Lecture 6
Resampling Methods

CHUNG-MING KUAN

Department of Finance & CRETA

National Taiwan University

March 5, 2020

Lecture Outline

Introduction

- Given the variable of interest y and the vector of variables $\boldsymbol{x}$ that may characterize the behavior of y, we have learned how to estimate a linear regression model using the sample: $(y_i, \boldsymbol{x}_i')'$, $i = 1, \ldots, n$. More generally, we may consider fitting (training) a possibly nonlinear or nonparametric model f, to this sample.
- Taking the mean squared error (MSE) $\mathbb{E}[y - f(\boldsymbol{x})]^2$ as the criterion function, f is trained by minimizing its sample counterpart:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}[y_i - f(\boldsymbol{x}_i)]^2;$$

this MSE will be referred to as the training MSE. When f is a linear function of $\boldsymbol{x}_i$, this is just the LS criterion we learned earlier.

- How do we select a trained model $\hat{f}$? It is common to select a model from a collection of models according to some measures based on (adjustment of) the training MSE, e.g., $\bar{R}^2$ in linear regressions.
- Yet, a different idea is to select a model according to some measures based on a previously unseen test sample: $(\eta_i, \boldsymbol{\xi}_i)$, $i = 1, \ldots, m$. That is, we want to examine whether $\hat{f}(\boldsymbol{\xi}_i)$ are close to η_i in the MSE sense. This amounts to evaluating the testing MSE:

$$\text{MSE}_{\text{test}} = \frac{1}{m} \sum_{i=1}^{m} [\eta_i - \hat{f}(\boldsymbol{\xi}_i)]^2.$$

- Clearly, a model with a small training MSE does not necessarily have a small testing MSE, and the selected model would be different when different testing samples are used.

Resampling

- Despite that we have only a given sample, modern statistics introduces the concept of resampling. The idea is to draw "new" samples from the original sample and fit the model to each of the "new" samples. This is in contrast with classical statistics in which a model is typically fit to the sample only once.
- Resampling is computationally demanding, but it enables us to obtain more information about the model of interest and to assess the model performance in very different ways. Note that given current computing power, computational simplicity is no longer a major concern.
- In this lecture we will discuss two important resampling methods: Cross Validation (CV) and Bootstrap.

The Validation Set Approach

- The Validation Set Approach randomly splits the sample into two sub-samples of equal size, a training sample and a validation set as the test sample.
- The training sample is used to train models, and every trained model is applied to predict the outcome of the observations in the validation set. The MSE based on the validation set provides an estimate of the testing MSE. A model is selected if it has the smallest testing MSE among a collection of models.
- The trained model and its performance in the validation set depend on how the sample is partitioned; different partitions may lead to very different results.

Example: Taiwan Traffic Data

The monthly traffic-related data in Taiwan from Feb. 2000 to July 2018 are taken from National Police Agency of the Ministry of Interior. We focus on the following variables:

