---
title: “Get data for RDF data cube observation”
author: "mja@statgroup.dk"
date: “2016-08-29”
output:
html_document:
toc: true
theme: united
pdf_document:
toc: true
highlight: zenburn
md_document:
variant: markdown_github

Introduction

Here I show how to get the data for for a statistics in a RDF data cube observation.

Setup

Loading libraries

```
library(rrdfancillary)
library(rrdfcdisc)
library(rrdfqb)
library(rrdfqbcrnd0)
library(knitr)
```

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. The function stores the sparql statements in a temporary file, which then is processed by `knitr::read_chunk`. The pdf generation with `rmarkdown::render` did not handle backslash-n (newline), so to avoid writing it I define the as a constant `const.newline`

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

const.newline<- "\n"
```

Here is an example of the R code to make the chunk:

```
mdwrite( dimensionsRq, "DEMOdimensions" )
```

The chunk is then shown using

```
    {r DEMOdimensions, results='asis', engine='highlight', engine.opts='-S n3 --inline-css'}
```

Setup

First, I define the location for the RDF data cube

```
rdf.data.cube.Name<- "TAB2X01"
rdf.data.cube.Dir<- "../res-ttl"
rdf.data.cube.File<- file.path(rdf.data.cube.Dir, paste("CDISC-pilot-", rdf.data.cube.Name, ".ttl", sep=
```

Second, I define the location for the `ADSL` data as turtle transformed using R2DQ (see `rrdfqbcrnd0/rrdfqbcrndex/inst/data-`

```
dataFilemap<- system.file("extdata/sample-rdf", "adsl-map.ttl", package="rrdfqbcrndex")
```

The final step is to load the turtle files into a RDF store, here named `storeCube`.

```
storeCube <- new.rdf(ontology=FALSE)
temp<- load.rdf(rdf.data.cube.File, format="TURTLE", appendTo= storeCube)
cat("Reading RDF Data Cube from file ", normalizePath(rdf.data.cube.File), const.newline)
```

```
## Reading RDF Data Cube from file  /home/ma/projects/poc-analysis-results-metadata/res-ttl/CDISC-pilot-
```

```
summarize.rdf(storeCube)
```

```
## [1] "Number of triples: 4949"
```

```
cat("Reading data set D2RQ map from file ", normalizePath(dataFilemap), const.newline)
```

```
## Reading data set D2RQ map from file  /home/ma/projects/poc-analysis-results-metadata/use-rrdfqbcrnd0/
```

```
temp<- load.rdf(dataFilemap, format="TURTLE", appendTo= storeCube)
dataFile<- system.file("extdata/sample-rdf", "adsl.ttl", package="rrdfqbcrndex")
cat("Reading data set in D2RQ format from file ", normalizePath(dataFile), const.newline)
```

```
## Reading data set in D2RQ format from file  /home/ma/projects/poc-analysis-results-metadata/use-rrdfql
```

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

```
## [1] "Number of triples: 18019"
```

Now, just to see if it works, I will get the data for one record. The key URI for a record is composed by the name of the dataset `ADSL` and the `usubjid` value `01-718-1254`.

```
d2rqbaseURL<- "http://www.example.org/datasets/"
d2rqvocab<- paste0(d2rqbaseURL, "vocab", "/", sep="")

s<- paste0("<", d2rqbaseURL,c("ADSL/01-718-1254"), ">")
data.records.rq<-paste(
    "select * ",
    "where { ?s ?p ?o.",
    " values(?s) {",
    paste("(",s,")", collapse=const.newline),
    "}",
    "}", sep=const.newline, collapse=const.newline )
mdwrite( data.records.rq, "ADSL-records-SELECT-query.rq" )
records.res<- data.frame(sparql.rdf(storeCube, data.records.rq),stringsAsFactors=FALSE)
```

The SPARQL query is

select *
where { ?s ?p ?o.
values(?s) {
(<http://www.example.org/datasets/ADSL/01-718-1254%3E)
}
}

The result is

cat(paste(apply(records.res,1,FUN=paste0, collapse=" “),collapse=const.newline),const.newline)

