Big Data Analysis and Application: Assignment #1

Geonsik Yu

2015 10 5

Question 1. NYT Click Through Rate

```
## 데이터 파일 nyt1.csv가 작업 디렉토리에 없을 경우,
## 관련 데이터 파일을 전부 다운로드.
data_array <- list()
if(!file.exists("./Data_NYT/nyt1.csv")){
  for(i in 1:31){
    temp <- paste("http://stat.columbia.edu/~rachel/datasets/nyt", i,".csv", sep="")
    data_array[[i]] <- read.csv(url(temp) ,header=TRUE)
    write.table(data_array[[i]], file = paste("./Data_NYT/nyt",i,".csv",sep=""),
                quote = FALSE, sep = ",", row.names = FALSE, col.names = TRUE)
  }
}
## 작업 디렉토리에 있는 csv 파일을 읽어 리스트에 각각 데이터 프레임으로 저장.
for(i in 1:31){
  data_array[[i]] <- read.csv( paste("./Data_NYT/nyt",i,".csv",sep="") ,header=TRUE)
}
```

Question 1-1. nyt1.csv를 호출해서 age를 기준으로 <=18, 18<&<=24, 24<&=34, 34<&<=44, 44<&<=54, 54<&<=64, 64< 로 나누어서 범주화한 값을 age_group에 입력하시오.

```
data1 = data_array[[1]]
data1$age_group <- cut(data1$Age,c(-Inf,0,18,24,34,44,54,64,Inf))
```

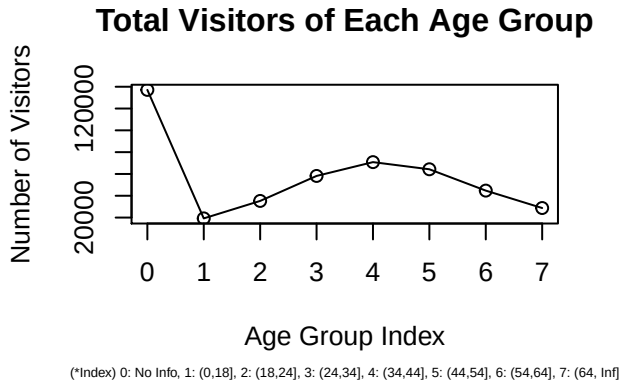
Question 1-2. Impressions 대비 Clicks 비율을 CTR이라는 변수로 만드시오.

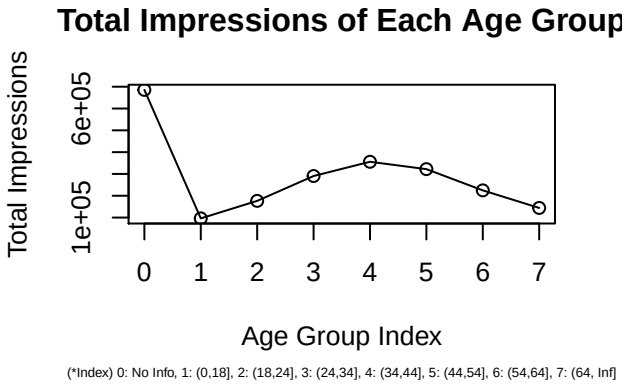
```
data1$CTR <- data1$Clicks / data1$Impressions
```

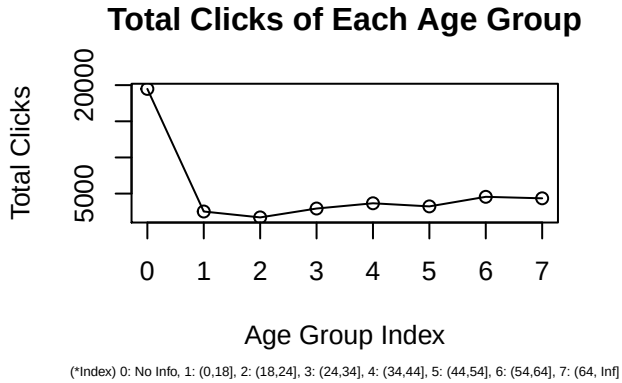
Question 1-3. nyt1.csv의 age_group별 총 방문자 수, 총 Impressions, 총 Clicks, 평균 CTR 을 그래프로 표현하시오.

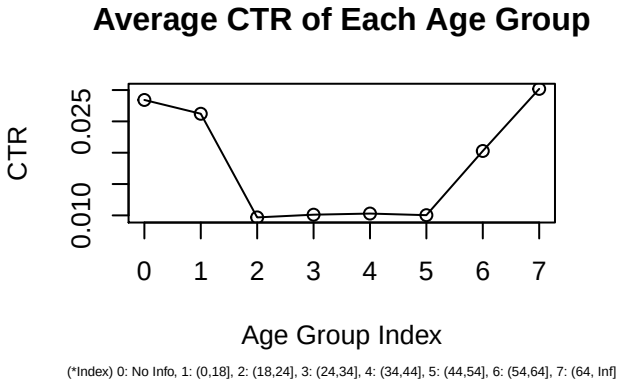
```
subsets_age = list()
lvls = levels(data1$age_group)
group_summary <- data.frame( "Count_Visitors" = numeric(0),
                             "Total_Impression" = numeric(0),
                             "Total_Clicks" = numeric(0),
                             "Average_CTR" = numeric(0) )
for( i in 1:length(lvls)){
  subsets_age[[i]]    <-  subset( data1, data1$age_group == lvls[i] )
  number_of_visitors  <-  nrow(subsets_age[[i]])
  total_impressions   <-  sum(subsets_age[[i]][,"Impressions"])
  total_clicks        <-  sum(subsets_age[[i]][,"Clicks"])
  total_CTR           <-  total_clicks / total_impressions
  group_summary[lvls[i],] <- c( number_of_visitors, total_impressions,
                                total_clicks, total_CTR )
}

par(mfrow=c(2, 2))
plot( 0:7, group_summary$Count_Visitors, type = 'o',
      main="Total Visitors of Each Age Group",
```

```
      ylab="Number of Visitors", xlab="Age Group Index",
      sub = paste("(*Index) 0: No Info, 1: (0,18], 2: (18,24], 3:",
      "(24,34], 4: (34,44], 5: (44,54], 6: (54,64], 7: (64, Inf]"), cex.sub=0.5)
plot( 0:7, group_summary$Total_Impression, type = 'o',
      main="Total Impressions of Each Age Group",
      ylab="Total Impressions", xlab="Age Group Index",
      sub = paste("(*Index) 0: No Info, 1: (0,18], 2: (18,24], 3:",
      "(24,34], 4: (34,44], 5: (44,54], 6: (54,64], 7: (64, Inf]"), cex.sub=0.5)
plot( 0:7, group_summary$Total_Clicks, type = 'o',
      main="Total Clicks of Each Age Group",
      ylab="Total Clicks", xlab="Age Group Index",
      sub = paste("(*Index) 0: No Info, 1: (0,18], 2: (18,24], 3:",
      "(24,34], 4: (34,44], 5: (44,54], 6: (54,64], 7: (64, Inf]"), cex.sub=0.5)
plot( 0:7, group_summary$Average_CTR, type = 'o',
      main="Average CTR of Each Age Group",
      ylab="CTR", xlab="Age Group Index",
      sub = paste("(*Index) 0: No Info, 1: (0,18], 2: (18,24], 3:",
      "(24,34], 4: (34,44], 5: (44,54], 6: (54,64], 7: (64, Inf]"), cex.sub=0.5)
```







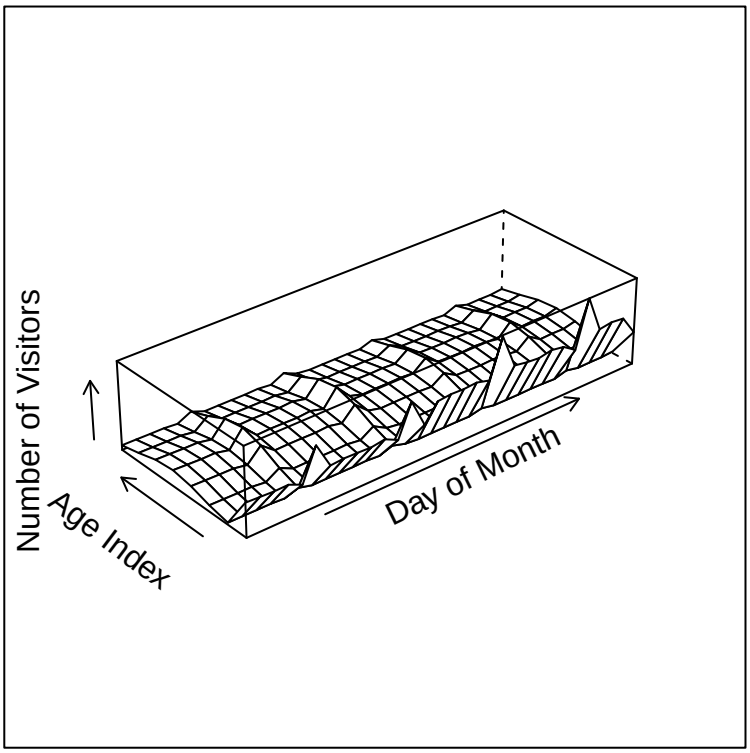


Question 1-4. 1일부터 31일까지 데이터를 모두 받아서 age_group별 총 방문자 수, 총 Impressions, 총 Clicks, 평균 CTR을 1일부터 31일까지 표현하는 그래프를 그리시오.

