

Derive results in RDF data cube and compare with results in data cube

mja@statgroup.dk

2016-03-08

Derive results in RDF data cube and compare with results in data cube

```
options(width=200) # long lines
```

```
library(xlsx)
```

```
## Loading required package: rJava
```

```
## Loading required package: methods
```

```
## Loading required package: xlsxjars
```

```
library(foreign)
```

```
library(rrdf)
```

```
## Loading required package: rrdflibs
```

```
library(rrdfqb)
```

```
## Loading required package: RCurl
```

```
## Loading required package: bitops
```

```
##
```

```
## Attaching package: 'RCurl'
```

```
## The following object is masked from 'package:rJava':
```

```
##
```

```
##      clone
```

```
## Loading required package: rrdfancillary
```

```
library(rrdfqbcrnd0)
```

```
## Loading required package: rrdgcdisc
```

```
## Loading required package: devtools
```

```

library(rrdfqbcindex)
library(rrdfqbcindexcheck)

obsFile<- system.file("extdata/sample-xpt", "adsl.xpt", package="rrdfqbcindex")
## TODO do not want factors in the data.frame
## http://stackoverflow.com/questions/2851015/convert-data-frame-columns-from-factors-to-characters
## better to use library(SASxport) - see inst/data-raw/create-dm-table-as-csv.Rmd
dataSet<-read.xport(obsFile)
ii <- sapply(dataSet, is.factor)
dataSet[ii] <- lapply(dataSet[ii], as.character)

```

The conversion to character can be avoided by using `library(SASxport)`, see `(../rrdfqbcindex/inst/data-raw/create-dm-table-as-csv.Rmd)`.

```

dataCubeFile<- system.file("extdata/sample-rdf", "DC-DM-sample.ttl", package="rrdfqbcindex")
store <- new.rdf() # Initialize
cat("Reading turtle definition from ", dataCubeFile, "\n")

```

```
## Reading turtle definition from /home/ma/R/x86_64-redhat-linux-gnu-library/3.2/rrdfqbcindex/extdata/
```

```

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

```

```
## [1] "Number of triples: 1450"
```

```

dsdName<- GetDsdNameFromCube( store )
domainName<- GetDomainNameFromCube( store )
forsparqlprefix<- GetForSparqlPrefix( domainName )
custom.prefixes <-Get.qb.crn.prefixes(domainName)

common.prefixes <-data.frame(
  prefix=names( Get.default.crn.prefixes() ),
  namespace=as.character( Get.default.crn.prefixes() )
)

```

```
## Prefix for storing the results of check each measure in the data cube
```

```

validation.measure.prefix<- data.frame(
  prefix=c("validmeas"),
  namespace=c(paste0("http://www.example.org/dc/",tolower(domainName),"/validmeas/"))
)

```

```
prefixes<- rbind(common.prefixes, custom.prefixes, validation.measure.prefix)
```

```
forsparqlprefix<- paste("prefix", paste(prefixes$prefix,":",sep=""), paste("<",prefixes$namespace,">"
```

```

## The qbfile also contains prefixes, which are part of the model
## So not adding the prefixes to the model, but using them for adding further
## information to the model when deriving statistics

```

```
myprefixes<- qb.def.prefixlist(store, prefixes )
```

```
res<- DeriveStatsForCube(store, forsparqlprefix, domainName, dsdName, dataSet, deriveMeasureList=NULL, c
```

```

## difference result 8.59016712714193 in cube 8.59016712714194

## Number of differences 1
## 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>
## prefix dc: <http://purl.org/dc/elements/1.1/>
## prefix dct: <http://purl.org/dc/terms/>
## 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/>
## prefix code: <http://www.example.org/dc/code/>
## prefix dccs: <http://www.example.org/dc/dm/dccs/>
## prefix ds: <http://www.example.org/dc/dm/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#>
## prefix validmeas: <http://www.example.org/dc/dm/validmeas/>
## prefix qbderiv: <http://www.example.org/dc/dm/validmeas/>
##
## select * where {
##     ?s a qb:Observation ;
##     qb:dataSet ds:dataset-DM ;
##     crnd-dimension:procedure ?procedure ;
##     crnd-measure:measure ?measure ;
##     .
##     optional{ ?s qbderiv:result ?result }
## } order by ?s
##
##
## [1] "If the result is <0 x 0> matrix then all value matches"

