R입문 기말고사

장윤서(201922034)

2021/ 12/ 21

```
library(tidyverse)
```

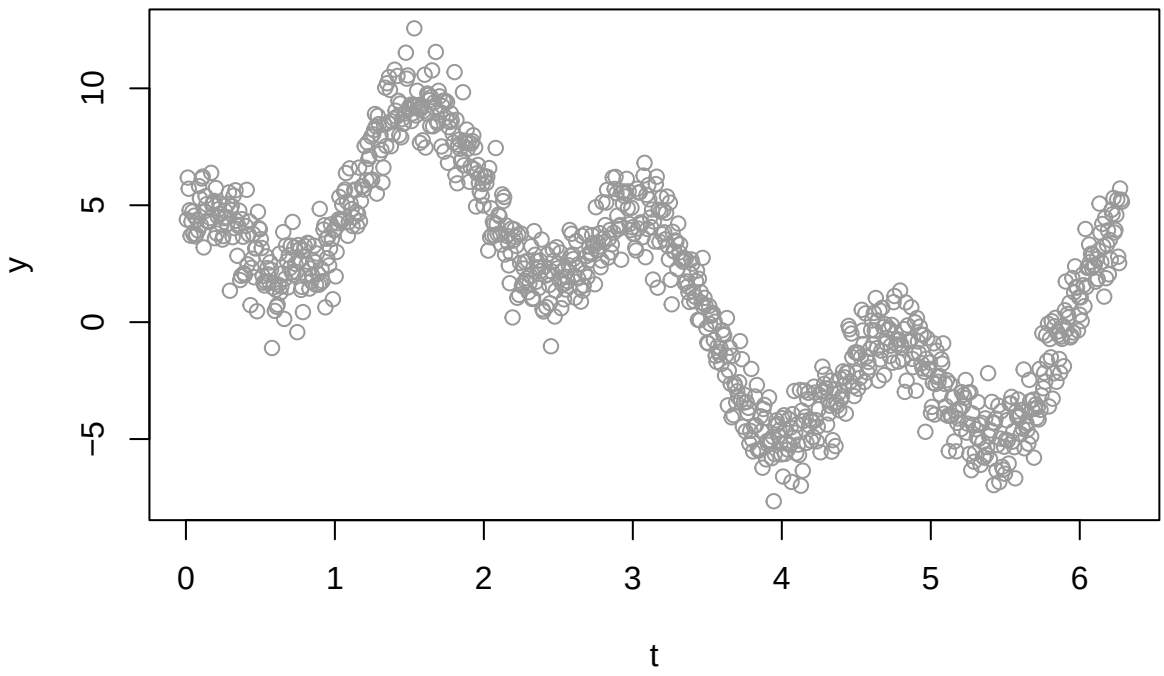
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

1번 문제

```
#1.1
epsilon <- rnorm(1000)
#1.2
a <- seq(1:1000)
t <- a*pi*2*(1/1000)
x1 <- sin(t)
x2 <- cos(4*t)
#1.3
y <-c()
for(i in 1:1000){
  y[i] <- 1.5 + 5*x1[i] + 3*x2[i] + epsilon[i]
}
plot(t,y,col = 'gray60')
```



