EESTI
STATISTIKA

ANDMEHALDUSE JUHISED

Andmekvaliteedi juhis
Lisa 1 Rakendamise näited R-is

Mai 2023

Versioon 1.1

Dokumendi ajalugu

ver	muutuse sisu	autor	kuupäev
1.0	Juhise aluseks on 2019 –2020. a koostatud juhis „Eesti andmehalduse metoodikaprojekt. Andmekvaliteedi juhis“	Raivo Ruusalepp, Peep Küngas, Siim Aben, Veiko Berendsen	aug 2020
1.1	Versiooni 1.0 peatükk rakendamise näited on tõstetud eraldi käesolevaks lisaks 1	Veiko Berendsen	mai 2023

Sisukord

1 Üldist

Antud sektsioonis on kirjeldatud profileerimise, andmekvaliteedi mõõtmise, andmekvaliteedi juhtimislaua loomise ja reeglite haldamise rakenduslikud näited. Näited on esitatud programmeerimiskeeles R[1], mida kasutatakse laialdaselt andmete analüüsiks. Näidisalgoritmide käivitamise eelduseks on, et arvutisse on paigaldatud programmeerimiskeele R interpretaator[2] ning arendustööriist RStudio[3].

Lisaks kasutavad järgnevalt esitatud andmekvaliteedi mõõtmise näited Mark van der Loo ja Edwin de Jonge poolt arendatud *validate*[4] paketti. *Validate* pakett lihtsustab R keeles andmete kvaliteedi mõõtmist ning andmekvaliteedi reeglite haldamist.

Pakett võimaldab:

- testida andmekomplekti vastu eeldefineeritud reegleid, kas siis andmekomplekti sees või nende üleselt;
- importida ja eksportida defineeritud andmekvaliteedi reegleid;
- uurida ja visualiseerida andmekvaliteedi tulemusi;
- teostada lihtsat reeglite haldust;
- defineerida ja hallata andmete kvaliteedi indikaatoreid andmetest eraldiseisvalt.

Paketi peamised objektid:

- *validator* - objekt, mis esitab hulka reegleid, millele andmed peavad vastama
- *indicator* - objekt, mis esitab hulka numbrilisi kvaliteedi indikaatoreid
- *confrontation* - objekt, mis esitab andmete andmekvaliteedi reeglite või indikaatoritega vastandamise tulemusi

Paketis on ka meetod *confront*, mis rakendab andmetele andmekvaliteedi reegleid või andmekvaliteedi indikaatoreid.

Lisaks *validate* paketile kasutatakse rakenduslikes näidetes paketti *dlookr*[5] andmete profileerimiseks ning pakette *data.table*[6], *stringr*[7] ja *tidyr*[8] andmete töötluse lihtsustamiseks.

[1] https://www.r-project.org/
[2] https://cran.r-project.org/bin/windows/base/
[3] https://rstudio.com/products/rstudio/download/
[4] https://cran.r-project.org/web/packages/validate/vignettes/introduction.html
[5] https://cran.r-project.org/web/packages/dlookr/
[6] https://cran.r-project.org/web/packages/data.table/
[7] https://cran.r-project.org/web/packages/stringr/
[8] https://cran.r-project.org/web/packages/tidyr/

2 Pakettide paigaldamine ja näidisandmed

Esimese sammuna on vaja arvutisse paigaldada näidetes kasutatavad paketid. Selleks tuleb jooksutada alljärgnevaid käske.

```
install.packages("dlookr")
install.packages("validate")
install.packages("data.table")
install.packages("stringr")
install.packages("tidyr")
```

Juhul kui eelnevalt toodud käskude käivitamine ebaõnnestub võib põhjuseks olla Rtools40 tööriista puudumine. Sellisel juhul tuleb järgmise sammuna nimetatud tööriist alla laadida ning arvutisse paigaldada. Tööriista saab alla laadida siit: https://cran.rstudio.com/bin/windows/Rtools/. Peale tööriista paigaldamist tuleb uus tööriistakomplekt seadistada. Selleks tuleb RStudios käivitada alljärgnev käsk.

```
writeLines('PATH="${RTOOLS40_HOME}\\usr\\bin;${PATH}"', con =
"~/.Renviron")
```

Peale kirjeldatud tegevuste teostamist tuleb eelnevalt toodud pakettide installeerimise käsud uuesti käivitada (juhul kui need eelnevalt ebaõnnestusid).

Järgmiseks tuleb äsja paigaldatud paketid sisse lugeda.

```
library(dlookr)
library(validate)
library(data.table)
library(stringr)
library(tidyr)
```

Viimaseks ettevalmistavaks sammuks on näidisandmete sisse lugemine. Antud juhul on näidisandmetena kasutatud Eesti Avaandmete Portaalist avalikult kättesaadavaid ehitisregistri[9] andmeid ning samuti Eesti Avaandmete Portaalist kättesaadavaid aadressiandmeid[10]. Näidisandmete komplekt koosneb kaheksast failist ning seda on võimalik tellida nimetatud portaalist või alla laadida siit: https://www.dropbox.com/sh/bf886osqo7kdya7/AACgIn1TbYv7JBROVTV5TJyDa?dl=0

Allalaetud fail tuleks arvutis salvestada sobivasse kausta. Näite puhul on failide asukohaks arvuti töölaual asuv kaust „Andmed“ ("C:\\Users\\Margus\\Desktop\\Andmed\\").

Järgmiseks tuleb RStudios määrata failide kataloog järgmise käsuga.

```
filedir <- "C:\\Users\\Margus\\Desktop\\Andmed\\"
```

Seejärel anname failidele nimed käivitades järgnevad käsud.

[9] https://opendata.riik.ee/andmehulgad/ehitisregister/
[10] https://opendata.riik.ee/andmehulgad/aadressiandmed-aadressid-koos-komponentide-ja-ajalooga/

```
eh_ehitised_file <- paste0(filedir, "eh_ehitised_1994-01_2020-02-09.csv")
eh_tehna_file <- paste0(filedir, "eh_tehna_1994-01_2020-02-09.csv")
kl_tehna_file <- paste0(filedir, "kl_tehna_1994-01_2020-02-09.csv")
aadressid_file <- paste0(filedir, "AADRESSID.csv")
eh_ehitis_osad_file <- paste0(filedir, "eh_ehitis_osad_1994-01_2020-02-
09.csv")
ehitis_aadress_file <- paste0(filedir, "ehitis_aadress_1994-01_2020-02-
09.csv")
hoone_energia_margised_file <- paste0(filedir,
"hoone_energia_margised_1994-01_2020-02-09.csv")
ad_aadress_file <- paste0(filedir, "ad_aadress_1994-01_2020-02-09.csv")
```

