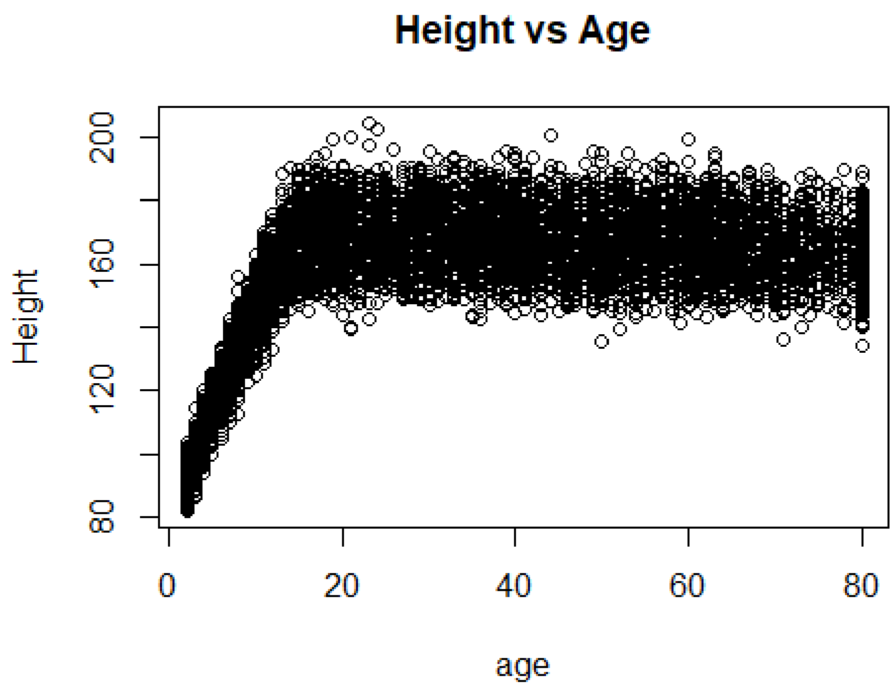
# multi_reg_3.R

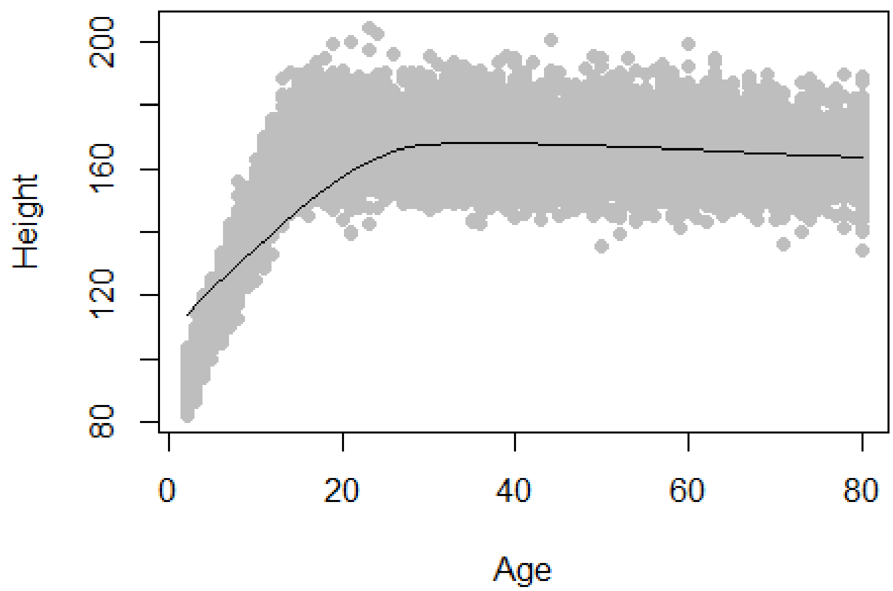
SANGHOOJEFFREY

Tue Jun 26 21:38:58 2018

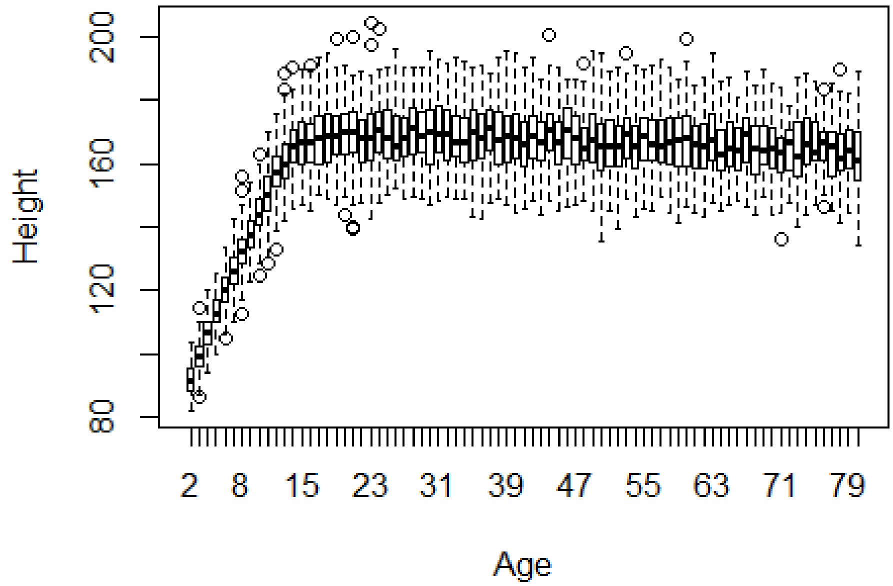
```r
library(car)

## Warning: package 'car' was built under R version 3.4.4

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.4.4

# 스플라인, 커널 회귀모형을 이용한 비선형모형 세우기


# 키, 몸무게, 나이, 성별 간의 관계를 알고싶다.
# 2011-2012 질병관리센터 국가보건영양검사에서 얻은 인체측정학적 데이터를 분석해보자.

body <- read.csv("https://scholar.harvard.edu/files/gerrard/files/nhanes_body.
