Минский инновационный университет

Кафедра информационных технологий

Гедранович О.Б.

Модели и технологии оперативного анализа данных

Лекция 5 OLAP operations in R

Facts and Dimensions tables. Multi-dimensional Cube. OLAP operations: Slice, Dice, Roll-Up, Drill-down, Pivot.

Facts and Dimensions tables

OLAP (Online Analytical Processing) is a very common way to analyze raw transaction data by aggregating along different combinations of dimensions. This is a well-established field in Business Intelligence / Reporting. In this post, I will highlight the key ideas in OLAP operation and illustrate how to do this in R.

The core part of OLAP is a so-called "multi-dimensional data model", which contains two types of tables; "Fact" table and "Dimension" table.

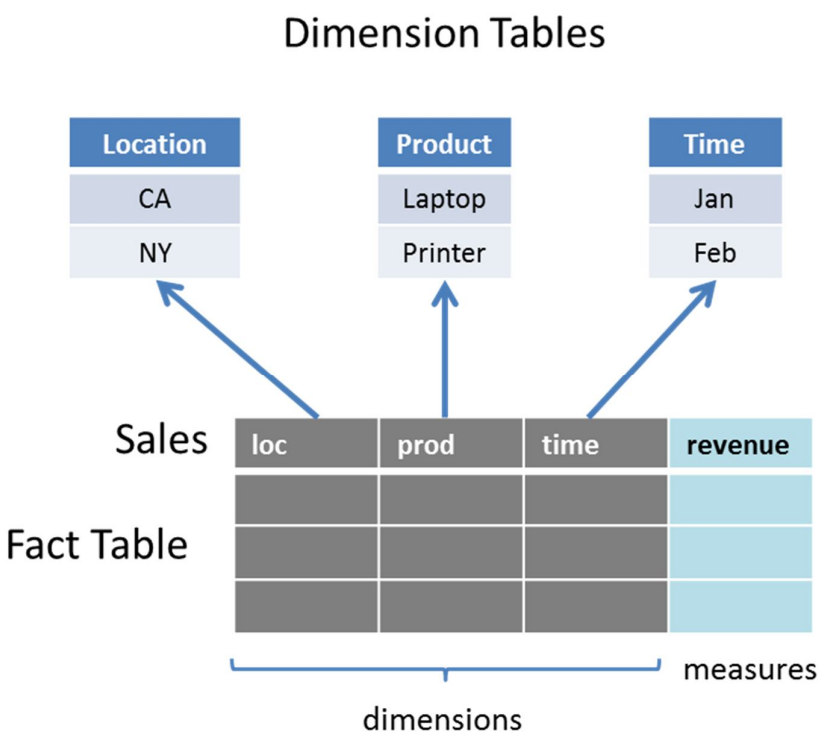
A Fact table contains records each describe an instance of a transaction. Each transaction record contains categorical attributes (which describe contextual aspects of the transaction, such as space, time, user) as well as numeric attributes (called "measures" which describe quantitative aspects of the transaction, such as number of items sold, price).

A Dimension table contains records that further elaborate the contextual attributes, such as user profile data, location details, etc.

In a typical setting of Multi-dimensional model:

- Each fact table contains foreign keys that reference the primary key of multiple dimension tables. In the simplest form it is called a STAR schema.
- Dimension tables can contain foreign keys that reference other dimensional tables. This provides a sophisticated detail breakdown of the contextual aspects. This is also called a SNOWFLAKE schema.
- Also this is not a hard rule, Fact table tends to be independent of other Fact table and usually doesn't contain reference pointer among each other.
- However, different Fact table usually share the same set of dimension tables. This is also called GALAXY schema.
- But it is a hard rule that Dimension table NEVER points / references Fact table.

A simple STAR schema is shown in following diagram.



Each dimension can also be hierarchical so that the analysis can be done at different degree of granularity. For example, the time dimension can be broken down into days, weeks, months, quarter and annual; Similarly, location dimension can be broken down into countries, states, cities, etc.

