# Slot_Machine_Vectorized

*Saurav Singh*

*17 January 2017*

## Slot Machine Vectorized

In this i have written the modified code for slot machine which executes faster than the previous version of code written.Later, we will execute this code for 10 million times.
Firstly,i have modified get_symbol() function to get_many_symbols() which returns the nth rows of randomed outputs of slot machine.

```r
get_many_symbols <- function(n) {
  wheel <- c("DD", "7", "BBB", "BB", "B", "C", "0")
  vec <- sample(wheel, size = 3 * n, replace = TRUE,
                prob = c(0.03, 0.03, 0.06, 0.1, 0.25, 0.01, 0.52))
  matrix(vec, ncol = 3)
}
get_many_symbols(3)
```

```
##      [,1] [,2]  [,3]
## [1,] "0"  "BB"  "BB"
## [2,] "0"  "BBB" "BB"
## [3,] "0"  "BB"  "0"
```

symbols variable is basically made like a matrix and is kind a lookup table created for scoring each type of a phase in the slot machine.

```r
symbols <- matrix(
  c("DD", "DD", "DD","C", "DD", "0",
    "B", "B", "B","B", "BB", "BBB",
    "C", "C", "0","7", "DD", "DD"), nrow = 6, byrow = TRUE)
```

score_many() function is modified accordingly to use variable **symbols** like a lookup table and be compatible with other functions of this modified version.

```r
score_many <- function(symbols) {
  cherries <- rowSums(symbols == "C")
  diamonds <- rowSums(symbols == "DD")
  prize <- c(0, 2, 5)[cherries + diamonds + 1]
  prize[!cherries] <- 0
  same <- symbols[, 1] == symbols[, 2] &
    symbols[, 2] == symbols[, 3]
  payoffs <- c("DD" = 100, "7" = 80, "BBB" = 40,
               "BB" = 25, "B" = 10, "C" = 10, "0" = 0)
  prize[same] <- payoffs[symbols[same, 1]]
  bars <- symbols == "B" | symbols == "BB" | symbols == "BBB"
  all_bars <- bars[, 1] & bars[, 2] & bars[, 3] & !same
  prize[all_bars] <- 5
  two_wilds <- diamonds == 2
  one <- two_wilds & symbols[, 1] != symbols[, 2] &
    symbols[, 2] == symbols[, 3]
  two <- two_wilds & symbols[, 1] != symbols[, 2] &
    symbols[, 1] == symbols[, 3]
  three <- two_wilds & symbols[, 1] == symbols[, 2] &
    symbols[, 2] != symbols[, 3]
  prize[one] <- payoffs[symbols[one, 1]]
  prize[two] <- payoffs[symbols[two, 2]]
  prize[three] <- payoffs[symbols[three, 3]]
  one_wild <- diamonds == 1
  wild_bars <- one_wild & (rowSums(bars) == 2)
  prize[wild_bars] <- 5
  one <- one_wild & symbols[, 1] == symbols[, 2]
  two <- one_wild & symbols[, 2] == symbols[, 3]
  three <- one_wild & symbols[, 3] == symbols[, 1]
  prize[one] <- payoffs[symbols[one, 1]]
  prize[two] <- payoffs[symbols[two, 2]]
  prize[three] <- payoffs[symbols[three, 3]]
  unname(prize * 2^diamonds)
}
```

play() function is modified to play_many() which plays slot machine for n times.

```
play_many <- function(n) {
  symb_mat <- get_many_symbols(n = n)
  data.frame(w1 = symb_mat[,1], w2 = symb_mat[,2],w3 = symb_mat[,3]
             , prize = score_many(symb_mat))
}
play_many(3)
```

```
##   w1  w2 w3 prize
## 1  7   0 BB     0
## 2  0   B  0     0
## 3 BB BBB  0     0
```

Finally,testing the play_many() function for 10 million times.

```
system.time(play_many(10000000))
```

```
##    user  system elapsed
##  20.296   0.948  21.253
```