txt")
attach(body)

dim(body) # 총 8,602 개의 자료, 5 개의 변수

## [1] 8602    5

plot(age, height, xlab="age", ylab="Height", main="Height vs Age")
```




```
# 두 변수의 관계가 선형이 아님을 확인할 수 있다. 그러면 이에 적합한 모형식을 세워보자.

scatter.smooth(height~age, xlab='Age', ylab='Height', col='gray', pch=16)
```

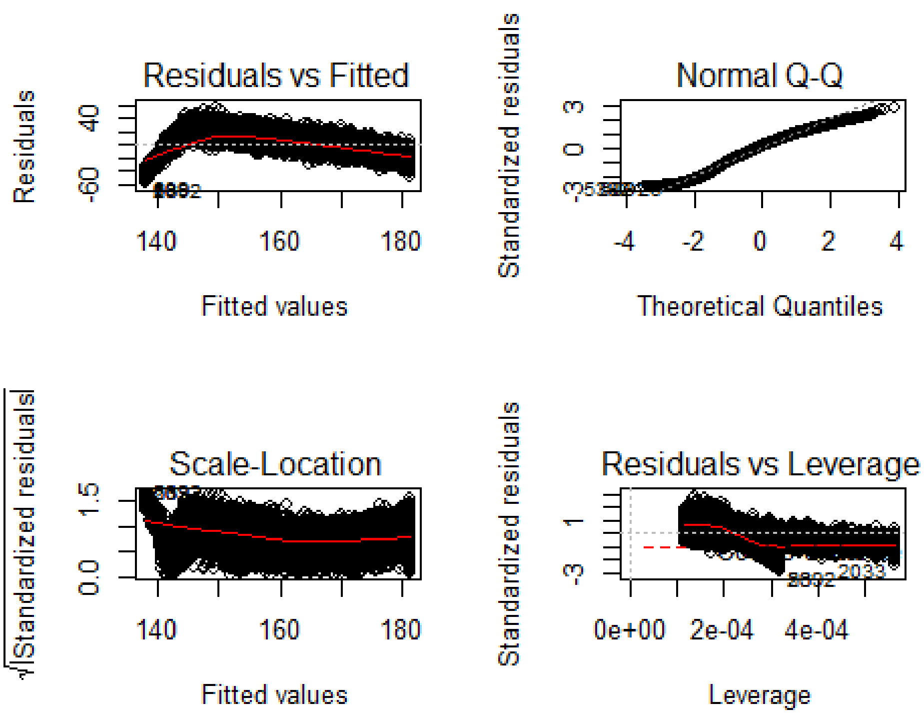



```
# 데이터의 전체적인 패턴을 시각화하고 싶으면 scatter.smooth()함수를 사용하면 된다.

boxplot(height~age, xlab='Age', ylab='Height')
```