```
## Get the values in the cube
dsdName<- GetDsdNameFromCube( storeCube )
domainName<- GetDomainNameFromCube( storeCube )
forsparqlprefix<- GetForSparqlPrefix( domainName )
## Get cube components
componentsRq<- GetComponentSparqlQuery( forsparqlprefix, dsdName )
components<- as.data.frame(sparql.rdf(storeCube, componentsRq), stringsAsFactors=FALSE)
components$vn<- gsub("crnd-dimension:|crnd-attribute:|crnd-measure:","",components$p)
knitr::kable(components[,c("vn", "label")])
```

vn	label
agegr1	Pooled Age Group 1
bmiblgr1	Pooled Baseline BMI Group 1
durdsgr1	Pooled Disease Duration Group 1
ethnic	Ethnicity
factor	Type of procedure (quantity, proportion...)
ittfl	Intent-To-Treat Population Flag
procedure	Statistical Procedure
sex	Sex
trt01p	Planned Treatment for Period 01

```
## Get code lists
codelistsRq<- GetCodeListSparqlQuery( forsparqlprefix, dsdName )
codelists<- as.data.frame(sparql.rdf(storeCube, codelistsRq), stringsAsFactors=FALSE)
codelists$vn<- gsub("crnd-dimension:|crnd-attribute:|crnd-measure:","",codelists$dimension)
```

```
codelists$clc<- gsub("code:","",codelists$cl)
knitr::kable(codelists[,c("vn", "clc", "clprefLabel")])
```

vn	clc	clprefLabel
agegr1	agegr1-65-80	65-80
agegr1	agegr1-_65	<65
agegr1	agegr1-_80	>80
agegr1	agegr1-*ALL*	*ALL*
agegr1	agegr1-*NONMISS*	*NONMISS*
bmiblgr1	bmiblgr1-25-_30	25-<30
bmiblgr1	bmiblgr1-_25	<25
bmiblgr1	bmiblgr1-_=30	>=30
bmiblgr1	bmiblgr1-*ALL*	*ALL*
bmiblgr1	bmiblgr1-*NONMISS*	*NONMISS*
durdsgr1	durdsgr1-_12	<12
durdsgr1	durdsgr1-_=12	>=12
durdsgr1	durdsgr1-*ALL*	*ALL*
durdsgr1	durdsgr1-*NONMISS*	*NONMISS*
ethnic	ethnic-HISPANIC_OR_LATINO	HISPANIC OR LATINO
ethnic	ethnic-NOT_HISPANIC_OR_LATINO	NOT HISPANIC OR LATINO
ethnic	ethnic-*ALL*	*ALL*
ethnic	ethnic-*NONMISS*	*NONMISS*
factor	factor-*ALL*	*ALL*
factor	factor-*NONMISS*	*NONMISS*
factor	factor-age	age
factor	factor-bmibl	bmibl
factor	factor-durdis	durdis
factor	factor-educlvl	educlvl
factor	factor-heightbl	heightbl
factor	factor-mmsetot	mmsetot
factor	factor-proportion	proportion
factor	factor-quantify	quantify
factor	factor-weightbl	weightbl
ittfl	ittfl-Y	Y
ittfl	ittfl-*ALL*	*ALL*
ittfl	ittfl-*NONMISS*	*NONMISS*
procedure	procedure-count	count
procedure	procedure-max	max
procedure	procedure-mean	mean
procedure	procedure-median	median
procedure	procedure-min	min
procedure	procedure-n	n
procedure	procedure-percent	percent
procedure	procedure-stddev	stddev
sex	sex-F	F
sex	sex-M	M
sex	sex-*ALL*	*ALL*
sex	sex-*NONMISS*	*NONMISS*
trt01p	trt01p-Placebo	Placebo
trt01p	trt01p-Xanomeline_High_Dose	Xanomeline High Dose
trt01p	trt01p-Xanomeline_Low_Dose	Xanomeline Low Dose
trt01p	trt01p-*ALL*	*ALL*
trt01p	trt01p-*NONMISS*	*NONMISS*

vn	clc	clprefLabel

```
## Get dimensions
dimensionsRq <- GetDimensionsSparqlQuery( forsparqlprefix )
dimensions<- sparql.rdf(storeCube, dimensionsRq)
knitr::kable(dimensions)
```

p

crnd-dimension:procedure crnd-dimension:trt01p
crnd-dimension:agegr1
crnd-dimension:factor
crnd-dimension:sex
crnd-dimension:durdsgr1
crnd-dimension:ethnic
crnd-dimension:bmiblgr1
crnd-dimension:ittfl

