Seoul National University
Dept. of Civil and Environmental Engineering

Instructor: Junho Song
junhosong@snu.ac.kr

# M1586.002500 Information Engineering for CE Engineers
## In-Class Material: Class 17
## Linear Model Selection and Regularization (ISL Chapter 6)

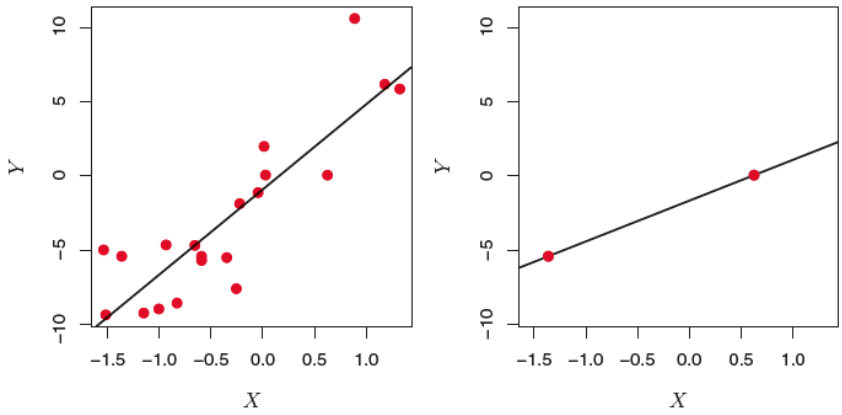
## 1. High-Dimensional Data

### (a) Definition

The number of features, $p$ is larger than that of observations, $n$. The consideration about high-dimension setting also applies when $p$ is slightly smaller than $n$

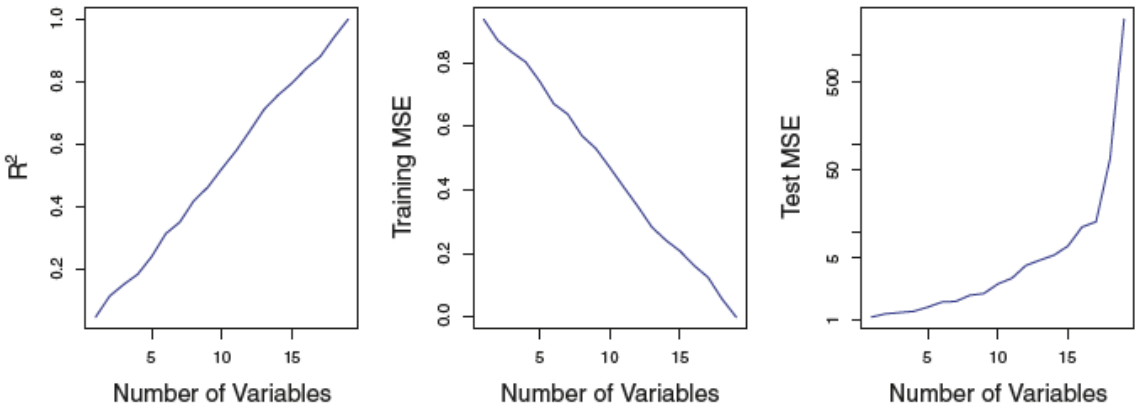
### (b) The limits of most traditional statistical techniques for regression and classification

Consider a low-dimensional setting in which the number of observations, $n$ is much greater than the number of features $p$ (left)




Least squares regression with $\boldsymbol{n = 20}$ and $\boldsymbol{n = 2}$

When $p > n$ or $p \approx n$ (right), even a simple least squares regression line can be too **flexible** and hence may **overfit** the data




### (c) As more features are included, the $R^2$ increases to 1, the training set MSE decreases to 0, but the test set MSE increases

Instructor: Junho Song
junhosong@snu.ac.kr

(d) "Model selection" method in the high-dimensional setting

In the high-dimensional setting, one should never use traditional measures such as p-value, $R^2$ statistics, $C_p$, AIC, and BIC.

