

TO DO:

ALGORITMO:

Cosa passiamo a main? -> S(1:n), beta, sigma, theta, csi(1:n)
[RAGIONARE]

```
MAIN <- function( S_0(1:n), beta_0, sigma_0, theta_0, csi(1:n)_0, R =  
1000)  
{  
  For r=1:R  
  - passo a  
    - update_roba -> LEZIONE 25/11 DA PENSARE  
  
  - passo b  
    - for (j = 1:k)  
      ### XI È UN VETTORE p-dim !!!!  
      xi_star[r][j] <- update_xi(xi_star[r-1]  
[j], ...) -> CERCARE SU INTERNET COME FARE, DA PENSARE  
  
  - passo c  
    -calcolare m1_barra <- calcolo m1_barra(S(i:n))  
    -sigma(r) <- update_sigma(m1, m1_barra, k,  
sigma(r-1), beta(r-1), freq(k))  
    -theta(y) <- update_theta(m1_barra, k, n,  
theta(r-1), sigma(r))  
  
  - passo d  
    - beta(r) <- update_beta(m1_barra, n, beta(i-1))  
  
  (Alla fine: summary della clustering structure)  
}
```

PASSO a

PASSO b

PASSO c

```
update_sigma <- function(m1, m1_barra, k, sigma_v, theta_v, freq(k),  
sd = 2)  
{
```

```
  sigma_star <- cambio di variabile sigma (sigma_v)
```

MH:

```

    {
        - Sampo y da normale_uni -> y = sigma_star +
N(0,sd)
        - Sampo u da U(0,1)
        - alpha <- calcolo_alpha_sigma(y, sigma_star,
theta_v, ...)
        - vedo se accettare o no
    } -> ottengo y nuovo passo

    - sigma <- inv_cambio_sigma(y)
    Return sigma
}

update_theta<- function(m1_barra, k, n, theta_v, sigma_n, sd = 2)
{
    theta_star <- cambio di variabile theta (theta_v)

    MH:
    {
        - Sampo y da normale_uni -> y = theta_star +
N(0,sd)
        - Sampo u da U(0,1)
        - alpha <- calcolo_alpha_theta(y, theta_star,
m1_barra, k, n, sigma_n)
        - vedo se accettare o no
    } -> ottengo y nuovo passo

    - theta <- inv_cambio_theta(y)
    Return theta
}

PASSO d
update_beta <- function(m1_barra, n, beta_vecchio, sd = 2)
{
    beta_star <- cambio di variabile beta (beta_vecchio)

    MH:
    {
        - Sampo y da normale_uni -> y = beta_star +
N(0,sd)
        - Sampo u da U(0,1)
        - alpha <- calcolo_alpha_beta(y, beta_star, ...)
        - vedo se accettare o no
    } -> ottengo y nuovo passo

    - beta <- inv_cambio_beta(y)
    Return beta
}

```

FUNZIONI AUSILIARIE

- Calcolo $m1_barra(S(i:n))$
- Calcolo $m1$
- $update_xi(xi_star[r-1][j], \dots)$
- cambio di variabile σ (σ_v) -> FORSE E UGUALE A β ???
- $calcolo_alpha_sigma(y, \sigma_star, \theta_v)$
- $inv_cambio_sigma(y)$
- cambio di variabile θ (θ_v)
- $calcolo_alpha_theta(y, \theta_star, m1_barra, k, n, \sigma_n)$
- $inv_cambio_theta(y)$
- cambio di variabile β ($\beta_vecchio$) -> UGUALE A σ
- $calcolo_alpha_beta(y, \beta_star)$ -> DIFFERENZIARE
- $inv_cambio_beta(y)$
-