

# R 입문 기말고사

김민지

202020772

```
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

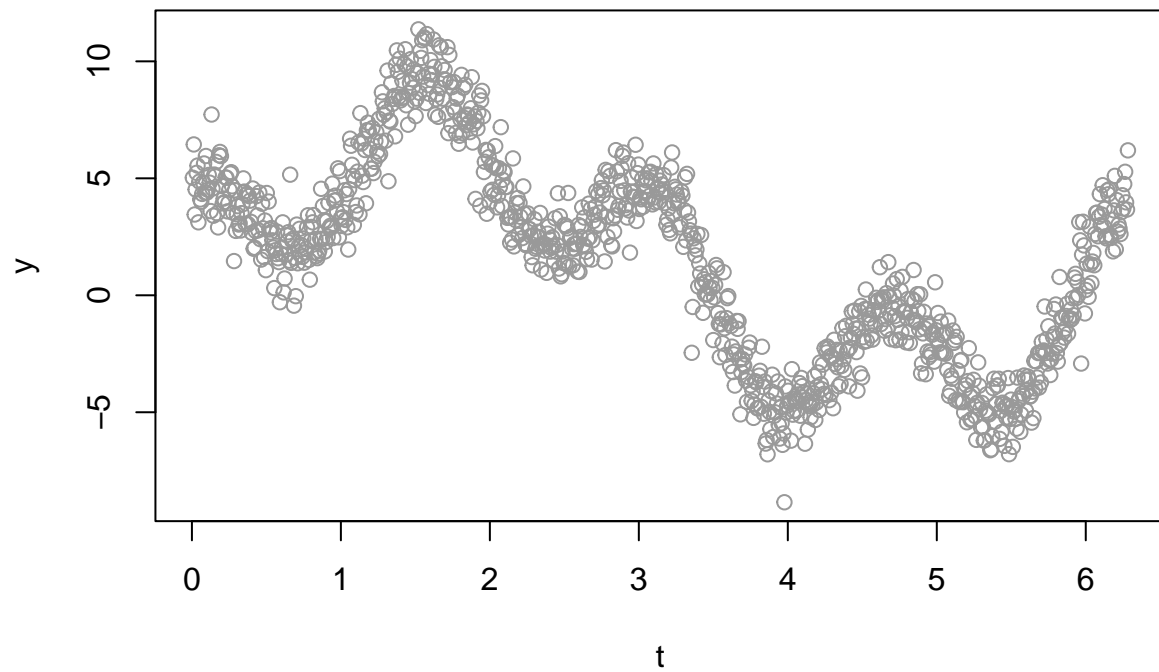
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(readr)
```

1번

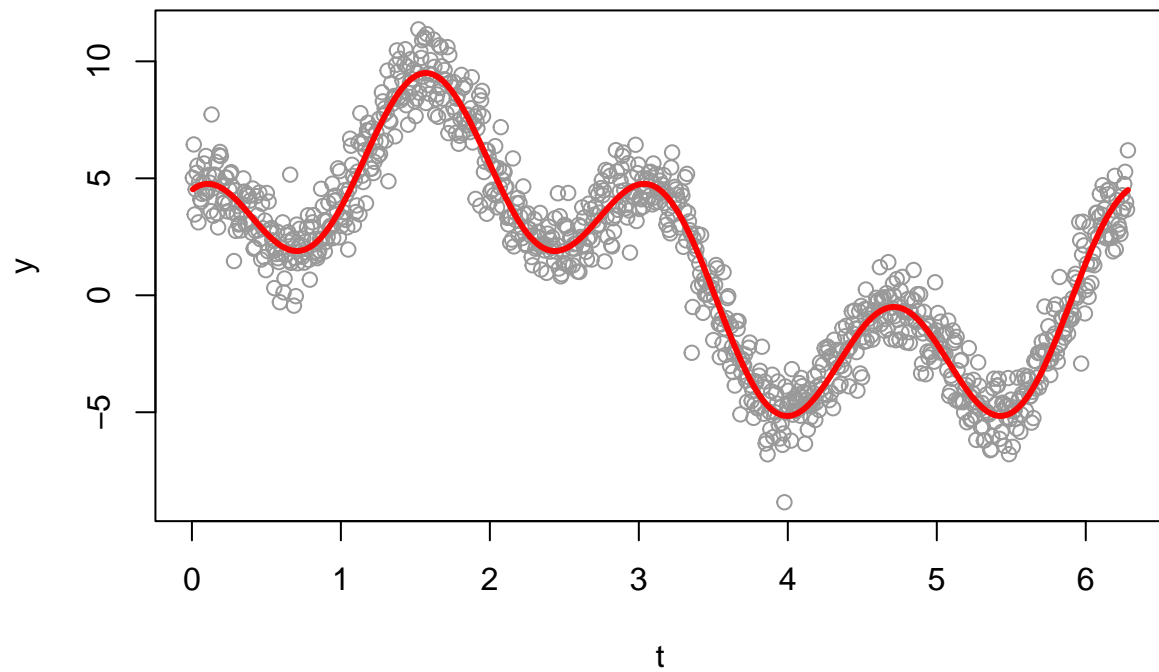
```
e=rnorm(1000)
i=1:1000
t= 2*pi/1000* i
x1 = sin(t)
x2=cos(4*t)
y=1.5+5*x1+3*x2+e

plot(t,y,col='gray60')
```