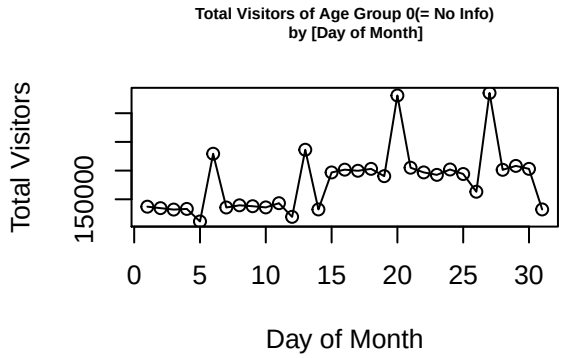
```
## 전체(1일~31일) 데이터를 일별 요약하여 group_summaries 리스트에 데이터 프레임 형태로 저장.
group_summaries = list()
for( k in 1:31){
  temp_subsets_age = list()
  lvls = levels(data1$age_group)
  temp_summary <- data.frame( "Count_Visitors" = numeric(0),
                              "Total_Impression" = numeric(0),
                              "Total_Clicks" = numeric(0),
                              "Average_CTR" = numeric(0) )
  data_array[[k]]$age_group <- cut(data_array[[k]]$Age,c(-Inf,0,18,24,34,44,54,64,Inf))
  for( i in 1:length(lvls)){
    temp_subsets_age[[i]]    <-  subset( data_array[[k]], data_array[[k]]$age_group == lvls[i] )
    number_of_visitors  <-  nrow(temp_subsets_age[[i]])
    total_impressions   <-  sum(temp_subsets_age[[i]][,"Impressions"])
    total_clicks        <-  sum(temp_subsets_age[[i]][,"Clicks"])
    total_CTR           <-  total_clicks / total_impressions
    temp_summary[lvls[i],] <- c( number_of_visitors, total_impressions,
                                 total_clicks, total_CTR )
  }
  group_summaries[[k]] <- temp_summary
}

## 1. 각각의 집계량(Visitors, Impressions, Clicks, CTR)에 대해서
##    x 축을 일별(Day of Month)로 하고, y 축을 연령그룹(Age Group)으로 하는 3차원 플롯
## 2. 연령 그룹(Age Roup)별로 일별 집계량을 시각화하는 총 8개(연령 그룹 수)의 2차원 플롯
## 집계량 데이터 처리.
if (!require("lattice", quietly = TRUE)){ install.packages("lattice") }
library(lattice)
data_grid <- expand.grid(x=1:31,y=0:7)
data_grid$visitors    <- 0
data_grid$impressions <- 0
data_grid$clicks      <- 0
data_grid$CTR         <- 0

for( i in 1:nrow(data_grid)){
  temp_x <- data_grid[i,"x"]
  temp_y <- data_grid[i,"y"]
  data_grid[i,"visitors"]     <- group_summaries[[temp_x]][temp_y+1,1]
  data_grid[i,"impressions"]  <- group_summaries[[temp_x]][temp_y+1,2]
  data_grid[i,"clicks"]       <- group_summaries[[temp_x]][temp_y+1,3]
  data_grid[i,"CTR"]          <- group_summaries[[temp_x]][temp_y+1,4]
}
```

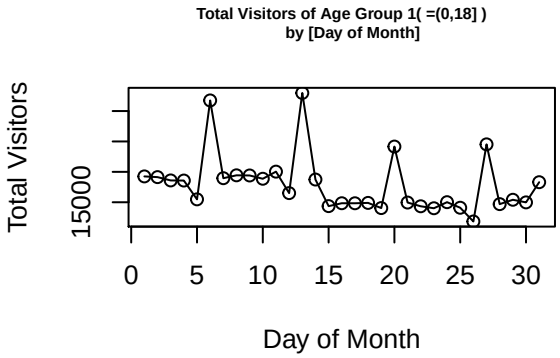
[시각화 1] '방문자 수(Number of Visitors)'

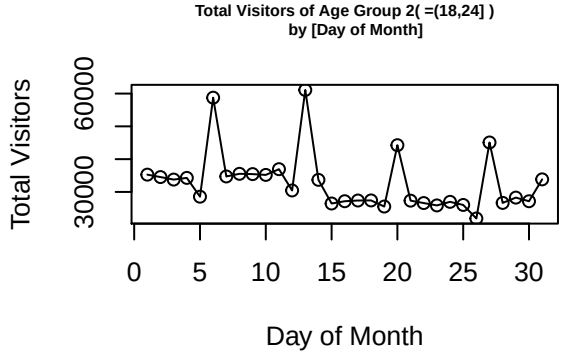
```
## '방문자 수(Number of Visitors)'에 대한 집계량을 3차원으로 시각화
##  x축: Day of Month(1~31), y축: 연령그룹 인덱스(0 ~ 7), z축: 방문자 수
wireframe(data_grid$visitors ~ data_grid$x+data_grid$y,data_grid,aspect=c(.4,.2),
          main="Number of Visitors by [ Day of Month + Age Group ]",
          xlab=list("Day of Month",rot=26),
          ylab=list("Age Index",rot=-37),
          zlab=list("Number of Visitors",rot=90))
```

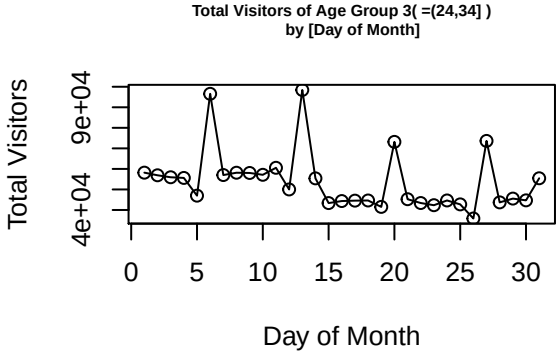



```
## '연령 그룹별 방문자 수(Number of Visitors)'에 대한 집계량을 2차원으로 시각화
##  x축: Day of Month(1~31), y축: 방문자 수, 총 8개의 플롯.
par(mfrow=c(2, 2))
plot(1:31, subset( data_grid$visitors, data_grid$y == 0 ), type='o',
     main="Total Visitors of Age Group 0(= No Info)
     by [Day of Month]",
     ylab="Total Visitors", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$visitors, data_grid$y == 1 ), type='o',
     main="Total Visitors of Age Group 1( =(0,18] )
     by [Day of Month]",
     ylab="Total Visitors", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$visitors, data_grid$y == 2 ), type='o',
     main="Total Visitors of Age Group 2( =(18,24] )
     by [Day of Month]",
     ylab="Total Visitors", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$visitors, data_grid$y == 3 ), type='o',
     main="Total Visitors of Age Group 3( =(24,34] )
     by [Day of Month]",
     ylab="Total Visitors", xlab="Day of Month", cex.main=.6)
```