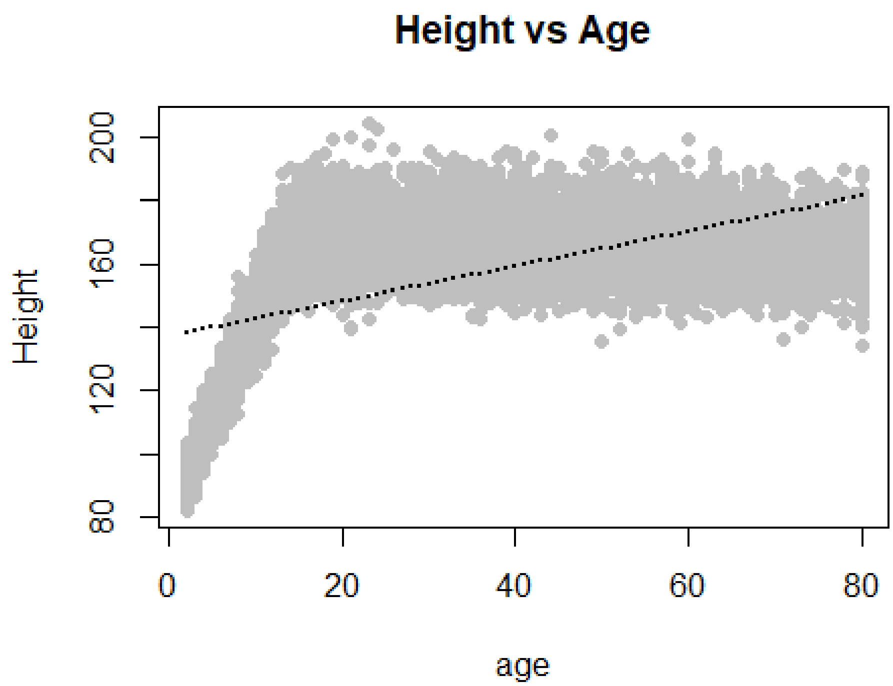



```r
#이 자료의 또 다른 시각화 방법


# 선형모형으로 세워보면
fit.l1 <- lm(height~age)
summary(fit.l1)

##
## Call:
## lm(formula = height ~ age)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.087 -12.657   2.087  14.693  54.811
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.370e+02  3.691e-01   371.1   <2e-16 ***
## age         5.524e-01  9.012e-03    61.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.85 on 8600 degrees of freedom
## Multiple R-squared:  0.3041, Adjusted R-squared:  0.304
## F-statistic:  3758 on 1 and 8600 DF,  p-value: < 2.2e-16
```

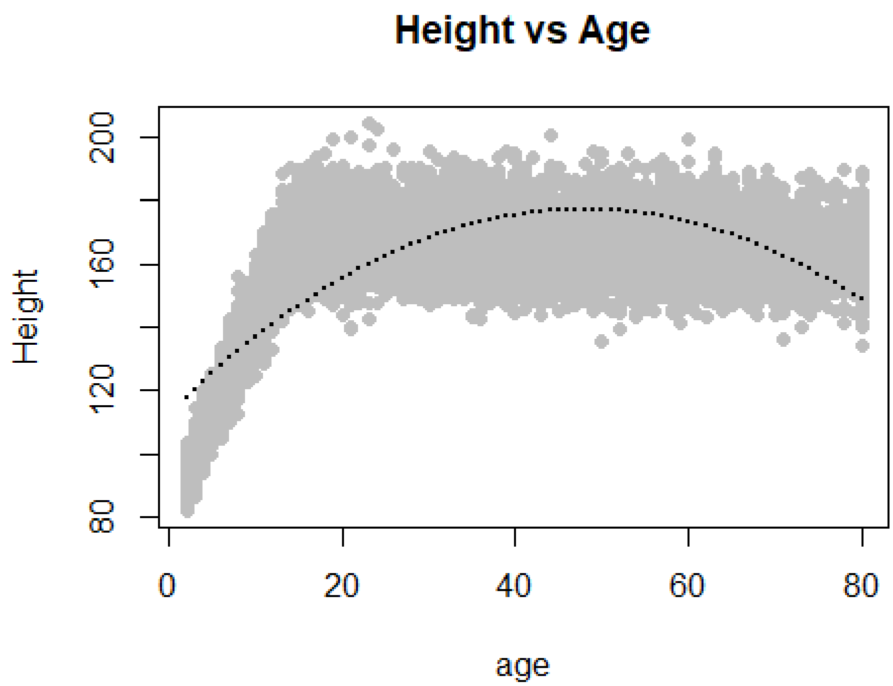
```r
par(mfrow=c(2,2))
plot(fit.l1)
```



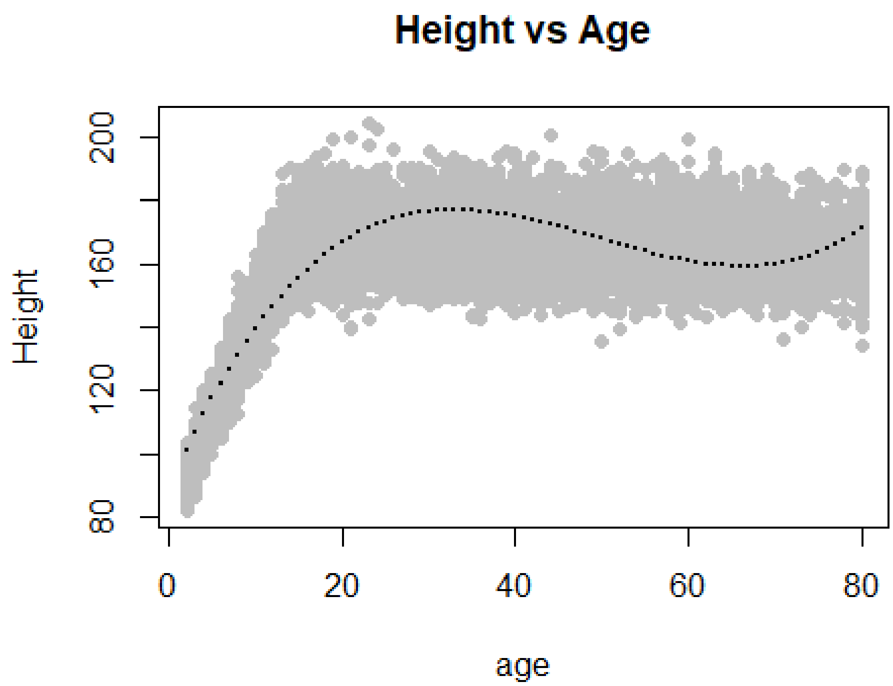

```r
par(mfrow=c(1,1))
plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height
 vs Age")
points(age, fit.l1$fitted.values, pch=16, cex=0.1)
```