It is so simple to obtain a model with $R^2 = 1$ when $p > n$, whereas this provides absolutely no evidence of having a compelling model

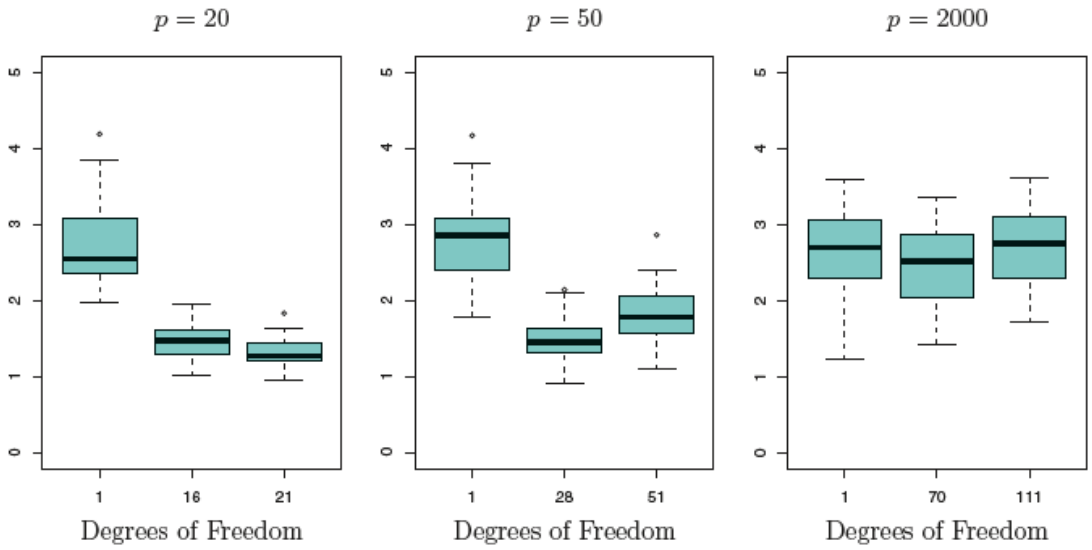
## 2. Regression in High Dimensions

(a) Less flexible least squares models are useful for performing regression in the high-dimensional setting

e.g. Ridge regression, the Lasso, and principal components regression

➔ These approaches avoid overfitting by using a less flexible fitting approach than least squares

(b) Consider the test MSE of Lasso regressions from $n = 100$ observations. In this simulated example, 20 of the predictors are actually associated with the response.




In each boxplot, the degree of freedom, i.e. the number of non-zero coefficient estimates, is displayed rather than reporting the values of $\lambda$ used. The figure highlights three important points:

① Shrinkage (regularization) plays a key role in high-dimensional problems
② Appropriate tuning parameter selection is crucial for good predictive performance
③ The test error tends to increase as the dimensionality of the problem increase, unless the additional features are truly associated with the response ➔ ***curse of dimensionality***

(c) **Curse of dimensionality**: Adding noise features unrelated with the response will lead to a deterioration of the model

Instructor: Junho Song
junhosong@snu.ac.kr

## 3. Interpreting results in High Dimensions

(a) In the high-dimensional setting, the ***multicollinearity*** problem that the variables in a regression is correlated with each other can be extreme

➔ **Multicollinearity**: Any variable in the model can be written as a linear combination of others

(b) It is impossible to know exactly which variables truly are predictive of the outcome, and identify the best coefficients

(c) One can incorrectly conclude that a model predict more effectively than the other models in the high-dimensional setting

- There may be many sets of the same number of components that would predict response $Y$ just as well as the selected model
- This does not disparage the value of the model obtained, but it is one of many effective models
- It must be further validated on independent data sets

Seoul National University
Dept. of Civil and Environmental Engineering

Instructor: Junho Song
junhosong@snu.ac.kr

# M1586.002500 Information Engineering for CE Engineers
## In-Class Material: Class 18
## Moving Beyond Linearity (ISL Chapter 7)

Problem: Standard "Linear" regression can have limitation in prediction power

Question: How can we improve regression models beyond linearity?

