# 재추출 방법들

서 울 대 학 교
통 계 학 과

2017년 8월

# 다루는 내용

1. 시험오차, 과적합, 평균분산균형
2. 교차검증
   - 2.1 검증자료방법
   - 2.2 하나남기기 교차검증
   - 2.3 k겹 교차검증
3. 붓스트랩

# 재추출(resampling) 방법이란?

## 정의

훈련자료에서 표본을 수많이 재추출하여 재추출된 표본에 모형을 적합하여 적합된 모형에 대한 정보를 얻는 방법들

## 두 가지 재추출 방법들

1. 교차검증(cross-validation, cv) 방법 : 시험오차를 추정하는데 주로 사용
2. 붓스트랩(bootstrap) 방법 : 파라미터 추정량의 정확성을 측정하기 위해 사용

# 노트.

1. 교차검정(cross-validation, cv) 방법 : 시험오차를 추정하는데 주로 사용된다. 시험오차로 모형의 성능을 측정하거나(model assessment) 모형을 선택(model selection)하는데 사용된다.
2. 붓스트랩(bootstrap) 방법 : 파라미터 추정량의 정확성을 측정하기 위해 사용된다.
3. 시험오차가 무엇인지는 이 다음 슬라이드에서 설명.

# 오차(error)의 종류

$$
\begin{aligned}
\mathbb{E}(Y - \hat{Y})^2 &= \mathbb{E}(f(X) + \epsilon - \hat{f}(X))^2 \\
&= \mathbb{E}(f(X) - \hat{f}(X))^2 + \mathbb{V}ar(\epsilon) \\
&= \text{축소가능한 오차} + \text{축소불가능한 오차}
\end{aligned}
$$

축소가능한 오차(reducible error) : 좋은 추정 방법으로 줄일 수 있는 에러

축소 불가능한 오차(irreducible error) : 추정방법을 바꾸어서 줄일수 없는 에러

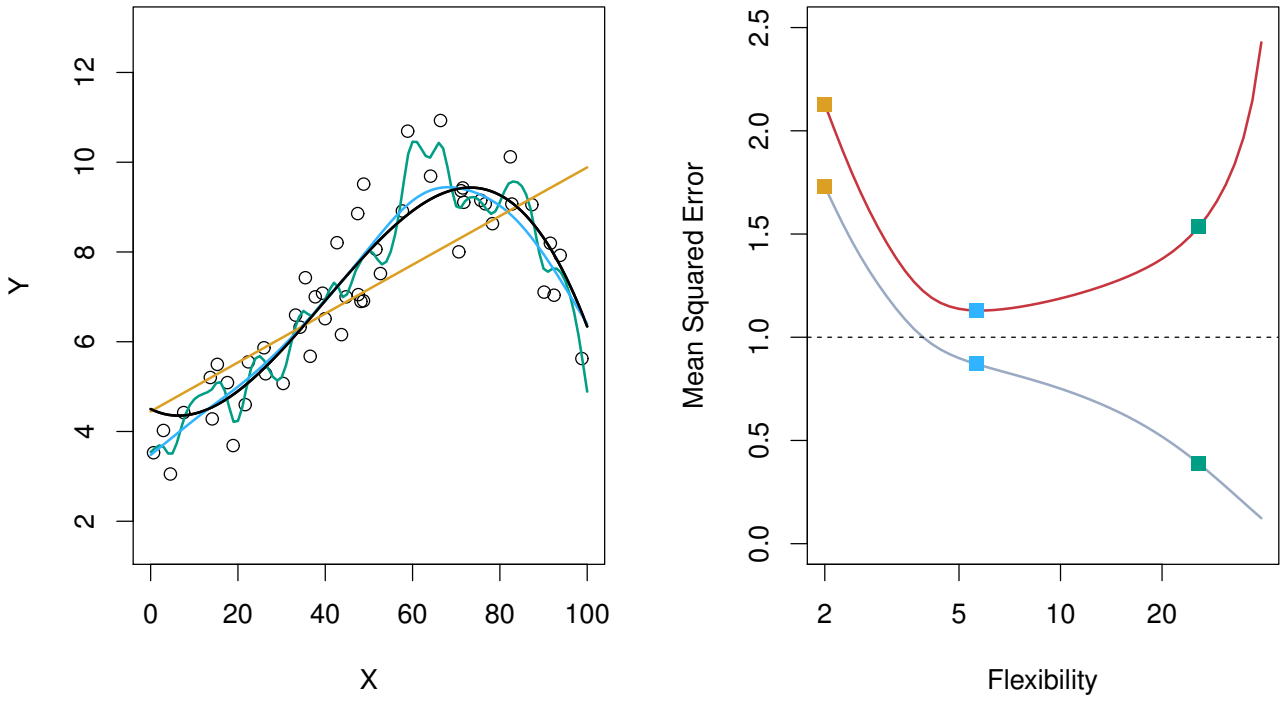
# 훈련자료와 시험자료

**훈련자료(training data)** : $f$를 추정하는데 사용되는 자료.
$(y_1, x_1), \ldots, (y_n, x_n)$, $x_i = (x_{i1}, \ldots, x_{ip})^T$

**시험자료 혹은 검증자료(test data)** : 회귀직선을 추정하는데 사용되지 않은 자료. 미래의 자료일 수도 있고, 주어진 자료 중 떼어낸 자료일 수도 있다.

# 평균제곱오차

복잡한 모형을 만들면, 현재 데이터에 딱 맞는 모형을 만들 수는 있지만,
새로운 데이터가 주어졌을 때는 오히려 MSE는 커진다.
이러한 경우를 과적합(overfitting)이라고 한다.




평균제곱오차(Mean squared error, MSE)는 오차의 크기를 재는 측도이다.

$$\text{훈련 평균제곱오차} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

$$\text{시험 평균제곱오차} = Ave(y_0 - \hat{f}(x_0))^2.$$

여기서 $(y_0, x_0)$는 미래의 관측치이다.

# 노트.

1. 시험 평균제곱오차가 더 중요하다.
2. 시험 평균제곱오차는 훈련 평균제곱오차 보다 보통 훨씬 더 크다.
3. 훈련 평균제곱오차가 작다고 시험 평균제곱오차가 작은 것은 아니다. 앞의 그림을 보면 자료를 너무 가까이 따르면 훈련 평균제곱오차는 작지만 시험 평균제곱오차는 오히려 더 커진다.
4. 그림은 유연성과 시험오차, 훈련오차의 관계를 보여준다
5. 보통 모형의 유연성이 커질수록 훈련오차는 작아지지만, 시험오차는 작아졌다가 다시 커진다. 다시 커졌을 때를 과적합이라고 한다.

# 과적합

과적합(overfitting)이란? $\hat{f}$가 자료를 너무 가까이 따라가는 경우를 말한다.

랜덤챈스(오차항)에 의한 자료의 패턴을 찾아내 $\hat{f}$라고 하는 경우이다.

이 경우 훈련평균제곱오차(training MSE)는 작지만 시험평균제곱오차(test MSE)는 커진다.

# 편향-분산 균형(Bias-variance trade-off)

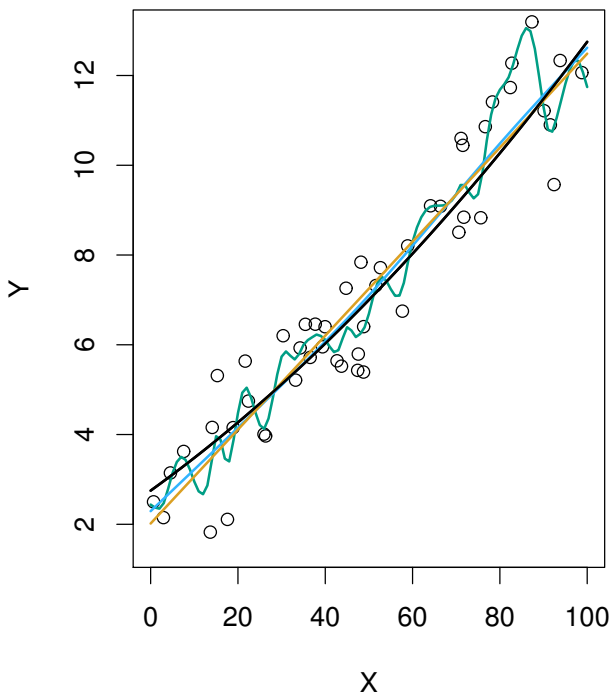
기대 시험 평균제곱오차(expected test MSE)를 분해하면,

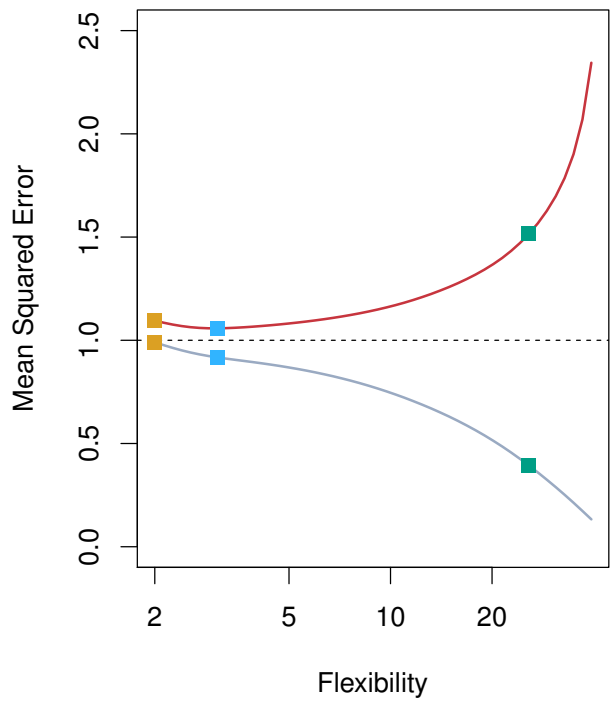
$$\begin{aligned}
\mathbb{E}(y_0 - \hat{f}(x_0))^2 &= \mathbb{E}(f(x_0) - \hat{f}(x_0) + \epsilon)^2 \\
&= \mathbb{E}\big((f(x_0) - \mathbb{E}\hat{f}(x_0)) + (\mathbb{E}\hat{f}(x_0) - \hat{f}(x_0)) + \epsilon\big)^2 \\
&= Bias(\hat{f}(x_0))^2 + \mathbb{V}ar(\hat{f}(x_0)) + \mathbb{V}ar(\epsilon).
\end{aligned}$$

1. $\mathbb{V}ar(\hat{f}(x_0))$ : 자료가 변함에 따라 $\hat{f}$이 변하는 정도를 나타낸다. 유연성(flexibility)가 큰 모형일 때 크다.
2. $Bias(\hat{f}(x_0))$ : 모형과 실제 $f$ 값의 차이를 나타낸다. 유연성이 큰 모형일 때 작다.