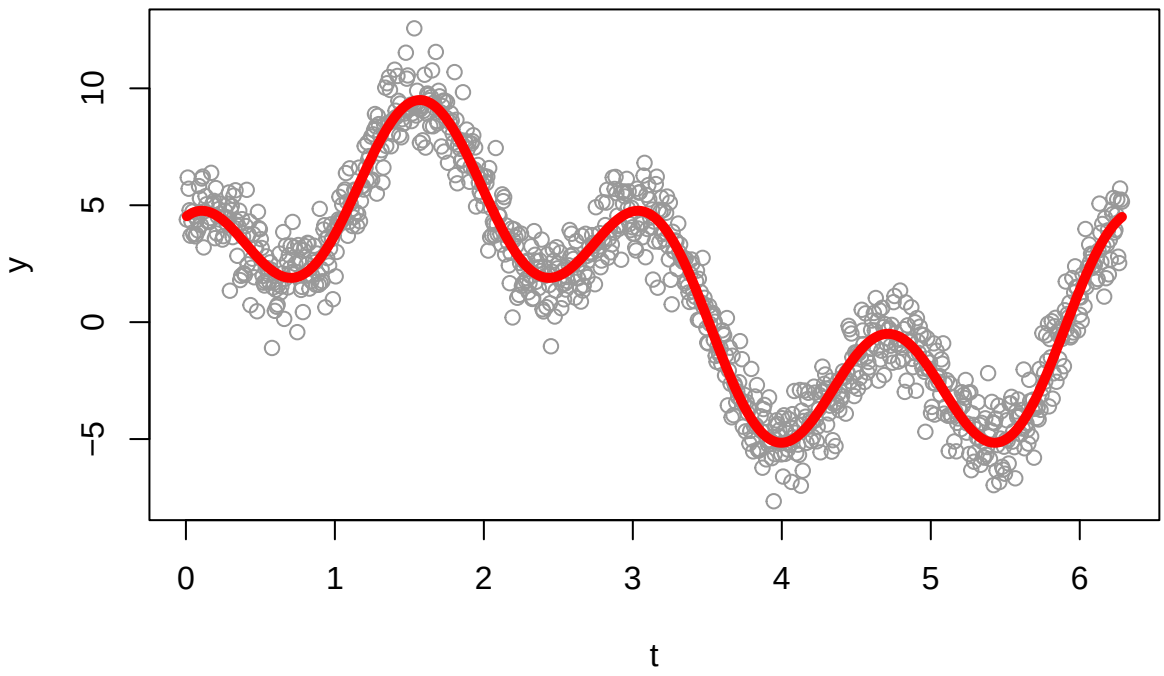

```
#1.4
a1 <- rep(1,1000)
X <- cbind(a1,x1,x2)
head(X)
```

```
##      a1          x1        x2
## [1,]  1 0.006283144 0.9996842
## [2,]  1 0.012566040 0.9987370
## [3,]  1 0.018848440 0.9971589
## [4,]  1 0.025130095 0.9949510
## [5,]  1 0.031410759 0.9921147
## [6,]  1 0.037690183 0.9886517
```

```
B <- rbind(1.5 , 5, 3)
B
```

```
##      [,1]
## [1,]  1.5
## [2,]  5.0
## [3,]  3.0
```

```
#1.5
XB <- X%*%B
plot(t,y,col = 'gray60')
lines(t,XB,col = 'red', lwd = '5')
```




```
#1.6
head(X)
```

```
##      a1          x1        x2
## [1,]  1 0.006283144 0.9996842
## [2,]  1 0.012566040 0.9987370
## [3,]  1 0.018848440 0.9971589
## [4,]  1 0.025130095 0.9949510
## [5,]  1 0.031410759 0.9921147
## [6,]  1 0.037690183 0.9886517
```

```
head(y)
```

```
## [1] 4.388536 6.183768 5.706285 4.788256 3.697580 4.297504
```

```
dim(y) <- c(1000,1)
head(y)
```

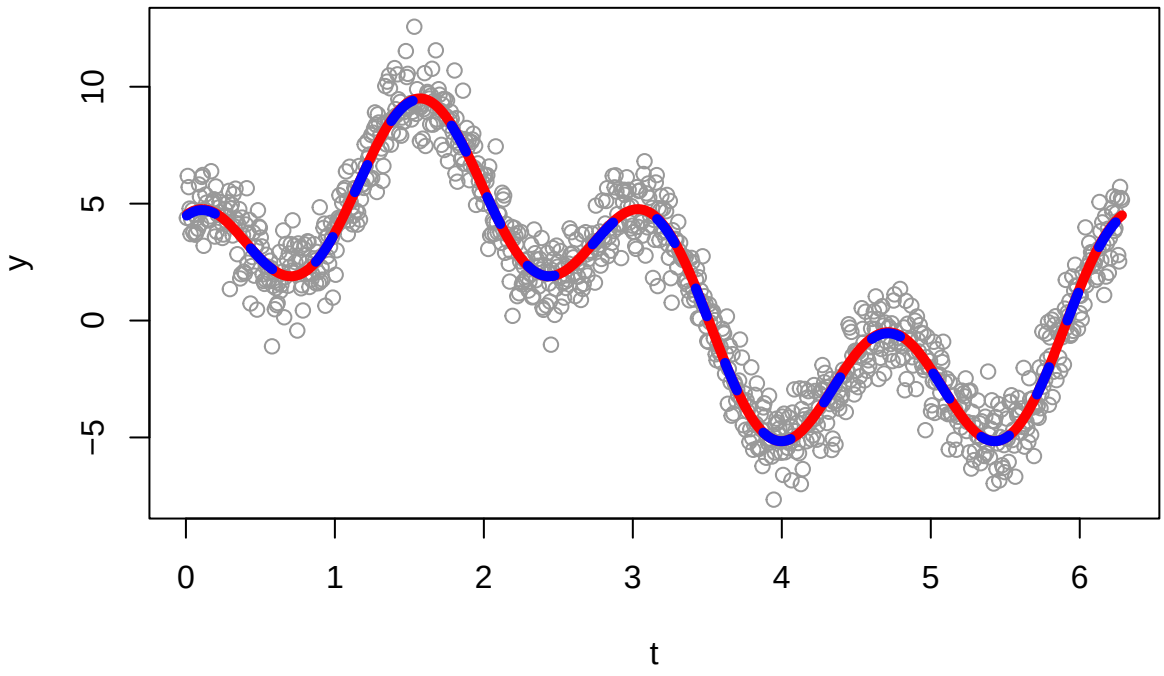
```
##          [,1]
## [1,] 4.388536
## [2,] 6.183768
## [3,] 5.706285
## [4,] 4.788256
## [5,] 3.697580
## [6,] 4.297504
```

```
B_ <- t(X)%*%X %>% solve()%*%t(X)%*%y
#1.7
```

```r
head(B_)
```

