

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

*mja@statgroup.dk*

*2016-06-07*

## 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-DEMO-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: 3095"
```

```

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, ...)
```

## difference in cube observation	ds:obs010	expected	61.627906977	got	61.6279069767442	relative	
## difference in cube observation	ds:obs011	expected	59.523809524	got	59.5238095238095	relative	
## difference in cube observation	ds:obs012	expected	47.619047619	got	47.6190476190476	relative	
## difference in cube observation	ds:obs016	expected	38.372093023	got	38.3720930232558	relative	
## difference in cube observation	ds:obs017	expected	40.476190476	got	40.4761904761905	relative	
## difference in cube observation	ds:obs018	expected	52.380952381	got	52.3809523809524	relative	
## difference in cube observation	ds:obs019	expected	8.5901671271	got	8.59016712714193	relative	
## difference in cube observation	ds:obs020	expected	8.2860505995	got	8.28605059954093	relative	
## difference in cube observation	ds:obs021	expected	7.8860938487	got	7.88609384869824	relative	
## difference in cube observation	ds:obs028	expected	75.209302326	got	75.2093023255814	relative	
## difference in cube observation	ds:obs029	expected	75.666666667	got	75.6666666666667	relative	
## difference in cube observation	ds:obs030	expected	74.380952381	got	74.3809523809524	relative	
## difference in cube observation	ds:obs052	expected	16.279069767	got	16.2790697674419	relative	
## difference in cube observation	ds:obs053	expected	9.5238095238	got	9.52380952380952	relative	
## difference in cube observation	ds:obs054	expected	13.095238095	got	13.0952380952381	relative	
## difference in cube observation	ds:obs058	expected	48.837209302	got	48.8372093023256	relative	
## difference in cube observation	ds:obs059	expected	55.952380952	got	55.952380952381	relative	6
## difference in cube observation	ds:obs060	expected	65.476190476	got	65.4761904761905	relative	
## difference in cube observation	ds:obs064	expected	34.88372093	got	34.8837209302326	relative	6
## difference in cube observation	ds:obs065	expected	34.523809524	got	34.5238095238095	relative	
## difference in cube observation	ds:obs066	expected	21.428571429	got	21.4285714285714	relative	
## difference in cube observation	ds:obs076	expected	90.697674419	got	90.6976744186046	relative	
## difference in cube observation	ds:obs077	expected	92.857142857	got	92.8571428571429	relative	
## difference in cube observation	ds:obs078	expected	88.095238095	got	88.0952380952381	relative	

```

## difference in cube observation ds:obs082 expected 9.3023255814 got 9.30232558139535 relative
## difference in cube observation ds:obs083 expected 7.1428571429 got 7.14285714285714 relative
## difference in cube observation ds:obs084 expected 10.714285714 got 10.7142857142857 relative
## difference in cube observation ds:obs090 expected 1.1904761905 got 1.19047619047619 relative
## difference in cube observation ds:obs091 expected 12.771543533 got 12.7715435329253 relative
## difference in cube observation ds:obs092 expected 14.123598649 got 14.1235986486909 relative
## difference in cube observation ds:obs093 expected 14.653433372 got 14.6534333717795 relative
## difference in cube observation ds:obs100 expected 62.759302326 got 62.7593023255814 relative
## difference in cube observation ds:obs101 expected 67.279518072 got 67.2795180722892 relative
## difference in cube observation ds:obs102 expected 70.004761905 got 70.0047619047619 relative
## difference in cube observation ds:obs124 expected 96.511627907 got 96.5116279069768 relative
## difference in cube observation ds:obs125 expected 92.857142857 got 92.8571428571429 relative
## difference in cube observation ds:obs126 expected 96.428571429 got 96.4285714285714 relative
## difference in cube observation ds:obs130 expected 3.488372093 got 3.48837209302326 relative
## difference in cube observation ds:obs131 expected 7.1428571429 got 7.14285714285714 relative
## difference in cube observation ds:obs132 expected 3.5714285714 got 3.57142857142857 relative

```

