SPARQL scripts for RDF data cubes

mja@statgroup.dk

2016-06-27

Contents

SPARQL scripts for the demographics cube (DC-DEMO-sample.ttl)

Setup

First load the package.

```
library(knitr)
library(rrdfancillary)
devtools::load_all(pkg="../..")
```

```
## Loading rrdfqbcrndex
```

Internals

The display of SPARQL script in markdown is done by first creating a chunk, and then using the chunk with the highlight engine in knitr. The advantage of this approach is that all formatting is handled by external packages. To make the highlight output work in markdown two blanks has to be added at the end of line according to markdown syntax.

```
mdwrite<- function( sparqlStatements, refname ) {
# fn<- file.path(tempdir(), paste0( refname, ".rq" ) )
fn<- file.path(system.file("extdata/sample-rdf", package="rrdfqbcrndex"), paste0( refname, ".rq" ) )
cat( paste0("## @knitr ", refname), gsub("\\n", "  \n", sparqlStatements), sep="  \n", file=fn)
knitr::read_chunk( fn, from=c(1))
invisible(fn)
}
```

SPARQL scripts

SPARQL scripts can be used to access the RDF triple store. In the package the scripts are made by a function generating the SPARQL script. The generated SPARQL scripts are shown here for the demographics cube in DC-DEMO-sample.ttl.

The turtle file and the scrips are stored in

```
system.file("extdata/sample-xpt", package="rrdfqbcrndex")
```

```
## [1] "/home/ma/projects/rrdfqbcrnd0/rrdfqbcrndex/inst/extdata/sample-xpt"
```

Setup for generating SPARQL scripts for the demographics cube (DC-DEMO-sample.TTL)

The DEMO data exists as a turtle file in the sample-rdf directory.

```
dataCubeFile<- system.file("extdata/sample-rdf", "DC-DEMO-sample.ttl", package="rrdfqbcrndex")
store <- new.rdf()  # Initialize
cat("Loading ", dataCubeFile, "\n")
```

```
## Loading  /home/ma/projects/rrdfqbcrnd0/rrdfqbcrndex/inst/extdata/sample-rdf/DC-DEMO-sample.ttl
```

```
temp<-load.rdf(dataCubeFile, format="TURTLE", appendTo= store)
summarize.rdf(store)
```

```
## [1] "Number of triples: 3095"
```

For the functions in the package the datasets definition in the cube is needed.

```
dsdName<- GetDsdNameFromCube( store )
domainName<- GetDomainNameFromCube( store )
forsparqlprefix<- GetForSparqlPrefix( domainName )
```

SPARQL query for codelists in RDF data cube

The SPARQL query for the dimensions is made by the function GetDimensionsSparqlQuery.

```
codelistRq <- GetCodeListSparqlQuery( forsparqlprefix, dsdName )
mdwrite( codelistRq, "DEMOcodelist" )
```

```
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix prov: <http://www.w3.org/ns/prov#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dcat: <http://www.w3.org/ns/dcat#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix pav: <http://purl.org/pav\T1\textgreater{}
prefix dc: <http://purl.org/dc/elements/1.1/\T1\textgreater{}
prefix dct: <http://purl.org/dc/terms/\T1\textgreater{}
prefix mms: <http://rdf.cdisc.org/mms#>
prefix cts: <http://rdf.cdisc.org/ct/schema#>
prefix cdiscs: <http://rdf.cdisc.org/std/schema#>
prefix cdash-1-1: <http://rdf.cdisc.org/std/cdash-1-1#>
prefix cdashct: <http://rdf.cdisc.org/cdash-terminology#>
prefix sdtmct: <http://rdf.cdisc.org/sdtm-terminology#>
prefix sdtm-1-2: <http://rdf.cdisc.org/std/sdtm-1-2#>
prefix sdtm-1-3: <http://rdf.cdisc.org/std/sdtm-1-3#>
prefix sdtms-1-3: <http://rdf.cdisc.org/sdtm-1-3/schema#>
prefix sdtmig-3-1-2: <http://rdf.cdisc.org/std/sdtmig-3-1-2#>
prefix sdtmig-3-1-3: <http://rdf.cdisc.org/std/sdtmig-3-1-3#>
prefix sendct: <http://rdf.cdisc.org/send-terminology#>
prefix sendig-3-0: <http://rdf.cdisc.org/std/sendig-3-0#>
prefix adamct: <http://rdf.cdisc.org/adam-terminology#>
prefix adam-2-1: <http://rdf.cdisc.org/std/adam-2-1#>
prefix adamig-1-0: <http://rdf.cdisc.org/std/adamig-1-0#>
prefix adamvr-1-2: <http://rdf.cdisc.org/std/adamvr-1-2#>
prefix qb: <http://purl.org/linked-data/cube#>
prefix rrdfqbcrnd0: <http://www.example.org/rrdfqbcrnd0/\T1\textgreater{}
prefix code: <http://www.example.org/dc/code/\T1\textgreater{}
prefix dccs: <http://www.example.org/dc/demo/dccs/\T1\textgreater{}
prefix ds: <http://www.example.org/dc/demo/ds/\T1\textgreater{}
prefix crnd-dimension: <http://www.example.org/dc/dimension#>
prefix crnd-attribute: <http://www.example.org/dc/attribute#>
prefix crnd-measure: <http://www.example.org/dc/measure#>

select distinct ?DataStructureDefinition ?dimension ?cprefLabel ?cl ?clprefLabel ?vn ?vct ?vnop ?vnval
where {
?DataStructureDefinition a qb:DataStructureDefinition ;
qb:component ?component .
?component a qb:ComponentSpecification .
?component qb:dimension ?dimension .

?dimension qb:codeList ?c .
OPTIONAL { ?c skos:prefLabel ?cprefLabel . }
OPTIONAL { ?c rrdfqbcrnd0:DataSetRefD2RQ ?vnop . }
OPTIONAL { ?c rrdfqbcrnd0:R-columnname ?vn . }
OPTIONAL { ?c rrdfqbcrnd0:codeType ?vct . }

?c skos:hasTopConcept ?cl .
OPTIONAL { ?cl skos:prefLabel ?clprefLabel . }
OPTIONAL { ?cl rrdfqbcrnd0:R-selectionoperator ?vnop . }
OPTIONAL { ?cl rrdfqbcrnd0:R-selectionvalue ?vnval . }
values ( ?DataStructureDefinition ) {
(ds:dsd-DEMO)
}
```

