EDA 8장 과제

주의사항을 숙지하였고 모든 책임을 지겠습니다.

2019122041 송유진

[8장 평활법 SUMMARY]

Smoothing은 시계열 데이터를 부드럽게 만들어서 데이터의 패턴을 살펴보는 것을 유용하게 도와주는 방법론

3 : 3개 자료의 median으로 smoothing을 하는 것
3R : 더 이상 자료가 바뀌지 않을 때까지 이를 반복하는 것
3RSS : 3R + splitting 두 번 반복
3RSSH : 3RSS + Hanning
3R : median으로 raw data를 smoothing하는 것
42 : raw data에 없는 값으로 smoothing 되어 더 부드럽게 평활
: 4253H는 3RSSH보다 더 부드럽게 평활된다는 장점이 있음

1. FRIDAY.DAT 자료는 영국에서 1986년 금요일에 발생한 자동차 사고 사망자 수이다. 첫 열은 하루를 24시간으로 표시한 것이고, 두 번째 열은 사망자 수이다. 시간에 따른 사망자 수를 평활하고 시간과 사망자 수의 관계를 서술하여라. (자료에는 나타나 있지 않으나 음주 운전을 염두에 두고 분석해 보면?)

```
setwd("D:\\2022-1(3-2)\\2022-01_탐자분\\Data")
friday <- read.csv("8장 FRIDAY.DAT", head=F, sep='\t')
head(friday)
tail(friday)
names(friday) <- c("time","death")
friday
attach(friday)
```

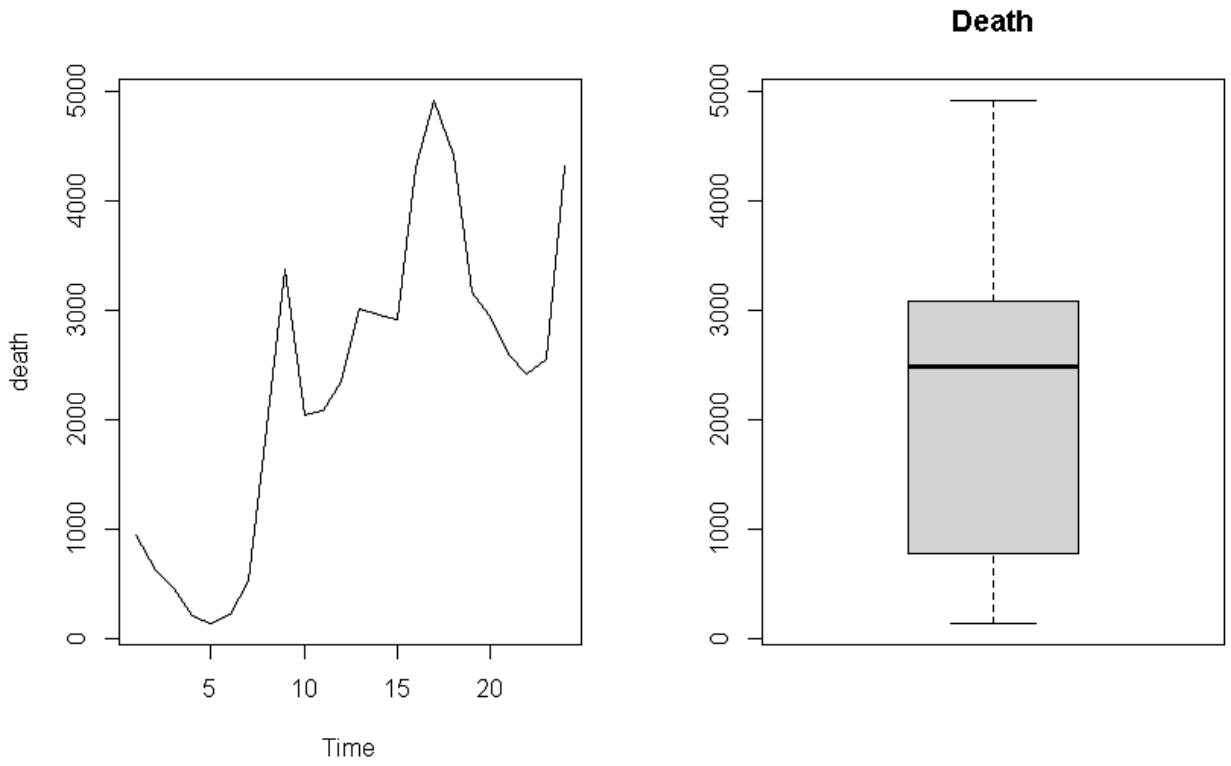
상단의 코드를 실행하여 아래와 같이 데이터프레임 형태를 정제해주었다.

```
> friday
    time death
1    0-1   938
2    1-2   621
3    2-3   455
4    3-4   207
5    4-5   138
6    5-6   215
7    6-7   526
8    7-8  1933
9    8-9  3377
10  9-10  2045
11 10-11  2078
12 11-12  2351
13 12-13  3015
14 13-14  2966
15 14-15  2912
16 15-16  4305
17 16-17  4923
18 17-18  4427
19 18-19  3164
20 19-20  2950
21 20-21  2601
22 21-22  2420
23 22-23  2557
24  23-0  4319
```

```
> summary(death)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  138.0   858.8  2488.5  2310.1  3052.2  4923.0
```

Friday 데이터의 사망자 수를 summary를 통해 살펴본 결과 최소값은 138, 중앙값은 2488.5, 최대값은 4923임을 알 수 있다. 시간의 흐름에 따른 사망자 수의 변화를 살펴보기 위해 time series plot과 boxplot을 그려보았다.

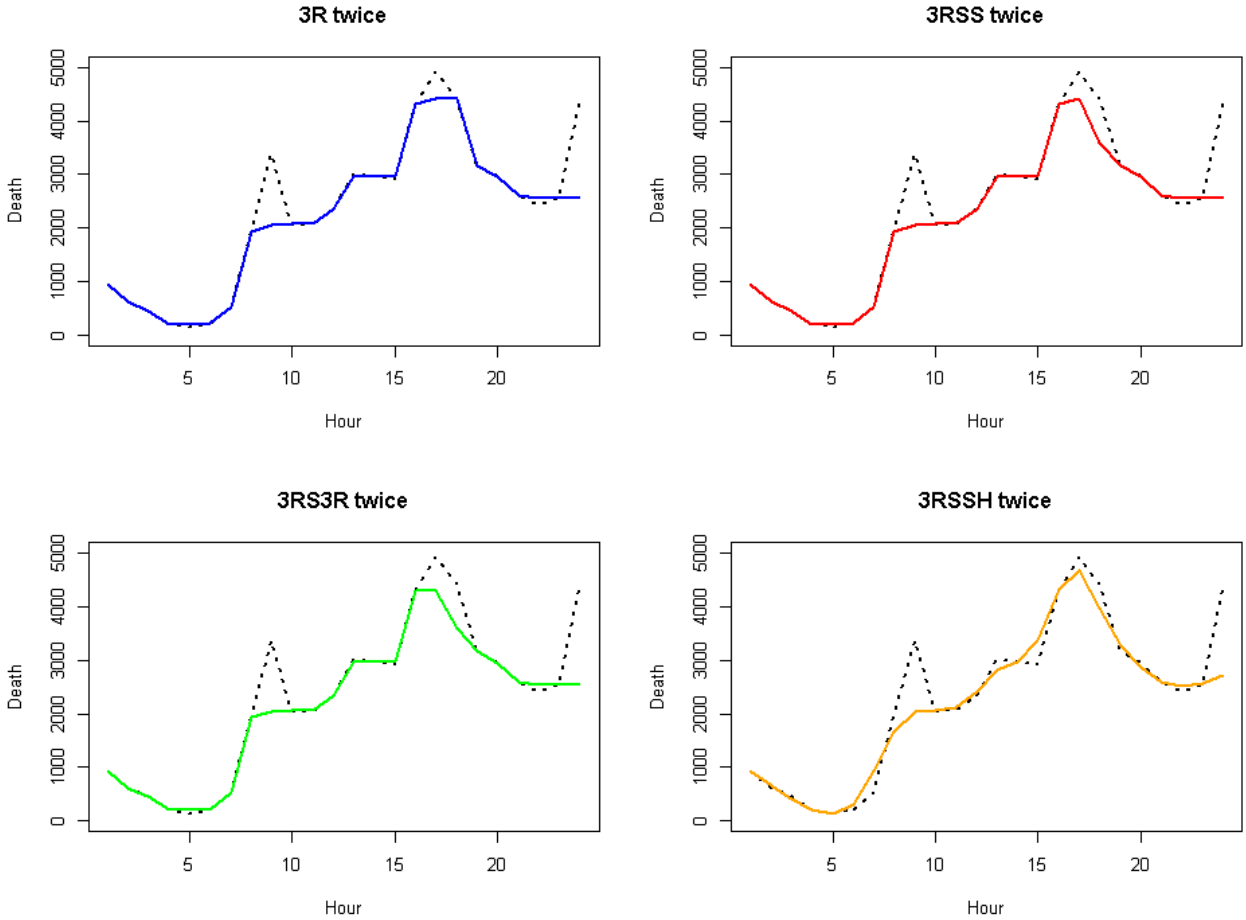
```
par(mfrow=c(1,2))
ts.plot(death)
boxplot(death,main="Death")
```



0~1시부터 07시까지는 1000명 이하로 비교적 사망자 수가 적었다. 그러나, 08~10시에 3000명 이상으로 사망자 수가 급등한다는 것을 알 수 있다. 이후로 2000명 대로 감소했다가 15~17시에 5000명대로 다시 증가한다. 또한, 22시까지 2500명대로 감소했다가 22~24시에 4000명으로 급등했다.
해당 결과를 해석해본 결과, 새벽 시간(0~7시)에는 통행량 자체가 적기 때문에 다른 시간대보다 적은 사망자 수가 기록된 것임을 유추할 수 있다. 이후 아침 시간 8시 이후부터 통행자 수 증가에 따라 사망자 수 역시 증가하고 있다. 이후 사망자수는 약간 감소했다가 오후, 저녁시간(15~18시)에 사망자 수가 다시 급등하고 있다. 이후 23~0시에는 사망자 수가 다시 증가했는데 이는 음주운전의 영향을 고려해볼 수 있다. 잔차를 제거한 데이터의 pattern을 살펴보기 위해 하단의 코드를 활용하여 다양한 평활법을 시도해보았다.

```
par(mfrow=c(2,2))
ts.friday = ts(death)
smooth.f=smooth(ts.friday,kind="3R",twiceit = T)
plot(death,type="l",ylab="Death",lty="dotted",xlab="Hour",ylim=c(0,5000),lwd=2);par(new=T)
plot(smooth.f,xlab="",ylab="",ylim=c(0,5000),col="blue",lwd=2,
main="3R twice")

ts.friday = ts(death)
smooth.f=smooth(ts.friday,kind="3RSS",twiceit = T)
plot(death,type="l",ylab="Death",lty="dotted",xlab="Hour",ylim=c(0,5000),lwd=2);par(new=T)
plot(smooth.f,xlab="",ylab="",ylim=c(0,5000),col="red",lwd=2,
      main="3RSS twice")

ts.friday = ts(death)
smooth.f=smooth(ts.friday,kind="3RS3R",twiceit = T)
plot(death,type="l",ylab="Death",lty="dotted",xlab="Hour",ylim=c(0,5000),lwd=2);par(new=T)
```

