r입문 기말고사

김혁일(201618610)

12/21/2021

1번 문제풀이

```
#1
epsilon = rnorm(1000)
head(epsilon)
```

```
## [1] -0.66140028  0.07076372 -0.41127294 -0.43232382  0.69640450 -1.42634181
```

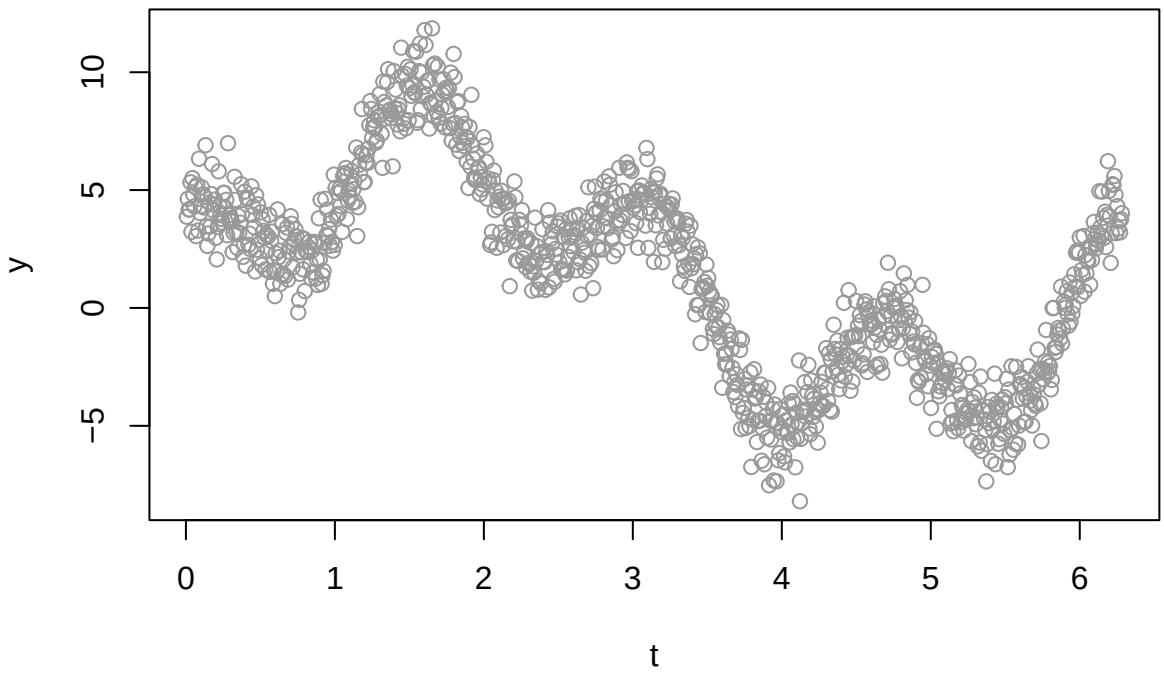
```
#2
i = (1:1000)
t <- (2*pi*i)/1000
X1 = sin(t)
X2 = cos(4*t)
head(X1)
```

```
## [1] 0.006283144 0.012566040 0.018848440 0.025130095 0.031410759 0.037690183
```

```
head(X2)
```

```
## [1] 0.9996842 0.9987370 0.9971589 0.9949510 0.9921147 0.9886517
```

```
#3
y <- 1.5+(5*X1)+(3*X2)+epsilon
plot(t,y,col='gray60')
```



