R입문 기말고사

202016052 김하늘

2021 12 21

```
library("tidyverse")
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     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()
```

1번. 회귀분석

(1)

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

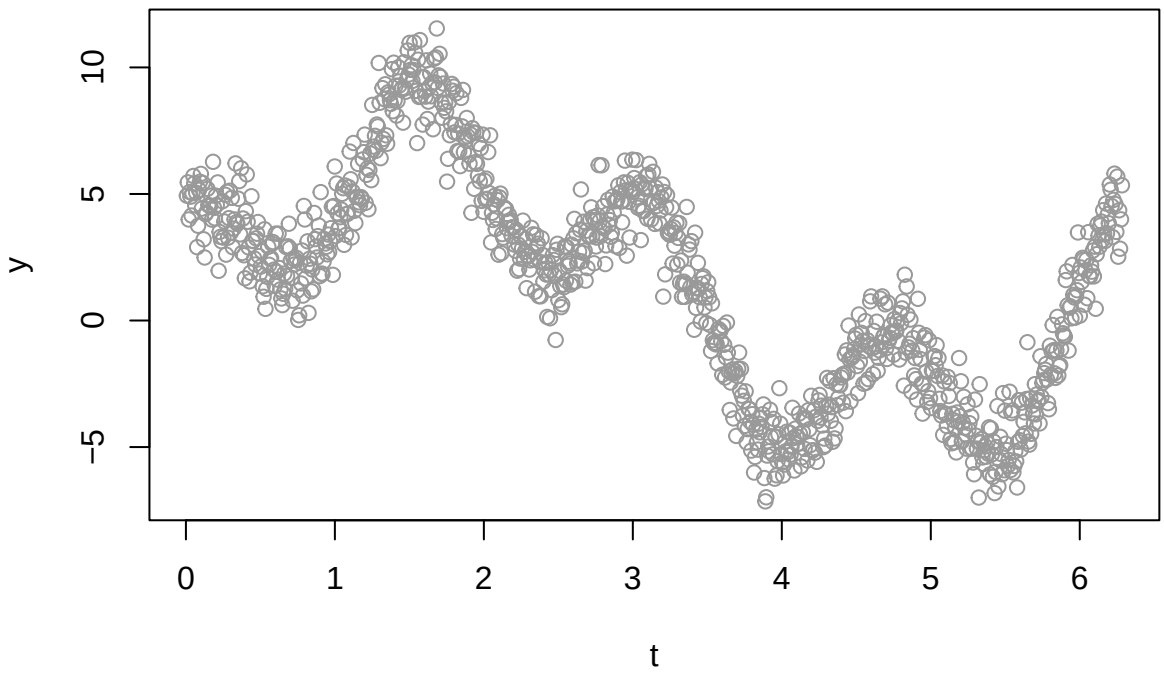
```
## [1]  0.4036144  0.9000285 -0.5927784  0.4666694  0.2584624 -0.5057589
```

(2)

```
x1 = c()
x2 = c()

i= 1:1000

t = 2*pi*i/1000

x1[i] = sin(t[i])
x2[i] = cos(4*t[i])
```

