

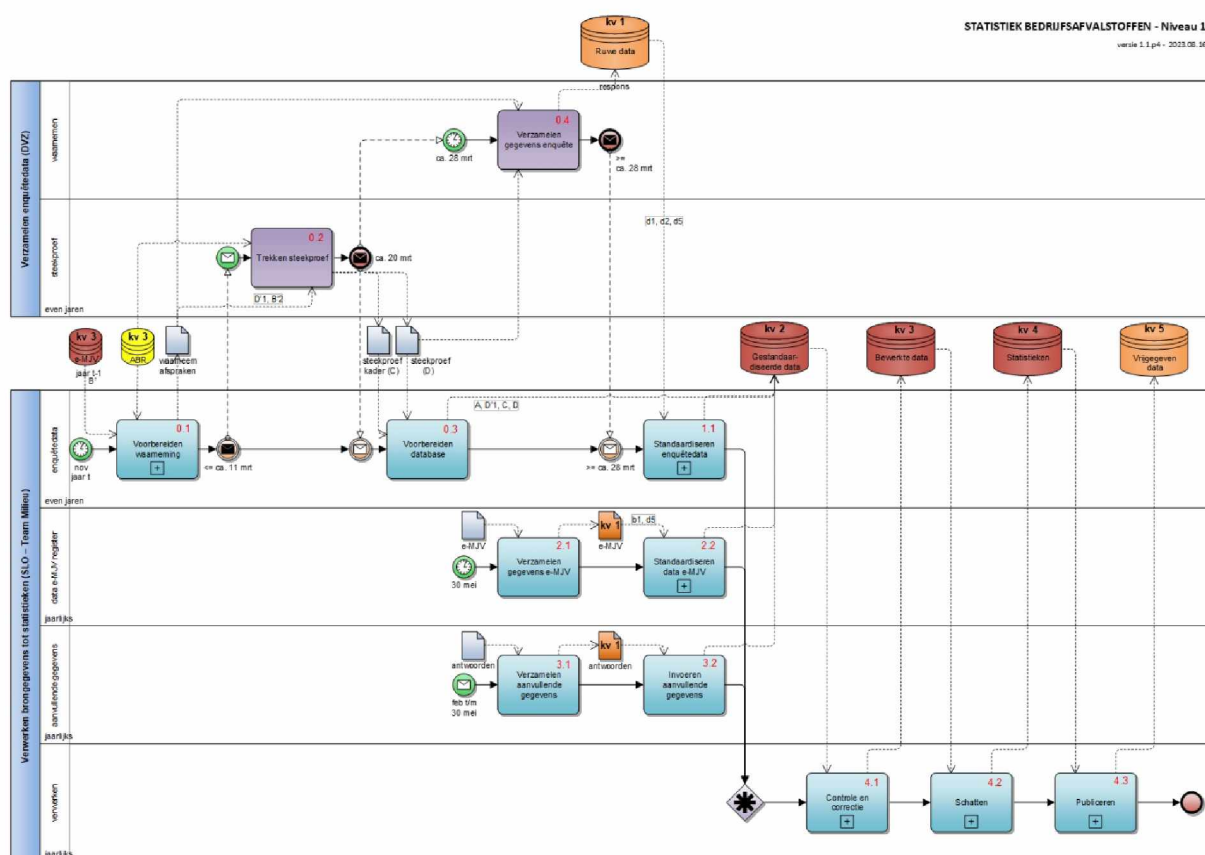
Beschrijving van automatiseringssysteem Bedrijfsafvalstoffen

Auteur: 5.1.2.e

1 Introductie

Het nieuwe automatiseringssysteem voor Bedrijfsafvalstoffen volgt de procesnummering zoals beschreven in het document *Verwerkingsproces Bedrijfsafvalstoffen naar KERS richtlijnen v.1.1.p.1* geschreven door 5.1.2.e en 5.1.2.e. De belangrijkste proceskeuzes worden in deze systeembeschrijving herhaald zodat duidelijk is waar de verschillende systeem modules betrekking op hebben.

Algemene procesplaat zoals in het document van 5.1.2.e en 5.1.2.e



Figuur 1 Procesplaat Statistiek Bedrijfsafvalstoffen - niveau 1

De statistiek Bedrijfsafvalstoffen kent drie databronnen met elk zijn eigen inhoud en kenmerken:

1. een door CBS uitgezonden enquête 'Bedrijfsafvalstoffen' (**zwembaan: enquête, hoofdstuk 3**),
2. het elektronisch Milieujaarverslag register van RIVM (**zwembaan: e-MJV, hoofdstuk 4**) en
3. aanvullende individuele dataverzameling via e-mail (**zwembaan: aanvullend, hoofdstuk 5**).

In figuur 1 is duidelijk te zien dat de verwerking van deze gegevens t/m koppelvak 2 Gestandaardiseerde data parallel uit te voeren processen zijn. Iedere databron wordt apart verwerkt en omgezet naar een CBS gestandaardiseerde vorm. Omdat de databronnen zo enorm verschillend zijn wil je ze eerst gelijk trekken in één datamodel voordat je verder gaat. Hierbij worden géén keuzes c.q. bewerkingen uitgevoerd die je specifiek doet voor je eigen output (ontwerpkeuze 2).

Zodra elke bron gestandaardiseerd is en beschikbaar gesteld in koppelvak 2 kan er een verwerkingsproces plaatsvinden waarin alle drie de bronnen gezamenlijk, inhoudelijk verwerkt worden (**zwembaan: verwerken, hoofdstuk 6**). Vanaf dat punt start de verwerking gericht op output (ontwerpkeuze 3). De enquête wordt een maal in de twee jaar (alleen voor de even jaren) uitgezonden. In de oneven jaren loopt het verwerkingsproces zonder deze databron.

1.1 Het automatiseringssysteem

Het automatiseringssysteem bestaat uit een SQL server database en een serie scripts. In de productie omgeving staat de database hier:

4.1.4

In de acceptatie omgeving is het adres:

4.1.4

De hoofdmap van het systeem is in de productie omgeving nog nader te bepalen. In de acceptatie omgeving is hij hier te vinden:

4.1.4

In de database staan de koppeltafelen en de tafelen die de koppelvlakken 2, 3 en 4 vormen. Overige rustpunten die worden gemaakt gedurende het draaien van de scripts worden in de logmap opgeslagen in afzonderlijke bestanden.

Naast de database, bevat het systeem (hoofdmap nog nader te bepalen) een map \scripts waar alle scripts te vinden zijn en de .cmd bestanden om de scrips mee te draaien.

Er is een map \instellingen, waar gebruikersinstellingen en regels in geplaatst kunnen worden. Alle instellingen en regels in deze map zijn met een standaard tekstverwerker zoals Kladblok of Notepad++ aan te passen. Dit is daarmee de enige plek waar de statistiek eigenaar/analist aanpassingen invoert buiten de koppeltafelen die in de database staan.

Verder is er een logmap genaamd \Logging waarin per datum/tijd te vinden is, welke resultaten de scripts hebben opgeleverd.

Er is ook een `\Import` map waar de analist data mee kan importeren in het systeem. Denk hierbij aan de EMJV data, RWS-LO data, Populatiebestand en Uitzendbestand.

Een belangrijke bestand is `algemene_instellingen.R`. Dit kan via RStudio of Notepad++ bewerkt worden. Instellingen die hier aangepast kunnen worden, zijn de locatie van de database en de locatie van het automatiseringssysteem zelf.

Elke afzonderlijke module schrijft het resultaat weg als `.rds` bestand (R eigen bestandstype) in de logmap. Dit zijn daarmee rustpunten. Elke volgende module leest het eindbestand van de voorgaande module als begin data in. De eerste modules van elk proces lezen data in uit het relevante koppelvlak. Naast elk weggeschreven `.rds` bestand wordt ook een `.csv` bestand opgeslagen. Deze is alleen ter info voor de analist om makkelijk te inhoud van het `.rds` bestand te kunnen zien. Het automatiseringssysteem maakt zelf geen gebruik van de `.csv` bestanden.

In de database staan de rustpunten van koppelvlak 2, 3 en 4. Deze tabellen bevatten allen de kolommen “InvoerDatum”, “EindDatum” en een autonummering kolom genaamd `Id`. Data die is aangepast wordt niet verwijderd, maar krijgt een einddatum mee. `Id` is alleen nodig om intern in de scripts te kunnen verwijzen naar de juiste record. Als “EindDatum” leeg is, dan zijn het nu geldige records. Er zijn voor alle tabellen ook Views aangemaakt waarmee direct altijd alleen de huidige records zichtbaar zijn.

Verder zijn er ook koppeltabellen in de database aanwezig zoals bijvoorbeeld “tbl_Koppel_1_1_3_Afvalverwerkingsmethode_DVZ_hercoderen” die verwerkingscodes van de ene vorm naar de andere zet. De methodiek van geldige records zou ik later een keer willen laten lijken op die van de tabellen in de rustpunten. Met het verschil dat er dan geen verslagjaar nodig is, maar een kolom die aangeeft vanaf welke datum een regel geldig is. Voor nu is echter de volgende methode in gebruik:

De koppeltabellen bevatten altijd een kolom “Jaar”. Als er geen wijzigingen in koppelingen zijn dan hoeft er niks veranderd te worden. Het automatiseringssysteem gebruikt dan altijd de meeste recente koppelingen. Als er bijvoorbeeld in verslagjaar 2022 geen wijzigingen zijn, zoekt het systeem naar koppelingen van het meest recente jaar. Als er koppelingen waren in 2018 en 2020, dan neemt het systeem de koppelingen van 2020 over.

Mochten er in 2022 wel wijzigingen optreden in 1 of meerdere koppelingen, dan moeten **alle** koppelingen voor 2022 worden bijgewerkt. Dit kan op een later moment eenvoudig gemaakt worden via een dashboard met achterliggend script. Op het huidige moment is het makkelijkste om deze stappen te volgen:

1. een kopie maken van alle 2020 records naar Excel,
2. het jaartal van alles aanpassen naar 2022,
3. de wijzigingen doorvoeren in de koppelingen waar iets is veranderd,
4. alle 2022 records toe te voegen aan de koppeltabel in de database.

Na die wijzigingen zullen de nieuwe 2022 koppelingen gebruikt worden voor het verwerken van nieuwe data. Mocht oude 2020 afvaldata opnieuw door het systeem gehaald worden, dan worden de oorspronkelijke 2020 koppelingen uit de koppeltabellen weer gebruikt. Alles blijft dus reproduceerbaar.

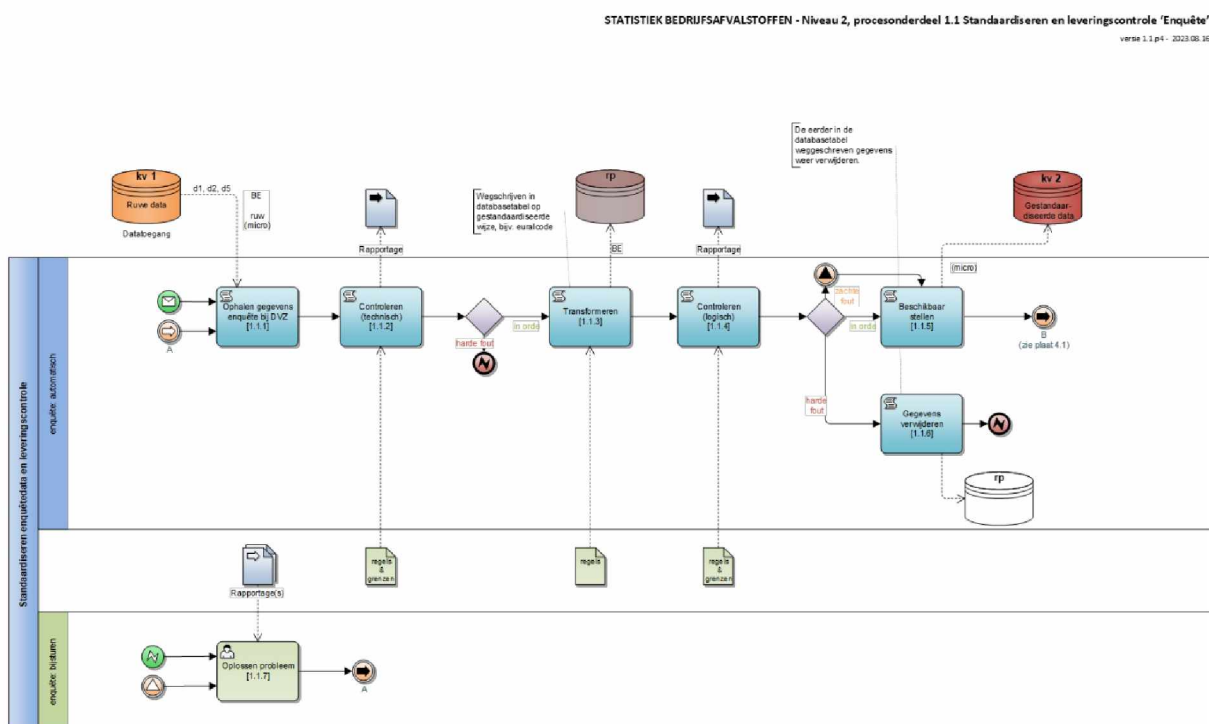
Er is een tabel in de database genaamd “tbl_HulpInformatie_Analysejaar”. Hierin staat maar één waarde en dat is het analysejaar. Daarmee weet het systeem voor welk jaar berekeningen gemaakt moeten worden en kan het checken of de invoerdata over het verslagjaar gaat.

2 Enquête

Eens in de twee jaar wordt een enquête gehouden. De enquête kent twee versies. Aan alle bedrijven in het onderzoek wordt gevraagd naar de afgevoerde afvalstromen. Bedrijven die ook recycle-activiteiten hebben (SBI 383) worden daarnaast ook gevraagd naar de aangevoerde afvalstromen.

2.1 Standaardiseren en leveringscontrole 'Enquête' [1.1]

Figuur 2 schetst het proces van standaardiseren en leveringscontrole van de enquêtegegevens meer in detail. Doel van deze processtappen is enerzijds om de ontvangen gegevens op een gestandaardiseerde wijze op te slaan in de eigen database en anderzijds om te controleren of de geleverde gegevens van voldoende kwaliteit zijn om op een verantwoorde manier statistieken mee te maken. Er worden nog geen correcties doorgevoerd of andere keuzes gemaakt die potentieel invloed hebben op de statistieken. De volgende paragrafen beschrijven de processtappen een-voor-een.



Figuur 2 Procesplaat Statistiek Bedrijfsafvalstoffen - niveau 2, procesonderdeel 1.1

Het gehele proces 1.1 wordt gedraaid via het opstarten van bestand "1.1-Standaardiseren_enquetedata.cmd".

Dit bestand voert alle afzonderlijke modules 1.1.1 t/m 1.1.5 in volgorde uit. Altijd wordt de volledige data van het verslagjaar die is gevonden bij Datatoegang opgehaald en in Koppelvlak 2 opgeslagen. Data die al eerder in Koppelvlak 2 was opgeslagen, blijft aanwezig, maar wordt aangepast met een einddatum zodat duidelijk is dat deze niet meer geldig is. Deze moet overigens altijd gelijk zijn, aangezien een bedrijf maar 1 keer kan responderen.

Output zijn de twee tabellen in de database Error! Reference source not found. en Error! Reference source not found. (zie LIM in hoofdstuk 7).

Nu volgt een beschrijving van de deelprocessen (modules).

2.1.1 Ophalen gegevens enquête bij DVZ

[1.1.1]

De steekproefwaarneming wordt door DVZ uitgevoerd en de resultaten worden beschikbaar gesteld in DataToegang (koppelvlak 1 Ruwe data). Het bestand dat klaarstaat is cumulatief: in één bestand zijn alle gegevens die tot dan toe binnen zijn opgenomen. Gaandeweg wordt het steeds aangevuld.

Script “1.1.1-Ophalen_gegevens_enquete_bij_DVZ.R” haalt de data op. Dit gaat aan de hand van de enquete instellingen in script “1.1.1-Enquete_instellingen.R” (instellingen map). De inhoud is dit:

```
INSTELLINGEN <- list(  
  baseUrl = "DTserver-PhoenixProductie.cbsp.nl",  
  
  afnemer = "BEDRAF_Afnemer_1",  
  enquetecode = "BEDRAF2022J",  
  datacontractcode = "MDC_BEDRAF22_V1",  
  uitvoeringsoort = "PROD", # PROD of LT  
  versieaanduiding = "1",  
  
  # Alleen van toepassing als link voor uitsluitend respons wordt gebruikt.  
  responsStatus <- "RESP"
```

Bij een nieuwe jaargang zal **enquetecode** en **datacontractcode** aangepast moeten worden. De rest kan gelijk blijven. Bij een logistieke test zal ook uitvoeringsoort aangepast moeten worden naar “LT”. De waarde bij **responsStatus** kan genegeerd worden. De gehele steekproef wordt altijd opgehaald, ook als er nog niet gerespondeerd is.