→ "**Extensions of linear models**" e.g. polynomial regression, regression splines

### 1. Polynomial regression

(a) The standard linear relationship between $X$ and $Y$,

$$Y = \beta_0 + \beta_1 X + \epsilon$$

can be extended by adding extra predictors defined as polynomial functions, i.e.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \cdots + \beta_d X^d + \epsilon$$

- $X, X^2, \ldots, X^d$: the "predictor" variables allow us to augment the inputs with polynomial terms to achieve higher-order Taylor expansions

- $d$: degree of freedom which determines the complexity of model

  → e.g. $d = 3$ : cubic polynomial regression

- Polynomial curve can become **"overly flexible"** and/or **"extremely non-linear"** curve for too large degree $d$ (See plots on the next page)

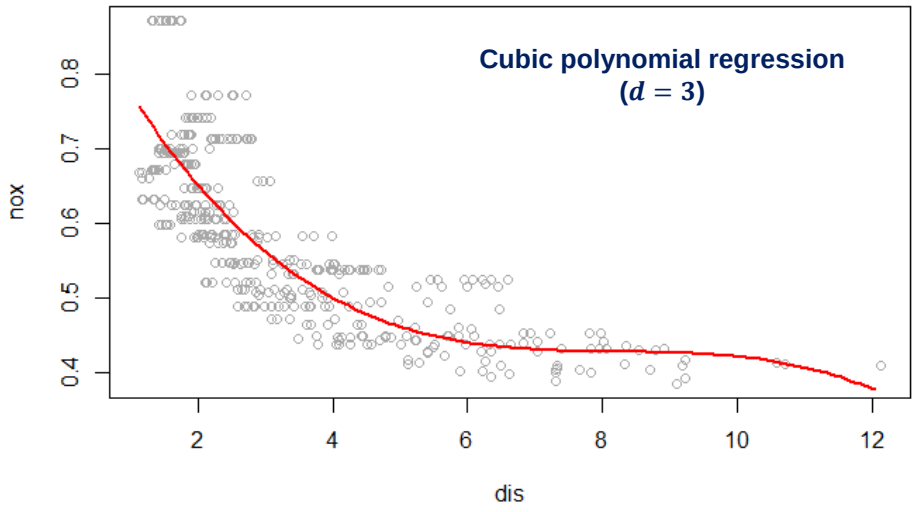
(b) Estimation of coefficients $\beta_i$

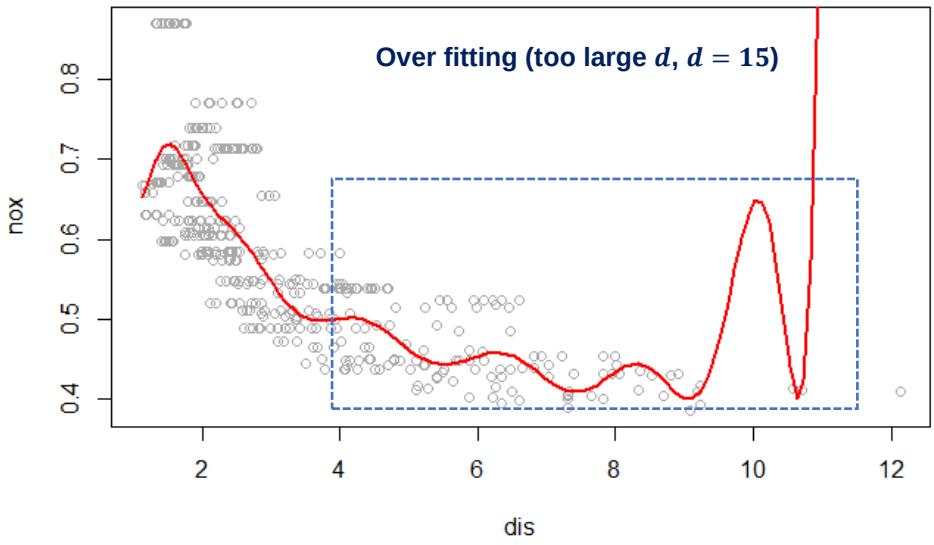
In the same way of linear regression, the model coefficients $\beta_i$ is calculated using least squares estimator:

$$RSS = \sum_{i=1}^{n} e_i^2$$
$$= \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i - \cdots \hat{\beta}_d x_i^d)^2$$

Find each $\hat{\beta}_i$ $(i = 1, \ldots, d)$ that minimize RSS → $\frac{\partial RSS}{\partial \hat{\beta}_i} = 0$