Here we first create a sales fact table that records each sales transaction.

```
# Setup the dimension tables
state_table <- data.frame(key = c("CA", "NY", "WA", "ON", "QU"),
                          name = c("California", "New York", "Washington",
"Ontario", "Quebec"),
                          country = c("USA", "USA", "USA", "Canada", "Canada"))
row.names(state_table) <- state_table$key
month_table <- data.frame(key = 1:12,
                          desc = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
"Jul", "Aug", "Sep", "Oct", "Nov", "Dec"),
                          quarter = c("Q1", "Q1", "Q1", "Q2", "Q2", "Q2", "Q3",
"Q3", "Q3", "Q4", "Q4", "Q4"))
row.names(month_table) <- month_table$key
prod_table <- data.frame(key = c("Printer", "Tablet", "Laptop"),
                         price = c(225, 570, 1120))
row.names(prod_table) <- prod_table$key
```

```
# Function to generate the Sales table
gen_sales <- function(no_of_recs) {
  # Generate transaction data randomly
  loc <- sample(state_table$key, no_of_recs,
                replace = TRUE, prob = c(2, 2, 1, 1, 1))
  country <- state_table[loc, ]$country
  time_month <- sample(month_table$key, no_of_recs, replace = TRUE)
  time_q <- month_table[time_month, ]$quarter
  time_year <- sample(c(2012, 2013), no_of_recs, replace = TRUE)
  prod <- sample(prod_table$key, no_of_recs, replace = TRUE, prob = c(1, 3, 2))
  unit <- sample(c(1, 2), no_of_recs, replace = TRUE, prob = c(10, 3))
  amount <- unit * prod_table[prod, ]$price
  sales <- data_frame(month = time_month,
                      quarter = time_q,
                      year = time_year,
                      loc = loc,
                      country = country,
                      prod = prod,
                      unit = unit,
                      amount = amount)
  # Sort the records by time order
  sales <- sales[order(sales$year, sales$month), ]
  sales
}

# Now create the sales fact table
sales_fact <- gen_sales(500)

# Look at a few records
head(sales_fact, 5)
  month year loc   prod unit amount
1     1 2012  NY Laptop    1    225
2     1 2012  CA Laptop    2    450
3     1 2012  ON Tablet    2   2240
4     1 2012  NY Tablet    1   1120
5     1 2012  NY Tablet    2   2240

# Look at all the records
sales_fact %>% View
```

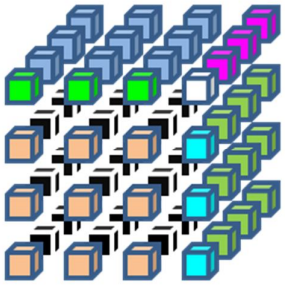
Multi-dimensional Cube

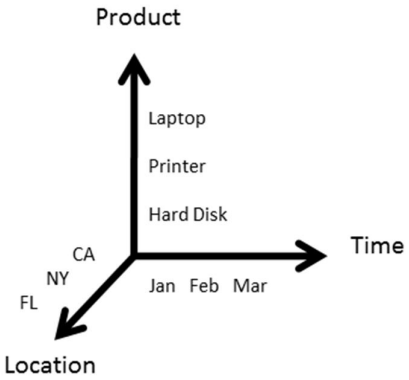
Now, we turn this fact table into a hypercube with multiple dimensions. Each cell in the cube represents an aggregate value for a unique combination of each dimension.

Sales Fact Table

state	product	year	month	unit	amount

Hypercube





3D Cuboid

Revenue of (Laptop, NY, Feb)

2D Cuboids

Revenue sum of (Laptop, Feb) over (locations)

Revenue sum of (NY, Feb) over (Products)

Revenue sum of (Laptop, NY) over (Time)

1D Cuboids

Revenue sum of (Feb) over (products, locations)

Revenue sum of (CA) over (products, time)

Revenue sum of (Laptop) over (time, locations)

0D Cuboid

Revenue sum over (products, time, locations)

```
# Build up a cube
revenue_cube <- tapply(sales_fact$amount,
                       sales_fact[, c("prod", "month", "year", "country")],
                       FUN = function(x) {return(sum(x))} )


# Showing the cells of the cude
revenue_cube
, , year = 2012, country = Canada

         month
prod         1    2    3    4    5    6    7    8    9   10   11   12
  Laptop   450  225  900  450  675  675  225  900   NA  225  225  450
  Printer   NA  570   NA  570  570 1140  570  570  570 1140  570   NA
  Tablet  1120 5600 5600 5600 2240 1120 1120 5600 8960 3360 2240 2240
```

```
, , year = 2013, country = Canada

         month
prod         1    2    3    4    5    6    7     8    9   10   11   12
  Laptop   900  450  450  675  450  225  675   225  675  450  450   NA
  Printer 1140 1710   NA   NA 2280 1140   NA  1140 1140   NA 1710  570
  Tablet  3360 1120 2240 4480 4480 3360 2240 10080 5600 4480 6720 3360

, , year = 2012, country = USA

         month
prod         1    2     3    4     5    6     7    8     9    10   11    12
  Laptop  2475  900  1575 2025   675 1350  1575 1800  2025   675 1350  1125
  Printer 1710 3420  3420 2280  2280 1710  1710 1140  2280  2280   NA  1140
  Tablet  1120 8960 13440 8960 11200 8960 14560 6720 12320 21280 8960 15680

, , year = 2013, country = USA

         month
prod         1     2     3     4    5    6     7     8    9   10    11   12
  Laptop  2025   900   900  3150 1575  450  1125  1575 2025 1350  2925  900
  Printer 3420   570  1710  1140 1140 1140  1140   570  570 2280  2280 1710
  Tablet  4480 11200 10080 13440 7840 7840 11200 13440 7840 5600 14560 7840



dimnames(revenue_cube)

$prod
[1] "Laptop"  "Printer" "Tablet"

$month
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12"

$year
[1] "2012" "2013"

$country
[1] "Canada" "USA"
```

