Description

Schrijft het rapport uit maak_benchmarkrapport() naar een .xlsx met vier tabbladen: Rapport, Toelichting (per kolom, afgestemd op analyseniveau en drempel), Validatie (controlepunten voor CEDA) en Metadata (peildatum, niveau, drempel, packageversie, laatste inschrijvingsjaar). Vereist het package writexl.

Usage

```
schrijf_benchmarkrapport(rapport, pad)
```

Arguments

rapport	Tibble zoals gemaakt door maak_benchmarkrapport()
pad	Pad van het uitvoerbestand (.xlsx)

Value

pad, onzichtbaar

Examples

```
df <- tibble::tibble(
  sector = "gezondheidszorg", opleidingsvorm = "voltijd",
  opleidingsniveau = "bachelor", inschrijvingsjaar = 2022L,
  uitval_1jr = factor("Uitgevallen binnen 1 jaar"),
  uitval_3jr = factor("Uitgevallen binnen 3 jaar"),
  rendement_3jr = factor("Geen diploma"),
  rendement_5jr = factor("Geen diploma"),
  rendement_8jr = factor("Geen diploma"),
  int_student = "geen internationale student",
  leeftijd_bij_instroom = 19L
)
if (requireNamespace("writexl", quietly = TRUE)) {
  rapport <- maak_benchmarkrapport(df, drempel = 1L, niveau = "student")
  schrijf_benchmarkrapport(rapport, tempfile(fileext = ".xlsx"))
}
```

---

start_dashboard *Start het Staat van Onderwijsinstelling dashboard*

Description

Opent de interactieve Shiny app waarmee je een 1CHO CSV-bestand kunt uploaden en studie-indicatoren kunt verkennen.

Usage

```
start_dashboard()
```

Value

No return value, called for side effects.

Examples

```
if (interactive()) {
  start_dashboard()
}
```

verrijk_met_bekostiging *Verrijk indicatoren met bekostigingsinformatie uit een VLPBEK-bestand*

Description

Koppelt bekostigingsstatus per student per opleiding aan het indicatorenbestand via `persoonsgebonden_nummer` en `opleidingscode`. Wanneer een student meerdere BRD-regels heeft voor dezelfde opleiding geldt: als ten minste een regel de status `"bekostigd"` of `"deels bekostigd"` heeft, is de student bekostigd voor die opleiding.

Usage

```
verrijk_met_bekostiging(indicatoren, bekostiging)
```

Arguments

`indicatoren`	Tibble zoals gemaakt door `combineer_indicatoren()`, met kolommen `persoonsgebonden_nummer` en `opleidingscode`
`bekostiging`	Tibble zoals gemaakt door `lees_bekostiging()`

Details

Een VLPBEK-bestand beschrijft de inschrijvingen van één bekostigingsjaar. De koppeling legt dus de huidige bekostigingsstatus naast elk instroomcohort; studenten die in dat jaar niet (meer) ingeschreven stonden krijgen `NA`. De kolom `bekostiging_jaar` geeft aan op welk inschrijvingsjaar de status betrekking heeft.

De functie meldt hoeveel rijen gekoppeld zijn en waarschuwt als dat minder dan de helft is: dan gebruiken de bestanden verschillende persoonsnummers. Dat is bij echte leveringen nu altijd zo, zie `lees_bekostiging()`. Het attribuut `koppeling` bevat de aantallen.

Studenten zonder overeenkomst in het VLPBEK-bestand krijgen `NA` voor `indicatie_bekostigd` en `indicatie_hoofdinschrijving`.

Value

De indicatoren-tibble uitgebreid met `indicatie_bekostigd` (`TRUE`/`FALSE`/`NA`), `indicatie_hoofdinschrijving` (`TRUE`/`FALSE`/`NA`), `reden_niet_bekostigd` (leesbare toelichting(en), `NA` als wel bekostigd) , `indicatie_herstelbaar` (`TRUE`/`FALSE`/`NA`, zie BEKOSTIGINGSTATUS_CODES) en `bekostiging_jaar`. Heeft een student meerdere BRD-regels met verschillende redenen voor dezelfde opleiding, dan worden de unieke redenen samengevoegd.