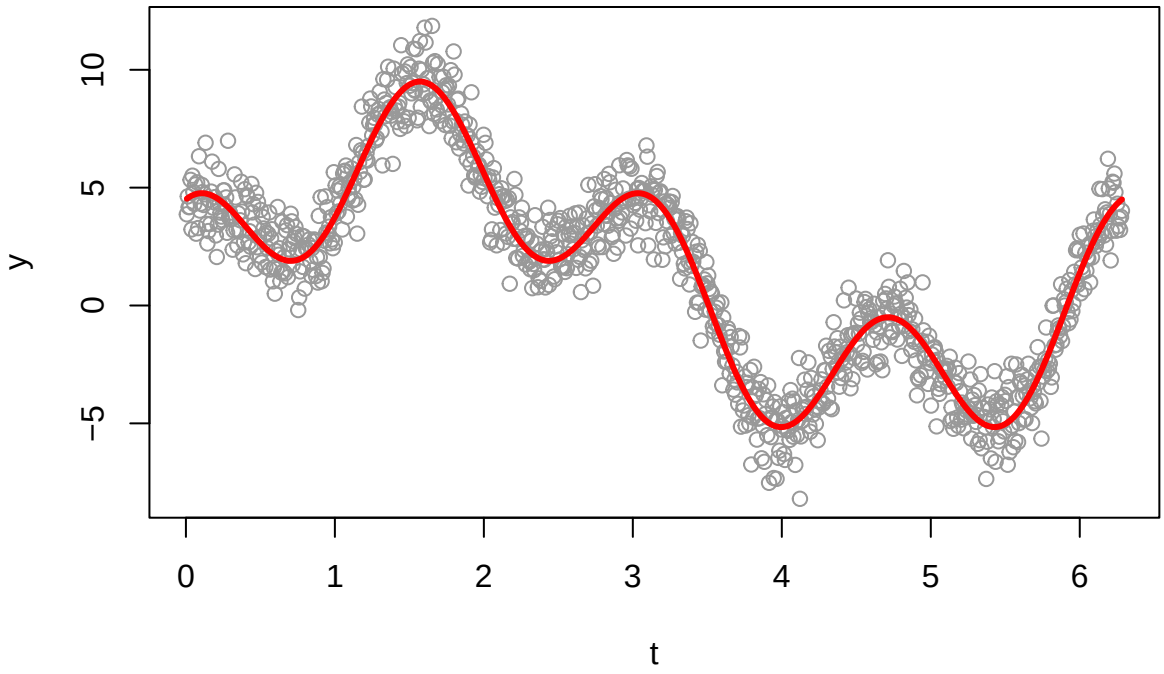

```
#4
a_ <- c()
for (i in 1:1000){
  a_[i]<- 1
}
X = cbind(a_,X1,X2)
head(X)
```

```
##      a_          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
```

```
#5
Beta = rbind(1.5,5,3)
XB=X %*% Beta
XBv = as.vector(XB)
head(XBv)
```

```
## [1] 4.530468 4.559041 4.585719 4.610504 4.633398 4.654406
```

```
plot(t,y,col='gray60')
lines(t,XBv,col='red',lwd='3')
```




```
#6
Bh = solve(t(X) %*% X) %*% t(X) %*% y
Bh
```

```
##         [,1]
## a_ 1.523286
## X1 5.029146
## X2 2.921906
```