Peale failidele nimede andmist loeme andmed sisse. Selleks tuleb käivitada järgmised käsud.

```
#ehitise andmed
h_ehitised <- fread(eh_ehitised_file, encoding="UTF-8")

#ehitise tehnilised andmed
eh_tehna <- fread(eh_tehna_file, encoding="UTF-8")

#tehniliste andmete klassifikaatorid
kl_tehna <- fread(kl_tehna_file, encoding="UTF-8")

#ehitise osad
eh_ehitis_osad <- fread(eh_ehitis_osad_file, encoding="UTF-8")

#energiamärgised
hoone_energia_margised <- fread(hoone_energia_margised_file,
encoding="UTF-8")

#seosed ehitise ja aadressi vahel
ehitis_aadress <- fread(ehitis_aadress_file, encoding="UTF-8")

#kohalik koopia aadressandmete tabelist
ad_aadress <- fread(ad_aadress_file, encoding="UTF-8")

#Aadressid aadressiandmete süsteemist
aadressid <- fread(aadressid_file, encoding="UTF-8")
```

3 Andmete profileerimise rakendamise näide

Andmekvaliteedi programmiga alustades tuleb esmalt saada ülevaade olemasolevatest andmetest. Selleks on mõislik toestada andmete profileerimine. Profileerimist on võimalik teostada mitmesuguste tööriistadega, kuid antud näide illustreerib *dlookr* paketi kasutamist. Paketi paigaldamiseks tehtavad tegevused on kirjeldatud eelnevas sektsioonis.

Ehitiste andmetabeli põhjal esmase andmekvaliteedi raporti (*Data Quality Diagnosis Report*) genereerimiseks tuleb käivitada järgnev käsk. Sarnaselt saab genereerida raporteid ka teiste andmetabelite põhjal.

```
eh_ehitised %>% diagnose_report(output_format = "html", output_file =
"Diagn_eh_ehitised.html")
```

Kui arvutisse on eelnevalt paigaldatud TeX[11] saab raportit genereerida ka pdf formaadis. Selleks tuleb käivitada alljärgnev käsk.

```
diagnose_report(eh_ehitised)
```

Käsu käivitamisel genereeritakse fail nimega Diagn_eh_ehitised.html (või fail nimega DataDiagnosis_Report.pdf juhul kui käivitati pdf formaadis faili genereerimise käsk). Kirjeldatud tegevuse eesmärgiks on saada esmane ülevaade andmetest. Saadud tulemused on abiks ka andmekvaliteedi reeglite kirjeldamisel.

dlookr paketi üldise andmekvaliteedi raporti dokumentatsiooniga on võimalik tutvuda siin:

https://cran.r-project.org/web/packages/dlookr/vignettes/diagonosis.html

Genereeritud raport annab põhjaliku esmase vaate olemasolevatele andmetele. Näiteks võimaldab raporti osana genereeritud arvväärtuste diagnoosimise tabel (Joonis 1) tuvastada võimalikke arvväärtuste kvaliteediprobleeme ning sõnastada andmekvaliteedi reegleid. Alltoodud ehitise andmete põhjal genereeritud näitest näeme, et andmeelemendi „ehitusalane_pind“ miinimumväärtus on negatiivne (-24). See võib viidata, et andmetes esineb kvaliteediprobleeme. Lisaks on tuvastatud probleemid toeks andmekvaliteedi reeglite sõnastamisel. Näiteks on kirjeldatud näite puhul võimalik sõnastada järgnev reegel: „Ehitise ehitusalane pind peab olema positiivne.“

variables	min	median	max	minus	minus ratio(%)
korgus	-300.0	2.6	15,205.0	7,205	0.681
sygavus	-150.0	0.0	832.4	2,106	0.199
pikkus	-275.0	9.2	5,361,019.0	1,166	0.110
maaaluste_korruste_arv	-2.0	0.0	17.0	593	0.056
max_korruste_arv	-23.0	1.0	120.0	10	0.001
laius	-65.0	3.0	60,482.0	9	0.001
abs_0_korgus	-16.0	37.0	7,700.0	4	0.000
maht_bruto	-10.0	150.0	10,452,091.0	3	0.000
ehitisalune_pind	-24.0	64.0	13,432,000.0	2	0.000
koetav_pind	-1.6	80.6	360,540.0	1	0.000

Joonis 1. Näidis dlookr üldisest andmekvaliteedi raportist

[11] https://miktex.org/

Lisaks eelpool toodud raportile võimaldab *dlookr* pakett genereerida ka andmeanalüüsi raportit (*Exploratory Data Analysis Report*). Nimetatud raporti ehitiste andmete põhjal genereerimiseks tuleb käivitada järgnev käsk.

```
eh_ehitised %>% eda_report(output_format = "html", output_file =
"EDA_Report_eh_ehitised.html")
```

Kirjeldatud käsu tulemusena genereeritakse fail nimega EDA_Report_eh_ehitised.html. Sarnaselt eelpool kirjeldatuga saab ka antud raportit genereerida pdf formaadis, juhul kui arvutisse on eelnevalt installeeritud TeX[12]. Selleks tuleb käivitada järgnev käsk.

```
eda_report(eh_ehitised)
```

Käsu käivitamise tulemusena genereeritakse fail nimega EDA_Report.pdf.

dlookr andmeanalüüsi raporti dokumentatsiooniga on võimalik tutvuda siin:

https://cran.r-project.org/web/packages/dlookr/vignettes/EDA.html

Kirjeldatud raportite genereerimine ja tulemustega tutvumine võimaldab saada hea esmase ülevaate andmetest. Seejuures on oluline meeles pidada, et profileerimine ei ole andmekvaliteedi mõõtmine ning siit tuleks edasi liikuda andmekvaliteedi süsteemse mõõtmise suunas.

[12] https://miktex.org/

4 Andmekvaliteedi mõõtmise näidisjuhtumid

4.1 Näide N1 (Täielikkus, puuduv väärtus)

Dimensioon: Täielikkus.
Probleem: Puuduv väärtus
Reegli kirjeldus: Igal ehitisel peab olema määratud tema tunnus (ehitisregistri kood).
Selgitus: Ehitisregistri tabelis *eh_ehitised* peab igal ehitisel olema määratud ehitisregistri kood *ehr_kood*. Täielikkuse testimiseks loome reeglite komplekti *existance_rules*, kuhu lisame reegli *exists_id,* mis kontrollib kas andmeobjektil on tunnus *ehr_kood* määratud.

```
existence_rules <- validator(
 exists_id = !is.na(ehr_kood)
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised kaks käsku.

```
check <- confront(eh_ehitised, existence_rules, key="ehr_kood")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
       name   items  passes fails nNA error warning       expression
1 exists_id 1058629 1058629     0   0 FALSE   FALSE !is.na(ehr_kood)
```

Tulemustest on näha, et kõigil ehitistel on ehitisregistri kood määratud (*fails* ehk reeglile mittevastavate kirjete arv on 0).

4.2 Näide N2 (Ühekordsus, unikaalse väärtuse rikkumine)

Dimensioon: Ühekordsus.
Probleem: Unikaalse väärtuse rikkumine
Reegli kirjeldus: Iga ehitise tunnus (ehitisregistri kood) peab olema unikaalne
Selgitus: Ehitisregistri tabelis *eh_ehitised* peab iga ehitise ehitusregistri kood olema unikaalne. Unikaalsuse testimiseks loome reeglite komplekti *uniqueness_rules,* kuhu lisame reegli *unique_id,* mis kontrollib kas andmeobjektil on tunnus *ehr_kood* unikaalne.

