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 13

## Resampling Methods (ISL Chapter 5)

### 1. The bootstrap

(a) The *bootstrap* is a widely applicable and extremely powerful statistical tool that can quantify the uncertainty associated with a given estimator or statistical learning method.

(b) The name of "bootstrap" comes from idiom "pulling oneself up by one's bootstraps" which means to improve one's situation or succeed through one's own efforts, without outside help.

©https://en.wikipedia.org/

### 2. Toy Example

(a) Determine the best investment allocation

- A fixed sum of money invested in two financial assets that returns $X$ and $Y$, where $X$ and $Y$ are *random quantities* – $\alpha$ of money invested in $X$, and remaining $(1-\alpha)$ in $Y$
- Since there is *variability* associated with the returns on these two assets, we wish to choose α to minimize the total risk, or variance $\mathrm{Var}(\alpha X + (1-\alpha)Y)$, of our investment. $\alpha$ is given by

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

where $\sigma_X^2 = Var(X)$, $\sigma_Y^2 = Var(Y)$, $\sigma_{XY} = COV(X,Y)$.

- However, in reality, these quantities are *unknown.*

(b) Solution A: Simulation

- Using simulation of 100 set of $X$ and $Y$ for 1,000 times, estimations for $\alpha$ and standard deviation of the $\alpha$ could be evaluated using

$$\alpha = \frac{1}{1{,}000}\sum_{r=1}^{1{,}000}\widehat{\alpha_r}$$

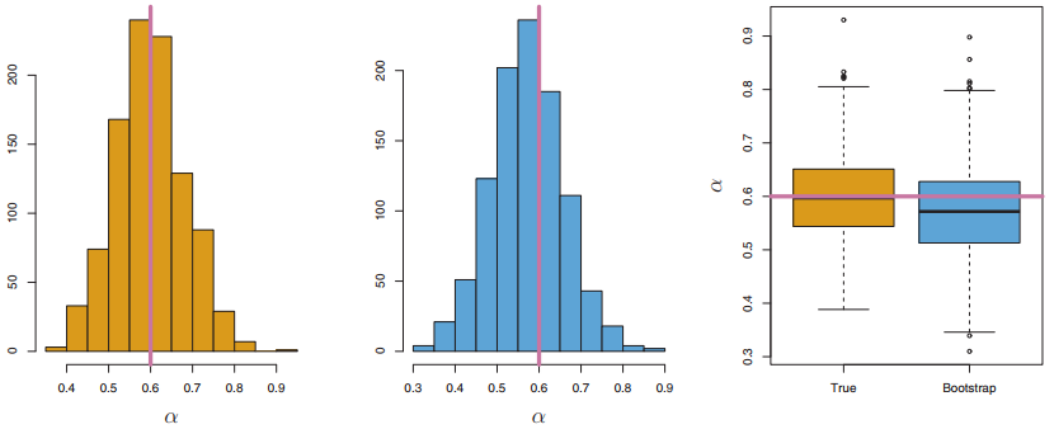
$$\mathrm{SE}(\hat{\alpha}) \approx \sqrt{\frac{1}{1{,}000-1}\sum_{r=1}^{1000}(\widehat{\alpha_r} - \bar{\alpha})^2}$$

- However, in reality, for real data we cannot *generate new samples* from the original population.