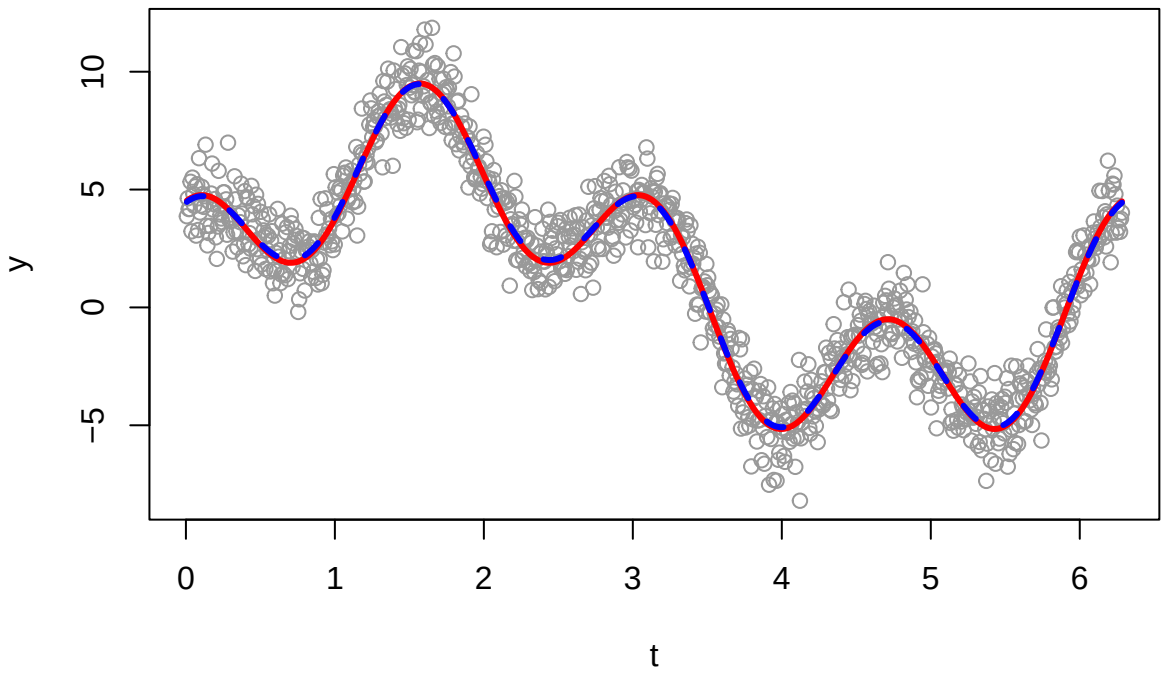
```
Beta
```

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

```
#7
XBh = X %*% Bh
XBhv = as.vector(XBh)
head(XBhv)
```

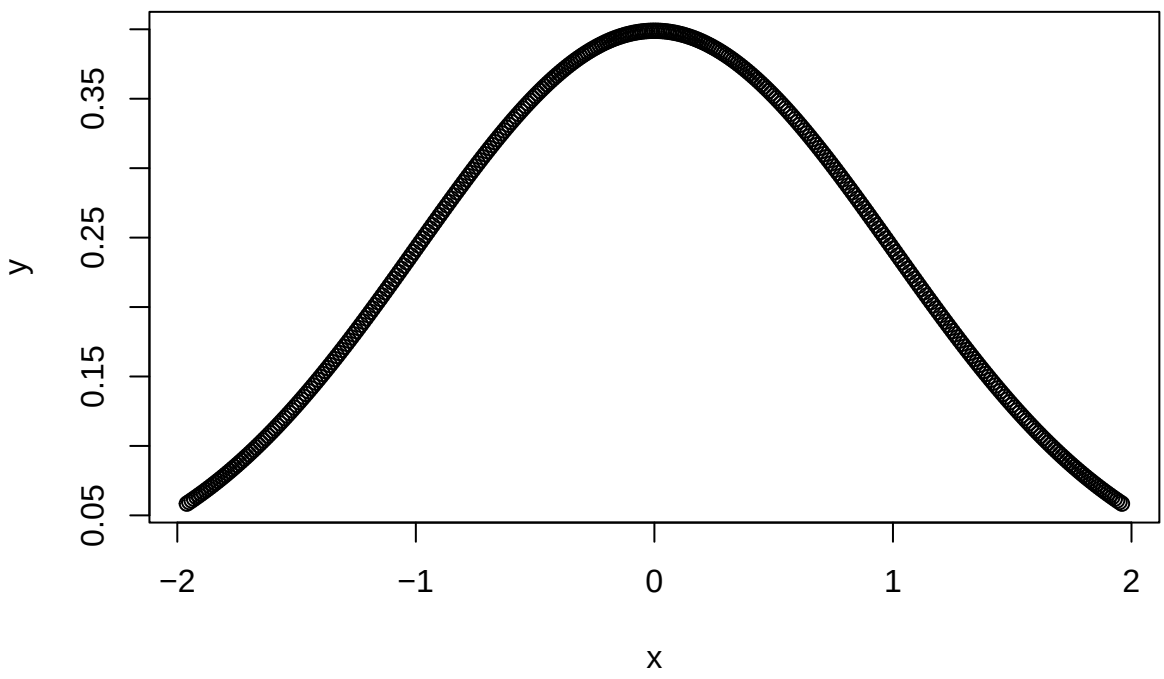
```
## [1] 4.475868 4.504698 4.531682 4.556822 4.580121 4.601583
```

```
plot(t,y,col='gray60')
lines(t,XBv,col='red',lwd='3')
lines(t,XBhv,lty=2,col='blue',lwd= '3')
```