```
uniqueness_rules <- validator(
 unique_id = is_unique(ehr_kood)
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised käsud.

```
check <- confront(eh_ehitised, uniqueness_rules, key="ehr_kood")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
       name   items  passes fails nNA error warning         expression
1 unique_id 1058629 1058629     0   0 FALSE   FALSE is_unique(ehr_kood)
```

Tulemustest on näha, et kõigil ehitistel on unikaalne tunnus (ehitisregistri kood).

4.3 Näide N3 (Õigsus, väärtusvahemiku rikkumine)

Dimensioon: Õigsus.
Probleem: Väärtusvahemiku rikkumine
Reegli kirjeldus: Omandi liigi ja seisundi väärtused peavad vastama etteantud loendi väärtutele.
Selgitus: Ehitisregistri tabelis *eh_ehitised* võivad atribuudi *omandi_liik* väärtused olla järgnevad:

- EHITIS_OMANDI_LIIK_VALLAS;
- EHITIS_OMANDI_LIIK_KINNIS;
- EHITIS_OMANDI_LIIK_VALLASRO.
-

Atribuudi *seisund* väärtused võivad olla järgnevad:

- EHITIS_SEISUND_EHITAM_LUBA;
- EHITIS_SEISUND_EHITAM_LUBA_EIKEH;
- EHITIS_SEISUND_EHITAMISEL;
- EHITIS_SEISUND_KASUT_LUBA_EIKEH;
- EHITIS_SEISUND_KASUT_MAAS;
- EHITIS_SEISUND_KASUT_OSALINE;
- EHITIS_SEISUND_KASUTUSEL;
- EHITIS_SEISUND_LAMMUT_LUBA;
- EHITIS_SEISUND_LAMMUTAMISEL;
- EHITIS_SEISUND_LAMMUTATUD;
- EHITIS_SEISUND_MAARAMATA;
- EHITIS_SEISUND_MENETLUSES;
- EHITIS_SEISUND_REG_OBJ_LOPP.

Seega kirjeldame vastavalt eeltoodud selgitusele reeglite komplekti *ehitised_rules,* kuhu lisame reeglid *property_type* ja *state_type*.

```
ehitised_rules <- validator(
 property_type = omandi_liik %in% c('EHITIS_OMANDI_LIIK_VALLAS',
'EHITIS_OMANDI_LIIK_KINNIS', 'EHITIS_OMANDI_LIIK_VALLASRO'),
 state_type = seisund %in% c('EHITIS_SEISUND_EHITAM_LUBA',
'EHITIS_SEISUND_EHITAM_LUBA_EIKEH', 'EHITIS_SEISUND_EHITAMISEL',
'EHITIS_SEISUND_KASUT_LUBA_EIKEH', 'EHITIS_SEISUND_KASUT_MAAS',
'EHITIS_SEISUND_KASUT_OSALINE', 'EHITIS_SEISUND_KASUTUSEL',
'EHITIS_SEISUND_LAMMUT_LUBA', 'EHITIS_SEISUND_LAMMUTAMISEL',
'EHITIS_SEISUND_LAMMUTATUD', 'EHITIS_SEISUND_MAARAMATA',
'EHITIS_SEISUND_MENETLUSES', 'EHITIS_SEISUND_REG_OBJ_LOPP')
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised käsud.

```
check <- confront(eh_ehitised, ehitised_rules, key="ehr_kood")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
           name   items  passes fails nNA error warning
1 property_type 1058629 1057500  1129   0 FALSE   FALSE
2    state_type 1058629 1058628     1   0 FALSE   FALSE
```

Väljavõte näitab, et 1058629-st kirjest ei ole vastavalt 1129-s ja 1-s kirjes omandi liik ja seisund õigesti kodeeritud.

Järgnevalt vaatame lähemalt millised kirjed ei vasta reeglile. Selleks konverteerime esmalt tulemuse kokkuvõtte sobivale kujule ning filtreerime tulemused nii, et alles jäävad vaid reeglitele mittevastavad kirjed. Kirjeldatu teostamiseks tuleb käivitada järgnevad kaks käsku.

```
output <- as.data.frame(check)
output[output$value == FALSE,]
```

Käskude käivitamise tulemusena kuvatakse RStudio konsoolis ülevaade probleemsetest kirjetest.

```
       ehr_kood          name value
   1: 120773369 property_type FALSE
   2: 120739741 property_type FALSE
   3: 120713061 property_type FALSE
   4: 120713124 property_type FALSE
   5: 120713112 property_type FALSE
  ---
1126: 120834655 property_type FALSE
1127: 120827890 property_type FALSE
1128: 120713778 property_type FALSE
1129: 120826592 property_type FALSE
1130: 221294327    state_type FALSE
```

Tuvastatud konkreetseid kirjeid saab ka täpsemalt uurida. Näiteks on alljärgnevalt toodud käsk kirje *ehr_kood*-iga 120773369 kuvamiseks.

```
eh_ehitised[eh_ehitised$ehr_kood == 120773369,c('id','nimetus',
'seisund','omandi_liik')]
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli probleemse kirje atribuutide *id*, *nimetus*, *seisund* ja *omandi_liik* väärtused.

```
       id nimetus                  seisund omandi_liik
1: 5537838  Kelder EHITIS_SEISUND_KASUTUSEL
```

Tulemusest on näha, et antud kirje puhul on *omandi_liik* väärtustamata ehk antud kirje ei vasta eelnevalt defineeritud andmekvaliteedi reeglile.

4.4 Näide N4 (Õigsus, väärtusvahemiku rikkumine)

Dimensioon: Õigsus.
Probleem: Väärtusvahemiku rikkumine.
Reegli kirjeldus: Ehitise tehniliste andmete kodeerimisel tuleb kasutada tehniliste andmete klassifikaatorit.
Selgitus: Ehitise tehniliste andmete kodeerimisel kasutatakse tehniliste andmete klassifikaatorit. Antud näites testime muutujaid eraldi tabelis esitatud klassifikaatori vastu. Ehitise tehnilised andmed on tabelis *eh_tehna*. Ehitise tehniliste andmete klassifikaator on tabelis *kl_tehna*. *eh_tehna* tabeli atribuut *tena_id* esitab seost *kl_tehna* tabeli atribuudiga *id* (*eh_tena.tena_id* <-> *kl_tehna.id*).

```
tech_classifier_rules <- validator(
 tech_data_conformance = tena_id %in% codelist
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised käsud.

