Assignment 1 -Statistical Programming Language-

Student ID : 201724570

Name : 정석규

Q1.

Source code)

floor(log10(factorial(100)))+1

floor(log10(exp(lfactorial(200))))+1

floor(log10(exp(lfactorial(500))))+1

Answer) 158; Inf; Inf; (R can't find way to calculate big number factorial)

R console img)

```
##Q1
floor(log10(factorial(100)))+1
floor(log10(exp(lfactorial(200))))+1
floor(log10(exp(lfactorial(500))))+1

> ##Q1
> floor(log10(factorial(100)))+1
[1] 158
> floor(log10(exp(lfactorial(200))))+1
[1] Inf
> floor(log10(exp(lfactorial(500))))+1
[1] Inf
```

Q2.

Source code)

seq(1,21,2)

10^(seq(0,9))

rep(0:3,6)

rep(0:4, each = 3)

seq(50,11,-1)

rep(c(1,2,5),5)*(10^seq(0,4))

Answer)

seq(1,21,2)

10^(seq(0,9))

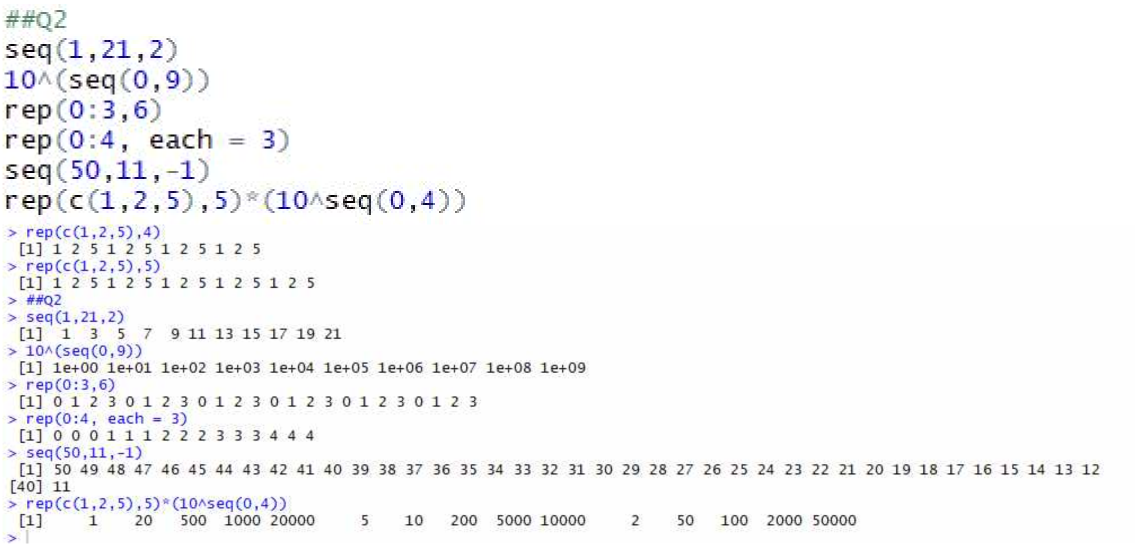
rep(0:3,6)

rep(0:4, each = 3)

seq(50,11,-1)

rep(c(1,2,5),5)*(10^seq(0,4))

R console img)



```
##Q2
seq(1,21,2)
10^(seq(0,9))
rep(0:3,6)
rep(0:4, each = 3)
seq(50,11,-1)
rep(c(1,2,5),5)*(10^seq(0,4))
> rep(c(1,2,5),4)
 [1] 1 2 5 1 2 5 1 2 5 1 2 5
> rep(c(1,2,5),5)
 [1] 1 2 5 1 2 5 1 2 5 1 2 5 1 2 5
> ##Q2
> seq(1,21,2)
 [1]  1  3  5  7  9 11 13 15 17 19 21
> 10^(seq(0,9))
 [1] 1e+00 1e+01 1e+02 1e+03 1e+04 1e+05 1e+06 1e+07 1e+08 1e+09
> rep(0:3,6)
 [1] 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3
> rep(0:4, each = 3)
 [1] 0 0 0 1 1 1 2 2 2 3 3 3 4 4 4
> seq(50,11,-1)
 [1] 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12
[40] 11
> rep(c(1,2,5),5)*(10^seq(0,4))
 [1]     1    20   500  1000 20000     5    10   200  5000 10000     2    50   100  2000 50000
> 
```

Q3.