Het resultaat van dit script “1.1.1-Ophalen_gegevens_enquete_bij_DVZ.R” is een vierkant bestand met ongeveer 400 kolommen. Net zoals bij alle andere modules wordt het resultaat weggeschreven in de logmap als .rds bestand (en een .csv ter info).

Outputbestand: 1.1.1_enquetedata_kv1.rds

2.1.2 Controle (technisch)

[1.1.2]

Hierna volgt direct een automatisch proces dat de opgehaalde brondata controleert op gemaakte afspraken met DVZ. Het betreft dan enkel controles op (het technische gedeelte van) de leveringsvorm en of deze aan de (leverings-) afspraken voldoen.

Inputbestand: 1.1.1_enquetedata_kv1.rds

Script: 1.1.2-Controle_technisch.R.

Er wordt nu gecontroleerd of het bestand een dataframe is. Mocht er een errormelding ontstaan bij het pogen de DVZ data in te lezen zal het bestand namelijk een vector zijn met de errormelding als waarde.

Daarnaast checkt het script of er minstens 1000 rijen in staan en minstens 100 kolommen. Dan weet je dat er echte data in staat. Elk bedrijf in het uitzendbestand moet in de DVZ data aanwezig zijn. Ook als ze nog niet gerespondeerd hebben.

En ten slotte wordt gecontroleerd of minstens de volgende variabelen aanwezig zijn:

"EnqueteCode", "CollectieEenheidId", "CollectieEenheidType", "SteekproefEenheidId",
"SteekproefEenheidType", "Correspondentienummer", "ResponsStatus", "RedenNonRespons",
"ProductCode", "DataModelVersie", "UitvoeringSoort", "Versieaanduiding", "DataPuntVersie", "
Personeel.PersSterkte".

Op de laatste na, zijn dit allen standaard variabelen uit Phoenix, zodat je dan zeker weet dat de data goed overgekomen is. "Personeel.PersSterkte" is specifieke variabele voor de Bedrijfsafval enquête zodat je weet dat de juiste enquête data is opgehaald. Controleren op alle enquête variabelen wordt niet gedaan omdat het er zo'n 300 zijn met hele lange namen die ook niet een allemaal altijd aanwezig hoeven te zijn. In het JSON bestand van DVZ komen variabelen namelijk niet voor als er geen enkele respondent is die daar een waarde bij heeft ingevuld.

Verder wordt gecontroleerd of de data over het verslagjaar gaat zoals opgegeven in de databasetabel "tbl_HulpInformatie_Analysejaar". Dit gaat aan de hand van het veld **Enquetecode**. Hier staat een code in zoals "BEDRAF2020J". De 2020 hierin betreft het verslagjaar.

Als de checks niet doorlopen worden, wordt het systeem hier afgebroken. De analist moet dan zorgen dat de invoer juist gaat verlopen. Dit zal dan gaan via aanpassen van de waarden in "1.1.1-Ophalen_gegevens_enquete_bij_DVZ.R" en/of contact met DVZ.

De resultaten van de checks gaan via meldingen op het scherm.

2.1.3 Transformeren

[1.1.3]

Het doel van **Transformeren** is om de verdere verwerking en gebruik van deze variabelen onafhankelijk te maken van de bron. Dus als de technische vorm van de bron wordt aangepast, bijvoorbeeld als DVZ besluit een variabelennaam te wijzigen, dan hoeft enkel het transformeren te worden aangepast. In deze stap worden bijvoorbeeld de euralcode en verwerkingsmethode uit de lange lijst variabelen uit het DVZ bestand afgeleid met de bijbehorende afval hoeveelheid erbij geplaatst. Zo krijgt de data dezelfde vorm en de variabelen krijgen eenzelfde schrijfwijze en datatype als dat bij het transformeren van de andere twee databronnen gebeurt (EMJV en Aanvullende data).

Inputbestanden:

- 1.1.1_enquedata_kv1.rds
- Database: "tbl_Koppel_1_1_3-variabelen_met_RowID_en_Classificatie"
- Database: "tbl_Koppel_1_1_3-standaard_waarden_formulier_met_RowID"
- Database: "tbl_Koppel_1_1_3_Afvalverwerkingsmethode_DVZ_hercoderen"

Script 1.1.3-Transformeren.R

Niet-afval data

Data die niet over afval gaat zoals Responsdatum, aantal werkzame personen en contactgegevens worden opgeslagen in een aparte tabel.

Afval data

Data die wel over afval gaat wordt met behulp van de hulpdata in database tabel "tbl_Koppel_1_1_3-variabelen_met_RowID_en_Classificatie" gegroepeerd in een long-format bestand. Zo komen de afvalstoffen en de hoeveelheden die daarop betrekking hebben bij elkaar te staan.

Verder worden uit de database tabel "tbl_Koppel_1_1_3-standaard_waarden_formulier_met_RowID" nog een aantal standaard records toegevoegd die altijd gelden en daarom niet op het enquêteformulier gevraagd worden.

Via de koppeltabel "tbl_Koppel_1_1_3_Afvalverwerkingsmethode_DVZ_hercoderen" worden de verwerkingscodes zoals ze zijn ingelezen uit de DVZ data omgezet naar een standaard codering.

Secundaire grondstoffen zijn geen afval, maar komen via de enquête toch binnen onder een Eural afvalcode en een bestemming van de grondstof in het Omschrijving veld. Het script voegt in deze gevallen de teksten onder Eural_code (naam grondstof) en Omschrijving (bestemming grondstof) samen in het Omschrijving veld en de Eural_code krijgt de codering "000006" mee.

Variabelen namen worden hernoemd naar een standaard naam.

Outputbestanden:

- 1.1.3_enquetedata_niet-afval_gettransformeerd.rds
- 1.1.3_enquetedata_gettransformeerd.rds

2.1.4 Controle (logisch)

[1.1.4]

De logische controle die hier bedoeld wordt, gaat over inhoudelijke verwachtingen van de opgehaalde data en gaat dus niet over vorm/meta (zie daarvoor **Controle (technisch)**). Het resultaat van dit proces is een oordeel of de data van voldoende kwaliteit is om statistiek mee te maken en een 'rapport' waarin inzichtelijk gemaakt is welke controles uitgevoerd zijn en wat daar de resultaten van waren.

Inputbestanden:

- 1.1.3_enquetedata_gettransformeerd.rds
- Database: "tbl_Koppel_1_1_3_Afvalverwerkingsmethode_DVZ_hercoderen"
- Database: "tbl_Koppel_4_1_4_Verwerking_code_naam"
- Database: "tbl_Koppel_4_1_2_EURAL_EWC"

Script: 1.1.4-Controle_logisch.R

Dit script checkt of er dubbele records gevonden worden op basis van de combinatie van de velden "Jaar", "CollectieEenheidId", "Eural_code", "Verwerking_code". Gevonden dubbele records worden op het scherm weergegeven en weggeschreven in de logmap onder "1.1.4_Enquete_data_dubbele_records.csv".

Verder controleert het of de in module 1.1.3-Transformeren omgezette Verwerking_codes en de ontvangen Eural_codes in onze koppeltabellen voorkomen.

Er wordt gekeken of er formulieren zijn waar geen enkele waarde is ingevuld. De ID's van de bedrijven worden op het scherm getoond en een lijst van deze bedrijven met hun enquête data worden weggeschreven in de logmap in "1.1.4_lege_formulieren.csv"

Uiteindelijk zullen records met zogeheten harde fouten uit de data verwijderd worden en de rest wordt klaargezet om op te slaan in de database. Op dit moment zijn er geen controles die een harde fout kunnen opleveren.

Outputbestanden:

- 1.1.4_enquetedata_gecontroleerd.rds
- 1.1.4_Enquete_data_dubbele_records.csv
- 1.1.4_lege_formulieren.csv

2.1.5 Beschikbaar stellen

[1.1.5]

Zodra de ontvangen en getransformeerde enquêtebestanden goed genoeg zijn bevonden om verder mee te werken en statistiek mee te maken (geen of zachte fouten) kunnen ze definitief opgeslagen worden in de database. Daarmee is effectief koppelpunt 2 vormgegeven.

Inputbestanden:

- 1.1.3_enquetedata_niet-afval_getransformeerd.rds
- 1.1.4_enquetedata_gecontroleerd.rds

Script: 1.1.5-Beschikbaar_stellen.R

Hier wordt alleen gezorgd dat variabelen de juiste types meekrijgen waarmee ze in de database opgeslagen kunnen worden. De database (koppelvlak 2) wordt bijgewerkt met de nieuw ingelezen data. Het veld "Bron" krijgt de waarde "enquete" mee zodat duidelijk is dat het om records gaat die via de enquête zijn binnengekomen.

Outputbestanden:

- Database: "tbl_Koppelvlak_2_niet_afval_data"
- 1.1.5_enquetedata_niet-afval_beschikbaar_stellen.rds
- Database: "tbl_Koppelvlak_2_gestandaardiseerde_data"
- 1.1.5_enquetedata_beschikbaar_stellen.rds

3 E-MJV

De grens waarbij een bedrijf meldingsplichtig is voor het e-MJV ligt bij > 2000 ton niet-gevaarlijk afval. Je mag ook zonder verplichting de Emjv invullen. Dit zal vooral gebeuren bij bedrijven die rondom de grens liggen met hun hoeveelheid afval. Bedrijven die verder van de aangiftegrens afliggen zullen dit minder snel doen.

3.1 Verzamelen gegevens e-MJV [2.1]

Team Milieu heeft een eigen inlogcode waarmee ze e-MJVs uit het register zelf kan downloaden¹. Dit gebeurt op 30 mei volgend op het verslagjaar (figuur 1). Alle e-MJV plichtige berichtgevers hebben dan hun data ingestuurd en de controlerende instanties hebben dan de kans gehad hun taak uit te voeren. De verwachting is dan ook dat de kwaliteit van deze data hoog is. Na downloaden worden de e-MJV registerdata verzameld in een netwerkmap.

Het gaat om drie bestanden die met een wachtwoord kunnen worden opgehaald van de website van het Emjv. Voor verslagjaar 2021 waren dat de tabellen:

- “2021_algemene_gegevens (3).csv”,
- “2021_afval_stoffen_verwerkwijze (1).csv” , en
- “toelichtingen oordelen en adviezen”.

Deze 3 bestanden vormen het koppelvlak 1 voor de EMJV data.

De tabel algemene gegevens bevat gegevens over de vestiging (o.a. adres) en gegevens over de contactpersoon van de e-MJV eenheden. De tabel “2021_afval_stoffen_verwerkwijze (1).csv” bevat de gerapporteerde afvalstromen. In de tabel “toelichtingen oordelen en adviezen” tot slot zijn vragen en antwoorden opgenomen die voortkwamen uit de controles die de controlerende overheidsinstanties hebben gedaan.

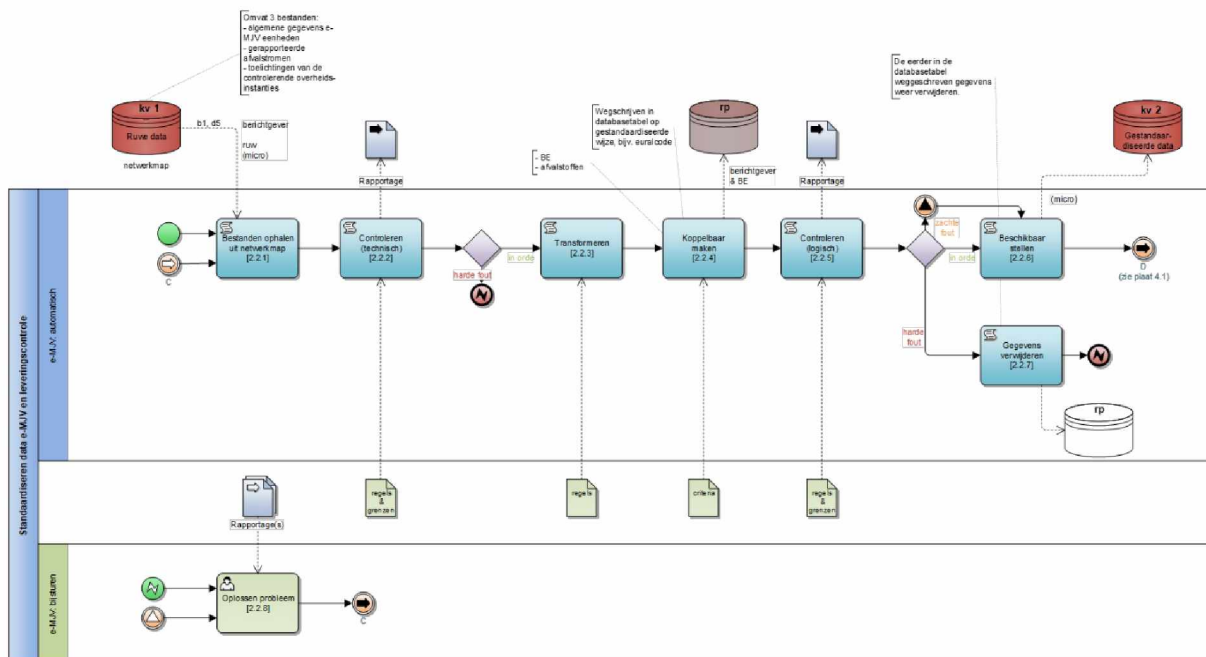
Na het **Verzamelen van e-MJV gegevens** (deelproces [2.1]) plaatst de analist de EMJV data in de map \Import.

3.2 Standaardiseren en leveringscontrole ‘e-MJV’ [2.2]

Figuur 3 geeft het proces van standaardiseren en leveringscontrole van de registerdata van het e-MJV. Doel van deze processtappen is enerzijds om de registergegevens op een gestandaardiseerde wijze op te slaan in de eigen database en anderzijds om te controleren of de opgehaalde gegevens van voldoende kwaliteit zijn om op een verantwoorde manier statistieken mee te maken. Er worden nog geen correcties doorgevoerd of andere keuzes gemaakt die potentieel invloed hebben op de statistieken.

De output van dit proces is dezelfde tabel Error! Reference source not found. (zie LIM in hoofdstuk 7) die ook al de output vormde van het proces over de enquête-data. Het doel is immers om alle data - van welke bron dan ook ontvangen – voor Koppelvlak 2 in een standaard format te plaatsen.

¹ Recentelijk zijn er gesprekken met de bronhouder geweest en mogelijk kunnen deze gegevens op afgesproken momenten geleverd gaan worden door de registerhouder (push).



Figuur 3 Procesplaat Statistiek Bedrijfsafvalstoffen - niveau 2, procesonderdeel 2.2

Het gehele proces 2.2 wordt gedraaid via het opstarten van bestand "2.2-Standardiseren_EMJVdata.cmd".

Dit bestand voert alle afzonderlijke modules 2.2.1 t/m 2.2.6 in volgorde uit. Altijd wordt de volledige data van het verslagjaar in Koppelvlak 2 opgeslagen. Data die al eerder in Koppelvlak 2 was opgeslagen, blijft aanwezig, maar wordt aangepast met een einddatum zodat duidelijk is dat deze niet meer geldig is.

Het bestand met Algemene Gegevens moet ingelezen zijn (module 0.0.3. EMJV algemene gegevens) voordat de EMJV afval data kan worden ingelezen. Voordat de EMJV afval data wordt ingelezen, wordt deze controle door het systeem uitgevoerd, waarbij de uitvoering van het script wordt afgebroken indien er niet aan voldaan is.

3.2.1 Gegevens ophalen uit netwerkmap

[2.2.1]

Inputbestanden:

- EMJV data (Excel bestand dat de analist in de map \Import heeft geplaatst)
- Database: tbl_HulpInformatie_Analysejaar

Script: 2.2.1-Ophalen_gegevens_EMJV.R

Het script komt al met de meest waarschijnlijk naam van het csv bestand met EMJV data aan door te kijken welk bestanden in de \Import map de tekst "afval_stoffen_verwerkwijze" bevatten. Analist kan het overrulen met specifieke naam. Script leest dan het bestand in.

Het script checkt of de data over het verslagjaar gaat. Zo niet, dan stopt het script. Als de data over het verslagjaar gaat, dan wordt het in de log weggeschreven en worden de volgende modules uitgevoerd.

Outputbestand: 2.2.1_EMJV_data_kv1.rds

3.2.2 Controle (technisch)

[2.2.2]

Hierna start direct een automatisch proces dat de opgehaalde data van het register controleert op de verwachte opzet van het bestand.