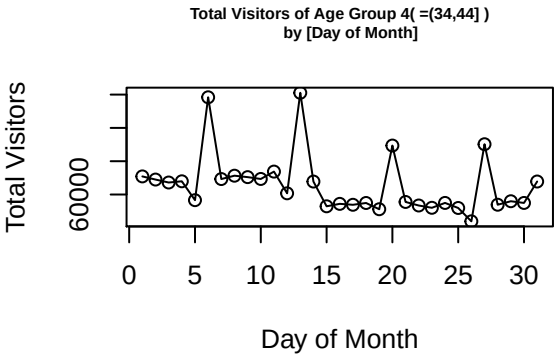


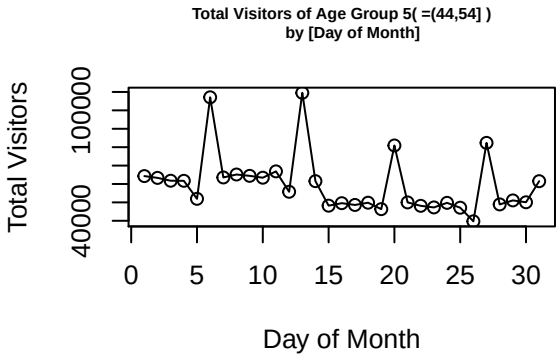


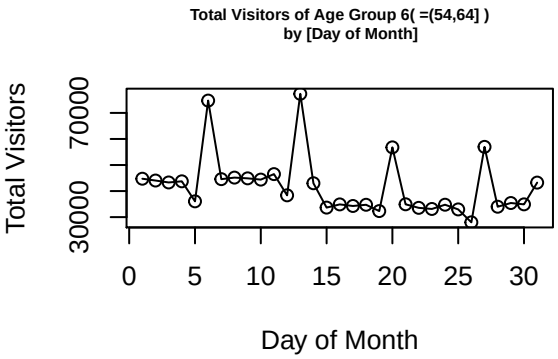


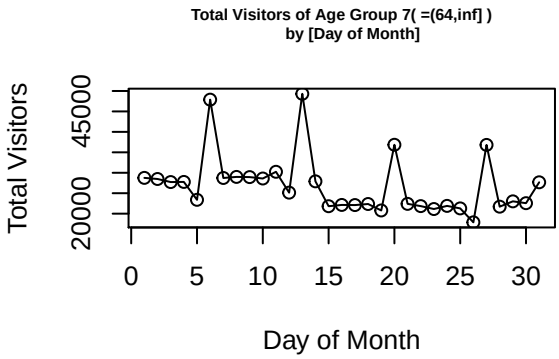



```
par(mfrow=c(2, 2))
plot(1:31, subset( data_grid$visitors, data_grid$y == 4 ), type='o',
     main="Total Visitors of Age Group 4( =(34,44] )
     by [Day of Month]",
     ylab="Total Visitors", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$visitors, data_grid$y == 5 ), type='o',
     main="Total Visitors of Age Group 5( =(44,54] )
     by [Day of Month]",
     ylab="Total Visitors", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$visitors, data_grid$y == 6 ), type='o',
     main="Total Visitors of Age Group 6( =(54,64] )
     by [Day of Month]",
     ylab="Total Visitors", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$visitors, data_grid$y == 7 ), type='o',
     main="Total Visitors of Age Group 7( =(64,inf] )
     by [Day of Month]",
     ylab="Total Visitors", xlab="Day of Month", cex.main=.6)
```


Total Visitors of Age Group 4(=(34,44])
by [Day of Month]
Total Visitors
60000
0
5
10
15
20
25
30
Day of Month

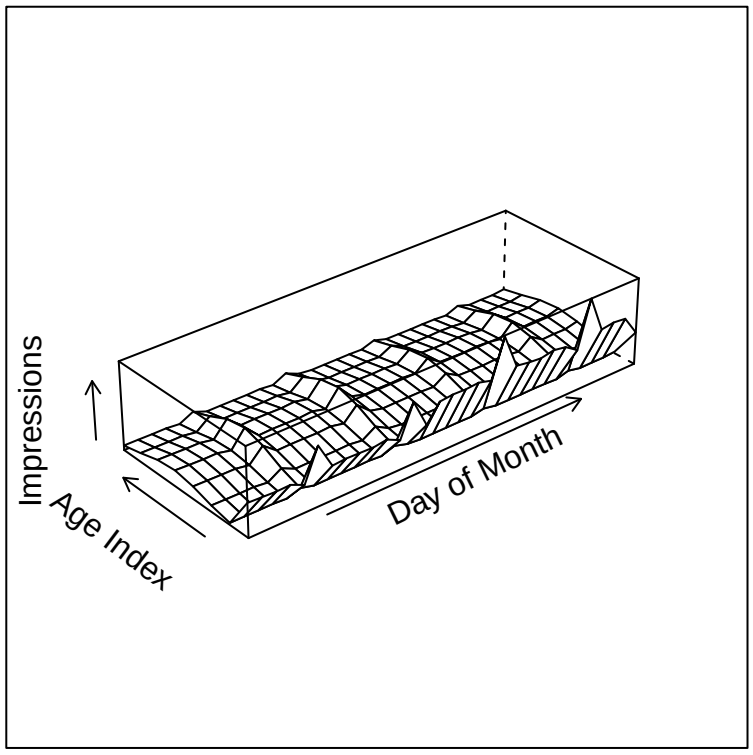

Total Visitors of Age Group 5(=(44,54])
by [Day of Month]
Total Visitors
40000
100000
0
5
10
15
20
25
30
Day of Month


Total Visitors of Age Group 6(=(54,64])
by [Day of Month]
Total Visitors
30000
70000
0
5
10
15
20
25
30
Day of Month

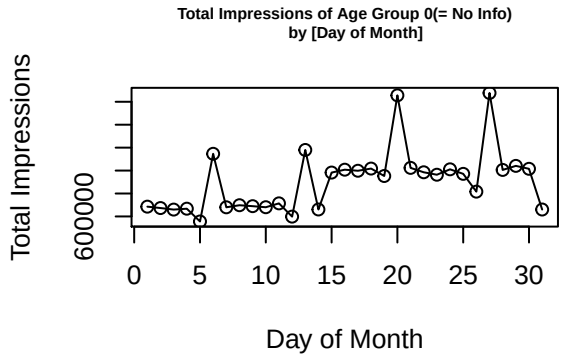

Total Visitors of Age Group 7(=(64,inf])
by [Day of Month]
Total Visitors
20000
45000
0
5
10
15
20
25
30
Day of Month

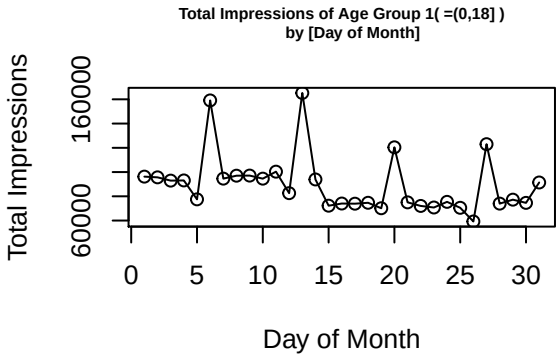
[시각화 2] 'Impressions'

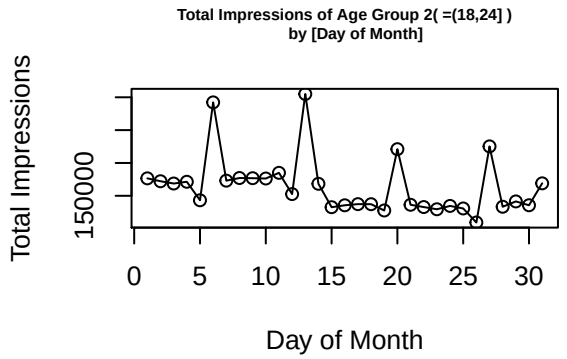
```
## 'Impression'에 대한 집계량을 3차원으로 시각화
##  x축: Day of Month(1~31), y축: 연령그룹 인덱스(0 ~ 7), z축: Impressions
wireframe(data_grid$impressions ~ data_grid$x+data_grid$y,data_grid,aspect=c(.4,.2),
          main="Total Impressions by [ Day of Month + Age Group ]",
          xlab=list("Day of Month",rot=26),
          ylab=list("Age Index",rot=-37),
          zlab=list("Impressions",rot=90))
```

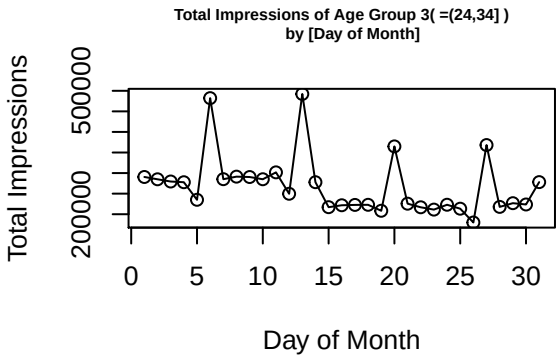



```
## '연령 그룹별 Impression'에 대한 집계량을 2차원으로 시각화
##  x축: Day of Month(1~31), y축: Impressions, 총 8개의 플롯.
par(mfrow=c(2, 2))
plot(1:31, subset( data_grid$impressions, data_grid$y == 0 ), type='o',
     main="Total Impressions of Age Group 0(= No Info)
     by [Day of Month]",
     ylab="Total Impressions", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$impressions, data_grid$y == 1 ), type='o',
     main="Total Impressions of Age Group 1( =(0,18] )
     by [Day of Month]",
     ylab="Total Impressions", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$impressions, data_grid$y == 2 ), type='o',
     main="Total Impressions of Age Group 2( =(18,24] )
     by [Day of Month]",
     ylab="Total Impressions", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$impressions, data_grid$y == 3 ), type='o',
     main="Total Impressions of Age Group 3( =(24,34] )
     by [Day of Month]",
     ylab="Total Impressions", xlab="Day of Month", cex.main=.6)
```