```
check <- confront(eh_tehna, tech_classifier_rules, ref = list(codelist =
kl_tehna$id), key="tena_id")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
                  name    items   passes fails nNA error warning expression
1 tech_data_conformance 19114165 19114165     0   0 FALSE   FALSE tena_id …
```

Tulemustest on näha, et ühtegi reegli rikkumist ei tuvastatud ning järelikult on tehniliste andmete kodeerimisel kasutatud korrektselt tehniliste andmete klassifikaatorit.

4.5 Näide N5 (Õigsus, süntaksi rikkumine)

Dimensioon: Õigsus.
Probleem: Süntaksi rikkumine
Reegli kirjeldus: Kuupäev peab vastama etteantud vormingule: Ehituse alustamise kuupäev (*eh_alust_kp*) peab olema vormingus yyyy-MM-dd ning kirje loomise kuupäev (*date_created)* vormingus yyyy-MM-dd'HH:mm:ss.SSSSSS.
Selgitus: Kuupäevad ehitiste tabelis *eh_ehitised* peavad vastama etteantud vormingule. Atribuudi *eh_alust_kp* kuupäeva õige vorming on yyyy-MM-dd (näiteks 2008-09-20) ja *date_created* õige vorming on yyyy-MM-dd'HH:mm:ss.SSSSSS (näiteks 2019-10-14 14:12:46.666714). Et kontrollida kuupäevade vormingule vastamist loome reeglite komplekti *date_rules,* kuhu lisame reegli *start_date_syntax* atribuudi *eh_alust_kp* väärtuste kontrollimiseks ning reegli *created_date_syntax* atribuudi *date_created* väärtuste kontrollimiseks.

```
date_rules <- validator(
 start_date_syntax = grepl("^[0-9]{4}-[0-9]{2}-[0-9]{2}$",
eh_alust_kp),
 created_date_syntax = grepl("^[0-9]{4}-[0-9]{2}-[0-9]{2}\ [0-
9]{2}\\:[0-9]{2}\\:[0-9]{2}\\.[0-9]{6}$", date_created)
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised käsud.

```
check <- confront(eh_ehitised, date_rules, key="ehr_kood")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
                 name   items  passes  fails nNA error warning expression
1   start_date_syntax 1058629   59942 998687   0 FALSE   FALSE grepl("^[0-…
2 created_date_syntax 1058629 1055767   2862   0 FALSE   FALSE grepl("^[0-…
```

Tulemustest on näha, et reeglile start_date_syntax ei vasta 998 687 kirjet ning reeglile created_date_syntax ei vasta 2862 kirjet. Järgnevalt vaatame taas lähemalt millised kirjed ei vasta reeglile. Selleks konverteerime esmalt tulemuse kokkuvõtte sobivale kujule ning filtreerime tulemused nii, et alles jäävad vaid reeglitele mittevastavad kirjed.

```
output <- as.data.frame(check)
output[output$value == FALSE,]
```

Käskude käivitamise tulemusena kuvatakse RStudio konsoolis ülevaade probleemsetest kirjetest.

```
         ehr_kood         name value                                expression
      1: 120231238 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
      2: 115011733 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
      3: 108034400 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
      4: 108028019 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
      5: 120295322 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
     ---
 623331: 121293239 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
 623332: 221296656 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
 623333: 221303569 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
 623334: 121303570 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
 623335: 121303683 pattern_year FALSE grepl("^[0-9]{4}$", esmane_kasutus)
```

Tuvastatud konkreetseid kirjeid saab ka täpsemalt uurida. Näiteks on alljärgnevalt toodud käsk kirje *ehr_kood*-iga 121303683 kuvamiseks.

```
eh_ehitised[eh_ehitised$ehr_kood == 121303683,c('id','eh_alust_kp',
'date_created')]
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli probleemse kirje atribuutide *id, eh_alust_kp* ja *date_created* väärtused. Nagu näha on antud probleemse kirje puhul eh_alust_kp väärtustamata.

```
        id eh_alust_kp              date_created
1: 5626327             2019-09-09 08:09:22.756168
```

4.6 Näide N6 (Õigsus, süntaksi rikkumine)

Dimensioon: Õigsus.
Probleem: Süntaksi rikkumine.
Reegli kirjeldus: Esmase kasutuse aasta talletamiseks peab olema kasutatud neljakohalist aastanumbrit (vastama etteantud vormingule: yyyy).
Selgitus: Tabelis *eh_ehitised* veerus *esmane_kasutus* on talletatud esmase kasutuse kuupäev ja kellaaeg. Kuupäeva süntaktilise õigsuse kontrollimiseks loome reeglite komplekti *pattern_rules*, kuhu lisame reegli *pattern_year.* Antud reegel kontrollib, et kasutatud oleks neljakohalist aastanumbrit.

```
pattern_rules <- validator(
 pattern_year = grepl("^[0-9]{4}$", esmane_kasutus)
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised käsud.

```
check <- confront(eh_ehitised, pattern_rules, key="id")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
          name   items passes  fails nNA error warning expression
1 pattern_year 1058629 435294 623335   0 FALSE   FALSE grepl("^[0-9]…
```

Saadud tulemusest on näha, et 623 335 kirjet ei vasta kontrollitavale reeglile. Konkreetsete probleeme saab lähemalt uurida näidetes N2 ja N4 näidatud viisil.

4.7 Näide N7 (Reeglipärasus, valdkonnakitsenduse rikkumine)

Osad reeglid kirjeldavad kitsendusi erinevate andmeveergude vahel. Need kitsendused on kasulikud ärireeglite esitamisel. Antud näide illustreerib sellist juhtu.

Dimensioon: Reeglipärasus.
Probleem: valdkonnakitsenduse rikkumine.
Reegli selgitus: Alates 01.01.2020 on liginullenergia nõue kohustuslik (v.a. erandid) üle 220 m² väikeelamutele. Liginullenergiahoone nõuet ei kohaldata väikeelamule köetava pinnaga kuni 220 m² (kehtivad madalenergia ehk B klassi nõuded). Samuti ei rakendata liginullenergianõuet (ka eramule suurema köetava pinnaga kui 220 m²) juhul, kui päikeseenergiasüsteemi paigaldamine ei ole majanduslikult põhjendatud või tehniliselt teostatav.
Selgitus: Hoonete energiatõhusust reguleerivad kolm määrust[13,14,15]. Neil määrustel põhinev reegel esitab hulga tingimusi, millele *eh_ehitised* tabelis olevad kirjed peavad vastama. Et kontrollida kirjeldatud reeglile vastavust konverteerime atribuudi *esmane_kasutus* numbriliseks väärtuseks. Selle tegevuse eesmärgiks on muuta võimalikuks andmete võrdlemine kasutades validator objekti. Järgmiseks loome reeglite komplekti *energy_class_rules*. Sinna lisame reegli *mandatory_energy_class*, mis kontrollib kõiki reeglis toodud tingimusi: kui esmane kasutus on 2020 või hilisem, hoone tüüp on väikeelamu ning köetav pind on suurem kui 220 m² peab ehitise energiaklass olema A.