Instructor: Junho Song
junhosong@snu.ac.kr

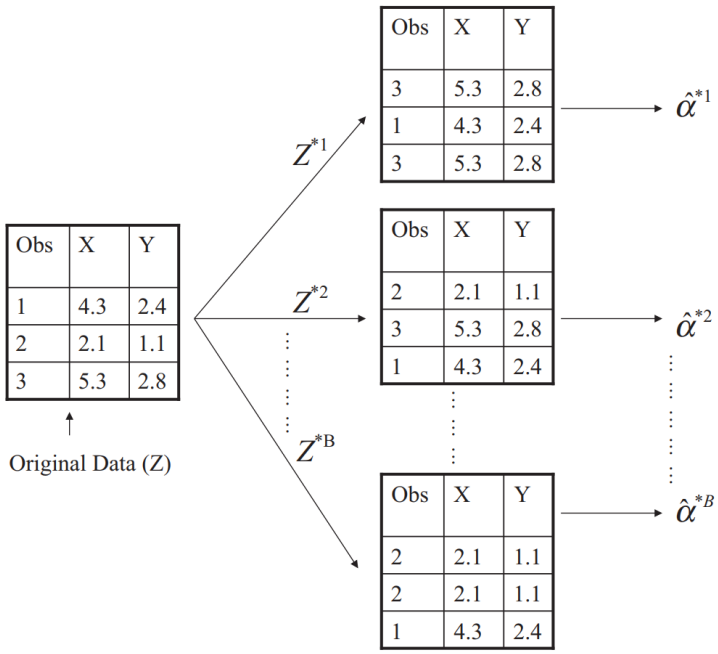
(c) Solution B: The bootstrap approach

- We can estimate the variability of $\hat{\alpha}$ without generating additional samples. Obtaining distinct data sets by *repeatedly sampling* observations *from the original data set* (with multiple selections of a sample allowed).




*In each panel, the pink line indicates the true value of α. (Left) Solution A: a histogram of the estimates of α from 1,000 simulated data (Center) Solution B: a histogram of the estimates of α obtained from 1,000 bootstrap samples (Right)*

(d) A small sample containing *n* = 3 observations.




*A graphical illustration of the bootstrap approach on a small sample containing n = 3 observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set and used to obtain an estimate of α.*

Instructor: Junho Song
junhosong@snu.ac.kr

- Randomly select *n* observations from the data set in order to produce a bootstrap data set, $Z^{*1}$. The sampling is performed with *replacement*.
- **Note** that both its $X$ and $Y$ values are included in observations.
- *Standard error* of the bootstrap estimates: We can use $Z^{*1}$ to produce a new bootstrap estimate for $\alpha$, which we call $\hat{\alpha}^{*1}$. Repeating such procedure *B* times and the estimate of the standard error of $\hat{\alpha}$ could be evaluated using

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}$$

(e) **R Example 1:** Estimating the accuracy of the statistic of Interest

```
library(ISLR) # to use 'Portfolio' & 'Auto' data
library(boot) # to use 'boot' function for bootstrap
alpha.fn=function(data,index){ # defined function
  X=data$X[index]
  Y=data$Y[index]
  return((var(Y)-cov(X,Y))/(var(X)+var(Y)-2*cov(X,Y)))} # the value of
the evaluated expression is returned

original = alpha.fn(Portfolio,1:100) # portfolio data set
set.seed(1)
alpha.fn(Portfolio,sample(100,100,replace=T)) # randomly select
observations

# generate 1,000 bootstrap samples
alpha_b = rep(0,1000)
for (i in 1:1000){
  set.seed(i)
  alpha_b[i]=alpha.fn(Portfolio,sample(100,100,replace=T))}
mean(alpha_b)
sd(alpha_b)

# generate 1,000 boostrap samples automatically using 'boot' function
boot(Portfolio,alpha.fn,R=1000)
```

(f) **R Example 2:** Estimating the accuracy of a linear regression model

```
boot.fn=function(data,index){
  return(coef(lm(mpg~horsepower,data=data,subset=index)))}
# "coef" Extract Model Coefficients
boot.fn(Auto,1:392) # original

set.seed(1)
boot.fn(Auto,sample(392,392,replace=T))
boot.fn(Auto,sample(392,392,replace=T))

boot(Auto, boot.fn, 1000) # 1000 bootstrap samplings
summary(lm(mpg~horsepower,data=Auto))$coef
```

Instructor: Junho Song
junhosong@snu.ac.kr

```
# bootstrap for second-order polynomial regression
boot.fn=function(data,index){
 return(coef(lm(mpg~horsepower+I(horsepower^2),data=data,subset=index)))}
set.seed(1)
boot(Auto, boot.fn, 1000)
```

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 14

## Linear Model Selection and Regularization (ISL Chapter 6)

The standard linear model: $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_P + \epsilon$

Question: How to improve the simple linear model in terms of ***prediction accuracy*** and ***model interpretability***?