Instructor: Junho Song
junhosong@snu.ac.kr



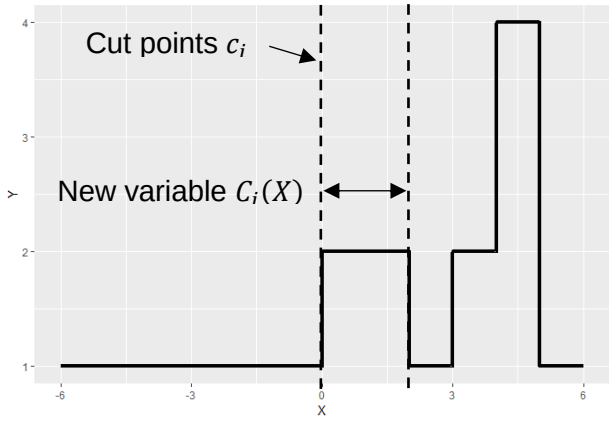

```
library(MASS) # MASS library
fix(Boston) # Load 'Boston' data
attach(Boston)
lm.fit = lm(nox ~ poly(dis, 3), data = Boston)
# Fit a cubic polynomial regression with degree 3 (we can change the
   degree d from 3 to any number)
summary(lm.fit) # show more detailed information
dislim = range(dis)
dis.grid = seq(from = dislim[1], to = dislim[2], by = 0.1)
lm.pred = predict(lm.fit, list(dis = dis.grid))
plot(nox ~ dis, data = Boston, col = "darkgrey") # scatter plot
lines(dis.grid, lm.pred, col = "red", lwd = 2)
# polynomial regression line
```

Instructor: Junho Song
junhosong@snu.ac.kr

## 2. Step functions

(a) "Step function": Convert a continuous variables into an *ordered categorical variable* with constructing $K$ cut points $c_i$, and $K+1$ new variables $C_i(X)$.

$$C_0(X) = I(X < c_1)$$
$$C_1(X) = I(c_1 \le X < c_2)$$
$$\vdots$$
$$C_K(X) = I(c_K \le X)$$




→ $I(\cdot)$ is an "indicator function" that returns a **1** if the condition is true, and **0** otherwise.

→ $C_0(X) + C_1(X) + \cdots + C_K(X) = 1$, since point $X$ must be in exactly one of the $K+1$ intervals.

**Basic intuition:** Break the range of $X$ into *bins*, and fit a different constant in each bin.

(b) Regression model:

$$Y = \beta_0 + \beta_1 C_1(X) + \beta_2 C_2(X) + \cdots + \beta_K C_K(X) + \epsilon$$

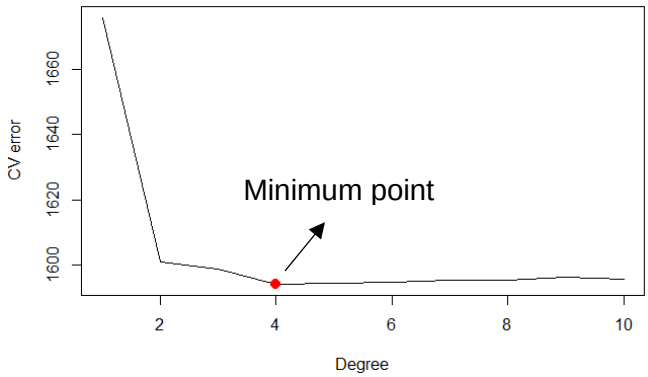
→ $\beta_0$ can be interpreted as the mean value of $Y$ for $X < c_1$, and $\beta_j$ represents the average increase in the response for $X$ in $c_j \le X < c_{j+1}$ relative to $X < c_1$.