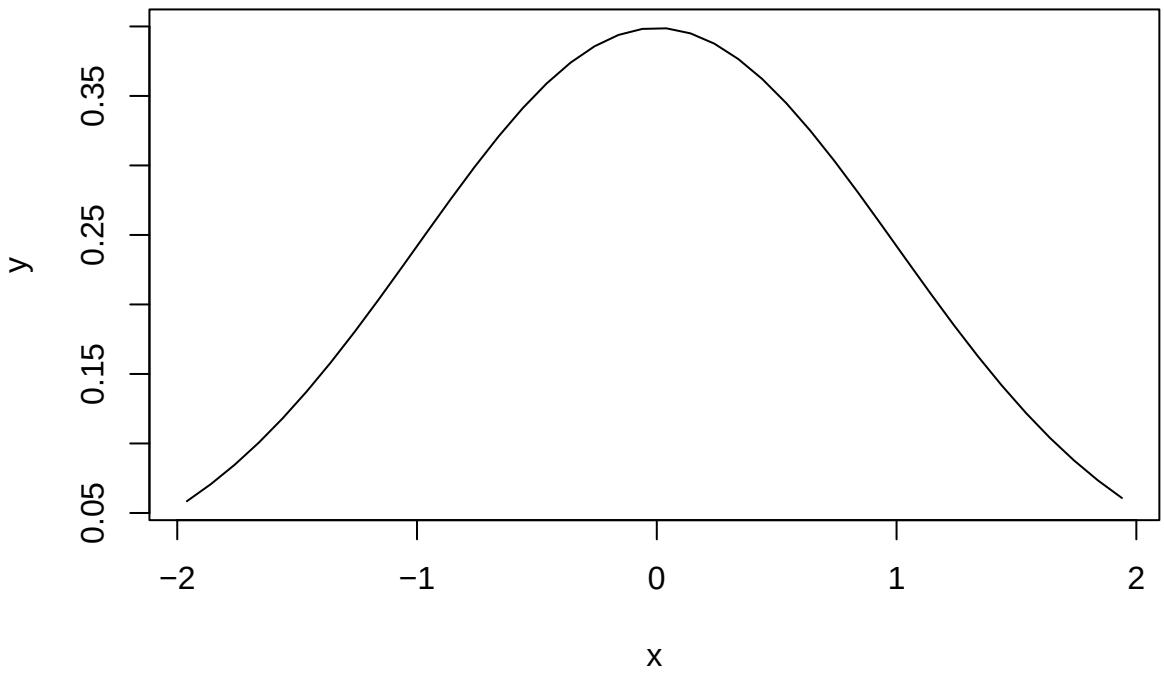
```
##        [,1]
## a1 1.479030
## x1 5.003676
## x2 2.978059
```

```r
XB_ <-X%*%B_
plot(t,y,col = 'gray60')
lines(t,XB,col = 'red', lwd = '5')
lines(t,XB_,col= 'blue', lty = 2, lwd = '5')
```