Inputbestand: 2.2.1_EMJV_data_kv1.rds

Script: 2.2.2-Controle_technisch.R

Het script controleert of de volgende kolommen aanwezig zijn zoals verwacht:

"jaar", "status", "nic", "bedrijf", "plaats", "provincie", "afval_code", "afval_categorie", "eural_code", "eural omschrijving", "eenheid", "afvalverwerkingsmethode", "ton_per_jaar", "percentage_drogestof", "afvalscheidingsinstallatie", "indicatie_eural_gevaarlijk", "indicatie_gevaarlijk_buitenland", "indicatie_zuiveringsslib".

Verder wordt gecontroleerd of de data over het verslagjaar gaat zoals opgegeven in de databasetabel "tbl_HulpInformatie_Analysejaar".

Mocht dit niet het geval zijn, stop het proces. Het is bij afkeuring raadzaam om contact op te nemen met het RIVM, de bronhouder van het register en dan te komen tot een oplossing. In figuur 3 is dit weergegeven in het deelproces **Oplossen probleem**.

3.2.3 Transformeren

[2.2.3]

Technische aanpassingen van de bron, waarbij de bron wordt omgezet naar CBS standaarden.

Vertaling van bronvariabelen naar CBS-definities waarbij geen informatie verloren gaat en geen informatie wordt toegevoegd. Het maken van koppelsleutels vormt hierop een uitzondering en is beschreven in **Koppelbaar maken**. De geplaatste bestanden worden ingelezen en weggeschreven op de juiste plekken in de database.

Inputbestand: 2.2.1_EMJV_data_kv1.rds

Script: 2.2.3-Transformeren.R

Via de koppeltabel "tbl_Koppel_2_2_3_Afvalverwerkingsmethode_EMJV_hercoderen" (uit de database) worden de verwerkingscodes zoals ze zijn ingelezen uit de EMJV data omgezet naar een standaard codering.

Variabelen namen worden hernoemd naar een standaard naam.

Outputbestand: 2.2.3_EMJV_data_gettransformeerd.rds

3.2.4 Controle (logisch)

[2.2.5]

Inputbestand:

- 2.2.3_EMJV_data_gettransformeerd.rds
- Database: "tbl_Koppel_4_1_4_Verwerking_code_naam"
- Database: "tbl_Koppel_4_1_2_EURAL_EWC"

Script: 2.2.4_EMJV_data_gecontroleerd.R

Dit script telt al het afval van alle bedrijven bij elkaar op en kijkt of dit hoger is dan 1000. Zo niet, dan komt er een harde fout en wordt het proces gestopt.

Verder controleert het script of de in module 2.2.3 omgezette Verwerking_codes en de ingevoerde Eural_codes in onze koppeltabellen voorkomen.

Aanwezig van dubbele records gevonden worden op basis van de combinatie van de velden "CollectieEenheidId", "Jaar", "Eural_code", "Verwerking_code". Gevonden dubbele records worden op het scherm weergegeven en weggeschreven in de logmap onder "2.2.4_EMJV_data_dubbele_records.csv". Dit is ter controle van de analist. Bij dubbele records wordt er niets weggegooid en inleesproces blijft doorlopen.

Outputbestanden:

- 2.2.4_EMJV_data_gecontroleerd.rds
- 2.2.4_EMJV_data_dubbele_records.csv

3.2.5 Beschikbaar stellen

[2.2.6]

Zodra het ontvangen EMJV bestand goed genoeg is bevonden om verder mee te werken en statistiek mee te maken (geen of zachte fouten) kunnen ze definitief opgeslagen worden in de database. Daarmee is effectief koppelpunt 2 vormgegeven.

Inputbestand: 2.2.4_EMJV_data_gecontroleerd.rds

Script: 2.2.5-Beschikbaar_stellen.R

De database (koppelvlak 2) wordt bijgewerkt met de nieuw ingelezen data.

Outputbestanden:

- Database: "tbl_Koppelvlak_2_gestandaardiseerde_data"
- 2.2.5_EMJV_data_beschikbaar_stellen.rds

4 Aanvullend

Naast de waarneming via de enquête zijn er aanvullende individuele dataverzamelingen een derde databron.

Er zijn 4-5 grote recyclingbedrijven die niet kunnen volstaan met de waarneming via de enquête. Zij verstrekken data per eigen bestand per mail (figuur 1). Deze bedrijven zijn vooraf bekend.

Het onderzoek bedrijfsafvalstoffen beoogt ook afvalstromen waar te nemen die door bedrijven niet altijd op voorhand onder afval worden geschaard (bijvoorbeeld omdat ze niet door een afvalbedrijf worden opgehaald). Ondanks dat de enquête ook hierin tracht te voorzien worden enige bedrijven aangeschreven middels email om langs deze weg aanvullende informatie op te halen bij bedrijven. De enquête van het CBS kan daar wel in voorzien maar wij kunnen aan het EMJV geen aanvullende vragen toevoegen.

Eenheden waarvoor nadere informatie wordt opgevraagd rond reststromen, veelal bijproducten die niet op voorhand altijd worden gemeld als afval.

4.1 Standaardiseren en leveringscontrole ‘aanvullende gegevens’ [3.2]

Inlezen gaat nu via Excel bestand waarin alle data wordt geplaatst. Dit is het koppelvak 1 bestand voor de aanvullende gegevens. Eventueel kan in een later stadium dit onderdeel aangepast worden naar bijvoorbeeld een R Shiny dashboard.

De output van dit proces is dezelfde tabel Error! Reference source not found. (zie LIM in hoofdstuk 7) die ook al de output vormde van de proces over de enquête- en de EMJV-data. Het doel is immers om alle data - van welke bron dan ook ontvangen – voor Koppelvak 2 in een standaard format te plaatsen.

Het gehele proces 3.2 wordt gedraaid via het opstarten van bestand “3.2-Standaardiseren_aanvullende_gegevens.cmd”.

Dit bestand voert alle afzonderlijke modules 3.2.1 t/m 3.2.5 in volgorde uit. Altijd wordt de volledige data van het verslagjaar in Koppelvak 2 opgeslagen. Data die al eerder in Koppelvak 2 was opgeslagen, blijft aanwezig, maar wordt aangepast met een einddatum zodat duidelijk is dat deze niet meer geldig is.

4.1.1 Gegevens ophalen uit netwerkmap [3.2.1]

Inputbestanden: Excel bestand met aanvullende gegevens

Script: 3.2.1-Ophalen_aanvullende_gegevens.R

Het script komt al met de meest waarschijnlijk naam van het Excel bestand met aanvullende data aan door te kijken welke bestanden in de \Import map beginnen met “3.3.1.” of “AANV”. Analist kan het overrulen met specifieke naam. Script leest dan het bestand in.

Outputbestand: 3.2.1_aanvullende_data_kv1.rds

4.1.2 Controle (technisch) [3.2.2]

Inputbestanden: 3.2.1_aanvullende_data_kv1.rds

Script: 3.2.2-Controle_technisch.R

Er vindt een technische controle plaats die kijkt of de volgende kolommen aanwezig zijn:

"Jaar", "CollectieEenheidId", "CollectieEenheidType", "Eural_code", "Omschrijving",
"Verwerking_code", "Hoeveelheid_ton", "DrogeStof_pct"

4.1.3 Transformeren

[3.2.3]

Inputbestanden:

- 3.2.1_aanvullende_data_kv1.rds
- Database: "tbl_Koppel_4_1_4_Verwerking_code_naam"
- Database: "tbl_Koppel_4_1_2_EURAL_EWC"

Script: 3.2.3-Transformeren.R

Omdat invullen in Excel vaak problemen geeft waarbij voorloophnullen verdwijnen, zal het script zelf zorgen dat deze terug worden geplaatst bij de velden CollectieEenheidId (12 posities zolang CollectieEenheidType gelijk is aan "CBS_V") en Eural_code (6 posities).

Het veld "Bron" krijgt de waarde "aanvullend" mee zodat duidelijk is waar deze records vandaan komen.

Outputbestand: 3.2.3_aanvullende_data_gettransformeerd.rds

4.1.4 Controle (logisch)

[3.2.4]

Inputbestanden:

- 3.2.3_aanvullende_data_gettransformeerd.rds
- Database: "tbl_Koppel_4_1_4_Verwerking_code_naam"
- Database: "tbl_Koppel_4_1_2_EURAL_EWC"

Script: 3.2.4-Controle_logisch.R

Er wordt gecheckt of er dubbele records gevonden worden op basis van de combinatie van de velden "Jaar", "CollectieEenheidId", "Eural_code", "Verwerking_code". Gevonden dubbele records worden op het scherm weergegeven en weggeschreven in de logmap onder "3.2.4_Aanvullende_data_dubbele_records.csv".

Verder controleert het of de ingevoerde Verwerking_codes en Eural_codes in onze koppeltabellen voorkomen.

Uiteindelijk zullen records met zogeheten harde fouten uit de data verwijderd worden en de rest wordt klaargezet om op te slaan in de database. Op dit moment zijn er geen controles die een harde fout kunnen opleveren.

Outputbestanden:

- 3.2.4_Aanvullende_data_dubbele_records.csv
- 3.2.4_Aanvullende_data_gecontroleerd.rds

4.1.5 Beschikbaar stellen

[3.2.5]

Inputbestanden: 3.2.4_Aanvullende_data_gecontroleerd.rds

Script: 3.2.5-Beschikbaar_stellen.R

De database (koppelvlak 2) wordt bijgewerkt met de nieuw ingelezen data. Zorg dat in het inputbestand alle aanvullende records van het verslagjaar behouden blijven. Anders wordt de niet meer opgenomen data als “verwijderd” gezien. In de database is het dan nog wel aanwezig, maar die regels hebben dan een Einddatum meegekregen die aangeeft dat ze niet meer geldig zijn.

Outputbestanden:

- Database: "tbl_Koppelvlak_2_gestandaardiseerde_data"
- 3.2.4_Aanvullende_data_dubbele_records.csv
- 3.2.4_Aanvullende_gegevens_beschikbaar_stellen.rds

5 Verwerken

Vanaf hier start de verwerking gericht op het maken van output (figuur 1 **zwembaan: verwerken**). In oneven jaren geldt dit proces voor de e-MJV data en de extra informatie, in even jaren geldt dit proces voor alle opgehaalde gegevens. De hier bedoelde “extra informatie” betreft dan (met hi index) opgehoogde gegevens van de enquête data uit een voorgaand jaar en de opgevraagde aanvullende gegevens.

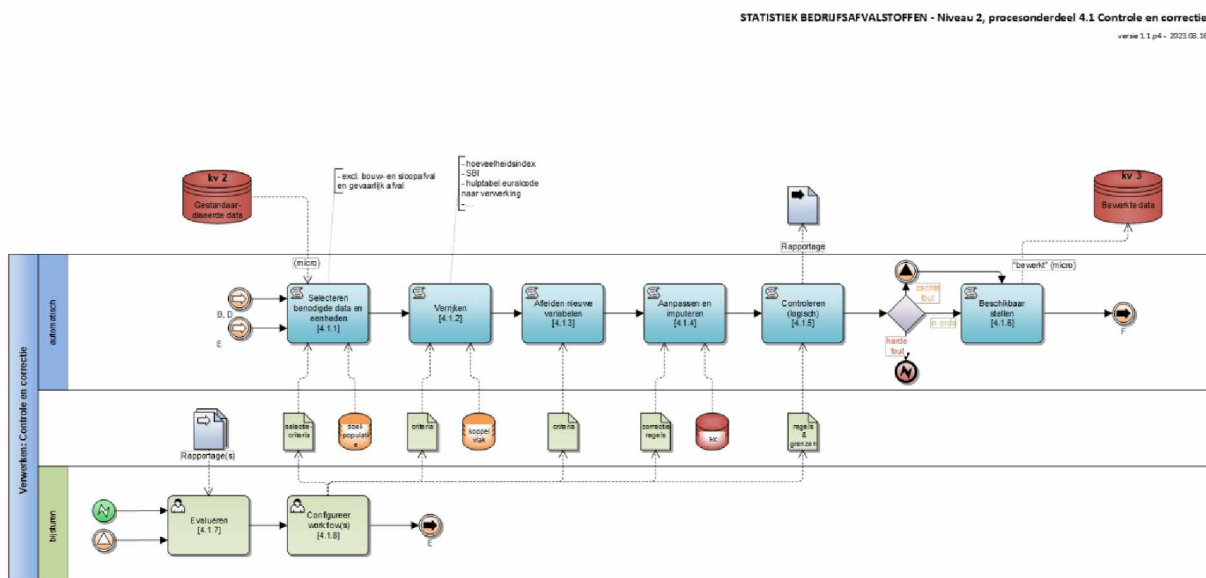
Voor de even jaren lopen de hierna volgende procedures tot en met koppelvlak 4 (= t/m hoofdstuk 6.2 **Schatten**) voor het eerst als het e-MJV register is. De enquête gegevens en de aanvullende gegevens zijn dan nog niet beschikbaar. Om vast te starten met rekenen heeft als voordeel dat grote meldingen die veel bijdragen aan de publicatiecijfers al vroeg in het jaar geverifieerd worden. Later wordt er weer gerekend tot en met koppelvlak 4 maar dan met alle data.

Let wel dat in de nu volgende procesonderdelen zo lang mogelijk de kleinste statistische bouwstenen gehandhaafd worden (KERS ontwerpkeuze ⑭). Het is dus expliciet niet de bedoeling direct te aggregeren naar EWC codes en dergelijke.

5.1 Controle en correctie

[4.1]

Controle en correctie is het eerste onderdeel van de outputgerichte verwerking. Het doel van dit deelproces is om de microdata klaar te maken voor (i) eigen schattingen en (ii) breder (her)gebruik. In figuur 4 is het proces in meer detail weergegeven. Het proces kan gedurende de looptijd van het verwerken meerdere keren doorlopen worden, bijvoorbeeld na het verhelpen van harde fouten of als er meer data is binnengekomen. Doorgaans zal het proces voor de eerste keer gaan lopen vanaf dat de e-MJV data beschikbaar is gesteld in koppelvlak 2 (zie D in [procesonderdeel 2.2]). Zowel de input als output van dit proces zijn microdata.



Figuur 4 Procesplaat Statistiek Bedrijfsafvalstoffen - niveau 2, procesonderdeel 4.1

Let op dat modules 4.1.1 en 4.1.2 omgewisseld zijn. Bovenstaand plaatje is daar niet op aangepast.

Het gehele proces 4.1 wordt gedraaid via het opstarten van bestand “4.1-Controlleren_en_corrigeren.cmd”.

Dit bestand voert alle afzonderlijke modules 4.1.1 t/m 4.1.6 in volgorde uit. Data van het verslagjaar die al eerder in Koppelvlak 3 was opgeslagen, wordt overschreven met de nieuwe data.

Output is de database tabel Error! Reference source not found. (zie LIM in hoofdstuk 7).

Dit koppelvak 3 bestand is de laatste met microdata. Merk op dat er geen ophoogfactoren in staan. Dat is een bewuste keuze van KERS. De ophoogfactoren worden uitgerekend en ingezet tussen koppelvak 3 en koppelvak 4 met als doel geaggregeerde koppelvak 4 data te maken. Een set ophoogfactoren hoort dan ook specifiek bij het aggregaat wat ermee wordt gemaakt. Het is expliciet niet de bedoeling de ophoogfactoren te delen met anderen. Als iemand dezelfde statistische behoefte heeft dan kan diegene het koppelpunt in koppelvak 4 gebruiken omdat dat bedoelt is om te delen met anderen.

Let op: GrondstoffenOmschrijving600 wordt nergens gebruikt bij de aggregaten of de publicaties. Waarom is deze variabele er?

Hadden we al eens besproken : (de opgehoogde (som van) de grondstoffen gaan mee daaR
(SOM)AFVAL INVOER == (SOM)AFVAL UITVOER + (SOM)GRONDSTOFFEN

En nr mogelijk zelfs eens een stofcode aan de afz genoeomde teruggewonnen grondstoffen kan hangen.

5.1.1 Verrijken

[4.1.2]

Het gaat hier om het toevoegen van hulpinformatie die je nodig hebt om je controles en/of correcties uit te voeren. Deze gegevens kunnen afkomstig zijn uit een combinatie van externe (primaire en secundaire waarneming) en/of interne bronnen en kunnen zowel uit de 'huidige' of eerdere verslagperiode(n) komen.