```
## Get attributes
attributesRq<- GetAttributesSparqlQuery( forsparqlprefix )
attributes<- sparql.rdf(storeCube, attributesRq)
knitr::kable(attributes)
```

p

crnd-attribute:unit
crnd-attribute:denominator The result provides for each observation the data values that must match.

Create the SPARQL query

Step 1: for an observation find the values that must match

```
qobs<- "ds:obs223"


dobs.dimensions.values.rq<- paste(
    "select * where {",
"
   ?obs ?dim ?codevalue .
   ?dim a qb:DimensionProperty .
   ?codelist skos:hasTopConcept ?codevalue .
   ?codelist rrdfqbcrnd0:DataSetRefD2RQ ?vnop .
   ?codelist rrdfqbcrnd0:R-columnname ?vn .
   ?codelist rrdfqbcrnd0:codeType     ?vct .
   ?codevalue skos:prefLabel ?clprefLabel .
   ?codevalue rrdfqbcrnd0:R-selectionoperator ?Rselectionoperator .
   ?codevalue rrdfqbcrnd0:R-selectionvalue ?Rselectionvalue .
",
"values (?obs) {", paste0("(", qobs, ")",collapse=const.newline), "}",
```

```
   "}"
)
dobs.rq<-     paste( forsparqlprefix, dobs.dimensions.values.rq )
# xx
dobsobs<- NULL
dobsobs<- as.data.frame(sparql.rdf(storeCube, dobs.rq ), stringsAsFactors=FALSE)
knitr::kable(dobsobs)
```

obs	dim	codevalue	codelist	vnop	v
ds:obs223	crnd-dimension:trt01p	code:trt01p-Xanomeline_High_Dose	code:trt01p	rrdfqbcrnd0:ADSL_TRT01P	tr
ds:obs223	crnd-dimension:ittfl	code:ittfl-Y	code:ittfl	rrdfqbcrnd0:ADSL_ITTFL	it

For subsequent use, the query is changed to only select the variables need for getting the data.

```
dobs1.dimensions.values.rq<- paste(
    "select ?vnop ?Rselectionvalue",
    " (replace(str(?vnop),'http://www.example.org/rrdfqbcrnd0/([A-Z0-9_]+)$', '$1', 'i' ) as ?d2rqname)"
    paste0(" ( concat(", "'", d2rqvocab, "'", ", ", "replace(str(?vnop),'http://www.example.org/rrdfqbc
    "where {",
"
   ?obs ?dim ?codevalue .
   ?dim a qb:DimensionProperty .
   ?codelist skos:hasTopConcept ?codevalue .
   ?codelist rrdfqbcrnd0:DataSetRefD2RQ ?vnop .
   ?codelist rrdfqbcrnd0:R-columnname ?vn .
   ?codelist rrdfqbcrnd0:codeType     ?vct .
   ?codevalue skos:prefLabel ?clprefLabel .
   ?codevalue rrdfqbcrnd0:R-selectionoperator ?Rselectionoperator .
   ?codevalue rrdfqbcrnd0:R-selectionvalue ?Rselectionvalue .
",
"values (?obs) {", paste0("(", qobs, ")",collapse=const.newline), "}",
   "}"
)
dobs1.rq<-     paste( forsparqlprefix, dobs1.dimensions.values.rq )
## xx
dobs1obs<- NULL
dobs1obs<- as.data.frame(sparql.rdf(storeCube, dobs1.rq ), stringsAsFactors=FALSE)
knitr::kable(dobs1obs)
```

vnop	Rselectionvalue	d2rqname	d2rqIRI
rrdfqbcrnd0:ADSL_TRT01P	Xanomeline High Dose	ADSL_TRT01P	http://www.example.org/datasets/vocab/ADSL
rrdfqbcrnd0:ADSL_ITTFL	Y	ADSL_ITTFL	http://www.example.org/datasets/vocab/ADSL

For subsequent use, the query is changed to only select the variables need for getting the data.

