odataR for Statistics Netherlands data

Han Oostdijk

2019-11-13

Introduction

The package was made because I am interested in the information of Statistics Netherlands.
The easiest tool method to do an adhoc query is using the Statline interface in the Dutch or English language. It offers the possibility to pivot the various dimensions of the information and download the result to a csv (comma separated values) file or an Excel spreadsheet. And then the csv or Excel file can be read into the R environment for further processing.
However especially the handling of the headers is laborious in this way because it is different for each table/layout combination.

Getting the information in R is made easier by the **odataR** package. Using this package avoids the intermediary csv or Excel file. The code for getting the information of a table in a *data.frame* can be as easy as `df=odataR_get_table(table_id='03759ned')`
when the identification code of the table is *03759ned* .

When you would execute this code it would result in a data.frame *df* with 45982080 rows and 7 columns. That is probably more data than information.
In this vignette we will show how to:

- indicate the location (*root*) of the OData structures. The default is set for the data of Statistics Netherlands so you don't have to do anything when you want to use that information. For other sources however you have to set the root.
- use the catalog to find out the available tables and their characteristics.
- find out the topics and dimensions of a table. There is a subtable for each dimension.
- determine with these subtables how to do queries on the dimensions to request more or less exactly the information that is needed.

For a full example for retrieving and using the data of Statistics Netherland we use the following case study.

Case study

For this case study we want to see if the number of persons in my municipality (Amstelveen) in various age groups remains relatively constant. So we need to retrieve information about this and create a plot of the relative group size over various years. For retrieval of the data we use of course the **odataR** package but to report, manipulate and plot the data we use the following packages (**knitr** is used to display the tables in this vignette):

```
library(knitr)
library(ggplot2)
library(magrittr)
library(dplyr)
library(stringr)
```

Indicate where the OData information of Statistics Netherlands is located

The user of the package has to indicate the location (*root*) of the OData structures. Because the package was written with a special interest for the information of Statistics Netherlands it is not necessary to specify the *root* for the standard data of Statistics Netherlands: when no *root* is specified the software will always use the Statistics Netherlands data. But you can always check which *root* will be used with the `odataR_get_root` function:

```
library(odataR)
print(odataR_get_root())
#> [1] "https://opendata.cbs.nl"
```

Indicate where OData information other than that of Statistics Netherlands is located

For other OData information the user of the package has to indicate the location (*root*) with the `odataR_set_root` function:

```
odataR_set_root("https://dataderden.cbs.nl")
print(odataR_get_root())
#> [1] "https://dataderden.cbs.nl"
```

From now on the indicated url will be used as *root* until the moment that it is changed again with the `odataR_set_root` function. This function has as default argument the url of the OData structure of Statistics Netherlands:

```
odataR_set_root()
print(odataR_get_root())
#> [1] "https://opendata.cbs.nl"
```