***Prediction accuracy***: If the number of data is not much larger than that of predictors, there can be a lot of variability in the least squares fit

***Model Interpretability***: Including irrelevant variables may lead to unnecessary complexity in the resulting model

Subset selection, shrinkage, and dimension reduction will be discussed as techniques to enhance the linear regression model

1. **Best Subset Selection**

   (a) A least square regression is separately conducted for each of *all* possible combinations of the $p$ predictors to obtain "Best" Subset Selection

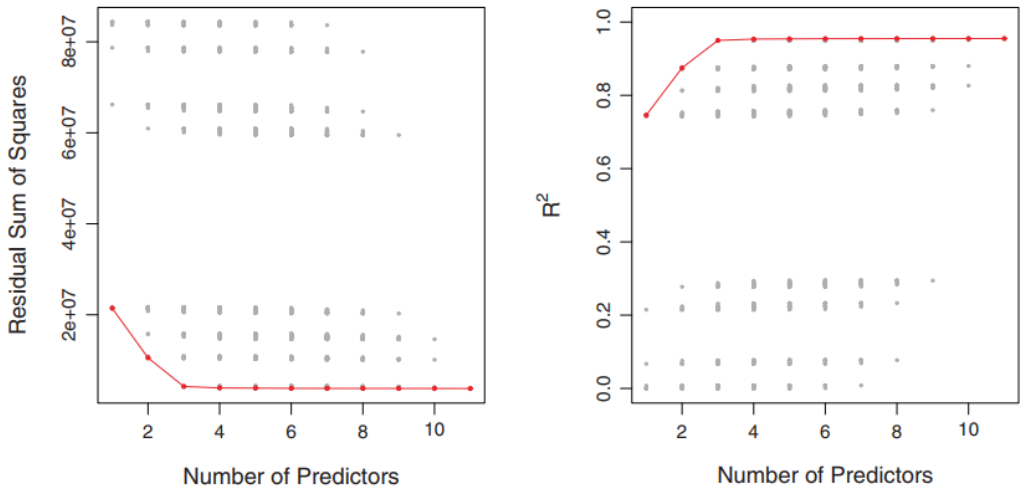
   (b) When $k$ (among $p$) predictors are used, all $\binom{p}{k}$ models are considered $(k = 1,2,\dots,p)$
   ➔ Eventually the best model is selected among $2^p$ models (See *Algorithm 6.1*)

**Algorithm 6.1** *Best subset selection*

1. Let $M_0$ denote the *null* model, which contains no predictors. This model simply predicts the response by the sample mean, i.e. $\hat{y}(x_i) = \bar{y}$.
2. For $k = 1,2,\dots,p$:
   A. Fit all $\binom{p}{k}$ models that contain exactly $k$ predictors.
   B. Pick the best among these $\binom{p}{k}$ models, and call it $M_k$. Here *best* is defined as having the smallest $RSS$, or equivalently largest $R^2$.
3. Select a single best model from among $M_0, \dots, M_p$ in terms of cross-validated prediction error, e.g., $C_p(AIC), BIC,$ Adjusted $R^2$.

   (c) While best subset selection is a simple and conceptually appealing approach, it suffers from large computational costs ➔ might want to use more efficient alternatives (as discussed below)

Instructor: Junho Song
junhosong@snu.ac.kr




Training errors by the models using subsets of predictor for the Credit data set

## 2. Stepwise Selection

(a) When $p$ is large, best subset selection may suffer from not only computational cost but also statistical problems: The larger the search space, the higher the chance of finding models that look good only on training data, i.e. overfitting and high variance of coefficient estimates

➔ One can use stepwise methods as alternatives (which explore a far more restricted set of models)

(b) **Forward stepwise selection** begins with a model containing no predictors (i.e. null model), then adds predictors to the models, one-at-a-time, until all of the predictors are in the model