Examples

```
indicatoren <- tibble::tibble(
  persoonsgebonden_nummer = c("1", "2", "3"),
  opleidingscode          = c("31001", "31002", "31003"),
  inschrijvingsjaar       = 2023L
)
bekostiging <- tibble::tibble(
  persoonsgebonden_nummer    = c("1", "2"),
  opleidingscode             = c("31001", "31002"),
  inschrijvingsjaar          = c(2023L, 2023L),
  indicatie_hoofdinschrijving = c(TRUE, TRUE),
  opleidingsvorm             = c("voltijd", "voltijd"),
  sector                     = c("gezondheidszorg", "economie"),
  bekostigingsstatus         = c("bekostigd", "niet bekostigd"),
  code_bekostigingstatus     = c(NA, "nf"),
 reden_niet_bekostigd     = c(NA, "Maximaal aantal bekostigde inschrijvingen overschreden."),
  indicatie_herstelbaar      = c(NA, FALSE)
)
verrijk_met_bekostiging(indicatoren, bekostiging)
```

`verrijk_met_vakhawv` *Verrijk indicatoren met VAKHAVW-vooropleidingsgegevens*

Description

Aggregeert vakcijferdata per student en koppelt ze aan het indicatorenbestand via `persoonsgebonden_nummer`. Berekent per student:

Usage

```
verrijk_met_vakhawv(indicatoren, vakhawv)
```

Arguments

`indicatoren`	Tibble zoals gemaakt door `combineer_indicatoren()`, met een kolom `persoonsgebonden_nummer`
`vakhawv`	Tibble zoals gemaakt door `lees_vakhawv()`

Details

- `vakhawv_gemiddeld_eindcijfer`: hoogste gemiddelde eindcijfer van de cijferlijst (max over meerdere opleidingsjaren als die voorkomen)
- `vakhawv_wiskundecijfer`: gemiddeld centraal examencijfer voor wiskunde (alle vakken waarvan de afkorting begint met `"wis"`)
- `vakhawv_aantal_vakken`: aantal unieke vakken op de cijferlijst

Studenten zonder overeenkomst in de VAKHAVW-data krijgen `NA` voor alle drie de kolommen. De functie meldt welk deel van de VAKHAVW-studenten is teruggevonden (attribuut `koppeling`)

en waarschuwt onder de 10%. Een lager aandeel dan 100% is normaal: VAKHAVW bevat ook studenten die voor het eerste cohort in de data begonnen.

DUO vult het persoonsgebonden nummer in EV aan met spaties en in VAKHAVW met nullen; de koppeling negeert voorloopnullen. Werkt op zowel student- als inschrijvingsniveau: bij inschrijvingsniveau worden de vooropleidingsgegevens van een student voor elke opleiding herhaald.

Value

De indicatoren-tibble uitgebreid met de kolommen `vakhawv_gemiddeld_eindcijfer`, `vakhawv_wiskundecijfer` en `vakhawv_aantal_vakken`

Examples

```
indicatoren <- tibble::tibble(
  persoonsgebonden_nummer = c("S001", "S002", "S003"),
  inschrijvingsjaar = 2020L
)
vakhawv <- tibble::tibble(
  persoonsgebonden_nummer = c("S001", "S001", "S002"),
  afkorting_vak = c("ne", "wis", "ne"),
  gemiddeld_cijfer_cijferlijst = c(7.2, 7.2, 6.5),
  cijfer_eerste_centraal_examen = c(7.0, 6.5, 6.2),
  cijfer_schoolexamen = c(7.5, 6.8, 6.8)
)
verrijk_met_vakhawv(indicatoren, vakhawv)
```

Index

∗ **datasets**
    BEKOSTIGINGSTATUS_CODES, 2
    DEFINITIES, 9

BEKOSTIGINGSTATUS_CODES, 2, *11*, *16*, *22*
bereken_rendement, 3
bereken_rendement(), *8, 9*, *12*
bereken_studiewissel, 4
bereken_studiewissel(), *8, 9*, *12*
bereken_uitval, 5
bereken_uitval(), *4*, *8, 9*, *12*
bereken_wissel_xjr, 6

combineer_indicatoren, 7
combineer_indicatoren(), *12*, *14, 15*, *22, 23*

DEFINITIES, 9, *13*

is_gepseudonimiseerd, 10

lees_bekostiging, 11
lees_bekostiging(), *22*
lees_vakhawv, 12
lees_vakhawv(), *23*

maak_analysebestand, 12
maak_analysebestand(), *20*
maak_basisbestand, 13
maak_basisbestand(), *4*, *6, 7*, *12*, *17*, *19*
maak_benchmarkrapport, 14
maak_benchmarkrapport(), *8*, *20*
maak_diploma_behaald, 16
maak_diploma_behaald(), *3, 4*, *6*, *12*
maak_instroom_cohort, 17
maak_instroom_cohort(), *3, 4*, *6*, *8, 9*, *12*
maak_synthetische_1cho, 18

schrijf_analysebestand, 19
schrijf_benchmarkrapport, 20
schrijf_benchmarkrapport(), *16*

start_dashboard, 21

verrijk_met_bekostiging, 22
verrijk_met_bekostiging(), *11*, *14*
verrijk_met_vakhawv, 23
verrijk_met_vakhawv(), *12*, *14*