(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')
```



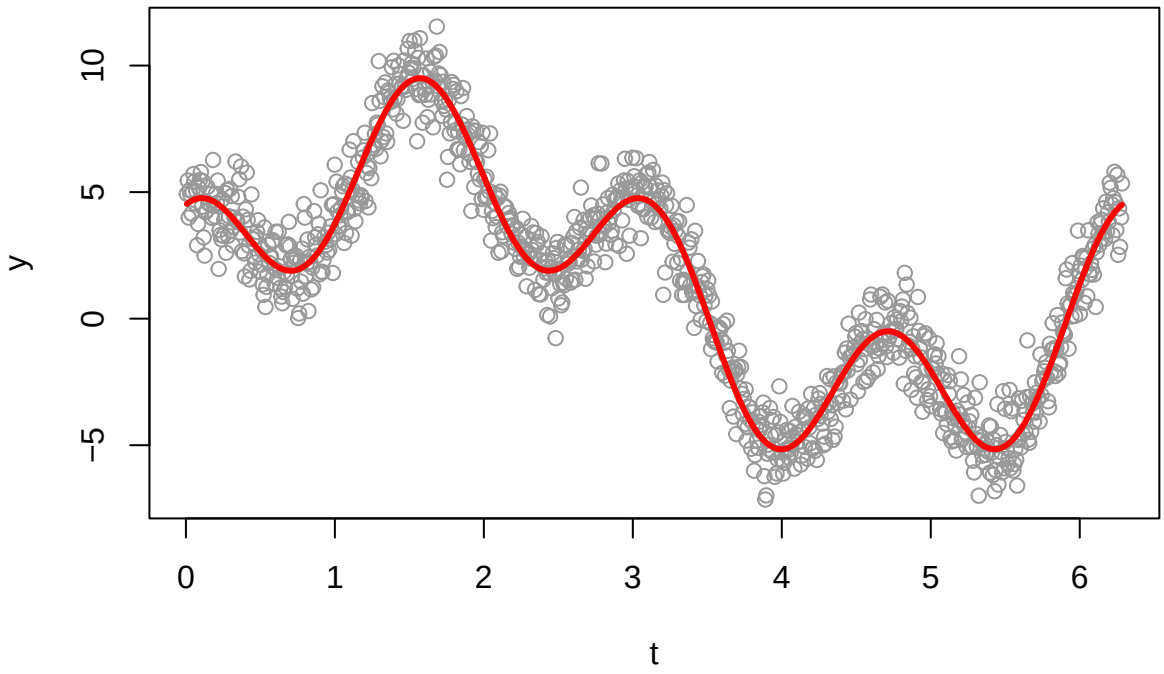
(4)

```
X = cbind(1, x1, x2)
dim(X) = c(1000, 3)

head(X)
```

```
##      [,1]        [,2]      [,3]
## [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
plot(t,y, col='gray60')
lines(t,XB, col='red', lwd=3)
```



(6)

```
dim(y) = c(1000,1)

X_t = t(X)
XtX_in =solve(X_t%*%X)

B_hat = XtX_in %*% X_t %*% y

B_hat
```

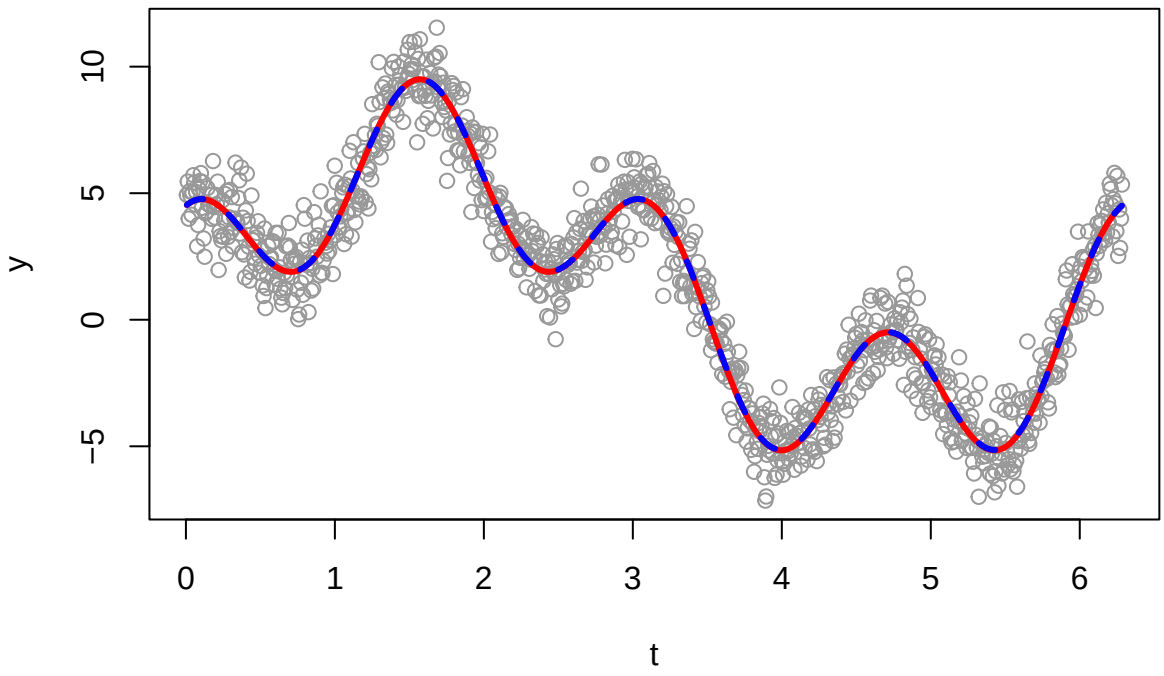
```
##          [,1]
## [1,] 1.509850
## [2,] 4.995290
## [3,] 2.997182
```