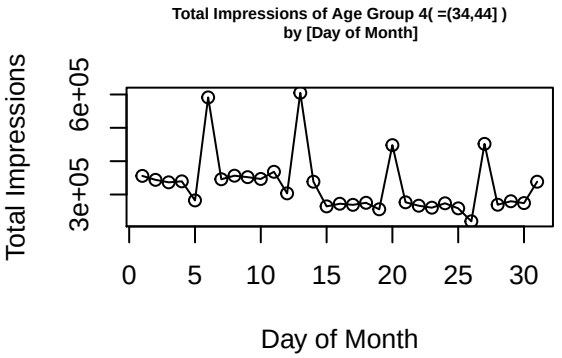


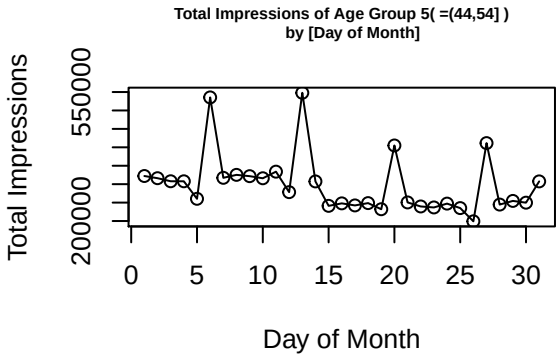


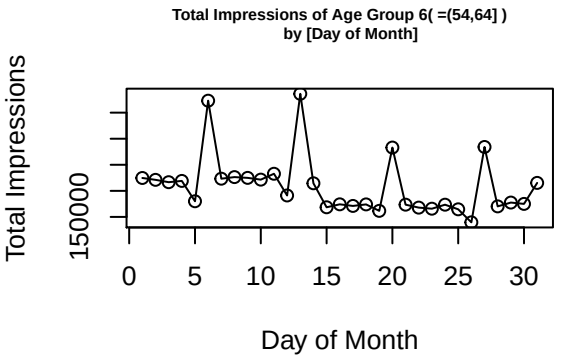


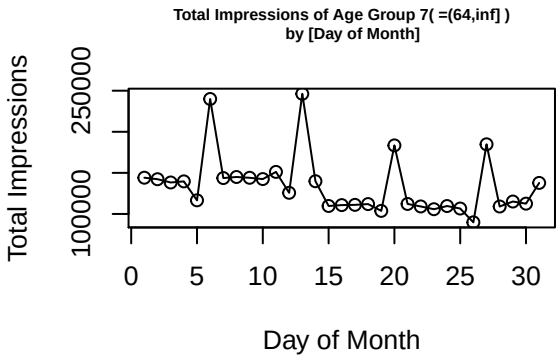



```
par(mfrow=c(2, 2))
plot(1:31, subset( data_grid$impressions, data_grid$y == 4 ), type='o',
     main="Total Impressions of Age Group 4( =(34,44] )
     by [Day of Month]",
     ylab="Total Impressions", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$impressions, data_grid$y == 5 ), type='o',
     main="Total Impressions of Age Group 5( =(44,54] )
     by [Day of Month]",
     ylab="Total Impressions", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$impressions, data_grid$y == 6 ), type='o',
     main="Total Impressions of Age Group 6( =(54,64] )
     by [Day of Month]",
     ylab="Total Impressions", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$impressions, data_grid$y == 7 ), type='o',
     main="Total Impressions of Age Group 7( =(64,inf] )
     by [Day of Month]",
     ylab="Total Impressions", xlab="Day of Month", cex.main=.6)
```


Total Impressions of Age Group 4(=(34,44])
by [Day of Month]
Total Impressions
3e+05
6e+05
0
5
10
15
20
25
30
Day of Month

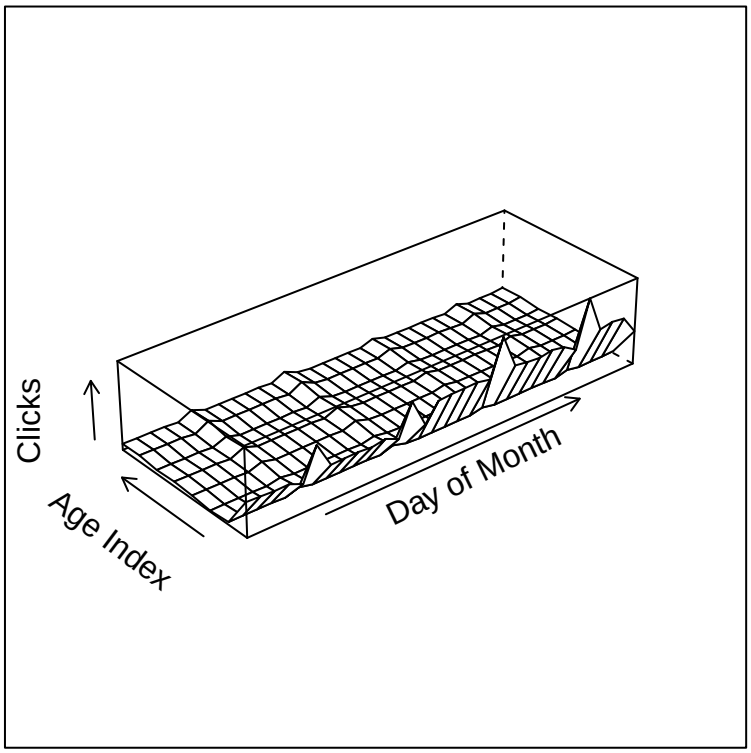

Total Impressions of Age Group 5(=(44,54])
by [Day of Month]
Total Impressions
200000
550000
0
5
10
15
20
25
30
Day of Month


Total Impressions of Age Group 6(=(54,64])
by [Day of Month]
Total Impressions
150000
0
5
10
15
20
25
30
Day of Month

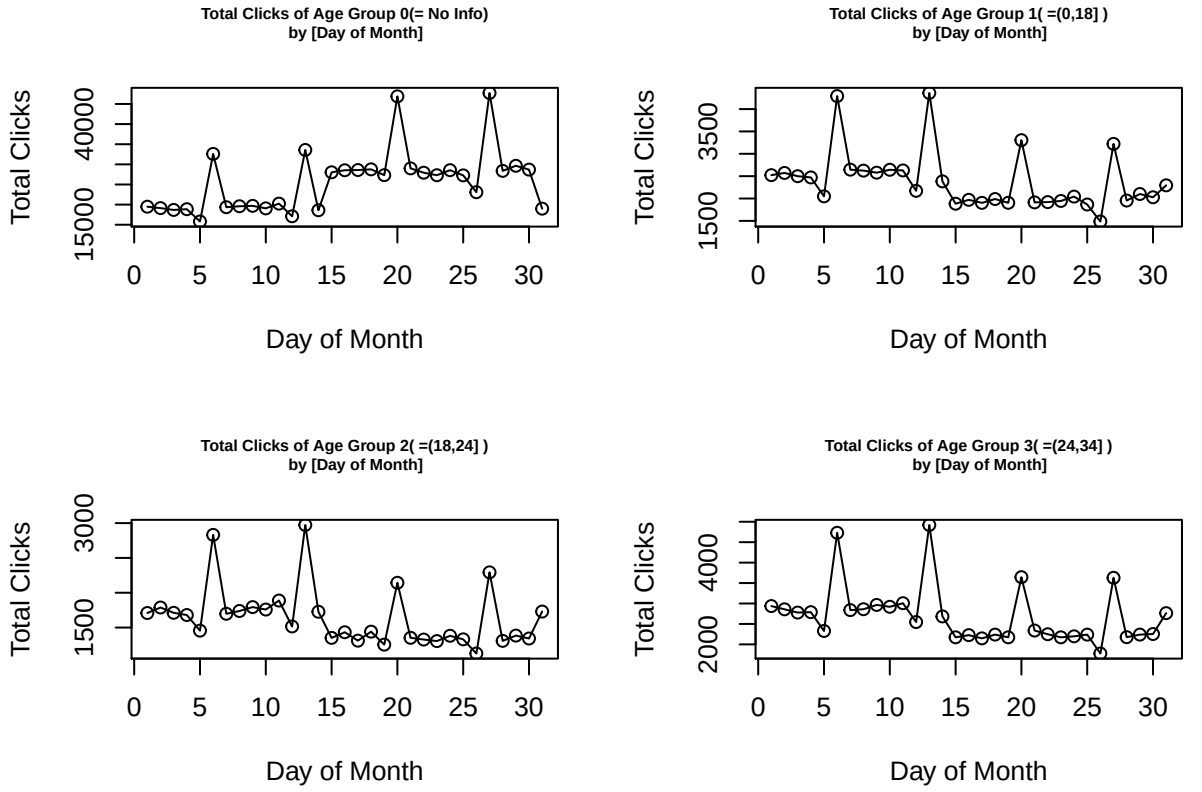

Total Impressions of Age Group 7(=(64,inf])
by [Day of Month]
Total Impressions
100000
250000
0
5
10
15
20
25
30
Day of Month

[시각화 3] ‘Clicks’

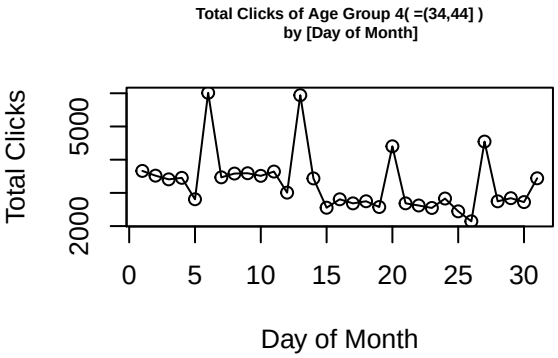
```
## 'Clicks'에 대한 집계량을 3차원으로 시각화
##  x축: Day of Month(1~31), y축: 연령그룹 인덱스(0 ~ 7), z축: Clicks
wireframe(data_grid$clicks ~ data_grid$x+data_grid$y,data_grid,aspect=c(.4,.2),
          main="Total Clicks by [ Day of Month + Age Group ]",
          xlab=list("Day of Month",rot=26),
          ylab=list("Age Index",rot=-37),
          zlab=list("Clicks",rot=90))
```