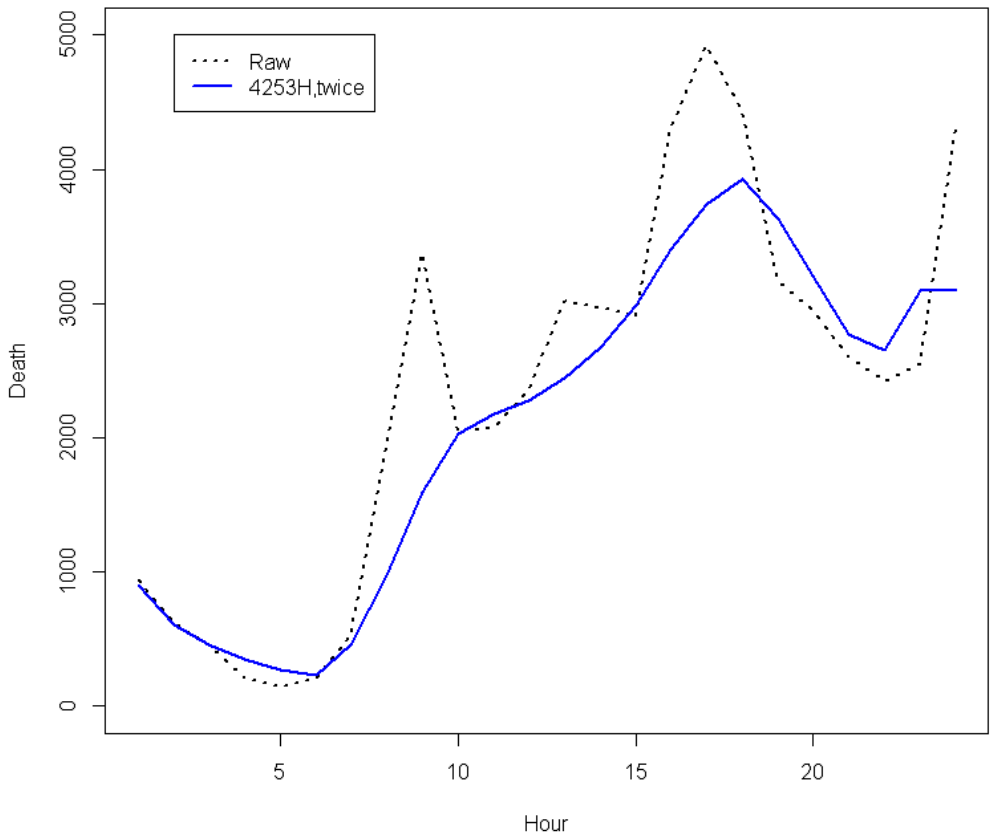
```
plot(smooth.f,xlab="",ylab="",ylim=c(0,5000),col="green",lwd=2,
      main="3RS3R twice")

a3rssh2=function(x){
  n=length(x)
  x3rss=smooth(x,kind="3RSS")
  x3rssh <- vector("numeric",n)
  x3rssh2 <- vector("numeric",n)
  for (i in 2:(n-1)) x3rssh[i] <- x3rss[i-1]/4 + x3rss[i]/2 + x3rss[i+1]/4
  x3rssh[1] <- x3rss[1]; x3rssh[n] <- x3rss[n]
  rough=x-x3rssh
  x3rss2=smooth(rough,kind="3RSS")
  for (i in 2:(n-1)) x3rssh2[i] <- x3rss2[i-1]/4 + x3rss2[i]/2 + x3rss2[i+1]/4
  x3rssh2[1] <- x3rss2[1]; x3rssh2[n] <- x3rss2[n]
  end=x3rssh+x3rssh2
  return(end)
}

h2=a3rssh2(death)
plot(death,type="l",ylab="Death",lty="dotted",xlab="Hour",ylim=c(0,5000),lwd=2);par(new=T)
plot(h2,type="l",ylab="Death", xlab="Hour",ylim=c(0,5000),col="orange", lwd=2,
      main="3RSSH twice")
```



상단의 코드를 통해 출력한 plot을 해석해보면 3~방법은 raw data의 median을 이용한 것이기 때문에 그래프가 전체적으로 rough하다는 것을 알 수 있다. 3RSSH는 splitting 후에 hanning을 수행하는 방법인데 다른 방법들보다 더 부드러운 plot 형태를 확인할 수 있다. 전체적으로 3R~방법은 23-24시에 사망자 수가 증가하는 추세를 반영하지 못하고 있음을 확인할 수 있었다.

따라서, 하단의 코드를 활용해 4253H, twice방법을 시도해보았다.

```
library(sleekts)
h4253=sleek(friday$death)
plot(friday$death,type="l",ylab="Death",lty="dotted",xlab="Hour",ylim=c(0,5000),lwd=2);par(new=T)
plot(h4253,type="l",ylab="Death",xlab="Hour",ylim=c(0,5000),col="blue",lwd=2)
legend(x = 2, y = 5000, c("Raw", "4253H,twice"), col = c("black","blue"), lty=c(3,1),lwd=2)
```



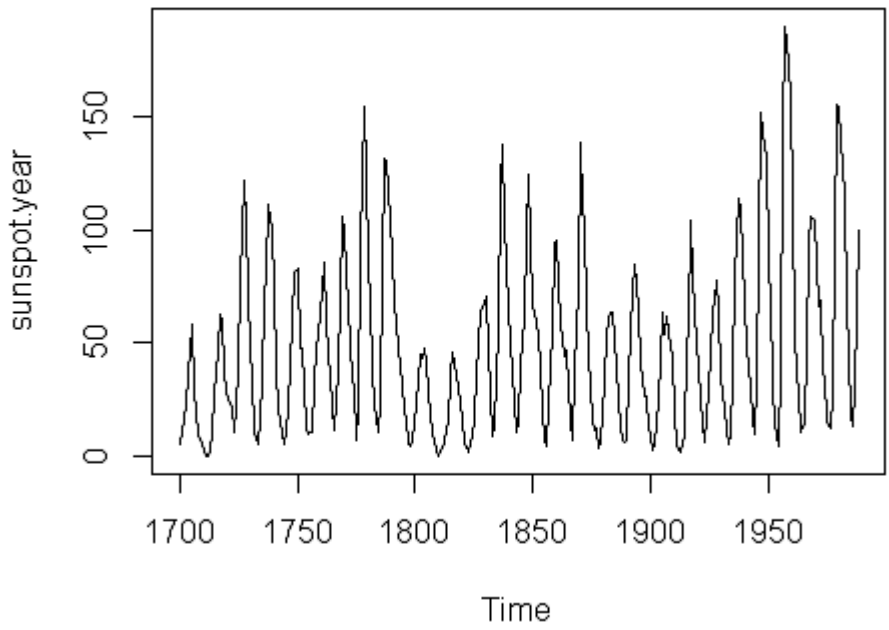
그 결과, 17시 peak값이 1000명 정도 줄었으며 9~10시의 peak도 1000명 이상 감소하여 peak의 양상이 거의 사라지게 되었음을 알 수 있다.
4253H 방법이 3R 계열의 방법과 비교했을 때 22시 이후의 추세를 어느정도 더 잘 반영하고 있음을 알 수 있다. 또한, 곡선의 형태를 보이고 있기 때문에 3R~ 평활법에 비해 추세를 더 쉽게 확인할 수 있다.

2. R의 datasets에 있는 sunspot.year 자료를 평활하고 그 패턴을 기술하라.

```
> length(sunspot.year)
[1] 289
> summary(sunspot.year)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   15.60   39.00   48.61   68.90  190.20
```

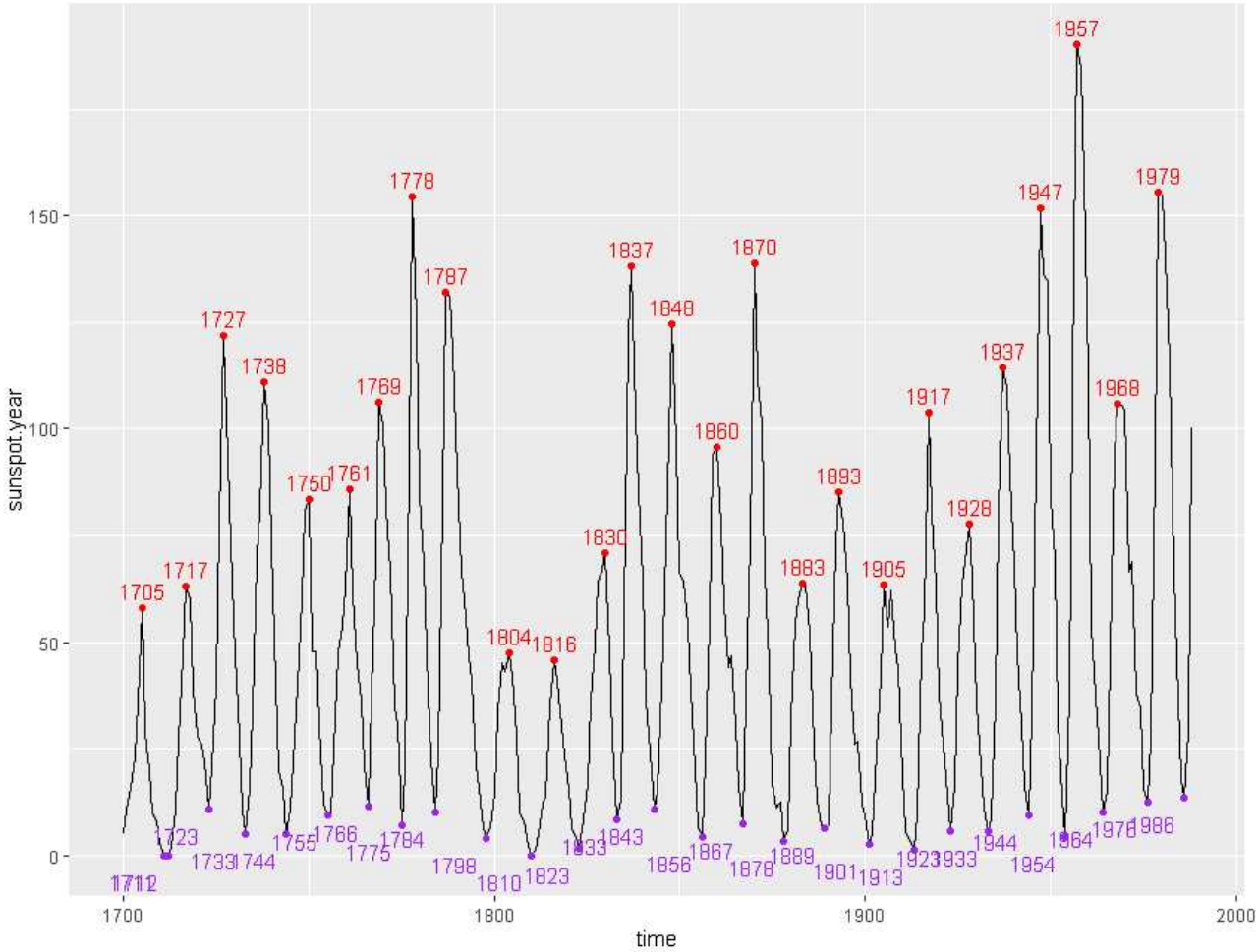
Sunspot.year는 1700년부터 1988년까지의 태양흑점 개수를 기록한 시계열 데이터이다.
대략적인 시계열 데이터의 흐름을 살펴보기 위해 plot을 그려보았다.

```
plot(sunspot.year)
```