```
dobs1.dimensions.values.rq<- paste(
    "select ",
    paste0(" ( iri(concat(", "'", d2rqvocab, "'", ", ", "replace(str(?vnop),'http://www.example.org/rrd
    "?matchvalue\n",
```

```
    "where {",
"
   ?obs ?dim ?codevalue .
   ?dim a qb:DimensionProperty .
   ?codelist skos:hasTopConcept ?codevalue .
   ?codelist rrdfqbcrnd0:DataSetRefD2RQ ?vnop .
   ?codelist rrdfqbcrnd0:R-columnname ?vn .
   ?codelist rrdfqbcrnd0:codeType     ?vct .
   ?codevalue skos:prefLabel ?clprefLabel .
   ?codevalue rrdfqbcrnd0:R-selectionoperator ?Rselectionoperator .
   ?codevalue rrdfqbcrnd0:R-selectionvalue ?matchvalue.
",
"values (?obs) {", paste0("(", qobs, ")",collapse=const.newline), "}",
   "}"
)
dobs1.rq<-     paste( forsparqlprefix, dobs1.dimensions.values.rq )
## xx
dobs1obs<- NULL
dobs1obs<- as.data.frame(sparql.rdf(storeCube, dobs1.rq ), stringsAsFactors=FALSE)
knitr::kable(dobs1obs)
```

variable	matchvalue
http://www.example.org/datasets/vocab/ADSL_TRT01P	Xanomeline High Dose
http://www.example.org/datasets/vocab/ADSL_ITTFL	Y

Step 2

Next step is to retrieve the data, using the values hardcoded. The next query shows the matching records in the usual rows and columns format, with the columns represeting the unique key for the record, defined by D2RQ, and the two columns in question `TRT01P` and `ITTFL`.

```
records.rq<- paste("select * ",
           "where {
?s d2rqvocab:ADSL_TRT01P ?TRT01P .
?s d2rqvocab:ADSL_ITTFL ?ITTFL .
?s d2rqvocab:ADSL_TRT01P 'Xanomeline High Dose' ;
   d2rqvocab:ADSL_ITTFL  'Y' .
",
"}",
"order by ?s",
sep=const.newline, collapse=const.newline )
dobsds.rq<-     paste( forsparqlprefix,
                       paste0( "prefix d2rqvocab: ", "<", d2rqvocab, ">", collapse=""),
                       records.rq )
## xx
dobsobs<- NULL
dobsobs<- as.data.frame(sparql.rdf(storeCube, dobsds.rq ), stringsAsFactors=FALSE)
knitr::kable(dobsobs)
```

s	TRT01P	ITTFL
http://www.example.org/datasets/ADSL/01-701-1028	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1034	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1133	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1146	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1148	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1180	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1181	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1239	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1275	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1287	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1302	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1360	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1383	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1444	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1076	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1258	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1295	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1335	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1403	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1439	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1008	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1017	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1065	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1074	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1093	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1241	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1266	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1332	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1280	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1281	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1303	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1310	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1377	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1382	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-706-1049	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1178	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1213	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1216	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1236	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1336	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1347	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1372	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1406	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1029	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1099	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1168	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1238	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1309	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1329	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1424	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1006	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1021	Xanomeline High Dose	Y

s	TRT01P	ITTFL
http://www.example.org/datasets/ADSL/01-710-1070	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1137	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1142	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1187	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1249	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1278	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1354	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1408	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-711-1012	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-711-1433	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-713-1106	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-713-1141	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-713-1209	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-714-1288	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-714-1425	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-715-1319	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-715-1321	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1030	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1071	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1189	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1229	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1364	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1373	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1418	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1447	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-717-1109	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-717-1174	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-717-1357	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-718-1101	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-718-1328	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-718-1371	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-718-1427	Xanomeline High Dose	Y

Another representation is showing the the variables one in each row.

```
obs.triples1.rq<-paste("select * ",
           "where {
?s ?variable ?value .
values (?variable) {
(d2rqvocab:ADSL_TRT01P)
(d2rqvocab:ADSL_ITTFL)
}
",
"}",
"order by ?s",
sep=const.newline, collapse=const.newline )
dobs.triples1.ds.rq<-     paste( forsparqlprefix,
                      paste0( "prefix d2rqvocab: ", "<", d2rqvocab, ">", collapse=""),
                      obs.triples1.rq )
## xx
dobs.triples1.obs<- NULL
```