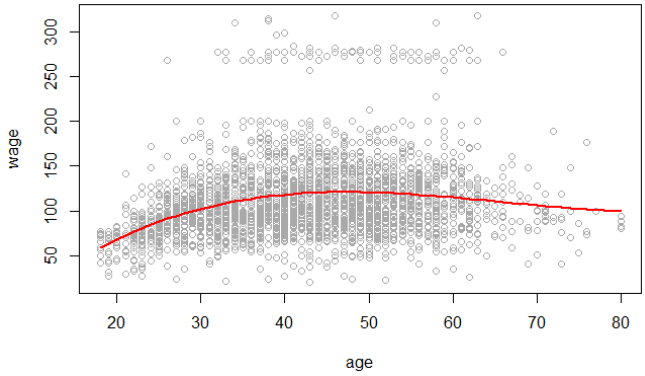
**Note:** $C_0(X), C_1(X), \ldots, C_K(X)$ are the predictor

(c) How to choose the breakpoints in the predictors?

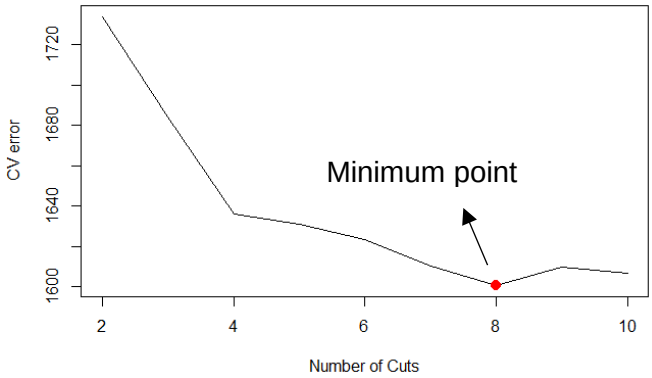
→ Use cross-validation approaches, as discussed in CM 12-13

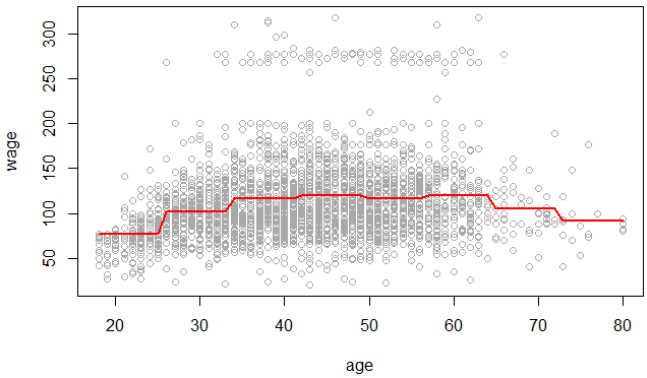
Instructor: Junho Song
junhosong@snu.ac.kr




→ The **degree of freedom** $d$ is determined in polynomial regression




→ The **number of cuts** is determined in step function regression

```r
library(ISLR) # ISLR library
library(boot) # boot library

## CV - Step
cvs <- rep(NA, 10)
for (i in 2:10) {
  Wage$age.cut <- cut(Wage$age, i)
  fit <- glm(wage ~ age.cut, data = Wage)
  cvs[i] <- cv.glm(Wage, fit, K = 10)$delta[1]
}
plot(2:10, cvs[-1], xlab = "Number of Cuts", ylab = "CV error", type = "l")
d.min <- which.min(cvs)
points(d.min, cvs[d.min], col = "red", cex = 2, pch = 20)

## step plot
plot(wage ~ age, data = Wage, col = "darkgrey")
agelims <- range(Wage$age)
age.grid <- seq(from = agelims[1], to = agelims[2])
fit <- glm(wage ~ cut(age, 8), data = Wage)
preds <- predict(fit, data.frame(age = age.grid))
lines(age.grid, preds, col = "red", lwd = 2)


# For CV of polynomial regression, see R codes at eTL
```

Instructor: Junho Song
junhosong@snu.ac.kr

## 3. Regression splines

(a) "Basis functions":

$$Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \cdots + \beta_n b_n(X) + \epsilon$$

- $\beta_0, \beta_1, \ldots, \beta_n$: "model coefficient"
- $b_1(X), b_2(X), \ldots, b_n(X)$: "basis function"

- For polynomial regression: $b_j(X) = X^j$, step functions: $b_j(X) = C_j(X)$.
  → Many alternatives are possible (e.g. wavelets or Fourier series)