흑점이 개수가 많아지고 적어지는 것을 반복하는 '주기'가 있는 데이터라는 사실을 한 눈에 살펴볼 수 있었다. 이를 더 자세히 살펴보기 위해서 peak와 valley의 연도를 plot에 표시해보았다.

```
library(ggplot2)
library(ggpmisc)
library(ggrepel)
library(broom)
ggplot(sunspot.year, as.numeric=F)+geom_line()+stat_peaks(col =
'red')+stat_peaks(geom='text',colour='red',vjust=-0.5,
x.label.fmt="%Y")+stat_valleys(col='purple')+stat_valleys(geom="text",colour='purple',vjust=2,
hjust=1.2, x.label.fmt="%Y")
```

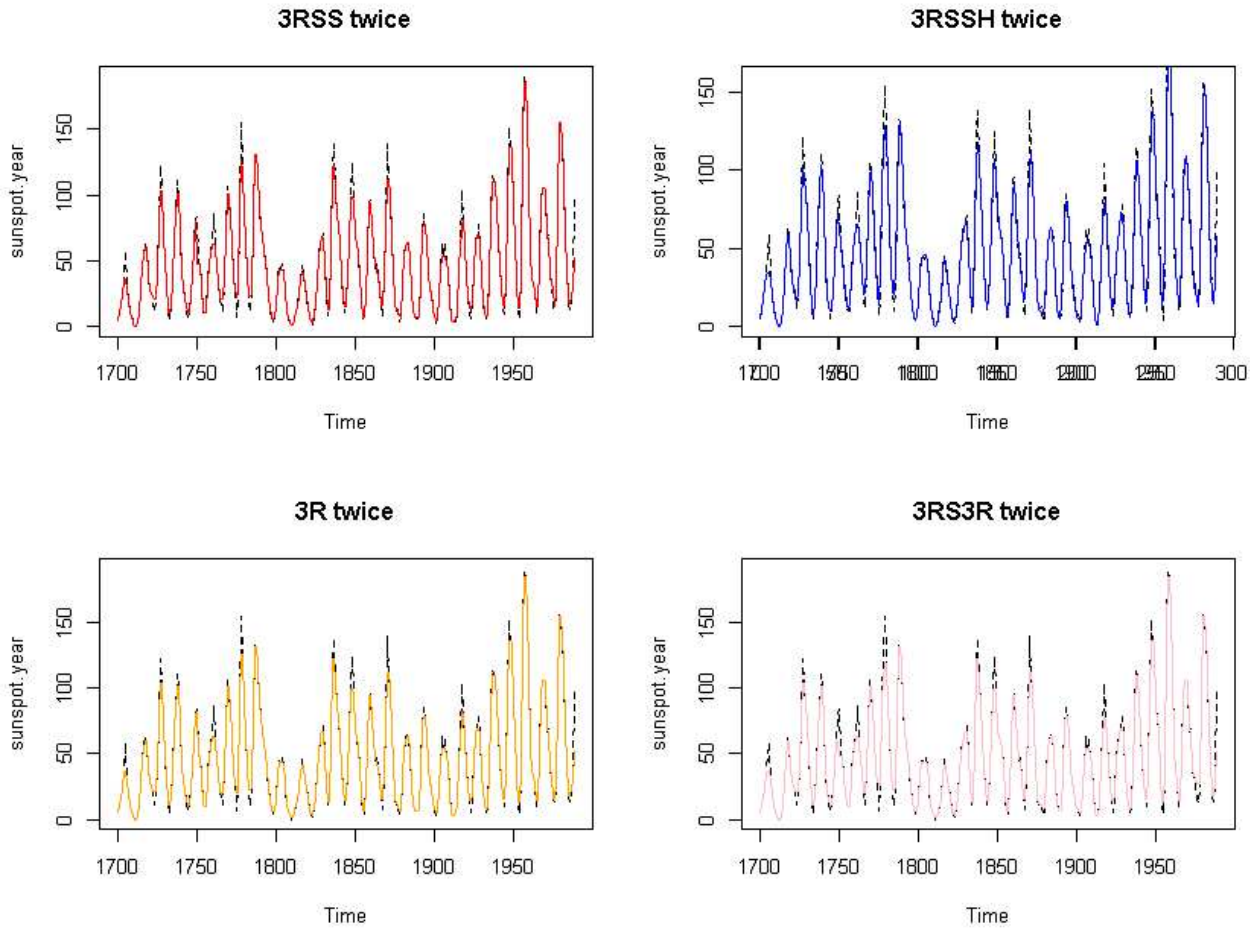


그 결과, 흑점의 개수가 peak인 지점은 대략 10년을 주기로 나타나고 있고, valley 지점 또한 대략 10년 주기로 나타나고 있다. 어느정도의 오차가 있지만 흑점은 대략 5년을 주기로 peak 그리고 valley를 반복하고 있다. 1957년에 흑점이 제일 많았고, 이때 흑점의 개수는 190.2개이다. 흑점이 가장 적었던 연도는 1711~1712년, 1810년으로 흑점의 개수는 0개임을 알 수 있다.

```
par(mfrow=c(2,2))
plot(sunspot.year, type='l', lty='dashed', twiceit = T, main="3RSS twice")
lines(smooth(sunspot.year, kind='3RSS'), col='red')

a3rssh2=function(x){
  n=length(x)
  x3rss=smooth(x,kind="3RSS")
  x3rssh <- vector("numeric",n)
  x3rssh2 <- vector("numeric",n)
  for (i in 2:(n-1)) x3rssh[i] <- x3rss[i-1]/4 + x3rss[i]/2 + x3rss[i+1]/4
  x3rssh[1] <- x3rss[1]; x3rssh[n] <- x3rss[n]
  rough=x-x3rssh
  x3rss2=smooth(rough,kind="3RSS")
  for (i in 2:(n-1)) x3rssh2[i] <- x3rss2[i-1]/4 + x3rss2[i]/2 + x3rss2[i+1]/4
  x3rssh2[1] <- x3rss2[1]; x3rssh2[n] <- x3rss2[n]
  end=x3rssh+x3rssh2
```

```
  return(end)
}
h2=a3rssh2(sunspot.year)
plot(sunspot.year, type='l', ylim=c(0,160), lty='dashed',xlab="Time", main="3RSSH twice",
ylab='sunspot.year');par(new=T)
plot(h2,type="l", xlab="Time", col='blue', ylim=c(0,160), ylab='sunspot.year')

plot(sunspot.year, type='l', twiceit = T, lty='dashed', main="3R twice")
lines(smooth(sunspot.year, kind='3R'), col='orange')

plot(sunspot.year, type='l', twiceit = T, lty='dashed', main="3RS3R twice")
lines(smooth(sunspot.year, kind='3RS3R'), col='pink')
```

상단의 코드를 통해 평활법을 적용해보았다. 적용해본 방법은 3RSS, 3RSSH, 3R 그리고 3RS3R으로서 twiceit=T로 인자를 지정해주었다.

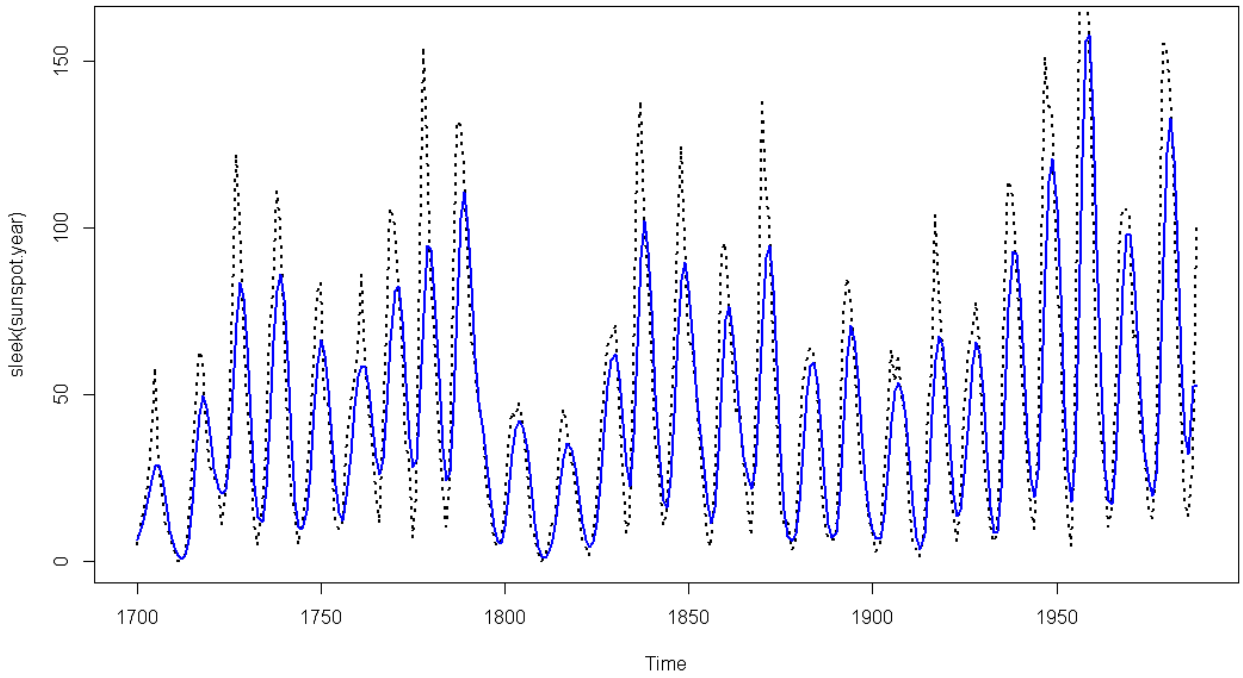


대체적으로 4가지 방법 모두 유사한 양상을 띠고 있음을 확인했다.
다만, 3가지 방법에 비해 3RS3R 방법이 좀 더 부드러우며 general한 형태를 보이고 있음을 알 수 있다.

다음으로, sleekts library를 활용해서 4253H 방법으로 평활을 시도해보았다.

```
library(sleekts)
```

```
h4253=sleek(sunspot.year)
plot(sunspot.year,type="l",ylab="sleek(sunspot.year)",xlab="Time",lty="dotted",ylim=c(0,160),lwd=2);par(new=T)
plot(h4253,type="l",ylab="sleek(sunspot.year)",xlab="Time",col="blue",ylim=c(0,160),lwd=2)
```



4253H방법은 raw data를 그대로 사용하지 않고 데이터에 없는 값을 smoothing에 사용하기 때문에 3R 방법보다 비교적 더 부드러운 평활 결과를 보여준다. 상단의 plot은 3R~ 방법론에 비해 훨씬 부드러운 양상을 보이고 있다. 흑점의 개수는 여전히 10년을 주기로 peak 그리고 valley가 반복된다는 것을 볼 수 있다. 그러나 흑점이 많아지고 줄어드는 속도는 시간의 흐름에 따라 계속해서 변화를 반복하고 있다. 1800-1820년대에는 비교적 변화 속도가 느리지만 1950년대 이후에는 흑점 개수의 변화 속도가 더 빨라지고 있음을 확인할 수 있다.

