Two dimensional representation of RDF data cube

PhuseSubTeamAnalysisResults@example.org

2016-02-18

Contents

Setup

First load the package.

```
library(rrdf)
library(rrdfqb)
library(rrdfqbcrnd0)
library(rrdfqbcrndex)
devtools::load_all(pkg="../..")
```

```
## Loading rrdfqbpresent
```

Current - 08-may-2015

The following code is under development. It creates HTML files under extdata/sample-cfg showing a two dimensional representation of the RDF data cube.

Here are some the features that are evaluated - drag and drop of measure - could be used for creating a new table from existing - store RDF as RDFa - could be used to embed the whole cube in the file; looks like it gets to big

Pending: including the rest of the cube into the html. Consider if in each cell only the observation and measure should be referenced and not all properties and objects for the observation.

Using with RDFa

The function `MakeHTMLfromQb` is used to create the HTML file. When invoked with `useRDFa=TRUE` the generated HTML will contain RDFa markup.

The HTML includes green-turtle (https://github.com/alexmilowski/green-turtle), jquery (http://jquery.com/) and jqueryUI (http://jqueryui.com/).

In my setup I store the project under packages, and can symlink to the files from the extdata/sample-cfg directory.

```
cd extdata/sample-cfg
ln -s ~/packages/green-turtle/build/RDFa.min.1.4.0.js
ln -s ~/packages/green-turtle/build/RDFaProcessor.min.1.4.0.js
ln -s ~/packages/jquery-2.1.3.min/jquery-2.1.3.min.js .
ln -s ~/packages/jquery-ui-1.11.3.custom .
```

```
MakeTable<- function( dataCubeFile, htmlfile, rowdim, coldim, idrow, idcol ) {
    store <- new.rdf()  # Initialize
    cat("Loading ", dataCubeFile, "\n")
    temp<-load.rdf(dataCubeFile, format="TURTLE", appendTo= store)
    summarize.rdf(store)
    dsdName<- GetDsdNameFromCube( store )
    domainName<- GetDomainNameFromCube( store )
    forsparqlprefix<- GetForSparqlPrefix( domainName )

    dimensions<- sparql.rdf(store, GetDimensionsSparqlQuery( forsparqlprefix ) )
    attributesDf<- sparql.rdf(store, GetAttributesSparqlQuery( forsparqlprefix ))

    outhtmlfile<- MakeHTMLfromQb( store, forsparqlprefix, dsdName, domainName,
                                  dimensions, rowdim, coldim, idrow, idcol,
                                  htmlfile, useRDFa=TRUE, compactDimColumns=FALSE,
                                  debug=FALSE)

    outhtmlfile
}
```

Create AE html example

```
dataCubeFile<- system.file("extdata/sample-rdf", "DC-AE-sample.ttl", package="rrdfqbcrndex")
# ToDo(mja): write to a temporary file  or move this to extdata
htmlfile<- file.path(system.file("extdata/sample-html", package="rrdfqbpresent"), "DC-AE-sample.html")
rowdim<- c("crnd-attribute:rowno", "crnd-dimension:aesoc", "crnd-dimension:aedecod" )
coldim<- c("crnd-attribute:colno", "crnd-attribute:cellpartno", "crnd-dimension:trta",
           "crnd-dimension:factor", "crnd-dimension:procedure" )
# idrow is a function of rowdim; writing it directly is easier for now
idrow<-  c( "aesocvalue", "aedecodvalue" )
idcol<-  c( "crnd-dimension:trta" )
resHtmlFile<- MakeTable( dataCubeFile, htmlfile, rowdim, coldim, idrow, idcol )
```

```
## Loading  /home/ma/R/x86_64-redhat-linux-gnu-library/3.2/rrdfqbcrndex/extdata/sample-rdf/DC-AE-sample
## [1] "Number of triples: 25289"
```

```
cat("HTML stored as: ", normalizePath(resHtmlFile), "\n")
```

```
## HTML stored as:  /home/ma/projects/rrdfqbcrnd0/rrdfqbpresent/inst/extdata/sample-html/DC-AE-sample.ht
```

Create DEMO html example

```
dataCubeFile<- system.file("extdata/sample-rdf", "DC-DEMO-sample.ttl", package="rrdfqbcrndex")
# ToDo(mja): write to a temporary file  or move this to extdata
htmlfile<- file.path(system.file("extdata/sample-html", package="rrdfqbpresent"), "DC-DEMO-sample.html")
coldim<- c("crnd-attribute:colno", "crnd-attribute:cellpartno", "crnd-dimension:trt01a",
           "crnd-dimension:factor", "crnd-dimension:procedure" )
rowdim<- c("crnd-attribute:rowno", "crnd-dimension:agegr1", "crnd-dimension:race",
           "crnd-dimension:ethnic", "crnd-dimension:sex" )
idrow<-  c( "agegr1value", "racevalue", "ethnicvalue", "sexvalue" )
idcol<-  c( "crnd-dimension:trt01a" )
resHtmlFile<- MakeTable( dataCubeFile, htmlfile, rowdim, coldim, idrow, idcol )
```

```
## Loading  /home/ma/R/x86_64-redhat-linux-gnu-library/3.2/rrdfqbcrndex/extdata/sample-rdf/DC-DEMO-sampl
## [1] "Number of triples: 3081"
```

```
cat("HTML stored as: ", normalizePath(resHtmlFile), "\n")
```

```
## HTML stored as:  /home/ma/projects/rrdfqbcrnd0/rrdfqbpresent/inst/extdata/sample-html/DC-DEMO-sample
```