```
dobs.triples1.obs<- as.data.frame(sparql.rdf(storeCube, dobs.triples1.ds.rq ), stringsAsFactors=FALSE)
knitr::kable(head(dobs.triples1.obs,10))
```

s	variable	value
http://www.example.org/datasets/ADSL/01-701-1015	d2rqvocab:ADSL_TRT01P	Placebo
http://www.example.org/datasets/ADSL/01-701-1015	d2rqvocab:ADSL_ITTFL	Y
http://www.example.org/datasets/ADSL/01-701-1023	d2rqvocab:ADSL_TRT01P	Placebo
http://www.example.org/datasets/ADSL/01-701-1023	d2rqvocab:ADSL_ITTFL	Y
http://www.example.org/datasets/ADSL/01-701-1028	d2rqvocab:ADSL_TRT01P	Xanomeline High Dose
http://www.example.org/datasets/ADSL/01-701-1028	d2rqvocab:ADSL_ITTFL	Y
http://www.example.org/datasets/ADSL/01-701-1033	d2rqvocab:ADSL_TRT01P	Xanomeline Low Dose
http://www.example.org/datasets/ADSL/01-701-1033	d2rqvocab:ADSL_ITTFL	Y
http://www.example.org/datasets/ADSL/01-701-1034	d2rqvocab:ADSL_TRT01P	Xanomeline High Dose
http://www.example.org/datasets/ADSL/01-701-1034	d2rqvocab:ADSL_ITTFL	Y

Next step is adding the matching value and reporting if its equal with 0 and 1. Note the BIND is after the values statement.

```
obs.triples2.rq<-paste("select * ",
           "where {
?s ?variable ?value .
values (?variable ?matchvalue) {
(d2rqvocab:ADSL_TRT01P 'Xanomeline High Dose')
(d2rqvocab:ADSL_ITTFL 'Y' )
}
BIND(IF(?value=?matchvalue,1,0) AS ?isequal)
",
"}",
"order by ?s",
sep=const.newline, collapse=const.newline )
dobs.triples2.ds.rq<-     paste( forsparqlprefix,
                      paste0( "prefix d2rqvocab: ", "<", d2rqvocab, ">", collapse=""),
                      obs.triples2.rq )
## xx
dobs.triples2.obs<- NULL
dobs.triples2.obs<- as.data.frame(sparql.rdf(storeCube, dobs.triples2.ds.rq ), stringsAsFactors=FALSE)
knitr::kable(head(dobs.triples2.obs,10))
```

s	variable	value	matchvalue
http://www.example.org/datasets/ADSL/01-701-1015	d2rqvocab:ADSL_TRT01P	Placebo	Xanomeline
http://www.example.org/datasets/ADSL/01-701-1015	d2rqvocab:ADSL_ITTFL	Y	Y
http://www.example.org/datasets/ADSL/01-701-1023	d2rqvocab:ADSL_TRT01P	Placebo	Xanomeline
http://www.example.org/datasets/ADSL/01-701-1023	d2rqvocab:ADSL_ITTFL	Y	Y
http://www.example.org/datasets/ADSL/01-701-1028	d2rqvocab:ADSL_TRT01P	Xanomeline High Dose	Xanomeline
http://www.example.org/datasets/ADSL/01-701-1028	d2rqvocab:ADSL_ITTFL	Y	Y
http://www.example.org/datasets/ADSL/01-701-1033	d2rqvocab:ADSL_TRT01P	Xanomeline Low Dose	Xanomeline
http://www.example.org/datasets/ADSL/01-701-1033	d2rqvocab:ADSL_ITTFL	Y	Y
http://www.example.org/datasets/ADSL/01-701-1034	d2rqvocab:ADSL_TRT01P	Xanomeline High Dose	Xanomeline
http://www.example.org/datasets/ADSL/01-701-1034	d2rqvocab:ADSL_ITTFL	Y	Y

Using SPARQL aggregate query grouping by `?s` provides the number of not-matching values. Now, the desired records `?s` are those where there is 0 not-equal variables.

```
obs.triples3.rq<-paste("SELECT ?s (SUM(?notequal) as ?nnotqual) ",
           "where {
?s ?variable ?value .
values (?variable ?matchvalue) {
(d2rqvocab:ADSL_TRT01P 'Xanomeline High Dose')
(d2rqvocab:ADSL_ITTFL 'Y' )
}
BIND(IF(?value!=?matchvalue,1,0) AS ?notequal)
",
"}",
"group by ?s",
"order by ?s",
sep=const.newline, collapse=const.newline )
dobs.triples3.ds.rq<-     paste( forsparqlprefix,
                      paste0( "prefix d2rqvocab: ", "<", d2rqvocab, ">", collapse=""),
                      obs.triples3.rq )
## xx
dobs.triples3.obs<- NULL
dobs.triples3.obs<- as.data.frame(sparql.rdf(storeCube, dobs.triples3.ds.rq ), stringsAsFactors=FALSE)
knitr::kable(head(dobs.triples3.obs))
```

s	nnotqual
http://www.example.org/datasets/ADSL/01-701-1015	1
http://www.example.org/datasets/ADSL/01-701-1023	1
http://www.example.org/datasets/ADSL/01-701-1028	0
http://www.example.org/datasets/ADSL/01-701-1033	1
http://www.example.org/datasets/ADSL/01-701-1034	0
http://www.example.org/datasets/ADSL/01-701-1047	1

Then to get the desired records, identified by `?s`, the SPARQL HAVING term is used.

