

Soluciones del Segundo Laboratorio

Bioestadística 2020

Ejercicio 1

```
p = 1
for(i in 1:50){
  p = p * i
}
p
```

```
## [1] 3.041409e+64
```

```
prod(1:50) # comparar con p
```

```
## [1] 3.041409e+64
```

Ejercicio 2

```
suma_sen = 0
for(i in 0:100){
  suma_sen = suma_sen + sin(i)
}
prom_sen = suma_sen/101
prom_sen
```

```
## [1] -0.001259119
```

```
mean(sin(0:100)) #comparar con prom_sen
```

```
## [1] -0.001259119
```

Ejercicio 3

```
suma_sen_pos = 0 # guarda suma de sin(i) positivos
cant_sen_pos = 0 # guarda cantidad de sin(i) positivos
for(i in 1:100){
  if(sin(i) > 0){
    suma_sen_pos = suma_sen_pos + sin(i)
    cant_sen_pos = cant_sen_pos + 1
  }
}
prom_sen_pos = suma_sen_pos/cant_sen_pos
prom_sen_pos
```

```
## [1] 0.6385647
```

```
sen_pos = sin(1:100)[sin(1:100) > 0]
mean(sen_pos) #comparar con prom_sen_pos
```

```
## [1] 0.6385647
```

Ejercicio 4

```
v = 1:100
# Inicializo un vector cualquiera 'v'.
# Lo importante es que tenga largo 100.
for(i in 1:100){
```

```
v[i] = i^2 - i + 1
}
```

Ejercicio 5

```
v.norm = rnorm(10000, 3, 2)
# Usando un for:
suma_norm = 0
cant_norm = 0
for(i in v.norm){
  if((i > -1)&(i < 3)){
    suma_norm = suma_norm + i
    cant_norm = cant_norm + 1
  }
}
prom_norm = suma_norm/cant_norm
prom_norm
```

```
## [1] 1.545846
```

Ejercicio 6

```
mean(2*rnorm(10000,0,2)-1)
```

```
## [1] -1.000118
```

Ejercicio 7

```
r.norm=rnorm(10000,0,2)
mean(r.norm[r.norm>-1&r.norm<3])
```

```
## [1] 0.6793159
```

Ejercicio 8

```
result=1:100
for(i in 1:100){
  r_i=rnorm(10000,3,2)
  result[i]=mean(r_i[(r_i>-1)&(r_i<3)])
}
result
```

```
## [1] 1.564910 1.561033 1.551018 1.567196 1.557163 1.537412 1.564088
## [8] 1.535783 1.537739 1.555866 1.558995 1.557346 1.568821 1.568882
## [15] 1.532084 1.572477 1.559524 1.555188 1.559857 1.549469 1.523707
## [22] 1.563710 1.558215 1.554534 1.583201 1.560729 1.568250 1.558232
## [29] 1.565494 1.550641 1.561257 1.542025 1.558857 1.559048 1.540966
## [36] 1.544525 1.555716 1.567605 1.556982 1.563353 1.543033 1.572030
## [43] 1.560752 1.547160 1.540680 1.569800 1.547011 1.548294 1.554516
## [50] 1.581974 1.545341 1.577270 1.553856 1.554452 1.556219 1.564984
## [57] 1.573410 1.532001 1.566099 1.558187 1.528612 1.573458 1.548432
## [64] 1.544174 1.532911 1.574246 1.569234 1.572350 1.556022 1.581007
## [71] 1.546222 1.526726 1.564325 1.589598 1.553619 1.567564 1.558444
## [78] 1.569849 1.560412 1.557434 1.537352 1.552844 1.538236 1.566235
## [85] 1.569889 1.541379 1.555868 1.538908 1.565655 1.560552 1.534256
## [92] 1.559177 1.548406 1.538906 1.573628 1.564626 1.558227 1.539979
## [99] 1.569125 1.580482
```

Ejercicio 9

```
n=100
cant=0
r=rnorm(n,0,1)
for(i in 1:n){
  if ((r[i]<qnorm(0.975))&(r[i]>qnorm(0.025))){
    cant=cant+1}
}
cant/n
```

```
## [1] 0.99
```

Ejercicio 10

```
n=100
alfa=0.86
cant=0
r=rnorm(n,0,1)
for(i in 1:n){
  if ((r[i]<qnorm((alfa+1)/2))&(r[i]>qnorm((1-alfa)/2))){
    cant=cant+1}
}
cant/n
```

```
## [1] 0.84
```