Inputbestanden:

- Database: "tbl_Koppelvak_2_gestandaardiseerde_data"
- Database: "tbl_Koppelvak_2_populatie"
- Database: "tbl_Koppelvak_2_uitzendbestand"
- Database: "tbl_Koppelvak_2_niet_afval_data"
- Database: "tbl_Koppelvak_3_bewerkte_data" (t-1 data)
- Database: "tbl_Koppelvak_3_EMJV_algemene_gegevens"
- Database: "tbl_Koppel_Bedrijfsklassen"
- Database: "tbl_Koppel_4_1_1_EURAL_EWC"
- Hoeveelheidsindex bestand
- 4.1.1_Standaard_verwerkingcodes.yaml

Script: 4.1.1-Verrijken.R

Aan de afvaldata "tbl_Koppelvak_2_gestandaardiseerde_data" wordt allereerst de BEID toegevoegd. Dat gaat via "tbl_Koppelvak_2_uitzendbestand". Deze BEID is van belang bij de CBS_V bedrijven. Vervolgens kunnen hiermee de SBI en GkSbs worden toegevoegd vanuit "tbl_Koppelvak_2_populatie".

Via de SBI worden dan Publicatiegroepen uit "tbl_Koppel_Bedrijfsklassen" toegevoegd.

Via de Eural_code wordt uit "tbl_Koppel_4_1_1_EURAL_EWC" EurGev en Eural_naam gekoppeld. EurGev wordt gebruikt om aan te geven of een euralcode gevaarlijk afval bevat of niet. Bij de EMJV data is dit al opgegeven. Dat wordt niet aangepast, dus het is alleen een toevoeging voor de ontvangen data via de enquête en via de aanvullende data.

De hoeveelheidsindex (HI index) wordt toegevoegd per SBI en Publicatiegroep_code. Let op dat dit nu nog slechts data uit een ingelezen Excel bestand bevat. Indien mogelijk zal het berekenen van deze cijfers in een later stadium in het systeem verwerkt gaan worden.

Via het regels bestand 4.1.1_Standaard_verwerkingcodes.yaml wordt bij (op dit moment ongeveer 40) Euralcodes een standaard verwerkingcode ingevuld. Dit wordt al in dit stadium gedaan en niet in stap 4.1.4 omdat bij de verrijking ook data van vorig jaar wordt toegevoegd welke onder andere op basis van de verwerkingcodes wordt gekoppeld. Bij de data van vorig jaar was de verwerkingcode al juist gezet. Dat moet dus voor dit jaar ook alvast, anders mis je die koppeling.

Het aantal Werkzame personen uit de respons wordt toegevoegd via "tbl_Koppelvlak_2_niet_afval_data". Het is niet nodig het aantal werkzame personen (WP) uit het ABR hier toe te voegen. WP wordt voor controle op duizendfouten gebruikt maar deze controle wordt alleen uitgevoerd voor berichtgevers uit de enquête en daarvoor volstaat WP die berichtgevers in de enquête hebben ingevuld. Deze WP bevat namelijk het aantal werkzame personen van het bedrijfsonderdeel die de berichtgever in gedachten had bij het invullen van de vragenlijst. Dit kan ook een vestiging zijn terwijl in het ABR alleen het aantal WP van de gehele bijbehorende BEID staat.

Uit "tbl_Koppelvlak_3_bewerkte_data" wordt afvaldata van het vorige verslagjaar per CollectieEenheidId, Eural_code en Verwerking_code toegevoegd. Deze worden in module 5.1.4 gebruikt voor de imputatie van missende waarden.

De NIC nummers vormen de bedrijfsidentificatie in het EMJV bestand. In proces 0.0.3.6 zijn de bijbehorende BEID's al toegevoegd aan de algemene gegevens: "tbl_Koppelvlak_3_EMJV_algemene_gegevens". Deze worden nu ook aan de afval data van EMJV bedrijven toegevoegd.

Er zijn records uit het EMJV waarbij de codering Eural_gevaarlijk anders is dan de standaard codering die wij hanteren bij de betreffende Eural_code. De opgave uit het EMJV wordt aangehouden. Een overzicht van deze records wordt weergegeven op het scherm en ook weggeschreven in "4.1.1_EMJV_Eural_gevaarlijk_anders.csv" (misschien kan dit beter in module 4.1.5. plaats vinden?).

Bij afvalregels waarbij de Verwerking_code gelijk is aan 600 is via module 1.1.3. de grondstof naam en grondstof bestemming in het Omschrijving veld geplaatst. Het script maakt hier het veld "GrondstoffenOmschrijving600" aan om de grondstoffenomschrijving over te nemen uit het veld Omschrijving.

Outputbestanden:

- 4.1.1_Verrijken.rds
- 4.1.1_EMJV_Eural_gevaarlijk_anders.csv

5.1.2 Selecteren benodigde data en eenheden

[4.1.1]

Uit koppelvlaak 2 Gestandaardiseerde data worden alleen de benodigde data en eenheden gehaald die nodig zijn voor het maken van de gewenste statistische output. Zo zitten bijvoorbeeld de

ziekenhuizen in de e-MJV terwijl de statistieken van bedrijfsafvalstoffen niet gaan over deze bedrijfseenheden. Deze vallen weg bij de selectie. Alleen de eenheden die aanwezig zijn in de doelpopulatie worden aangekoppeld.

Inputbestanden:

- 4.1.1_Verrijken.rds
- 4.1.2-Selecties_benodigde_data_instellingen.yaml
- Database: "tbl_Koppelvlak_2_populatie"
- Database: "tbl_Koppelvlak_2_uitzendbestand"

Script: 4.1.2-Selecteren_benodigde_data_en_eenheden.R

Het script leest alle data van het verslagjaar uit Koppelvlak 2 in en verwijdert allereerst data van bedrijven die niet in de populatie zitten. BEID's kunnen direct in de populatie worden opgezocht. CBS_V nummers worden gecontroleerd op aanwezigheid in het uitzendbestand. Data die verwijderd wordt, is ter controle in te zien in de logmap in het bestand "4.1.2_VerwijderdeRecords_niet_in_populatie.csv".

Er zijn ook afvalstoffen voor dit onderzoek uitgezonderd, te weten:

1. radioactief afval,
2. gevaarlijk afval,
3. bouw- en sloopafval (zijn alle euralcodes die beginnen met 17 excl. 170504),
4. verbrandingsresten uit eigen afvalverbranding en
5. stoffen die geloosd worden met afvalwater of worden uitgestoten in de lucht.

De selectieregels van niet mee te nemen stoffen zijn op te stellen op basis van euralcodes welke kunnen worden ingevoerd in het bestand "4.1.2-Selecties_benodigde_data_instellingen.yaml" in de \Instellingen map. Dit is de invulling van regelgestuurd werken (KERS ontwerpkeuze 8).

Dit zijn de variabelen namen waarmee de regels opgebouwd kunnen worden:

"Jaar", "CollectieEenheidId", "CollectieEenheidType", "Eural_code", "Verwerking_code", "Publicatiegroep_code", "SBI", "BEID", "RowID", "Omschrijving", "Verwerking_naam", "Hoeveelheid_ton", "DrogeStof_pct", "Bron", "GkSbs", "Publicatiegroep_omschrijving", "Pub_SL", "SL_Code", "SL_omschrijving", "SL_Code2", "Eural_gevaarlijk", "Eural_naam", "EWC_code", "EWC_omschrijving", "HoeveelheidsIndex", "WP", "Hoeveelheid_ton_tmin1", "GrondstoffenOmschrijving600".

Op dit moment komen de selecties in de volgende regels tot uiting:

```
rules:
-
  expr: (str_detect(Eural_code, "^17") & Eural_code != "170504") & !str_detect(Verwerking_code, "^21") & !str_detect(SBI, "^383")
  name: nietmeenemen01
  label: bouw- en sloopafval
  description: Bouw- en sloopafval (zijn alle euralcodes die beginnen met 17 excl. 170504) met uitzondering van interne verwerking (21*) en VTR bedrijven (SBI 383*).
-
```

```

expr: Eural_code %in% c("150110")

name: nietmeenemen02

label: overige euralcodes niet meenemen

description: Euralcodes welke niet worden meegenomen in verdere proces (radioactief afval, gevaarlijk afval,
verbrandingsresten uit eigen afvalverbranding en stoffen die geloosd worden met afvalwater of worden uitgestoten in
de lucht).

```

Regel nietmeenemen01 is al juist voor de huidige situatie en hoeft alleen aangepast te worden als er door veranderde regelgeving andere selecties aan bouw- en sloopafval nodig zijn. Het ^-teken in de regel betekent hier “begint met”. Er wordt hier dus bijvoorbeeld gekeken naar Eural codes die beginnen met “17”. Het !-teken betekend niet. De ! voor !str_detect(Verwerking_code, "^21") betekent dus “zoek Verwerking_code die NIET beginnen met 21”. De == betekent is gelijk aan, en != betekent is NIET gelijk aan.

Een deel van de Gevaarlijke afvalstoffen wordt hier verwijderd, maar ze kunnen bij interne verwerking nog wel aanwezig zijn.

Indien nodig kunnen ook extra regels worden toegevoegd. De labels gebruiken we nog niet, maar kunnen we later verwerken tot informatie in de log.

Het script neemt data over de afvalsoorten die opgegeven zijn in het “4.1.2-Selecties_benodigde_data_instellingen.yaml” bestand niet verder mee naar koppelvlaak 3.

Records die verwijderd zijn, worden ter controle in de logmap opgeslagen in het bestand “4.1.2_VerwijderdeRecords_afvalsoorten.csv”.

Als bij opgaves van bedrijven meerdere records met dezelfde Eural_code en Verwerking_code zijn opgegeven zijn ze al naar voren gekomen bij de logische controles in modules 1.1.4 en 2.2.5. Deze records kunnen dan behandeld worden in de bestanden met regels (yaml bestanden). Indien er (nog) niks aan gedaan is worden ze hier samengeteld tot 1 regel. Dit houdt er rekening mee dat de hoeveelheid drogestof in deze fase van het proces nog als een percentage in de database staat.

Outputbestanden:

- 4.1.2_Selecteren_benodigde_data_en_eenheden.rds
- 4.1.2_VerwijderdeRecords_niet_in_populatie.csv
- 4.1.2_VerwijderdeRecords_afvalsoorten.csv

5.1.3 Afleiden nieuwe variabelen

[4.1.3]

We bedoelen hier statistisch inhoudelijke afleidingen. Het aantal ton afval per werkzame persoon is zo’n afgeleide variabele. Als je de geproduceerde hoeveelheid afval, en het aantal werkzame personen hebt, dan kun je daar de hoeveelheid afval per werkzame persoon uit afleiden. Deze afleiding wordt gebruikt om te controleren of een opgave plausibel is.

De ‘Marker’ is geen afgeleide variabele. Dat is hulpinformatie voor het statistisch proces.

Inputbestand: 4.1.2_Verrijken.rds

Script: 4.1.3-Afleiden_nieuwe_variabelen.R

Dit script doet niets meer. De inhoud is verplaatst naar 4.1.4.

Outputbestand: 4.1.3_Afleiden_nieuwe_variabelen.rds

5.1.4 Aanpassen en imputeren

[4.1.4]

Deze activiteit bestaat uit twee deelstappen: (i) automatisch (!) opsporen van eenheden waar ingrijpen nodig is en (ii) doorvoeren van de correctie. Dit gebeurt in een op regels gebaseerde aanpak. In figuur 4 zie je de regels weergegeven die in het proces binnenkomen.

Merk op dat een imputatie een (nieuwe) waarde moet opleveren op een plaats waar een waarde ontbreekt of 'onjuist' is, én dat deze geïmputeerde waarde een betere schatting vormt dan de waarde die er stond. De betrouwbaarheid van het nieuwe cijfer moet dus groot zijn. Anders gezegd betekent dit, dat aanpassen en imputeren ten behoeve van schattingen hier niet behoort plaats te vinden. Als men bijvoorbeeld enkel imputeert om later (eenvoudig) te kunnen aggregeren dan moet dat gebeuren bij de stap **Schatten**.

Belangrijk is ook dat records worden gemarkeerd bij het toevoegen / wijzigen / verwijderen van gegevenswaarden. De bedrijfsafvalstoffenstatistiek doet dit al met het toekennen van een marker.

Inputbestanden:

- 4.1.3_Afleiden_nieuwe_variabelen.rds
- 4.1.4_Correctieregels_algemeen.R
- 4.1.4_Correctieregels_specifiek.csv
- Database: "tbl_Koppel_4_1_4_Verwerking_code_naam"

Script: 4.1.4-Aanpassen_en_imputeren.R

De aanpassingen in imputaties gaan volgens de regels in de yaml bestanden. Ze zijn in twee groepen gesplitst. Bij de algemene correctieregels (4.1.4_Correctieregels_algemeen.R in de instellingen \Correctieregels_algemeen map) zet de analist de automatische correcties welke in principe elk jaar opnieuw uitgevoerd moeten worden tenzij er betere correctieregels bedacht worden.

Bij de specifieke correctieregels (4.1.4_Correctieregels_specifiek.csv in de instellingen \Correctieregels_specifiek map) kan een analist specifieke wijzigingen voor een bepaald bedrijf doorvoeren die niet met algemene regels gevonden of gecorrigeerd kunnen worden. Zie verder toelichting vanuit KERS bij proces 4.1.8. Configureer workflows.

Dit zijn de variabelen namen waarmee de algemene en specifieke correctieregels opgebouwd kunnen worden:

"CollectieEenheidId", "Eural_code", "Verwerking_code", "Jaar", "Publicatiegroep_code", "SBI", "BEID", "CollectieEenheidType", "Eural_gevaarlijk", "Verwerking_naam", "Bron", "Hoeveelheid_ton", "RowID", "Omschrijving", "DrogeStof_pct", "GkSbs", "Publicatiegroep_omschrijving", "Pub_SL", "SL_Code", "SL_omschrijving", "SL_Code2", "Eural_naam", "EWC_code", "EWC_omschrijving", "HoeveelheidsIndex", "WP", "Hoeveelheid_ton_tmin1", "GrondstoffenOmschrijving600", "afval_som_sbi", "WP_respons_som_sbi", "duizendfout_grens", "afval_som", "afval_som_per_WP")

Alle bovengenoemde regels kunnen ook in hetzelfde bestand worden opgenomen, maar door de opsplitsing in algemeen en specifiek, is sneller duidelijk om wat voor type regels het gaat. Het verslagjaar moet altijd meegegeven worden aan de regels met de specifieke wijzigingen, zodat

uitkomsten uit het verleden altijd te reproduceren zijn en het geeft ook een goed overzicht van de correcties die bij bedrijven in het verleden gedaan moesten worden. Bij de algemene regels is het waarschijnlijk ook het handigste om het jaartal op te nemen aangezien deze ook aan evolutie onderhevig kunnen zijn.

Via de database tabel “tbl_Koppel_4_1_4_Verwerking_code_naam” wordt aan de Verwerking_code de Verwerking_naam gekoppeld. Verwerking_naam was al ingevuld bij de EMJV bedrijven, maar die wordt hier overschreven zodat bij elke Verwerking_code slechts één Verwerking_naam kan staan. Het origineel dat via het EMJV was ingevuld is altijd terug te vinden in koppelvak 2. In koppelvak 3 zijn de namen die horen bij de verwerking_codes daarmee gestructureerd in vaste waarden in plaats van in alle verschillende omschrijvingen die door berichtgevers opgegeven waren.

Correcties voor duizendfouten worden allereerst gedaan door een grens voor duizendfouten te berekenen. Deze duizendfout_grens wordt berekend als “som afval in de SBI” / “som wp (respons) in de SBI”.

Vervolgens wordt Afval totaal per CollectieEenheidld uitgerekend, om voor het opsporen van duizendfouten te kunnen vergelijken met de duizendfout_grens en “som afval in de SBI”.

Er is nog een tweede methode om duizendfouten te markeren, die het script zelf uitvoert. Deze duizendfouten worden gelokaliseerd door te kijken of de gemiddelde hoeveelheid van een stofsoort per werkzame persoon binnen 300 maal van de gestelde grens ligt. Vervolgens worden de waarden van de via beide methoden gemarkeerde duizendfouten gecorrigeerd door de hoeveelheid te delen door 1000. Bij de gecorrigeerde waarden krijgt het Marker veld de waarde "dzd fout aangepast" mee.

Hierna worden missende waarden gecorrigeerd.

Als er missings gevonden zijn (volgens bovenstaande definitie) dan worden die gecorrigeerd met de opgegeven waarde uit (t-1) * HI index. Het Marker veld krijgt in deze gevallen de waarde "imputatie" mee.

Outputbestand: 4.1.4_Aanpassen_en_Imputeren.rds

5.1.5 Controle (logisch)

[4.1.5]

T.b.v. [4.1.4] worden altijd eerst een paar controles uitgevoerd, je wilt de eenheden namelijk opsporen waar ingrijpen nodig is. Achteraf wil je echter wel nog eens nagaan of nu wel aan alle logische controles voldaan wordt. Vandaar dat dit een aparte expliciete stap is.