Source code)

```
i = seq(1, 10)
S_10 = 4*sum((-1^(i+1))/(2*i-1))
i = seq(1, 100)
S_100 = 4*sum((-1^(i+1))/(2*i-1))
i = seq(1, 1000)
S_100 = 4*sum((-1^(i+1))/(2*i-1))

j = seq(1,100)
S_j = ((-1)^(j+1))/(2*j-1)
S_v = cumsum(S_j)
```

Answer) is in R console img

R console img)

```
##Q3
i = seq(1, 10)
s_10 = 4*sum((-1^(i+1))/(2*i-1))
i = seq(1, 100)
s_100 = 4*sum((-1^(i+1))/(2*i-1))
i = seq(1, 1000)
s_100 = 4*sum((-1^(i+1))/(2*i-1))

j = seq(1,100)
s_j = ((-1)^(j+1))/(2*j-1)
s_v = cumsum(s_j)
```

s_1	-13.1373687572065
s_10	-8.53302212063822
s_100	-17.7425306933404
s_j	num [1:100] 1 -0.333 0.2 -0.143 0.111 ...
s_v	num [1:100] 1 0.667 0.867 0.724 0.835 ...

Q4.

Source code)

x = seq(0.1, 4.9, by = 0.1)

x_1 = seq(0.2, 5, by = 0.1)

cal = (1/sqrt(2*pi))*exp(-x^2/2)*(x_1-x)

sum(cal)

x = seq(0.001, 99.999, by = 0.001)

x_1 = seq(0.002, 100, by = 0.001)

cal = (1/sqrt(x))*exp(-x)*(x_1-x)

sum(cal)

x = seq(0.01, 0,99, by = 0.01)

x_1 = seq(0.02, 1, by = 0.01)

cal = x^4*(1-x^4)*(x_1-x)

sum(cal)

Answer) 0.4800525 1.72628 9.942795e+21

R console img)

```
##Q4
x = seq(0.1, 4.9, by = 0.1)
x_1 = seq(0.2, 5, by = 0.1)
cal = (1/sqrt(2*pi))*exp(-x^2/2)*(x_1-x)
sum(cal)

x = seq(0.001, 99.999, by = 0.001)
x_1 = seq(0.002, 100, by = 0.001)
cal = (1/sqrt(x))*exp(-x)*(x_1-x)
sum(cal)

x = seq(0.01, 0,99, by = 0.01)
x_1 = seq(0.02, 1, by = 0.01)
cal = x^4*(1-x^4)*(x_1-x)
sum(cal)
```

```
> ##Q4
> x = seq(0.1, 4.9, by = 0.1)
> x_1 = seq(0.2, 5, by = 0.1)
> cal = (1/sqrt(2*pi))*exp(-x^2/2)*(x_1-x)
> sum(cal)
[1] 0.4800525
>
> x = seq(0.001, 99.999, by = 0.001)
> x_1 = seq(0.002, 100, by = 0.001)
> cal = (1/sqrt(x))*exp(-x)*(x_1-x)
> sum(cal)
[1] 1.72628
>
> x = seq(0.01, 0,99, by = 0.01)
Error in seq.default(0.01, 0, 99, by = 0.01) :
  너무 많은 인자들이 있습니다
> x_1 = seq(0.02, 1, by = 0.01)
> cal = x^4*(1-x^4)*(x_1-x)
Warning message:
In x_1 - x :
  longer object length is not a multiple of shorter object length
> sum(cal)
[1] 9.942795e+21
```

Q5.

Source code)

n = c(5,10,20,50,100)

n_m = data.frame(n)

fact <- factorial(n)

appx <- sqrt(2*pi*n)*(n/exp(1))^n

sim <- (appx/fact)*100

data.frame(n_m, fact = fact, appx = appx, sim = sim)

Answer) factorial and Stirling's approximation are similar each other about 99%(mean)

R console img)

```
##Q5
n = c(5,10,20,50,100)
n_m = data.frame(n)
fact <- factorial(n)
appx <- sqrt(2*pi*n)*(n/exp(1))^n
sim <- (appx/fact)*100
data.frame(n_m, fact = fact, appx = appx, sim = sim)
```

```
> ##Q5
> n = c(5,10,20,50,100)
> n_m = data.frame(n)
> fact <- factorial(n)
> appx <- sqrt(2*pi*n)*(n/exp(1))^n
> sim <- (appx/fact)*100
> data.frame(n_m, fact = fact, appx = appx, sim = sim)
    n          fact          appx      sim
1   5  1.200000e+02  1.180192e+02 98.34931
2  10  3.628800e+06  3.598696e+06 99.17040
3  20  2.432902e+18  2.422787e+18 99.58423
4  50  3.041409e+64  3.036345e+64 99.83347
5 100 9.332622e+157 9.324848e+157 99.91670
```

Q6.

Source code)

year = 1919

q = 1; m = 3; K = year%%100 ; J = floor(year%%100)

h = (q + floor(13*(m+1)/5)+K+floor(K/4)+floor(J/4)-2*J)

h

Answer) 0 -> ‘Saturday’

R console img)

```
##Q6
year = 1919
q = 1; m = 3; K = year%%100 ; J = floor(year%%100)
h = (q + floor(13*(m+1)/5)+K+floor(K/4)+floor(J/4)-2*J)
h
```

```
> ##Q6
> year = 1919
> q = 1; m = 3; K = year%%100 ; J = floor(year%%100)
> h = (q + floor(13*(m+1)/5)+K+floor(K/4)+floor(J/4)-2*J)
> h
[1] 0
```