}
order by ?dimension ?cl ?dimensionrefLabel

Executing the SPARQL query gives:

```
codelists<- sparql.rdf(store, codelistRq)
knitr::kable(codelists,caption="Codelists")
```

DataStructureDefinition	dimension	cprefLabel	cl
ds:dsd-DEMO	crnd-dimension:agegr1	Codelist scheme: agegr1	code:agegr1-65-80
ds:dsd-DEMO	crnd-dimension:agegr1	Codelist scheme: agegr1	code:agegr1-__65
ds:dsd-DEMO	crnd-dimension:agegr1	Codelist scheme: agegr1	code:agegr1-__80
ds:dsd-DEMO	crnd-dimension:agegr1	Codelist scheme: agegr1	code:agegr1-*ALL*
ds:dsd-DEMO	crnd-dimension:agegr1	Codelist scheme: agegr1	code:agegr1-*NONMISS*
ds:dsd-DEMO	crnd-dimension:ethnic	Codelist scheme: ethnic	code:ethnic-HISPANIC__OR__LATINO
ds:dsd-DEMO	crnd-dimension:ethnic	Codelist scheme: ethnic	code:ethnic-NOT__HISPANIC__OR__L
ds:dsd-DEMO	crnd-dimension:ethnic	Codelist scheme: ethnic	code:ethnic-*ALL*
ds:dsd-DEMO	crnd-dimension:ethnic	Codelist scheme: ethnic	code:ethnic-*NONMISS*
ds:dsd-DEMO	crnd-dimension:factor	Codelist scheme: factor	code:factor-*ALL*
ds:dsd-DEMO	crnd-dimension:factor	Codelist scheme: factor	code:factor-*NONMISS*
ds:dsd-DEMO	crnd-dimension:factor	Codelist scheme: factor	code:factor-age
ds:dsd-DEMO	crnd-dimension:factor	Codelist scheme: factor	code:factor-proportion
ds:dsd-DEMO	crnd-dimension:factor	Codelist scheme: factor	code:factor-quantity
ds:dsd-DEMO	crnd-dimension:factor	Codelist scheme: factor	code:factor-weightbl
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-count
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-max
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-mean
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-median
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-min
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-n
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-percent
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-q1
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-q3
ds:dsd-DEMO	crnd-dimension:procedure	Codelist scheme: procedure	code:procedure-std
ds:dsd-DEMO	crnd-dimension:race	Codelist scheme: race	code:race-AMERICAN__INDIAN__OR
ds:dsd-DEMO	crnd-dimension:race	Codelist scheme: race	code:race-ASIAN
ds:dsd-DEMO	crnd-dimension:race	Codelist scheme: race	code:race-BLACK__OR__AFRICAN__A
ds:dsd-DEMO	crnd-dimension:race	Codelist scheme: race	code:race-NATIVE__HAWAIIAN__OR_
ds:dsd-DEMO	crnd-dimension:race	Codelist scheme: race	code:race-WHITE
ds:dsd-DEMO	crnd-dimension:race	Codelist scheme: race	code:race-*ALL*
ds:dsd-DEMO	crnd-dimension:race	Codelist scheme: race	code:race-*NONMISS*
ds:dsd-DEMO	crnd-dimension:sex	Codelist scheme: sex	code:sex-F
ds:dsd-DEMO	crnd-dimension:sex	Codelist scheme: sex	code:sex-M
ds:dsd-DEMO	crnd-dimension:sex	Codelist scheme: sex	code:sex-U
ds:dsd-DEMO	crnd-dimension:sex	Codelist scheme: sex	code:sex-UN
ds:dsd-DEMO	crnd-dimension:sex	Codelist scheme: sex	code:sex-*ALL*
ds:dsd-DEMO	crnd-dimension:sex	Codelist scheme: sex	code:sex-*NONMISS*
ds:dsd-DEMO	crnd-dimension:trt01a	Codelist scheme: trt01a	code:trt01a-Placebo
ds:dsd-DEMO	crnd-dimension:trt01a	Codelist scheme: trt01a	code:trt01a-Xanomeline__High__Dose
ds:dsd-DEMO	crnd-dimension:trt01a	Codelist scheme: trt01a	code:trt01a-Xanomeline__Low__Dose
ds:dsd-DEMO	crnd-dimension:trt01a	Codelist scheme: trt01a	code:trt01a-*ALL*
ds:dsd-DEMO	crnd-dimension:trt01a	Codelist scheme: trt01a	code:trt01a-*NONMISS*

