# sliced_row_by_row

*Xuelong Wang*

*2017-11-14*

## What

The following code is trying to read the data row-by-row and calculate statistics for each slice. In this case, the statistics is just the standardized mean vector for each slice $M_h$

## How

Using the sufficient statistics which can be calculated row by row

### Based on the first 3 steps of SIR

(1) Standardize $x$ to $\tilde{x} = \Sigma_{xx}^{-1/2}(x_i - \bar{x})$
(2) Divide range ofy into H slices, $I_1, \ldots, I_H$
(3) Within each slice, compute the sample mean $\hat{m}_h$ of the $\tilde{x}_i$'s

### Sufficient Statistis for calculating the slice mean vector

In each slice $h$, we want to find the $m_h$

$$\begin{aligned} m_h &= \frac{1}{n_h}(\sum_{i=1}^{n_h} \tilde{x}) \\ &= \frac{1}{n_h}(\sum_{i=1}^{n_h} \Sigma_{xx}^{-1/2}(x_i - \bar{x})) \\ &= \Sigma_{xx}^{-1/2} \frac{1}{n_h} \sum_{i=1}^{n_h}(x_i - \bar{x}) \\ &= \Sigma_{xx}^{-1/2}(\bar{x}_{ih} - \bar{x}) \\ &= \Sigma_{xx}^{-1/2}(m_h^* - \bar{x}) \end{aligned}$$

Where the $m_h^*$ is just the mean vector of original (non-standardized).

$$
\begin{aligned}
cor(X) &= \frac{1}{n}\left(X-\bar{X}\right)^T\left(X-\bar{X}\right) \\
&= \sum_{i=1}^{n}\left(x_i-\bar{x}\right)\left(x_i-\bar{x}\right)^T \\
\text{where } x_i \text{ is the } X_{i\cdot}^T &= \frac{1}{n}\sum_{i=1}^{n}(x_i x_i^T) - \bar{x}\bar{x}^T
\end{aligned}
$$

Thus the only thing we need to record for each slice $h$ is $x_i x_i^T$, $x_i = \frac{1}{n}\sum x_i$ and the total number $n_h$

$$
\Sigma_{xx}^{-1/2}(\bar{x}_{ih} - n_h\bar{x})
$$

```
library(tidyverse)
library(data.table)
```

```
# simulation data
set.seed(1014)
n <- 10^(3)
y <- rnorm(sd = 50, n)
x <- rnorm(n*3,c(1, 20, 100)) %>% matrix(. , nrow = n, ncol = 3, byrow = TRUE)

# assume we can read all the data in the ram and sort the data by y's value
labels_factor <- LETTERS[1:8] %>% as.factor()
data_set <- data.table(y = y, x = x) %>% setorder(., y)
labels <- data_set$y %>% cut(., breaks = 8, labels = labels_factor)
data_set_labled <- data.table(data_set, labels = labels)

# iterating each row to calculate covariance matrix and sliced mean vector
x_sum <- numeric(3)
x_x_t <- rep(0, ncol(x)) %>% diag()

x_each_slice <- matrix(0, nrow = length(labels_factor), ncol = 3)
rownames(x_each_slice) <- levels(labels_factor)

total_number_each_slice <- numeric(8)
names(total_number_each_slice) <- labels_factor %>% levels()
for (i in (1:nrow(data_set_labled))) {
  x_i <- data_set_labled[i,!c("y", "labels"), with = FALSE]
  labels <- data_set_labled[i, "labels", with = FALSE]
  x_x_t <- x_x_t + tcrossprod(x_i %>% unlist() %>% matrix(., ncol = 1)) # x*x^t
  x_sum <- x_sum + x_i
  slice_posistion <- match(labels[[1]], labels_factor)
  x_each_slice[slice_posistion,] <- unlist(x_i) + x_each_slice[slice_posistion,]
  total_number_each_slice[slice_posistion] <- total_number_each_slice[slice_posistion] + 1
}

# calculate the covariance matrix
x_mean <- (x_sum/n) %>% unlist()
covariance <- (x_x_t - n*tcrossprod(x_mean))/n
var(x) - covariance
```

```r
# calcuate sliced mean vector
h_mean_unnormalized <- sweep(x_each_slice, MARGIN = 1, total_number_each_slice, '/')
h_mean <- (sweep(h_mean_unnormalized, MARGIN = 2, x_mean, FUN = "-")) %*% solve(covariance)^(-1/2)
h_mean
```