OLAP Operations

Here are some common operations of OLAP

- Slice
- Dice
- Rollup
- Drilldown
- Pivot

"Slice" is about fixing certain dimensions to analyze the remaining dimensions. For example, we can focus in the sales happening in "2012", "Jan", or we can focus in the sales happening in "2012", "Jan", "Tablet".

```
# cube data in Jan, 2012
revenue_cube[, "1", "2012", ]
         country
prod      Canada  USA
  Laptop     450 2475
  Printer     NA 1710
  Tablet    1120 1120

# cube data in Jan, 2012
revenue_cube["Tablet", "1", "2012",]
Canada    USA
  1120   1120
```

"Dice" is about limited each dimension to a certain range of values, while keeping the number of dimensions the same in the resulting cube. For example, we can focus in sales happening in [Jan/Feb/Mar, Laptop/Tablet, USA].

```
revenue_cube[c("Tablet", "Laptop"),
             c("1", "2", "3"),
             ,
             "USA"]

, , year = 2012

        month
prod        1    2     3
  Tablet 1120 8960 13440
  Laptop 2475  900  1575

, , year = 2013

        month
prod        1     2     3
  Tablet 4480 11200 10080
  Laptop 2025   900   900
```

"Roll-up" is about applying an aggregation function to collapse a number of dimensions. For example, we want to focus in the annual revenue for each product and collapse the location dimension (i.e.: we don't care where we sold our product).

```
apply(revenue_cube, c("year", "prod"),
      FUN=function(x) {return(sum(x, na.rm=TRUE))})
      prod
year   Laptop Printer Tablet
  2012  22950   29640 176960
  2013  24525   28500 166880
```

"Drill-down" is the reverse of "roll-up". It is performed by either of the following ways: by stepping down a concept hierarchy for a dimension; or by introducing a new dimension. For example, we want to step down from countries to states (then we form a new cube out of our initial data frame).

```
revenue_cube1 <-
  tapply(sales_fact$amount,
         sales_fact[, c("prod", "month", "year", "loc")],
         FUN = function(x) {return(sum(x))} )

dimnames(revenue_cube1)
$prod
[1] "Laptop"  "Printer" "Tablet"
$month
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12"
$year
[1] "2012" "2013"
$loc
[1] "CA" "NY" "ON" "QU" "WA"
```

"Pivot" is about analyzing the combination of a pair of selected dimensions. For example, we want to analyze the revenue by products and countries, probably only on February 2012. Also we can rotate the axis.

```
apply(revenue_cube, c("prod", "country"),
      FUN = function(x) {return(sum(x, na.rm = TRUE))} ) %>%
  t()
```

```
        prod
country  Laptop Printer Tablet
  Canada  11025   17100  96320
  USA     36450   41040 247520

revenue_cube[ , "2", "2012", ] %>%
  t()
        prod
country  Laptop Printer Tablet
  Canada    225     570   5600
  USA       900    3420   8960
```

However, since R is doing all the processing in RAM. This requires your data to be small enough so it can fit into the local memory in a single machine.

Литература:

OLAP-системы [Electronic resource] / TAdviser. – Mode of access: http://www.tadviser.ru/index.php/Статья:OLAP-системы. – Date of access: 20.01.2017.

Ho, R. OLAP operation in R [Electronic resource] / Pragmatic Programming Techniques. – Mode of access: http://horicky.blogspot.com.by/2013/07/olap-operation-in-r.html. – Date of access: 20.01.2017.