Inputbestand:

- 4.1.4_Aanpassen_en_Imputeren.rds
- Database: "tbl_Koppelvak_2_populatie"

Script: 4.1.5-Controle_Logisch.R

Er worden een aantal controles op de data gedaan. Dit omvat test op dubbele records en of bedrijven in de populatie zitten. Ook tests op duizendfouten. Het resultaat hiervan wordt weggeschreven in Controle_KV_data.html in de logmap.

Outputbestand:

- 4.1.5_Controle_logisch.rds
- Controle_KV_data.html

5.1.6 Beschikbaar stellen

[4.1.6]

Inputbestand: 4.1.5_Controle_logisch.rds

Script: 4.1.6-Beschikbaar_stellen.R

De database (koppelvlak 3) wordt bijgewerkt met de nieuw ingelezen data. Eventueel nog iets bouwen voor de records met fouten via module 4.1.7? Analist kan fouten corrigeren in module 4.1.4 en vervolgens heel het Verwerken proces opnieuw draaien.

Outputbestanden:

- Database: "tbl_Koppelvlak_3_bewerkte_data"
- 4.1.6-Beschikbaar_stellen.rds

5.1.7 Evalueren

[4.1.7]

Dit stuk is nog niet gebouwd. Later kijken wat de wensen van analist op dit vlak zijn.

Wanneer er harde en/of zachte fouten uit de logische controle komen volgt dit deelproces. Als basis voor de evaluatie wordt het rapport gebruikt dat automatisch gegenereerd is bij de controle. Deze bevat niet enkel een overzicht van de gebruikte indicatoren en welke tot fouten hebben geleid, maar bijvoorbeeld ook tellingen, grafieken en diagrammen. Door het rapport te gebruiken in samenhang met de beschikbare data in de koppelvlakken en eventuele lokale rustpunten kan worden achterhaald wat de oorzaak is van de fout en wat de gewenste vervolgactie. . Bijvoorbeeld kan de e-MJV-tabel "toelichtingen oordelen en adviezen" uit **hoofdstuk 4.1 Verzamelen gegevens e-MJV**, verklaringen geven voor de ontwikkelingen in afvalstromen. Overigens kan op basis hiervan dus ook aangepast worden wat er in de rapporten wordt gegenereerd om zo bijvoorbeeld meer informatie te verkrijgen voor de toekomst.

T.b.v. evalueren kun je ook kijken naar groepsgemiddelden e.d. Hier kan ook eventueel een top-down tool ingezet worden zoals team Generiek die ontwikkeld heeft. Belangrijk is dat het een hulpmiddel is en geen onderdeel van het reguliere productieproces.

5.1.8 Configureer workflow(s)

[4.1.8]

Dit is geen aparte module, maar een beschrijving van hoe gevonden problemen in regels moeten worden verwerkt. De volgende alinea's beschrijven dit proces.

Hierin zit de wendbaarheid van het KERS proces. Nadat op basis van het proces Evalueren de oorzaak van een fout is geïdentificeerd, of op basis van het proces Evalueren & innoveren een vernieuwing c.q. verbetering geïdentificeerd is die op korte termijn door te voeren is, moet de oplossing vastgelegd worden door middel van het configureren van workflows. Dit houdt in dat zo vroeg mogelijk in het verwerkingsproces de regelsturing zodanig wordt aangepast dat de opgemerkte fouten en verbeteringen naar wens afgehandeld worden. De correcties zijn enkel mogelijk op basis van het wijzigen, toevoegen of verwijderen van regels, parameters, criteria en grenzen (ontwerpkeuze 10: regelsturing aanpassen). Daarmee blijft het proces transparant, beheersbaar en

reproduceerbaar, en worden lessen altijd teruggelegd in de standaard verwerking. Het is mogelijk dat een gevonden fout een wijziging op meerdere plaatsen noodzakelijk maakt. Bijvoorbeeld door een bepaalde fout wel door Controleren (logisch) te laten, maar deze fout pas aan te passen door middel van een correctieregel bij Aanpassen & imputeren.

Dit betekent dus ook dat handmatige correcties, zoals nu in het proces van Bedrijfsafvalstoffen gemaakt worden, niet wenselijk zijn. Hoewel het lof verdient dat het ingrijpen in het huidige proces transparant is gemaakt. In theorie kan voor een specifieke eenheid, voor specifieke omstandigheden de aanpassing worden vastgelegd als regel die maar voor één moment geldig is. In de praktijk zorgt dit dan voor een reproduceerbaar en transparant proces. Maar dit is op de lange duur niet beheersbaar en biedt bovendien geen structurele oplossing. Indien mogelijk zou je zo'n wijziging dus liever in automatische correctieregel willen opschrijven die voor elke eenheid van toepassing is en niet alleen voor de 30 uitschieters waar nu naar gekeken wordt.

Bestaande controleregels zijn hier van belang om goed te kunnen bewaken dat met deze aanpassingen nog steeds de gewenste kwaliteit wordt behaald. Zodra nieuwe problemen ontstaan heeft dit dus doorgaans ook aanpassingen aan controleregels.

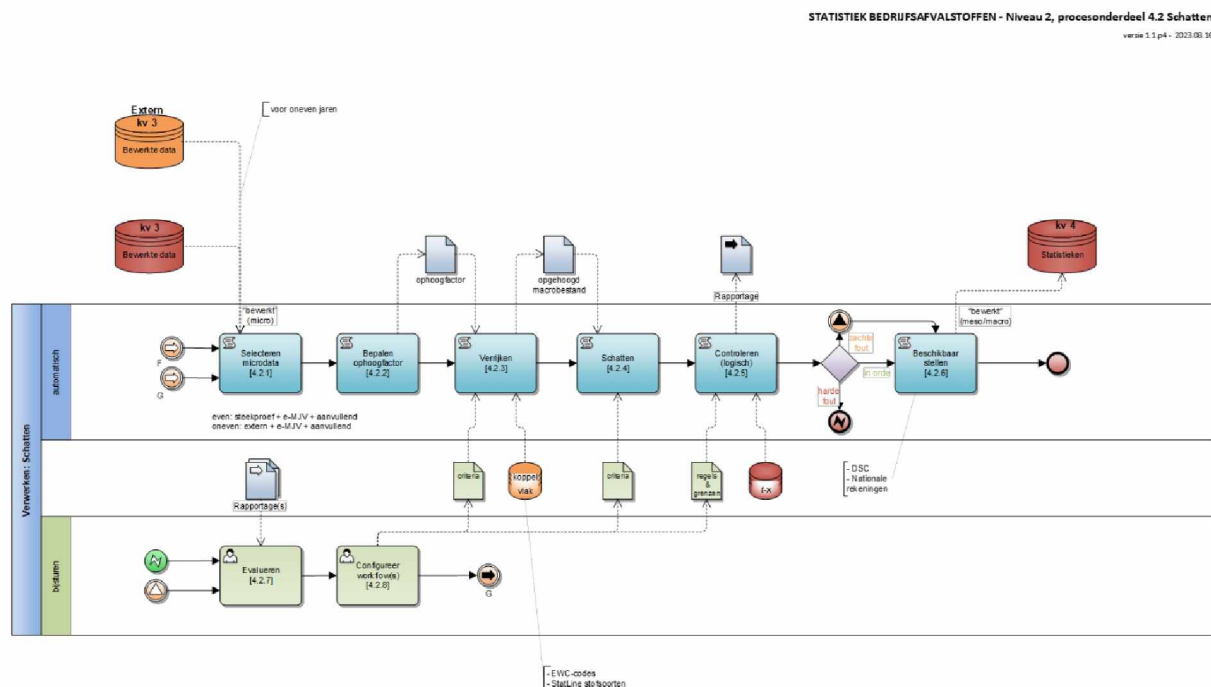
Dit proces kan samen met het proces Evalueren gezien worden als de basis van het reguliere productieproces. Het zorgt er voor dat tijdens het productieproces het werk gericht is op die elementen waar het meeste effect kan worden behaald en dat dit snel en iteratief kan plaats vinden.

Na het configureren van de workflow(s) start het automatische proces weer van voor af aan (letter E in figuur 4).

5.2 Schatten

[4.2]

Schatten is het tweede onderdeel van de outputgerichte verwerking (figuur 1). In figuur 5 is het proces in detail weergegeven. In dit proces wordt de overstap gemaakt van gegevens op microniveau naar gegevens op meso-/macroniveau. Het doel van dit deelproces is cijfers maken voor publicatie. Door gebruik te maken van schattingsmethoden kunnen, op geaggregeerd niveau, publiceerbare cijfers berekend worden. Vanaf dit proces komen ook aggregaten van afvalsoorten om de hoek kijken.



Figuur 5 Procesplaat Statistiek Bedrijfsafvalstoffen - niveau 2, procesonderdeel 4.2

In de gegeven procesonderdelen is het mogelijk gelijktijdig cijfers voor verschillende gewenste aggregaten te maken of na elkaar. Het proces kan namelijk meerdere keren doorlopen worden in meerdere iteraties. In de eerste iteratie is bijvoorbeeld alleen nog maar data van het e-MJV beschikbaar. En omdat aan de verschillende publicatieverplichtingen niet allemaal gelijke deadlines hangen is dit ook wenselijk. Daarbij wil je het proces doorlopen met de meest recente beschikbare data (brondata en hulpinfo) van de hoogste kwaliteit. Ook als bijvoorbeeld voor maatwerk en aanvullende statistische diensten andere aggregaten nodig zijn dan loopt dit proces opnieuw.

Het voornaamste verschil tussen alle publicaties zit hem in de gebruikte labels cq. omschrijvingen op koppelvlaak 5. Zo'n omschrijving is iets dat los staat van de data zelf. Wanneer je een SBI-aggregaat publiceert op StatLine dan moet daar de omschrijving aangehangen worden uit de codevijver. T.b.v. Eurostat krijgt precies datzelfde SBI-aggregaat (dat dus maar één keer op koppelvlaak 4 aanwezig is) een andere omschrijving (label) mee. Het aggregaat zelf hoeft dus geen labels te bevatten. Afhankelijk van voor wie je een publicatie maakt voeg je daar de juiste labels aan toe in het proces tussen koppelvlaak 4 en 5.

Koppelvlaak 4 bevat alle aggregaten die je nodig hebt voor het maken van je publicaties (koppelvlaak 5), en misschien nog wel wat meer. Als er tussen de publicaties t.b.v. Eurostat, StatLine etc. een grote mate van overlap zit, dan worden de aggregaten maar één keer uitgerekend en komen dus ook maar één keer voor op koppelvlaak 4. Op koppelvlaak 5 komen ze wel vaker voor, namelijk in elke publicatie apart. **Bij ons worden er geen aggregaten hergebruikt voor meerdere publicaties, dus voor elke**

publicatie zijn er aparte aggregaten. Bij de CLO publicatie worden andere aggregaten gebruikt als voor StatLine omdat bij het CLO de euralcodes "170504", "170506", "010412" niet meegenomen worden. Per saldo publiceren we dus ook de som van die 3 euralcodes. Wellicht heeft dit goede redenen, maar anders overwegen waard om over zelfde groepen te gaan publiceren?

Dit mag ik aanpassen! CLO gelijktrekken aan StatLine!

Het gehele proces 4.2 wordt gedraaid via het opstarten van bestand 4.2-Schatten.cmd

Dit bestand voert alle afzonderlijke modules 4.2.1 t/m 4.2.6 in volgorde uit. Data van het verslagjaar die al eerder in Koppelvlak 4 was opgeslagen, wordt overschreven met de nieuwe data.

Output is de database tabel Error! Reference source not found. (zie LIM in hoofdstuk 7).

5.2.1 Selecteren microdata

[4.2.1]

Inputbestand: Database: "tbl_Koppelvlak_3_bewerkte_data"

Script: 4.2.1-Selecteren_microdata.R

Dit script doet niets anders dan de data uit de database inlezen.

Outputbestand: 4.2.1-Selecteren_microdata.rds

5.2.2 Bepalen ophoogfactor

[4.2.2]

Dit onderdeel gebeurt op basis van de beschikbare brondata en kent gewichten toe aan de responderende bedrijven met als doel in onderdeel [4.2.4] totaalschattingen voor de populatie te maken.

Inputbestanden:

- 4.2.1-Selecteren_microdata.rds
- Database: "tbl_Koppelvlak_3_populatie"
- Database: "tbl_Koppelvlak_3_uitzendbestand"
- Database: "tbl_Koppelvlak_2_niet_afval_data"
- Database: "tbl_Koppel_Bedrijfsklassen"
- 4.2.2_Ophoogfactor1.yaml

Script: 4.2.2-Bepalen_ophoogfactor.R

De ophoging gaat via de classificaties van SBI_stratum. Voorheen werd ook de GK_stratum gebruikt, maar uit onderzoek van Methodologie bleek dat het aantal werkzame personen per persoon niet verschilt per grootteklasse. Daarom hoeft deze ook niet meer gebruikt te worden bij het ophogen.

SBI_stratum is niet gebruikt voor het maken van de steekproef, maar is een classificatie die gebruikt wordt voor de publicaties. Het SBI_stratum komt uit het veld Publicatiegroep_code in "tbl_Koppel_Bedrijfsklassen".

Het Populatiebestand wordt verrijkt met het SBI_stratum. Vervolgens worden de variabelen "CollectieEenheidId", en "CollectieEenheidType" via het uitzendbestand gekoppeld. Daarmee kan vervolgens de niet_afval_data worden gekoppeld waarin staat of een berichtgever via de enquête heeft gerespondeerd of niet. Met deze informatie worden per stratum ophoogfactoren uitgerekend

via de methodiek uitgedacht door ^{5.1.2.e} van Bedrijvensteekproeven. Hierbij vindt ophoging plaats via het aantal Werkzame Personen uit het ABR (dus niet wat op het enquête formulier is opgegeven).

In die methodiek is echter geen rekening gehouden met CBS_V bedrijven en binnenkomst via het EMJV. Dit is opgelost door bij CBS_V bedrijven het aantal werkzame personen van de BEID te delen door het aantal CBS_V eenheden die bij de BEID horen. Zo kan er maximaal tot het totaal van de BEID worden opgehoogd.

Als een bedrijf heeft gerespondeerd via het EMJV wordt het aantal werkzame personen (voor de ophoogmethodiek) op 0 gezet, zodat bedrijven uit de enquête niet worden opgehoogd voor een bedrijf dat al ontvangen is via het EMJV. Bedrijven die respondeerden via het EMJV krijgen altijd ophoogfactor 1.

Ook kan er in het bestand 4.2.2_Ophoogfactor1.yaml een lijst CollectieEenheidId's worden opgegeven van bedrijven uit andere bronnen die ook ophoogfactor 1 moeten krijgen. Deze worden op dezelfde manier behandeld als bedrijven die via het EMJV zijn ontvangen. Dus er wordt niet naartoe opgehoogd door het aantal werkzame personen op 0 te zetten, en de ophoogfactor van deze bedrijven wordt op 1 gezet.

Nu kan het feitelijke ophogen plaatsvinden. Dit gaat als volgt:

Hoeveelheid_ton_opgehoogd = Hoeveelheid_ton * Ophoogfactor

De ophoogfactoren mogen volgens KERS-principes niet in de koppelvlakken opgenomen worden. Daarom worden de ophoogfactoren per Collectie-eenheid apart opgeslagen in de database in tabel "tbl_Ophoogfactoren". In de log wordt ook een bestand met ophoogfactoren per stratum geplaatst (4.2.2-ophoogfactoren.csv).

Outputbestanden:

- Database: "tbl_Ophoogfactoren"
- 4.2.2-ophoogfactoren.csv
- 4.2.2-Bepalen_ophoogfactor.rds

5.2.3 Verrijken

[4.2.3]

Het gaat hier om het toevoegen van hulpinformatie die je nodig hebt voor het schatten en om te kunnen controleren. Classificaties die je toevoegt om over te kunnen publiceren vallen hier ook onder. Voor dit 'verrijken' van de data mag data uit alle koppelvlakken genomen worden, al hebben data uit koppelvlak (3) Bewerkte data of (4) Statistieken (bijvoorbeeld als randtotalen) de voorkeur omdat dit het minste extra werk oplevert en de kans op dubbel werk verkleint. Als namelijk data uit een eerder koppelvlak worden gebruikt zullen deze eerst geschikt gemaakt moeten worden om mee te schatten volgens het 'Prepareren data voor eigen schattingen' proces.

Inputbestanden:

- 4.2.2-Bepalen_ophoogfactor.rds
- Database: "tbl_Koppel_Bedrijfsklassen"
- Database: "tbl_Koppelvlak_2_RWSLO"

Script: 4.2.3-Verrijken.R

In dit script worden publicatiegroepen uit "tbl_Koppel_Bedrijfsklassen" aan de afvaldata ("4.2.2-Bepalen_ophoogfactor.rds") gekoppeld op basis van de SBI.