2번 문제

```r
x <- seq(from = -1.96 , to = 1.96, by = 0.1)
y <- (1/sqrt(2*pi))*exp(-0.5*x^2)
plot(x,y,type = 'l')
```




```
xx= runif(10000,-1.96,1.96)
yy= runif(10000)
plot(xx,yy,col='grey')
lines(x,y,col='red',lwd=3)
test <- function(xx,yy){
  yy < (1/sqrt(2*pi))*exp(-0.5*xx^2)
}
print(c(xx[1],yy[1]))
```

```
## [1] 0.5305611 0.5232577
```

```
print((1/sqrt(2*pi))*exp(-0.5*xx[1]^2))
```

```
## [1] 0.3465646
```

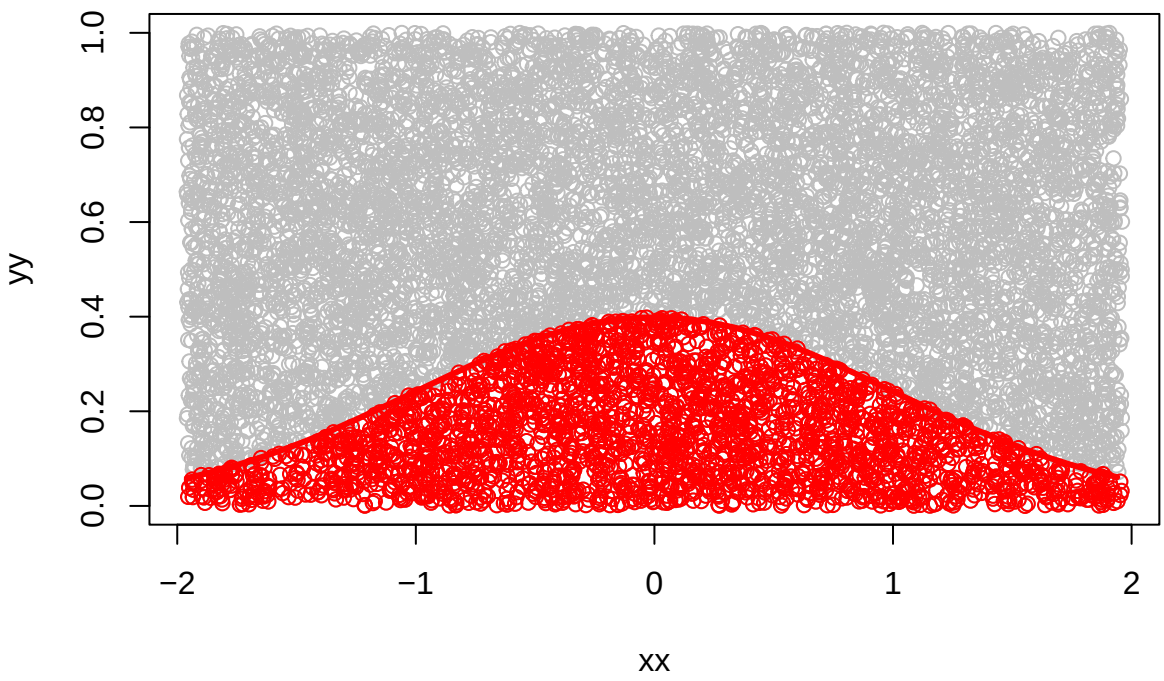
```
test(xx[1],yy[1])
```

```
## [1] FALSE
```

```
tst <- c()
for(i in 1:10000) tst[i] =test(xx[i],yy[i])
head(tst)
```

```
## [1] FALSE  TRUE  TRUE FALSE  TRUE  TRUE
```

```
points(xx[tst],yy[tst],col='red')
```




```
sum(tst)
```

```
## [1] 2431
```

```
sum(tst)/10000 *(1.96*2*1)
```

```
## [1] 0.952952
```

```
#2.2
a <- rnorm(1000)
head(a)
```

```
## [1]  1.1593225  0.5968120 -0.5317742  1.1741308 -0.6437992 -0.5053599
```

```
count = 0
for( i in 1:1000){
  if(abs(a[i]) < 1.96){
    count= count +1
  }
}
count
```

```
## [1] 954
```