Esimese sammuna konverteerime atribuudi *esmane_kasutus* numbriliseks väärtuseks.

```
eh_ehitised$esmane_kasutus <- as.numeric(eh_ehitised$esmane_kasutus)
```

Järgmiseks loome eelnevalt kirjeldatud reeglite komplekti energy_class_rules.

```
energy_class_rules <- validator(
 mandatory_energy_class = if (esmane_kasutus >= 2020 & hoone_tyyp ==
'VAIKEELAMUD' & koetav_pind > 220) energia_klass == 'ENERGIAKL_A'
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised kaks käsku.

[13] https://www.riigiteataja.ee/akt/122082019002
[14] https://www.riigiteataja.ee/akt/122082019005
[15] https://www.riigiteataja.ee/akt/122082019004

```
check <- confront(hoone_energia_margised, energy_class_rules, key="id")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
                    name items passes fails nNA error warning expression
1 mandatory_energy_class 30649  30644     1   4 FALSE   FALSE !( …
```

Tulemusest on näha, et üks kirje ei vasta reeglile ning neli kirjet on väärtustamata (nNA). Konkreetsete probleeme saab lähemalt uurida näidetes N2 ja N4 näidatud viisil.

4.8 Näide N8 (Õigsus, valdkonnakitsenduse rikkumine)

Dimensioon: Reeglipärasus
Probleem: Valdkonnakitsenduse rikkumine.
Reegli kirjeldus: Kui ehitise köetav pind on määratud, siis peab see olema väiksem või võrdne kasuliku pinnaga.
Selgitus: Määruses[16] "Ehitise tehniliste andmete loetelu ja pindade arvestamise alused" on kirjeldatud hoone köetav pind järgnevalt:

- § 28(2) Hoone köetav pind on hoone kõigi sisekliima tagamisega ruumide suletud netopindade summa;
- § 22(1) Korruse suletud netopind ehk kasulik pind on korruse suletud brutopind, millest on maha arvatud korruse välistarindite alune pind, sisetarindite alune pind ja mittekandvate tarindite alune pind;
- § 22(2) Hoone suletud netopind on kõigi korruste suletud netopindade summa.

Sellest tulenevalt kontrollime järgnevalt, et kui köetav pind on olemas, siis see on väiksem või võrdne kasuliku pinnaga.

```
conditional_rules <- validator(
 condition1 = if (!is.na(koetav_pind)) koetav_pind <= kasulik_pind
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised käsud.

```
check <- confront(eh_ehitised, conditional_rules, key="id")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
       name   items passes fails   nNA error warning  expression
1 condition1 1058629 993968 10145 54516 FALSE   FALSE  !(!is.na(…
```

Tulemusest on näha, et üks 10 145 kirjet ei vasta reeglile ning neli kirjet on väärtustamata (nNA). Konkreetsete probleeme saab lähemalt uurida näidetes N2 ja N4 näidatud viisil.

[16] https://www.riigiteataja.ee/akt/107102014003

4.9 Näide N9 (Reeglipärasus, funktsionaalse sõltuvuse rikkumine)

Dimensioon: Reeglipärasus.
Probleem: funktsionaalse sõltuvuse rikkumine.
Reegli kirjeldus: Omavalitsus on seotud vaid ühe maakonnaga.
Selgitus: Kontrollimaks, et omavalitsus on seotud vaid ühe maakonnaga loome reeglite komplekti *dependency_rules*. Sinna lisame reegli *dependency_county*, mis kirjeldab funktsionaalset sõltuvust omavalitsuse ja maakonna vahel.

```
dependency_rules <- validator(
 dependency_county = omavalitsus ~ maakond
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised käsud.

```
check <- confront(eh_ehitised, dependency_rules, key="ehr_kood")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
               name   items  passes fails nNA error warning expression
1 dependency_county 1058629 1002331 56298   0 FALSE   FALSE omavalitsus …
```

Tulemused näitavad, et 1058629-st kirjest ei ole 56298-s kirjes omavalitsus üheselt maakonnaga seotud.

Järgmiseks vaatame lähemalt milline kirje ei vasta reeglile. Selleks tuleb käivitada järgnev käsk.

```
output <- as.data.frame(check)
output[output$value == FALSE,]
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev ülevaade kirjetest.

```
        ehr_kood              name value            expression
    1: 114011772 dependency_county FALSE omavalitsus ~ maakond
    2: 114011505 dependency_county FALSE omavalitsus ~ maakond
    3: 220838926 dependency_county FALSE omavalitsus ~ maakond
    4: 121309384 dependency_county FALSE omavalitsus ~ maakond
    5: 221309392 dependency_county FALSE omavalitsus ~ maakond
   ---
56295: 121302502 dependency_county FALSE omavalitsus ~ maakond
56296: 121300602 dependency_county FALSE omavalitsus ~ maakond
56297: 221300400 dependency_county FALSE omavalitsus ~ maakond
```

Vaatame viimast kirjet lähemalt. Selleks tuleb käivitada järgnev käsku.

```
eh_ehitised[eh_ehitised$ehr_kood == 121303567,c('id','nimetus',
'maakond', 'omavalitsus')]
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
       id    nimetus maakond omavalitsus
1: 5626247 Üksikelamu      68         809
```

Järgmiseks vaatame millist maakonna koodi kasutatakse veel sama omavalitsuse kirjetes.

```
eh_ehitised[eh_ehitised$omavalitsus == 809, c('id','nimetus',
'maakond', 'omavalitsus')]
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
             id    nimetus maakond omavalitsus
    1: 3162780    alajaam      67         809
    2: 2501880      Elamu      68         809
    3: 2531178 Üksikelamu      68         809
    4: 5548142 Üksikelamu      68         809
    5: 5630137   Puurkaev      68         809
   ---
15588: 5624925 üksikelamu      68         809
15589: 5624926   Puurkaev      68         809
15590: 5629827       Kuur      68         809
15591: 5625477    Aiamaja      68         809
15592: 5626247 Üksikelamu      68         809
```

Tulemustest selgub, et omavalitsuse 809 (Tori vald) maakonnaks on andmetes 67 ja 68. Tegelikkuses peab seal olema vaid 68 (Pärnu maakond).

4.10Näide N10 (Ajakohasus, aegunud väärtus)

Dimensioon: Ajakohasus.
Probleem: vale väärtus (aegunud).
Reegli kirjeldus: Ehitisregistrisse imporditud aadresside lühiaadress ja täisaadress peab vastama aadressiandmete süsteemis hallatavate aadresside samadele andmeelementidele.
Selgitus: Antud näites võrdleme sama andmehulga eri versioone tuvastamaks lokaalse koopia tabelis *ad_aadress* ajakohasust võrreldes tabeliga *aadressid*.