**Algorithm 6.2** *Forward stepwise selection*

1. Let $M_0$ denote the *null* model, which contains no predictors

2. For $k = 0, \ldots, p-1$:
   A. Consider all $p-k$ models that augment the predictors in $M_k$ with one additional predictor
   B. Choose the best among these $(p-k)$ models, and call it $M_{k+1}$

   Example

   - $k = 0$: compare $p$ models each of which has a single predictor
   - Suppose the model using $X_3$ gives lowest RSS and highest $R^2$
   - $k = 1$: compare $(p-1)$ models each of which has $(X_3, X_1), (X_3, X_2), (X_3, X_4), \ldots$

3. Select a single best model from among $M_0, \ldots, M_P$ in terms of cross-validated prediction error, e.g., $C_p(AIC)$ , $BIC$, Adjusted $R^2$.

Instructor: Junho Song
junhosong@snu.ac.kr

(c) Forward stepwise selection involves fitting one null model, along with $p-k$ models in the $k$th iteration, for $k=0,\ldots,p-1$

The total number of fitted models: $1+\sum_{k=0}^{p-1}(p-k)$

(d) Forward stepwise selection's computational advantage over best subset selection is clear, but it is not guaranteed to find the best possible model out of all $2^p$ models

| # of Variables | Best subset | Forward stepwise |
|---|---|---|
| 1 | rating | rating |
| 2 | rating, income | rating, income |
| 3 | rating, income, student | rating, income, student |
| 4 | cards, income, student, limit | rating, income, student, limit |

Example: best models by Best Subset and Forward Stepwise methods for the Credit data set

(e) Unlike forward stepwise selection, **backward stepwise selection** begins with the full least squares model containing all predictors, and then removes the least useful predictors (in terms of RSS or $R^2$) one-at-a-time

Using backward stepwise selection, it is not guaranteed to find the best possible model

**Algorithm 6.3** *Backward stepwise selection*

1. Let $M_p$ denote the *full* model, which contains all predictors
2. For $k=p,p-1,\ldots,1$:
   A. Consider all $k$ models that contain all but one of the predictors in $M_k$, for a total of $(k-1)$ predictors
   B. Choose the best among these $k$ models, and call it $M_{k-1}$
3. Select a single best model from among $M_0,\ldots,M_P$ in terms of cross-validated prediction error, e.g., $C_p(AIC), BIC,$ Adjusted $R^2$.

(f) Backward stepwise selection requires that the number of samples, $n$ (information or data) is larger than the number of predictors $p$ (so that the full model can be fitted)

By contrast, forward stepwise can be used even when $n<p$, and so it is the only viable subset method when $p$ is large

(g) The best subset, forward stepwise, and backward stepwise selection approaches generally give similar but no identical models

➔ Hybrid Approaches: after adding a new variable, any variables that no longer provide an improvement are removed

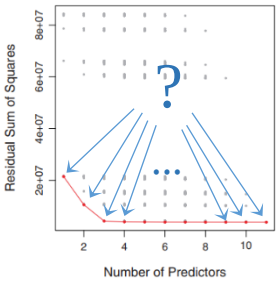
Hybrid versions attempts to mimic best subset selection while retaining computational advantages

Instructor: Junho Song
junhosong@snu.ac.kr

```
library(ISLR) # ISLR library
fix(Hitters) # Load 'Hitters' data
Hitters=na.omit(Hitters)
# remove all of the rows that have missing values in any variable

library(leaps) # leaps library for the following regsubsets() function
regfit.full=regsubsets(Salary~.,data=Hitters,nvmax=19)
# perform best subset selection up to nvmax variables
reg.summary=summary(regfit.full)
# summary() returns following things
# "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"
reg.summary$rsq # R^2 values for each subset size
coef(regfit.full,6) # show the coefficient estimates with subset size 6

# Forward stepwise selection
regfit.fwd=regsubsets(Salary~.,data=Hitters,nvmax=19,method="forward")
coef(regfit.fwd,6)

# Backward stepwise selection
regfit.bwd=regsubsets(Salary~.,data=Hitters,nvmax=19,method="backward")
coef(regfit.bwd,6)
```

```
> coef(regfit.full,6)
 (Intercept)        AtBat         Hits        Walks       CRBI    DivisionW      PutOuts
  91.5117981   -1.8685892    7.6043976    3.6976468  0.6430169 -122.9515338    0.2643076

> coef(regfit.fwd,6)
 (Intercept)        AtBat         Hits        Walks       CRBI    DivisionW      PutOuts
  91.5117981   -1.8685892    7.6043976    3.6976468  0.6430169 -122.9515338    0.2643076

> coef(regfit.bwd,6)
 (Intercept)        AtBat         Hits        Walks       CRuns    DivisionW      PutOuts
  78.2664070   -1.8158931    7.3597644    3.5123248  0.6187876 -113.7958600    0.2995788
```