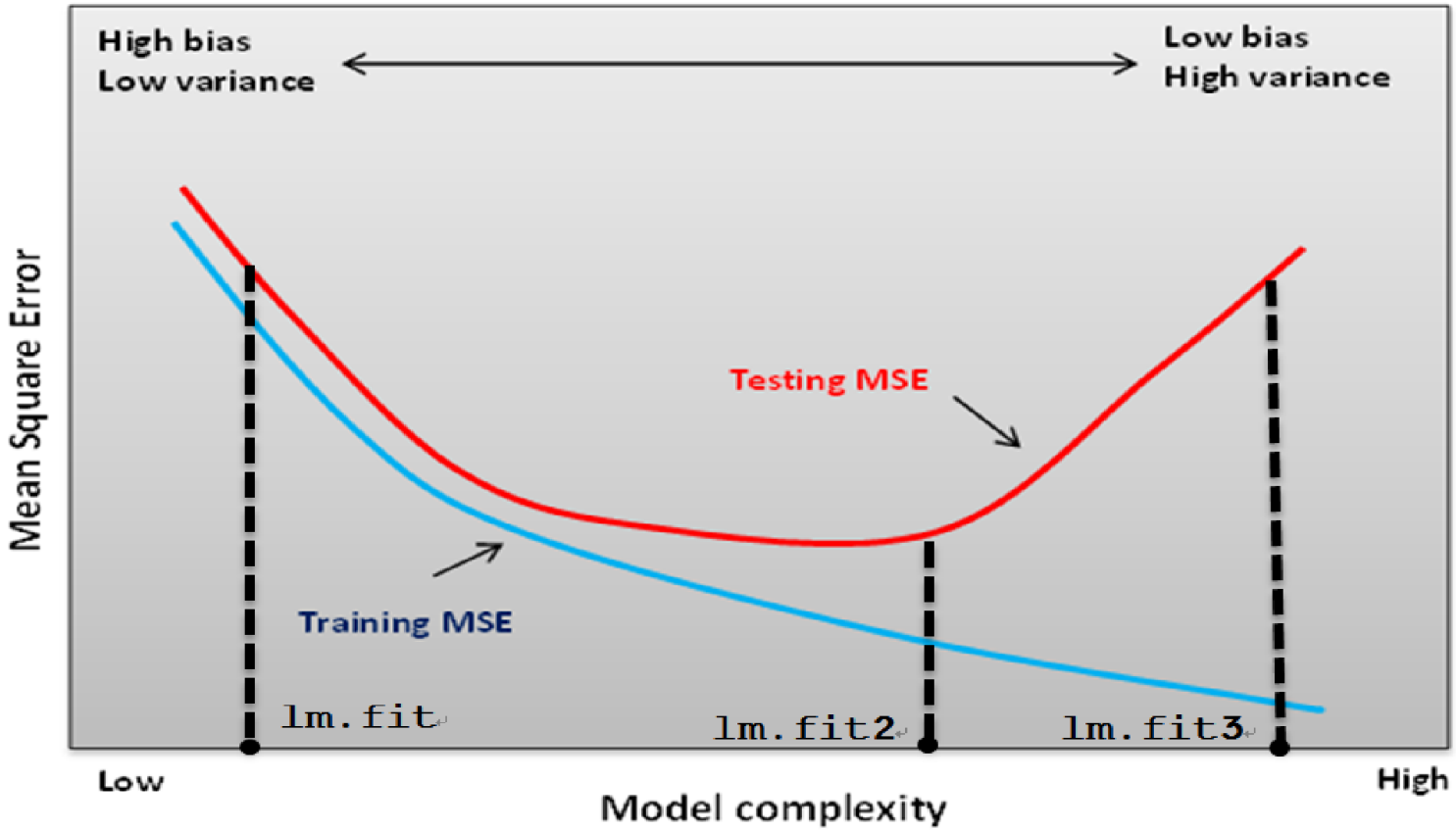
- `Death`: The number of death due to traffic accidents in a month.
- `AutoIncr`: The number of increased automobiles in a month.
- `MotorIncr`: The number of increased motorcycles in a month.
- `Time`: Linear time trend.

As an example, we consider the following models:

$$\texttt{lm.fit:} \quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + u,$$

$$\texttt{lm.fit2:} \quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + \beta_2 \texttt{AutoIncr} + u,$$