Kõigepealt mestime kahe andmehulga (*ad_address* ja *aadressid*) aadresside *adr_id* väärtuse alusel ja loome referentstabeli *ads_aadress*, mis sisaldab vaid lähiaadressi *(lahiaadress)* ja täisaadressi *(taisaadress)* kirjeid.

```
adsdiff = merge(x=ad_aadress, y=aadressid, by.x="adr_id",
by.y="ADR_ID", all.x=TRUE)
ads_aadress <- adsdiff[,c('LAHIAADRESS','TAISAADRESS')]
colnames(ads_aadress) <- c('lahiaadress', 'taisaadress')
ehr_aadress <- adsdiff[,c('lahiaadress', 'taisaadress')]
```

Seejärel loome reeglid, mis kontrollivad vastavalt kas andmetabeli elemendid *lahiaadress* ja *taisaadress* vastavad referentstabeli (*ads_reference*) analoogsetele väärtustele.

```
timelyness_rules <- validator(
 timelyness_lahiaddress = lahiaadress == ads_reference$lahiaadress,
 timelyness_taisaddress = taisaadress == ads_reference$taisaadress
)
```

Tabeli *ehr_aadress* reeglitele vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb käivitada järgmised käsud, millega ütleme, et *ehr_aadress* tabeli referentstabelina (*ads_reference*) kasutatakse *ads_aadress* andmeid.

```
check <- confront(ehr_aadress, timelyness_rules, ref =
list(ads_reference = ads_aadress))
summary(check)
```

Käivitatud käsu tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
                    name   items  passes fails   nNA error expression
1 timelyness_lahiaddress 2464347 2413787    77 50483 FALSE lahi…
2 timelyness_taisaddress 2464347 2413785    79 50483 FALSE tais…
```

4.11 Näide N11 (Õigsus, süntaksi rikkumine)

Antud näites kontrollime kuupäevade süntaksile vastavust mitme muutuja puhul korraga, laiendades näitest N5 tuttava reegli *start_date_syntax* kasutust.

Dimensioon: Õigsus.
Probleem: Süntaksi rikkumine
Reegli kirjeldus: Kuupäev peab vastama etteantud vormingule: Ehituse alustamise kuupäev (*eh_alust_kp*), kultuurimälestiseks muutmise aeg (*kultuurimalestis_kp*), kavandatav kasutusele võtmise aeg (*kav_kasutus_kp*), ajutise ehitise kasutuse lõpp (*ajeh_kasutlopp_kp*) ja ajutise ehitise kasutusele võtmise algus (*ajeh_kasutalg_kp*) peavad olema vormingus yyyy-MM-dd.
Selgitus: Kuupäevad ehitiste tabelis *eh_ehitised* peab vastama etteantud vormingule. Kuupäeva õige vorming on yyyy-MM-dd (näiteks 2008-09-20). Antud näiteks kontrollitakse atribuute *eh_alust_kp, kultuurimalestis_kp, kav_kasutus_kp, ajeh_kasutlopp_kp* ja *ajeh_kasutalg_kp.* Et kontrollida kuupäevade vormingule vastamist loome reeglite komplekti *date_rules,* kuhu lisame reegli *start_date_syntax.*

```
date_rules <- validator(
eh_alust_kp_syntax = grepl("^[0-9]{4}-[0-9]{2}-[0-9]{2}$",
eh_alust_kp),
kultuurimalestis_kp_syntax = grepl("^[0-9]{4}-[0-9]{2}-[0-9]{2}$",
kultuurimalestis_kp),
kav_kasutus_kp_syntax = grepl("^[0-9]{4}-[0-9]{2}-[0-9]{2}$",
kav_kasutus_kp),
ajeh_kasutlopp_kp_syntax = grepl("^[0-9]{4}-[0-9]{2}-[0-9]{2}$",
ajeh_kasutlopp_kp),
ajeh_kasutalg_kp_syntax = grepl("^[0-9]{4}-[0-9]{2}-[0-9]{2}$",
ajeh_kasutalg_kp)
)
```

Nagu eeltoodud näitest näha kasvab atribuutide lisandumisel oluliselt vajaminevat koodiridade arv. Sellise olukorra vältimiseks saab alternatiivina kasutada lühemat ja loetavamat, kuid samaväärset käsku.

```
date_rules <- validator(
 G := var_group(eh_alust_kp, kultuurimalestis_kp, kav_kasutus_kp,
ajeh_kasutlopp_kp, ajeh_kasutalg_kp), grepl("^[0-9]{4}-[0-9]{2}-[0-
9]{2}$", G)
)
```

Reeglile vastavuse kontrollimiseks ja tulemuste kuvamiseks tuleb endiselt käivitada järgmised käsud.

```
check <- confront(eh_ehitised, date_rules, key="ehr_kood")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
  name   items passes   fails nNA error warning expression
1 V2.1 1058629  59942  998687   0 FALSE   FALSE grepl("^[0-9]{4}…
2 V2.2 1058629      4 1058625   0 FALSE   FALSE grepl("^[0-9]{4}…
3 V2.3 1058629 118499  940130   0 FALSE   FALSE grepl("^[0-9]{4}…
4 V2.4 1058629   6015 1052614   0 FALSE   FALSE grepl("^[0-9]{4}…
5 V2.5 1058629      2 1058627   0 FALSE   FALSE grepl("^[0-9]{4}…
```

5 Andmekvaliteedi juhtimislaua loomine

5.1 Üldist

Eeltoodud reeglite näidatud viisil realiseerimise tulemusi on lihtne esitada andmekvaliteedi juhtimislaual. Juhtimislaua loomise eesmärgiks on tuua välja andmete kvaliteediga seotud kitsaskohad ning kommunikeerida edasiminekuid. Vaata ka juhtimislauda näidet aadressilt:

https://datastudio.google.com/u/0/reporting/e34c9f7b-7c05-4870-a5b7-2cb1e6b6299c/page/FKO9.

Allpool kirjeldame sellise juhtimislaua loomist, mis aitab anda vastused järgmistele küsimustele:

- Kui suur osa hallatud andmeobjektidest vastab andmekvaliteedi nõuetele?
- Millised andmeelemendid ja mil määral on kaetud andmekvaliteedi reeglitega?
- Millised andmekvaliteedi reeglid on kirjeldatud?
- Milliste andmeobjektide kvaliteeti on vaja tõsta?

5.2 Andmeobjektide vastavus nõuetele

Selleks, et visualiseerida kõikide andmeobjektide nõuetele vastavust tuleb kokku koondada kõik andmekvaliteedi reeglid. Antud puhul koondame kokku ehitiste andmeobjektide kohta käivad reeglid (*eh_ehitised_rules*).

```
eh_ehitised_rules <- existence_rules + uniqueness_rules +
ehitised_rules + pattern_rules + conditional_rules + dependency_rules
```

Ehitise andmete vastavust eelnevalt kirjeldatud reeglitele on võimalik korraga kontrollida käivitades järgnevalt toodud käsud.