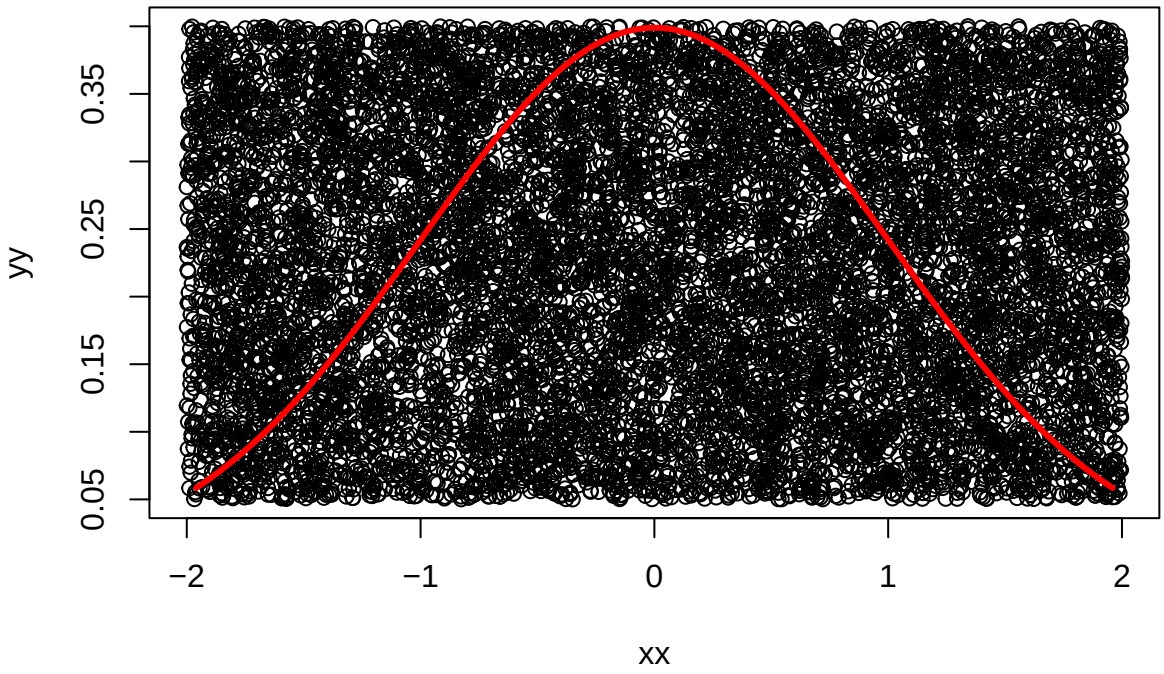


2번 문제풀이

```
#1
x = seq(-1.96,1.96,0.01)
y = (1/sqrt(2*pi))*exp((-(x^2))/2)
plot(x,y)
```




```
xx = runif(n=10000,-2,2)
yy = runif(n=10000,0.05,0.4)
plot(xx,yy)
lines(x,y,col='red',lwd=3)
```




```
test = function(xx,yy){
    yy < (1/sqrt(2*pi))*exp((-(xx^2))/2)
}
print(c(xx[1],yy[1]))
```

```
## [1] 1.4743250 0.1885057
```

