

# Package ‘staat1cho’

September 30, 2026

**Title** Study Indicators Based on Dutch Higher Education Data (ICHO)

**Version** 0.2.0

**Description** Calculates enrolment, graduation, dropout, and programme-switch indicators from the Dutch higher education registration data (ICHO) 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

**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

**NeedsCompilation** no

**Author** Aslam Tanjung [aut, cre],  
Veerle van Son [aut],  
Damiëtte Bakx-van den Brink [aut]

**Maintainer** Aslam Tanjung <aslam.tanjung@surf.nl>

**Repository** CRAN

**Date/Publication** 2026-09-30 21:20:32 UTC

## 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 . . . . .         | 10        |
| lees_vakhawv . . . . .             | 11        |
| maak_basisbestand . . . . .        | 12        |
| maak_benchmarkrapport . . . . .    | 12        |
| maak_diploma_behaald . . . . .     | 15        |
| maak_instroom_cohort . . . . .     | 16        |
| maak_synthetische_lcho . . . . .   | 17        |
| schrijf_benchmarkrapport . . . . . | 18        |
| start_dashboard . . . . .          | 19        |
| verrijk_met_bekostiging . . . . .  | 20        |
| verrijk_met_vakhawv . . . . .      | 21        |
| <b>Index</b>                       | <b>23</b> |

---

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 | <i>Bereken rendementsindicatoren per cohort</i> |
|-------------------|-------------------------------------------------|

---

## 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 <a href="#">maak_instroom_cohort()</a>                                                                                                                                                                                                                         |
| diploma_behaald   | Tibble zoals gemaakt door <a href="#">maak_diploma_behaald()</a>                                                                                                                                                                                                                         |
| niveau            | Analyseniveau: "student" (standaard) of "inschrijving". Moet overeenkomen met het niveau waarop cohorten_instroom en diploma_behaald zijn aangeemaakt.                                                                                                                                   |
| 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 | <i>Bereken uitvalindicatoren per cohort</i> |
|----------------|---------------------------------------------|

---

## 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 <code>maak_instroom_cohort()</code>                                                                                                                                                                                                                                                                   |
| 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 <a href="#">maak_basisbestand()</a>                                                           |
| zittend         | Tibble met de populatie waarvoor de wissel bepaald wordt                                                                |
| verblijfsjaren  | Integer vector van lengte 2: begin- en eindjaar van het meetvenster, bijv. <code>c(1, 2)</code> of <code>c(1, 4)</code> |
| 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 <code>maak_instroom_cohort()</code>                                                                                                                                               |
| rendement_indicatoren    | Tibble zoals gemaakt door <code>bereken_rendement()</code>                                                                                                                                                  |
| uitval_indicatoren       | Tibble zoals gemaakt door <code>bereken_uitval()</code>                                                                                                                                                     |
| studiewissel_indicatoren | Tibble zoals gemaakt door <code>bereken_studiewissel()</code> , 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 ICHO-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 ICHO-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(strep("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 ICHO-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 ICHO-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_basisbestand | <i>Lees het ICHO-bestand in en voeg label-kolommen toe</i> |
|-------------------|------------------------------------------------------------|

---

### 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)

### Value

Een tibble met alle ICHO-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 | <i>Maak een geaggregeerd benchmarkrapport</i> |
|-----------------------|-----------------------------------------------|

---

### 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 <code>combineer_indicatoren()</code> , eventueel aangevuld via <code>verrijk_met_vakhawv()</code> of <code>verrijk_met_bekostiging()</code> .                                                                                        |
| 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 celderdrukking uit.                                                                                                                                                                   |
| niveau      | Analyseniveau waarop indicatoren is gemaakt ("student" of "inschrijving"). Standaard het attribuut niveau dat <code>combineer_indicatoren()</code> 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 <code>combineer_indicatoren()</code> 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 verraaft. 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 leegge maakt 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:

- **Totale 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 | <i>Maak een bestand met het vroegst behaalde diploma per student (of inschrijving)</i> |
|----------------------|----------------------------------------------------------------------------------------|