모형의 유연성이 커질수록 분산은 커지고, 편향은 작아진다.

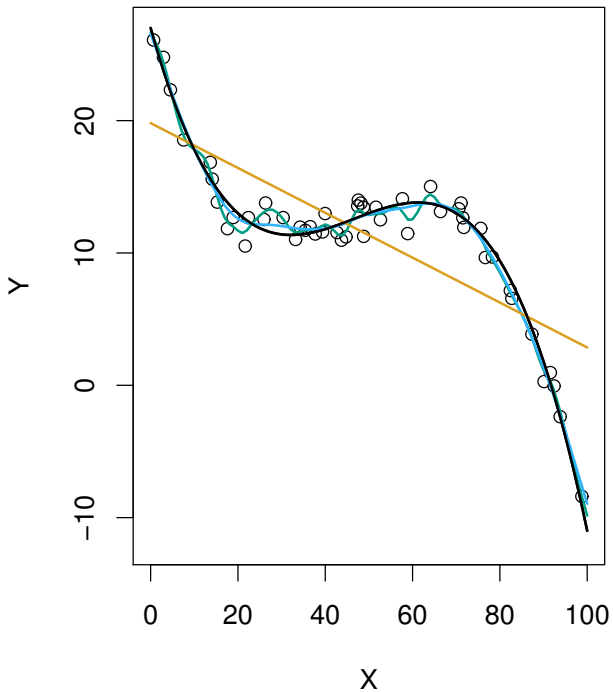
# 유연성, 시험오차, 훈련오차

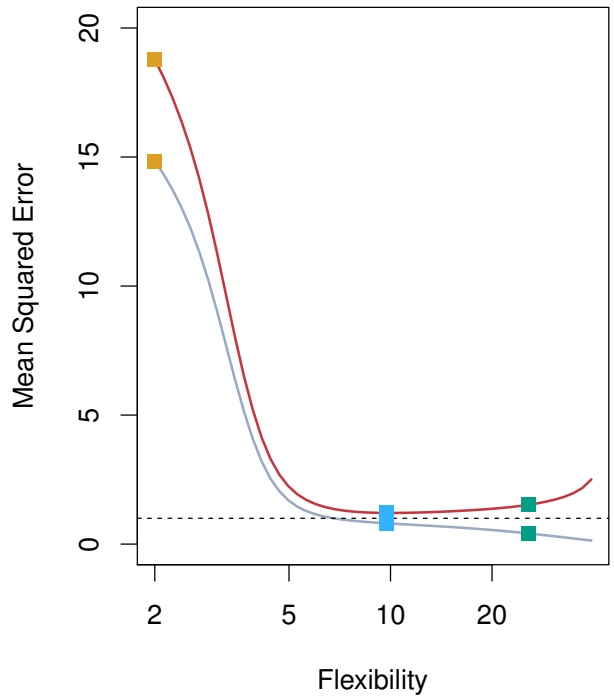


# 노트.

1. 처음 두 개의 그림 은 유연성과 시험오차(붉은색), 훈련오차(푸른색)의 관계를 보여준다.
2. 마지막 그림은 모형의 유연성과 분산, 편향, 시험오차의 관계를 보여준다.
3. 진짜 회귀함수가 직선에 가깝다. 처음 직선을 적합한 것이 참값을 잘 맞추어서 시험 mse가 분산(점선 $\sigma = 1$)과 거의 같다. 훈련 mse도 거의 1이다. 그러나 모형의 유연성이 커지며, 분산이 커지고, 시험 mse가 오히려 커진다. 이때도 훈련 mse는 계속 작아진다.

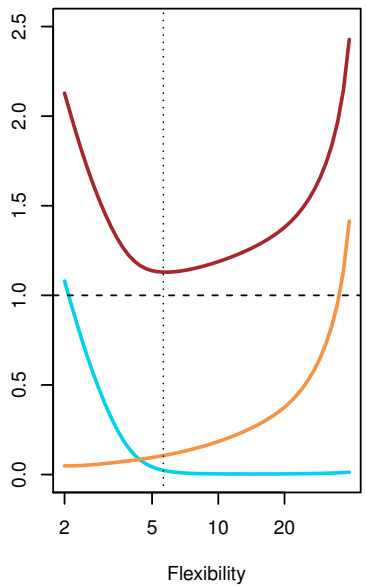

Y
20
10
0
−10
0
20
40
60
80
100
X

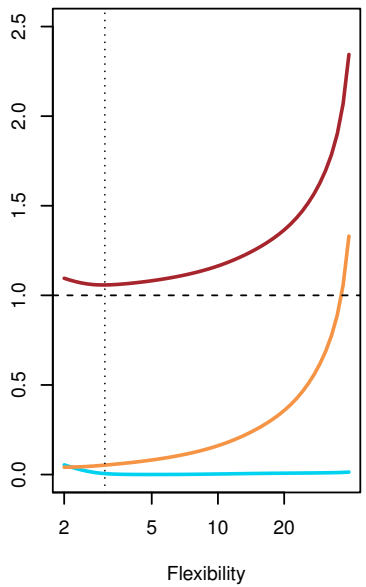

Mean Squared Error
0
5
10
15
20
2
5
10
20
Flexibility

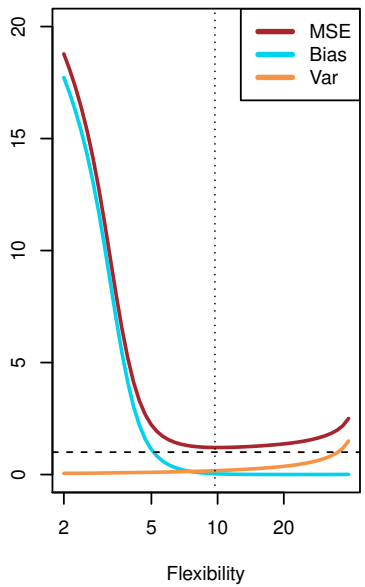
# 노트.

1. 처음 적합한 선형회귀가 참값의 변동을 못쫓아간다. 편향이 커서 시험, 훈련 mse 모두 크다. 유연성이 커지며 추정값이 참값에 가까와지고, 시험 훈련 mse 모두 작아진다.

# 유연성, 분산, 편향, 시험오차



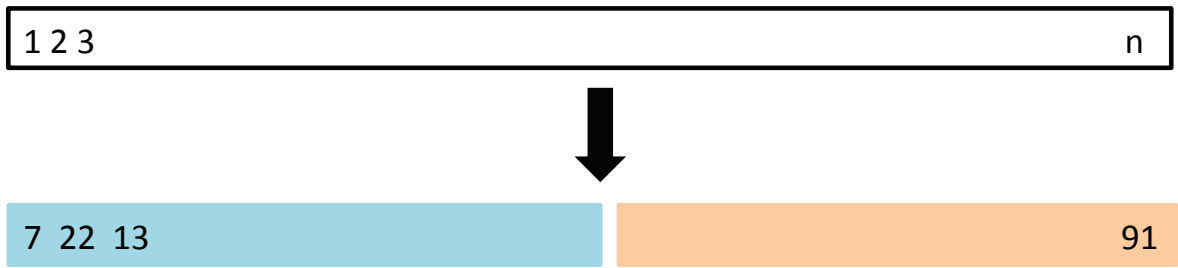



# 노트.

1. 모형의 유연성과 분산(오렌지), 편향(파란색), 시험오차(붉은색)의 관계를 보여준다.
2. 유연성이 커질수록 편향은 작아지지만 분산은 커진다.
3. 왼쪽 그림 : 참 값이 가운데 유연성 모형에서 커버가 된다. 유연성이 작을 때 모형이 충분히 크지 않아, 편향이 크고 시험 mse가 크다. 유연성이 커지면서 편향이 작아지고 시험 mse도 작아진다. 유연성이 더 커지면서 분산이 커지기 시작하며 시험 mse도 커진다.
4. 중간그림 : 참값이 낮은 유연성 모형에서 커버가 된다. 유연성이 작은 모형이 이미 작은 편향을 가지고 있다. 유연성이 커지며 분산이 커지고 시험 mse도 커진다.
5. 오른쪽 그림 : 참값이 유연성이 큰 모형에 커버가 된다. 유연성이 작은 모형은 편향이 너무 크다 따라서 시험 mse도 매우 크다. 유연성을 크게 하며 편향이 작아지고 시험 mse도 작아진다.