Find in the catalog which tables are available

We know that the information we need is in the table with identification code *03759ned* that was used in the example above.
But assuming that we don't know this, we have to use the catalog to find out which tables contain information about persons, in regions, with certain ages. The easiest way to find out this information is using the English or Dutch visual web interface. When you have navigated through the catalog and reached the table that contains the necessary information you can read in the catalog all meta data about the table. Because you want to use the **odataR** package it is important to take note of the identifier that you will later use as *table_id*.
Apart from viewing the catalog we can also do a query on the catalog. In the next code section we show how to retrieve the whole catalog in data.frame `cbscat` and to check which describing elements (fields) it contains by looking at the first record in the catalog:

```
cbscat = odataR_get_cat()
(dimcat = dim(cbscat))
#> [1] 4680   26
```

```
str(cbscat[1,]
```

```
#> 'data.frame':    1 obs. of  26 variables:
#>  $ Updated            : chr "2019-11-08T02:00:00"
#>  $ ID                 : int 0
#>  $ Identifier         : chr "83583NED"
#>  $ Title              : chr "Banen van werknemers; bedrijfsgrootte en ec"| __truncated__
#>  $ ShortTitle         : chr "Banen van werknemers; bedrijfsgrootte"
#>  $ ShortDescription   : chr "\nDeze tabel geeft informatie over het gemi"| __truncated__
#>  $ Summary            : chr "Gemiddeld aantal banen van werknemers in de"| __truncated__
#>  $ Modified           : chr "2019-11-08T02:00:00"
#>  $ MetaDataModified   : chr "2019-11-08T02:00:00"
#>  $ ReasonDelivery     : chr "Actualisering"
#>  $ ExplanatoryText    : chr ""
#>  $ OutputStatus       : chr "Regulier"
#>  $ Source             : chr "CBS."
#>  $ Language           : chr "nl"
#>  $ Catalog            : chr "CBS"
#>  $ Frequency          : chr "Perjaar"
#>  $ Period             : chr "2010 december - 2018 december"
#>  $ SummaryAndLinks    : chr "Gemiddeld aantal banen van werknemers in de"| __truncated__
#>  $ ApiUrl             : chr "https://opendata.cbs.nl/ODataApi/OData/83583NED"
#>  $ FeedUrl            : chr "https://opendata.cbs.nl/ODataFeed/OData/83583NED"
#>  $ DefaultPresentation: chr "ts=1572875720081&graphtype=Table&r=Perioden"| __truncated__
#>  $ DefaultSelection   : chr "$filter=((Bedrijfsgrootte eq 'T001097') or "| __truncated__
#>  $ GraphTypes         : chr "Table,Bar,Line"
#>  $ RecordCount        : int 6696
#>  $ ColumnCount        : int 4
#>  $ SearchPriority     : chr "2"
```

From this we see that the catalog has 4680 rows and 26 columns and that the field *ShortDescription* is probably suited to do a query on bevolking (population), leeftijd (age) and regio (region):

```
x = odataR_get_cat(query = paste0(
  "?$filter=substringof('leeftijd',tolower(ShortDescription))",
  "and substringof('regio',tolower(ShortDescription)) ",
  "and substringof('bevolking',tolower(ShortDescription)) ",
  "and substringof('bevolking',tolower(ShortTitle))",
  "&$select=Identifier,Title,ShortTitle,RecordCount,ColumnCount"))
kable(x,caption='results query leeftijd,bevolking,regio')
```

Table 1: results query leeftijd,bevolking,regio

Identifier	Title	ShortTitle
84527NED	Regionale prognose 2020-2050; bevolking, intervallen, regio-indeling 2018	Bevolking; intervallen, re
84525NED	Regionale prognose 2020-2050; bevolking, regio-indeling 2018	Bevolking; leeftijd, regio,
83491NED	Regionale prognose 2017-2040; bevolking, intervallen, regio-indeling 2015	Bevolking; intervallen, 20
83489NED	Regionale prognose 2017-2040; bevolking, regio-indeling 2015	Bevolking; leeftijd, 2017-
70648ned	Bevolking op 1 januari; leeftijd, geboorteland en regio	Bevolking; geboorteland
37713	Bevolking; leeftijd, migratieachtergrond, geslacht en regio, 1 januari	Bevolking; migratieachte
70634ned	Bevolking; geslacht, leeftijd, nationaliteit en regio, 1 januari	Bevolking; nationaliteit,
03759ned	Bevolking op 1 januari en gemiddeld; geslacht, leeftijd en regio	Bevolking; geslacht, leeft
71887ned	Beroepsbevolking; regio's 1996-2013	Beroepsbevolking; regio's
82220NED	Regionale prognose 2014-2040; kerncijfers, regio-indeling 2013	Bevolking; kerncijfers, re
82172NED	Regionale prognose 2014-2040; bevolking, regio-indeling 2013	Bevolking; leeftijd, regio,
81273ned	Regionale prognose bevolkingsopbouw; 2011-2040	Bevolkingsopbouw gemee
80283ned	Regionale prognose bevolkingsopbouw;2009-2040	Bevolkingsopbouw gemee
71548ned	Regionale prognose bevolkingsopbouw; 2007-2025	Bevolkingsopbouw gemee
71188ned	Regionale prognose bevolkingsopbouw; 2005-2025	Bevolkingsopbouw gemee
70233ned	Gemiddelde bevolking; geslacht, leeftijd, burg. staat, regio, 1995-2016	Gem. bevolking;leeftijd,