## 3. Choosing the Optimal Model




(a) RSS and $R^2$, i.e. *training* error measures, are not suitable for selecting the best model among a collection of models with different numbers of predictors

(b) $C_p$, AIC (*Akaike Information Criterion*), BIC (*Bayesian Information Criterion*), and Adjusted $R^2$ – indirect estimates of test error – are often used to choose a best model

These measures give penalty to models with a larger number of predictors, $d$

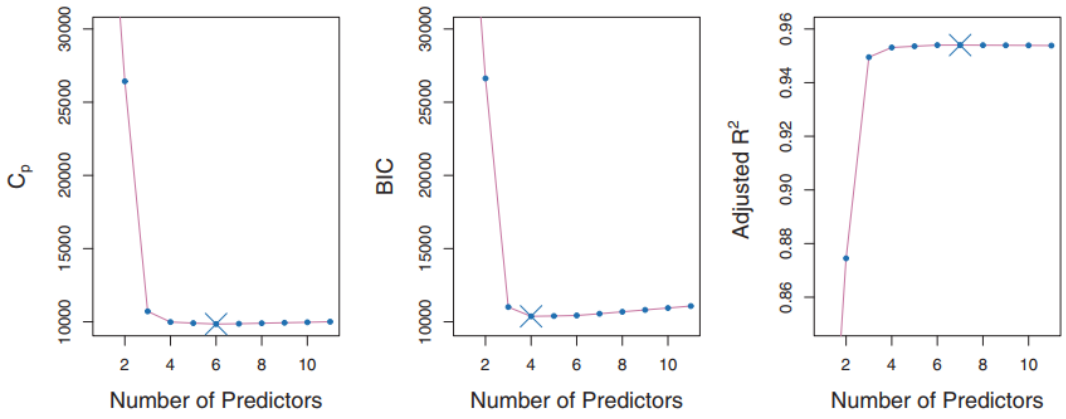
$$C_p = \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2)$$

$$\text{AIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2d\hat{\sigma}^2)$$

$$\text{BIC} = \frac{1}{n}(\text{RSS} + \log(n)\, d\hat{\sigma}^2)$$

$$\text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$$

Instructor: Junho Song
junhosong@snu.ac.kr




$C_p$, **BIC**, and **Adjusted** $R^2$ for the best subset selection on Credit data set

(c) Validation set and cross-validation will be alternatives to the above four approaches because these provide a *direct* estimation of the test error, and make fewer assumptions about true underlying model

## 4. Choosing the Optimal Model Using the Validation Set Approach and Cross-Validation (R code)

(a) In validation set and cross validation approaches, data are divided into training set and test set
(b) For accurate estimates of the test error, *only the training set* should be used to perform all aspects of model-fitting including variable selection

