Working on Problems

Neeraj Jain

17-02-2020

1. Function to create dummy variable

```
rm(list = ls())
#Generating Data
df <- data.frame(Signal = sample(x = c("A", "B", "C"), size = 50, replace = T))
df$Signal <- as.character(df$Signal)

#Manually Creating the variables
df$A <- ifelse(df$Signal == "A", 1, 0)
df$B <- ifelse(df$Signal == "B", 1, 0)

#Writing function
y <- sample(x = c("A", "B", "Aadi", "Abhijit"), size = 50, replace = TRUE)

Dummy <- function(x) {
  mat <- matrix(data = x, ncol = 1)
  uniquex <- unique(x)
  n <- length(uniquex)
  for(i in 1:(n-1)) {
    mat <- cbind(mat, ifelse(x == uniquex[i], 1, 0))
    #mat[, i+1] <- ifelse(x == uniquex[i], 1, 0) #donot do this
  }
  return(mat)
}

#Ex:
Dum <- Dummy(y)
head(Dum)
```

```
##      [,1]   [,2] [,3] [,4]
## [1,] "B"    "1"  "0"  "0"
## [2,] "A"    "0"  "1"  "0"
## [3,] "A"    "0"  "1"  "0"
## [4,] "Aadi" "0"  "0"  "1"
## [5,] "Aadi" "0"  "0"  "1"
## [6,] "A"    "0"  "1"  "0"
```

```
#alternative: In this case, we first created matrix of desired dimensions, then
#   according we replace them from desired value.
Dummy1 <- function(x) {
  uniquex <- unique(x)
  n <- length(uniquex)
  mat <- matrix(data = NA, nrow = length(x), ncol = (n - 1))
```

```
  for(i in 1:(n-1)) {
    #mat <- cbind(mat, ifelse(x == uniquex[i], 1, 0))
    mat[, i] <- ifelse(x == uniquex[i], 1, 0)
  }
  return(mat)
}

#Ex:
Dum1 <- Dummy1(y)
head(Dum1)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    1    0
## [4,]    0    0    1
## [5,]    0    0    1
## [6,]    0    1    0
```

2. Arranging data into order.

```
#For atomic Vector
b <- c(23, 43, 12, 45, 55)
o <- order(b, decreasing = T)
b[o]
```

```
## [1] 55 45 43 23 12
```

```
#For data.frame
Df <- data.frame(Name = letters[1:20], marks = rnorm(20, 50, 10))

Df <- Df[order(Df$marks, decreasing = T), ]
head(Df)
```

```
##    Name    marks
## 19    s 66.17669
## 8     h 63.16996
## 12    l 63.06406
## 15    o 60.16833
## 16    p 58.00614
## 3     c 57.80463
```

3. General Problem: Count Consecutive zero

```
#count number of zero after one
a <- sample(x = c(1, 0), size = 50, replace = T)

sum(diff(a) < 0)
```

```
## [1] 13
```

```r
#count number of zero after two consecutive one
suppressWarnings( sum((a == 0 & c(NA, a) == 1 & c(NA, NA, a) == 1)[1:length(a)],
                      na.rm= T) )
```

```
## [1] 5
```