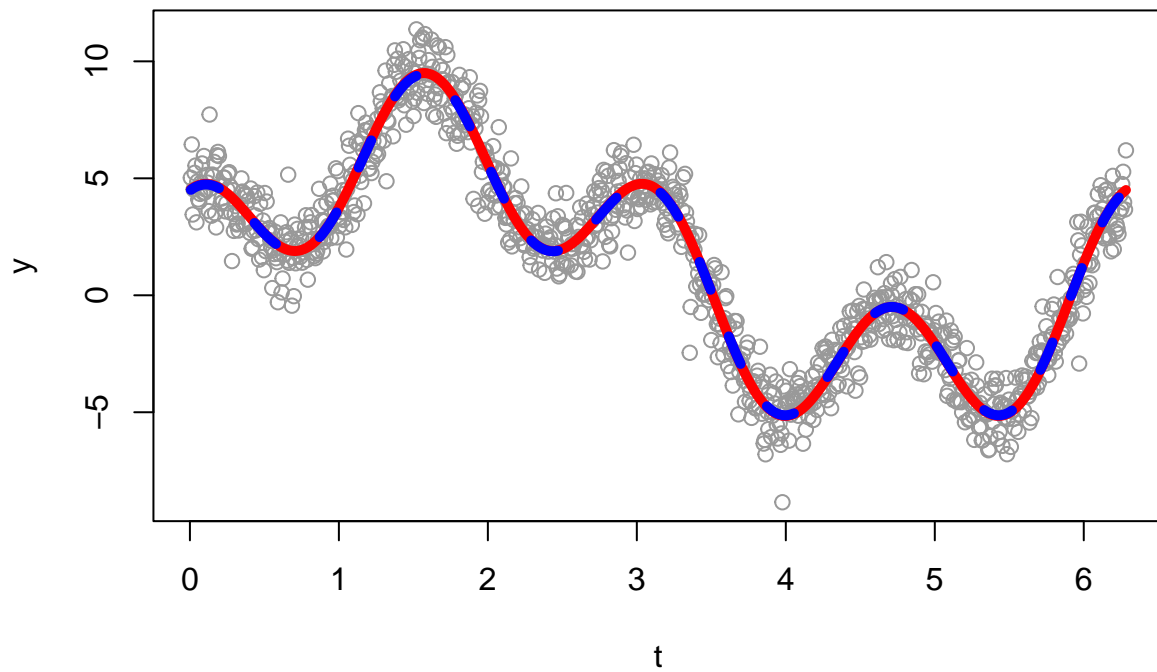
```
a=cbind(1,x1,x2)
b=rbind(1.5,5,3)
x4=a%*%b
x4=as.vector(x4)
```

```
plot(t,y,col='gray60')
lines(t,x4,col='red',lwd=3)
```



```
x5=solve(t(x4) %*% x4 )%*% t(x4) %*% y
x6= x4 %*% x5
```