```
obs.triples4.rq<-paste("SELECT ?s ",
           "where {
?s ?variable ?value .
values (?variable ?matchvalue) {
(d2rqvocab:ADSL_TRT01P 'Xanomeline High Dose')
(d2rqvocab:ADSL_ITTFL 'Y' )
}
BIND(IF(?value!=?matchvalue,1,0) AS ?notequal)
",
"}",
"group by ?s",
"having(SUM(?notequal)=0)",
"order by ?s",
sep=const.newline, collapse=const.newline )
dobs.triples4.ds.rq<-     paste( forsparqlprefix,
                      paste0( "prefix d2rqvocab: ", "<", d2rqvocab, ">", collapse=""),
                      obs.triples4.rq )
## xx
```

```
dobs.triples4.obs<- NULL
dobs.triples4.obs<- as.data.frame(sparql.rdf(storeCube, dobs.triples4.ds.rq ), stringsAsFactors=FALSE)
knitr::kable(head(dobs.triples4.obs))
```

s

http://www.example.org/datasets/ADSL/01-701-1028 http://www.example.org/datasets/ADSL/01-701-1034 http://www.example.org/datasets/ADSL/01-701-1133 http://www.example.org/datasets/ADSL/01-701-1146 http://www.example.org/datasets/ADSL/01-701-1148 http://www.example.org/datasets/ADSL/01-701-1180

The query above can then be used as a subquery in the full querqy.

```
records2.rq<-paste("select * ",
           "where {
?s d2rqvocab:ADSL_TRT01P ?TRT01P .
?s d2rqvocab:ADSL_ITTFL ?ITTFL .
",
"{",
obs.triples4.rq,
"}",
"}",
"order by ?s",
sep=const.newline, collapse=const.newline )

dobs2ds.rq<-     paste( forsparqlprefix,
                     paste0( "prefix d2rqvocab: ", "<", d2rqvocab, ">", collapse=""),
                     records2.rq )
## xx
dobs2obs<- NULL
dobs2obs<- as.data.frame(sparql.rdf(storeCube, dobs2ds.rq ), stringsAsFactors=FALSE)
knitr::kable(dobs2obs)
```

s	TRT01P	ITTFL
http://www.example.org/datasets/ADSL/01-701-1028	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1034	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1133	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1146	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1148	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1180	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1181	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1239	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1275	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1287	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1302	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1360	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1383	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-701-1444	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1076	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1258	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1295	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1335	Xanomeline High Dose	Y

s	TRT01P	ITTFL
http://www.example.org/datasets/ADSL/01-703-1403	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-703-1439	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1008	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1017	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1065	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1074	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1093	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1241	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1266	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-704-1332	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1280	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1281	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1303	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1310	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1377	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-705-1382	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-706-1049	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1178	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1213	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1216	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1236	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1336	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1347	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1372	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-708-1406	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1029	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1099	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1168	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1238	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1309	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1329	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-709-1424	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1006	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1021	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1070	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1137	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1142	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1187	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1249	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1278	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1354	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-710-1408	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-711-1012	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-711-1433	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-713-1106	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-713-1141	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-713-1209	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-714-1288	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-714-1425	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-715-1319	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-715-1321	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1030	Xanomeline High Dose	Y

s	TRT01P	ITTFL
http://www.example.org/datasets/ADSL/01-716-1071	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1189	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1229	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1364	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1373	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1418	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-716-1447	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-717-1109	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-717-1174	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-717-1357	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-718-1101	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-718-1328	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-718-1371	Xanomeline High Dose	Y
http://www.example.org/datasets/ADSL/01-718-1427	Xanomeline High Dose	Y

The values part of the query above can be replaced by a subquery.