# 검증자료 방법(validation set approach)



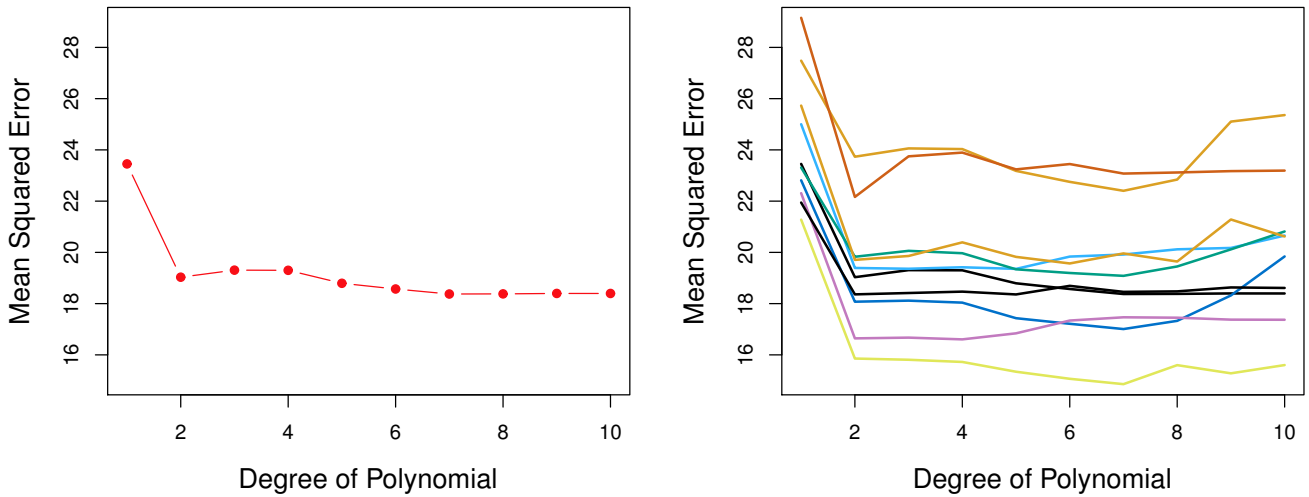

주어진 자료를 훈련자료와 검증자료로 나눈다.

훈련자료를 이용해 모형을 적합하고 타당성 자료를 이용해 적합된 모형의 시험오차를 추정한다.

# 노트.

1. 주어진 자료를 훈련자료와 검증자료로 나누는 그림.
2. 훈련자료로 mse를 계산하면 유연성이 커지면서 mse를 아주 작게 만들수 있다. 작은 mse는 모형의 유연성이 크다는 얘기이지 모형의 성능이 좋다는 얘기는 아니다. 모형을 적합한 자료로 모형을 평가하는 것은 문제를 주고 공부를 시킨 후에 그 문제로 시험을 보게해서 평가하는 것과 마찬가지 이다. 이 문제를 해결하기 위해 문제를 반만 주고 공부를 하게하고 나머지 반은 평가를 위해 남겨두는 방법이다. 주어진 자료를 반만 써서 모형을 적합하고 나머지 반은 모형을 평가하는데 쓴다.

# 검증자료 방법의 예




$$mpg = \beta_0 + \beta_1 \times horsepower + \beta_2 \times horsepower^2 + \ldots + \beta_p \times horsepower^p + \epsilon$$

검증자료 방법을 이용하여 $p$를 선택하는 예를 보여준다.

# 노트.

1. 그림은 자동차 자료에

   $$mpg = \beta_0 + \beta_1 \times horsepower + \beta_2 \times horsepower^2 + \ldots + \beta_p \times horsepower^p + \epsilon$$

   의 모형을 적합하는데, 검증자료 방법을 이용하여 $p$를 선택하는 예를 보여준다.

2. 392개의 자료를 196개의 훈련자료와 196개의 검증자료로 나누었다.
3. 왼쪽 그림은 $p$값에 따른 시험오차의 추정량을 보여준다.
4. 오른쪽 그림은 훈련자료와 검증자료의 분리를 랜덤하게 10번해서 시험오차를 추정한 것을 보여준다.
5. 1차식은 적당하지 않다는 것을 알수있다. 모든 곡선에서 1차후에 시험오차가 갑자기 작아진다.
6. $p = 2, 3, 4, 5$ 중에서 어떤 $p$가 좋은지는 명확하지 않다.
7. 패턴은 모두 비슷한데 시험오차 추정치의 값이 큰 차이를 보인다.

# 검증자료 방법의 문제점

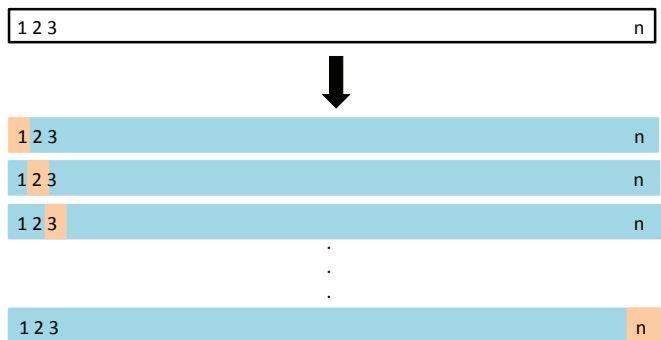
1. 어떻게 훈련자료와 검증자료로 나누었냐에 따라 시험오차 추정량의 변동이 매우 크다.
2. 훈련자료를 주어진 자료의 반만 사용했기 때문에 시험오차를 과대추정하게 된다.

# 하나남기기 교차검증(leave-one-out cross-validation, LOOCV)

**하나남기기 교차검증이란?**

1. 한개의 자료 $(x_i, y_i)$를 뺀 나머지 자료를 훈련자료로 사용하여 모형을 적합
2. 적합된 모형을 이용해 $\hat{y}_i$를 추정
3. $MSE_i := (y_i - \hat{y}_i)^2$
4. 시험오차

$$CV_{(n)} := \frac{1}{n} \sum_{i=1}^{n} MSE_i$$



# 노트.

1. **하나남기기 교차검증이란?** 한개의 자료 $(x_i, y_i)$를 뺀 나머지 자료를 훈련자료로 사용하여 모형을 적합한다. 이 모형을 이용해 $\hat{y}_i$를 추정하고

$$MSE_i := (y_i - \hat{y}_i)^2$$

이라 정의한다. 시험오차는

$$CV_{(n)} := \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

로 추정한다.

2. 그림 : 하나남기기 교차검증 방법을 위해 자료를 나누는 방식의 도식화.

# 하나남기기 교차검증의 예 : 코드

```
> library(boot)
> glm.fit=glm(mpg~horsepower,data=Auto)
> cv.err=cv.glm(Auto,glm.fit)
> cv.err$delta
[1] 24.23151 24.23114
> cv.error=rep(0,5)
> for (i in 1:5){
+  glm.fit=glm(mpg~poly(horsepower,i),data=Auto)
+  cv.error[i]=cv.glm(Auto,glm.fit)$delta[1]
+  }
> cv.error
[1] 24.23151 19.24821 19.33498 19.42443 19.03321
```