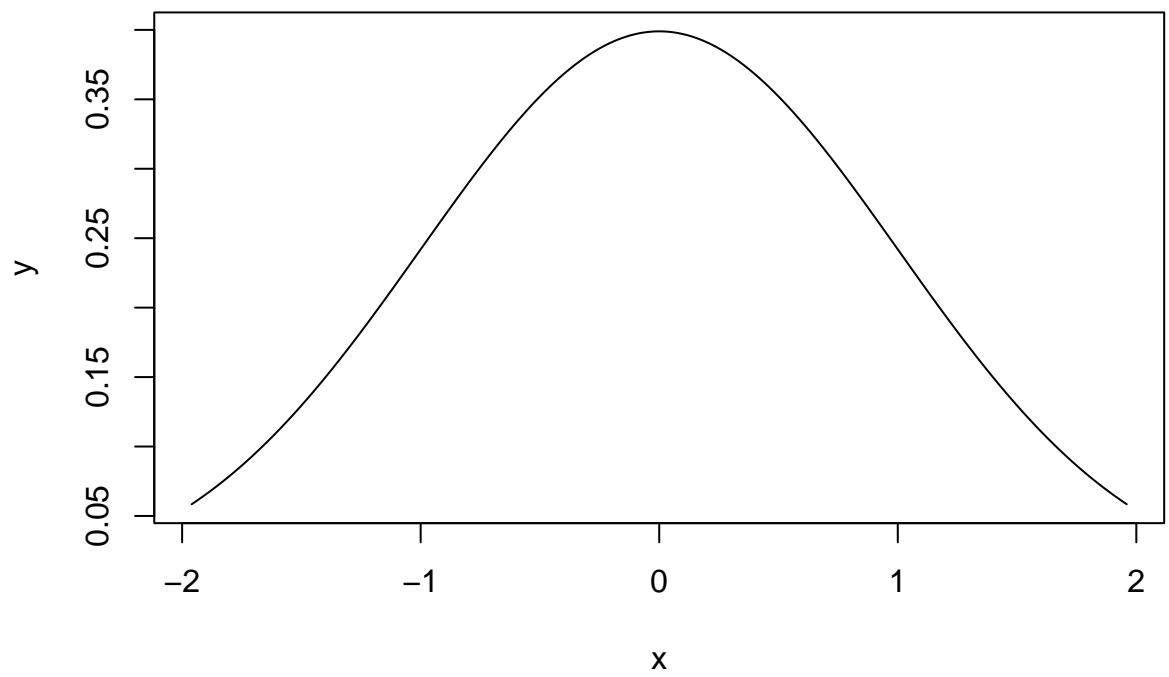
```
plot(t,y,col='gray60')
lines(t,x4,col='red',lwd=5)
lines(t,x6,col='blue',lty=2, lwd=5)
```



2번

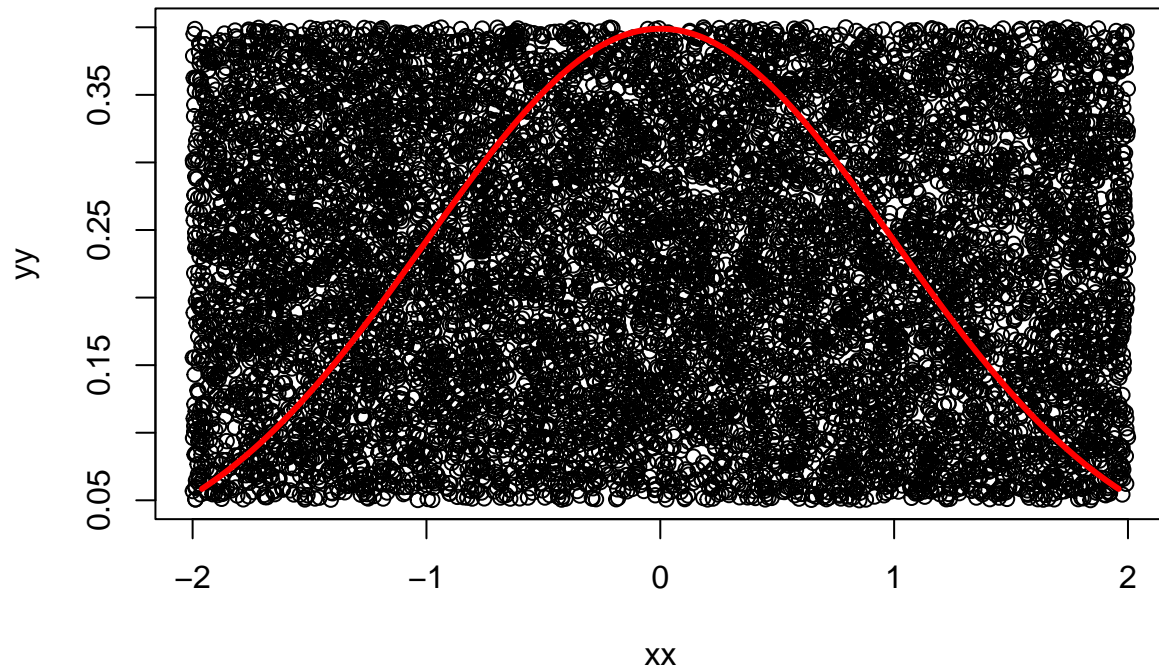
```
x=seq(from=-1.96, to=1.96, by=0.01)  
y=1/sqrt(2*pi)*exp((-1/2)*x^2)
```

```
plot(x,y,type='l')
```