```
print(res)
```

```

##      s      procedure      measure      result      diff
## [1,] "ds:obs001" "code:procedure-count" "86"      "86"      "0.0"
## [2,] "ds:obs002" "code:procedure-count" "84"      "84"      "0.0"
## [3,] "ds:obs003" "code:procedure-count" "84"      "84"      "0.0"
## [4,] "ds:obs004" "code:procedure-percent" "100"     "100"     "0.0"
## [5,] "ds:obs005" "code:procedure-percent" "100"     "100"     "0.0"
## [6,] "ds:obs006" "code:procedure-percent" "100"     "100"     "0.0"
## [7,] "ds:obs007" "code:procedure-count" "53"      "53"      "0.0"
## [8,] "ds:obs008" "code:procedure-count" "50"      "50"      "0.0"
## [9,] "ds:obs009" "code:procedure-count" "40"      "40"      "0.0"
## [10,] "ds:obs010" "code:procedure-percent" "61.627906977" "61.6279069767442" "0.0"
## [11,] "ds:obs011" "code:procedure-percent" "59.523809524" "59.5238095238095" "0.0"
## [12,] "ds:obs012" "code:procedure-percent" "47.619047619" "47.6190476190476" "0.0"
## [13,] "ds:obs013" "code:procedure-count" "33"      "33"      "0.0"
## [14,] "ds:obs014" "code:procedure-count" "34"      "34"      "0.0"

```

##	[15,]	"ds:obs015"	"code:procedure-count"	"44"	"44"	"0.0"
##	[16,]	"ds:obs016"	"code:procedure-percent"	"38.372093023"	"38.3720930232558"	"0.0"
##	[17,]	"ds:obs017"	"code:procedure-percent"	"40.476190476"	"40.4761904761905"	"0.0"
##	[18,]	"ds:obs018"	"code:procedure-percent"	"52.380952381"	"52.3809523809524"	"0.0"
##	[19,]	"ds:obs019"	"code:procedure-std"	"8.5901671271"	"8.59016712714193"	"0.0"
##	[20,]	"ds:obs020"	"code:procedure-std"	"8.2860505995"	"8.28605059954093"	"0.0"
##	[21,]	"ds:obs021"	"code:procedure-std"	"7.8860938487"	"7.88609384869824"	"0.0"
##	[22,]	"ds:obs022"	"code:procedure-n"	"86"	"86"	"0.0"
##	[23,]	"ds:obs023"	"code:procedure-n"	"84"	"84"	"0.0"
##	[24,]	"ds:obs024"	"code:procedure-n"	"84"	"84"	"0.0"
##	[25,]	"ds:obs025"	"code:procedure-median"	"76"	"76"	"0.0"
##	[26,]	"ds:obs026"	"code:procedure-median"	"77.5"	"77.5"	"0.0"
##	[27,]	"ds:obs027"	"code:procedure-median"	"76"	"76"	"0.0"
##	[28,]	"ds:obs028"	"code:procedure-mean"	"75.209302326"	"75.2093023255814"	"0.0"
##	[29,]	"ds:obs029"	"code:procedure-mean"	"75.666666667"	"75.6666666666667"	"0.0"
##	[30,]	"ds:obs030"	"code:procedure-mean"	"74.380952381"	"74.3809523809524"	"0.0"
##	[31,]	"ds:obs031"	"code:procedure-q3"	"82"	"82"	"0.0"
##	[32,]	"ds:obs032"	"code:procedure-q3"	"82"	"82"	"0.0"
##	[33,]	"ds:obs033"	"code:procedure-q3"	"80"	"80"	"0.0"
##	[34,]	"ds:obs034"	"code:procedure-q1"	"69"	"69"	"0.0"
##	[35,]	"ds:obs035"	"code:procedure-q1"	"71"	"71"	"0.0"
##	[36,]	"ds:obs036"	"code:procedure-q1"	"70.5"	"70.5"	"0.0"
##	[37,]	"ds:obs037"	"code:procedure-max"	"89"	"89"	"0.0"
##	[38,]	"ds:obs038"	"code:procedure-max"	"88"	"88"	"0.0"
##	[39,]	"ds:obs039"	"code:procedure-max"	"88"	"88"	"0.0"
##	[40,]	"ds:obs040"	"code:procedure-min"	"52"	"52"	"0.0"
##	[41,]	"ds:obs041"	"code:procedure-min"	"51"	"51"	"0.0"
##	[42,]	"ds:obs042"	"code:procedure-min"	"56"	"56"	"0.0"
##	[43,]	"ds:obs043"	"code:procedure-count"	"86"	"86"	"0.0"
##	[44,]	"ds:obs044"	"code:procedure-count"	