R입문 기말고사

김지은(201816911)

1번 회귀분석

-(1) e벡터생성

```
set.seed(1)
ei=rnorm(1000)
head(ei)
```

```
## [1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078 -0.8204684
```

-(2) x1,x2 생성

```
i=1:1000
ti= (2*pi*i) / 1000

i=1:1000
x1=sin(ti[i])
x2=cos(4*ti[i])
head(x1)
```

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

```
head(x2)
```

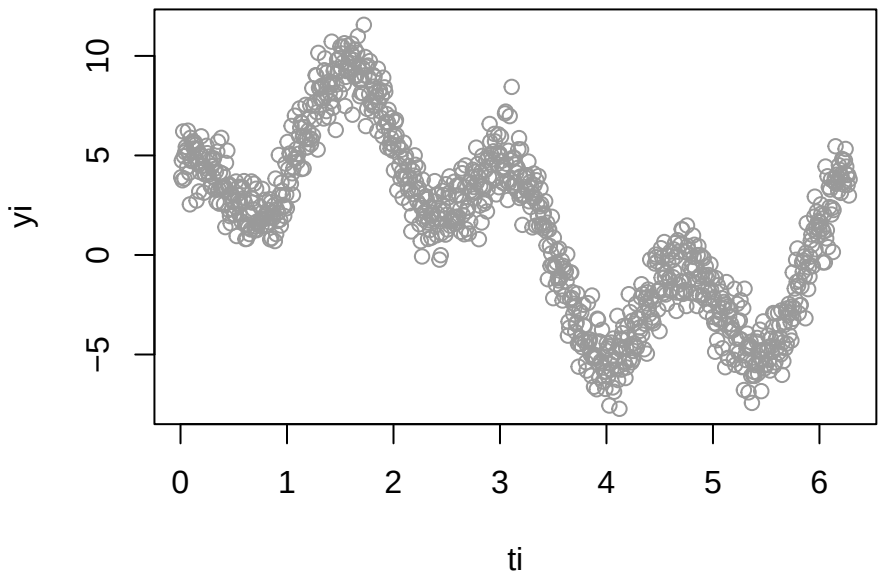
```
## [1] 0.9996842 0.9987370 0.9971589 0.9949510 0.9921147 0.9886517
```

-(3) y계산, 시각화

```
i=1:1000
yi=1.5+ (5*x1[i]) + (3*x2[i]) + ei[i]
head(yi)
```

```
## [1] 3.904014 4.742684 3.750090 6.205784 4.962906 3.833938
```

```
df <- data.frame(ti,yi)
plot(x=ti,y=yi,col="gray60")
```



-(4) 매트릭스 X 생성

```
X= cbind(1,x1,x2)
head(X)
```

```
##                  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
```

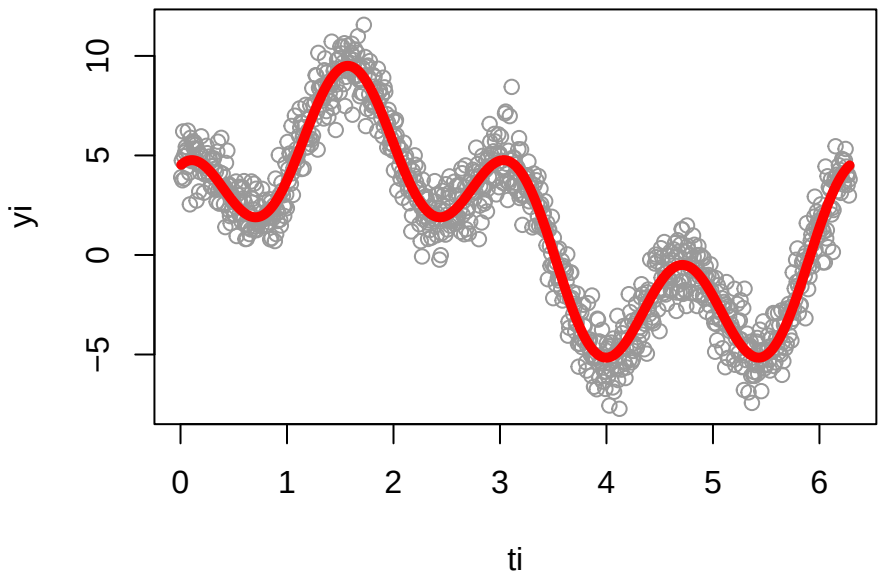
---

(5) 매트릭스 B 생성, 매트릭스곱으로 XB 계산, 시각화

```
B=matrix(c(1.5,5,3))
B
```

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

```
XB=X%*%B  #X와 B의 곱
XB_v=XB[,1]   #벡터화

plot(x=ti,y=yi,col="gray60")
lines(ti,XB_v , col = "Red",lwd=5)  #선 추가
```