SPARQL query for dimensions in RDF data cube

The SPARQL query for the dimensions is made by the function GetDimensionsSparqlQuery.

```
dimensionsRq <- GetDimensionsSparqlQuery( forsparqlprefix )
mdwrite( dimensionsRq, "DEMOdimensions" )
```

prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix prov: <http://www.w3.org/ns/prov#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dcat: <http://www.w3.org/ns/dcat#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix pav: <http://purl.org/pav\T1\textgreater{}
prefix dc: <http://purl.org/dc/elements/1.1/\T1\textgreater{}
prefix dct: <http://purl.org/dc/terms/\T1\textgreater{}
prefix mms: <http://rdf.cdisc.org/mms#>
prefix cts: <http://rdf.cdisc.org/ct/schema#>
prefix cdiscs: <http://rdf.cdisc.org/std/schema#>
prefix cdash-1-1: <http://rdf.cdisc.org/std/cdash-1-1#>
prefix cdashct: <http://rdf.cdisc.org/cdash-terminology#>
prefix sdtmct: <http://rdf.cdisc.org/sdtm-terminology#>
prefix sdtm-1-2: <http://rdf.cdisc.org/std/sdtm-1-2#>
prefix sdtm-1-3: <http://rdf.cdisc.org/std/sdtm-1-3#>
prefix sdtms-1-3: <http://rdf.cdisc.org/sdtm-1-3/schema#>
prefix sdtmig-3-1-2: <http://rdf.cdisc.org/std/sdtmig-3-1-2#>
prefix sdtmig-3-1-3: <http://rdf.cdisc.org/std/sdtmig-3-1-3#>
prefix sendct: <http://rdf.cdisc.org/send-terminology#>
prefix sendig-3-0: <http://rdf.cdisc.org/std/sendig-3-0#>
prefix adamct: <http://rdf.cdisc.org/adam-terminology#>
prefix adam-2-1: <http://rdf.cdisc.org/std/adam-2-1#>
prefix adamig-1-0: <http://rdf.cdisc.org/std/adamig-1-0#>
prefix adamvr-1-2: <http://rdf.cdisc.org/std/adamvr-1-2#>
prefix qb: <http://purl.org/linked-data/cube#>
prefix rrdfqbcrnd0: <http://www.example.org/rrdfqbcrnd0/\T1\textgreater{}
prefix code: <http://www.example.org/dc/code/\T1\textgreater{}
prefix dccs: <http://www.example.org/dc/demo/dccs/\T1\textgreater{}
prefix ds: <http://www.example.org/dc/demo/ds/\T1\textgreater{}
prefix crnd-dimension: <http://www.example.org/dc/dimension#>
prefix crnd-attribute: <http://www.example.org/dc/attribute#>
prefix crnd-measure: <http://www.example.org/dc/measure#>

select * where
{ [] qb:dimension ?p . }

Executing the SPARQL query gives:

```
dimensions<- sparql.rdf(store, dimensionsRq)
knitr::kable(dimensions,col.names=c("dimension"),caption="Dimensions")
```