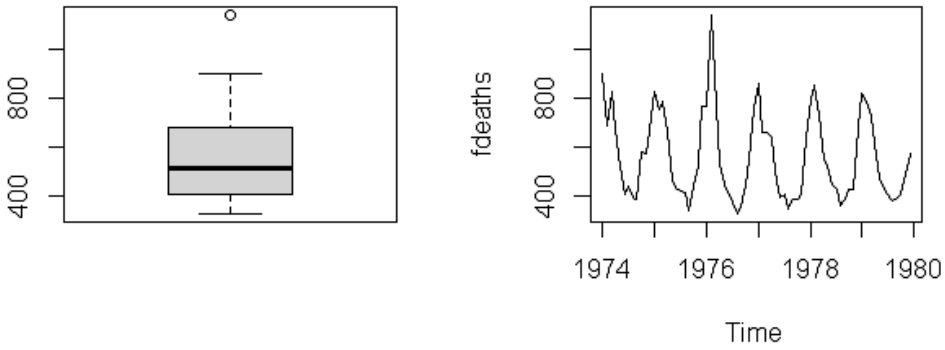
3. R의 fdeaths 자료를 평활하고 분석하여라.

fdeaths 데이터는 1974년부터 1979년까지 영국에서 bronchitis 기관지염, emphysema 폐기종 그리고 asthma 천식으로 사망한 월별 여성의 수를 기록한 시계열 데이터이다.

```
> summary(fdeaths)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  330.0   411.0   512.0   560.7   681.5  1141.0
```

fdeaths를 요약한 결과를 살펴보면 최솟값은 330, 1Q는 411, 중앙값은 512, 3Q는 681이고 최댓값이 1141임을 알 수 있다. 최솟값에서부터 3Q값까지 거의 100명대로 일정하게 증가하지만, 최댓값에서 급격하게 증가했음을 확인할 수 있다.

```
par(mfrow=c(1,2))
boxplot(fdeaths)
ts.plot(fdeaths)
```



boxplot과 시계열 plot을 상단의 코드를 실행하여 살펴보았다. outlier가 하나 나타나고 있음을 알 수 있으며 이는 max값이었던 1141임을 알 수 있다.

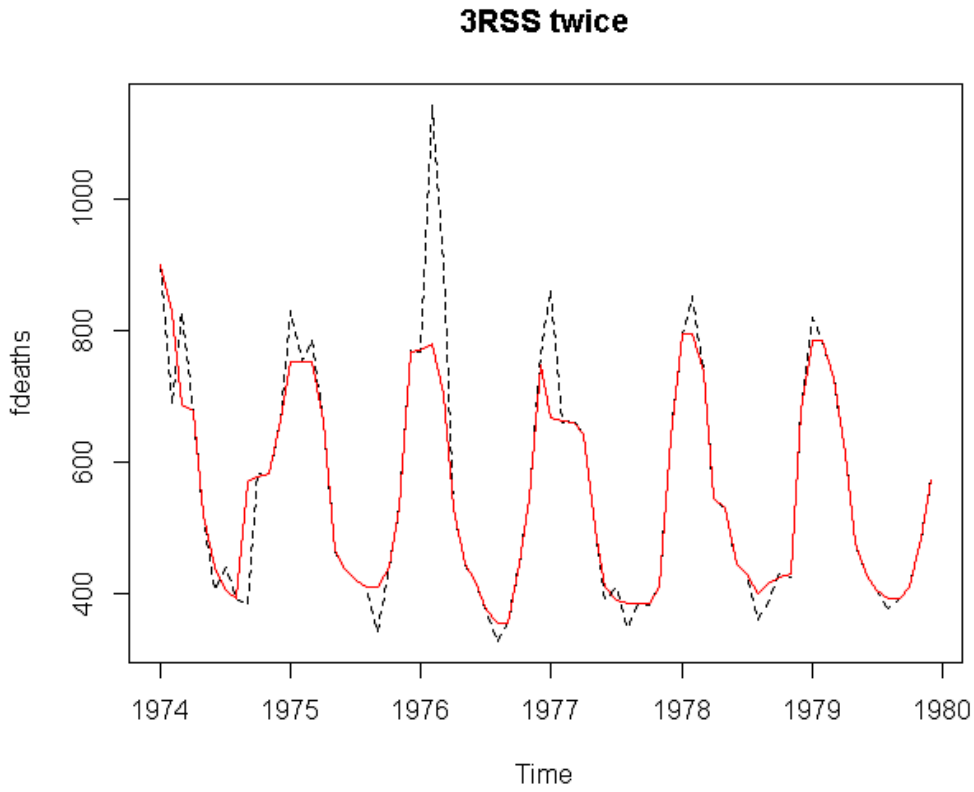
fdeaths의 plot을 살펴보면 일정한 주기를 가지고 순환하는 시계열 데이터임을 알 수 있다.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1974	901	689	827	677	522	406	441	393	387	582	578	666
1975	830	752	785	664	467	438	421	412	343	440	531	771
1976	767	1141	896	532	447	420	376	330	357	445	546	764
1977	862	660	663	643	502	392	411	348	387	385	411	638
1978	796	853	737	546	530	446	431	362	387	430	425	679
1979	821	785	727	612	478	429	405	379	393	411	487	574

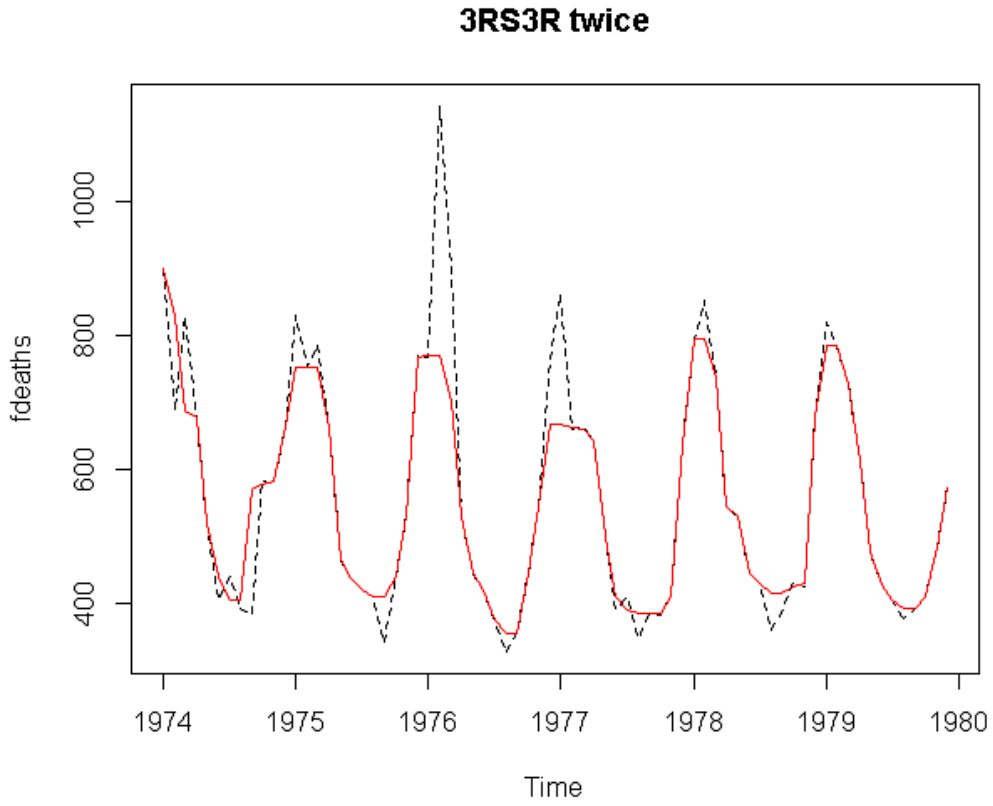
실제로 raw data를 살펴보면 August 8월을 기준으로 사망자가 줄어드는 것을 확인할 수 있다. 즉, 어느 정도의 계절성 또한 엿볼 수 있었다.

다음으로, 평활법을 적용하여 데이터를 살펴보았다.

```
plot(fdeaths, type='l', lty='dashed', twiceit = T, main="3RSS twice")
lines(smooth(fdeaths, kind='3RSS'), col='red')
```



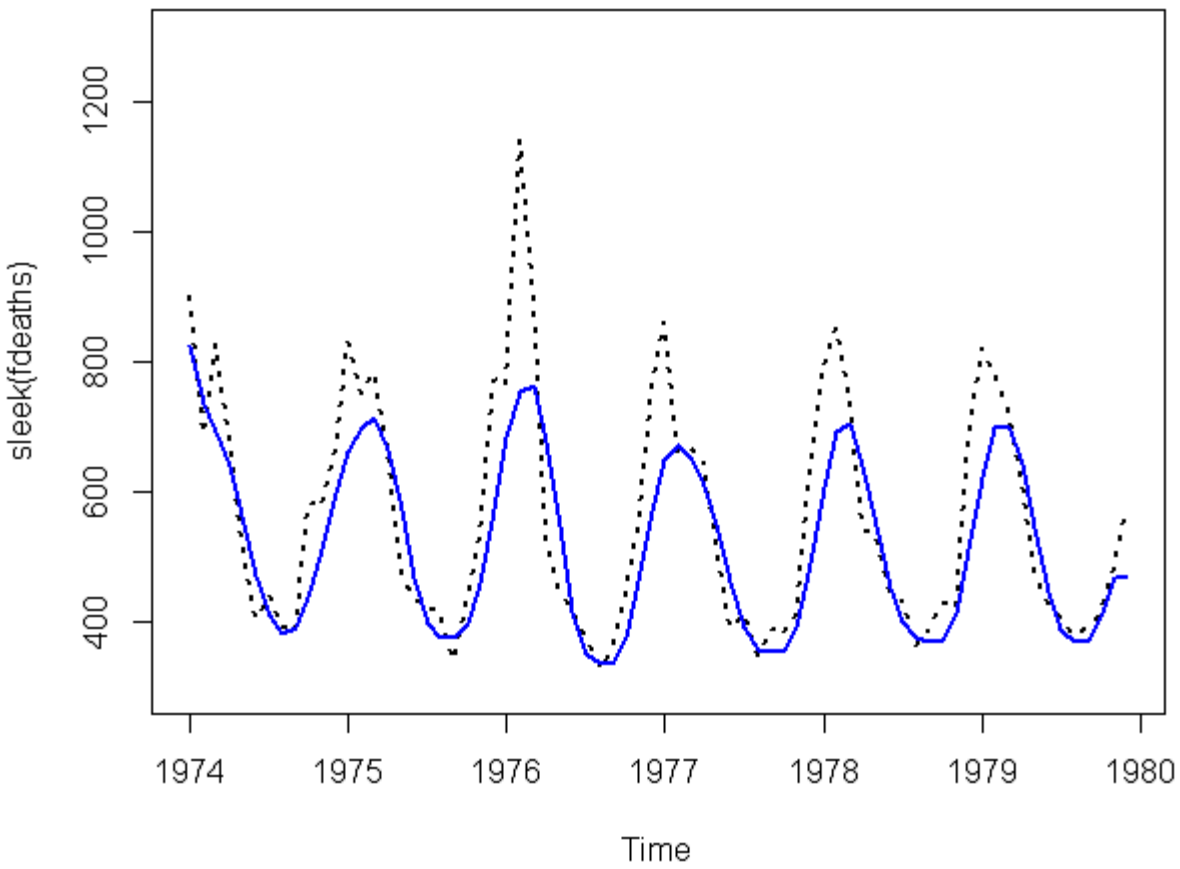

```
plot(fdeaths, type='l', twiceit = T, lty='dashed', main="3RS3R twice")
lines(smooth(fdeaths, kind='3RS3R'), col='red')
```