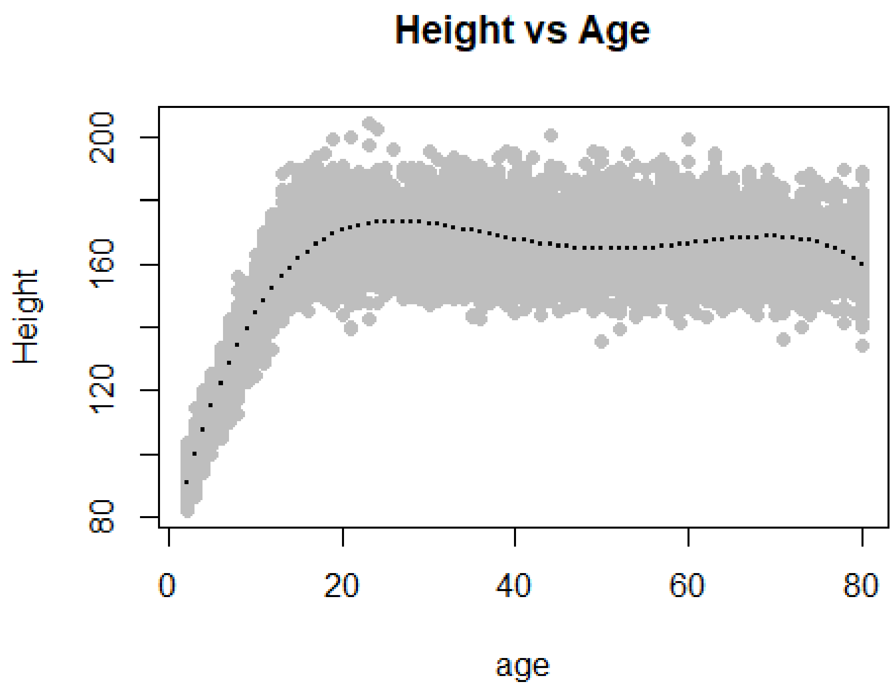

```
#2 차 선형모형
fit.l2 <- lm(height~age+I(age^2))
plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height
 vs Age")
points(age, fit.l2$fitted.values, pch=16, cex=0.1)
```



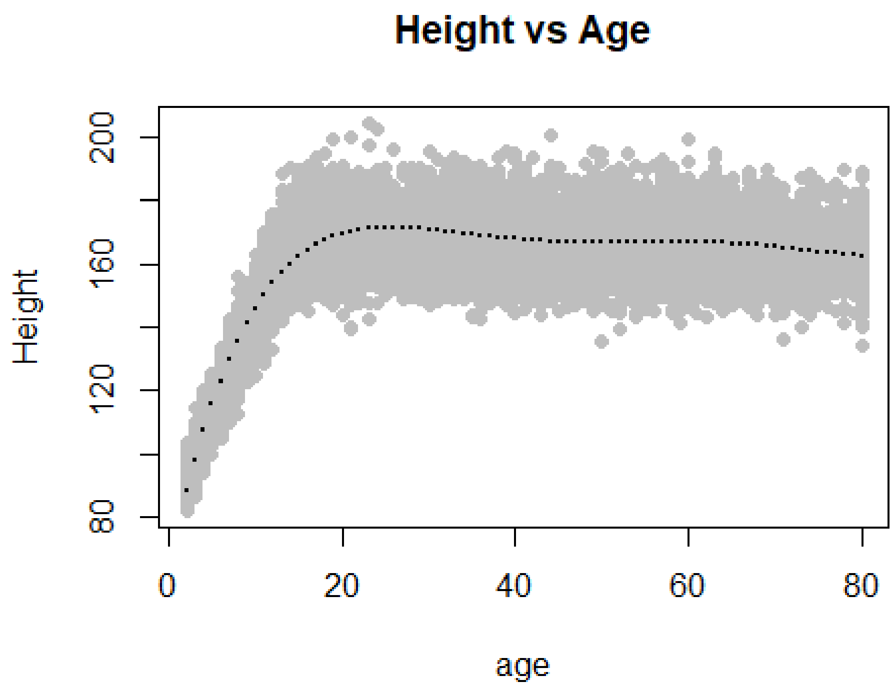

```
#3 차 선형모형
fit.l3 <- lm(height~age+I(age^2)+I(age^3))
plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height
 vs Age")
points(age, fit.l3$fitted.values, pch=16, cex=0.1)
```