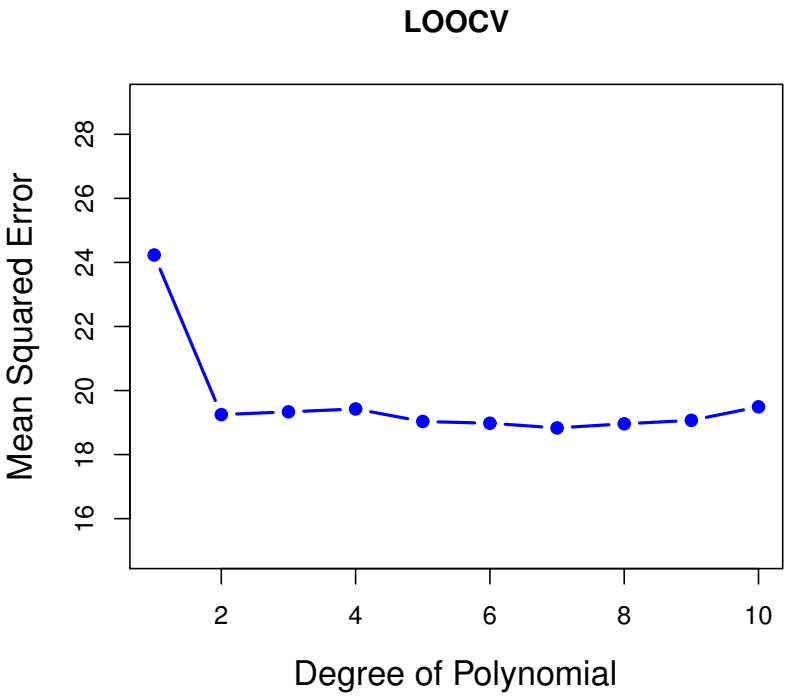
**cv.glm(data, glmfit, cost, K)**

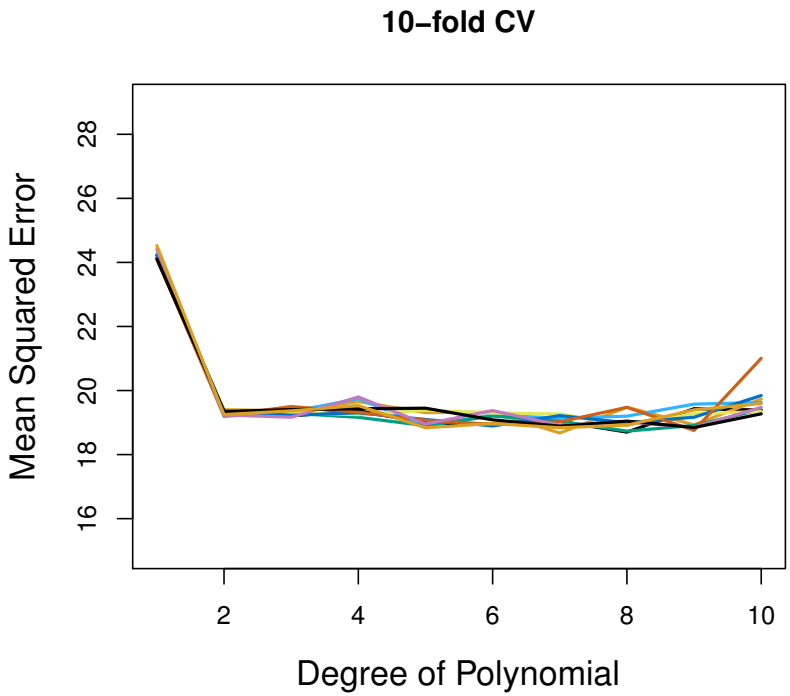
**2-standard error RULE**

# 노트.

1. lm 대신에 glm을 썼다. glm의 옵션 중에 family의 디폴트 값이 gaussian이다. 이는 정규선형회귀모형을 의미한다. lm과 동일하다. lm대신에 glm을 쓰는 이유는 cv.glm함수를 쓰기 위해서이다.
2. cv.glm은 K-겹 교차검증오차를 계산한다. cv.glm의 옵션으로 K가 있는데 이의 디폴트 값은 자료의 개수이다. 따라서 하나남기기 교차검증오차를 계산한다.
3. `cv.err$delta`는 항상 길이가 2인 벡터이다. 첫번째는 예측오차의 교차검증추정량이다. 두번째 것은 하나남기기 교차검증을 사용하지 않을 때 생기는 편이를 교정한 값이다. 하나남기기 교차검증을 사용하지 않을 때는 편이가 있다는 얘기다. 여기서는 하나남기기 방법을 사용했기 때문에 두 개의 값이 거의 동일하다. 같지는 않은 모양이다.
4. 다항회귀의 차수별로 하나남기기 교차검증으로 추정한 예측오차를 구한다.

# 하나남기기 교차검증의 예



# 노트.

1. 자동차 자료에 $p$차 다항회귀를 적합할 때 $p$에 따른 시험오차 추정값들을 보여준다. 왼쪽 그림 : 하나남기기 교차검증 방법으로 추정한 시험오차. 오른쪽 그림 : 10겹 교차검증 방법으로 추정한 시험오차.
2. **하나남기기 교차검증 방법의 장점** 검증자료 방법의 문제점을 해결한다.
   - 2.1 검증자료 방법에 비해 편향이 매우 적다.
   - 2.2 검증자료의 임의성에 의한 변동성이 없다.

# 교차검증 공식

선형모형의 경우

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2 .$$

$\hat{y}_i$는 전체자료를 이용해 적합한 후 예측한 $y_i$의 값

$h_i$는 $i$번째 관측치의 지렛대 통계량

# 노트.

1. n이 매우 크고 모형적합의 속도가 느리면 하나남기기 교차검증 방법의 계산량이 매우 커질 수 있다. 선형모형의 경우 다음의 간단한 식으로 한 번의 적합을 통해 시험오차 추정량을 구할 수 있다.
2. 여기서 $\hat{y}_i$는 전체자료를 이용해 적합한 후 예측한 $y_i$의 값이고, $h_i$는 $i$ 번째 관측치의 지렛대 통계량이다. i번째 관측치의 지렛대 통계량은 햇형렬의 i번째 대각원소이다.

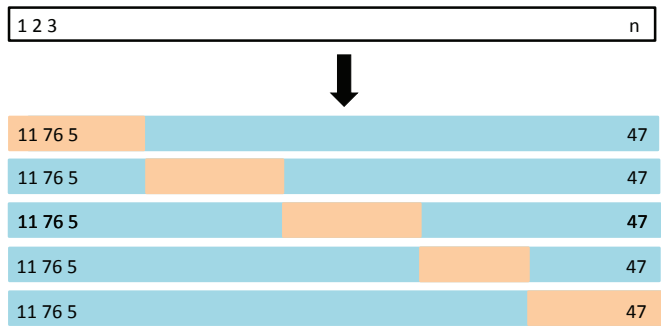
# k-겹 혹은 k-묶음 교차검증법(k-fold cross-validation)

LOOCV는 CV를 n번 해야 한다. 하지만, 큰 데이터가 주어졌을 때는 계산의 어려움이 발생할 수 있다.
그래서 고안된 것이 k-fold CV이다.

1. 전체 자료를 $k$ 개의 묶음으로 나눈다.
2. i번째 묶음을 제외하고 나머지 자료로 모형을 적합하고
3. 적합한 모형으로 i번째 묶음의 자료를 예측하고 실제 관측치와 비교하여 평균제곱오차 $MSE_i$를 구한다.
4. 시험오차추정량은

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

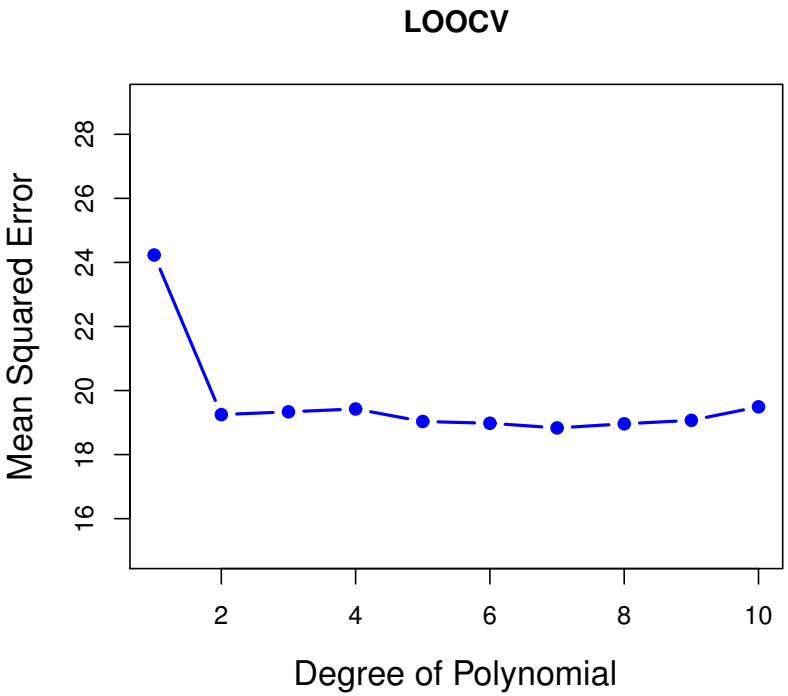
가 된다.