```
obs.triples5.rq<-paste(
paste(
    "select ",
    paste0(" ( iri(concat(", "'", d2rqvocab, "'", ", ", "replace(str(?vnop),'http://www.example.org/rrdi
    "?matchvalue\n",
    "where {",
"
   ?obs ?dim ?codevalue .
   ?dim a qb:DimensionProperty .
   ?codelist skos:hasTopConcept ?codevalue .
   ?codelist rrdfqbcrnd0:DataSetRefD2RQ ?vnop .
   ?codelist rrdfqbcrnd0:R-columnname ?vn .
   ?codelist rrdfqbcrnd0:codeType     ?vct .
   ?codevalue skos:prefLabel ?clprefLabel .
   ?codevalue rrdfqbcrnd0:R-selectionoperator ?Rselectionoperator .
   ?codevalue rrdfqbcrnd0:R-selectionvalue ?matchvalue.
",
"values (?obs) {", paste0("(", qobs, ")",collapse=const.newline), "}",
"}"
),
sep=const.newline, collapse=const.newline )
dobs.triples5.ds.rq<-     paste( forsparqlprefix,
                      paste0( "prefix d2rqvocab: ", "<", d2rqvocab, ">", const.newline, collapse=""),
                      obs.triples5.rq )
## xx
dobs.triples5.obs<- NULL
dobs.triples5.obs<- as.data.frame(sparql.rdf(storeCube, dobs.triples5.ds.rq ), stringsAsFactors=FALSE)
knitr::kable(head(dobs.triples5.obs))
```

variable	matchvalue
d2rqvocab:ADSL_TRT01P	Xanomeline High Dose
d2rqvocab:ADSL_ITTFL	Y

