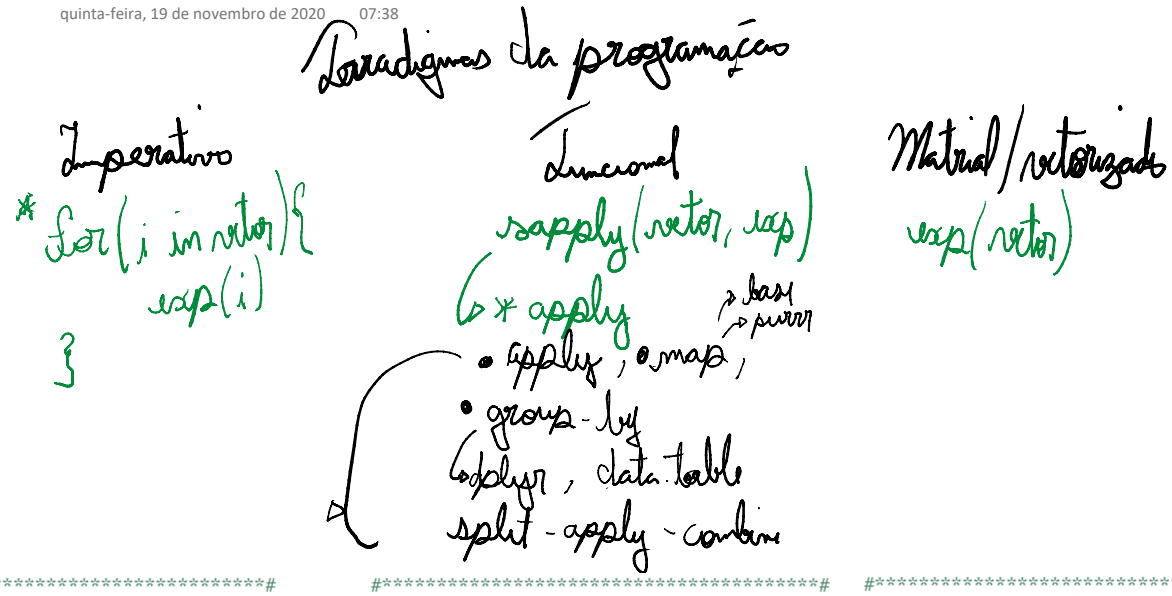
quinta-feira, 19 de novembro de 2020 07:38




```r
#########################################
#------------ Imperativo ---------------#
vetor <- c(5, 1.5, 3, 0.75)
N <- length(vetor)
vetor_exp <- numeric(N) #vector("numeric", N)
for (i in 1:N) {
   vetor_exp[i] <- exp(vetor[i])
}
```

```r
#########################################
#------------ Funcional ----------------#
vetor <- c(5, 1.5, 3, 0.75)
vetor_exp <- sapply(vetor, exp)
```

```r
#########################################
#------------ Vetorizado ---------------#
vetor <- c(5, 1.5, 3, 0.75)
vetor_exp <- exp(vetor)
```

```r
#########################################
#------------ Imperativo ---------------#
niveis <- c(0.1, 5, 1.75, 3)
N <- length(niveis)
acao <- character(N)
for (i in 1:N) {
   if (niveis[i] >= 2) {
      acao[i] <- "SIM"
   } else {
      acao[i] <- "NAO"
   }
}
```

```r
#########################################
#------------ Funcional ----------------#
niveis <- c(0.1, 5, 1.75, 3)
decisao <- function(nivel) {
   if (nivel >= 2) {
      return("SIM")
   } else {
      return("NAO")
   }
}
#decisao(1)
#decisao(2.2)
acao <- sapply(niveis, decisao)
```

```r
#########################################
#------------ Vetorizado ---------------#
niveis <- c(0.1, 5, 1.75, 3)
acao <- ifelse(niveis >= 2, "SIM", "NAO")
```

```r
# Mais complexo, grupo de elementos
## Dados
proj <- c(1, 1, 1, 1, 2, 2, 2, 2, 2,
          3, 3, 3, 3)
x <-    c(1, 2, 3, 4, 1, 2, 3, 4, 5,
          1, 2, 3, 4)
y <-    c(1, 2, 2, 3, 8, 6, 5, 3, 2,
          2, 5, 8, 9)
d <- data.frame(proj = proj, x = x, y = y)
```

```r
#########################################
#------------ Imperativo ---------------#
un_proj <- unique(d$proj)
N <- length(un_proj)
coefs <- data.frame(proj = integer(N),
                    b0 = numeric(N),
                    b1 = numeric(N))
for (i in 1:N) {
   d_i <- subset(d, d$proj == un_proj[i])
   lm_i <- lm(y ~ x, d_i)
   coef_i <- coef(lm_i)
   coefs[i, "proj"] <- un_proj[i]
   coefs[i, "b0"] <- coef_i[[1]]
   coefs[i, "b1"] <- coef_i[[2]]
}
```

```r
#########################################
#------------ Funcional ----------------#
ajuste_lm <- function(dados) {
   lm_func <- lm(y ~ x, dados)
   coef_func <- coef(lm_func)
   dados_func <- data.frame(proj = dados$proj[1],
                            b0 = coef_func[[1]],
                            b1 = coef_func[[2]])
   return(dados_func)
}
# ajuste_lm(d) # testar função ajuste_lm
d_split <- split(d, d$proj) # split
aplicar_lm <- lapply(d_split, ajuste_lm) # apply
coefs <- Reduce(rbind, aplicar_lm) # combine
```

```r
#########################################
#------------ Vetorizado ---------------#
# ?
```

quinta-feira, 19 de novembro de 2020 08:50