```
beta
```

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

(7)

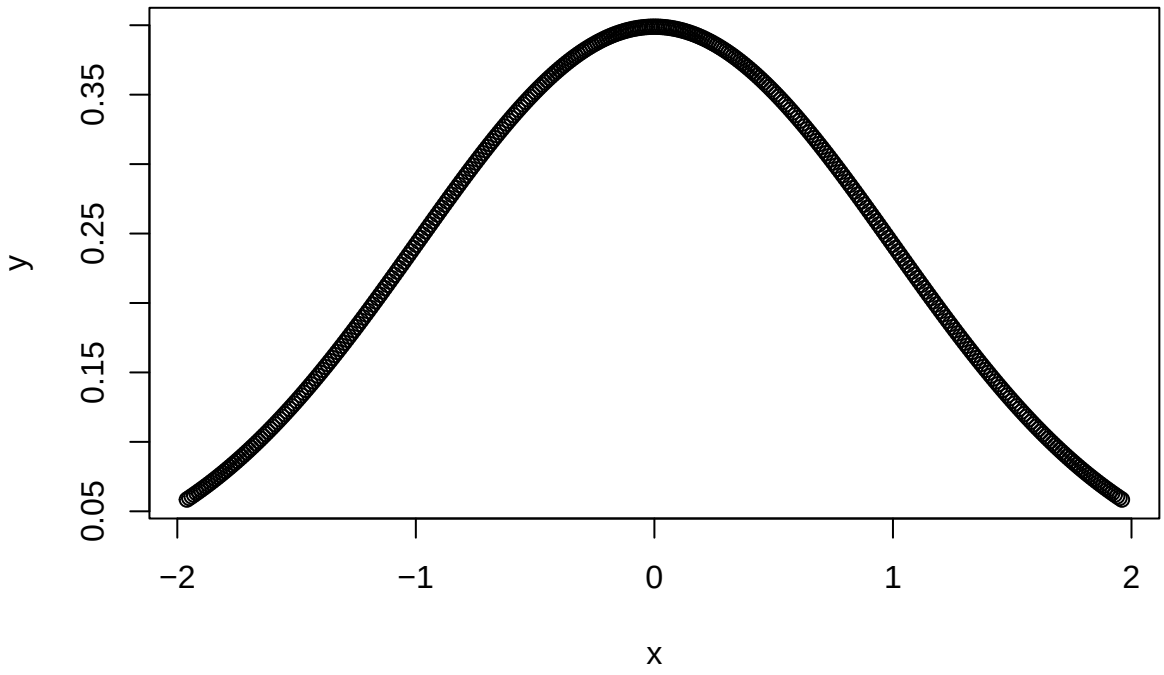
```
XB_hat = c()
XB_hat <- X %*% B_hat