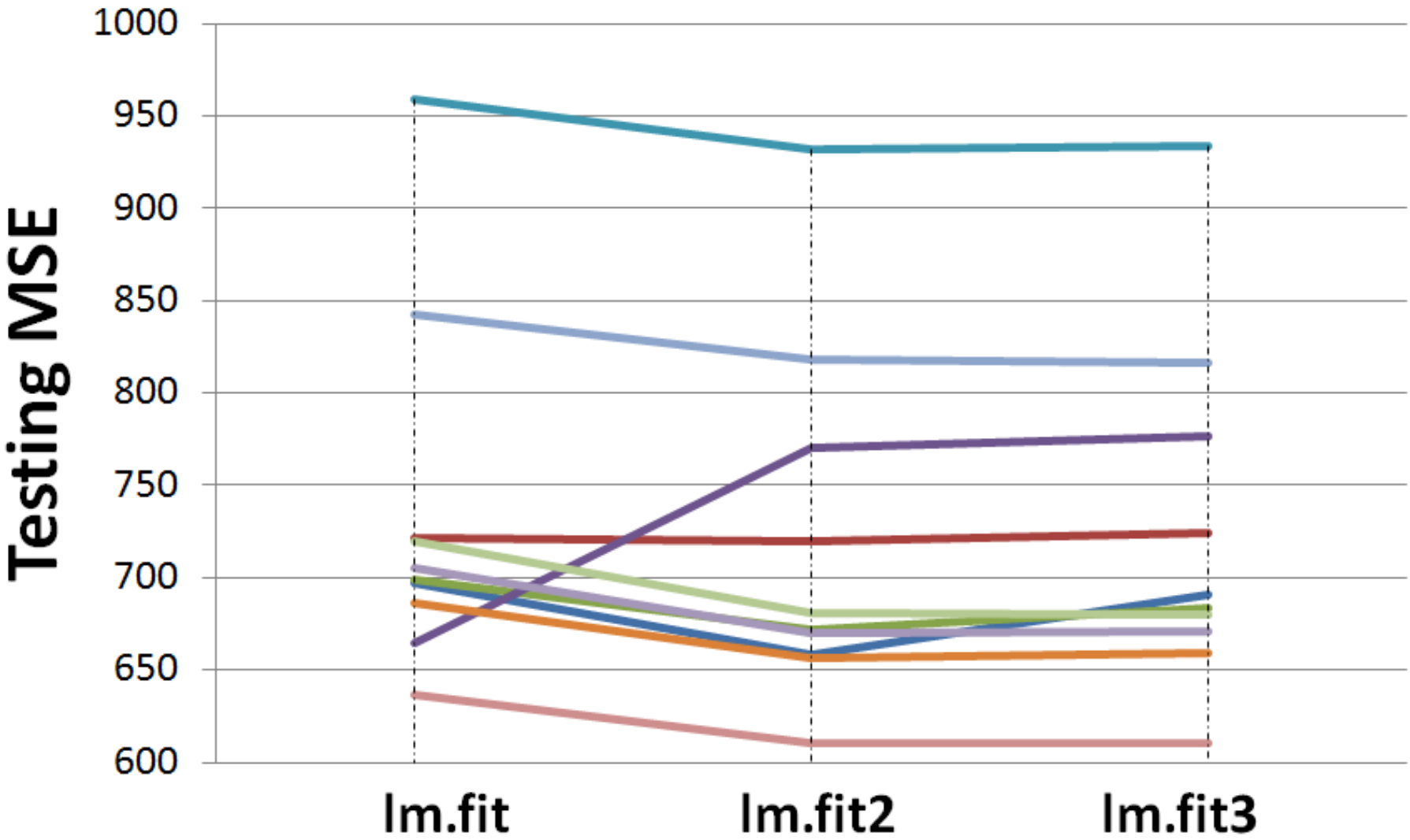
$$\texttt{lm.fit3:} \quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + \beta_2 \texttt{AutoIncr} + \beta_3 \texttt{MotorIncr} + u.$$



The estimated testing MSEs for `lm.fit`, `lm.fit2`, and `lm.fit3` are, respectively, 696.6768, 658.6364, and 690.9322.

Drawbacks of the Validation Set Approach

- The results of the Validation Set Approach depend on random partition of the sample and hence are quite arbitrary. To see this, we conduct 10 random partitions and plot the resulting testing MSEs in the next page. It turns out that the model with the smallest testing MSE may be any of the 3 competing models, depending on which partition is used.
- The Validation Set Approach does not fully utilize the sample information because only half of the sample is used for model training. This problem remains even when the sample is not partitioned equally.


Testing MSE
1000
950
900
850
800
750
700
650
600
lm.fit
lm.fit2
lm.fit3

Leave-One-Out Cross Validation

- Instead of using half of the sample as the validation set, we may take only one observation, say $(y_1, \boldsymbol{x}_1')$, for model validation and the remaining observations, $\{(y_2, \boldsymbol{x}_2'), ..., (y_n, \boldsymbol{x}_n')\}$, for training $\hat{f}_{-1}$. The estimate of the testing MSE is then:

$$\text{MSE}_1 = [y_1 - \hat{f}_{-1}(\boldsymbol{x}_1)]^2.$$

- Take $(y_i, \boldsymbol{x}_i')$ as the validation set and the remaining observations for training $\hat{f}_{-i}$, $i = 2, \ldots, n$. The estimates of the testing MSE are:

$$\text{MSE}_i = [y_i - \hat{f}_{-i}(\boldsymbol{x}_i)]^2, \quad i = 2, \ldots, n.$$

Thus, every observation in complete sample will be "validated" using the model estimated from the remaining $n - 1$ observations.

- The Leave-One-Out Cross Validation (LOOCV) estimate of the testing MSE is:

$$\text{MSE}_{\text{LOOCV}} = \frac{1}{n}\sum_{i=1}^{n}\text{MSE}_i.$$

  A model is selected if it has the smallest $\text{MSE}_{\text{LOOCV}}$.

- Compared with the Validation Set Approach, LOOCV avoids random partition of the sample and utilizes almost the complete sample to train each model. Yet, LOOCV is computationally much demanding when n is large or when the model is difficult to train (e.g., a highly nonlinear model).

- The result below shows that, when the model f is linear and trained by minimizing MSE, the LOOCV estimate can be easily computed by doing LS regression once using the entire sample.