The results of the query, to which also the extra condition on the field *ShortTitle* was added to limit the number of results, can be found in *Table1*. Note that we did the selection query directly on the catalog and not on `cbscat`. The mean reason for that is showing how this can be done but in general selections done on the webserver are more efficient because the amount of data to transport from webserver to client is decreased in this way. Apart from the filter specification we also included a select

statement in order to decrease the width of the table in this document. Normally such an intermediate table would be excluded from a document and in that case the select specification can be omitted. The *kable* statement serves only to get the result table included in the document. To avoid confusion further calls to *kable* will be hidden in this document.
Of course the table with identification code *03759ned* is in the table.

Find information about table *03759ned*

In the CBS database a table name (table_id) points to a set of subtables that together provide the information. The 'main' subtable *TypedDataset* (or alternatively *UntypedDataset*) contains the topic data with dimensions in coded form indicating where the topic data relates to. The other subtables convey the meaning of the coded dimensions. E.g. a topic field could be the number of married male persons and the dimensions could be region and period.

We see from *Table1* (not in the pdf version of the document) that table *03759ned* has 45982080 rows and 7 columns. So it is worthwhile to create a query that exactly selects the information that is needed.

DataProperties subtable

First we use function `odataR_get_meta` to retrieve subtable *DataProperties* to see which are the topics and dimensions of the main table.

```
props = odataR_get_meta(table_id='03759ned',metatype = 'DataProperties')
x= props %>% select(Position,ParentID,Type,Key,Title)
```

Table 2: properties of table 03759ned (first fields only)

Position	ParentID	Type	Key	Title
0	NA	Dimension	Geslacht	Geslacht
1	NA	Dimension	Leeftijd	Leeftijd
2	NA	Dimension	BurgerlijkeStaat	Burgerlijke staat
3	NA	GeoDimension	RegioS	Regio's
4	NA	TimeDimension	Perioden	Perioden
5	NA	Topic	BevolkingOp1Januari_1	Bevolking op 1 januari
6	NA	Topic	GemiddeldeBevolking_2	Gemiddelde bevolking

In *Table2* we see that there are five dimensions (in position 0 till 4) and 2 topics (the other positions) for which we will (in the case study) consider only the second topic:

- dimension *Geslacht*: dimension for gender
- dimension *Leeftijd*: dimension for ages
- dimension *BurgerlijkeStaat*: dimension for marital state
- dimension *RegioS*: dimension for regions in the Netherlands
- dimension *Perioden*: dimension for periods
- topic *BevolkingOp1Januari_1* : the topic of the total number of persons on January 1 (of the year indicated by the period dimension) in the classes indicated by the others dimensions
- topic *GemiddeldeBevolking_2* : the topic of the average number of persons in the year indicated by the period dimension in the classes indicated by the others dimensions

Because the case study wants to use the number of persons in certain age groups in Amstelveen we first have to check the subtables for the dimensions to see how we can recognize these persons. Remember that one can easily code to retrieve the whole table with `df=odataR_get_table(table_id='03759ned')` but this would result in a *data.frame* of 45982080 rows and 7 columns.

Check the *Geslacht* dimension (gender)

We will not distinguish by gender but we need the code for all genders together i.e. `T001038`. By executing `x = odataR_get_meta(table_id='03759ned',metatype = 'Geslacht')` we see the various genders . *Table3* shows these.

Table 3: keys for the genders

Key	Title	Description	CategoryGroupID
T001038	Totaal mannen en vrouwen		NA
3000	Mannen		NA
4000	Vrouwen		NA

Check the *Leeftijd* dimension (ages and agegroups)

By executing `x = odataR_get_meta(table_id='03759ned',metatype = 'Leeftijd')` we see the age groups. *Table4* shows the first five of these. Note that key `10000` is the total for all the age groups.