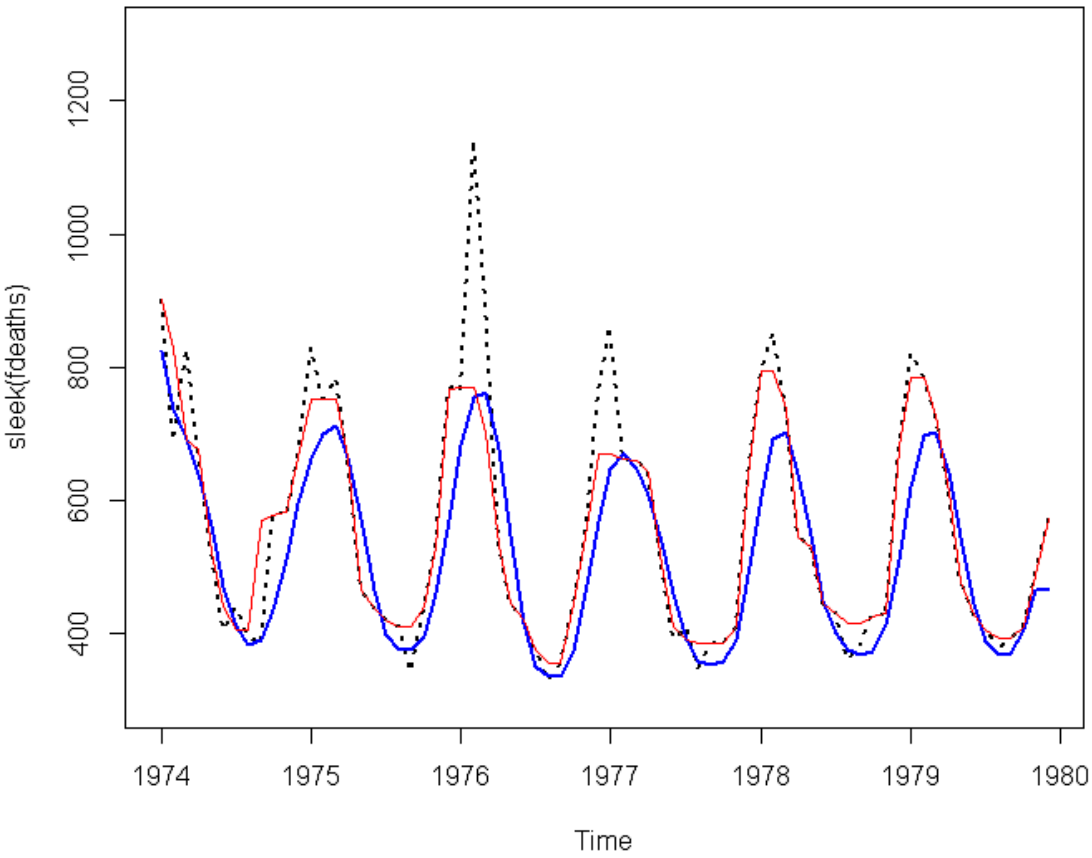
3RSS 방법은 data에 한 번, rough에 한 번 smoothing을 시행한 것이고, 두 번째 방법 3RS3R는 data에

한 번, rough에 한 번 시행한 것이다. 이로 인해 1976년 최댓값 1141(outlier)가 800명대로 감소했으며, raw data의 뾰족했던 peak 또한 부드러워졌음을 알 수 있다. 두 방법 중에서 3RSS이 raw data의 경향성을 조금 더 잘 보여주고 있음을 알 수 있다.
또한, 추운 겨울 1~2월에 사망자 수가 가장 많고, 기온이 올라가는 여름 8~9월에는 사망자 수가 적은 데이터의 경향성을 확인할 수 있었다.
다음으로, h4253 방법도 적용해보았다.

```
library(sleekts)
h4253=sleek(fdeaths)
plot(fdeaths,type="l",ylab="sleek(fdeaths)",xlab="Time",lty="dotted",ylim=c(300,1300),lwd=2);par(new=T)
plot(h4253,type="l",ylab="sleek(fdeaths)",xlab="Time",col="blue",ylim=c(300,1300),lwd=2)
```




```
library(sleekts)
h4253=sleek(fdeaths)
plot(fdeaths,type="l",ylab="sleek(fdeaths)",xlab="Time",lty="dotted",ylim=c(300,1300),lwd=2);par(new=T)
plot(h4253,type="l",ylab="sleek(fdeaths)",xlab="Time",col="blue",ylim=c(300, 1300),lwd=2)
lines(smooth(fdeaths, kind='3RS3R'), col='red')
```



더 나은 비교를 위해 한 번에 plot으로 표현해보았다.

3RSS방법에서는 bottom과 top이 생기며 비교적 rough한 값들이 보인다. 그러나 4253H에서는 raw data를 그대로 이용하지 않고 연산을 한 이후 smoothing을 실시하기 때문에 훨씬 더 smooth한 plot이 보인다. 실제로 1976년에 존재하였던 outlier 1141(명)이 800대로 감소했고 몇몇 peak에서도 200 이상 감소하였다.
4253H에서는 전체적으로 raw data보다 1-2달 정도 delay가 되고 있음을 볼 수 있다. 또한, 전체적으로 3RS3R방법보다 값이 작게 나오는 추세를 보인다. 즉, raw data의 추세를 더 잘 반영하는 것은 3RS3R방법이라는 결론을 내릴 수 있다. 그러나 4253H 방법은 top, bottom 없이 훨씬 부드러운 결과를 보여주고 있으며 3R~방법에 비해 굴곡이 덜해 분석의 용이성이 더 높을 것으로 기대된다.

4. R의 datasets에 있는 jumping.dat

첫 칼럼은 연도, 둘째 칼럼부터 high jump, pole vault, long jump, tripple jump의 네 가지 뛰기 관련 종목에 대한 올림픽 기록이다. 단위 미터. 네 경기 종목에 대한 기록을 각각 평활하여 어느 종목의 기록이 점점 기록 갱신하기 어려워지고 있는지 분석하라. (어느 종목에서 기록 갱신의 상대적 변화가 점점 줄어드는가?)

	high	pole	long	tripple
year	jump	vault	jump	jump
1900	1.90	3.30	7.19	14.43

1904 1.80 3.51 7.34 14.33
1908 1.90 3.71 7.48 14.92
.....

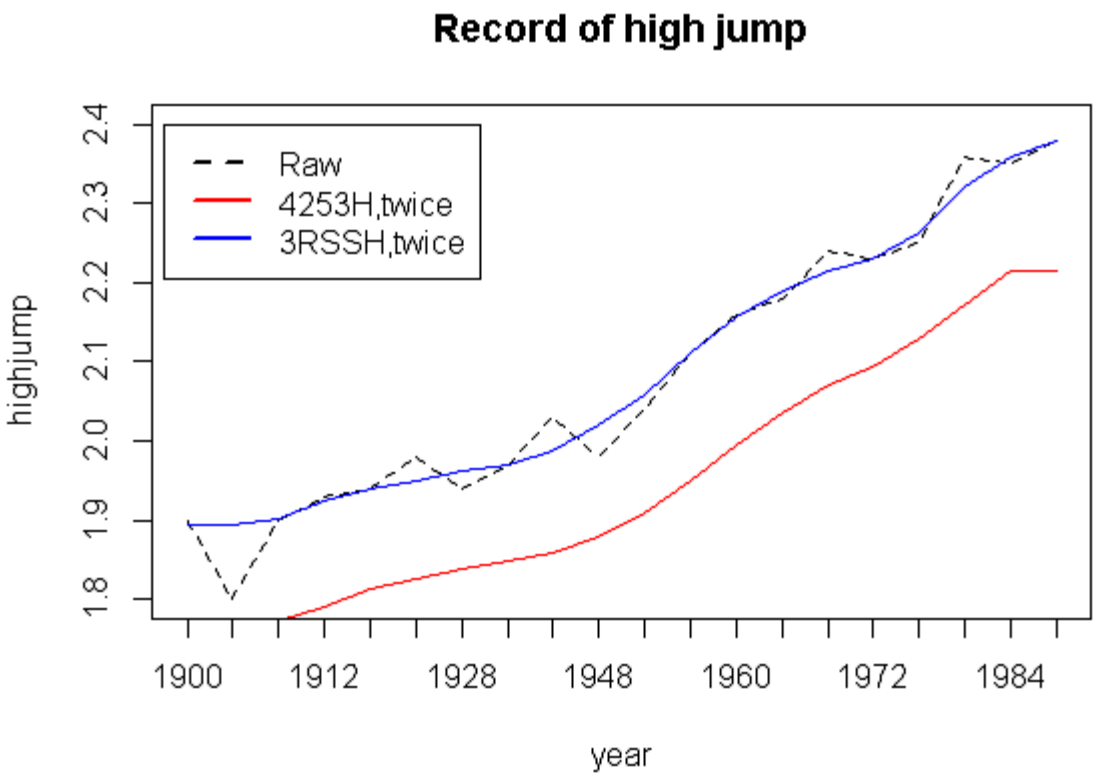
```
> setwd("D:\\2022-1(3-2)\\2022-01_탐자분\\Data")
> jump <- read.csv("JUMPING.DAT", head=F, sep='\t')
> colnames(jump)=c("year","highjump",
+                  "polevault", "longjump", "tripplejump")
> attach(jump)
The following objects are masked from jump (pos = 3):

    highjump, longjump, polevault, tripplejump, year

> jump
   year highjump polevault longjump tripplejump
1  1900     1.90      3.30     7.19       14.43
2  1904     1.80      3.51     7.34       14.33
3  1908     1.90      3.71     7.48       14.92
4  1912     1.93      3.95     7.60       14.76
5  1920     1.94      4.09     7.15       14.50
6  1924     1.98      3.95     7.45       15.53
7  1928     1.94      4.20     7.74       15.21
8  1932     1.97      4.31     7.64       15.72
9  1936     2.03      4.35     8.06       16.00
10 1948     1.98      4.30     7.82       15.40
11 1952     2.04      4.55     7.57       16.22
12 1956     2.11      4.56     7.83       16.34
13 1960     2.16      4.70     8.12       16.81
14 1964     2.18      5.10     8.07       16.85
15 1968     2.24      5.40     8.90       17.39
16 1972     2.23      5.50     8.24       17.35
17 1976     2.25      5.50     8.34       17.29
18 1980     2.36      5.78     8.54       17.35
19 1984     2.35      5.75     8.54       17.56
20 1988     2.38      5.90     8.72       17.61
```

JUMPING.DAT 파일을 불러온 뒤 각 열 이름을 year, high jump, pole vault, long jump, tripple jump와 같이 지정했다.

```
a3rssh2=function(x){
  n=length(x)
  x3rss=smooth(x,kind="3RSS")
  x3rssh <- vector("numeric",n)
  x3rssh2 <- vector("numeric",n)
  for (i in 2:(n-1)) x3rssh[i] <- x3rss[i-1]/4 + x3rss[i]/2 + x3rss[i+1]/4
  x3rssh[1] <- x3rss[1]; x3rssh[n] <- x3rss[n]
  rough=x-x3rssh
  x3rss2=smooth(rough,kind="3RSS")
  for (i in 2:(n-1)) x3rssh2[i] <- x3rss2[i-1]/4 + x3rss2[i]/2 + x3rss2[i+1]/4
  x3rssh2[1] <- x3rss2[1]; x3rssh2[n] <- x3rss2[n]
  end=x3rssh+x3rssh2
  return(end)
}

library(sleekts)
```