```
check <- confront(eh_ehitised, eh_ehitised_rules, key="ehr_kood")
summary(check)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsooli järgnev tulemuste kokkuvõte.

```
                   name   items  passes  fails   nNA error warning
1             exists_id 1058629 1058629      0     0 FALSE   FALSE
2             unique_id 1058629 1058629      0     0 FALSE   FALSE
3         property_type 1058629 1057500   1129     0 FALSE   FALSE
4            state_type 1058629 1058628      1     0 FALSE   FALSE
5          pattern_year 1058629  435294 623335     0 FALSE   FALSE
6 condition_koetav_pind 1058629  993968  10145 54516 FALSE   FALSE
7     dependency_county 1058629 1002331  56298     0 FALSE   FALSE
```

Järgmiseks käivitame käsud reeglite tabelikujule viimiseks.

```
output <- as.data.frame(check)
head(output)
```

Käsu käivitamise tulemusena kuvatakse RStudio konsoolis loodud tabelkuju kuut esimest kirjet.

```
    ehr_kood      name value       expression
1: 220258718 exists_id  TRUE !is.na(ehr_kood)
2: 220258678 exists_id  TRUE !is.na(ehr_kood)
3: 221267603 exists_id  TRUE !is.na(ehr_kood)
4: 120741879 exists_id  TRUE !is.na(ehr_kood)
5: 120231238 exists_id  TRUE !is.na(ehr_kood)
6: 120535518 exists_id  TRUE !is.na(ehr_kood)
                     …
```

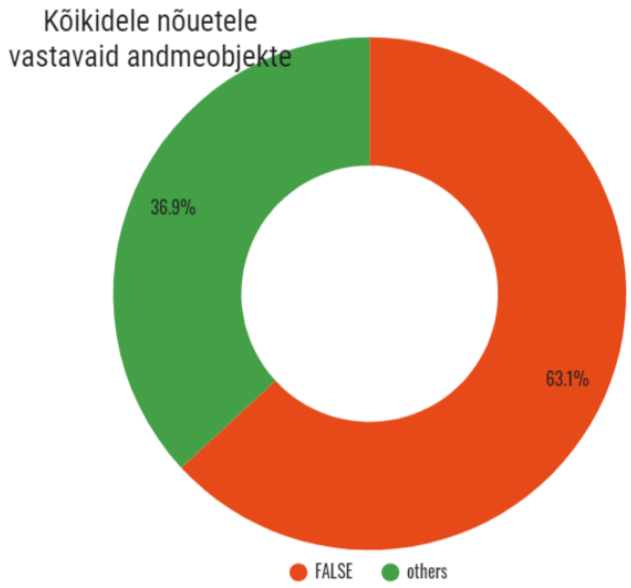
Andmeobjektide kõikidele nõuetele vastavuse visualiseerimiseks võtame näidisena 100 000 kirjet (tuleneb Google Sheets, mida lihtsustamise huvides kasutame, kitsendustest) ning kirjutame need CSV faili nimega data-conformance-long.csv. Selleks tuleb käivitada alljärgnevad käsud.

```
output <- output[sample(.N, 100000)]
write.csv(output[,c('ehr_kood', 'name', 'value')], paste0(filedir,
"data-conformance-long.csv"), row.names = FALSE, na="")
```

Käsu käivitamise tulemusena genereeritud CSV faili saab kasutada nõuetele vastavate reeglite visualiseerimiseks kasutades näiteks MS Excel-it, Google Sheets-i, Google Data Studio-t vmt tarkvara.

Näites ettevalmistatud andmete põhjal koostatud näidisraportiga on võimalik tutvuda siin: https://datastudio.google.com/u/0/reporting/e34c9f7b-7c05-4870-a5b7-2cb1e6b6299c/page/FKO9

Eeltoodud tegevuste tulemusena ettevalmistatud andmete põhjal on Google Data Studio abil koostatud graafik pealkirjaga „Kõikidele nõuetele vastavaid andmeobjekte“, mis on toodud ka alljärgneval joonisel 1.



Joonis 1. Kõikidele nõuetele vastavaid andmeobjekte

5.3 Parandamist vajavad andmeobjektid

Selleks, et saaksime ülevaatlikult tuua välja konkreetsed andmeobjektid, mille kvaliteet vajab tõstmist viime andmed nn laiale kujule, milleks tuleb käivitada järgmised käsud.

```
output <- as.data.frame(check)
data_wide <- spread(output[, c('ehr_kood', 'name', 'value')], name, value)
```

Seejärel võtame näitlikustamiseks 60000 suvalist kirjet ning kirjutame need CSV faili nimega data-conformance-wide.csv.

```
data_wide <- data_wide[sample(.N, 60000)]
write.csv(data_wide, paste0(filedir, "data-conformance-wide.csv"),
row.names = FALSE, na="")
```

Genereeritud CSV faili saab kasutada andmekvaliteedile vastavuse visualiseerimiseks kasutades näiteks MS Excel-it, Google Sheets-i, Google Data Studio-t vmt tarkvara.

Näidistulemusi on võimalik grupeeritult vaadata näidisjuhtimislaualt (Joonis2):

https://datastudio.google.com/u/0/reporting/e34c9f7b-7c05-4870-a5b7-2cb1e6b6299c/page/H0bVB

ehr_kood	exists_id	unique_id	conditio...	depende...	property_ty...	pattern_year	state_type
101000002	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
101000021	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
101000051	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
101000072	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
101000114	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
101000134	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
101000151	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
101000193	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
101000196	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
101000243	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
101000278	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
101000282	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
101000285	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
101000318	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

Joonis 2. Ehitise andmete vastavus andmekvaliteedi reeglitele

5.4 Millised reeglid on kaetud?

Reeglitest ülevaate saamiseks saab neid esitada tabelkujul (Joonis 3), tuues välja milliste ärimõistete, andmetabelite ja andmekvaliteedi dimensioonidega on nad seotud. Vaata ka: https://datastudio.google.com/u/0/reporting/e34c9f7b-7c05-4870-a5b7-2cb1e6b6299c/page/TWQLB

Id	Ärimõiste	Andmetabel	Dimensio...	Probleemi liik	Reegli kirjeldus	Reegel (R)
exists_id	Ehitis	ehr.eh_ehitised	Täielikkus	Puuduv väärtus	Igal ehitisel peab olema määratud tema tunnus (ehitis...	is_unique(ehr_kood)
unique_id	Ehitis	ehr.eh_ehitised	Ühekordsus	Unikaalse väärtuse rikkumine	Iga ehitise tunnus (ehitisregistri kood) peab olema uni...	!is.na(ehr_kood)
property_type	Ehitis	ehr.eh_ehitised	Õigsus	Väärtusvahemiku rikkumine	Ehitise omandi liik vastab etteantud loendile	omandi_liik %in% c('E...
state_type	Ehitis	ehr.eh_ehitised	Õigsus	Väärtusvahemiku rikkumine	Ehitise seisund vastab etteantud loendile	seisund %in% c('EHIT...
tech_data_conforman...	Ehitise tehnilise...	ehr.eh_tehna	Õigsus	Väärtusvahemiku rikkumine	Ehitise tehniliste andmete kodeerimisel kasutatakse t...	tena_id %in% codelist
start_date_syntax	Ehitis	ehr.eh_ehitised	Õigsus	Süntaksi rikkumine	Ehituse alustamise ajatempel peab olema esitatud va...	grepl('^[0-9]{4}-[0-9]{2...
created_date_syntax	Ehitis	ehr.eh_ehitised	Õigsus	Süntaksi rikkumine	Kirje loomise kuupäev peab olema esitatud vastavalt ...	grepl('^[0-9]{4}-[0-9]{2...
pattern_year	Ehitis	ehr.eh_ehitised	Õigsus	Süntaksi rikkumine	Esmase kasutamise aastaarv peab vastama ettenäht...	grepl('^[0-9]{4}$', es...
mandatory_energy_cla...	Energiamärgis	ehr.hoone_energia_margis...	Reeglipäras...	Valdkonnakitsenduse rikkumine	Alates 01.01.2020 on liginullenergia nõue kohustuslik ...	if (esmane_kasutus =...
condition_koetav_pind	Ehitis	ehr.eh_ehitised	Reeglipäras...	Valdkonnakitsenduse rikkumine	Kui kõetav pind on olemas, siis koetav_pind <= kasulik...	if (!is.na(koetav_pind)...
dependency_county	Ehitis	ehr.eh_ehitised	Reeglipäras...	Funktsionaalse sõltuvuse rikkum...	Omavalitsus saab olla vaid ühe maakonna koosseisus	omavalitsus ~ maak...
timelyness_lahiaddress	Aadress	ehr.ad_aadress	Ajakohasus	Aegunud väärtus	EHR'i lähiaadress erineb ADS lähiaadressist	lahiaadress == ads_r...
timelyness_taisaddress	Aadress	ehr.ad_aadress	Ajakohasus	Aegunud väärtus	EHR'i täisaadress erineb ADS täisaadressist	taisaadress == ads_r...