Table 4: keys for the age groups

Key	Title	Description	CategoryGroupID
10000	Totaal		1
10010	0 jaar		2
10100	1 jaar		2

Key	Title	Description	CategoryGroupID
10200	2 jaar		2
10300	3 jaar		2

Check the ***BurgerlijkeStaat*** dimension (marital status)

We will not distinguish by marital status but we need the code for the total together i.e. `T001019`. By executing `x = odataR_get_meta(table_id='03759ned',metatype = 'BurgerlijkeStaat')` we see the various genders . *Table5* shows the first three fields (with Description field capped at 40 characters).

Table 5: keys for marital status

Key	Title	Description
T001019	Totaal burgerlijke staat	Burgerlijke staat: Formele positie van
1010	Ongehuwd	Vanaf 2010: burgerlijke staat die aangee
1020	Gehuwd	Vanaf 1998: wettig gehuwd plus partnersc
1050	Verweduwd	Vanaf 2010: verweduwd na wettig huwelijk
1080	Gescheiden	Vanaf 2010: gescheiden na wettig huwelij

Check the ***RegioS*** dimension (regions)

We can do a query on the *RegioS* subtable to try to find the code that is used for the municipality Amstelveen. We see in *Table6* that we will have to use key 'GM0362'.

```
x = odataR_get_meta(table_id='03759ned',metatype ='RegioS',
        query="?$filter=substringof('amstelveen',tolower(Title))") %>%
  select(Key,Title)
```

Table 6: key(s) for region Amstelveen

Key	Title
GM0362	Amstelveen

Check the ***Perioden*** dimension (periods)

By executing `x = odataR_get_meta(table_id='03759ned',metatype = 'Perioden')` we see in the results (only the first five are shown in *Table7*) the codes that are used for the periods.

Table 7: keys for periods

Key	Title	Description	Status
1988JJ00	1988	NA	Definitief
1989JJ00	1989	NA	Definitief
1990JJ00	1990	NA	Definitief
1991JJ00	1991	NA	Definitief
1992JJ00	1992	NA	Definitief

Compose the query to get the required data

So we now have the information to do a precise query for the information we need:

```
Aveen= odataR_get_table(table_id='03759ned',
  query=paste0(
    "?$filter=startswith(RegioS,'GM0362') ",
    "and Leeftijd ne '10000' " ,
    "and Geslacht eq 'T001038' ",
    "and BurgerlijkeStaat eq 'T001019'",
   "&$select=Perioden,Leeftijd,GemiddeldeBevolking_2"
  ))
```

Data.frame *Aveen* has 3424 rows and 3 columns. Remember the dimensions of the full table: 45982080 rows and 7 columns.

The remainder of the vignette shows how this information can be plotted.

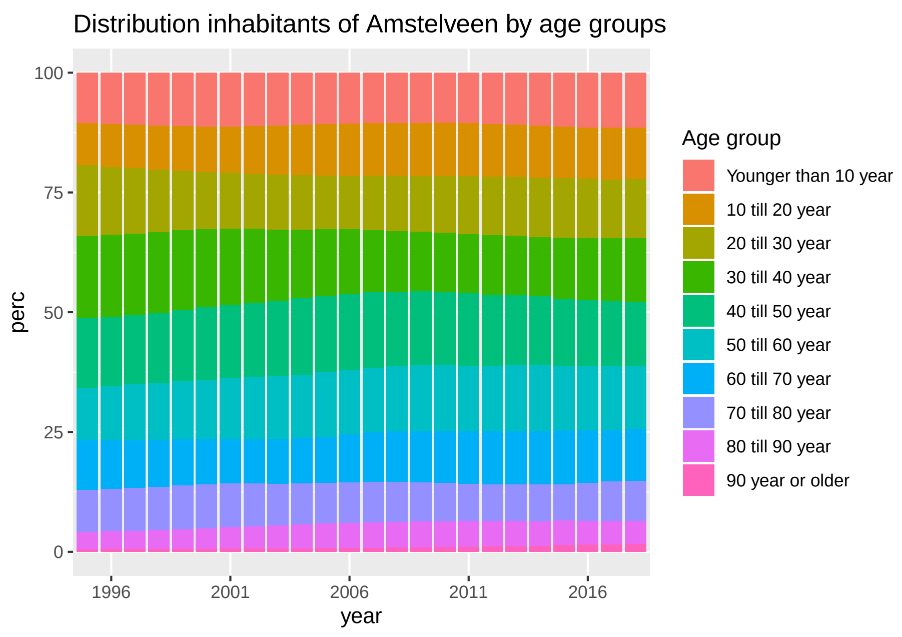
Determine the distribution per period over age groups