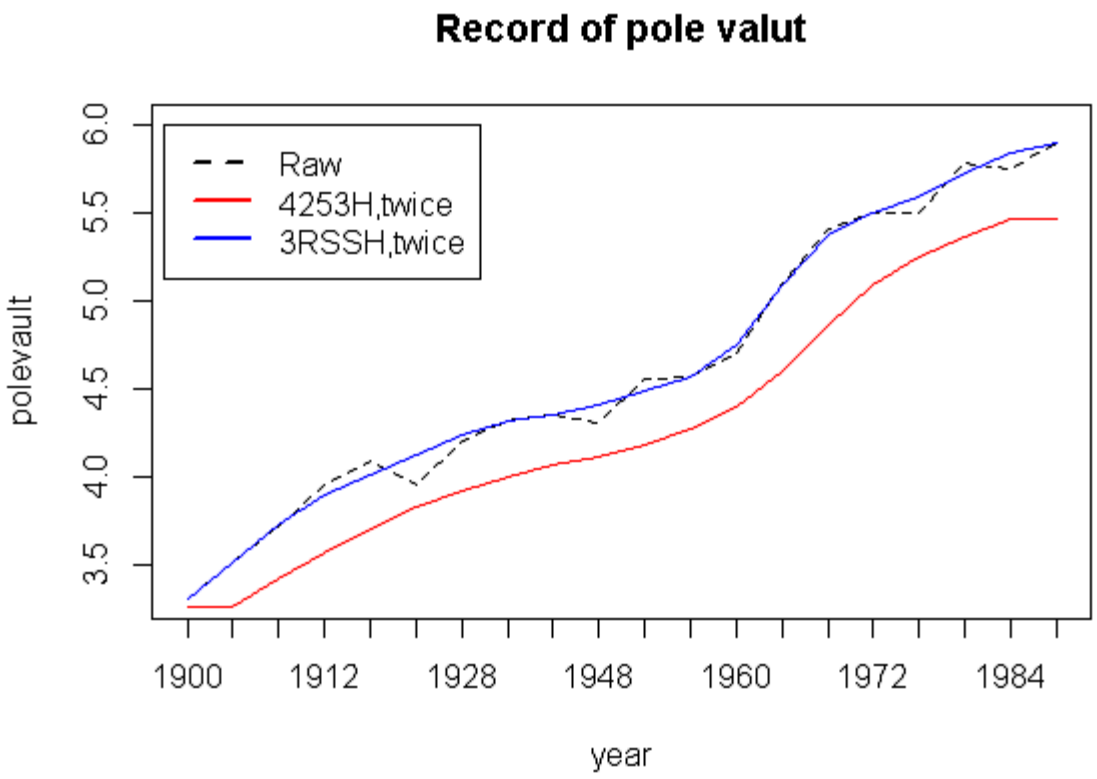
```
plot(highjump,xaxt="n",xlab="year",type="l",ylim=c(1.8,2.4),lty=2,main="Record of high jump");par(new=T)
plot(sleek(highjump),xaxt="n",col="red",type="l",ylim=c(1.8,2.4),ann=F);par(new=T)

h4253=a3rssh2(highjump)

plot(h4253,xaxt="n",xlab="year",type="l",ylim=c(1.8,2.4),col="blue",ann=F)
axis(1,at=1:20,labels=jump$year)
legend(x = 0.5, y = 2.4, c("Raw", "4253H,twice", "3RSSH,twice"),
       col = c("black","red","blue"),
       lty=c(2,1,1),lwd=2)
```

상단의 코드를 실행하여 아래와 같이 high jump plot을 그려주었다.



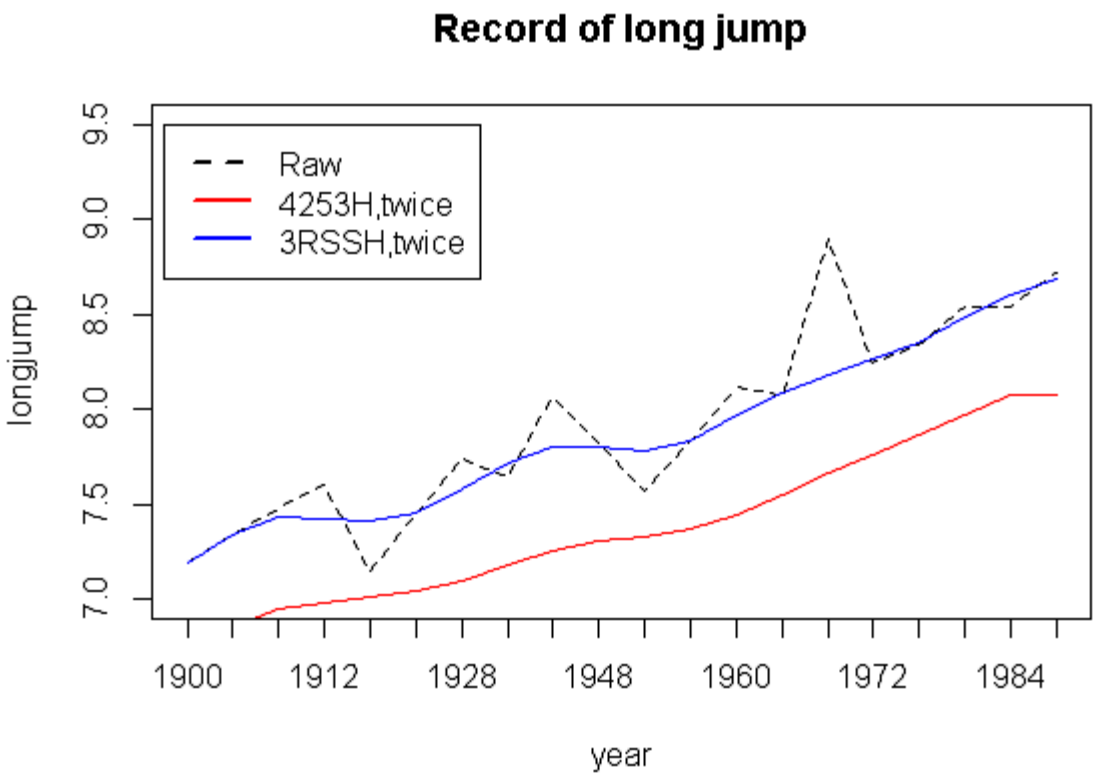

```
plot(polevault,xaxt="n",xlab="year",type="l",lty=2,ylim=c(3.3,6.0),main="Record of pole valut");par(new=T)
plot(sleek(polevault),xaxt="n",col="red",type="l",ylim=c(3.3,6.0),ann=F);par(new=T)
p4253=a3rssh2(jump$polevault)
plot(p4253,xaxt="n",xlab="year",type="l",ylim=c(3.3,6.0),col="blue",ann=F)
legend(x = 0.5, y = 6, c("Raw", "4253H,twice", "3RSSH,twice"),
       col = c("black","red","blue"),
       lty=c(2,1,1),lwd=2)
axis(1,at=1:20,labels=jump$year)
```

상단의 코드를 실행하여 아래와 같이 pole valut plot을 그려주었다.



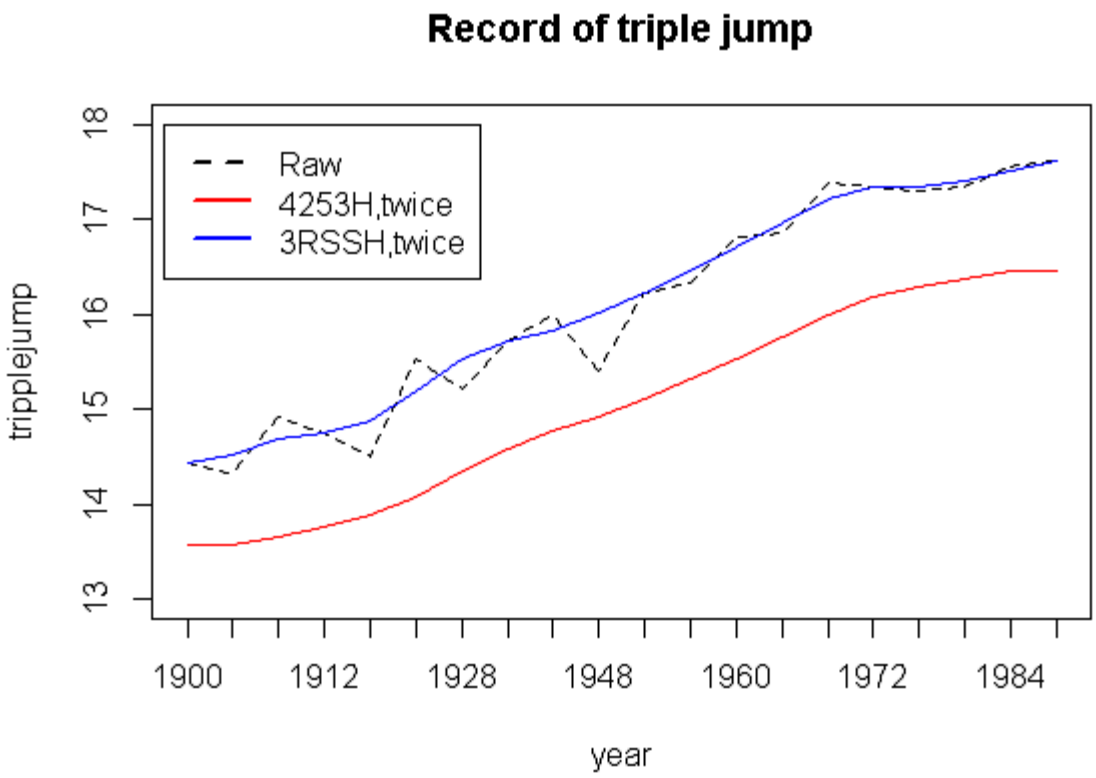

```
plot(longjump,xaxt="n",xlab="year",type="l",lty=2,ylim=c(7,9.5),main="Record of long
 jump");par(new=T)
plot(sleek(longjump),xaxt="n",col="red",type="l",ylim=c(7,9.5),ann=F);par(new=T)
l4253=a3rssh2(jump$longjump)
plot(l4253,xaxt="n",xlab="year",type="l",ylim=c(7,9.5),col="blue",ann=F)
legend(x = 0.5, y = 9.5, c("Raw", "4253H,twice", "3RSSH,twice"),
       col = c("black","red","blue"),
       lty=c(2,1,1),lwd=2)

axis(1,at=1:20,labels=jump$year)
```

상단의 코드를 실행하여 아래와 같이 long jump plot을 그려주었다.




```
plot(tripplejump,xaxt="n",lty=2,xlab="year",type="l",ylim=c(13,18),
main="Record of triple jump");par(new=T)
plot(sleek(tripplejump),xaxt="n",col="red",type="l",ylim=c(13,18),ann=F);par(new=T)
t4253=a3rssh2(jump$tripplejump)
plot(t4253,xaxt="n",xlab="year",type="l",ylim=c(13,18),col="blue",ann=F)
legend(x = 0.5, y = 18, c("Raw", "4253H,twice", "3RSSH,twice"),
       col = c("black","red","blue"),
       lty=c(2,1,1),lwd=2)
axis(1,at=1:20,labels=jump$year)
```

상단의 코드를 실행하여 아래와 같이 triple jump plot을 그려주었다.



4253H, twice & 3RSSH, twice 방법을 활용해 평활법을 적용해본 결과, 4가지의 모든 종목에서 시간의 흐름에 따라 기록이 상향하고 있음을 알 수 있었다.