---

(6) Bhat 계산 , B와 비교

```
X_t=t(X)  #전치행렬
B_hat=solve(X_t%*%X)%*%X_t%*%yi  # bhat구하기

B_hat
```

```
##         [,1]
##    1.488352
## x1 5.042240
## x2 3.020600
```

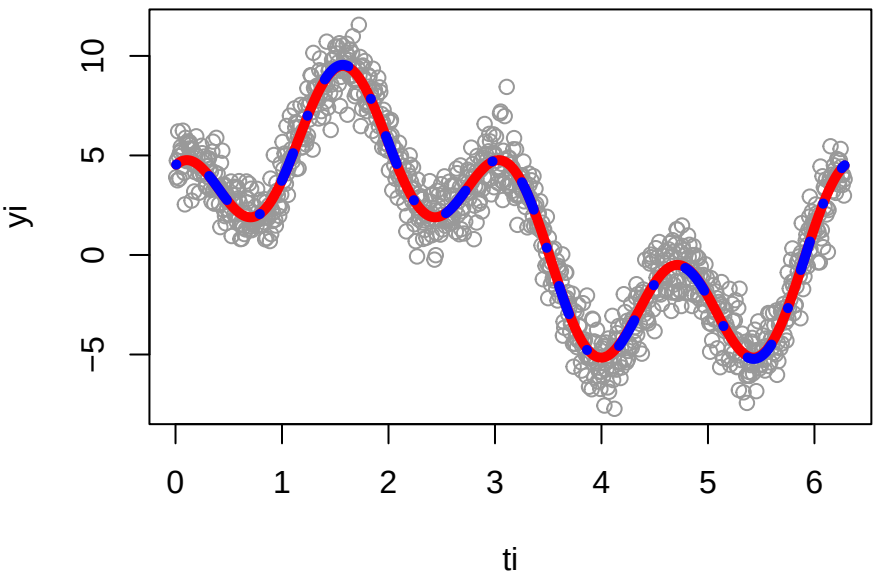
```
B
```

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

---

(7)XBhat 계산, 시각화

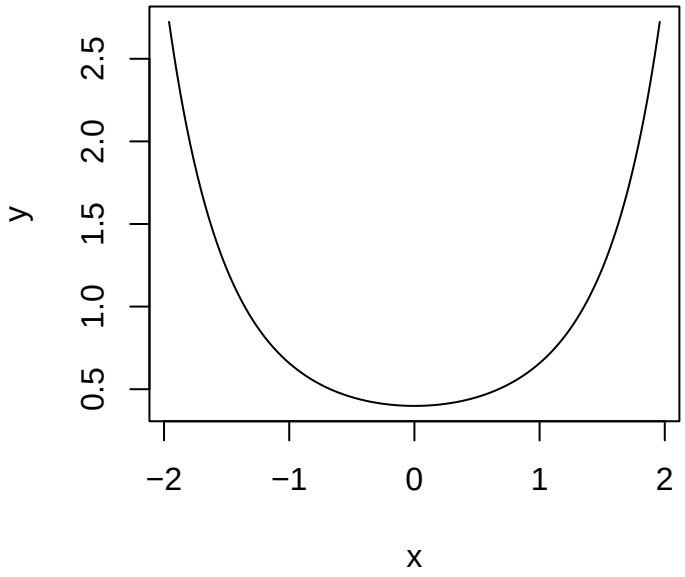
```
XB_hat = X%*%B_hat

plot(x=ti,y=yi,col="gray60")
lines(ti,XB_v , col = "Red",lwd=5)
lines(ti,XB_hat , col = "Blue",lwd=5, lty="dotdash")  #푸른점선추가
```