"84"	"84"	"0.0"
##	[45,]	"ds:obs045"	"code:procedure-count"	"84"	"84"	"0.0"
##	[46,]	"ds:obs046"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[47,]	"ds:obs047"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[48,]	"ds:obs048"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[49,]	"ds:obs049"	"code:procedure-count"	"14"	"14"	"0.0"
##	[50,]	"ds:obs050"	"code:procedure-count"	"8"	"8"	"0.0"
##	[51,]	"ds:obs051"	"code:procedure-count"	"11"	"11"	"0.0"
##	[52,]	"ds:obs052"	"code:procedure-percent"	"16.279069767"	"16.2790697674419"	"0.0"
##	[53,]	"ds:obs053"	"code:procedure-percent"	"9.5238095238"	"9.52380952380952"	"0.0"
##	[54,]	"ds:obs054"	"code:procedure-percent"	"13.095238095"	"13.0952380952381"	"0.0"
##	[55,]	"ds:obs055"	"code:procedure-count"	"42"	"42"	"0.0"
##	[56,]	"ds:obs056"	"code:procedure-count"	"47"	"47"	"0.0"
##	[57,]	"ds:obs057"	"code:procedure-count"	"55"	"55"	"0.0"
##	[58,]	"ds:obs058"	"code:procedure-percent"	"48.837209302"	"48.8372093023256"	"0.0"
##	[59,]	"ds:obs059"	"code:procedure-percent"	"55.952380952"	"55.952380952381"	"0.0"
##	[60,]	"ds:obs060"	"code:procedure-percent"	"65.476190476"	"65.4761904761905"	"0.0"
##	[61,]	"ds:obs061"	"code:procedure-count"	"30"	"30"	"0.0"
##	[62,]	"ds:obs062"	"code:procedure-count"	"29"	"29"	"0.0"
##	[63,]	"ds:obs063"	"code:procedure-count"	"18"	"18"	"0.0"
##	[64,]	"ds:obs064"	"code:procedure-percent"	"34.88372093"	"34.8837209302326"	"0.0"
##	[65,]	"ds:obs065"	"code:procedure-percent"	"34.523809524"	"34.5238095238095"	"0.0"
##	[66,]	"ds:obs066"	"code:procedure-percent"	"21.428571429"	"21.4285714285714"	"0.0"
##	[67,]	"ds:obs067"	"code:procedure-count"	"86"	"86"	"0.0"
##	[68,]	"ds:obs068"	"code:procedure-count"	"84"	"84"	"0.0"

##	[69,]	"ds:obs069"	"code:procedure-count"	"84"	"84"	"0.0"
##	[70,]	"ds:obs070"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[71,]	"ds:obs071"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[72,]	"ds:obs072"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[73,]	"ds:obs073"	"code:procedure-count"	"78"	"78"	"0.0"
##	[74,]	"ds:obs074"	"code:procedure-count"	"78"	"78"	"0.0"
##	[75,]	"ds:obs075"	"code:procedure-count"	"74"	"74"	"0.0"
##	[76,]	"ds:obs076"	"code:procedure-percent"	"90.697674419"	"90.6976744186046"	"0.0"
##	[77,]	"ds:obs077"	"code:procedure-percent"	"92.857142857"	"92.8571428571429"	"0.0"
##	[78,]	"ds:obs078"	"code:procedure-percent"	"88.095238095"	"88.0952380952381"	"0.0"
##	[79,]	"ds:obs079"	"code:procedure-count"	"8"	"8"	"0.0"
##	[80,]	"ds:obs080"	"code:procedure-count"	"6"	"6"	"0.0"
##	[81,]	"ds:obs081"	"code:procedure-count"	"9"	"9"	"0.0"
##	[82,]	"ds:obs082"	"code:procedure-percent"	"9.3023255814"	"9.30232558139535"	"0.0"
##	[83,]	"ds:obs083"	"code:procedure-percent"	"7.1428571429"	"7.14285714285714"	"0.0"
##	[84,]	"ds:obs084"	"code:procedure-percent"	"10.714285714"	"10.7142857142857"	"0.0"
##	[85,]	"ds:obs085"	"code:procedure-count"	"0"	"0"	"0.0"
##	[86,]	"ds:obs086"	"code:procedure-count"	"0"	"0"	"0.0"
##	[87,]	"ds:obs087"	"code:procedure-count"	"1"	"1"	"0.0"
##	[88,]	"ds:obs088"	"code:procedure-percent"	