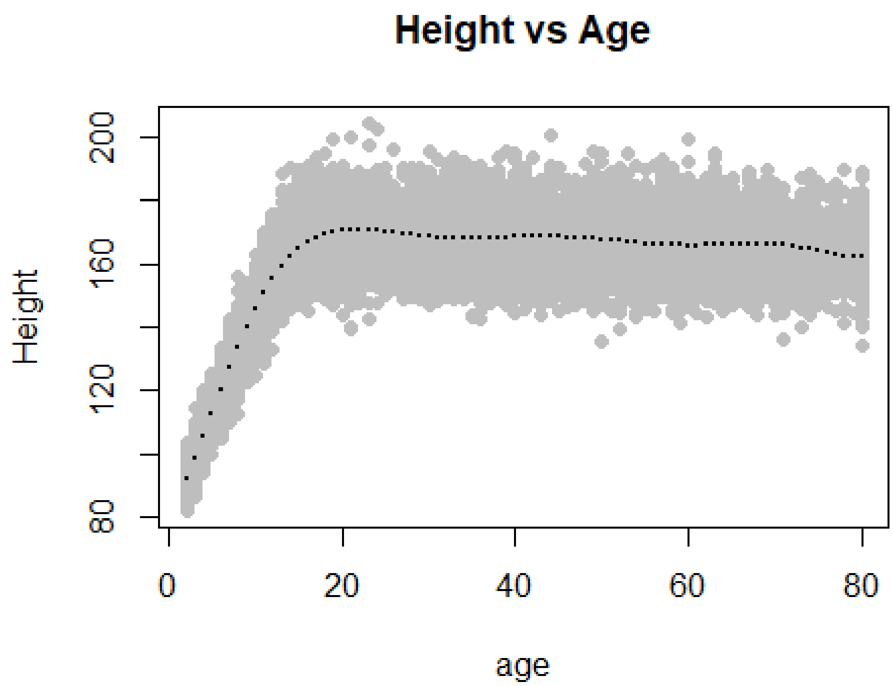

```
#4 차 선형모형
fit.l4 <- lm(height~age+I(age^2)+I(age^3)+I(age^4))
plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height
 vs Age")
points(age, fit.l4$fitted.values, pch=16, cex=0.1)
```



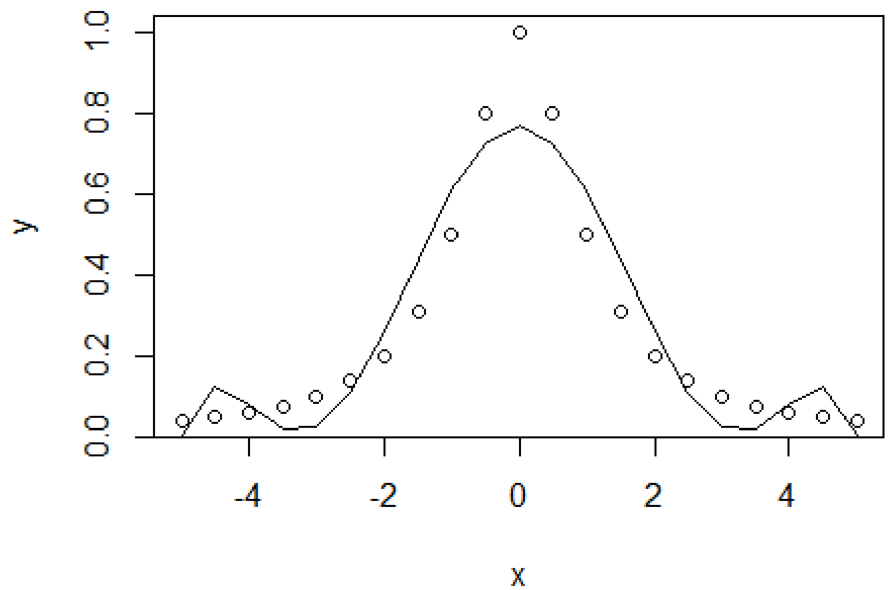

```r
#5 차 선형모형
fit.l5 <- lm(height~age+I(age^2)+I(age^3)+I(age^4)+I(age^5))
plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height
 vs Age")
points(age, fit.l5$fitted.values, pch=16, cex=0.1)
```