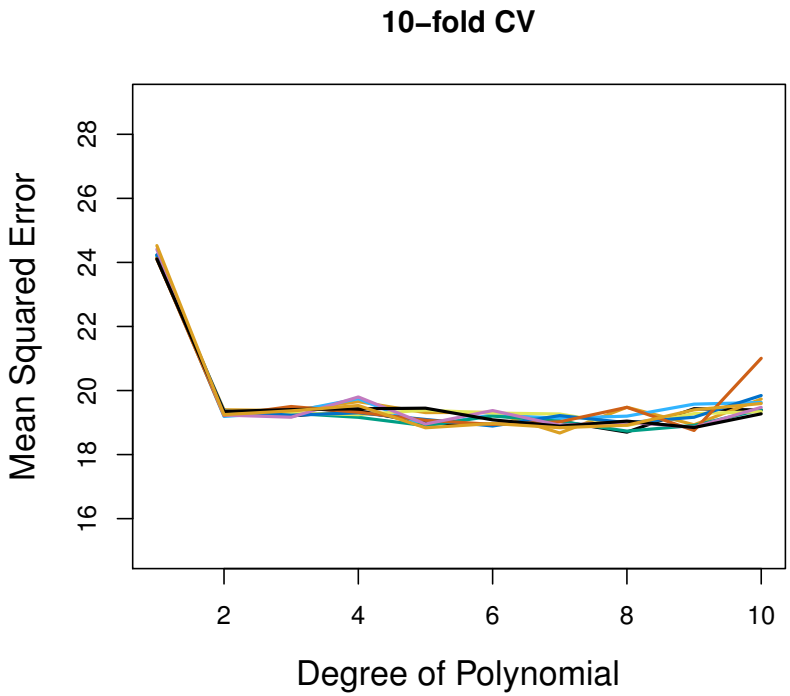


# 노트.

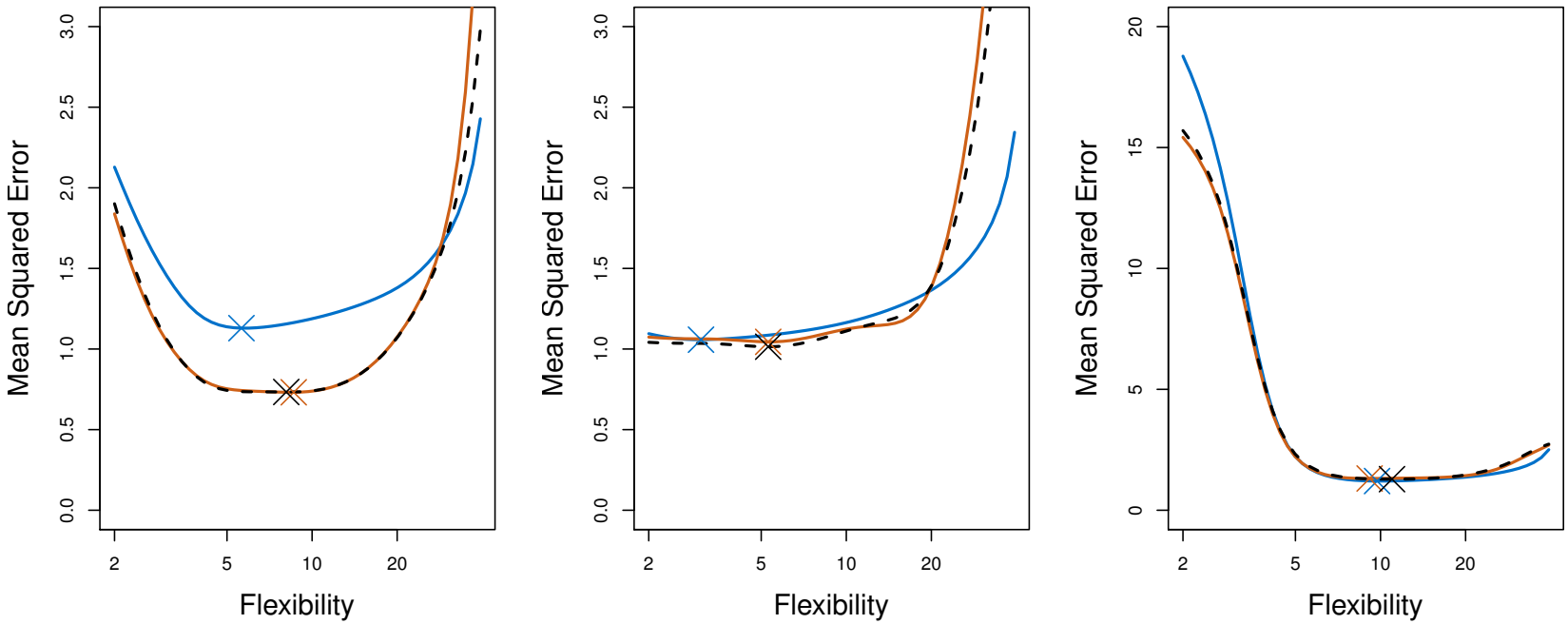
1. 하나남기기 교차검증은 k겹 교차검증의 특별한 경우이다.
2. 그림은 k겹 교차검증을 위해 자료를 나누는 방식의 도식화.
3. $k = 5, 10$을 많이 쓴다.
4. 하나남기기 교차검증법보다 계산량이 엄청 적다.
5. 시험오차 추정량의 묶음의 랜덤성에 의한 변동이 매우 작다. 다음 쪽의 그림의 오른쪽 그림은 10겹 교차검증을 9번한 결과를 보여준다. 한 번 교차검증을 할 때마다 10 묶음은 램덤하게 정한 것이다. 9번 모두 매우 비슷한 시험오차 추정량을 보여준다. 시험오차 추정량의 묶음의 랜덤성에 의한 변동이 매우 작다는 것을 보여준다.

# k-겹 교차검증법의 예


LOOCV
Mean Squared Error
16
18
20
22
24
26
28
2
4
6
8
10
Degree of Polynomial


10−fold CV
Mean Squared Error
16
18
20
22
24
26
28
2
4
6
8
10
Degree of Polynomial

# 모의실험




1. 세가지 모형에 대한 모의실험결과.
2. 진짜 시험평균제곱오차(파란색), 하나남기기 방법에 의한 시험오차추정량(검은색 대쉬라인), 10겹 교차검증에 의한 시험오차추정량(붉은색)을 보여준다.

# 노트.

1. 세 가지 모형에 의한 모의실험결과를 보여준다.
2. 진짜 시험평균제곱오차(파란색), 하나남기기 방법에 의한 시험오차추정량(검은색 대쉬라인), 10겹 교차검증에 의한 시험오차추정량(붉은색)을 보여준다.
3. 왼쪽그림은 교차검증추정량들이 진짜시험오차를 과소추정하고, 나머지 두개는 잘 추정한다고 보인다.
4. 모든 경우에 하나남기기 교차검증과 10겹 교차검증 추정량은 거의 동일하다. 10겹 교차검증이 계산량이 작으니 10겹 추정량을 쓰는것이 좋다는 결과이다.
5. 진짜 시험오차를 과소추정하기도 하지만 그 때도 어디서 시험오차가 최소가 되는지는 비슷하다. 모형선택에서는 값자체보다는 최소가 되는 지점이 중요하다.

# k겹 교차검증의 편향-분산 균형

1. 하나남기기 교차검증의 시험오차 추정량은 k겹 교차검증의 추정량보다 편향이 작다.
2. k겹 교차검증 추정량은 하나남기기 방법의 추정량보다 분산이 작다.
3. 경험적으로 5겹, 10겹 교차검증이 평균제곱오차 관점에서 하나남기기 방법보다 좋은 것으로 알려져 있다.

# 노트.

1. 하나남기기 교차검증의 시험오차 추정량은 k겹 교차검증의 추정량보다 편향이 작다. n개의 자료를 적합하는 모형을 하나남기기 방법은 n-1개의 자료를 적합한 모형으로 흉내내고, k겹 교차검증 추정량은 n-k개의 자료를 적합한 모형으로 흉내내기 때문이다.
2. k겹 교차검증 추정량은 하나남기기 방법의 추정량보다 분산이 작다. 하나남기기 방법의 추정량은 n개의 값을 평균내서 구하고, k겹 추정량은 $n \times k$개의 값을 평균을 내기 때문이다.
3. 경험적으로 5겹, 10겹 교차검증이 평균제곱오차 관점에서 하나남기기 방법보다 좋은 것으로 알려져 있다.

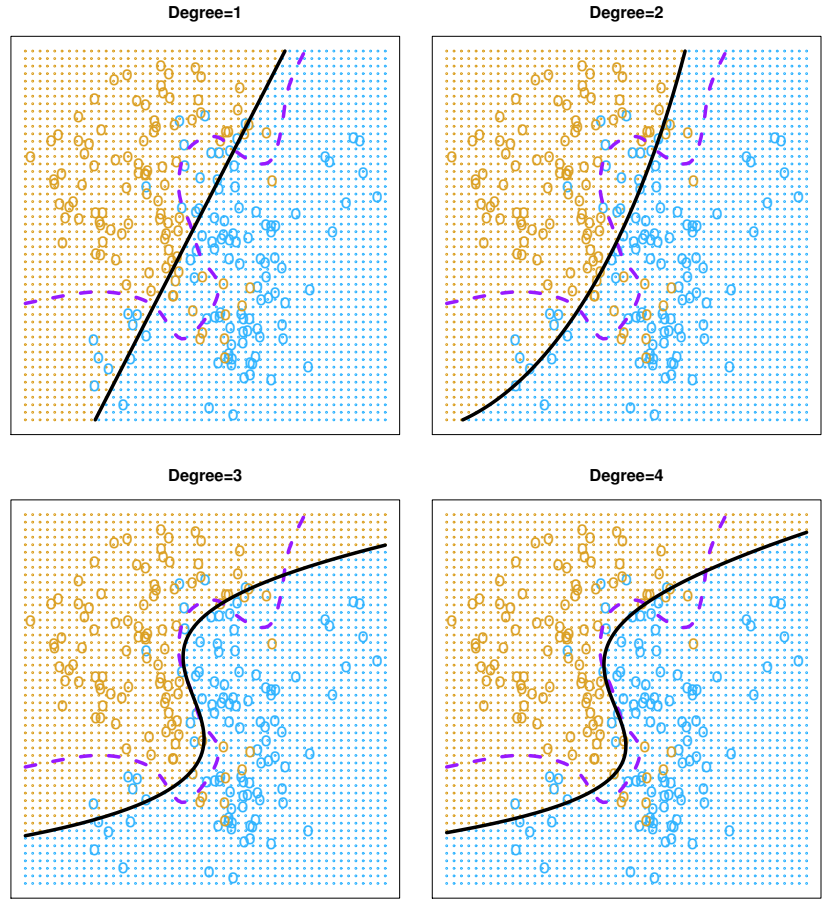
# 분류에서 교차검증

그럼 도대체 얼마나 복잡한, 또는 간단한 모형을 쓸 것인가? 생각해본다면, CV가 가장 작은 모형을 쓰면 좋다.

하나남기기 추정량은

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} Err_i, \;\; Err_i = I(y_i \neq \hat{y}_i)$$

와 같이 정의된다. 여기서 $\hat{y}_i$는 $y_i$를 빼고 적합한 모형으로 예측한 $y_i$ 값이다. k겹 교차검증 추정량도 동일하게 정의된다.