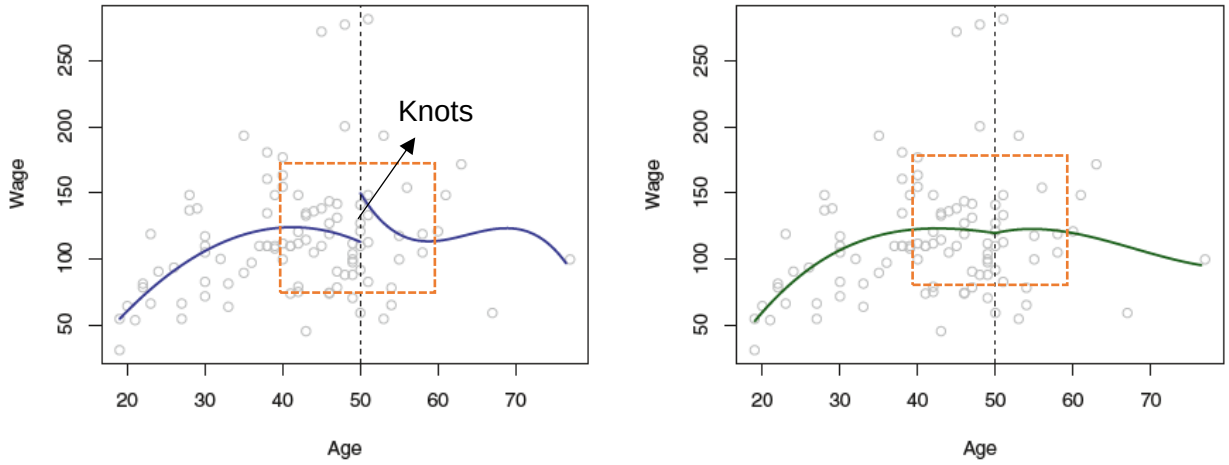
**Note:** The basis functions $b_1(X), b_2(X), \ldots, b_K(X)$ are fixed and known in many cases.

(b) Regression Splines:

**Basic intuition:** Instead of fitting a high-level polynomial, fit separate low-degree polynomials over different regions of $X$.

The extension of previous regression:

→ Dividing the range of $X$ in ***K* distinct regions** with **polynomial functions**
→ The polynomials are **constrained** so that they join smoothly at the region boundaries, or ***knots***




For example, a piecewise "cubic" with a single knot at a point $c$ takes the form:

$$Y = \begin{cases} \beta_{01} + \beta_{11}X + \beta_{21}X^2 + \beta_{31}X^3 + \epsilon & if\ X < c \\ \beta_{02} + \beta_{12}X + \beta_{22}X^2 + \beta_{32}X^3 + \epsilon & if\ X \geq c \end{cases}$$

Instructor: Junho Song
junhosong@snu.ac.kr

(c) Degree of freedom $d$ for splines

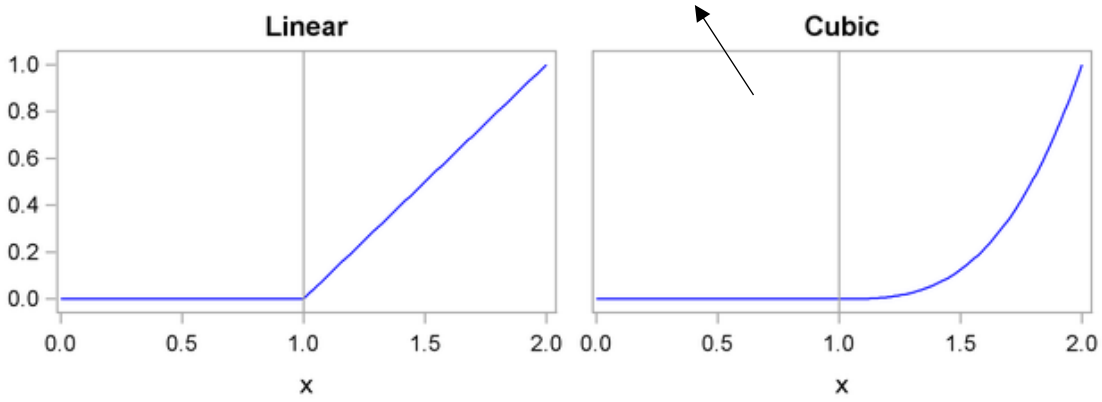
To fit **continuous** and **smooth**, degree-$d$ splines is a piecewise degree-$d$ polynomial, with continuity in derivatives up to degree $d-1$ at each knot.