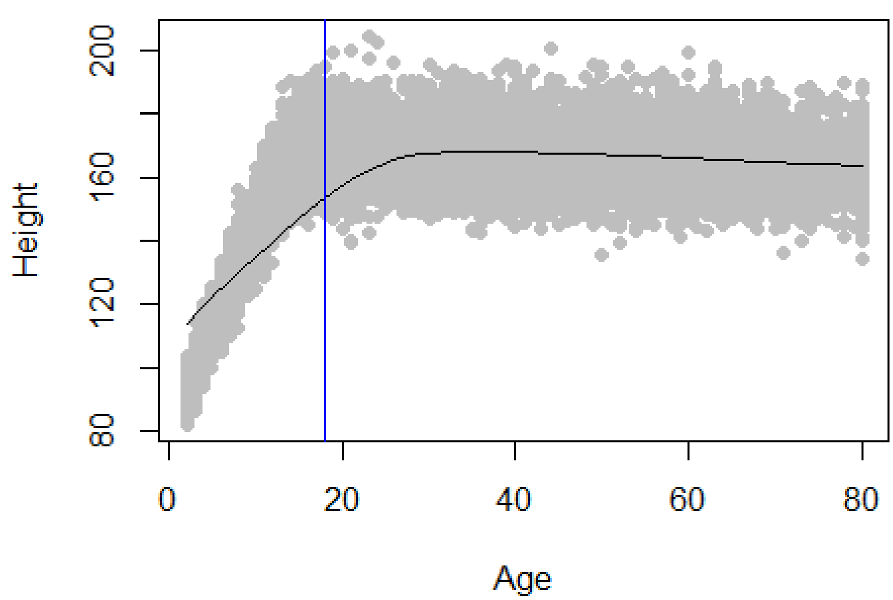

```
#8 차 선형모형
fit.l8 <- lm(height~age+I(age^2)+I(age^3)+I(age^4)+I(age^5)+I(age^6)+I(age^7)
+I(age^8))
plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height
 vs Age")
points(age, fit.l8$fitted.values, pch=16, cex=0.1)
```



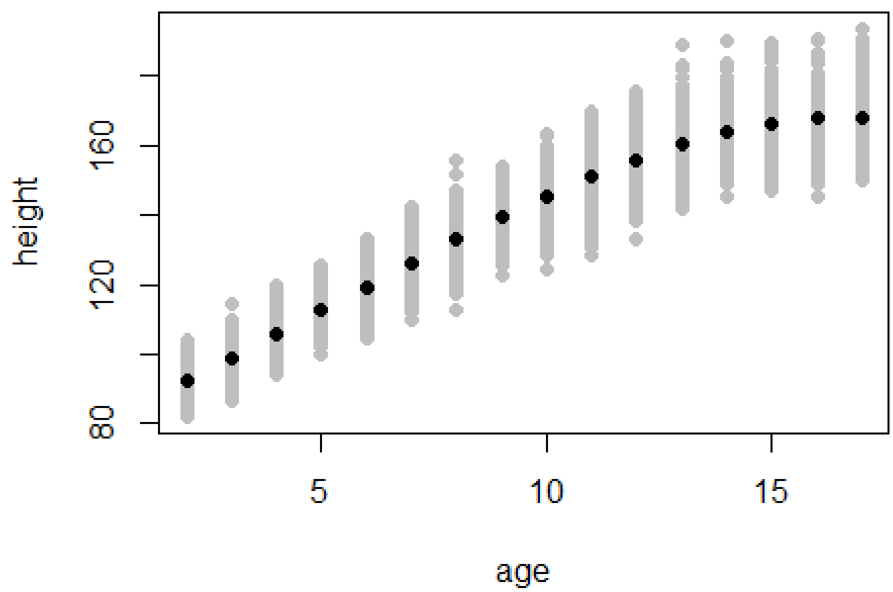

```
# 높은 차수의 다항식을 이용하면 결과가 좋아진다.
# 하지만 비수렴의 문제가 있어 3 차식 이하로 모형을 세워야 한다.

# 비수렴 문제의 예
runge <- function(x) {return(1/(1+x^2))}
x <- seq(-5, 5, 0.5)
y <- runge(x)

plot(y~x)

fit.runge <- lm(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6))
lines(fit.runge$fitted.values ~ x)
```



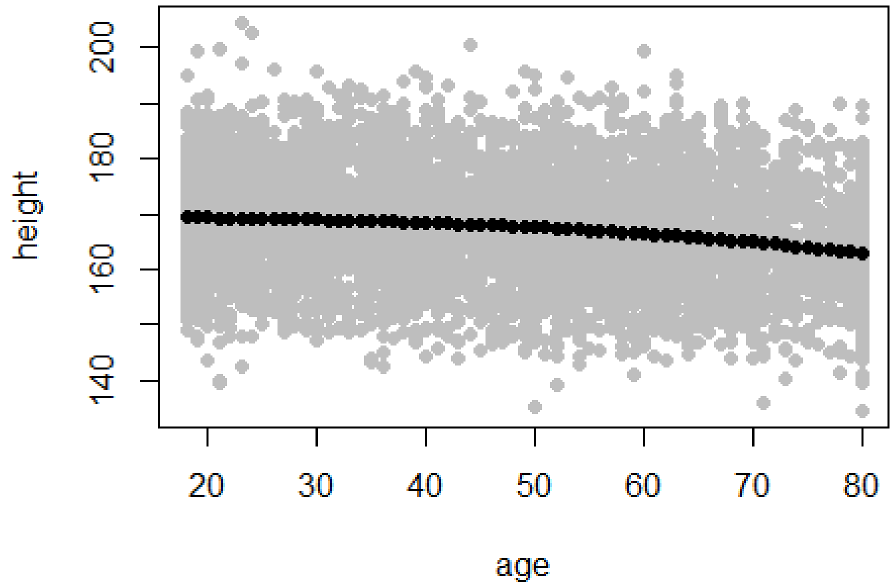

```r
scatter.smooth(height~age, xlab='Age', ylab='Height', col='gray', pch=16)
abline(v=18, col="blue")
```