3번 문제

```
###TYPE A
##N1 ~ N8 = 참가자 1 ~ 참가자 8, A =유리식별 가능한 참가자9 N9= 참가자 10
ARR = c('N1','N2','N3','N4','N5','N6','N7','N8','A','N9')
SURV = 10
PLAYER = ARR[SURV]
PLAYER
```

```
## [1] "N9"
```

```
STAGE=0
PROB = 0.5
TOSSRSLT = NA

toss= function(p) rbinom(n=1,size=1,prob=p) %>% as.logical()
reset = function(){
  TOSSRSLT <<- NA
  SURV <<- 10
  STAGE <<- 0
  PLAYER <<- ARR[SURV]
}
record <- function(){
  list(PRE_TOSSRSLT=TOSSRSLT,SURV=SURV,STAGE=STAGE,PLAYER=ARR[SURV])
}

go <- function(){
  PROB <<- 0.5 + (PLAYER=='A')*0.45
  TOSSRSLT <<- toss(PROB)
  if(TOSSRSLT==FALSE) SURV <<- SURV -1
  STAGE <<- STAGE + 1
  PLAYER <<- ARR[SURV]
}

gogo <- function(){
  for(i in 1:20){
    go()
    if(SURV==0) break
  }
}
simulate_once = function(){
  reset()
  gogo()
  return(record()$SURV)
}

simrslt = c()
for(i in 1:1000) simrslt[i] = simulate_once()
mean(simrslt)
```

```
## [1] 5.497
```

```
###TYPEB
ARR1= c('N9','A','N8','N7','N6','N5','N4','N3','N2','N1')
SURV=10
PLAYER=ARR1[SURV]
```

```
STAGE=0
PROB=0.5
TOSSRSLT=NA
go1 <- function(){
  PROB <<- 0.5 + (PLAYER=='A')*0.45
  TOSSRSLT <<- toss(PROB)
  if(TOSSRSLT==FALSE) SURV <<- SURV -1
  STAGE <<- STAGE +1
  PLAYER<<-ARR1[SURV]
}
gogo1 <- function(){
  for(i in 1:20){
    go1()
    if(SURV==0) break
  }
}
simulate_once1 <- function(){
  reset()
  gogo1()
  return(record()$SURV)
}
simrslt1 <- c()
for(i in 1:1000) simrslt1[i] <- simulate_once1()
mean(simrslt1)
```

```
## [1] 1.979
```

```
head(simrslt);head(simrslt1)
```

```
## [1] 9 6 1 9 9 0
```

```
## [1] 2 2 2 5 4 2
```

```
simrslt[simrslt>=8]
```

```
##   [1] 9 9 9 9 9 9 9 9 8 9 9 9 9 9 9 8 9 9 9 9 9 9 9 9 9 8 9 9 9 9 9 9 9 9 9
##  [38] 9 9 9 9 9 8 9 9 8 9 9 9 9 9 9 9 8 9 9 9 9 9 8 8 9 9 8 9 8 9 9 9 9 9 9 8 8
##  [75] 9 9 9 9 8 9 9 9 9 9 9 9 8 9 9 9 8 9 9 9 9 9 8 9 9 9 9 9 9 9 9 9 9 9 8 8
## [112] 9 9 8 9 9 9 9 9 9 9 9 9 9 9 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
## [149] 9 9 9 9 9 9 9 8 9 9 8 9 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 9 9 9
## [186] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 9 9 9
## [223] 8 9 9 8 8 9 9 9 9 9 9 9 8 8 9 9 9 9 9 9 9 9 9 8 9 9 9 9 9 8 9 9 9 9 9 9 9
## [260] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
## [297] 9 9 9 8 9 9 9 9 9 8 9 9 8 9 9 9 9 9 8 8 9 9 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9
## [334] 9 9 8 9 9 9 9 8 9 9 9 9 9 8 9 9 9 9 9 9 9 9 8 9 9 9 9 8 9 9 9 9 9 9 9 9 9
## [371] 9 9 8 9 9 9 9 9 9 9 9 9 9 8 9 9 9 9 9 9 9 9 9 9 8 9 9 9 9 9 9 9 9 9 9 9 8
## [408] 9 9 9 9 8 9 9 9 9 9 9 9 9 9 9 9 9 8 9 9 9 9 8 9 9 9 9 9 9 9 9 9 8 9 9 9 9 9 9 8
```

```
simrslt1[simrslt1>=3]
```

```
##   [1] 5 4 3 3 3 5 5 4 4 4 5 3 3 3 3 3 5 5 3 7 3 4 3 3 4 4 3 3 3 3 4 5 5 3 3 3 3
##  [38] 5 3 3 4 3 3 6 3 3 3 4 3 3 5 3 4 4 3 4 4 4 3 4 5 5 3 5 3 3 3 7 5 3 4 3 3 4
##  [75] 3 4 3 3 4 3 4 6 3 3 4 3 4 3 4 3 5 5 4 4 3 3 5 6 3 3 3 4 4 3 3 3 3 6 3 3 3
## [112] 3 5 3 6 4 4 3 3 5 3 5 4 3 4 3 5 3 3 3 5 5 3 3 3 4 4 3 3 4 3
```

```
count = 0
for(i in 1:1000){
```

```
  if(simrslt[i]>=8){
  count= count +1
  }
}
count1=0
for(i in 1:1000){
  if(simrslt1[i]>=3){
    count1 = count1 +1
  }
}
#count는 TYPE A 에서 일반인8이 1000번중에 살아남는 횟수,count1은 TYPE B에서 일반인8이 살아남는 횟수이다.
count/1000;count1/1000
```

```
## [1] 0.444
```

```
## [1] 0.141
```

```
#그러므로 TYPE A 가 살아남을 확률이 더 높다
```