yi
10
5
0
−5
0
1
2
3
4
5
6
ti

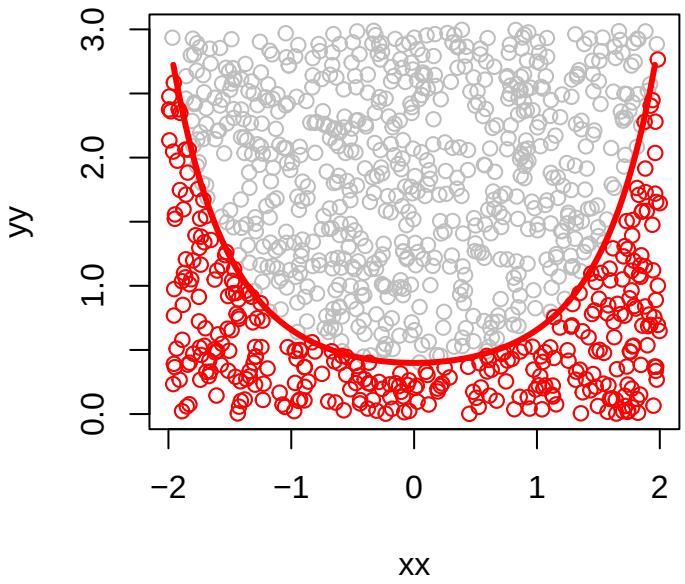
2번 몬테카를로 적분

(1) runif()이용. 몬테카를로 적분

```
set.seed(1)
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')
```




```
xx=runif(1000, min=-2, max=2)  #-2~2사이 임의 점 뽑기
yy=runif(1000, min=0, max=3)

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

tst = c()
for (i in 1:1000) 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')  #선 아래 점 표시
```




```
sum(tst)
```

```
## [1] 340
```

```
340/1000 * 12
```

```
## [1] 4.08
```

=> 답: 4.08이다.

---

(2) rnorm() 이용, 구간 값 count

```
set.seed(1)
x_2=rnorm(1000)
sum ( x_2 > -1.96 & x_2 <1.96 )
```

```
## [1] 932
```

=> 답: 932개

3번 징검다리

- Type A : 10번->9번->8번->7번 .. 1번

```
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()
```

```
ARR= c('N1','N2','N3','N4','N5','N6','N7','N8','N9','N10')

toss = function(p) rbinom(n=1,size=1,prob=p) %>% as.logical  #실행

reset = function(){
    TOSSRSLT <<- NA
    SURV <<- 10
    STAGE <<- 0
    PLAYER <<- ARR[SURV]
}

record = function(){
    list(PRE_TOSSRSLT=TOSSRSLT, SURV=SURV, STAGE=STAGE, PLAYER=PLAYER)
}

go = function(){
    PROB <<- 0.5 + (PLAYER=='N9')*0.45   #9번은 0.95확률
    TOSSRSLT <<- toss(PROB)
    if (TOSSRSLT==FALSE) SURV <<- SURV - 1   #일반유리 일때
    STAGE <<- STAGE + 1
    PLAYER <<- ARR[SURV]
}

gogo = function() for(i in 1:20) go()  #20번 실행

simulate_once = function(){
    reset()
    gogo()
    return(record()$SURV )  #생존자 반환
}
```

=> 8번 참가자 생존 하는 경우 = SURV가 8명 이상인 경우 = 8,9,10명인 경우

```
simrslt = c()
for (i in 1:1000) simrslt[i] = simulate_once()
sum(simulate_once()>=8)/1000
```

- Type B: 1번->2번->3번,..,8번->9번->10번

```
ARR= c('N10','N9','N8','N7','N6','N5','N4','N3','N2','N1')

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=PLAYER)
}

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

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

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

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

```
## $PRE_TOSSRSLT
## [1] NA
##
## $SURV
## [1] 10
##
## $STAGE
## [1] 0
##
## $PLAYER
## [1] "N1"
```

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

```
## $PRE_TOSSRSLT
## [1] FALSE
##
## $SURV
## [1] -1
##
```

```
## $STAGE
## [1] 20
##
## $PLAYER
## [1] "N9" "N8" "N7" "N6" "N5" "N4" "N3" "N2" "N1"
```

=> 8번 참가자가 생존하는 경우 = 3명이상 생존하는 경우 = 3~10명인 경우

```
simrslt = c()
for (i in 1:1000) simrslt[i] = simulate_once()
sum(simulate_once()>=3)/1000
```

```
## [1] 0
```

=> 생존확률이 거의 0 ~ 0.001 정도 나옴

4번 COVID19

(1) 2020년, 2021년 총합

```
library(tidyverse)
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 %>% group_by(year) %>% summarise(sum_cases = sum(cases))
```

```
## # A tibble: 2 x 2
##    year sum_cases
##   <dbl>     <dbl>
## 1  2020     60726
## 2  2021    396886
```

=> 2020년 확진자 수 = 60726명, 2021년 확진자 수 = 396886명으로 나타난다

(2) 2020년 2월1일 ~ 2020년 2월15일 기간 확진자의 총합 지역별

```
df %>% filter(year==2020 & day >=1 & day <=15 & month ==2) %>% 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
```

=> 경기지역이 9명으로 가장 많은 확진자가 발생했다.

---

(3) 2020년 2월16일 ~ 2020년 2월29일 기간 확진자의 총합 지역별

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

```
## # 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 2055명으로 가장 많은 확진자가 발생했다.