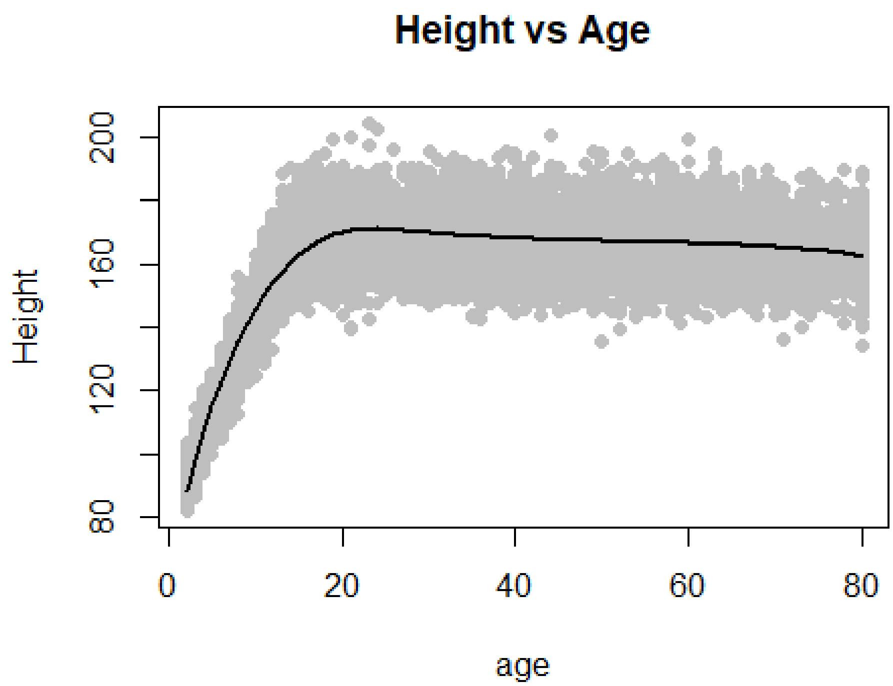
```r
# 성장 중인 것으로 기대되는 사람 (18 세 미만)
# 성장이 끝난 것으로 기대되는 사람 (18 세 이상)

detach(body)

youths <- body[body$age<18,]
adults <- body[body$age>=18,]

fit.youths.l3 <- lm(height~age+I(age^2)+I(age^3), data=youths)
plot(height~age, data=youths, pch=16, col="gray")
points(youths$age, fit.youths.l3$fitted.values, pch=16)
```



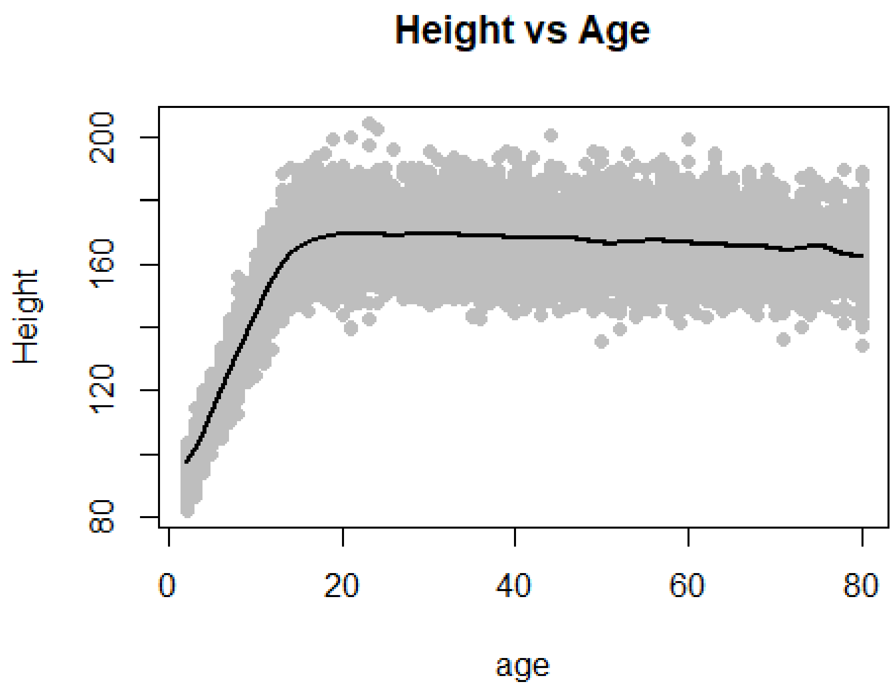

```r
fit.adults.l3 <- lm(height~age+I(age^2)+I(age^3), data=adults)
plot(height~age, data=adults, pch=16, col="gray")
points(adults$age, fit.adults.l3$fitted.values, pch=16)
```