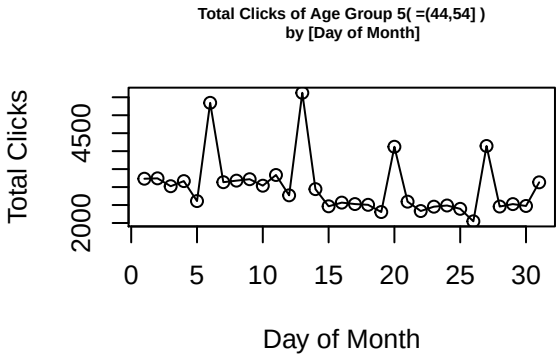


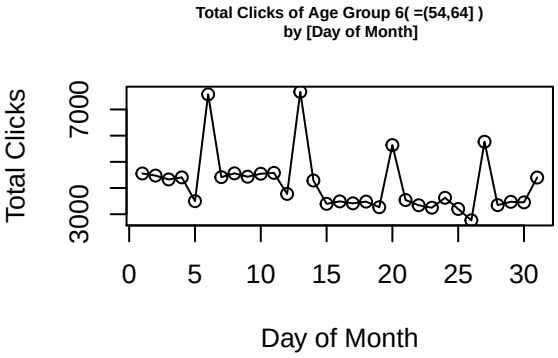

```
## '연령 그룹별 Clicks'에 대한 집계량을 2차원으로 시각화
##  x축: Day of Month(1~31), y축: Clicks, 총 8개의 플롯.
par(mfrow=c(2, 2))
plot(1:31, subset( data_grid$clicks, data_grid$y == 0 ), type='o',
     main="Total Clicks of Age Group 0(= No Info)
     by [Day of Month]",
     ylab="Total Clicks", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$clicks, data_grid$y == 1 ), type='o',
     main="Total Clicks of Age Group 1( =(0,18] )
     by [Day of Month]",
     ylab="Total Clicks", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$clicks, data_grid$y == 2 ), type='o',
     main="Total Clicks of Age Group 2( =(18,24] )
     by [Day of Month]",
     ylab="Total Clicks", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$clicks, data_grid$y == 3 ), type='o',
     main="Total Clicks of Age Group 3( =(24,34] )
     by [Day of Month]",
     ylab="Total Clicks", xlab="Day of Month", cex.main=.6)
```

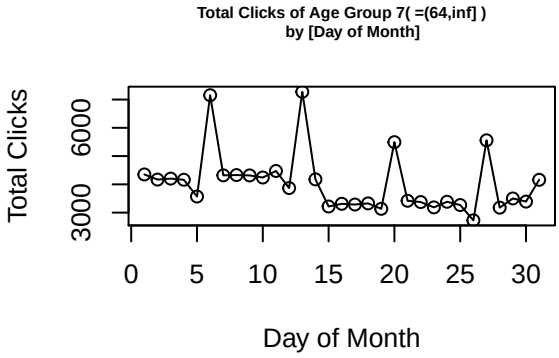



```
par(mfrow=c(2, 2))
plot(1:31, subset( data_grid$clicks, data_grid$y == 4 ), type='o',
     main="Total Clicks of Age Group 4( =(34,44] )
     by [Day of Month]",
     ylab="Total Clicks", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$clicks, data_grid$y == 5 ), type='o',
     main="Total Clicks of Age Group 5( =(44,54] )
     by [Day of Month]",
     ylab="Total Clicks", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$clicks, data_grid$y == 6 ), type='o',
     main="Total Clicks of Age Group 6( =(54,64] )
     by [Day of Month]",
     ylab="Total Clicks", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$clicks, data_grid$y == 7 ), type='o',
     main="Total Clicks of Age Group 7( =(64,inf] )
     by [Day of Month]",
     ylab="Total Clicks", xlab="Day of Month", cex.main=.6)
```


Total Clicks of Age Group 4(=(34,44])
by [Day of Month]
Total Clicks
2000
5000
0
5
10
15
20
25
30
Day of Month

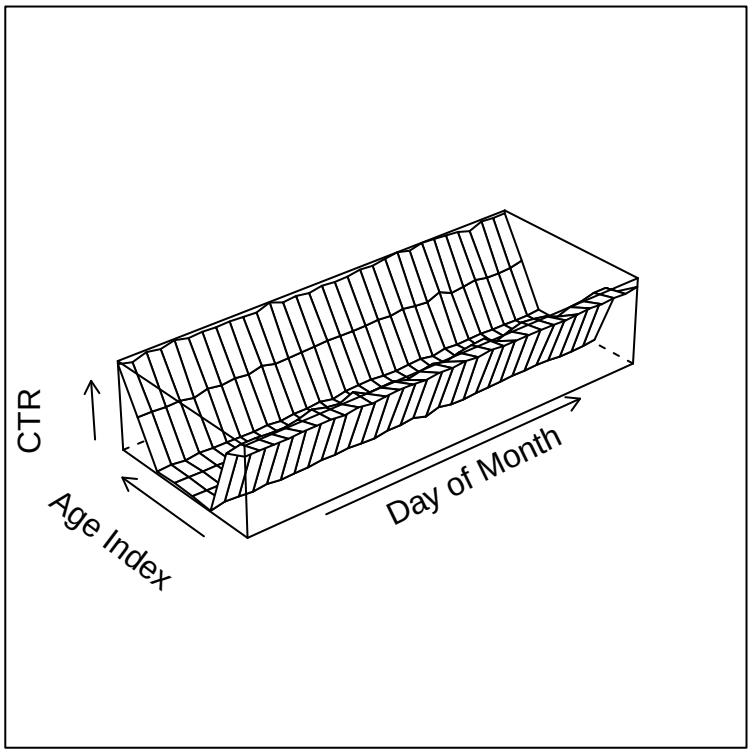

Total Clicks of Age Group 5(=(44,54])
by [Day of Month]
Total Clicks
2000
4500
0
5
10
15
20
25
30
Day of Month


Total Clicks of Age Group 6(=(54,64])
by [Day of Month]
Total Clicks
3000
7000
0
5
10
15
20
25
30
Day of Month

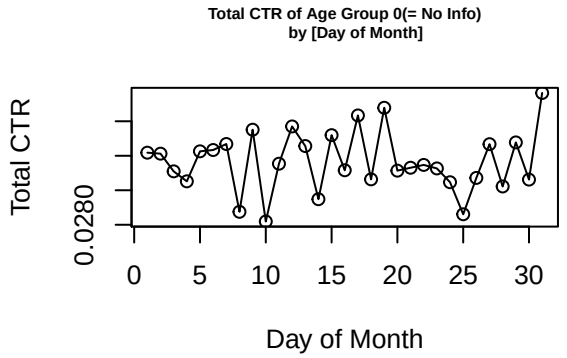

Total Clicks of Age Group 7(=(64,inf])
by [Day of Month]
Total Clicks
3000
6000
0
5
10
15
20
25
30
Day of Month

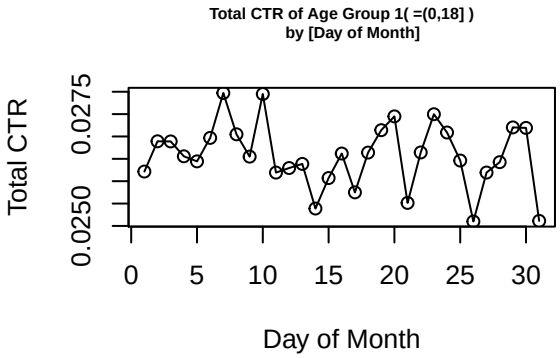
[시각화 4] ‘CTR’

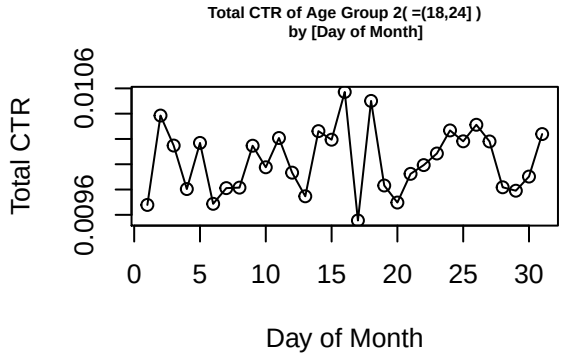
```
## 'CTR'에 대한 집계량을 3차원으로 시각화
##  x축: Day of Month(1~31), y축: 연령그룹 인덱스(0 ~ 7), z축: CTR
wireframe(data_grid$CTR ~ data_grid$x+data_grid$y,data_grid,aspect=c(.4,.2),
          main="Average CTR by [ Day of Month + Age Group ]",
          xlab=list("Day of Month",rot=26),
          ylab=list("Age Index",rot=-37),
          zlab=list("CTR",rot=90))
```