4번 문제

```
#4.1
df=read_csv('https://raw.githubusercontent.com/guebin/2021IR/master/_notebooks/covid19.csv')
```

```
## Rows: 12294 Columns: 5
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): prov
## dbl (4): year, month, day, cases
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(df)
```

```
## # A tibble: 6 x 5
##    year month   day prov  cases
##   <dbl> <dbl> <dbl> <chr> <dbl>
## 1  2020     1    20 서울      0
## 2  2020     1    20 부산      0
## 3  2020     1    20 대구      0
## 4  2020     1    20 인천      1
## 5  2020     1    20 광주      0
## 6  2020     1    20 대전      0
```

```
df %>% filter(year == 2020) %>% mutate(sum1 = sum(cases))
```

```
## # A tibble: 6,246 x 6
##    year month   day prov  cases  sum1
##   <dbl> <dbl> <dbl> <chr> <dbl> <dbl>
## 1  2020     1    20 서울      0 60726
## 2  2020     1    20 부산      0 60726
## 3  2020     1    20 대구      0 60726
## 4  2020     1    20 인천      1 60726
## 5  2020     1    20 광주      0 60726
```

```
##  6  2020     1    20 대전       0 60726
##  7  2020     1    20 울산       0 60726
##  8  2020     1    20 세종       0 60726
##  9  2020     1    20 경기       0 60726
## 10  2020     1    20 강원       0 60726
## # ... with 6,236 more rows
```

```
df %>% filter(year == 2021) %>% mutate(sum2 = sum(cases))
```

```
## # A tibble: 6,048 x 6
##     year month   day prov  cases   sum2
##    <dbl> <dbl> <dbl> <chr> <dbl>  <dbl>
##  1  2021     1     1 서울    357 396886
##  2  2021     1     1 부산     57 396886
##  3  2021     1     1 대구     43 396886
##  4  2021     1     1 인천     65 396886
##  5  2021     1     1 광주     17 396886
##  6  2021     1     1 대전     11 396886
##  7  2021     1     1 울산     43 396886
##  8  2021     1     1 세종      1 396886
##  9  2021     1     1 경기    284 396886
## 10  2021     1     1 강원     30 396886
## # ... with 6,038 more rows
```

```
#4.2
df %>% group_by(prov) %>% filter(year==2020, month ==2, day <= 15) %>% summarise(prov_sum = sum(cases))
```

```
## # A tibble: 18 x 2
##    prov  prov_sum
##    <chr>    <dbl>
##  1 강원         0
##  2 검역         0
##  3 경기         9
##  4 경남         0
##  5 경북         0
##  6 광주         2
##  7 대구         0
##  8 대전         0
##  9 부산         0
## 10 서울         5
## 11 세종         0
## 12 울산         0
## 13 인천         0
## 14 전남         1
## 15 전북         0
## 16 제주         0
## 17 충남         0
## 18 충북         0
```

```
#4.3
df %>% group_by(prov) %>% filter(year==2020, month ==2, day > 15) %>% summarise(prov_sum = sum(cases))
```

```
## # A tibble: 18 x 2
##    prov  prov_sum
##    <chr>    <dbl>
##  1 강원         7
```

```
##  2 검역          0
##  3 경기         65
##  4 경남         59
##  5 경북        472
##  6 광주          7
##  7 대구       2055
##  8 대전         13
##  9 부산         75
## 10 서울         62
## 11 세종          1
## 12 울산         17
## 13 인천          5
## 14 전남          1
## 15 전북          4
## 16 제주          2
## 17 충남         48
## 18 충북         10
```

4.1;

2020확진자 총 합= 60726

2021 확진자 총합 = 396886

4.2; 경기 지역에서 가장 많이 확진자가 나옴

4.3; 대구지역에서 가장 많이 확진자가 나옴