(1)

```
xx=runif(10000)*4-2  
yy=runif(10000)*0.35+0.05  
plot(xx,yy)  
lines(x,y,col='red',lwd=3)
```



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

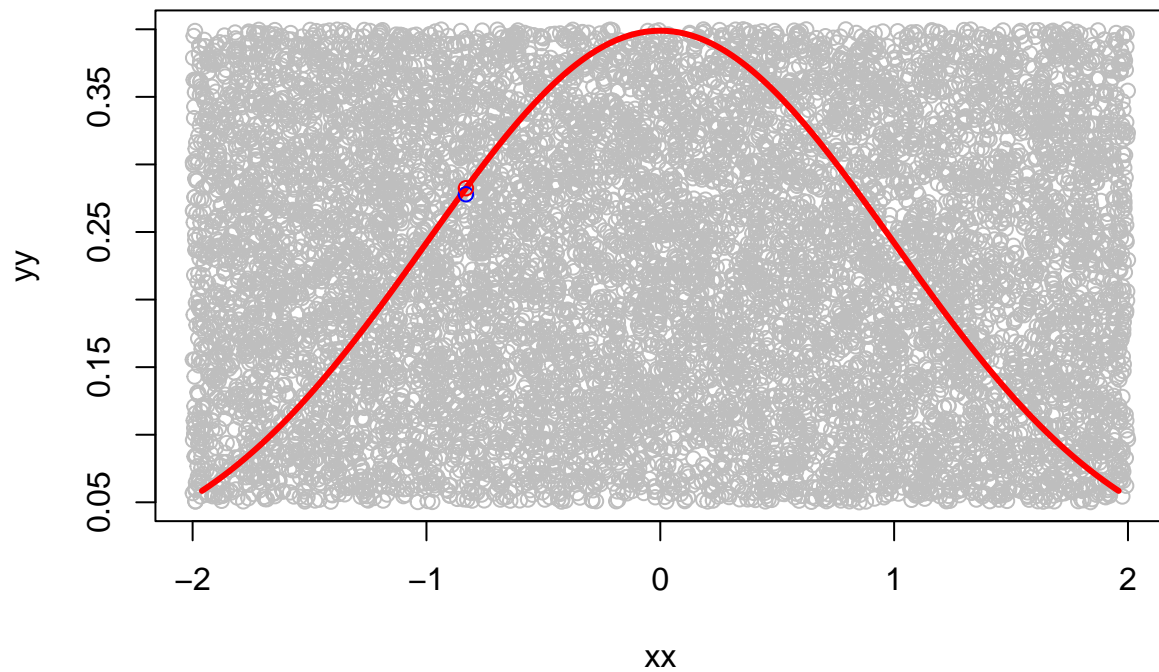
print(c(xx[1],yy[1]))