```

```
print(res)
```

##	s	procedure	measure	result	diff
## [1,]	"ds:obs01"	"code:procedure-count"	"143"	"143"	"0.0"
## [2,]	"ds:obs02"	"code:procedure-count"	"111"	"111"	"0.0"
## [3,]	"ds:obs03"	"code:procedure-count"	"1"	"1"	"0.0"
## [4,]	"ds:obs04"	"code:procedure-count"	"23"	"23"	"0.0"
## [5,]	"ds:obs05"	"code:procedure-count"	"230"	"230"	"0.0"
## [6,]	"ds:obs06"	"code:procedure-max"	"89"	"89"	"0.0"
## [7,]	"ds:obs07"	"code:procedure-mean"	"75.2093023255814"	"75.2093023255814"	"0.0"
## [8,]	"ds:obs08"	"code:procedure-median"	"76"	"76"	"0.0"
## [9,]	"ds:obs09"	"code:procedure-min"	"52"	"52"	"0.0"
## [10,]	"ds:obs10"	"code:procedure-stdev"	"8.59016712714194"	"8.59016712714193"	"0.0"
## [11,]	"ds:obs11"	"code:procedure-percent"	"61.6279069767442"	"61.6279069767442"	"0.0"
## [12,]	"ds:obs12"	"code:procedure-percent"	"38.3720930232558"	"38.3720930232558"	"0.0"
## [13,]	"ds:obs13"	"code:procedure-count"	"86"	"86"	"0.0"
## [14,]	"ds:obs14"	"code:procedure-count"	"53"	"53"	"0.0"
## [15,]	"ds:obs15"	"code:procedure-count"	"33"	"33"	"0.0"
## [16,]	"ds:obs16"	"code:procedure-max"	"86.2"	"86.2"	"0.0"
## [17,]	"ds:obs17"	"code:procedure-mean"	"62.7593023255814"	"62.7593023255814"	"0.0"
## [18,]	"ds:obs18"	"code:procedure-median"	"60.55"	"60.55"	"0.0"
## [19,]	"ds:obs19"	"code:procedure-min"	"34"	"34"	"0.0"
## [20,]	"ds:obs20"	"code:procedure-stdev"	"12.7715435329253"	"12.7715435329253"	"0.0"
## [21,]	"ds:obs21"	"code:procedure-percent"	"0"	"0"	"0.0"
## [22,]	"ds:obs22"	"code:procedure-count"	"0"	"0"	"0.0"
## [23,]	"ds:obs23"	"code:procedure-percent"	"9.30232558139535"	"9.30232558139535"	"0.0"
## [24,]	"ds:obs24"	"code:procedure-count"	"8"	"8"	"0.0"
## [25,]	"ds:obs25"	"code:procedure-percent"	"90.6976744186046"	"90.6976744186046"	"0.0"
## [26,]	"ds:obs26"	"code:procedure-count"	"78"	"78"	"0.0"
## [27,]	"ds:obs27"	"code:procedure-max"	"88"	"88"	"0.0"
## [28,]	"ds:obs28"	"code:procedure-mean"	"74.3809523809524"	"74.3809523809524"	"0.0"
## [29,]	"ds:obs29"	"code:procedure-median"	"76"	"76"	"0.0"
## [30,]	"ds:obs30"	"code:procedure-min"	"56"	"56"	"0.0"
## [31,]	"ds:obs31"	"code:procedure-stdev"	"7.88609384869824"	"7.88609384869824"	"0.0"
## [32,]	"ds:obs32"	"code:procedure-percent"	"47.6190476190476"	"47.6190476190476"	"0.0"
## [33,]	"ds:obs33"	"code:procedure-percent"	"52.3809523809524"	"52.3809523809524"	"0.0"
## [34,]	"ds:obs34"	"code:procedure-count"	"84"	"84"	"0.0"
## [35,]	"ds:obs35"	"code:procedure-count"	"40"	"40"	"0.0"
## [36,]	"ds:obs36"	"code:procedure-count"	"44"	"44"	"0.0"
## [37,]	"ds:obs37"	"code:procedure-max"	"108"	"108"	