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

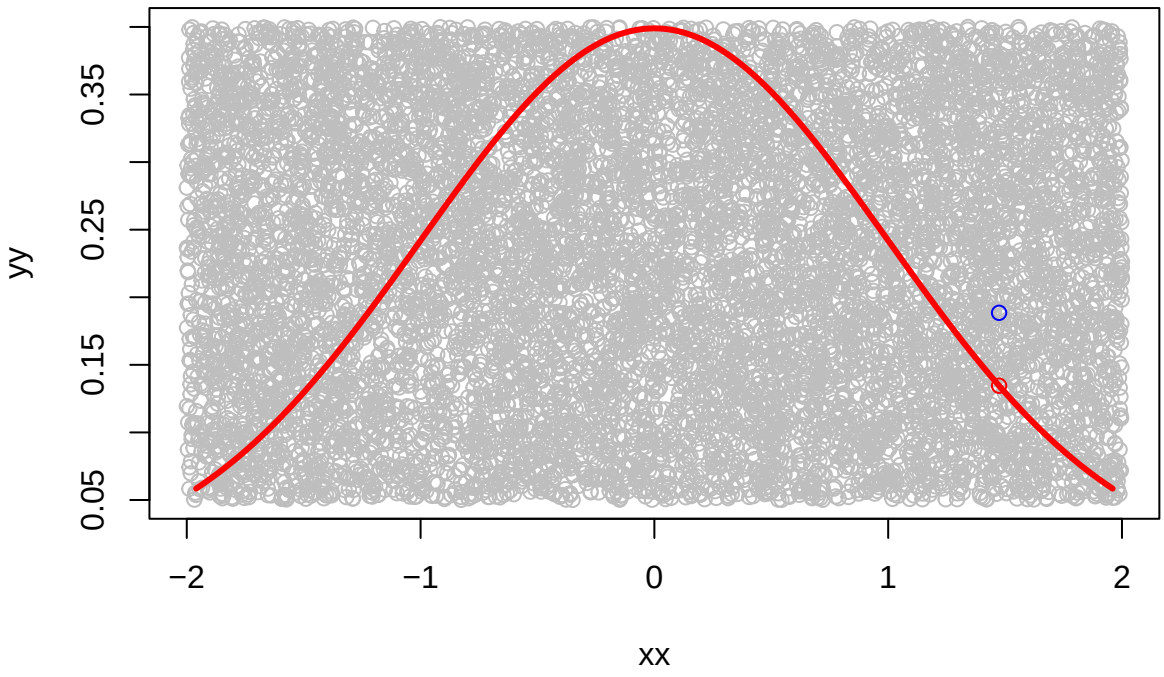
```
## [1] 0.3989423
```

```
## [1] 0.1345586
```