---

## Description

Filtret 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 <a href="#">maak_basisbestand()</a>                                           |
| 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 | <i>Maak een eerstejaarscohort per instelling</i> |
|----------------------|--------------------------------------------------|

---

## Description

Filtret 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 <a href="#">maak_basisbestand()</a>                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| soort_ho     | Character vector met toegestane waarden van soort_hoger_onderwijs, bijv. <code>c("hoger beroepsonderwijs", "hbo")</code>                                                                                                                                                                                                                                                                                                                                                                      |
| 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_instelling).<br>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

|                                                                                 |                                                                                                    |
|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <code>n_per_jaar</code>                                                         | Aantal instromers per cohort                                                                       |
| <code>jaren</code>                                                              | Integer vector met instroomjaren; het laatste jaar is ook het laatste inschrijvingsjaar in de data |
| <code>p_uitval_1jr</code> , <code>p_uitval_later</code> , <code>p_wissel</code> | Kansen, zie hierboven                                                                              |
| <code>seed</code>                                                               | 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\_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**

|                      |                                                                |
|----------------------|----------------------------------------------------------------|
| <code>rapport</code> | Tibble zoals gemaakt door <code>maak_benchmarkrapport()</code> |
| <code>pad</code>     | Pad van het uitvoerbestand ( <code>.xlsx</code> )              |

**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 ICHO 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 <a href="#">combineer_indicatoren()</a> , met kolommen persoonsgebonden_nummer en opleidingscode |
| bekostiging | Tibble zoals gemaakt door <a href="#">lees_bekostiging()</a>                                                               |

## 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\_hoofdschrijving.

## Value

De indicatoren-tibble uitgebreid met indicatie\_bekostigd (TRUE/FALSE/NA), indicatie\_hoofdschrijving (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 | <i>Verrijk indicatoren met VAKHAVW-vooropleidingsgegevens</i> |
|---------------------|---------------------------------------------------------------|

---

## 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 <a href="#">combineer_indicatoren()</a> , met een kolom persoonsgebonden_nummer |
| vakhawv     | Tibble zoals gemaakt door <a href="#">lees_vakhawv()</a>                                                  |

## 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 voorloophnullen. 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](#), [14](#), [20](#)

bereken\_rendement, [3](#)

bereken\_rendement(), [8](#), [9](#)

bereken\_studiewissel, [4](#)

bereken\_studiewissel(), [8](#), [9](#)

bereken\_uitval, [5](#)

bereken\_uitval(), [4](#), [8](#), [9](#)

bereken\_wissel\_xjr, [6](#)

combineer\_indicatoren, [7](#)

combineer\_indicatoren(), [13](#), [14](#), [20](#), [21](#)

DEFINITIES, [9](#)

is\_gepseudonimiseerd, [10](#)

lees\_bekostiging, [10](#)

lees\_bekostiging(), [20](#)

lees\_vakhawv, [11](#)

lees\_vakhawv(), [21](#)

maak\_basisbestand, [12](#)

maak\_basisbestand(), [4](#), [5](#), [7](#), [15](#), [16](#), [18](#)

maak\_benchmarkrapport, [12](#)

maak\_benchmarkrapport(), [8](#), [18](#)

maak\_diploma\_behaald, [15](#)

maak\_diploma\_behaald(), [3–5](#)

maak\_instroom\_cohort, [16](#)

maak\_instroom\_cohort(), [3](#), [4](#), [6](#), [8](#), [9](#)

maak\_synthetische\_1cho, [17](#)

schrijf\_benchmarkrapport, [18](#)

schrijf\_benchmarkrapport(), [14](#)

start\_dashboard, [19](#)

verrijk\_met\_bekostiging, [20](#)

verrijk\_met\_bekostiging(), [11](#), [13](#)

verrijk\_met\_vakhawv, [21](#)

verrijk\_met\_vakhawv(), [11](#), [13](#)