"0.0"
## [38,]	"ds:obs38"	"code:procedure-mean"	"70.0047619047619"	"70.0047619047619"	"0.0"
## [39,]	"ds:obs39"	"code:procedure-median"	"69.2"	"69.2"	"0.0"
## [40,]	"ds:obs40"	"code:procedure-min"	"41.7"	"41.7"	"0.0"
## [41,]	"ds:obs41"	"code:procedure-stdev"	"14.6534333717795"	"14.6534333717795"	"0.0"
## [42,]	"ds:obs42"	"code:procedure-percent"	"1.19047619047619"	"1.19047619047619"	"0.0"
## [43,]	"ds:obs43"	"code:procedure-count"	"1"	"1"	"0.0"
## [44,]	"ds:obs44"	"code:procedure-percent"	"10.7142857142857"	"10.7142857142857"	"0.0"
## [45,]	"ds:obs45"	"code:procedure-count"	"9"	"9"	"0.0"
## [46,]	"ds:obs46"	"code:procedure-percent"	"88.0952380952381"	"88.0952380952381"	"0.0"
## [47,]	"ds:obs47"	"code:procedure-count"	"74"	"74"	"0.0"
## [48,]	"ds:obs48"	"code:procedure-max"	"88"	"88"	"0.0"
## [49,]	"ds:obs49"	"code:procedure-mean"	"75.6666666666667"	"75.6666666666667"	"0.0"
## [50,]	"ds:obs50"	"code:procedure-median"	"77.5"	"77.5"	"0.0"

```

## [51,] "ds:obs51" "code:procedure-min"      "51"          "51"          "0.0"
## [52,] "ds:obs52" "code:procedure-stdev"    "8.28605059954093" "8.28605059954093" "0.0"
## [53,] "ds:obs53" "code:procedure-percent"  "59.5238095238095" "59.5238095238095" "0.0"
## [54,] "ds:obs54" "code:procedure-percent"  "40.4761904761905" "40.4761904761905" "0.0"
## [55,] "ds:obs55" "code:procedure-count"    "84"          "84"          "0.0"
## [56,] "ds:obs56" "code:procedure-count"    "50"          "50"          "0.0"
## [57,] "ds:obs57" "code:procedure-count"    "34"          "34"          "0.0"
## [58,] "ds:obs58" "code:procedure-max"      "106.1"        "106.1"        "0.0"
## [59,] "ds:obs59" "code:procedure-mean"     "67.2795180722892" "67.2795180722892" "0.0"
## [60,] "ds:obs60" "code:procedure-median"    "64.9"         "64.9"         "0.0"
## [61,] "ds:obs61" "code:procedure-min"      "45.4"         "45.4"         "0.0"
## [62,] "ds:obs62" "code:procedure-stdev"    "14.1235986486909" "14.1235986486909" "0.0"
## [63,] "ds:obs63" "code:procedure-percent"  "0"            "0"            "0.0"
## [64,] "ds:obs64" "code:procedure-count"    "0"            "0"            "0.0"
## [65,] "ds:obs65" "code:procedure-percent"  "7.14285714285714" "7.14285714285714" "0.0"
## [66,] "ds:obs66" "code:procedure-count"    "6"            "6"            "0.0"
## [67,] "ds:obs67" "code:procedure-percent"  "92.8571428571429" "92.8571428571429" "0.0"
## [68,] "ds:obs68" "code:procedure-count"    "78"          "78"          "0.0"

```

TODO(mja): naming of parameters to function *DeriveStatsForCube* to be updated.

End of file.