

# Best\_code\_prac\_lab

*Samuel Perini*

6/11/2018

## 1.2 Task: Obscure Code

```
rm(list = ls())

myIterAtoR.max <- 5
second_iterator.max<-7
col.NUM= 10
row.cnt =10
fwzy45 <- matrix(rep(1, col.NUM*row.cnt),nrow=row.cnt)
for(haystack in (2-1):col.NUM){
  for(needle in 1:row.cnt) {
    if(haystack>=myIterAtoR.max){
      fwzy45[haystack, needle]<-NA}
    }}
fwzy45
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    1    1    1    1    1    1
## [2,]    1    1    1    1    1    1    1    1    1    1
## [3,]    1    1    1    1    1    1    1    1    1    1
## [4,]    1    1    1    1    1    1    1    1    1    1
## [5,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [6,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [7,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [8,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [9,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [10,]  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
```

```
for(i in 1:ncol(fwzy45)){
  for(j in 1:nrow(fwzy45)) {
    if(i>=myIterAtoR.max){
      fwzy45[i, j]<-NA}
    }}
fwzy45
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    1    1    1    1    1    1
## [2,]    1    1    1    1    1    1    1    1    1    1
## [3,]    1    1    1    1    1    1    1    1    1    1
## [4,]    1    1    1    1    1    1    1    1    1    1
## [5,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [6,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [7,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [8,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [9,]   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## [10,]  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
```

### 1.3 Task: Better Formatting

```
simulate_genotype <- function( q, N=100 ) {  
  if( length(q)==1 ){  
    p <- (1 - q)  
    f_gt <- c(p^2, 2*p*q, q^2) # AA, AB, BB  
  }else{  
    f_gt<-q  
  }  
  tmp <- sample( c('AA','AB','BB'), size =N, prob=f_gt, replace=T )  
  return(tmp)  
}
```

### 1.4 Task: Hidden Variable

```
rev(month.abb)  
  
## [1] "Dec" "Nov" "Oct" "Sep" "Aug" "Jul" "Jun" "May" "Apr" "Mar" "Feb"  
## [12] "Jan"
```

### 1.5 Task: Pipeline-friendly Function

```
library(tidyverse)  
df = tibble(x=runif(5),y=runif(5))  
my_filter <- function(threshold = 1, data, scalar = 5) {  
  data[data >= threshold] <- NA  
  data <- data * scalar  
  return(data)  
}
```

```
my_filter(data = df) %>% summary()
```

```
##           x           y  
## Min.      :0.9514   Min.      :1.606  
## 1st Qu.:1.1288   1st Qu.:2.865  
## Median :1.6813   Median :3.053  
## Mean     :2.2684   Mean      :3.326  
## 3rd Qu.:3.5767   3rd Qu.:4.249  
## Max.     :4.0038   Max.      :4.858
```

```
my_filter2 <- function(data, threshold = 1, scalar = 5) {  
  data[data >= threshold] <- NA  
  data <- data * scalar  
  return(data)  
}  
my_filter2(df) %>% summary()
```

```
##           x           y  
## Min.      :0.9514   Min.      :1.606  
## 1st Qu.:1.1288   1st Qu.:2.865  
## Median :1.6813   Median :3.053  
## Mean     :2.2684   Mean      :3.326
```

```
## 3rd Qu.:3.5767    3rd Qu.:4.249
## Max.      :4.0038    Max.      :4.858
```

## 1.6 Task: Untidy Code?

```
simulate_phenotype <- function(pop_params, gp_map, gtype) {
  pop_mean <- pop_params[1]
  pop_var <- pop_params[2]
  pheno <- rnorm(n = N, mean = pop_mean, sd = sqrt(pop_var))
  effect <- rep(0, times = length(N))
  for (gt_iter in c('AA', 'AB', 'BB')) {
    effect[gtype == gt_iter] <- rnorm(n = sum(gtype == gt_iter),
                                       mean = gp_map[gt_iter, 'mean_eff'],
                                       sd = sqrt(gp_map[gt_iter, 'var_eff']))
  }
  dat <- data.frame(gt = gtype, raw_pheno = pheno, effect = effect, pheno = pheno + effect)
  return(dat)
}
```