"0"	"0"	"0.0"
##	[89,]	"ds:obs089"	"code:procedure-percent"	"0"	"0"	"0.0"
##	[90,]	"ds:obs090"	"code:procedure-percent"	"1.1904761905"	"1.19047619047619"	"0.0"
##	[91,]	"ds:obs091"	"code:procedure-std"	"12.771543533"	"12.7715435329253"	"0.0"
##	[92,]	"ds:obs092"	"code:procedure-std"	"14.123598649"	"14.1235986486909"	"0.0"
##	[93,]	"ds:obs093"	"code:procedure-std"	"14.653433372"	"14.6534333717795"	"0.0"
##	[94,]	"ds:obs094"	"code:procedure-n"	"86"	"86"	"0.0"
##	[95,]	"ds:obs095"	"code:procedure-n"	"83"	"83"	"0.0"
##	[96,]	"ds:obs096"	"code:procedure-n"	"84"	"84"	"0.0"
##	[97,]	"ds:obs097"	"code:procedure-median"	"60.55"	"60.55"	"0.0"
##	[98,]	"ds:obs098"	"code:procedure-median"	"64.9"	"64.9"	"0.0"
##	[99,]	"ds:obs099"	"code:procedure-median"	"69.2"	"69.2"	"0.0"
##	[100,]	"ds:obs100"	"code:procedure-mean"	"62.759302326"	"62.7593023255814"	"0.0"
##	[101,]	"ds:obs101"	"code:procedure-mean"	"67.279518072"	"67.2795180722892"	"0.0"
##	[102,]	"ds:obs102"	"code:procedure-mean"	"70.004761905"	"70.0047619047619"	"0.0"
##	[103,]	"ds:obs103"	"code:procedure-q3"	"74.4"	"74.4"	"0.0"
##	[104,]	"ds:obs104"	"code:procedure-q3"	"77.8"	"77.8"	"0.0"
##	[105,]	"ds:obs105"	"code:procedure-q3"	"80.3"	"80.3"	"0.0"
##	[106,]	"ds:obs106"	"code:procedure-q1"	"53.5"	"53.5"	"0.0"
##	[107,]	"ds:obs107"	"code:procedure-q1"	"55.8"	"55.8"	"0.0"
##	[108,]	"ds:obs108"	"code:procedure-q1"	"56.75"	"56.75"	"0.0"
##	[109,]	"ds:obs109"	"code:procedure-max"	"86.2"	"86.2"	"0.0"
##	[110,]	"ds:obs110"	"code:procedure-max"	"106.1"	"106.1"	"0.0"
##	[111,]	"ds:obs111"	"code:procedure-max"	"108"	"108"	"0.0"
##	[112,]	"ds:obs112"	"code:procedure-min"	"34"	"34"	"0.0"
##	[113,]	"ds:obs113"	"code:procedure-min"	"45.4"	"45.4"	"0.0"
##	[114,]	"ds:obs114"	"code:procedure-min"	"41.7"	"41.7"	"0.0"
##	[115,]	"ds:obs115"	"code:procedure-count"	"86"	"86"	"0.0"
##	[116,]	"ds:obs116"	"code:procedure-count"	"84"	"84"	"0.0"
##	[117,]	"ds:obs117"	"code:procedure-count"	"84"	"84"	"0.0"
##	[118,]	"ds:obs118"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[119,]	"ds:obs119"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[120,]	"ds:obs120"	"code:procedure-percent"	"100"	"100"	"0.0"
##	[121,]	"ds:obs121"	"code:procedure-count"	"83"	"83"	"0.0"
##	[122,]	"ds:obs122"	"code:procedure-count"	"78"	"78"	"0.0"

```

## [123,] "ds:obs123" "code:procedure-count" "81" "81" "0.0"
## [124,] "ds:obs124" "code:procedure-percent" "96.511627907" "96.5116279069768" "0.0"
## [125,] "ds:obs125" "code:procedure-percent" "92.857142857" "92.8571428571429" "0.0"
## [126,] "ds:obs126" "code:procedure-percent" "96.428571429" "96.4285714285714" "0.0"
## [127,] "ds:obs127" "code:procedure-count" "3" "3" "0.0"
## [128,] "ds:obs128" "code:procedure-count" "6" "6" "0.0"
## [129,] "ds:obs129" "code:procedure-count" "3" "3" "0.0"
## [130,] "ds:obs130" "code:procedure-percent" "3.488372093" "3.48837209302326" "0.0"
## [131,] "ds:obs131" "code:procedure-percent" "7.1428571429" "7.14285714285714" "0.0"
## [132,] "ds:obs132" "code:procedure-percent" "3.5714285714" "3.57142857142857" "0.0"

```

End of file.