해당 문제에서는 “어느 종목의 기록이 점점 기록 갱신하기 어려워지고 있는지 분석하라. (어느 종목에서 기록 갱신의 상대적 변화가 점점 줄어드는가?)”를 요구하고 있다. 따라서, 4개의 종목을 함께 비교 분석하기 위해서는 표준화가 필수적이다.

```
> fivenum(highjump)
[1] 1.800 1.940 2.035 2.235 2.380
> fivenum(tripplejump)
[1] 14.330 15.065 16.110 17.320 17.610
```

상단의 결과를 통해 highjump와 tripplejump의 기록 범위의 정도는 다르다는 것을 확인할 수 있다.

```
stand= function(x){
  std=(x-median(x))/(fivenum(x)[4]-fivenum(x)[2])
  return(std)
}
highjump=stand(jump$highjump)
polevault=stand(jump$polevault)
longjump=stand(jump$longjump)
```

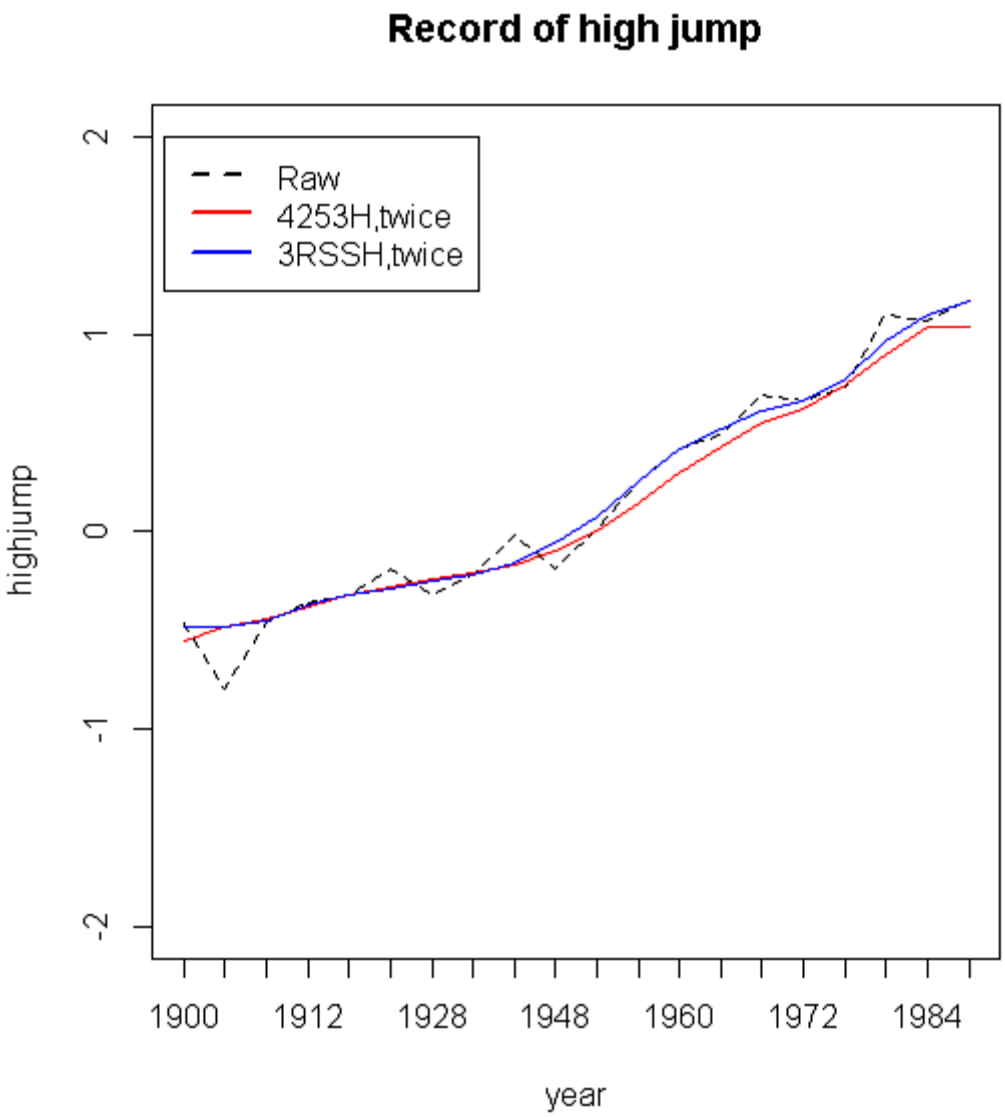
```
tripplejump=stand(jump$tripplejump)
```

상단의 코드를 통해 아래와 같이 4가지 종목을 표준화해주었다.

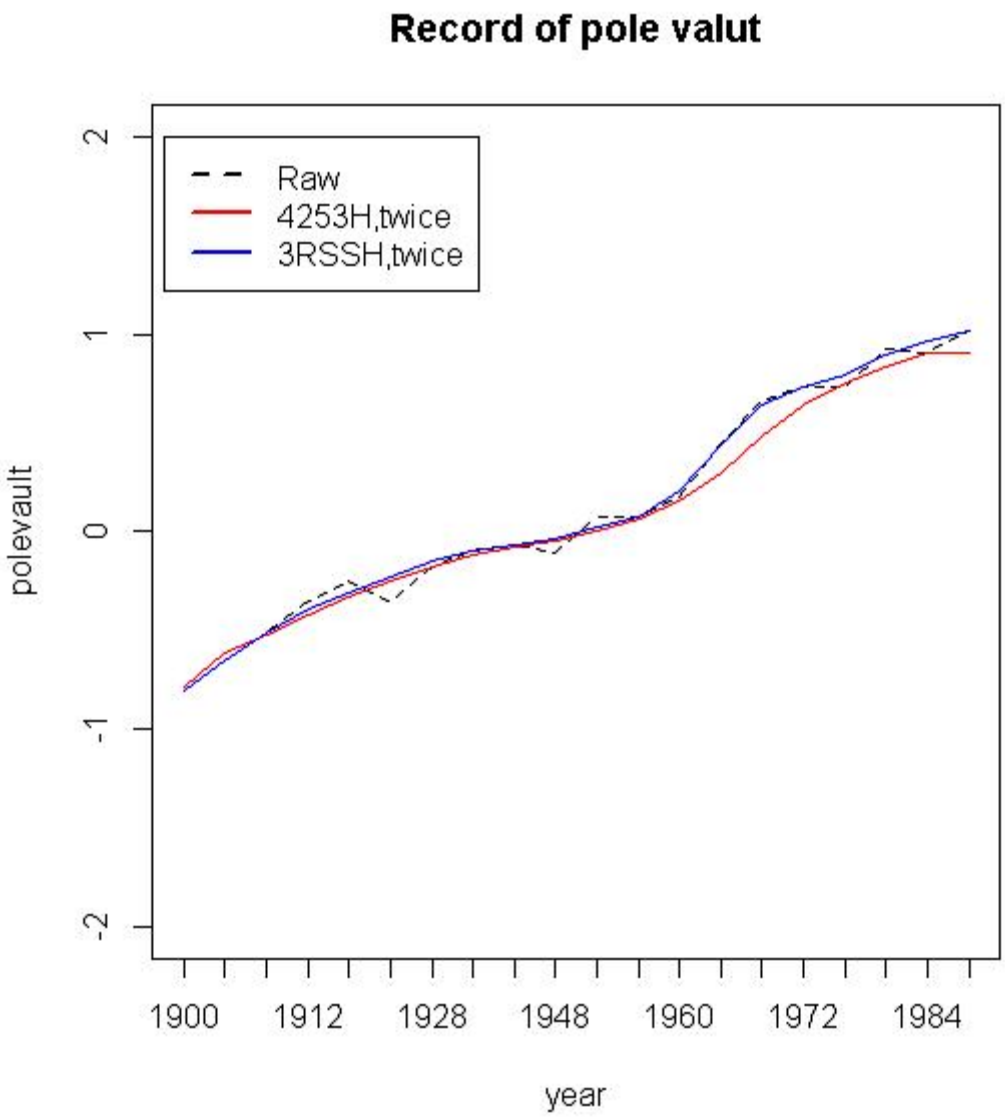
```
> highjump
 [1] -0.45762712 -0.79661017 -0.45762712 -0.35593220
 [5] -0.32203390 -0.18644068 -0.32203390 -0.22033898
 [9] -0.01694915 -0.18644068  0.01694915  0.25423729
[13]  0.42372881  0.49152542  0.69491525  0.66101695
[17]  0.72881356  1.10169492  1.06779661  1.16949153
> polevault
 [1] -0.80419580 -0.65734266 -0.51748252 -0.34965035
 [5] -0.25174825 -0.34965035 -0.17482517 -0.09790210
 [9] -0.06993007 -0.10489510  0.06993007  0.07692308
[13]  0.17482517  0.45454545  0.66433566  0.73426573
[17]  0.73426573  0.93006993  0.90909091  1.01398601
> longjump
 [1] -0.830065359 -0.633986928 -0.450980392
 [4] -0.294117647 -0.882352941 -0.490196078
 [7] -0.111111111 -0.241830065  0.307189542
[10] -0.006535948 -0.333333333  0.006535948
[13]  0.385620915  0.320261438  1.405228758
[16]  0.542483660  0.673202614  0.934640523
[19]  0.934640523  1.169934641
> tripplejump
 [1] -0.74501109 -0.78935698 -0.52771619 -0.59866962
 [5] -0.71396896 -0.25720621 -0.39911308 -0.17294900
 [9] -0.04878049 -0.31485588  0.04878049  0.10199557
[13]  0.31042129  0.32815965  0.56762749  0.54988914
[17]  0.52328160  0.54988914  0.64301552  0.66518847
```

표준화 이후 위에서 plot을 그렸던 방식과 동일하게 다시 평활법을 적용해서 plot을 그려주었다.

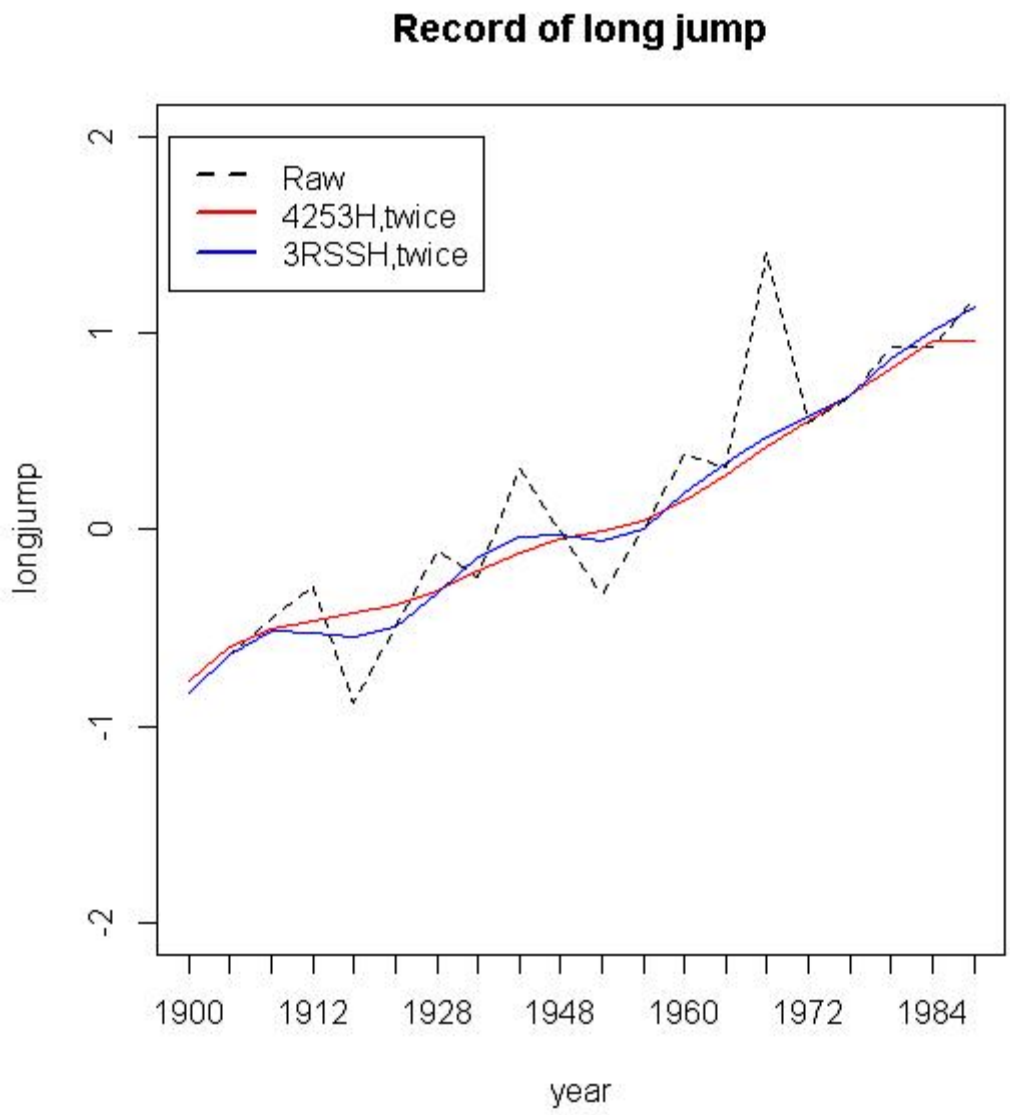
```
plot(highjump,xaxt="n",xlab="year",type="l",ylim=c(-2,2),lty=2,main="Record of high
jump");par(new=T)
plot(sleek(highjump),xaxt="n",col="red",type="l",ylim=c(-2,2),ann=F);par(new=T)
h4253=a3rssh2(highjump)
plot(h4253,xaxt="n",xlab="year",type="l",ylim=c(-2,2),col="blue",ann=F)
axis(1,at=1:20,labels=jump$year)
legend(x = 0.5, y = 2, c("Raw", "4253H,twice", "3RSSH,twice"),
       col = c("black","red","blue"),
       lty=c(2,1,1),lwd=2)
```