**Note**: Each constraint effectively frees up **one** degree, by reducing the complexity of the resulting piecewise polynomial fit. (→ Compatibility condition)

Then, how to represent the general regression mode?

→ The most direct way to represent a cubic spline is a **"truncated power basis function"**

$$h(X,\xi) = (X-\xi)_+^3 = \begin{cases} (X-\xi)^3 & if\ X > \xi \\ 0 & if\ otherwise, \end{cases}$$



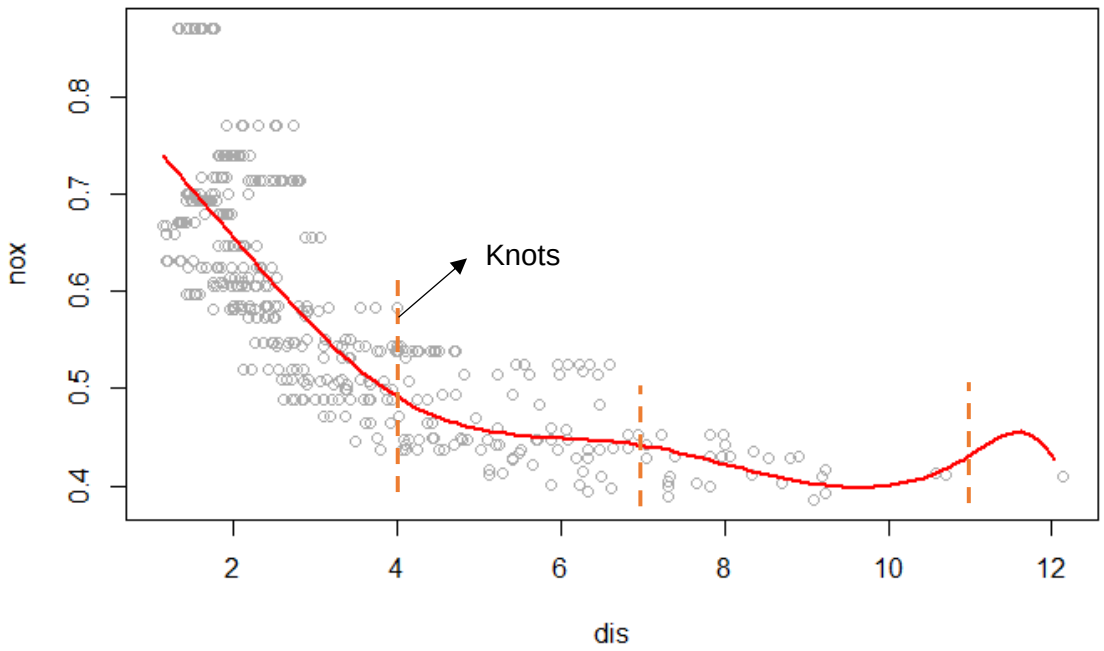

A "cubic" spline with $K$ knots can be modeled as ($\xi_K$ is $K$-th knot)

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 h(X,\xi_1) + \cdots + \beta_{K+3} h(X,\xi_K) + \epsilon$$

→ The function remain continuous, with continuous first and second derivatives, at each of the knots.

**Note**: Total $K+4$ degrees of freedom → 1 intercept + $(K+3)$ predictors

Instructor: Junho Song
junhosong@snu.ac.kr




→ Regression spline with 4 intervals (knots at [4,7,11])

```
library(MASS) # MASS library
attach(Boston)
library(splines) # splines library
dislims <- range(Boston$dis)
dis.grid <- seq(from = dislims[1], to = dislims[2], by = 0.1)
fit <- lm(nox ~ bs(dis, df = 4, knots = c(4, 7, 11)), data = Boston)
# Use the “bs()” function to fit a regression spline
summary(fit) # Report summary
sp.pred <- predict(fit, list(dis = dis.grid))
plot(nox ~ dis, data = Boston, col = "darkgrey")
lines(dis.grid, sp.pred, col = "red", lwd = 2) # Plot regression splines
```