Table 2: Dimensions

dimension
crnd-dimension:ethnic
crnd-dimension:race
crnd-dimension:procedure
crnd-dimension:agegr1
crnd-dimension:factor
crnd-dimension:trt01a
crnd-dimension:sex

SPARQL query for attributes in RDF data cube

The SPARQL query for the attributes is made by the function GetAttributesSparqlQuery.

```
attributesRq<- GetAttributesSparqlQuery( forsparqlprefix )
mdwrite( attributesRq, "DEMOattributes" )
```

prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix prov: <http://www.w3.org/ns/prov#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dcat: <http://www.w3.org/ns/dcat#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix pav: <http://purl.org/pav\T1\textgreater{}
prefix dc: <http://purl.org/dc/elements/1.1/\T1\textgreater{}
prefix dct: <http://purl.org/dc/terms/\T1\textgreater{}
prefix mms: <http://rdf.cdisc.org/mms#>
prefix cts: <http://rdf.cdisc.org/ct/schema#>
prefix cdiscs: <http://rdf.cdisc.org/std/schema#>
prefix cdash-1-1: <http://rdf.cdisc.org/std/cdash-1-1#>
prefix cdashct: <http://rdf.cdisc.org/cdash-terminology#>
prefix sdtmct: <http://rdf.cdisc.org/sdtm-terminology#>
prefix sdtm-1-2: <http://rdf.cdisc.org/std/sdtm-1-2#>
prefix sdtm-1-3: <http://rdf.cdisc.org/std/sdtm-1-3#>
prefix sdtms-1-3: <http://rdf.cdisc.org/sdtm-1-3/schema#>
prefix sdtmig-3-1-2: <http://rdf.cdisc.org/std/sdtmig-3-1-2#>
prefix sdtmig-3-1-3: <http://rdf.cdisc.org/std/sdtmig-3-1-3#>
prefix sendct: <http://rdf.cdisc.org/send-terminology#>
prefix sendig-3-0: <http://rdf.cdisc.org/std/sendig-3-0#>
prefix adamct: <http://rdf.cdisc.org/adam-terminology#>
prefix adam-2-1: <http://rdf.cdisc.org/std/adam-2-1#>
prefix adamig-1-0: <http://rdf.cdisc.org/std/adamig-1-0#>
prefix adamvr-1-2: <http://rdf.cdisc.org/std/adamvr-1-2#>
prefix qb: <http://purl.org/linked-data/cube#>
prefix rrdfqbcrnd0: <http://www.example.org/rrdfqbcrnd0/\T1\textgreater{}
prefix code: <http://www.example.org/dc/code/\T1\textgreater{}
prefix dccs: <http://www.example.org/dc/demo/dccs/\T1\textgreater{}
prefix ds: <http://www.example.org/dc/demo/ds/\T1\textgreater{}
prefix crnd-dimension: <http://www.example.org/dc/dimension#>

prefix crnd-attribute: <http://www.example.org/dc/attribute#>
prefix crnd-measure: <http://www.example.org/dc/measure#>

select * where
{ ?p a qb:AttributeProperty . }

Executing the SPARQL query gives:

```
attributes<- sparql.rdf(store, attributesRq)
knitr::kable(attributes,col.names=c("attribute"),caption="Attributes")
```

Table 3: Attributes

attribute
crnd-attribute:cellpartno
crnd-attribute:measurefmt
crnd-attribute:colno
crnd-attribute:denominator
crnd-attribute:unit
crnd-attribute:rowno

SPARQL query for observations from RDF data cube in workbook format

The SPARQL query for the attributes is made by the function GetAttributesSparqlQuery, where the domainName, dimensions and attributes for the cube is passed as parameters.

```
observationsRq<- GetObservationsSparqlQuery( forsparqlprefix, domainName, dimensions, attributes )
mdwrite( observationsRq, "DEMOobservations" )
```

prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix prov: <http://www.w3.org/ns/prov#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dcat: <http://www.w3.org/ns/dcat#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix pav: <http://purl.org/pav\T1\textgreater{}
prefix dc: <http://purl.org/dc/elements/1.1/\T1\textgreater{}
prefix dct: <http://purl.org/dc/terms/\T1\textgreater{}
prefix mms: <http://rdf.cdisc.org/mms#>
prefix cts: <http://rdf.cdisc.org/ct/schema#>
prefix cdiscs: <http://rdf.cdisc.org/std/schema#>
prefix cdash-1-1: <http://rdf.cdisc.org/std/cdash-1-1#>
prefix cdashct: <http://rdf.cdisc.org/cdash-terminology#>
prefix sdtmct: <http://rdf.cdisc.org/sdtm-terminology#>
prefix sdtm-1-2: <http://rdf.cdisc.org/std/sdtm-1-2#>
prefix sdtm-1-3: <http://rdf.cdisc.org/std/sdtm-1-3#>
prefix sdtms-1-3: <http://rdf.cdisc.org/sdtm-1-3/schema#>
prefix sdtmig-3-1-2: <http://rdf.cdisc.org/std/sdtmig-3-1-2#>
prefix sdtmig-3-1-3: <http://rdf.cdisc.org/std/sdtmig-3-1-3#>
prefix sendct: <http://rdf.cdisc.org/send-terminology#>
prefix sendig-3-0: <http://rdf.cdisc.org/std/sendig-3-0#>

```
prefix adamct: <http://rdf.cdisc.org/adam-terminology#>
prefix adam-2-1: <http://rdf.cdisc.org/std/adam-2-1#>
prefix adamig-1-0: <http://rdf.cdisc.org/std/adamig-1-0#>
prefix adamvr-1-2: <http://rdf.cdisc.org/std/adamvr-1-2#>
prefix qb: <http://purl.org/linked-data/cube#>
prefix rrdfqbcrnd0: <http://www.example.org/rrdfqbcrnd0/\T1\textgreater{}
prefix code: <http://www.example.org/dc/code/\T1\textgreater{}
prefix dccs: <http://www.example.org/dc/demo/dccs/\T1\textgreater{}
prefix ds: <http://www.example.org/dc/demo/ds/\T1\textgreater{}
prefix crnd-dimension: <http://www.example.org/dc/dimension#>
prefix crnd-attribute: <http://www.example.org/dc/attribute#>
prefix crnd-measure: <http://www.example.org/dc/measure#>

select * where {
?s a qb:Observation ;
qb:dataSet ds:dataset-DEMO ;
crnd-dimension:ethnic ?ethnic;
crnd-dimension:race ?race;
crnd-dimension:procedure ?procedure;
crnd-dimension:agegr1 ?agegr1;
crnd-dimension:factor ?factor;
crnd-dimension:trt01a ?trt01a;
crnd-dimension:sex ?sex;
crnd-attribute:cellpartno ?cellpartno;
crnd-attribute:measurefmt ?measurefmt;
crnd-attribute:colno ?colno;
crnd-attribute:denominator ?denominator;
crnd-attribute:unit ?unit;
crnd-attribute:rowno ?rowno;
crnd-measure:measure ?measure .
optional{ ?ethnic skos:prefLabel ?ethnicvalue . }
optional{ ?race skos:prefLabel ?racevalue . }
optional{ ?procedure skos:prefLabel ?procedurevalue . }
optional{ ?agegr1 skos:prefLabel ?agegr1value . }
optional{ ?factor skos:prefLabel ?factorvalue . }
optional{ ?trt01a skos:prefLabel ?trt01avalue . }
optional{ ?sex skos:prefLabel ?sexvalue . }
}
order by ?s
```

The first 2 rows of result of the query is:

```
observations<- data.frame(sparql.rdf(store, observationsRq),stringsAsFactors=FALSE)
knitr::kable(head(observations,2),caption="Observations (only 2)")
```

s	ethnic	race	procedure	agegr1	factor	trt01a
ds:obs001	code:ethnic-*ALL*	code:race-*ALL*	code:procedure-count	code:agegr1-*ALL*	code:factor-quantity	code:trt0
ds:obs002	code:ethnic-*ALL*	code:race-*ALL*	code:procedure-count	code:agegr1-*ALL*	code:factor-quantity	code:trt0

TODO(mja): SPARQL scripts parametrised

This vignettes shows the contents of the scripts. Some of the scripts are parametrised by one or more parameters. The parameter are shown with as $p, $q etc, following the same convention as in the (http://www.w3.org/TR/vocab-data-cube/#ic-20)[RDF Data Cube Vocabulary]. This does not really work here, as some of the parameters in the R functions are intended for vectors with more than one parameter.

TODO(mja): Generating this output using the information in the documentation

TODO(mja): enter the commands here as example in each of the .Rd files. Then the output here can be generated from the .Rd files.