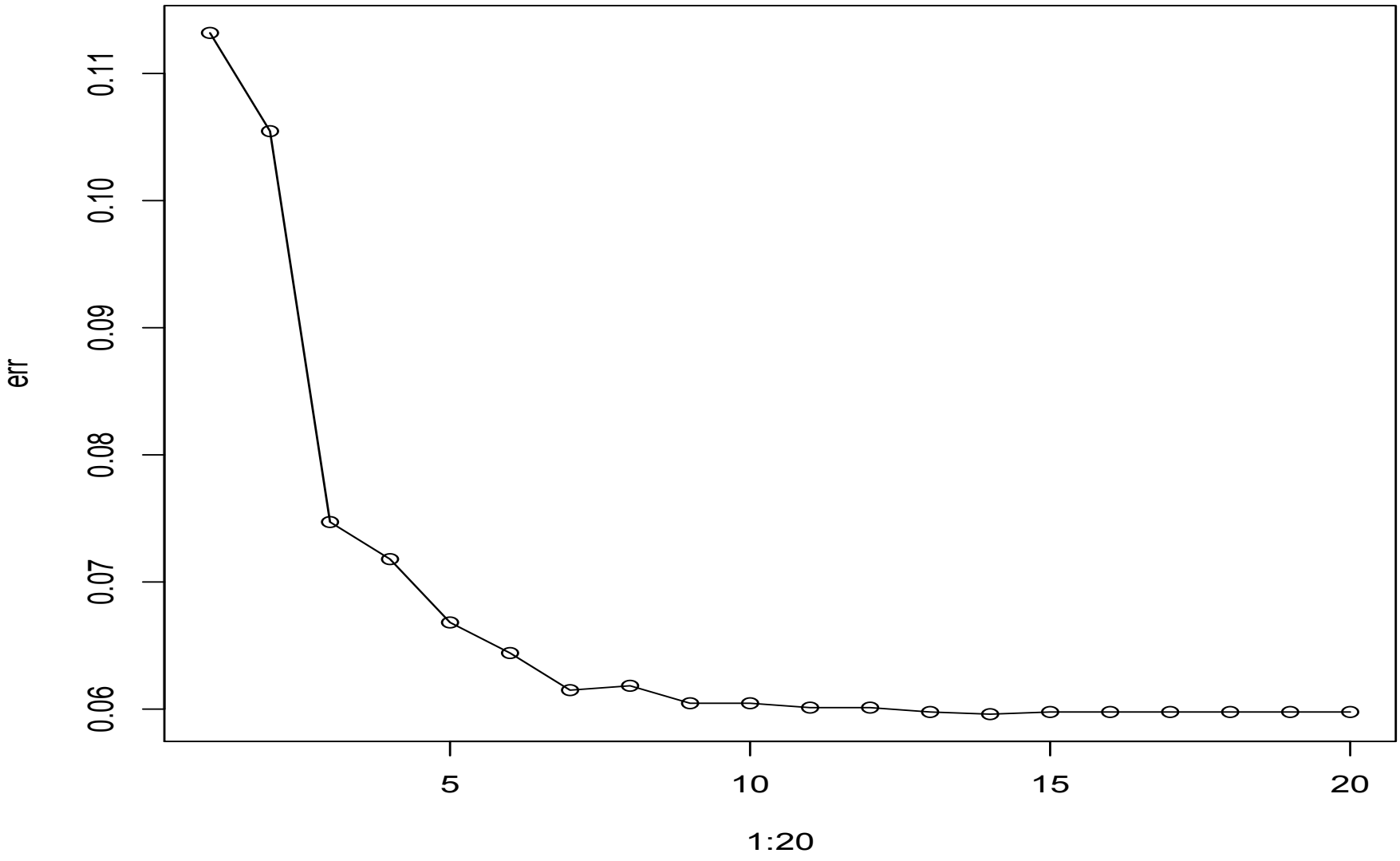
# 노트.

1. 그림. 1차 – 4차 로지스틱회귀모형으로 적합한 분류의 경계를 보여준다. 점선은 베이즈 결정 경계이다. 네 모형의 시험오차는 각각 0.201, 0.197, 0.160, 0.162이다. 베이즈분류의 시험오차는 0.133이다.

# KNN의 K값을 CV를 이용하여 정하기

```
library(ISLR)
library(class)
dim(Caravan)
attach(Caravan)
summary (Purchase)
allX=scale(Caravan[,-86])
allY=Purchase
n=length(allY)
#
# 6 fold cross-validation을 통하여 K를 정하는 R code
#
#  n=5822, p=85
#

misclass=matrix(0,20,6)
```

```
for(l in (1:20)){
    for(m in (1:6)){
        start=(m-1)*1000+1
        end=m*1000
        if(m==6){end=n}
        test.X=allX[start:end,]
        test.Y=allY[start:end]
        train.X=allX[-(start:end),]
        train.Y=allY[-(start:end)]
        set.seed (1)
        knn.pred=knn(train.X,test.X,train.Y,k=l)
        misclass[l,m]=sum(test.Y!=knn.pred)
    }
    cat("\n")
    cat(l)
}

err=apply(misclass,1,sum)/n
plot(1:20,err,type="o")
```


err
0.06
0.07
0.08
0.09
0.10
0.11
5
10
15
20
1:20

# 붓스트랩

## 투자분할의 예

- 두 개의 투자 방법이 있다. 백만원을 투자하면 1년후에 회수금이 각각 X원 Y원이 된다. X와 Y는 랜덤이다.
- 투자를 $\alpha$와 $1-\alpha$의 비율로 하면 회수금은 $\alpha X + (1-\alpha)Y$가 된다.
- 투자의 위험은 회수금의 분산으로 정의된다. 위험을 최소화하는 $\alpha$를 구하고자 한다.
- $\alpha$를 $\mathbb{V}ar(\alpha X + (1-\alpha)Y)$를 최소화하도록 구하면

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

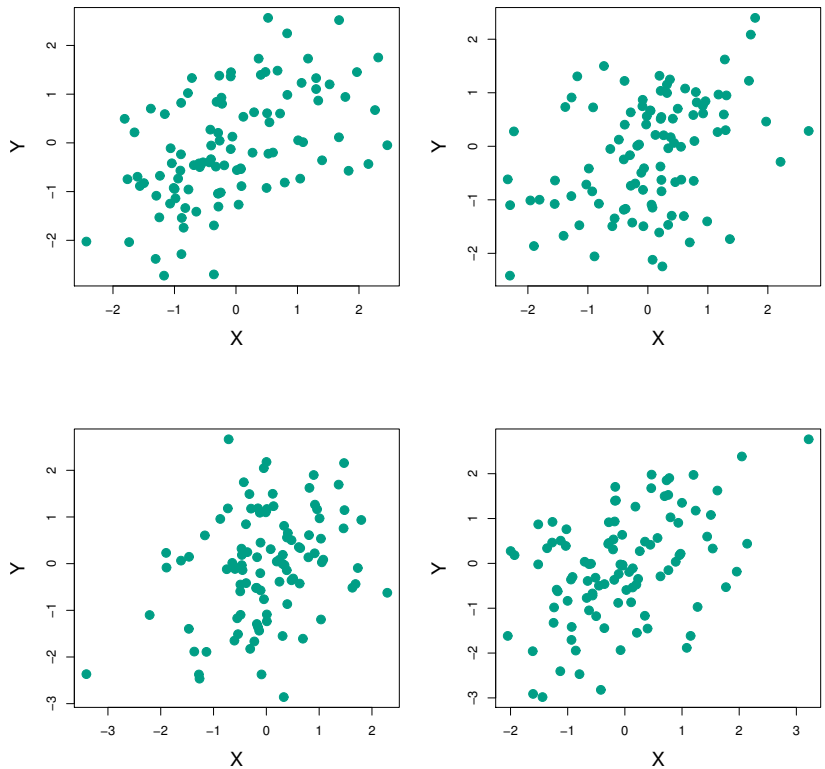
  가 된다.
- 과거의 자료($n = 100$)가 있다면 $\hat{\sigma}_X, \hat{\sigma}_Y, \hat{\sigma}_{XY}$를 구하고

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

  을 이용하면 된다.
- $SE(\hat{\alpha})$을 구하고자 한다.

# $SE(\hat{\alpha})$을 구하는 방법 : 모집단을 아는 경우

1. $(X, Y)$의 분포에서 크기 $n = 100$의 표본을 $M$개 발생시킨다. 각 표본을 이용해 $\hat{\alpha}_r$, $m = 1, 2, \ldots, M$을 구한다.
2. $\hat{\alpha}_m$들의 표준편차로 $SE(\hat{\alpha})$를 근사한다.
3. 오른쪽 그림에서 각 표본을 이용해 구한 $\hat{\alpha}$은 각각 0.576, 0.532, 0.657, 0.651이다.
4. 1000개의 표본을 발생시켜 $\hat{\alpha}_r$, $r = 1, 2, \ldots, 1000$을 구하고 다음을 구하였다. 참고로 $\alpha$의 참값은 0.6이다.




$$\bar{\alpha} = \frac{1}{1000}\sum_{r=1}^{1000}\hat{\alpha}_r = 0.5996$$

$$SE \approx \sqrt{\frac{1}{1000-1}\sum_{r=1}^{1000}(\hat{\alpha}_r - \bar{\alpha})^2} = 0.083.$$

# 노트.

1. 만약 우리가 $(X, Y)$의 분포를 정확히 알고 있다면 $SE(\hat{\alpha})$를 구할 수 있다. 여러 가지 방법이 있겠지만, 표본추출을 이용해 다음과 같이 구할 수 있다.
2. 모집단에서 발생시킨 4개의 표본 그림이다. 각 표본을 이용해 구한 $\hat{\alpha}$은 각각 0.576, 0.532, 0.657, 0.651이다.
3. 하지만 이 방법으로 $\bar{\alpha}$와 $SE(\hat{\alpha})$을 구할 수는 없다. 모집단을 모르기 때문이다.

# $SE(\hat{\alpha})$을 구하는 방법 : 붓스트랩 방법

## 붓스트랩 알고리듬

$Z$를 크기 $n$인 주어진 표본이라 하자. 이를 이용해 $\hat{\alpha}$을 구했다. $SE(\hat{\alpha})$을 구하고 싶다.

Step 1. $r = 1, 2, \ldots, B$에 대해,

- Step 1..1 붓스트랩 표본 $Z_r^*$를 추출한다. $Z_r^*$은 $Z$로부터 복원추출로 추출한 크기 $n$의 표본이다.
- Step 2..2 $Z_r^*$를 이용해 $\hat{\alpha}$을 구하고 이를 $\hat{\alpha}_r^*$이라 한다.