For development

Setup

```
store <- new.rdf()  # Initialize
cat("Loading ", dataCubeFile, "\n")
```

```
## Loading  /home/ma/R/x86_64-redhat-linux-gnu-library/3.2/rrdfqbcrndex/extdata/sample-rdf/DC-DEMO-sampl
```

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

```
## [1] "Number of triples: 3081"
```

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

Check GetTwoDimTableFromQb

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

```
## Loading  /home/ma/R/x86_64-redhat-linux-gnu-library/3.2/rrdfqbcrndex/extdata/sample-rdf/DC-DEMO-sampl
```

```
temp<-load.rdf(dataCubeFile, format="TURTLE", appendTo= store)
dsdName<- GetDsdNameFromCube( store )
domainName<- GetDomainNameFromCube( store )
forsparqlprefix<- GetForSparqlPrefix( domainName )
cat("dsdName: ", dsdName, "\n")
```

```
## dsdName:  dsd-DEMO
```

```
cat("domainName: ", domainName, "\n")
```

```
## domainName:  DEMO
```

```
cat("forsparqlprefix: ", forsparqlprefix, "\n")
```

```
## forsparqlprefix:  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 qb: <http://purl.org/linked-data/cube#>
## prefix pav: <http://purl.org/pav>
## prefix dct: <http://purl.org/dc/terms/>
## prefix mms: <http://rdf.cdisc.org/mms#>
## prefix cts: <http://rdf.cdisc.org/ct/schema#>
## prefix qb: <http://purl.org/linked-data/cube#>
## prefix rrdfqbcrnd0: <http://www.example.org/rrdfqbcrnd0/>
## prefix code: <http://www.example.org/dc/code/>
## prefix dccs: <http://www.example.org/dc/demo/dccs/>
## prefix ds: <http://www.example.org/dc/demo/ds/>
## 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#>
##
```

```
coldim<- c("crnd-attribute:colno", "crnd-attribute:cellpartno", "crnd-dimension:trt01a" )
rowdim<- c("crnd-attribute:rowno", "crnd-dimension:agegr1", "crnd-dimension:race",
           "crnd-dimension:ethnic", "crnd-dimension:sex", "crnd-dimension:procedure" )
xqbtest<- GetTwoDimTableFromQb( store, forsparqlprefix, domainName, rowdim, coldim )

dimensionsRq <- GetDimensionsSparqlQuery( forsparqlprefix )
dimensions<- sparql.rdf(store, dimensionsRq)
```

```
attributesRq<- GetAttributesSparqlQuery( forsparqlprefix )
attributes<- sparql.rdf(store, attributesRq)

observationsDescriptionRq<- GetObservationsWithDescriptionSparqlQuery( forsparqlprefix, domainName, dim
observationsDesc<- as.data.frame(sparql.rdf(store, observationsDescriptionRq ), stringsAsFactors=FALSE)
```

View a specific observation.

The observation is specified in the values part of the SPARQL query.

```
cube.observations.rq<-  paste( forsparqlprefix,
'
select *
where {
?s a qb:Observation ;
?p ?o .
values (?s) {
(ds:obs029)
}
}
',
"\n"
)

cube.observations<- sparql.rdf(store, cube.observations.rq)
knitr::kable(cube.observations)
```

s	p	o
ds:obs029	crnd-dimension:factor	code:factor-age
ds:obs029	crnd-attribute:cellpartno	1
ds:obs029	crnd-measure:measure	75.666666667
ds:obs029	crnd-dimension:race	code:race-*ALL*
ds:obs029	crnd-dimension:sex	code:sex-*ALL*
ds:obs029	crnd-dimension:procedure	code:procedure-mean
ds:obs029	rdf:type	qb:Observation
ds:obs029	rdfs:comment	Statistic for number of records/Statistics for factor with the dimensions XX
ds:obs029	crnd-dimension:trt01a	code:trt01a-Xanomeline_Low_Dose
ds:obs029	crnd-attribute:measurefmt	%6.1f
ds:obs029	crnd-attribute:rowno	7
ds:obs029	crnd-attribute:unit	NA
ds:obs029	crnd-dimension:agegr1	code:agegr1-*ALL*
ds:obs029	qb:dataSet	ds:dataset-DEMO
ds:obs029	crnd-dimension:ethnic	code:ethnic-*ALL*
ds:obs029	crnd-attribute:denominator	
ds:obs029	rdfs:label	29
ds:obs029	crnd-attribute:colno	2