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



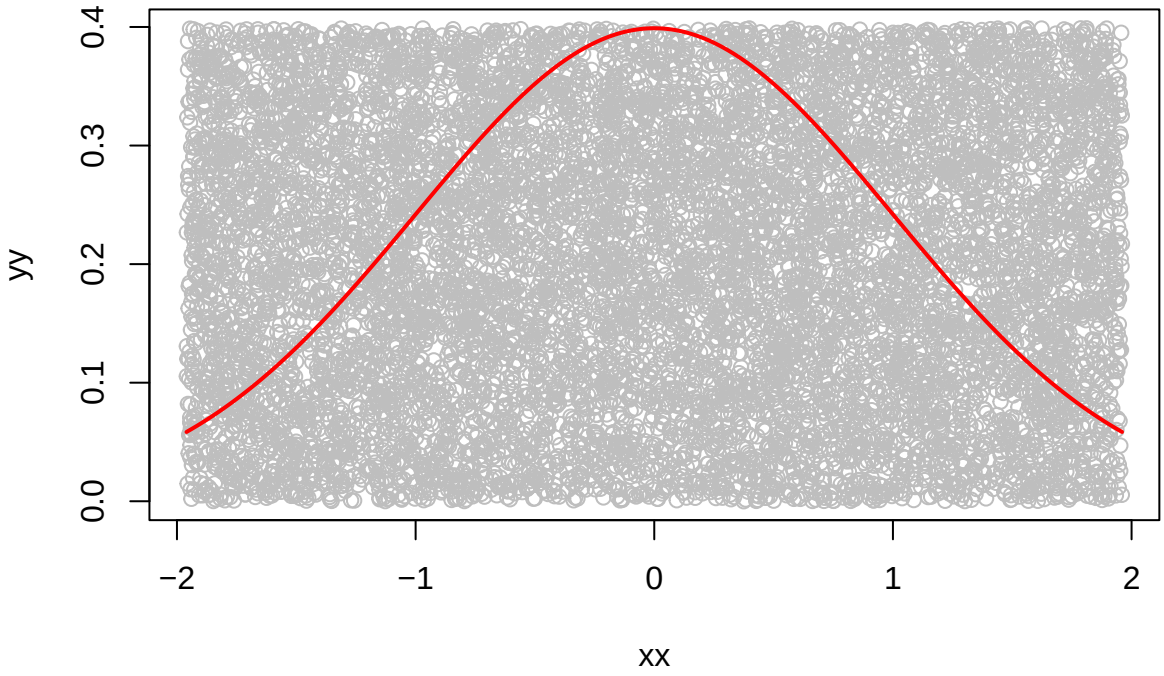
2번. 몬테카를로 적분

(1)

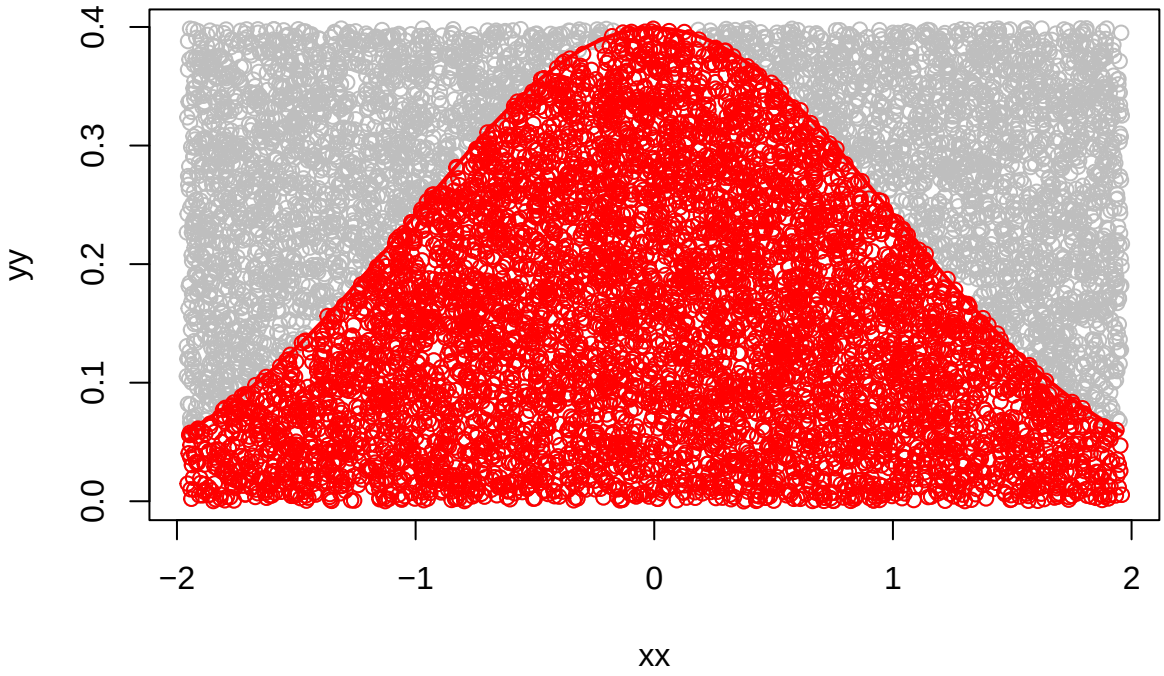
```r
x=seq(from=-1.96, to=1.96, by=0.01)
y=(1/sqrt(2*pi))*exp(-1*x^2/2)

plot(x,y)
```




```r
xx = runif(n=10000, min=-1.96, max=1.96)
yy = runif(n=10000, min=0, max=1/sqrt(2*pi))

plot(xx,yy, col='gray')
lines(x,y, col='red', lwd=2)
```