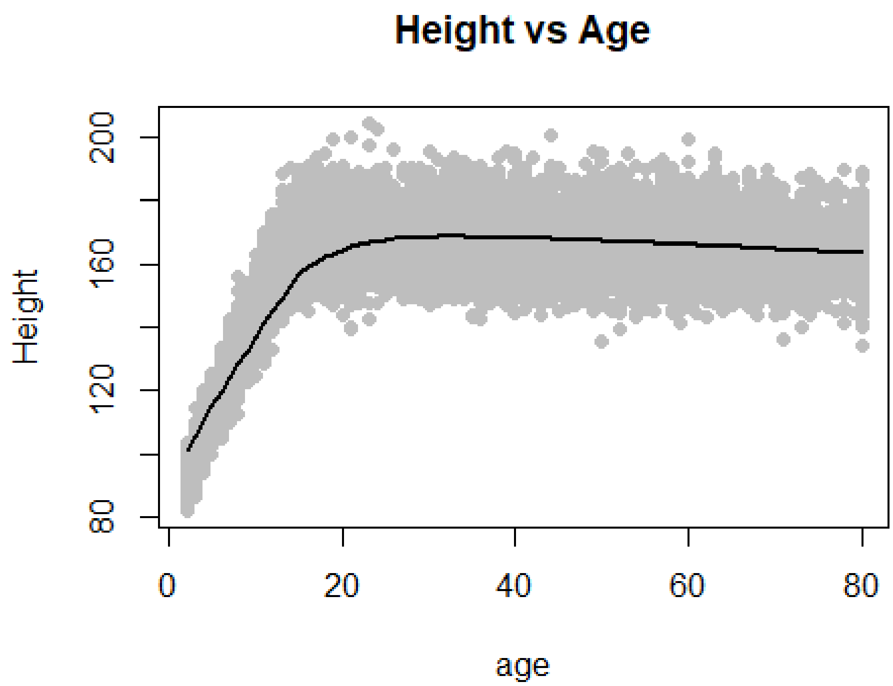

```
# 이러한 회귀분석 접근법을 스플라인(spline)이라 한다.
attach(body)
fit.spline <- smooth.spline(height ~ age, nknots=4)
# 여기서 nknots 는 매듭점이라 한다.

plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height
 vs Age")
lines(fit.spline, pch=16, lwd=2)
```




```
# 부드러운 함수를 통해 데이터를 설명하고 있으나, 수학적인 성질에 대해서는 이야기 하지 못하는 단점
fit.spline

## Call:
## smooth.spline(x = height ~ age, nknots = 4)
##
## Smoothing Parameter  spar= -0.3368845  lambda= 9.717504e-05 (16 iterations)
## Equivalent Degrees of Freedom (Df): 5.998716
## Penalized Criterion (RSS): 30177.52
## GCV: 82.41412

# 커널 회귀를 이용하면 비선형 자료에 부드러운 함수를 적합시켜준다
smooth.height <- ksmooth(age, height, bandwidth=4, kernel='normal')
# bandwidth : 평활량 값 (평활량이 클수록 부드럽고, 작을수록 거친 그림이 그려진다)

plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height vs Age")
lines(smooth.height, pch=16, lwd=2)
```




```
#국소 가중선형회귀를 이용한 비모수적 방법으로도 부드러운 함수를 그릴 수 있다.
# 여기서 f 는 평활량 값
lowess.height <- lowess(height~age, f=0.5)
plot(age, height, pch=16, col="gray", xlab="age", ylab="Height", main="Height
 vs Age")
lines(lowess.height, lwd=2)
```




```
# Piecewise regression
x <- c(1:10, 13:22)
y <- numeric(20)
## Create first segment
set.seed(124134)
y[1:10] <- 20:11 + rnorm(10, 0, 1.5)
## Create second segment
y[11:20] <- seq(11, 15, len=10) + rnorm(10, 0, 1.5)

plot(x,y,  pch=16)

#segmented 패키지를 이용하면 컴퓨터가 최적 breakpoint 를 찾아준다.
if(!require(segmented)) install.packages("segmented"); library(segmented)

## Loading required package: segmented

## Warning: package 'segmented' was built under R version 3.4.3

lin.mod <- lm(y~x)
segmented.mod <- segmented(lin.mod, seg.Z = ~x)
summary(segmented.mod)

##
##  ***Regression Model with Segmented Relationship(s)***
##
## Call:
```

```
## segmented.lm(obj = lin.mod, seg.Z = ~x)
##
## Estimated Break-Point(s):
##    Est. St.Err
##  8.310  0.743
##
## Meaningful coefficients of the linear terms:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23.4070     1.1225  20.852 5.02e-13 ***
## x            -1.5121     0.2223  -6.803 4.25e-06 ***
## U1.x          1.8193     0.2457   7.405       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.441 on 16 degrees of freedom
## Multiple R-Squared: 0.8348,  Adjusted R-squared: 0.8038
##
## Convergence attained in 4 iterations with relative change 4.279568e-16
```

```r
plot(x,y, pch=16)
plot(segmented.mod, add=T, col="blue")
```