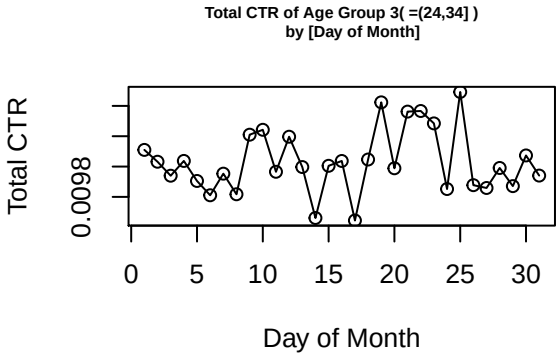



```
## '연령 그룹별 CTR'에 대한 집계량을 2차원으로 시각화
##  x축: Day of Month(1~31), y축: CTR, 총 8개의 플롯.
par(mfrow=c(2, 2))
plot(1:31, subset( data_grid$CTR, data_grid$y == 0 ), type='o',
     main="Total CTR of Age Group 0(= No Info)
     by [Day of Month]",
     ylab="Total CTR", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$CTR, data_grid$y == 1 ), type='o',
     main="Total CTR of Age Group 1( =(0,18] )
     by [Day of Month]",
     ylab="Total CTR", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$CTR, data_grid$y == 2 ), type='o',
     main="Total CTR of Age Group 2( =(18,24] )
     by [Day of Month]",
     ylab="Total CTR", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$CTR, data_grid$y == 3 ), type='o',
     main="Total CTR of Age Group 3( =(24,34] )
     by [Day of Month]",
     ylab="Total CTR", xlab="Day of Month", cex.main=.6)
```

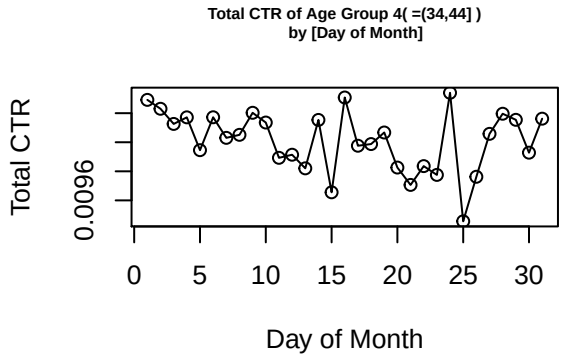


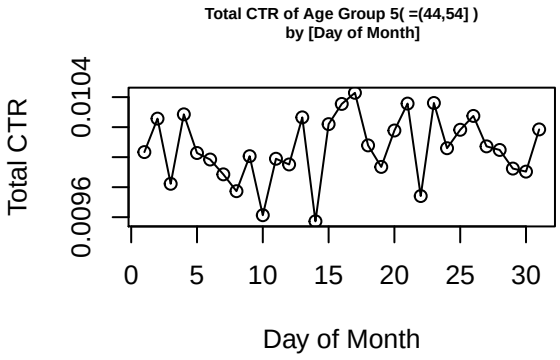


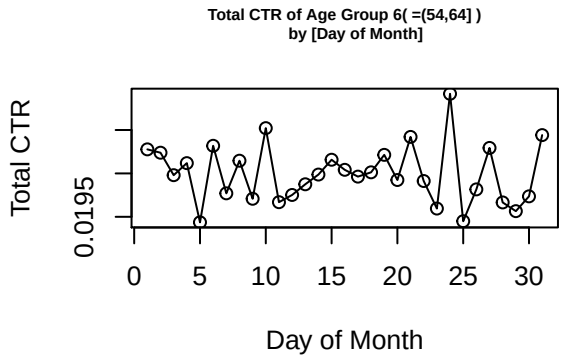


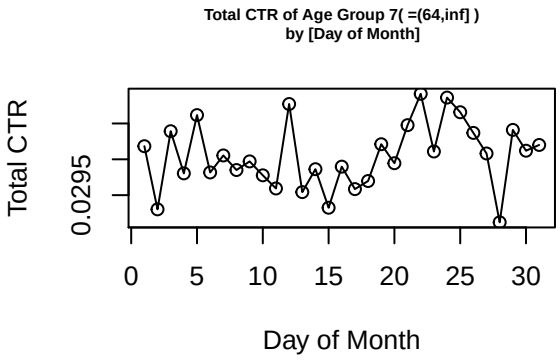



```
par(mfrow=c(2, 2))
plot(1:31, subset( data_grid$CTR, data_grid$y == 4 ), type='o',
     main="Total CTR of Age Group 4( =(34,44] )
     by [Day of Month]",
     ylab="Total CTR", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$CTR, data_grid$y == 5 ), type='o',
     main="Total CTR of Age Group 5( =(44,54] )
     by [Day of Month]",
     ylab="Total CTR", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$CTR, data_grid$y == 6 ), type='o',
     main="Total CTR of Age Group 6( =(54,64] )
     by [Day of Month]",
     ylab="Total CTR", xlab="Day of Month", cex.main=.6)
plot(1:31, subset( data_grid$CTR, data_grid$y == 7 ), type='o',
     main="Total CTR of Age Group 7( =(64,inf] )
     by [Day of Month]",
     ylab="Total CTR", xlab="Day of Month", cex.main=.6)
```


Total CTR of Age Group 4(=(34,44])
by [Day of Month]
Total CTR
0.0096
0
5
10
15
20
25
30
Day of Month


Total CTR of Age Group 5(=(44,54])
by [Day of Month]
Total CTR
0.0096
0.0104
0
5
10
15
20
25
30
Day of Month


Total CTR of Age Group 6(=(54,64])
by [Day of Month]
Total CTR
0.0195
0
5
10
15
20
25
30
Day of Month


Total CTR of Age Group 7(=(64,inf])
by [Day of Month]
Total CTR
0.0295
0
5
10
15
20
25
30
Day of Month

Question 2. Hitters

```
## ISLR 페키지가 먼서 설치되어있다고 가정하고 라이브러리를 호출했다.
if (!require("ISLR", quietly = TRUE)){ install.packages("ISLR") }
library(ISLR)
```

Quesiton 2-1. 학습집합을 이용하여 Salary를 가장 잘 예측하는 변수조합을 찾으시오. (분석자에 의해 제거해야하는 변수가 있으면 자의적인 제거 가능)

Answer)

메이저 리그는 총 2개의 리그(League: American / National)와 4개의 디비전(American - West/East, National West/East)로 이루어져 있다. 이 중 각의 디비전은 5개의 팀으로 구성되어 있고, 각 리그는 2개의 디비전(W/E)으로 이루어져 있다.(중부리그는 97년 이후에 창설) 이는 포스트시즌 진행 방식에 따른 구성 방식인데, 중요한 점은 이 구분으로 인해서 팀간 경기수가 다르다는 점이다. 다시 말하자면, 메이저리그에 속한 어떤 팀은 전체 경기를 진행함에 있어 상대적으로 더 많은 경기를 같은 지구(디비전, Division)의 팀과 수행하며, 같은 리그의 다른 팀이나 다른 리그 팀과는 그보다는 적은 경기 수를 수행한다.

주어진 자료에서 볼수 있는 League 변수(A 또는 N)나 Division 변수(W 또는 E) 각각으로는 이와같은 일정 관계를 모형에 포함시키기 어려우며, 따라서 이 두 변수를 모두 제거하거나 결합시켜(eg. AW, AE, NW, NE) 사용하는 것이 옳다. 이 중 본 분석은 두 변수를 제거하는 것을 선택했다.

또한 현재 소속 리그라는 변수를 제거함으로써 재개약으로 뛰게될 새 리그를 나타내는 변수인 NewLeague 역시 분석에서 제거해야 한다고 판단하고, 이를 제거했다.

```
## 문제의 지시대로 결측치를 포함한 데이터 엔티티를 삭제했다.
hitters_omit <- na.omit( Hitters )

## 위의 설명대로, League / Division / NewLeague 변수를 제거했다.
hitters_omit$League    <- NULL
hitters_omit$Division  <- NULL
hitters_omit$NewLeague <- NULL

## 트레이닝 파티션과 테스트 파티션으로 분할했다.
training_set <- hitters_omit[1:200,]
test_set     <- hitters_omit[201:nrow(hitters_omit),]
```

데이터의 사이즈가 크지 않고 변수의 수 또한 적으므로, Exhaustive Method 를 이용하여 모든 조합을 비교해보기로 했다.