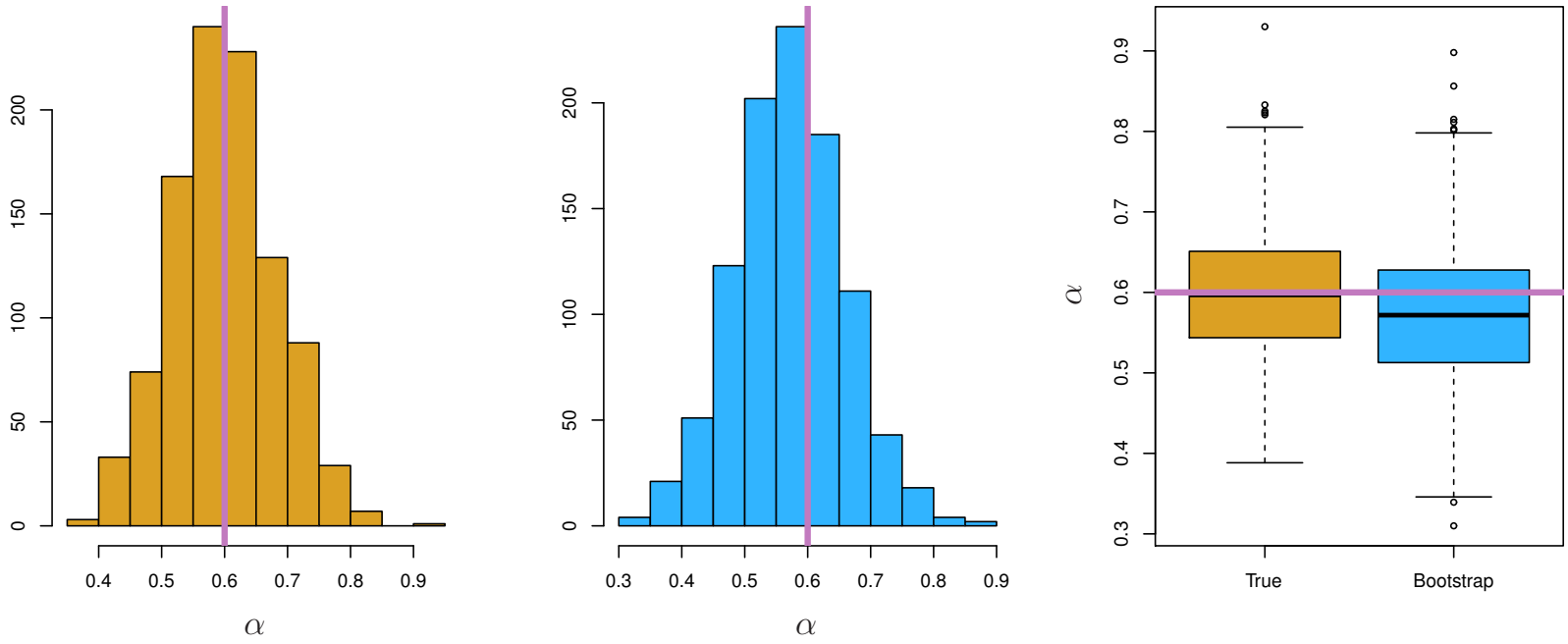
Step 2. $SE(\hat{\alpha})$의 붓스트랩 추정량은

$$\hat{SE}(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\alpha}_r^* - \bar{\alpha}^*)^2}, \quad \bar{\alpha}^* = \frac{1}{B}\sum_{r=1}^{B}\hat{\alpha}_r^*.$$

# 노트.

1. 모집단을 아는 경우는 모집단에서 1000개의 표본을 발생시켜 $\hat{\alpha}_r$값을 구했다. 붓스트랩 방법은 모집단에서 표본을 추출하는 대신, 주어진 표본을 모집단이라 생각하고, 표본에서 복원추출로 1000개의 표본을 재추출한다. 복원추출로 재추출된 표본을 붓스트랩 표본이라 한다. 붓스트랩 표본으로 $\hat{\alpha}_r$을 구해서 $SE(\hat{\alpha})$을 구한다.

# 모집단을 알 때와 모를 때 비교




왼쪽 그림 : 모집단에서 추출한 표본으로 구한 $\hat{\alpha}$의 히스토그램.

중간 그림 : 붓스트랩표본으로 구한 $\hat{\alpha}$의 히스토그램.

오른쪽 그림 : 두 $\hat{\alpha}$ 표본들의 상자그림.

# 노트.

1. 그림은 모집단에서 추출한 표본으로 구한 $\hat{\alpha}$의 히스토그램과 붓스트랩표본으로 구한 $\hat{\alpha}$의 히스토그램을 비교한다. 두 개의 분포가 거의 비슷한 것을 알 수 있다.

# 붓스트랩 코드

```
> str(Portfolio)
'data.frame': 100 obs. of  2 variables:
 $ X: num  -0.895 -1.562 -0.417 1.044 -0.316 ...
 $ Y: num  -0.235 -0.885 0.272 -0.734 0.842 ...

 > alpha.fn=function(data,index){
+  X=data$X[index]
+  Y=data$Y[index]
+  return((var(Y)-cov(X,Y))/(var(X)+var(Y)-2*cov(X,Y)))
+  }
> boot(Portfolio, alpha.fn, R=1000)
ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = Portfolio, statistic = alpha.fn, R = 1000)

Bootstrap Statistics :
     original      bias    std. error
t1* 0.5758321 0.002705445  0.09197062
```

# 노트.

1. alpha.fn 함수는 X와 Y를 콤포넌트로 갖고 있는 data와 data의 관측치 중 일부분의 인덱스를 나타내는 index를 받아들여서, 그 인덱스에 해당하는 자료만을 이용하여

$$\frac{\mathbb{V}ar(Y) - \mathbb{C}ov(X,Y)}{\mathbb{V}ar(X) + \mathbb{V}ar(Y) - 2\mathbb{C}ov(X,Y)}$$

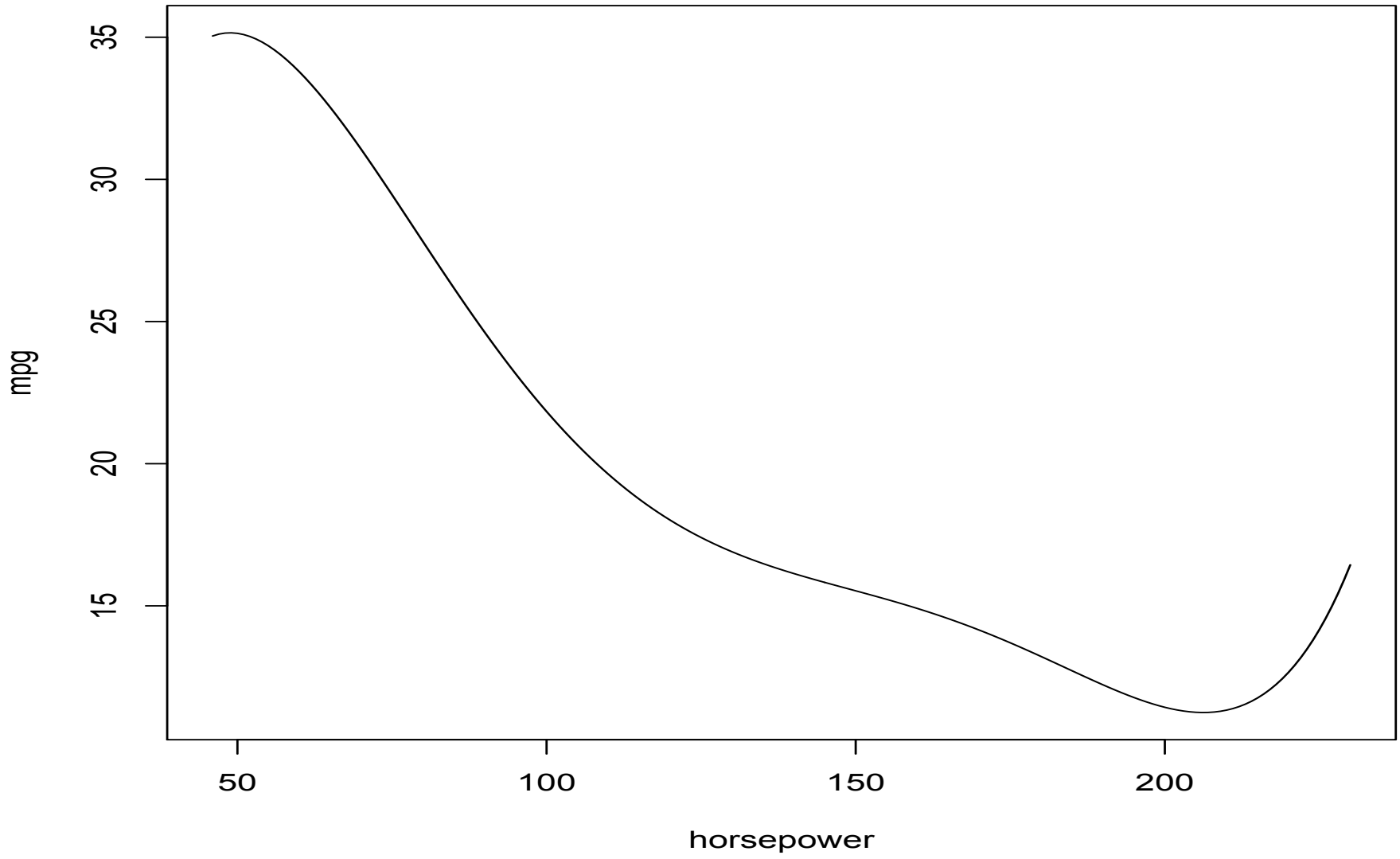
   를 반환한다.
2. Portfolio는 X, Y를 콤포넌트로 갖고 있는 데이터프레임이다.
3. boot은 자료 Portfolio에 통계량 alpha.fn의 붓스트랩을 적용하여 bias와 표준오차를 구한다.