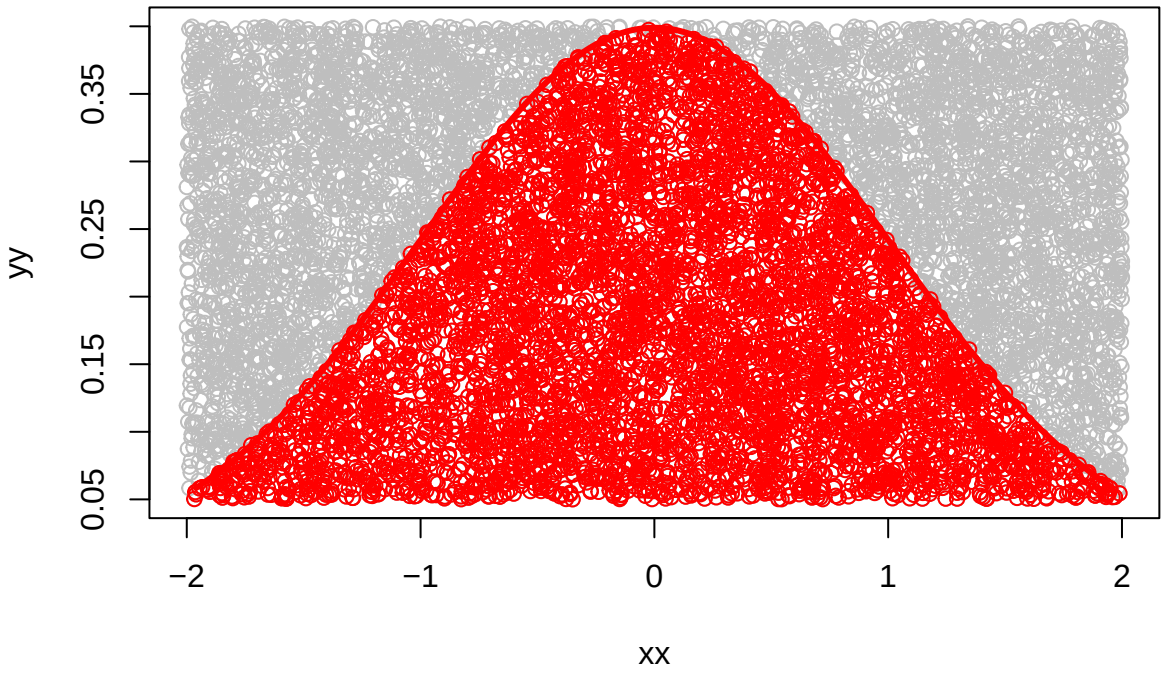
```
test(xx[1],yy[1])
```

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

```
plot(xx,yy,col='gray')
lines(x,y,col='red',lwd=3)
points(xx[1],yy[1],col='blue')
points(xx[1],(1/sqrt(2*pi))*exp((-(xx[1]^2))/2),col='red')
```




```
tst = c()
for (i in 1:10000) tst[i] = test(xx[i],yy[i])
plot(xx,yy,col='gray')
lines(x,y,col='red',lwd=3)
points(xx[tst],yy[tst],col='red')
```




```
sum(tst)
```

```
## [1] 5392
```

```
#10000개 중에서 5406개가 붉은점 따라서 5406/10000 * 1.4
#(-2에서 2까지 4, 0.05에서0.4까지 0.35, 4*0.35=1.4)

5406/10000*1.4 #약 0.75가 나온다
```

```
## [1] 0.75684
```

```
#2
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

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

```
x_ <- rnorm(1000)
```

```
dim(x_) = c(1000,1)
x_ = as.tibble(x_)
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.
## Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
x__ = x_ %>% filter(1.96 > V1 & -1.96 < x_)
dim(x__) #남은 행의 개수가 구간사이에 있는 확률변수의 개수이다.
```

```
## [1] 951   1
```

3번 문제풀이

```
# Type A
#10번의 성공 확률을 제외해도 9번이 19번 건너면 9번 8번 모두 살수있다.
#9번이 19번 성공하는 것 말고도 10번의 성공확률 8번의 성공확률이 있지만
#그것을 제외하고 계산해도 Type B의 확률보다 높다.
A = 0.95^19 #9번 혼자 건너가는 확률
x_ = rbinom(19,size=1,0.95)
#cumprod(x_)
x=sum(cumprod(x_))
xx_ = 19-x-1
y = rbinom(1, size=xx_, 0.5)
```

```
## Warning in rbinom(1, size = xx_, 0.5): NA가 생성되었습니다
```

```
rslt <- c()
for (i in 1:10000){
    x=sum(cumprod(rbinom(19,size=1,0.95)))
    xx_=19-x-1
    if (xx_>0) y = rbinom(1, size=xx_, 0.5) else y=0
    rslt[i]=x+y-10
}
mean(rslt)
```

```
## [1] 5.1246
```

```
# 평균 5명이 살아남는다

#Type B
#8번이 살기 위해서는 일반인 8명이서 끝까지 건너야한다
B=0.5^20
mean(rbinom(100000,20,0.5))
```

```
## [1] 9.99463
```

```
#하지만 일반인이 20개의다리를 건너기 위해서는 평균 10명이 필요하다

A>B# True가 나옴으로 9번혼자 건널 때 성공 확률이 더크다고 볼수있다
```

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

4번 문제풀이

```
df=read_csv('https://raw.githubusercontent.com/guebin/2021IR/master/_notebooks/covid19.c
```

```
## 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
```

```
#1
sum(df %>% filter(year==2020) %>% select(cases))#2020 cases
```

```
## [1] 60726
```

```
sum(df %>% filter(year==2021) %>% select(cases))#2021 cases
```

```
## [1] 396886
```

```
#2
df1 = df %>% filter(year==2020 & month==2 & day<=15)
df1%>% group_by(prov) %>% summarise(sum = sum(cases))
```

```
## # A tibble: 18 x 2
##    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
```

```
#경기가 9명으로 제일 많다.
```

```
#3
df2 = df %>% filter(year==2020 & month==2 & day>=16)
df2 %>% group_by(prov) %>% summarise(sum = sum(cases))
```

```
## # A tibble: 18 x 2
##    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
```

```
#대구가 2055명으로 제일 많다.
```