Daarna worden de categorieën van verwerkingsmethodes verder ingedikt tot 5 hoofdgroepen.

Verder worden diverse afvalsoorten samengeteld. Bijvoorbeeld het ijzerhoudend metaal, niet-ijzerhoudend metaal en gemixt metaal, worden samengevoegd tot "Metaal".

Ook worden sommaties gemaakt naar verwerkingsmethodes onafhankelijk van welk type afval er verwerkt is.

Daarna wordt data over SBI 37, 38.1, 38.2 en 39 toevoegen. Dit is afkomstig van RWO/LO en is klaargezet in "tbl_Koppelvlak_2_RWSLO". Dat koppelvlak is al klaargezet in module "0.0.4 - RWSLO data invoeren".

Outputbestand: 4.2.3-Verrijken.rds

5.2.4 Schatten

[4.2.4]

Dit onderdeel maakt, zoals in paragraaf 6.2.2 genoemd, totaalschattingen van populatieparameters. Zo is een doel voor de doelpopulatie de omvang van de totale hoeveelheid afgevoerd afval te schatten. Het komt hier neer op gewogen optellen van je respons naar de gewenste aggregaten. In [4.2.2] zijn ophoogfactoren toegekend aan de responderende bedrijven. Schattingen van populatieparameters worden verkregen door de gegevens met deze ophoogfactoren te vermenigvuldigen en vervolgens op te tellen. In de eerste iteratie betekent dit dat de schattingen op basis van alleen e-MJV input zijn aangezien gegevens van de enquête dan nog niet binnen zijn. Elke iteratie levert zo andere ophoogfactoren op.

Inputbestanden:

- 4.2.3-Verrijken.rds
- Database: "tbl_Koppel_Bedrijfsklassen"

Script: 4.2.4-Schatten.R

Dit script maakt de nodige aggregaten. CLO tabellen gaan nu gebruik maken van dezelfde aggregaten als Statline. Dat was in het verleden niet zo toen euralcodes "170504", "170506" en "010412" voor CLO werden uitgesloten, maar dat is nu niet meer nodig. Ook Milieurekeningen krijgt een tabel gebaseerd op StatLine output, maar dan alleen wat betreft interne verwerking.

Er worden aggregaten gemaakt voor:

- StatLine Bedrijfsafval tabel
- StatLine Voorbereiding tot recycling (VTR) tabel

Sommige publicaties berekenen bepaalde combinaties van afvalstoffen. In deze module worden alle combinaties van afvalstoffen voor alle aggregaten berekend. In proces "4.3 Publiceren", zal alleen hetgeen wat nodig is er uit worden geselecteerd.

Outputbestand: 4.2.4-Schatten.rds

5.2.5 Controle (logisch)

[4.2.5]

Deze module moet nog gemaakt worden.

Een voorbeeld van een mogelijke controle is:

- Zijn de schattingen voor jaar t van dezelfde orde van grootte als de schattingen voor jaar $t-1$
- Is de afvalindex(AI) binnen bepaalde grenzen van de hoeveelheidsindex (HI)?
- Klopt het randtotaal met de som van de afzonderlijke groepen, bijvoorbeeld sbi's?

Inputbestanden: 4.2.4-Schatten.rds

Script: 4.2.5-Controle_logisch.R

Outputbestand: 4.2.5_Controle_logisch.rds

5.2.6 Beschikbaar stellen

[4.2.6]

Inputbestanden: 4.2.5_Controle_logisch.rds

Script: 4.2.5-Controle_logisch.R

De database (koppelvlak 4) wordt bijgewerkt met de nieuw ingelezen data.

Outputbestanden:

- Database: "tbl_Koppelvlak_4_statistieken"
- 4.2.6-Beschikbaar_stellen.rds

5.3 Publiceren

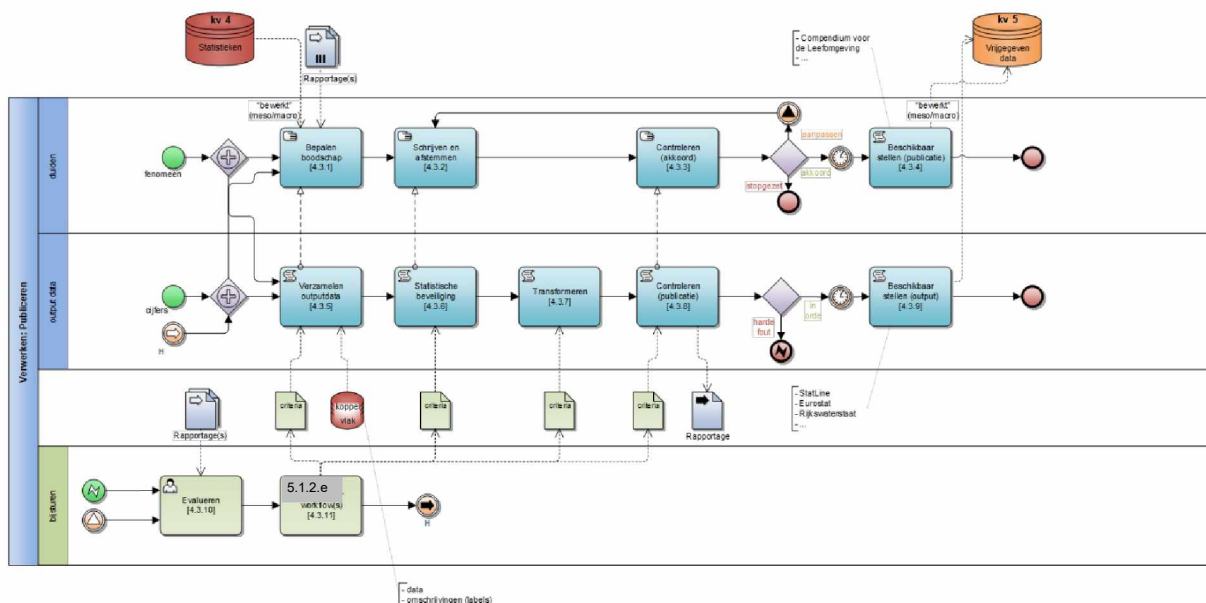
[4.3]

Publiceren is het derde en tevens slotonderdeel van de outputgerichte verwerking van bedrijfsafvalstoffen (figuur 1). Het heeft als doel de kennis over de omvang en ontwikkeling van bedrijfsafvalstoffen tot maatschappelijke waarde te brengen en die kennis ook daadwerkelijk te verspreiden; de kernactiviteit van het CBS.

Bedrijfsafvalstoffen maakt cijfers voor:

- Leveringen aan Nationale rekeningen
- StatLinetabel jaar
- Levering Rijkswaterstaat voor totaal tabel aan Eurostat
- Compendium voor de Leefomgeving
- StatLine alle jaren

Figuur 6 geeft het deelproces meer in detail.



Figuur 6 Procesplaat Statistiek Bedrijfsafvalstoffen - niveau 2, procesonderdeel 4.3

De twee onderdelen van het ontsluiten van de publiceerbare gegevens zijn daarin terug te zien door middel van twee zwembanen. De bovenste zwembaan geeft het proces van het duiden van de data; het betekenis geven aan de data door ze in samenhang te brengen. Denk daarbij aan artikelen in het Compendium voor de Leefomgeving. Die betekenis kan vergroot worden door de cijfers te publiceren met cijfers uit eerdere perioden, maar ook door ze te publiceren met cijfers uit een hele andere bron.

De zwembaan daaronder laat het proces van het beschikbaar stellen van de data zien. Belangrijk is op te merken dat het per outputverplichting een selectie betreft uit de aggregaten die in koppelvlaak 4 staan. De naar buiten gebrachte data heet in de B&I-architectuur Vrijgegeven data. Het is mogelijk data te plaatsen zonder die te duiden (bijvoorbeeld voor het updaten van open data). Het plaatsingsproces verloopt geheel automatisch.

Hierna wordt alleen het deel van dit proces beschreven dat tot het automatiseringssysteem behoort.

5.3.1 Verzamelen outputdata [4.3.5]

Het proces 4.3.5 wordt gedraaid via het opstarten van bestand 4.3.5-Publiceren.cmd. De tabellen die worden samengesteld worden opgeslagen in de Publicatiemap en vormen samen Koppelvlaak 5. Data voor het verslagjaar dat al in een vorige ronde was weggeschreven wordt overschreven zodat daar altijd de laatste versie staat. Daarnaast worden alle tabellen ook in de Logmap weggeschreven. Aangezien elke keer draaien daar een nieuwe mapnaam krijgt, zijn daar uitkomsten van eerdere keren dat dit script gedraaid is wel terug te vinden.

Afhankelijk van de doelgroep zullen verschillende datasets samengesteld moeten worden waarvoor outputdata verzameld moet worden.

Inputbestanden:

- Database: "tbl_Koppelvlaak_4_statistieken"
- Database: "tbl_Koppelvlaak_3_bewerkte_data"

- Database: "tbl_Ophoogfactoren"
- Database: "tbl_Koppel_Bedrijfsklassen"

Script: 4.3.5-Publiceren.R

Dit script haalt de juiste aggregaten op uit Koppelvlak 4 voor de afzonderlijke publicaties. Verder worden specifieke labels of samentellingen (zoals de StatLine verplichte groepen) toegevoegd.

Data voor alle in Koppelvlak 4 aanwezige jaren wordt gebruikt voor een volledig nieuwe StatLine tabel. Eventuele aanpassingen in eerdere jaren worden daarmee dus ook meegenomen.

Outputbestanden (al deze output is Koppelvlak 5, ze komen zowel in de \Publicatiemap als in de \Logging map):

- \StatLine\84970NED_[JAAR].dat (Bedrijfsafval 1)
- \StatLine\84971NED_[JAAR].dat (Bedrijfsafval 2)
- \StatLine\81416NED_[JAAR].dat (VTR)
- \Nationale_Rekeningen\interne_verwerking.csv
- \CLO\basisdata_table_0116.csv
- \CLO\basisdata_table_0117.csv
- \CLO\basisdata_table_0118.csv
- \CLO\basisdata_table_0124.csv
- \DSC\DSC_BAS_[JAAR].csv
- \DSC\DSC_VTR_[JAAR].csv
- \RWS-LO\1_Delfstoffenwinning_[JAAR].csv
- \RWS-LO\2_Elektriciteitsbedrijven_[JAAR].csv
- \RWS-LO\3_Waterwinbedrijven_[JAAR].csv
- \RWS-LO\4_VorbereidingTR_[JAAR].csv
- \RWS-LO\5_AanvoerVTR_[JAAR].csv
- \RWS-LO\6_Industrie_[JAAR].csv

DSC krijgt geen ophoogfactoren meer, aangezien ophoogfactoren niet in koppelvlakken horen te staan.

RWS-LO krijgt in hun microdata output wel ophoogfactoren en opgehoogde hoeveelheden geleverd.

5.3.2 Statistische beveiliging

[4.3.6]

Uiteraard zullen de op het CBS geldende statistische beveiligingsregels toegepast moeten worden. Statistische beveiliging is het zodanig beschermen van statistische informatie dat het risico op onthulling de vooraf gestelde grenzen niet overschrijdt. De gegevens (en metadata) in de output mogen niet in strijd zijn met zowel de AVG als de CBS-wet. De mate en methode van statistische beveiliging kan variëren voor verschillende soorten output. De benadering die wordt gebruikt voor microdatasets voor onderzoeksdoeleinden zal bijvoorbeeld anders zijn dan die voor gepubliceerde tabellen of visualisaties op kaarten.

Dit is voor later, maar moet nog gebeuren. Tau Argus is de voor de hand liggende tool waarbij het wel complex kan zijn om de diverse output in een datamodel te zetten. Dit komt vooral door het kleine verschil tussen CLO en StatLine output en de extra uitsplitsing voor RWSLO naar gevaarlijke en niet-gevaarlijke euralcodes.

De extra uitsplitsing rwslo , gevaarlijk en niet gevaarlijk, is niet nodig. In principe hoort er geen gevaarlijk afval in het bestand van rwslo. Dat dat daar nog zit is een inconsequentie die ze er nu daar wel uit halen.

6 Overige modules

6.1.1 Populatiebestand

[0.0.1]

6.1.1.1 Beschikbaar stellen in koppelvlak 2

Het Populatiebestand wordt ingelezen via 0.0.1.1-Populatiebestand_inlezen_kv2.cmd

Dit is voldoende voor het inlezen van het uitzendbestand en voor het inlezen van alle uitkomsten tot en met koppelvlak 3. Voor het draaien van de koppelvlak 4 data (4.2-Schatten.cmd) moet het populatiebestand ook in koppelvlak 3 zijn ingelezen. Zie hiervoor 0.0.1.2-Populatiebestand_inlezen_kv3.cmd toegelicht onder kop 6.1.1.2.

Inputbestanden:

- Populatiebestand (CSV bestand dat de analist in de map \Import heeft geplaatst)
- Database: tbl_HulpInformatie_Analysejaar

Script: 0.0.1.1-Populatiebestand_inlezen_kv2.R

Het script komt al met de meest waarschijnlijk naam van het CSV bestand met aanvullende data aan door te kijken welke bestanden in de \Import map beginnen met “MABAS” of “POP”. Analist kan het overrulen met specifieke naam. Script leest dan het bestand in.

Er vindt technische controle plaats of de volgende kolommen aanwezig zijn:

"Beldentificatie", "BaseStratum", "Classification", "Status", "InSurvey", "PanelIndicator",
"StandId", "SbiGecoordineerd", "GkSbsGecoordineerd", "WPActueel", "RechtsvormCode",
"MaandBegin", "SectorCodeGecoordineerd", "TopXIndicator", "TopWIndicator", "CongoIndicator",
"N", "n"

Variabelen worden indien nodig naar numeriek geconverteerd.

De database (“tbl_Koppelvlak_2_populatie”) wordt bijgewerkt met de nieuw ingelezen data.

Aan de data wordt het verslagjaar gekoppeld zoals afkomstig uit “tbl_HulpInformatie_Analysejaar”.

Outputbestanden:

- Database: “tbl_Koppelvlak_2_populatie”
- 0.0.1.1_Populatiebestand_kv2.rds
- 0.0.1.1_Populatiebestand_kv2.csv

6.1.1.2 Beschikbaar stellen in koppelvlak 3

Het Populatiebestand wordt ingelezen via 0.0.1.2-Populatiebestand_inlezen_kv3.cmd

Inputbestanden:

- Database: “tbl_Koppelvlak_2_populatie”
- Database: “tbl_Koppel_Bedrijfsklassen”
- Database: tbl_HulpInformatie_Analysejaar
- Koppeltabel uit Excel die SBI aanpast bij een deel van de bedrijven in SBI 383

Bedrijven in SBI 383 krijgen een extra nummer mee om hun specifieke verwerkingsactiviteit te registreren. Dit is van belang voor het ophogen.

Er wordt gevraagd om de naam in te voeren van de koppeltabel in Excel.

Er vindt technische controle plaats of de volgende kolommen aanwezig zijn:

"BEID", "Sbi_be"

Met de waarde uit "Sbi_be" wordt de SBI in het veld "SbiGecoördineerd" aangepast.

De publicatiegroep code wordt gekoppeld. Dat is namelijk de variabele waarnaar gestratificeerd wordt bij de ophoging. N_pop is net als in koppelvak 2 het aantal bedrijven in de populatie en n is het aantal bedrijven in het uitzendbestand. Deze zijn gemaakt op basis van de steekproef en bevatten daarom de insluitkans van elke BE. Ze worden verder gebruikt in script 4.2.2-Bepalen_ophoogfactor.R bij het berekenen van de ophoogfactoren.

De database ("tbl_Koppelvak_3_populatie") wordt bijgewerkt met de nieuw ingelezen data.

Outputbestanden:

- Database: "tbl_Koppelvak_3_populatie"
- 0.0.1.2_Populatiebestand_kv3.rds
- 0.0.1.2_Populatiebestand_kv3.csv

6.1.2 Uitzendbestand

[0.0.2]

Het Uitzendbestand wordt ingelezen via 0.0.2-Uitzendbestand_inlezen.cmd

6.1.2.1 Beschikbaar stellen in koppelvak 2

Inputbestand:

- Uitzendbestand (Excel bestand dat de analist in de map \Import heeft geplaatst)
- Database: tbl_HulpInformatie_Analysejaar

Script: 0.0.2.1-Beschikbaar_stellen_kv2.R

Het script komt al met de meest waarschijnlijk naam van het XLSX uitzendbestand door te kijken welke bestanden in de \Import map beginnen met "UITZ". Analist kan het overrulen met specifieke naam. Script leest dan het bestand in.