```
obs.triples6.rq<-paste(
    "SELECT ?s",
          "where {
?s ?variable ?value
",
"{",
paste(
    "select ",
    paste0(" ( iri(concat(", "'", d2rqvocab, "'", ", ", "replace(str(?vnop),'http://www.example.org/rrdf
    "?matchvalue\n",
    "where {",
"
   ?obs ?dim ?codevalue .
   ?dim a qb:DimensionProperty .
   ?codelist skos:hasTopConcept ?codevalue .
   ?codelist rrdfqbcrnd0:DataSetRefD2RQ ?vnop .
   ?codelist rrdfqbcrnd0:R-columnname ?vn .
   ?codelist rrdfqbcrnd0:codeType     ?vct .
   ?codevalue skos:prefLabel ?clprefLabel .
   ?codevalue rrdfqbcrnd0:R-selectionoperator ?Rselectionoperator .
   ?codevalue rrdfqbcrnd0:R-selectionvalue ?matchvalue.
",
"values (?obs) {", paste0("(", qobs, ")",collapse=const.newline), "}",
"}"
),
"}",
"BIND(IF(?value!=?matchvalue,1,0) AS ?notequal)",
"}",
"group by ?s",
"having(SUM(?notequal)=0)",
"order by ?s",
sep=const.newline, collapse=const.newline )
dobs.triples6.ds.rq<-     paste( forsparqlprefix,
                      paste0( "prefix d2rqvocab: ", "<", d2rqvocab, ">", const.newline, collapse=""),
                      obs.triples6.rq )
## xx
dobs.triples6.obs<- NULL
dobs.triples6.obs<- as.data.frame(sparql.rdf(storeCube, dobs.triples6.ds.rq ), stringsAsFactors=FALSE)
knitr::kable(dobs.triples6.obs)
```

s

http://www.example.org/datasets/ADSL/01-701-1028 http://www.example.org/datasets/ADSL/01-701-1034
http://www.example.org/datasets/ADSL/01-701-1133 http://www.example.org/datasets/ADSL/01-701-1146
http://www.example.org/datasets/ADSL/01-701-1148 http://www.example.org/datasets/ADSL/01-701-1180
http://www.example.org/datasets/ADSL/01-701-1181 http://www.example.org/datasets/ADSL/01-701-1239
http://www.example.org/datasets/ADSL/01-701-1275 http://www.example.org/datasets/ADSL/01-701-1287
http://www.example.org/datasets/ADSL/01-701-1302 http://www.example.org/datasets/ADSL/01-701-1360
http://www.example.org/datasets/ADSL/01-701-1383 http://www.example.org/datasets/ADSL/01-701-1444
http://www.example.org/datasets/ADSL/01-703-1076 http://www.example.org/datasets/ADSL/01-703-1258
http://www.example.org/datasets/ADSL/01-703-1295 http://www.example.org/datasets/ADSL/01-703-1335
http://www.example.org/datasets/ADSL/01-703-1403 http://www.example.org/datasets/ADSL/01-703-1439

http://www.example.org/datasets/ADSL/01-704-1008 http://www.example.org/datasets/ADSL/01-704-1017
http://www.example.org/datasets/ADSL/01-704-1065 http://www.example.org/datasets/ADSL/01-704-1074
http://www.example.org/datasets/ADSL/01-704-1093 http://www.example.org/datasets/ADSL/01-704-1241
http://www.example.org/datasets/ADSL/01-704-1266 http://www.example.org/datasets/ADSL/01-704-1332
http://www.example.org/datasets/ADSL/01-705-1280 http://www.example.org/datasets/ADSL/01-705-1281
http://www.example.org/datasets/ADSL/01-705-1303 http://www.example.org/datasets/ADSL/01-705-1310
http://www.example.org/datasets/ADSL/01-705-1377 http://www.example.org/datasets/ADSL/01-705-1382
http://www.example.org/datasets/ADSL/01-706-1049 http://www.example.org/datasets/ADSL/01-708-1178
http://www.example.org/datasets/ADSL/01-708-1213 http://www.example.org/datasets/ADSL/01-708-1216
http://www.example.org/datasets/ADSL/01-708-1236 http://www.example.org/datasets/ADSL/01-708-1336
http://www.example.org/datasets/ADSL/01-708-1347 http://www.example.org/datasets/ADSL/01-708-1372
http://www.example.org/datasets/ADSL/01-708-1406 http://www.example.org/datasets/ADSL/01-709-1029
http://www.example.org/datasets/ADSL/01-709-1099 http://www.example.org/datasets/ADSL/01-709-1168
http://www.example.org/datasets/ADSL/01-709-1238 http://www.example.org/datasets/ADSL/01-709-1309
http://www.example.org/datasets/ADSL/01-709-1329 http://www.example.org/datasets/ADSL/01-709-1424
http://www.example.org/datasets/ADSL/01-710-1006 http://www.example.org/datasets/ADSL/01-710-1021
http://www.example.org/datasets/ADSL/01-710-1070 http://www.example.org/datasets/ADSL/01-710-1137
http://www.example.org/datasets/ADSL/01-710-1142 http://www.example.org/datasets/ADSL/01-710-1187
http://www.example.org/datasets/ADSL/01-710-1249 http://www.example.org/datasets/ADSL/01-710-1278
http://www.example.org/datasets/ADSL/01-710-1354 http://www.example.org/datasets/ADSL/01-710-1408
http://www.example.org/datasets/ADSL/01-711-1012 http://www.example.org/datasets/ADSL/01-711-1433
http://www.example.org/datasets/ADSL/01-713-1106 http://www.example.org/datasets/ADSL/01-713-1141
http://www.example.org/datasets/ADSL/01-713-1209 http://www.example.org/datasets/ADSL/01-714-1288
http://www.example.org/datasets/ADSL/01-714-1425 http://www.example.org/datasets/ADSL/01-715-1319
http://www.example.org/datasets/ADSL/01-715-1321 http://www.example.org/datasets/ADSL/01-716-1030
http://www.example.org/datasets/ADSL/01-716-1071 http://www.example.org/datasets/ADSL/01-716-1189
http://www.example.org/datasets/ADSL/01-716-1229 http://www.example.org/datasets/ADSL/01-716-1364
http://www.example.org/datasets/ADSL/01-716-1373 http://www.example.org/datasets/ADSL/01-716-1418
http://www.example.org/datasets/ADSL/01-716-1447 http://www.example.org/datasets/ADSL/01-717-1109
http://www.example.org/datasets/ADSL/01-717-1174 http://www.example.org/datasets/ADSL/01-717-1357
http://www.example.org/datasets/ADSL/01-718-1101 http://www.example.org/datasets/ADSL/01-718-1328
http://www.example.org/datasets/ADSL/01-718-1371 http://www.example.org/datasets/ADSL/01-718-1427

Final stuff

Here is how to make the PDF file

```
rmarkdown::render('get-data-for-cube-observation.Rmd', c("html_document", "pdf_document"), clean=TRUE)
```