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

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=2)
points(xx[tst], yy[tst], col='red')
```




```
1/sqrt(2*pi)*3.92*sum(test(xx,yy))/10000
```

```
## [1] 0.9522305
```

(2)

```
x= rnorm(1000)
sum(-1.96<x & x<1.96)
```

```
## [1] 956
```

3. 징검다리

```
#변수 선언
arr= c('N1','N2','N3','N4','N5','N6','N7','N8','A9','N10')
surv = 10
stage = 0
player = arr[surv]
p=0.5
toss_rslt = NA
```

```
#함수
toss = function(p){
  if(surv>=1){as.logical(rbinom(n=1, size=1, prob=p))
    }else{TRUE}
    }

reset= function(){
    toss_rslt <<- NA
    surv <<- 10
    stage <<- 0
    player <<- arr[surv]
    }

record = function(){
    list(toss_rslt=toss_rslt, surv=surv, stage=stage, player=player)}

go = function(){
    prob <<- 0.5+ (player=='A9')*0.45
    toss_rslt <<- toss(prob)
    if(surv>=1){
        if(toss_rslt==FALSE){
            surv <<- surv-1
            stage <<- stage+1
            player<<- arr[surv]
        }
    }else if(surv==0){
        toss_rslt <<- NA
        surv <<- 0
        stage <<- stage
        player <<- NA
        }
}

gogo = function(){
  for(i in 1:20) go()
}
```

```
reset()
gogo()
record()
```

```
## $toss_rslt
## [1] TRUE
##
## $surv
## [1] 9
##
## $stage
## [1] 1
##
## $player
## [1] "A9"
```

```
simulate_once = function(){
    reset()
    gogo()
    return(record()$surv)
    }
simulate_once()
```

```
## [1] 4
```

(Type A)

```
sim_rslt = c()
for(i in 1:1000){
  sim_rslt[i] = simulate_once()
}

sum(sim_rslt>=8)/1000
```

```
## [1] 0.457
```

(Type B)

```
arr=rev(arr)

sim_rslt = c()
for(i in 1:1000){
  sim_rslt[i] = simulate_once()
}

sum(sim_rslt>=3)/1000
```

```
## [1] 0.133
```

8번 참가자는 Type A일때 생존확률이 더 높다.

4번. COVID19

```
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
```

(1)

```
sumc = df %>% group_by(year) %>% summarise(sum(cases))
sumc
```

```
## # A tibble: 2 x 2
##    year `sum(cases)`
##   <dbl>        <dbl>
## 1  2020        60726
## 2  2021       396886
```

```
list = list(sumc$`sum(cases)`[1], sumc$`sum(cases)`[2])
names(list) <- c('2020년 확진자 총합', '2021년 확진자 총합')
list
```

```
## $`2020년 확진자 총합`
## [1] 60726
##
## $`2021년 확진자 총합`
## [1] 396886
```

(2)

```
summa <- df %>% filter(year==2020 & month==2 & day>=1 & day<=15) %>% group_by(prov) %>% summarise(sum(ca

summa$prov[summa$`sum(cases)`==max(summa$`sum(cases)`)]
```

```
## [1] "경기"
```

경기지역이 가장 많다.

(3)

```
summs = df %>% filter(year==2020 & month==2 & day>=16 & day<=29) %>% group_by(prov) %>% summarise(sum(ca

summs$prov[summs$`sum(cases)`==max(summs$`sum(cases)`)]
```

```
## [1] "대구"
```

대구지역이 가장 많다.