Er vindt technische controle plaats of de volgende kolommen aanwezig zijn:

"Id", "EENHEID", "EENHEIDSOORT", "VAN BE", "SbiGecoördineerd", "stuurcode".

Aan de data wordt het verslagjaar gekoppeld zoals afkomstig uit "tbl_HulpInformatie_Analysejaar".

De database tabel "tbl_Koppelvak_2_uitzendbestand" wordt bijgewerkt.

Outputbestand: Database: "tbl_Koppelvak_2_uitzendbestand"

6.1.2.2 Beschikbaar stellen in koppelvlak 3

Inputbestanden:

- Database: "tbl_Koppelvlak_2_uitzendbestand"
- Database: "tbl_Koppelvlak_2_populatie"

Script: 0.0.2.2-Beschikbaar_stellen_kv3.R

Aantal Werkzame Personen (WP) van BEID uit Populatiebestand wordt aan de data gekoppeld.

De database tabel "tbl_Koppelvlak_3_uitzendbestand" wordt bijgewerkt.

Outputbestand: Database: "tbl_Koppelvlak_3_uitzendbestand"

6.1.3 EMJV algemene gegevens bestand

[0.0.3]

Het gehele proces 0.0.3 wordt gedraaid via het opstarten van bestand "0.0.3-EMJV_algemene_gegevens_inlezen.cmd".

Dit bestand voert alle afzonderlijke modules 0.0.3.1 t/m 0.0.3.7 in volgorde uit.

6.1.3.1 Ophalen EMJV algemene gegevens [0.0.3.1]

Inputbestanden:

- EMJV bestand met algemene gegevens (CSV bestand dat de analist in de map \Import heeft geplaatst)
- Database: tbl_HulpInformatie_Analysejaar

Script: 0.0.3.1-Ophalen_EMJV_algemene_gegevens.R

Het script komt al met de meest waarschijnlijk naam van het csv bestand door te kijken welke bestanden in de \Import map de tekst "algemene_gegevens" bevatten. Analist kan het overrulen met specifieke naam. Script leest dan het bestand in. Het gaat er van uit dat er geen enkele aanpassing is gedaan aan het invoerbestand sinds het is gedownload.

Het script leest de data in en zorgt dat kolomnamen ontdaan worden van spaties en vreemde tekens.

Outputbestand: 0.0.3.1_EMJV_algemene_gegevens_kv1.rds

6.1.3.2 Controle technisch [0.0.3.2]

Inputbestand: 0.0.3.1_EMJV_algemene_gegevens_kv1.rds

Script: 0.0.3.2-Controle_technisch.R

Er vindt technische controle plaats of de volgende kolommen aanwezig zijn:

"nic", "kvk_nummer", "kvk_vestigingsnummer", "werknemer_aantal".
--

Deze kolommen zijn essentieel voor het verwerkingsproces. Als één van deze kolommen mist, dan komt er een harde fout en stopt het script. Als andere kolommen missen, dan komt daarvan een melding op het scherm met de betreffende namen en worden door het script lege kolommen aangemaakt, zodat het wegschrijven in de database geen problemen oplevert.

Verder wordt er gecontroleerd of de data over het verslagjaar gaat. Doordat er geen kolom met het jaar in de dataset staat, wordt gekeken naar de bestandsnaam, welke - als er niks aan veranderd wordt na het downloaden - het jaartal aan het begin van de naam heeft staan. Als het jaar waarover de dataset gaat niet overeenkomt met het verslagjaar, dan stopt het script.

6.1.3.3 Transformeren [0.0.3.3]

Inputbestand: 0.0.3.1_EMJV_algemene_gegevens_kv1.rds

Script: 0.0.3.3-Transformeren.R

Eventuele ontbrekende kolommen die in database verwacht worden, worden leeg aangemaakt. Kolommen krijgen maximale lengte en namen worden aangepast.

Outputbestand: 0.0.3.3_EMJV_algemene_gegevens_getransformeerd.rds

6.1.3.4 Controle logisch [0.0.3.4]

Inputbestand: 0.0.3.3_EMJV_algemene_gegevens_getransformeerd.rds

Script: 0.0.3.4-Controle_logisch.R

Er is een controle op dubbel voorkomende NIC nummers in de rijen. Als die gevonden worden, dan komt er een melding op het scherm en worden de gevonden rijen weggeschreven in het bestand "0.0.3.4_EMJV_algemene_gegevens_dubbele_records.csv"

Outputbestanden:

- 0.0.3.4_EMJV_algemene_gegevens_dubbele_records.csv
- 0.0.3.4_EMJV_algemene_gegevens_gecontroleerd.rds

6.1.3.5 Beschikbaar stellen [0.0.3.5]

Inputbestand: 0.0.3.4_EMJV_algemene_gegevens_gecontroleerd.rds

Script: 0.0.3.5-Beschikbaar_stellen.R

Data wordt opgeslagen in tbl_Koppelvlak_2_EMJV_algemene_gegevens

Outputbestand:

- Database: tbl_Koppelvlak_2_EMJV_algemene_gegevens

6.1.3.6 Koppelen BE [0.0.3.5]

Inputbestanden:

- Database: tbl_Koppelvlak_2_EMJV_algemene_gegevens

- EHB22037_VEST_OUTPUT_geen_dubbelen.rds
- EHB22037_OUTPUT_geen_dubbelen.rds
- Database: tbl_Koppelvlak_3_bewerkte_data
- 0.0.3_NIC_BE_koppelingen.csv

Script: 0.0.3.6-Koppelen_BE.R

Via een apart system wordt in de map \\cbsp.nl\Productie\Primair\EAV\Werk\KvK_lijst_uit_ABR automatisch elke maand een koppelbestand tussen alle KvK nummers en BE-ID (EHB22037_OUTPUT.CSV) en een bestand met koppelingen tussen KvK vestigingen en BE-ID (EHB22037_VEST_OUTPUT.CSV) bijgewerkt. Er is al een R script dat dit opgeschoond (dubbelen er uit bijvoorbeeld) en er RDS bestanden van maakt. Dit wordt later aangepast zodat het automatisch gaat draaien.

Script “0.0.3-EMJV_algemene_gegevens_inlezen.R” maakt gebruik van deze opgeschoonde koppelingen om in eerste instantie van KvK vestiging nummers te converteren naar BE-ID.

In gevallen waarbij het KvK vestiging nummer niet is opgegeven, wordt de koppeling gelegd via het KvK nummer.

Als er hierna nog lege koppelingen zijn, dan wordt in data van eerdere jaren (database tabel “tbl_Koppelvlak_3_bewerkte_data”) gezocht naar de koppelingen van toen. Deze worden vervolgens voor het huidige verslagjaar overgenomen.

Als er vervolgens KVK zijn met meerdere KvK vestiging nummer en sommige vestigingen hebben een gekoppelde BEID en andere niet, dan wordt bij koppeling van resterende vestigingen zonder koppeling de BEID ingeschat op basis van een vestigingen met BEID die bij dezelfde KVK hoort. In het geval bij 1 KVK meerdere vestigingen horen die verschillende BEID's hebben, wordt de koppeling gekozen die het meeste voorkomt.

Mochten er nu nog lege koppelingen zijn, of als de analist vindt dat bepaalde koppelingen overschreven moeten worden, dan kan de analist handmatig aanpassingen doen in het bestand “0.0.3_NIC_BE_koppelingen.csv” in de \Instellingen map.

Outputbestand: 0.0.3.6-Koppelen_BE.rds

6.1.3.7 Beschikbaar stellen [0.0.3.7]

Inputbestand: 0.0.3.6-Koppelen_BE.rds

Script: 0.0.3.7-Aanpassingen_EMJV_algemeen.R

EMJV bedrijven die ook in de steekproef zitten, worden verwijderd uit het EMJV.

Outputbestand: 0.0.3.7-Aanpassingen_EMJV_algemeen.rds

6.1.3.8 Beschikbaar stellen [0.0.3.8]

Inputbestand: 0.0.3.7-Aanpassingen_EMJV_algemeen.rds

Script: 0.0.3.8-Beschikbaar_stellen.R

Data wordt opgeslagen in tbl_Koppelvlak_3_EMJV_algemene_gegevens.

Outputbestand:

- Database: tbl_Koppelvlak_3_EMJV_algemene_gegevens

6.1.4 RWSLO data invoeren

[0.0.4]

Het bestand met extra gegevens ontvangen van RWSLO wordt ingelezen via 0.0.4-RWSLO_gegevens_inlezen.cmd

Inputbestanden:

- RWSLO bestand (XLSX bestand dat de analist in de map \Import heeft geplaatst)

Script: 0.0.4-RWSLO_gegevens_inlezen.R

Het RWSLO bestand wordt door analist in een standaard vorm gezet. Dit resultaat moet komen in een tabblad met de naam "IMPORT". Het script leest alleen de data in die op een tabblad met deze naam voorkomt.

Er vindt technische controle plaats of de volgende kolommen aanwezig zijn:

"Jaar", "Verwerking_SL", "Pub_SL", "Totaal_bedrijfsafvalstoffen", "Totaal_niet_chemisch", "Metaal", "Glas", "Papier", "Kunststof_Rubber", "Hout", "Textiel_Leder", "Dierlijk_Plantaardig", "Gemengd", "Slib", "Mineralen", "Overig_niet_chemisch", "Chemisch", "Totaal_nuttige_toepassing", "Energierterugwinning", "Recycling", "Totaal_eindverwerking", "Verbranden", "Storten", "Scheiden"

Outputbestanden:

- Database: "tbl_Koppelvlak_2_RWSLO"
- 0.0.4-RWSLO_gegevens_inlezen.rds

7 Logisch Informatie Model (LIM)

7.1 dbo.tbl_HulpInformatie_Analysejaar

```
CREATE TABLE dbo.tbl_HulpInformatie_Analysejaar (  
  Id int NOT NULL IDENTITY PRIMARY KEY,  
  Jaar nvarchar(4) NOT NULL  
);
```

7.2 dbo.tbl_Koppelvlak_2_populatie

```
REATE TABLE dbo.tbl_Koppelvlak_2_populatie (  
  Id int NOT NULL IDENTITY,  
  Jaar nvarchar(4) NOT NULL,  
  BEID nvarchar(8) NOT NULL,  
  BaseStratum nvarchar(20),  
  Classification nvarchar(20),  
  InSurvey bit,  
  SbiGecoordineerd nvarchar(10),  
  GkSbsGecoordineerd nvarchar(2),  
  WPActueel int,  
  N_pop int,  
  n int,  
  InvoerDatum datetime NOT NULL,  
  EindDatum date  
  CONSTRAINT pk_tbl_Koppelvlak_2_populatie PRIMARY KEY (Jaar, BEID, InvoerDatum)  
);
```

7.3 dbo.tbl_Koppelvlak_3_populatie

```
CREATE TABLE dbo.tbl_Koppelvlak_3_populatie (  
  Id int NOT NULL IDENTITY,  
  Jaar nvarchar(4) NOT NULL,  
  BEID nvarchar(8) NOT NULL,  
  BaseStratum nvarchar(20),  
  Classification nvarchar(20),  
  InSurvey bit,  
  SbiGecoordineerd nvarchar(10),  
  GkSbsGecoordineerd nvarchar(2),  
  WPActueel int,  
  Publicatiegroep_code int,  
  N_pop int,  
  n int,  
  InvoerDatum datetime NOT NULL,  
  EindDatum date  
  CONSTRAINT pk_tbl_Koppelvlak_3_populatie PRIMARY KEY (Jaar, BEID, InvoerDatum)  
);
```

Hier is N_pop het aantal bedrijven per Publicatiegroep_code in de populatie en n is aantal bedrijven in het uitzendbestand.

7.4 dbo.tbl_Koppelvlak_2_uitzendbestand

```
CREATE TABLE dbo.tbl_Koppelvlak_2_uitzendbestand (  
  Id int NOT NULL IDENTITY,  
  Jaar nvarchar(4) NOT NULL,  
  CollectieEenheidId nvarchar(12) NOT NULL,  
  CollectieEenheidType nvarchar(10) NOT NULL,  
  Parent_BE nvarchar(8),
```

```

    InvoerDatum datetime NOT NULL,
    EindDatum date
CONSTRAINT pk_tbl_Koppelvlak_2_uitzendbestand PRIMARY KEY (Jaar,
CollectieEenheidId, CollectieEenheidType, InvoerDatum)
);

```

7.5 dbo.tbl_Koppelvlak_3_uitzendbestand

```

CREATE TABLE dbo.tbl_Koppelvlak_3_uitzendbestand (
    Id int NOT NULL IDENTITY,
    Jaar nvarchar(4) NOT NULL,
    CollectieEenheidId nvarchar(12) NOT NULL,
    CollectieEenheidType nvarchar(10) NOT NULL,
    Parent_BE nvarchar(8),
    WPActueel int,
    InvoerDatum datetime NOT NULL,
    EindDatum date
CONSTRAINT pk_tbl_Koppelvlak_3_uitzendbestand PRIMARY KEY (Jaar,
CollectieEenheidId, CollectieEenheidType, InvoerDatum)
);

```

7.6 dbo.tbl_Koppelvlak_2_EMJV_algemene_gegevens

```

CREATE TABLE dbo.tbl_Koppelvlak_2_EMJV_algemene_gegevens (
    Id int NOT NULL IDENTITY,
    Jaar nvarchar(4) NOT NULL,
    NIC nvarchar(12) NOT NULL,
    KvK_nummer nvarchar(255),
    KVK_Vestigingsnummer nvarchar(255),
    bedrijfsgroep_id nvarchar(255),
    bedrijfsgroep nvarchar(255),
    model_nummer nvarchar(255),
    model_naam nvarchar(255),
    bedrijf nvarchar(255),
    straat nvarchar(255),
    huisnr nvarchar(255),
    huisnr_toevoeging nvarchar(255),
    postcode nvarchar(255),
    plaats nvarchar(255),
    provincie nvarchar(255),
    x_coordinaat nvarchar(255),
    y_coordinaat nvarchar(255),
    n_coordinaat nvarchar(255),
    o_coordinaat nvarchar(255),
    stroomgebied nvarchar(255),
    sbi_code nvarchar(255),
    sbi_omschrijving nvarchar(255),
    milieu_coordinator nvarchar(255),
    bedrijf_contactpers_mja_mee nvarchar(255),
    milieu_coord_tel nvarchar(255),
    milieu_coord_fax nvarchar(255),
    milieu_coord_email nvarchar(255),
    milieu_coord_notificatie nvarchar(255),
    verantw_bedr nvarchar(255),
    wm nvarchar(255),
    water_zuiveringsschap nvarchar(255),
    rws nvarchar(255),
    contact_wm nvarchar(255),
    contact_wm_email nvarchar(255),
    contact_wm_notificatie nvarchar(255),
    contact_water_zuiv nvarchar(255),

```

```

contact_water_zuiv_email nvarchar(255),
contact_water_zuiv_notificatie nvarchar(255),
contact_rws nvarchar(255),
contact_rws_email nvarchar(255),
contact_rws_notificatie nvarchar(255),
waterkwaliteit_coord_gezag nvarchar(255),
prtr_coord_gezag nvarchar(255),
novem_code nvarchar(255),
agnl_code nvarchar(255),
emjv nvarchar(255),
eptr nvarchar(255),
afgesloten nvarchar(255),
concern nvarchar(255),
postadres_straat nvarchar(255),
postadres_huisnr nvarchar(255),
postadres_huisnr_toevoeging nvarchar(255),
postadres_postcode nvarchar(255),
postadres_plaats nvarchar(255),
ets_code nvarchar(255),
productie_volume nvarchar(255),
hoeveelheid nvarchar(255),
installatie_aantal nvarchar(255),
bedrijfstijd nvarchar(255),
werknemer_aantal nvarchar(255),
website_adres nvarchar(255),
email nvarchar(255),
opmerking nvarchar(255),
InvoerDatum datetime NOT NULL,
EindDatum date
CONSTRAINT pk_tbl_Koppelvlak_2_EMJV_algemene_gegevens PRIMARY KEY (Jaar, NIC,
InvoerDatum)
);