Per year we want to show the relative size of each age group. First we will translate the Dutch terms to English and calculate the number of persion by age group. In the second step we calculate the total number of persons in a year and in the third step we merge this total to the first table so that we can calculate percentages. In the last step we also ensure the correct sorting order by using the `factor` function.

```
av1 = Aveen %>%
  select(period=Perioden,age=Leeftijd,persons=GemiddeldeBevolking_2) %>%
  filter(!is.na(persons)) %>%
  mutate(
         age = as.numeric(stringr::str_extract(age,"\\d{1,3}")),
         ageg = case_when(
    age < 10  ~ "Younger than 10 year",
    age < 20  ~ "10 till 20 year",
    age < 30  ~ "20 till 30 year",
    age < 40  ~ "30 till 40 year",
    age < 50  ~ "40 till 50 year",
    age < 60  ~ "50 till 60 year",
    age < 70  ~ "60 till 70 year",
    age < 80  ~ "70 till 80 year",
    age < 90  ~ "80 till 90 year",
    TRUE      ~ "90 year or older")) %>%
  group_by(period,ageg) %>%
  summarise(persons = sum(persons))
av2 = av1 %>%
  group_by(period) %>%
  summarise(tot = sum(persons))
grps =  c("Younger than 10 year", "10 till 20 year", "20 till 30 year",
         "30 till 40 year", "40 till 50 year", "50 till 60 year",
         "60 till 70 year", "70 till 80 year", "80 till 90 year",
         "90 year or older")
av3 = av1 %>%
  inner_join(av2,by=c(period='period')) %>%
  mutate(perc = 100*persons/tot,ageg = factor(ageg,levels=grps,ordered=T))
```

Create barchart

```
ggplot(data = av3, aes(x = period, y = perc, fill = ageg)) +
  geom_bar(stat = "identity") +
  scale_fill_discrete(guide = guide_legend(title = 'Age group')) +
  scale_x_discrete("year",breaks=seq(1986,2016,5)) +
  ggtitle('Distribution inhabitants of Amstelveen by age groups ')
```



References

- A introduction to OData : Introducing OData
- Details OData : OData - the best way to REST
- CBS (Statistics Netherlands) OData environment: Handleiding CBS Open Data Services (in Dutch)
- CBS (Statistics Netherlands) services: Open data (in Dutch)
- OData protocol v4
- OData protocol v3

Session Info

```
sessionInfo()
#> R version 3.6.0 (2019-04-26)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 18362)
#>
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=English_United States.1252
#> [2] LC_CTYPE=English_United States.1252
#> [3] LC_MONETARY=English_United States.1252
#> [4] LC_NUMERIC=C
#> [5] LC_TIME=English_United States.1252
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#> [1] stringr_1.4.0 dplyr_0.8.3   magrittr_1.5  ggplot2_3.2.1 knitr_1.25
#> [6] odataR_0.1.2
#>
#> loaded via a namespace (and not attached):
#>  [1] Rcpp_1.0.3       tidyselect_0.2.5 munsell_0.5.0    colorspace_1.4-1
#>  [5] R6_2.4.0         rlang_0.4.1      highr_0.8        tools_3.6.0
#>  [9] grid_3.6.0       gtable_0.3.0     xfun_0.10        withr_2.1.2
#> [13] htmltools_0.4.0  assertthat_0.2.1 yaml_2.2.0       lazyeval_0.2.2
#> [17] digest_0.6.22    tibble_2.1.3     crayon_1.3.4     purrr_0.3.3
#> [21] curl_4.2         glue_1.3.1       evaluate_0.14    rmarkdown_1.16
#> [25] labeling_0.3     stringi_1.4.3    compiler_3.6.0   pillar_1.4.2
#> [29] scales_1.0.0     jsonlite_1.6     pkgconfig_2.0.3
```