View formating of measure

```
cube.measurefmt.rq<-  paste( forsparqlprefix,
'
select distinct ?procedure ?measurefmt
where {
?s a qb:Observation ;
crnd-dimension:procedure ?procedure ;
crnd-attribute:measurefmt ?measurefmt .
}
',
"\n"
)

cube.measurefmt<- sparql.rdf(store, cube.measurefmt.rq)
knitr::kable(cube.measurefmt)
```

procedure	measurefmt
code:procedure-q3	%6.1f
code:procedure-count	%6.0f
code:procedure-percent	%6.1f
code:procedure-std	%6.1f
code:procedure-min	%6.1f
code:procedure-n	%6.0f
code:procedure-median	%6.1f
code:procedure-q1	%6.1f
code:procedure-max	%6.1f
code:procedure-mean	%6.1f

Columns

The contents of the

```
cols.rq<- GetRownoColnoCellpartnoSparqlQuery( forsparqlprefix )
cols<- data.frame(sparql.rdf(store, cols.rq))
knitr::kable(cols)
```

rowno	colno	cellpartno
1	1	1
1	1	2
1	2	1
1	2	2
1	3	1
1	3	2
10	1	1
10	2	1
10	3	1
11	1	1
11	2	1

rowno	colno	cellpartno
11	3	1
18	1	1
18	1	2
18	2	1
18	2	2
18	3	1
18	3	2
19	1	1
19	1	2
19	2	1
19	2	2
19	3	1
19	3	2
2	1	1
2	1	2
2	2	1
2	2	2
2	3	1
2	3	2
20	1	1
20	1	2
20	2	1
20	2	2
20	3	1
20	3	2
21	1	1
21	1	2
21	2	1
21	2	2
21	3	1
21	3	2
22	1	1
22	1	2
22	2	1
22	2	2
22	3	1
22	3	2
23	1	1
23	1	2
23	2	1
23	2	2
23	3	1
23	3	2
24	1	1
24	1	2
24	2	1
24	2	2
24	3	1
24	3	2
25	1	1
25	1	2
25	2	1

rowno	colno	cellpartno
25	2	2
25	3	1
25	3	2
26	1	1
26	2	1
26	3	1
27	1	1
27	2	1
27	3	1
28	1	1
28	2	1
28	3	1
29	1	1
29	2	1
29	3	1
3	1	1
3	1	2
3	2	1
3	2	2
3	3	1
3	3	2
30	1	1
30	2	1
30	3	1
31	1	1
31	2	1
31	3	1
32	1	1
32	2	1
32	3	1
33	1	1
33	2	1
33	3	1
36	1	1
36	1	2
36	2	1
36	2	2
36	3	1
36	3	2
38	1	1
38	1	2
38	2	1
38	2	2
38	3	1
38	3	2
4	1	1
4	2	1
4	3	1
40	1	1
40	1	2
40	2	1
40	2	2

rowno	colno	cellpartno
40	3	1
40	3	2
5	1	1
5	2	1
5	3	1
6	1	1
6	2	1
6	3	1
7	1	1
7	2	1
7	3	1
8	1	1
8	2	1
8	3	1
9	1	1
9	2	1
9	3	1

Session information

```
sessionInfo()
```

```
## R version 3.2.3 (2015-12-10)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: Fedora 23 (Workstation Edition)
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8          LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8           LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8       LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8          LC_NAME=en_GB.UTF-8
##  [9] LC_ADDRESS=en_GB.UTF-8        LC_TELEPHONE=en_GB.UTF-8
## [11] LC_MEASUREMENT=en_GB.UTF-8    LC_IDENTIFICATION=en_GB.UTF-8
##
## attached base packages:
## [1] methods   stats     graphics  grDevices utils     datasets  base
##
## other attached packages:
##  [1] rrdfqbpresent_0.2.1 rrdfqbcrndex_0.2.1  rrdfqbcrnd0_0.2.1
##  [4] rrdfcdisc_0.2.1     devtools_1.9.1      rrdfqb_0.2.1
##  [7] rrdfancillary_0.2.1 RCurl_1.95-4.7      bitops_1.0-6
## [10] xlsx_0.5.7          xlsxjars_0.6.1      rrdf_2.1.2
## [13] rrdflibs_1.4.0      rJava_0.9-8
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.3     knitr_1.12      magrittr_1.5    roxygen2_5.0.1
##  [5] highr_0.5.1     stringr_1.0.0   tools_3.2.3     htmltools_0.3
##  [9] yaml_2.1.13     digest_0.6.9    formatR_1.2.1   memoise_0.2.1
## [13] evaluate_0.8    rmarkdown_0.9.2 stringi_1.0-1
```