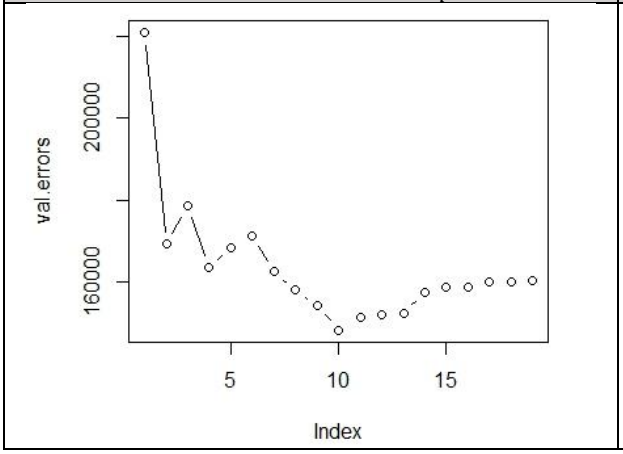
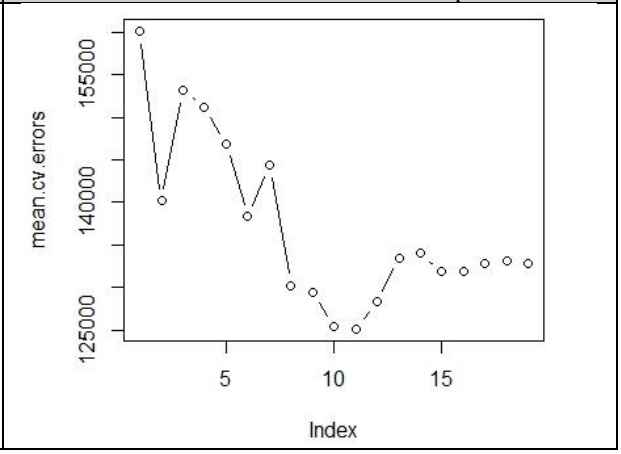
***The Validation Set Approach***

```
library(ISLR)
fix(Hitters)
Hitters=na.omit(Hitters)
set.seed(1)
train = sample(c(TRUE,FALSE),nrow(Hitters),replace=TRUE)
# sample TRUE or FALSE
test = (!train)

library(leaps)
regfit.best = regsubsets(Salary~.,data=Hitters[train,],nvmax=19)
# perform best subset selection using training set

test.mat = model.matrix(Salary~.,data=Hitters[test,])
# building an "X" matrix from data

val.errors = rep(NA,19)
for(i in 1:19){
  coefi = coef(regfit.best,id=i)
  pred = test.mat[,names(coefi)]%*%coefi # matrix multiplication
  val.errors[i] = mean((Hitters$Salary[test]-pred)^2)
}

min_t = which.min(val.errors) # number of variables w/ minimum test error
regfit.best = regsubsets(Salary~.,data=Hitters,nvmax=min_t) #fitting for
   all data w/ optimal number of variables
coef(regfit.best,min_t) # coefficients of the optimal model
```

Instructor: Junho Song
junhosong@snu.ac.kr

***Cross-validation Approach***

```
library(ISLR)
fix(Hitters)
Hitters=na.omit(Hitters)
k=10
set.seed(1)
folds = sample(1:k,nrow(Hitters),replace=TRUE)
cv.errors = matrix(NA,k,19, dimnames=list(NULL, paste(1:19)))
dim(cv.errors)

library(leaps)
#let's define the function for prediction of regsubsets
predict.regsubsets = function(object,newdata,id,...){
  form=as.formula(object$call[[2]]) # extract a formula from object
  mat=model.matrix(form,newdata)
  coefi=coef(object,id=id)
  xvars=names(coefi)
  mat[,xvars]%*%coefi
} # this function provides a prediction using the model in 'object'
   (created by 'regsubsets') at 'newdata' for the level 'id'

# Find Cross-validation errors
for(j in 1:k){
  best.fit = regsubsets(Salary~.,data=Hitters[folds!=j,],nvmax=19)
  for(i in 1:19){
    pred=predict(best.fit,Hitters[folds==j,],id=i)
    cv.errors[j,i] = mean((Hitters$Salary[folds==j]-pred)^2)
  }
}

mean.cv.errors=apply(cv.errors,2,mean) # Average over the columns
plot(1:19,mean.cv.errors,type="b")
min_t = which.min(mean.cv.errors) # number of subset with minimum test
   error
regfit.best = regsubsets(Salary~.,data=Hitters,nvmax=min_t)
coef(regfit.best,min_t) # coefficients of the optimal model
```

| *Validation Set Approach* ($d_{opt} = 10$) | *Cross-Validation Approach* ($d_{opt} = 11$) |
|---|---|
|  |  |