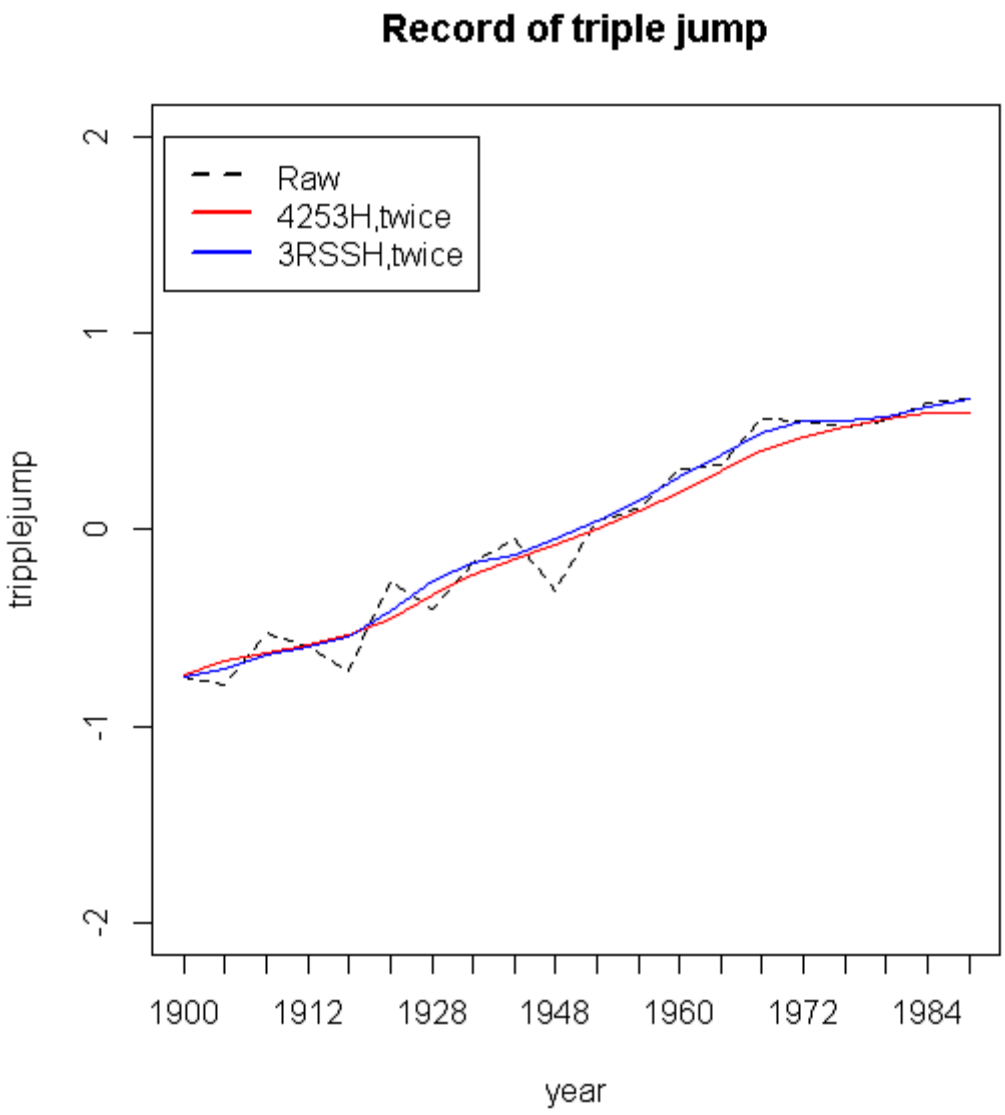

```
plot(polevault,xaxt="n",xlab="year",type="l",lty=2,ylim=c(-2,2),main="Record of pole
valut");par(new=T)
plot(sleek(polevault),xaxt="n",col="red",type="l",ylim=c(-2,2),ann=F);par(new=T)
p4253=a3rssh2(polevault)
plot(p4253,xaxt="n",xlab="year",type="l",ylim=c(-2,2),col="blue",ann=F)
legend(x = 0.5, y = 2, c("Raw", "4253H,twice", "3RSSH,twice"),
       col = c("black","red","blue"),
       lty=c(2,1,1),lwd=2)
axis(1,at=1:20,labels=jump$year)
```




```
plot(longjump,xaxt="n",xlab="year",type="l",lty=2,ylim=c(-2,2),main="Record of long
jump");par(new=T)
plot(sleek(longjump),xaxt="n",col="red",type="l",ylim=c(-2,2),ann=F);par(new=T)
l4253=a3rssh2(longjump)
plot(l4253,xaxt="n",xlab="year",type="l",ylim=c(-2,2),col="blue",ann=F)
legend(x = 0.5, y = 2, c("Raw", "4253H,twice", "3RSSH,twice"),
       col = c("black","red","blue"),
       lty=c(2,1,1),lwd=2)

axis(1,at=1:20,labels=jump$year)
```

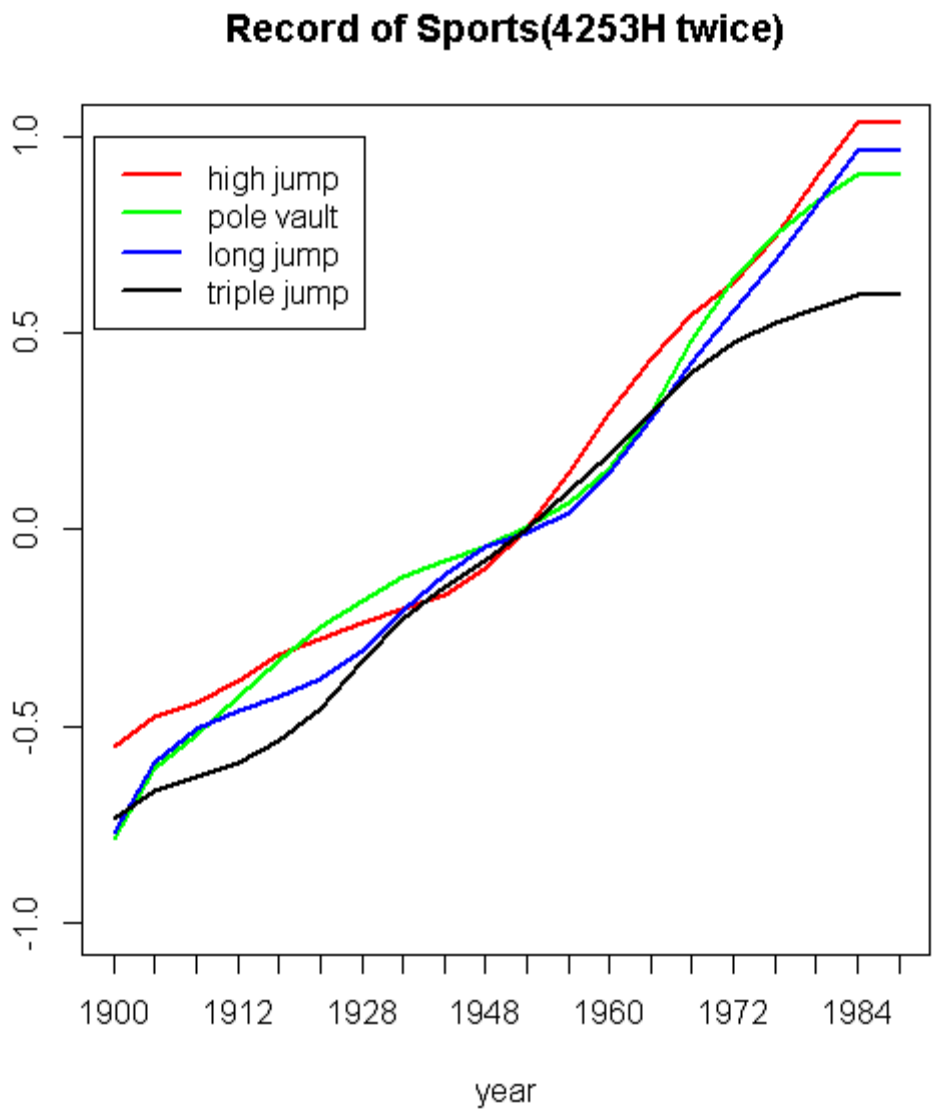



```
plot(tripplejump,xaxt="n",lty=2,xlab="year",type="l",ylim=c(-2,2),main="Record of triple
jump");par(new=T)
plot(sleek(tripplejump),xaxt="n",col="red",type="l",ylim=c(-2,2),ann=F);par(new=T)
t4253=a3rssh2(tripplejump)
plot(t4253,xaxt="n",xlab="year",type="l",ylim=c(-2,2),col="blue",ann=F)
legend(x = 0.5, y = 2, c("Raw", "4253H,twice", "3RSSH,twice"),
       col = c("black","red","blue"),
       lty=c(2,1,1),lwd=2)
axis(1,at=1:20,labels=jump$year)
```



4가지 종목을 한 번에 비교분석하기 위해 아래 코드를 실행하여 함께 비교해보았다.

```
plot(lwd=2,sleek(highjump),xaxt="n",xlab="year",type="l",ylim=c(-1,1),col="red",ylab="",main="Record of Sports(4253H twice)");par(new=T)
plot(lwd=2,sleek(polevault),xaxt="n",xlab="year",type="l",ylim=c(-1,1),col="green",ann=F);par(new=T)
plot(lwd=2,sleek(longjump),xaxt="n",xlab="year",type="l",ylim=c(-1,1),col="blue",ann=F);par(new=T)
plot(lwd=2,sleek(tripplejump),xaxt="n",xlab="year",type="l",ylim=c(-1,1),col="black",ann=F)
axis(1,at=1:20,labels=jump$year)
legend(x = 0.5, y = 1, c("high jump", "pole vault", "long jump", "triple jump"),
       col = c("red","green","blue","black"), lty=c(1,1,1,1),lwd=2)
```



상단의 plot을 통해 시간의 흐름에 따라 기울기가 가장 큰 종목이 가장 기록이 많이 변화한 종목이라는 결론을 내릴 수 있다. triple jump의 경우 1968년 이후 기울기가 다른 3개의 종목에 비해 매우 완만해졌음을 알 수 있다. 즉, 다른 종목에 비해 기록 갱신이 어려워졌음을 확인할 수 있다.