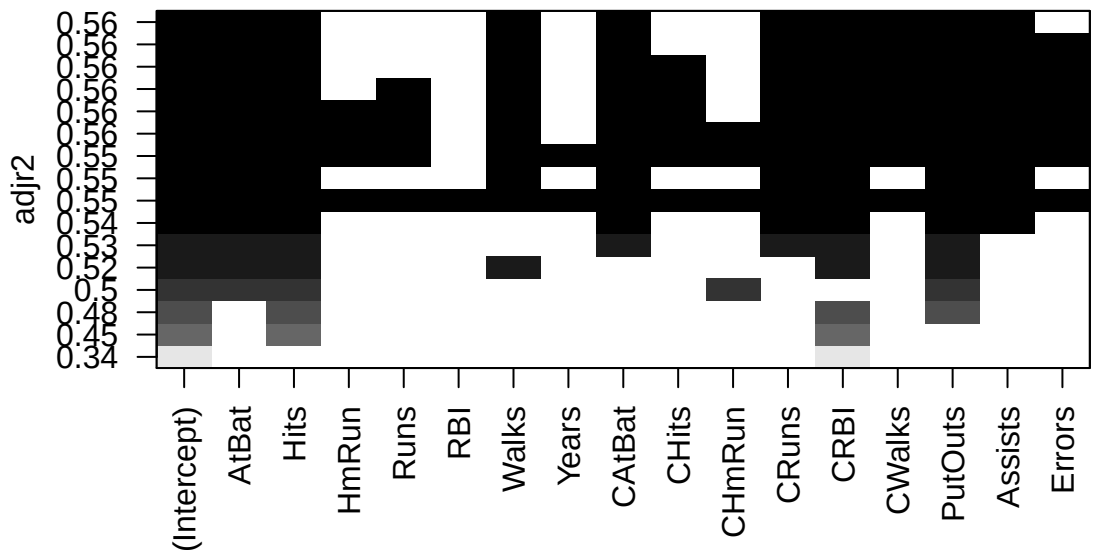
```
## leaps 페키지는 이미 설치되어있음을 가정했다.
if (!require("leaps", quietly = TRUE)){ install.packages("leaps") }
library(leaps)
regsubsets.result <- regsubsets(training_set$Salary~., data=training_set,
                                method="exhaustive", nvmax=16)
summary(regsubsets.result)
```

```
## Subset selection object
## Call: regsubsets.formula(training_set$Salary ~ ., data = training_set,
##     method = "exhaustive", nvmax = 16)
## 16 Variables  (and intercept)
##         Forced in Forced out
## AtBat       FALSE      FALSE
## Hits        FALSE      FALSE
## HmRun       FALSE      FALSE
## Runs        FALSE      FALSE
## RBI         FALSE      FALSE
## Walks       FALSE      FALSE
```

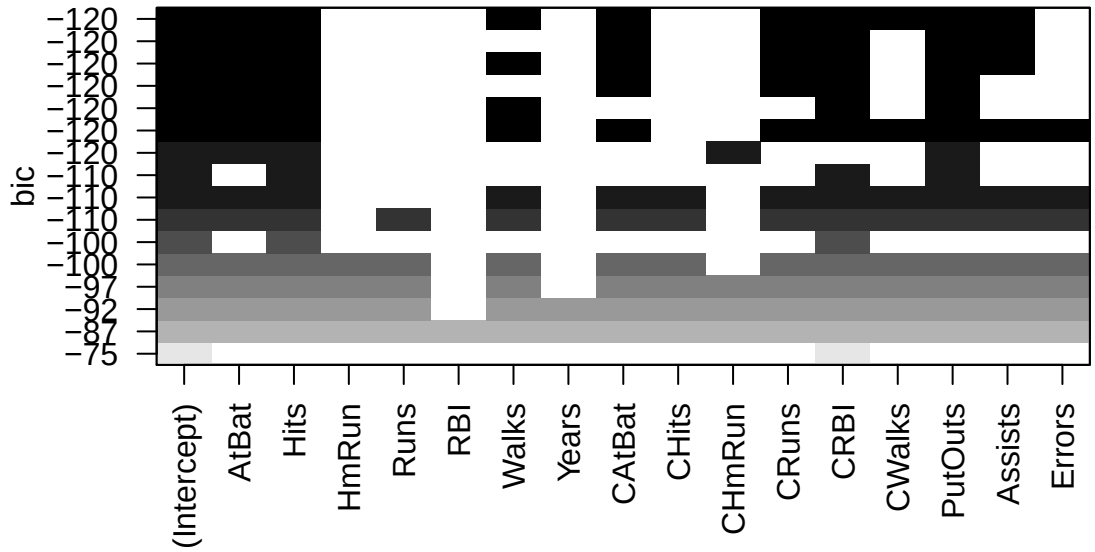
```
## Years       FALSE      FALSE
## CAtBat      FALSE      FALSE
## CHits       FALSE      FALSE
## CHmRun      FALSE      FALSE
## CRuns       FALSE      FALSE
## CRBI        FALSE      FALSE
## CWalks      FALSE      FALSE
## PutOuts     FALSE      FALSE
## Assists     FALSE      FALSE
## Errors      FALSE      FALSE
## 1 subsets of each size up to 16
## Selection Algorithm: exhaustive
##           AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns
## 1  ( 1 )  " "   " "  " "   " "  " " " " " "   " "   " "    " "   " "    " "
## 2  ( 1 )  " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
## 3  ( 1 )  " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
## 4  ( 1 )  "*"   "*"  " "   " "  " " " " " "   " "   " "    " "   "*"    " "
## 5  ( 1 )  "*"   "*"  " "   " "  " " " " "*"   " "   " "    " "   " "    " "
## 6  ( 1 )  "*"   "*"  " "   " "  " " " " " "   " "   "*"    " "   " "    "*"
## 7  ( 1 )  "*"   "*"  " "   " "  " " " " " "   " "   "*"    " "   " "    "*"
## 8  ( 1 )  "*"   "*"  " "   " "  " " " " "*"   " "   "*"    " "   " "    "*"
## 9  ( 1 )  "*"   "*"  " "   " "  " " " " "*"   " "   "*"    " "   " "    "*"
## 10  ( 1 ) "*"   "*"  " "   " "  " " " " "*"   " "   "*"    " "   " "    "*"
## 11  ( 1 ) "*"   "*"  " "   " "  " " " " "*"   " "   "*"    "*"   " "    "*"
## 12  ( 1 ) "*"   "*"  " "   "*"  " " " " "*"   " "   "*"    "*"   " "    "*"
## 13  ( 1 ) "*"   "*"  "*"   "*"  " " " " "*"   " "   "*"    "*"   " "    "*"
## 14  ( 1 ) "*"   "*"  "*"   "*"  " " " " "*"   " "   "*"    "*"   "*"    "*"
## 15  ( 1 ) "*"   "*"  "*"   "*"  " " " " "*"   "*"   "*"    "*"   "*"    "*"
## 16  ( 1 ) "*"   "*"  "*"   "*"  "*" "*" "*"   "*"   "*"    "*"   "*"    "*"
##           CRBI CWalks PutOuts Assists Errors
## 1  ( 1 )  "*"  " "    " "     " "     " "
## 2  ( 1 )  "*"  " "    " "     " "     " "
## 3  ( 1 )  "*"  " "    "*"     " "     " "
## 4  ( 1 )  " "  " "    "*"     " "     " "
## 5  ( 1 )  "*"  " "    "*"     " "     " "
## 6  ( 1 )  "*"  " "    "*"     " "     " "
## 7  ( 1 )  "*"  " "    "*"     "*"     " "
## 8  ( 1 )  "*"  " "    "*"     "*"     " "
## 9  ( 1 )  "*"  "*"    "*"     "*"     " "
## 10  ( 1 ) "*"  "*"    "*"     "*"     "*"
## 11  ( 1 ) "*"  "*"    "*"     "*"     "*"
## 12  ( 1 ) "*"  "*"    "*"     "*"     "*"
## 13  ( 1 ) "*"  "*"    "*"     "*"     "*"
## 14  ( 1 ) "*"  "*"    "*"     "*"     "*"
## 15  ( 1 ) "*"  "*"    "*"     "*"     "*"
## 16  ( 1 ) "*"  "*"    "*"     "*"     "*"
```

주어진 결과(summary)로부터 최적 조합에 대한 정보를 확인했다. regsubsets 함수에서는 BIC 척도와 수정 R-squared 를 제공하므로 일단 그 둘을 이용하여 확인해보도록 했다.

```
plot(regsubsets.result, scale = "adjr2", main = "Adjusted R-Squared")
```




```
plot(regsubsets.result, scale = "bic", main = "BIC")
```



수정 결정계수(Adjusted R^2)의 경우 큰 값이 좋은 모형을, BIC의 경우 작은 값이 좋은 모형을 지시하므로, 각각의 플롯에서 가장 위에 위치한 조합이 각각의 기준에서 가장 좋은 모형이며, 두 가지 기준에서 최적 모형은 같다고 볼 수 있다.

즉, AtBat, Hits, Walks, CAtBat, CRuns, CRBI, CWalks, PutOuts, Assists 총 9개 변수를 고려한 모형이 Salary 변수에 대해 두 가지 척도에서 가장 좋은 모형으로 평가 받고 있다.