Joonis 3. Kaetus andmekvaliteedi reeglitega

5.5 Andmeelementide kaetus reeglitega

Et teha kindlaks kas kõik olulised andmeelemendid on kaetud andmekvaliteedi reeglitega, koondame esmalt kokku kõik eelnevald loodud reeglid.

```
all_rules <- existence_rules + uniqueness_rules + ehitised_rules +
tech_classifier_rules + date_rules + pattern_rules + energy_class_rules
+ conditional_rules + dependency_rules + timelyness_rules
```

Reeglite ja muutujate maatrikskujul vaatamiseks tuleb käivitada alljärgnev käsk.

```
variables(all_rules, as='matrix')
```

Käsu tulemusena kuvatakse atribuutide kaetus reeglitega maatriksina. Allpool on näidisena toodud 4 esimest.

```
rule                    ehr_kood omandi_liik seisund tena_id codelist
  exists_id                 TRUE       FALSE   FALSE   FALSE    FALSE
  unique_id                 TRUE       FALSE   FALSE   FALSE    FALSE
  property_type            FALSE        TRUE   FALSE   FALSE    FALSE
  state_type               FALSE       FALSE    TRUE   FALSE    FALSE
  tech_data_conformance    FALSE       FALSE   FALSE    TRUE     TRUE
                                          …
```

Järgmiseks tuvastame need andmekvaliteedi veerud, mille koha andmekvaliteedi reeglid puuduvad. Selleks tuleb käivitada alljärgnev käsk.

```
names(eh_ehitised)[!names(eh_ehitised) %in%
variables(eh_ehitised_rules)]
```

Käsu käivitamise tulemusena kuvatakse atribuudid ehk veerud, mis pole veel reeglitega kaetud.

```
[1] "id"                        "ehit_id"                       …
 [6] "rajatis_hoone"             "max_korruste_arv"             …
[11] "kav_kasutus_kp"            "ajeh_kasutalg_kp"             …
[16] "energia_valjast_kp"        "energia_margise_kp"           …
[21] "date_created"              "date_updated"                 …
[26] "mitteeluruumide_arv"       "korterite_arv"                …
[31] "kultuurimalestis_viit"     "abs_0_korgus"                 …
[36] "elamispind"                "eluruumi_pind"                …
[41] "laius"                     "lift"                         …
[46] "pind"                      "suletud_netopind"             …
[51] "abiruumide_pind"           "toimiku_nr"                   …
```

Eelnev võimaldab küll analüüsida andmeelementide kaetust reeglitega, kuid pole mugav viis ülevaate saamiseks. Parema ülevaate saamiseks võib samasid andmeid esitada andmekvaliteedi juhtimislaual. Selleks tuleb käivitada järgmised andmeid ettevalmistavad käsud.

```
df <- data.frame(variables(all_rules, as='matrix'))
df[names(eh_ehitised)[!names(eh_ehitised) %in%
variables(eh_ehitised_rules)]]=FALSE
t_df <- transpose(df)
colnames(t_df) <- rownames(df)
rownames(t_df) <- colnames(df)
write.csv(t_df, paste0(filedir, "variable-coverage-with-
rules.csv"), row.names = TRUE, na="")
```

Käskude tulemusena genereeritakse varem vaadeldud reeglitega kaetuse andmed maatrikskujul CSV faili nimega variable-coverage-with-rules.csv. Nimetatud failis olevaid andmeid saab kasutada juhtimislaua koostamisel (Joonis 424). Juhtimislaua näide on juurdepääsetav siit:

https://datastudio.google.com/u/0/reporting/e34c9f7b-7c05-4870-a5b7-2cb1e6b6299c/page/yISVB

field_name	exists...	unique_...	dependenc...	property_t...	tech_data_confo...	timelynes...	state_type	pattern_y...	mandatory_...	condition_koetav_pind
ehr_kood	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
omandi_liik	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
seisund	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
tena_id	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
codelist	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
eh_alust_kp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
kultuurimalestis_kp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
kav_kasutus_kp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
ajeh_kasutlopp_kp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
ajeh_kasutalg_kp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
esmane_kasutus	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE
hoone_tyyp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
koetav_pind	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
energia_klass	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

Joonis 42. Juhtimislaua koostamiseks kasutatavad andmed

5.6 Reeglite eksport ja import

Võib tekkida olukordi, kus kirja pandud reegleid on vaja jagada. Sellisel juhul on reeglid võimalik eksportida reeglite faili käivitades alljärgnevalt toodud käsud. Käskude käivitamisel genereeritakse yml formaadis fail nimega rules.yml.

```
rule_filename <- paste0(filedir, "rules.yml")
export_yaml(all_rules, file=rule_filename)
```

Võib tekkida ka olukord, kus reegild on vaja failist sisse lugeda. Seda saab teha järgneva käsuga.

```
rules2 <- validator(.file=rule_filename)
```

6 Soovituslikud töövahendid

Järgnevalt on toodud nimekiri tööriistades, mis on abiks andmekvaliteediga seotud eri töölõikude realiseerimisel.

Profileerimine:

- RStudio (https://rstudio.com/products/rstudio/)
- R paketid profileerimise toestamiseks:
  - DataExplorer R pakett (https://cran.r-project.org/web/packages/DataExplorer/)
  - dlookr R pakett (https://cran.r-project.org/web/packages/dlookr/)

Kvaliteedi mõõtmine:

- Validate R pakett (https://cran.r-project.org/web/packages/validate/)

Andmekvaliteedi reeglite haldamine:

- Confluence (https://www.atlassian.com/software/confluence)
- Jira (https://www.atlassian.com/software/jira)