```

7.7 dbo.tbl_Koppelvlak_3_EMJV_algemene_gegevens

```

CREATE TABLE dbo.tbl_Koppelvlak_3_EMJV_algemene_gegevens (
    Id int NOT NULL IDENTITY,
    Jaar nvarchar(4) NOT NULL,
    NIC nvarchar(12) NOT NULL,
    KvK_nummer nvarchar(255),
    KvK_Vestigingsnummer nvarchar(255),
    BeIdentificatie nvarchar(255),
    bedrijfsgroep_id nvarchar(255),
    bedrijfsgroep nvarchar(255),
    model_nummer nvarchar(255),
    model_naam nvarchar(255),
    bedrijf nvarchar(255),
    straat nvarchar(255),
    huisnr nvarchar(255),
    huisnr_toevoeging nvarchar(255),
    postcode nvarchar(255),
    plaats nvarchar(255),
    provincie nvarchar(255),
    x_coordinaat nvarchar(255),
    y_coordinaat nvarchar(255),
    n_coordinaat nvarchar(255),
    o_coordinaat nvarchar(255),
    stroomgebied nvarchar(255),
    sbi_code nvarchar(255),
    sbi omschrijving nvarchar(255),
    milieu_coordinator nvarchar(255),

```



```

        bedrijf_contactpers_mja_mee nvarchar(255),
        milieu_coord_tel nvarchar(255),
        milieu_coord_fax nvarchar(255),
        milieu_coord_email nvarchar(255),
        milieu_coord_notificatie nvarchar(255),
        verantw_bedr nvarchar(255),
        wm nvarchar(255),
        water_zuiveringsschap nvarchar(255),
        rws nvarchar(255),
        contact_wm nvarchar(255),
        contact_wm_email nvarchar(255),
        contact_wm_notificatie nvarchar(255),
        contact_water_zuiv nvarchar(255),
        contact_water_zuiv_email nvarchar(255),
        contact_water_zuiv_notificatie nvarchar(255),
        contact_rws nvarchar(255),
        contact_rws_email nvarchar(255),
        contact_rws_notificatie nvarchar(255),
        waterkwaliteit_coord_gezag nvarchar(255),
        prtr_coord_gezag nvarchar(255),
        novem_code nvarchar(255),
        agnl_code nvarchar(255),
        emjv nvarchar(255),
        eprr nvarchar(255),
        afgesloten nvarchar(255),
        concern nvarchar(255),
        postadres_straat nvarchar(255),
        postadres_huisnr nvarchar(255),
        postadres_huisnr_toevoeging nvarchar(255),
        postadres_postcode nvarchar(255),
        postadres_plaats nvarchar(255),
        ets_code nvarchar(255),
        productie_volume nvarchar(255),
        hoeveelheid nvarchar(255),
        installatie_aantal nvarchar(255),
        bedrijfstijd nvarchar(255),
        werknemer_aantal nvarchar(255),
        website_adres nvarchar(255),
        email nvarchar(255),
        opmerking nvarchar(255),
        InvoerDatum datetime NOT NULL,
        EindDatum date
        CONSTRAINT pk_tbl_Koppelvlak_3_EMJV_algemene_gegevens PRIMARY KEY (Jaar, NIC,
InvoerDatum)
    );

```

7.8 dbo.tbl_Koppelvlak_2_RWSLO

```

CREATE TABLE dbo.tbl_Koppelvlak_2_RWSLO (
    Id int NOT NULL IDENTITY,
    Jaar nvarchar(4) NOT NULL,
    Verwerking_SL int NOT NULL,
    Pub_SL int,
    Totaal_bedrijfsafvalstoffen int,
    Totaal_niet_chemisch int,
    Metaal int,
    Glas int,
    Papier int,
    Kunststof_Rubber int,
    Hout int,
    Textiel_Leder int,

```

```

Dierlijk_Plantaardig int,
Gemengd int,
Slib int,
Mineralen int,
Overig_niet_chemisch int,
Chemisch int,
Totaal_nuttige_toepassing int,
Energieterugwinning int,
Recycling int,
Totaal_eindverwerking int,
Verbranden int,
Storten int,
Scheiden int,
InvoerDatum datetime NOT NULL,
EindDatum date
CONSTRAINT pk_tbl_Koppelvlak_2_RWSLO PRIMARY KEY (Jaar, Verwerking_SL, Pub_SL,
InvoerDatum)
);

```

7.9 dbo.tbl_Koppelvlak_2_niet_afval_data

```

CREATE TABLE dbo.tbl_Koppelvlak_2_niet_afval_data (
    Id int NOT NULL IDENTITY,
    Jaar nvarchar(4) NOT NULL,
    CollectieEenheidId nvarchar(12) NOT NULL,
    CollectieEenheidType nvarchar(10) NOT NULL,
    Correspondentienummer nvarchar(255),
    ResponsStatus nvarchar(255),
    RedenNonRespons nvarchar(255),
    Inzend_Datum nvarchar(255),
    Inzend_Tijd nvarchar(255),
    WP int,
    Contact_Aanhef nvarchar(255),
    Contact_Voorletters nvarchar(255),
    Contact_Achternaam nvarchar(255),
    Contact_EmailAdres nvarchar(255),
    Contact_Telefoon nvarchar(255),
    Contact_Status nvarchar(255),
    Contact_Opmerkingen nvarchar(255),
    BestedeTijd_Uren int,
    BestedeTijd_Minuten int,
    InvoerDatum datetime NOT NULL,
    EindDatum date
    CONSTRAINT pk_tbl_Koppelvlak_2_niet_afval_data PRIMARY KEY (Jaar,
CollectieEenheidId, CollectieEenheidType, InvoerDatum)
);

```

7.10 dbo.tbl_Koppelvlak_2_gestandaardiseerde_data

```

CREATE TABLE dbo.tbl_Koppelvlak_2_gestandaardiseerde_data (
    Id int NOT NULL IDENTITY PRIMARY KEY,
    Jaar nvarchar(4) NOT NULL,
    CollectieEenheidId nvarchar(12) NOT NULL,
    CollectieEenheidType nvarchar(10) NOT NULL,
    RowID int,
    Eural_code nvarchar(10),
    Omschrijving nvarchar(255),
    Eural_gevaarlijk tinyint,
    Verwerking_code nvarchar(3),
    Verwerking_naam nvarchar(50),

```

```
Hoeveelheid_ton numeric(15, 3),
DrogeStof_pct int,
Bron nvarchar(10),
InvoerDatum datetime NOT NULL,
EindDatum date
);
```

7.11 dbo.tbl_Koppelvlak_3_bewerkte_data

```
CREATE TABLE dbo.tbl_Koppelvlak_3_bewerkte_data (
  Id int NOT NULL IDENTITY PRIMARY KEY,
  Jaar nvarchar(4) NOT NULL,
  CollectieEenheidId nvarchar(12) NOT NULL,
  CollectieEenheidType nvarchar(10) NOT NULL,
  BEID nvarchar(8),
  RowID int,
  SBI nvarchar(10),
  GkSbs nvarchar(2),
  Eural_code nvarchar(10),
  Eural_naam nvarchar(255),
  Eural_gevaarlijk tinyint,
  EWC_code nvarchar(5),
  Verwerking_code nvarchar(3),
  Verwerking_naam nvarchar(50),
  Hoeveelheid_ton numeric(15, 3),
  DrogeStof_ton numeric(15, 3),
  Marker nvarchar(255),
  Opmerkingen nvarchar(255),
  GrondstoffenOmschrijving600 nvarchar(255),
  Bron nvarchar(10),
  InvoerDatum datetime NOT NULL,
  EindDatum date
);
```

7.12 dbo.tbl_Koppelvlak_4_statistieken

```
CREATE TABLE dbo.tbl_Koppelvlak_4_statistieken (
  Id int NOT NULL IDENTITY PRIMARY KEY,
  Jaar nvarchar(4) NOT NULL,
  Aggregaat nvarchar(255) NOT NULL,
  Pub_SL int,
  Verwerking_SL int,
  Verwerking_SL_naam nvarchar(50),
  Verwerking_code nvarchar(3),
  Verwerking_naam nvarchar(50),
  IE_verwerking nvarchar(10),
  Eural_gevaarlijk tinyint,
  EWC_code nvarchar(5),
  SBI_Code nvarchar(10),
  SBI_tekst nvarchar(50),
  SL_Code nvarchar(10),
  Afvalsoort nvarchar(50),
  Hoeveelheid_ton numeric(15, 0),
  InvoerDatum datetime NOT NULL,
  EindDatum date
);
```

Sommige afvalsoorten tellen op tot andere afvalsoorten. Ze komen allen in deze tabel voor. Om geen dubbeltellingen te krijgen selecteer je de afvalstoffen **Metaal, Glas, Papier, Kunststof_Rubber, Hout, Textiel_Leder, Dierlijk, Plantaardig, Dierlijk_Plantaardig_gemengd, Gemengd, Slib, Slak_en_as, Mineralen_overig, Overig_niet_chemisch, Chemisch.**

Dit zijn de samengestelde afvalsoorten die voor publicaties gebruikt worden:

- **Dierlijk_Plantaardig** = som(Dierlijk, Plantaardig, Dierlijk_Plantaardig_gemengd).
- **Mineralen** = som(Slak_en_as, Mineralen_overig).
- **Totaal_niet_chemisch** = som(Metaal, Glas, Papier, Kunststof_Rubber, Hout, Textiel_Leder, Dierlijk_Plantaardig, Gemengd, Slib, Mineralen, Overig_niet_chemisch).
- **EWC_7** = som(Glas, Papier, Kunststof_Rubber, Hout, Textiel_Leder).
- **Totaal_bedrijfsafvalstoffen** = som(Totaal_niet_chemisch, Chemisch).

De verwerkingsmethode staat in verschillende aggregatieniveaus in Verwerking_SL en Verwerking_code, en is daarnaast ook toegevoegd aan Afvalsoort, analoog aan hoe het was in het oude systeem.

Er is een view genaamd view_Koppelvlak_4_statistieken_Totaal_per_Afvalsoort die deze tabel samenteld tot een compactere tabel met totalen per Afvalsoort ongeacht verwerkingmethodes. Zie kop 7.23.

7.13 dbo.tbl_schatting_kleine_gk

```
CREATE TABLE dbo.tbl_schatting_kleine_gk (
  Id int NOT NULL IDENTITY PRIMARY KEY,
  Jaar nvarchar(4) NOT NULL,
  CollectieEenheidId nvarchar(12) NOT NULL,
  CollectieEenheidType nvarchar(10) NOT NULL,
  Publicatiegroep_code int,
  Verwerking_code nvarchar(3),
  Eural_code nvarchar(10),
  EWC_code nvarchar(5),
  som_hoev numeric(15, 3),
  som_wp numeric(15, 3),
  bron int,
  model int,
  factor numeric(15, 3),
  Hoeveelheid_ton_opgehoogd numeric(15, 3),
  InvoerDatum datetime NOT NULL,
  EindDatum date
);
```

7.14 dbo.tbl_Koppel_1_1_3_variabelen_met_RowID_en_Classificatie

```
CREATE TABLE dbo.tbl_Koppel_1_1_3_variabelen_met_RowID_en_Classificatie (
  Jaar nvarchar(4) NOT NULL,
  variable nvarchar(255) NOT NULL,
  RowID int NOT NULL,
  Category int NOT NULL,
  CONSTRAINT pk_tbl_Koppel_1_1_3_variabelen_met_RowID_en_Classificatie PRIMARY KEY
(Jaar, variable, RowID, Category)
);
```

7.15 dbo.tbl_Koppel_1_1_3_standaard_waarden_formulier_met_RowID

```
CREATE TABLE dbo.tbl_Koppel_1_1_3_standaard_waarden_formulier_met_RowID (
  Jaar nvarchar(4) NOT NULL,
  RowID int NOT NULL,
  Eural_code nvarchar(255),
  Omschrijving nvarchar(255),
  Verwerking_code nvarchar(3),
```



```

    CONSTRAINT pk_tbl_Koppel_1_1_3_standandaard_waarden_formulier_met_RowID PRIMARY KEY
(Jaar, RowID)
);

```

7.16 dbo.tbl_Koppel_1_1_3_Afvalverwerkingsmethode_DVZ_hercoderen

```

CREATE TABLE dbo.tbl_Koppel_1_1_3_Afvalverwerkingsmethode_DVZ_hercoderen (
    Jaar nvarchar(4) NOT NULL,
    Verwerking_code_phoenix nvarchar(50) NOT NULL,
    Verwerking_code nvarchar(3),
    Verwerking_naam nvarchar(50)
    CONSTRAINT pk_tbl_Koppel_1_1_3_Afvalverwerkingsmethode_DVZ_hercoderen PRIMARY KEY
(Jaar, Verwerking_code_phoenix)
);

```

7.17 dbo.tbl_Koppel_4_1_4_Verwerking_code_naam

```

CREATE TABLE dbo.tbl_Koppel_4_1_4_Verwerking_code_naam (
    Jaar nvarchar(4) NOT NULL,
    Verwerking_code nvarchar(3),
    Verwerking_naam nvarchar(50)
    CONSTRAINT pk_tbl_Koppel_4_1_4_Verwerking_code_naam PRIMARY KEY (Jaar,
Verwerking_code)
);

```

7.18 dbo.tbl_Koppel_4_1_1_EURAL_EWC

```

CREATE TABLE dbo.tbl_Koppel_4_1_1_EURAL_EWC (
    Jaar nvarchar(4) NOT NULL,
    Eural_code nvarchar(10) NOT NULL,
    Eural_naam nvarchar(255),
    Eural_gevaarlijk tinyint,
    EWC_code nvarchar(5),
    EWC_omschrijving nvarchar(255)
    CONSTRAINT pk_tbl_Koppel_4_1_1_EURAL_EWC PRIMARY KEY (Jaar, Eural_code)
);

```

7.19 dbo.tbl_Koppel_2_2_3_Afvalverwerkingsmethode_EMJV_hercoderen

```

CREATE TABLE dbo.tbl_Koppel_2_2_3_Afvalverwerkingsmethode_EMJV_hercoderen (
    Jaar nvarchar(4) NOT NULL,
    afvalverwerkingsmethode nvarchar(50) NOT NULL,
    Verwerking_code nvarchar(3)
    CONSTRAINT pk_tbl_Koppel_2_2_3_Afvalverwerkingsmethode_EMJV_hercoderen PRIMARY KEY
(Jaar, afvalverwerkingsmethode)
);

```

7.20 dbo.tbl_Koppel_4_2_3_EWC_StatLine_Categorie

```

CREATE TABLE dbo.tbl_Koppel_4_2_3_EWC_StatLine_Categorie (
    Jaar nvarchar(4) NOT NULL,
    EWC_code nvarchar(10) NOT NULL,
    EWC_omschrijving nvarchar(255),
    StatLine_Categorie_nr nvarchar(5),
    StatLine_Categorie_Naam nvarchar(255)
    CONSTRAINT pk_tbl_Koppel_4_2_3_EWC_StatLine_Categorie PRIMARY KEY (Jaar, EWC_code)
);

```

```
);
```

7.21 dbo.tbl_Koppel_Bedrijfsklassen

```
CREATE TABLE dbo.tbl_Koppel_Bedrijfsklassen (  
    Jaar nvarchar(4) NOT NULL,  
    SBI nvarchar(10) NOT NULL,  
    Publicatiegroep_code int,  
    Publicatiegroep_omschrijving nvarchar(255),  
    Pub_SL int,  
    SL_Code nvarchar(50),  
    SL_omschrijving nvarchar(255),  
    SL_Code2 nvarchar(10)  
    CONSTRAINT pk_tbl_Koppel_Bedrijfsklassen PRIMARY KEY (Jaar, SBI)  
);
```

7.22 dbo.tbl_Ophoogfactoren

```
CREATE TABLE dbo.tbl_Ophoogfactoren (  
    Id int NOT NULL IDENTITY,  
    Jaar nvarchar(4) NOT NULL,  
    CollectieEenheidId nvarchar(12) NOT NULL,  
    CollectieEenheidType nvarchar(10) NOT NULL,  
    BEID nvarchar(8),  
    Ophoogfactor numeric(12, 5),  
    InvoerDatum datetime NOT NULL,  
    EindDatum date  
    CONSTRAINT pk_tbl_Ophoogfactoren PRIMARY KEY (Id)  
);
```

7.23 view_Koppelvlak_4_statistieken_Totaal_per_Afvalsoort

```
CREATE VIEW dbo.view_Koppelvlak_4_statistieken_Totaal_per_Afvalsoort  
AS  
SELECT Jaar, Aggregaat, Pub_SL, SBI_Code, SBI_tekst, SL_Code, Afvalsoort,  
SUM(Hoeveelheid_ton) AS Hoeveelheid_ton  
FROM dbo.tbl_Koppelvlak_4_statistieken  
GROUP BY Jaar, Aggregaat, Pub_SL, SBI_Code, SBI_tekst, SL_Code, Afvalsoort,  
Verwerking_SL_naam, Verwerking_naam, IE_verwerking, EindDatum  
HAVING Verwerking_SL_naam = 'Totaal' AND Verwerking_naam = 'Totaal' AND IE_verwerking  
= 'Totaal' AND EindDatum IS NULL;
```