본 분석에서는 또 다른 척도인 AIC 를 이용한 Stepwise Method 에서도 동일한 결과를 얻는지 확인하기 위해 아래와 같은 함수를 호출했다.

```
model0 <- lm(training_set$Salary~., data=training_set)
## 명료한 보고서 작성을 위해 step 함수의 트레이스 기능은 FALSE 로 설정했다.
steps <- step(model0, direction = "both", trace = FALSE)
summary(steps)
```

```
##
## Call:
## lm(formula = training_set$Salary ~ AtBat + Hits + Walks + CAtBat +
##     CRuns + CRBI + CWalks + PutOuts + Assists, data = training_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -726.14 -186.11  -32.09  120.92 1907.27
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 109.34579   74.67078   1.464 0.144745
## AtBat        -2.99752    0.69589  -4.307 2.65e-05 ***
## Hits          9.53034    2.12438   4.486 1.25e-05 ***
## Walks         5.86393    1.84417   3.180 0.001722 **
## CAtBat       -0.22812    0.06068  -3.759 0.000227 ***
## CRuns         1.73979    0.42725   4.072 6.83e-05 ***
## CRBI          1.07924    0.24627   4.382 1.94e-05 ***
## CWalks       -0.71075    0.29594  -2.402 0.017283 *
## PutOuts       0.41043    0.09442   4.347 2.25e-05 ***
## Assists       0.56059    0.19229   2.915 0.003980 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 313.8 on 190 degrees of freedom
## Multiple R-squared:  0.5824, Adjusted R-squared:  0.5626
## F-statistic: 29.44 on 9 and 190 DF,  p-value: < 2.2e-16
```

위의 결과에서도 볼 수 있듯이 BIC, Adjusted R^2에서 최적 모형이었던 모형이 AIC 척도를 이용한 stepwise method 에서도 가장 적합한 모형으로 뽑혔음을 알 수 있다.

Additional Work)

추가적으로, 데이터 표준화를 수행에 대한 논의가 있을 수 있어 다음과 같은 명령을 추가했다. 주어진 데이터를 보면, 각각의 변수에 해당하는 관측치들의 스케일에 상당한 차이가 있음을 확인할 수 있다. 앞에 C가 붙는 CAtBat 등과 같은 커리어(Career) 전체로부터의 기록들은 선수의 누적 기록이므로, 선수의 연차에 따라 1986년 기록에 해당하는 변수에 비해 10배 이상 차이가 나기도 했다. 따라서 주어진 데이터에 대해 표준화를 수행하여 AIC 척도 기준의 Stepwise Method 를 통해 최적 모형을 탐색해보았다.

```
standard_training_set <- scale(training_set)
standard_training_set <- as.data.frame(standard_training_set)
model_standard <- lm(standard_training_set$Salary~., data=standard_training_set)
## 명료한 보고서 작성을 위해 step 함수의 트레이스 기능은 FALSE 로 설정했다.
steps <- step(model_standard, direction = "both", trace = FALSE)
summary(steps)
```

```
##
## Call:
## lm(formula = standard_training_set$Salary ~ AtBat + Hits + Walks +
##     CAtBat + CRuns + CRBI + CWalks + PutOuts + Assists, data = standard_training_set)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5303 -0.3922 -0.0676  0.2548  4.0194
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.184e-19  4.677e-02   0.000 1.000000
## AtBat       -9.093e-01  2.111e-01  -4.307 2.65e-05 ***
## Hits         8.842e-01  1.971e-01   4.486 1.25e-05 ***
## Walks        2.646e-01  8.323e-02   3.180 0.001722 **
## CAtBat      -1.128e+00  3.000e-01  -3.759 0.000227 ***
## CRuns        1.246e+00  3.060e-01   4.072 6.83e-05 ***
## CRBI         7.423e-01  1.694e-01   4.382 1.94e-05 ***
## CWalks      -4.087e-01  1.702e-01  -2.402 0.017283 *
## PutOuts      2.214e-01  5.092e-02   4.347 2.25e-05 ***
## Assists      1.577e-01  5.408e-02   2.915 0.003980 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6614 on 190 degrees of freedom
## Multiple R-squared:  0.5824, Adjusted R-squared:  0.5626
## F-statistic: 29.44 on 9 and 190 DF,  p-value: < 2.2e-16
```

표준화 전과 마찬가지로 "Salary ~ AtBat + Hits + Walks + CAtBat + CRuns + CRBI + CWalks + PutOuts + Assists" 모형을 얻을 수 있는 것을 확인할 수 있다.

Question 2-2. 학습된 모형을 가지고 테스트 집합을 예측하고, RMSE 오차를 계산하시오.

Answer)

```
model0 <- lm(training_set$Salary ~ AtBat + Hits + Walks + CAtBat + CRuns + CRBI
             + CWalks + PutOuts + Assists, data=training_set)
#RMSE of model0 on Test Partition
sqrt( mean( (predict(model0, newdata=test_set)-test_set$Salary)^2 ) )
```

```
## [1] 345.8279
```

Question 2-3. 가장 잘 예측하지 않는 임의의 다른 변수조합 2개를 가지고 모형을 생성하고, 테스트 집합의 RMSE 오차를 계산하시오.

Answer)

문제 2-1에서 확인한 수정 결정계수(Adjusted R^2) 척도에서 2번째와 3번째로 좋은 모형으로 파악되었던 모형을 바탕으로 진행했다.

```
model1 <- lm(training_set$Salary ~ AtBat + Hits + Walks + CAtBat + CRuns + CRBI
             + CWalks + PutOuts + Assists + Errors, data=training_set)
#RMSE of model1 on Test Partition
sqrt( mean( (predict(model1, newdata=test_set)-test_set$Salary)^2 ) )
```

```
## [1] 347.3018
```

```
model2 <- lm(training_set$Salary ~ AtBat + Hits + Walks + CAtBat + CHits + CRuns + CRBI
             + CWalks + PutOuts + Assists + Errors, data=training_set)
#RMSE of model2 on Test Partition
sqrt( mean( (predict(model2, newdata=test_set)-test_set$Salary)^2 ) )
```

```
## [1] 350.7295
```

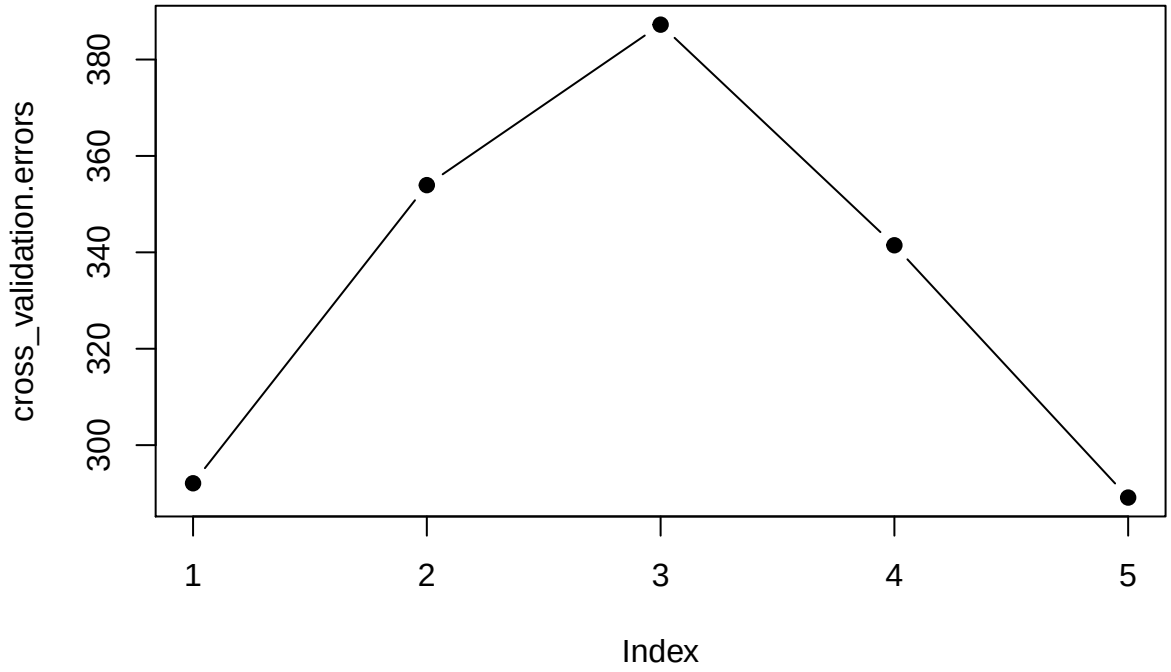
Question 2-4. 1번에서 찾은 변수조합을 활용하여 전체 데이터를 가지고 5겹 교차 검증을 수행하라. RMSE로 오차를 계산하고, 5겹 교차검증의 평균 오차를 보고하라.

Answer)

```
## 교차 검증 문제에는 수업시간에 배운 Manual Cross Validation 을 사용했다.
cross_validation.errors <- c(rep(0,5))
folds <- sample(rep(1:5, length=nrow(hitters_omit)))

for(k in 1:5){
  model <- lm(Salary ~ AtBat + Hits + Walks + CAtBat + CRuns
              + CRBI + CWalks + PutOuts + Assists, data = hitters_omit[folds!=k,])
  pred <- predict(model, hitters_omit[folds==k,])
  cross_validation.errors[k] <- sqrt(mean( (hitters_omit$Salary[folds==k] - pred)^2))
}
```

5회의 교차검증의 오차(RMSE)를 그래프를 이용하여 보고하면 다음과 같다.

```
plot(cross_validation.errors, pch=19, type="b")
```



최종적으로 5겹 교차검증의 평균을 보고하면 다음과 같다.

```
mean(cross_validation.errors)
```

```
## [1] 332.7706
```