# 보충 R 코드 pdf 파일

1. 포트폴리오
2. **붓스트랩을 이용해서 선형모형의 정확도를 구하는 코드**

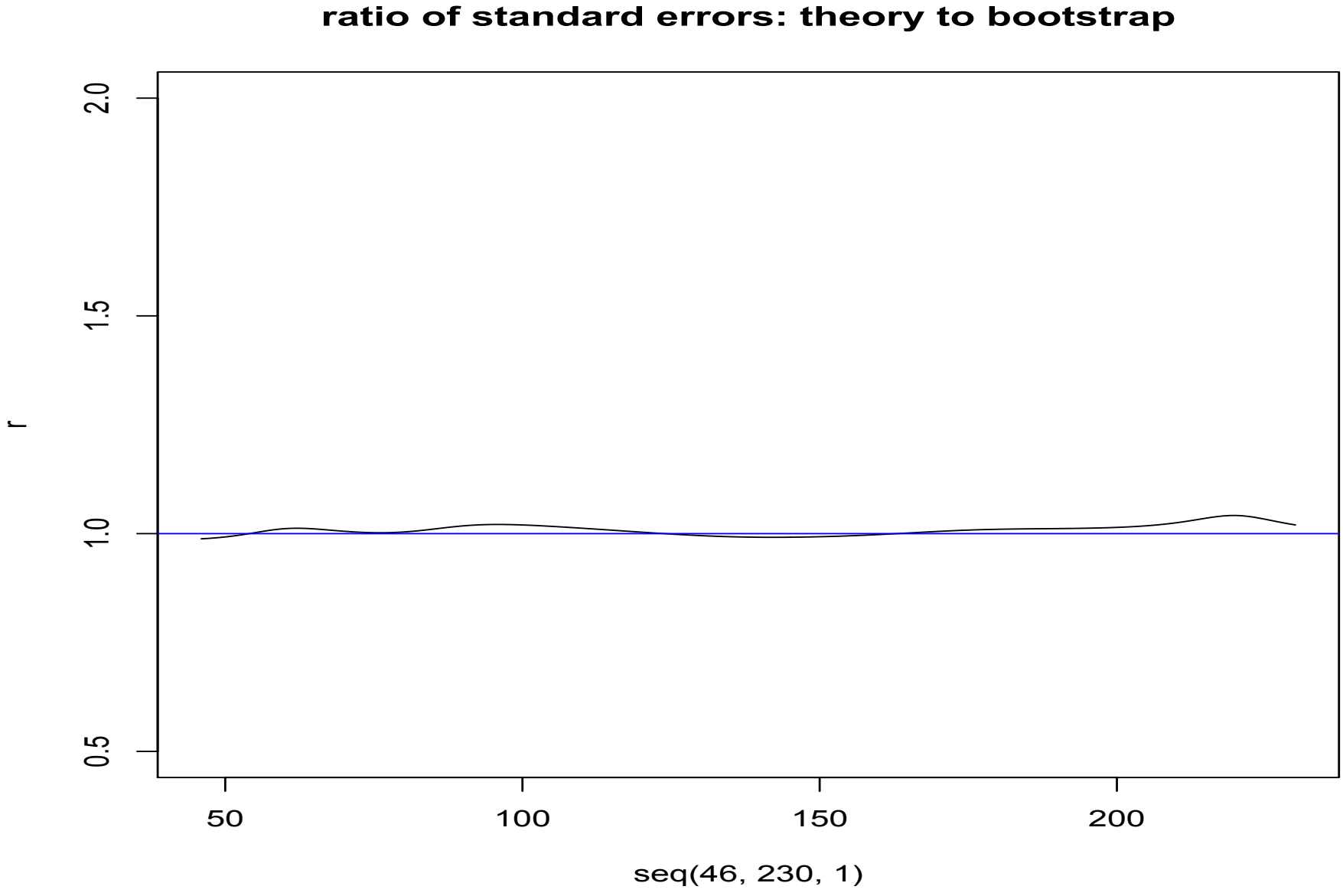
# 회귀분석에서의 붓스트랩: confidence band

```
library(ISLR)
library(boot)
head(Auto)
attach(Auto)

mpg.fit=lm(mpg~poly(horsepower,5))
pred.x=data.frame(horsepower=seq(46,230,1))
mpg.pred=predict.lm(mpg.fit,newdata=pred.x,se.fit=T)
plot(seq(46,230,1),mpg.pred$fit,type="l",
  xlab="horsepower",ylab="mpg")

#
#
# Two ways of bootstrap/ naive and residual bootstrap
# Here, residual bootstrap
#
#

oresid=mpg.fit$resid
ofit=mpg.fit$fitted
betaest=mpg.fit$coefficients
n=length(oresid)
```

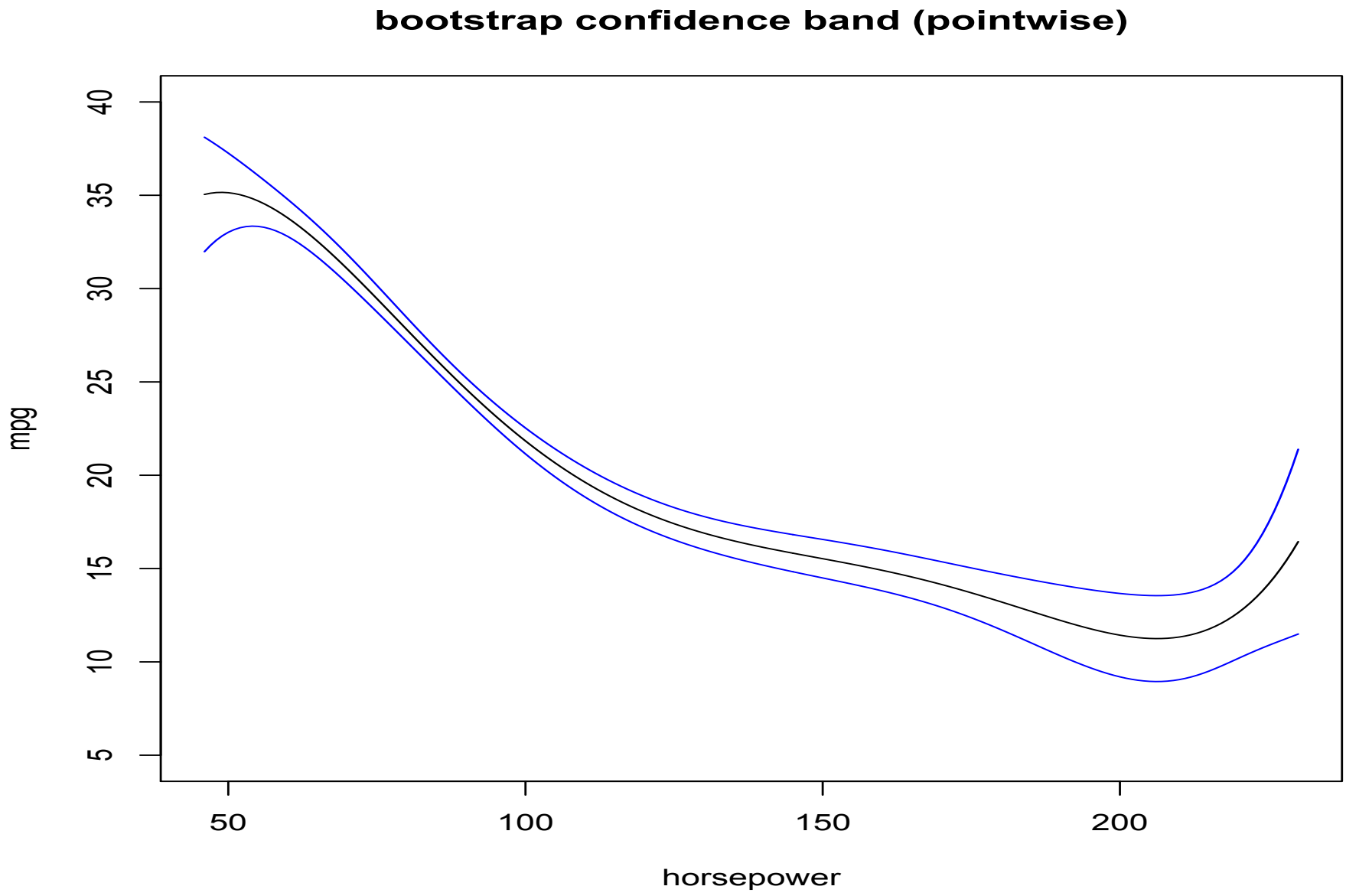

mpg
35
30
25
20
15
50
100
150
200
horsepower

```
B=1000
d=length(seq(46,230,1))
bootfit=matrix(0,B,d)
bootsamp=bootresid=rep(0,n)

for(b in (1:B)){

     sampid=sample(n,n,replace=T)
     bmpg=ofit+oresid[sampid]
     bmpg.fit=lm(bmpg~poly(horsepower,5))
     bmpg.pred=predict.lm(bmpg.fit,newdata=pred.x,se.fit=T)
     bootfit[b,]=as.numeric(bmpg.pred$fit)
        cat("\n")
     cat(b)
}
```

```
bse.fit=sqrt(apply(bootfit,2,var))
ose.fit=mpg.pred$se.fit
r=ose.fit/bse.fit
plot(seq(46,230,1),r,type="l",ylim=c(0.5,2))
abline(h=1,col="blue")
title("ratio of standard errors: theory to bootstrap")
```


ratio of standard errors: theory to bootstrap
r
2.0
1.5
1.0
0.5
50
100
150
200
seq(46, 230, 1)

```
upper=mpg.pred$fit+1.96*bse.fit
lower=mpg.pred$fit-1.96*bse.fit

plot(seq(46,230,1),mpg.pred$fit,type="l",xlab="horsepower",
  ylab="mpg",ylim=c(5,40))
lines(seq(46,230,1),upper,col="blue")
lines(seq(46,230,1),lower,col="blue")
title("bootstrap confidence band (pointwise)")
```


bootstrap confidence band (pointwise)
mpg
5
10
15
20
25
30
35
40
50
100
150
200
horsepower

# 참고문헌

아래의 책에서 제공하는 그림들을 사용하였다.


1. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*. Springer, 2013.