## [1] -0.8312705  0.2779126
print(1/sqrt(2*pi)*exp((-1/2)*xx[1]^2))

## [1] 0.2823963
test(xx[1],yy[1])

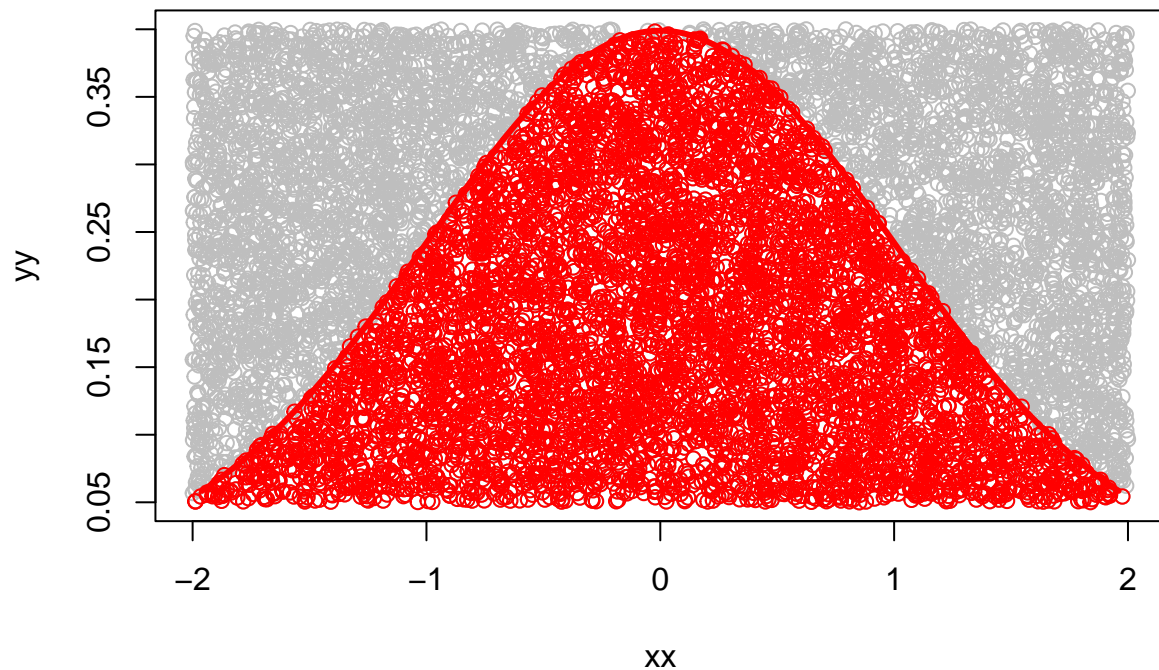
## [1] TRUE

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((-1/2)*xx[1]^2),col='red')
```



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

plot(xx,yy,col='grey')
lines(x,y, col='red',lwd=3)
points(xx[tst],yy[tst],col='red')
```



```
sum(tst)
```

```
## [1] 5381
```

```
5503/10000 *0.8
```

```
## [1] 0.44024
```

답은 0.44024이다

(2)

```
a=rnorm(1000)
```

```
sum((a>-1.96)*(a<1.96))
```

```
## [1] 961
```

3번

```
a=0 #아무도 안죽은 경우
```

```
a=a+(choose(20,0)*(5^0)*(0.5^(19))*0.95)
```

```
#10만 죽은 경우
```

```
for(i in 0:19)
```

```
{
```

```
a=a+((0.5^1)*(0.5^i)*(0.95^(19-i)))
```

```
}
```



```
#10과 9만 죽은 경우
a=a+(choose(20,2)*(0.5 ^1)*(0.5^(18))*(0.95^1))

print(a*100)
```

a타입

```
## [1] 39.86627
```

```
b=0 #b 타입
for(i in 0:7)
{
b=b+(choose(20,i)*(0.5^i)*(0.5^(20-i)))
}
print(b*100)
```

b타입

```
## [1] 13.1588
```

a타입이 살 확률이 더 높다

4번

```
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.

library(tidyverse)
a=as_tibble(df)
```

```
a %>% filter(year==2020) %>% summarise(sum_=sum(cases))
```

1번

```
## # A tibble: 1 x 1
##   sum_
##   <dbl>
## 1 60726
```

```
a %>% filter(year==2021) %>% summarise(sum_=sum(cases))
```

2020년의 확진자는 60726이다

```
## # A tibble: 1 x 1
##   sum_
##   <dbl>
## 1 396886
```

2021 확진자의 총합은 396886이다

2번

```
a %>% filter(year==2020,month==2) %>% filter(day>=1 & day<=15) %>% group_by(prov) %>% summarise(sum_cases=
```

```
## # A tibble: 18 x 2
##   prov sum_cases
##   <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
```

```
a %>% filter(year==2020,month==2) %>% filter(day>=15 & day<=29) %>% group_by(prov) %>% summarise(sum_ca
```

경기가 가장 많다.

```
## # A tibble: 18 x 2
##   prov sum_cases
##   <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 가장 많다.