A Special Case of LOOCV: Linear Regression

Let $\hat{y}_i$ denote the fitted values of the linear model trained by OLS. Then,

$$\text{MSE}_{\text{LOOCV}} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2,$$

where $h_i = \boldsymbol{x}_i'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{x}_i$, the i th diagonal term of $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$.

Proof: Given the data $\boldsymbol{y}$ $(n \times 1)$ and $\boldsymbol{X}$ $(n \times p)$, let $\boldsymbol{y}_{-i}$ and $\boldsymbol{X}_{-i}$ denote the sub-matrices of $\boldsymbol{y}$ and $\boldsymbol{X}$, each with the i th row deleted. The OLS estimators of regressing $\boldsymbol{y}$ on $\boldsymbol{X}$ and $\boldsymbol{y}_{-i}$ on $\boldsymbol{X}_{-i}$ are, respectively, $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$, and $\hat{\boldsymbol{\beta}}_{-i} = (\boldsymbol{X}'_{-i}\boldsymbol{X}_{-i})^{-1}\boldsymbol{X}'_{-i}\boldsymbol{y}_{-i}$. We can write $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{X}'_{-i}\boldsymbol{y}_{-i} + \boldsymbol{x}_i y_i$ and

$$\boldsymbol{X}'\boldsymbol{X} = \sum_{j=1}^{n}\boldsymbol{x}_j\boldsymbol{x}_j' = \boldsymbol{X}'_{-i}\boldsymbol{X}_{-i} + \boldsymbol{x}_i\boldsymbol{x}_i'.$$

Writing $\boldsymbol{A} = \boldsymbol{X}'\boldsymbol{X}$, a well-known matrix inversion formula (the third equality) shows that:

$$\begin{aligned}
\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_{-i} &= \boldsymbol{x}_i'(\boldsymbol{X}_{-i}'\boldsymbol{X}_{-i})^{-1}\boldsymbol{X}_{-i}'\boldsymbol{y}_{-i} \\
&= \boldsymbol{x}_i'(\boldsymbol{X}'\boldsymbol{X} - \boldsymbol{x}_i\boldsymbol{x}_i')^{-1}(\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{x}_i y_i) \\
&= \boldsymbol{x}_i'\left(\boldsymbol{A}^{-1} + \frac{\boldsymbol{A}^{-1}\boldsymbol{x}_i\boldsymbol{x}_i'\boldsymbol{A}^{-1}}{1 - \boldsymbol{x}_i'\boldsymbol{A}^{-1}\boldsymbol{x}_i}\right)(\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{x}_i y_i) \\
&= \left(\boldsymbol{x}_i'\boldsymbol{A}^{-1} + \frac{h_i\boldsymbol{x}_i'\boldsymbol{A}^{-1}}{1 - h_i}\right)(\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{x}_i y_i),
\end{aligned}$$

As $\hat{y}_i = \boldsymbol{x}_i'\hat{\boldsymbol{\beta}} = \boldsymbol{x}_i'\boldsymbol{A}^{-1}\boldsymbol{X}'\boldsymbol{y}$, we have

$$\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_{-i} = \left(\frac{\boldsymbol{x}_i'\boldsymbol{A}^{-1}}{1 - h_i}\right)(\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{x}_i y_i) = \frac{\hat{y}_i - h_i y_i}{1 - h_i}.$$

This is the prediction based on x_i and the training OLS estimator $\hat{\boldsymbol{\beta}}_{-1}$.

It follows that the prediction error of y_i is

$$y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_{-i} = \frac{y_i(1-h_i) - \hat{y}_i + h_i y_i}{1-h_i} = \frac{y_i - \hat{y}_i}{1-h_i},$$

and the testing MSE is

$$\text{MSE}_i = [y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_{-i}]^2 = \left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2. \quad i = 1, \ldots, n,$$

Consequently,

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

Remark: As $p = \text{trace}(\boldsymbol{H}) = \sum_{i=1}^n h_i$, h_i may be approximated by p/n. With this approximation, $\text{MSE}_{\text{LOOCV}}$ is proportional to the sample MSE.

k-Fold Cross-Validation

- A CV simpler than LOOCV is k-Fold Cross-Validation: Randomly partition the sample into k groups of equal size and repeatedly take one group as a validation set and remaining data as the training sample. This leads to k estimates of the testing MSEs: $\text{MSE}_1, \ldots, \text{MSE}_k$. The resulting k-fold estimate of the testing MSE is:

$$\text{MSE}_{k\text{-fold}} = \frac{1}{k} \sum_{j=1}^{k} \text{MSE}_j.$$

- k-fold CV is computationally simpler because it requires fitting only k ($k < n$) models. When $k = n$, this is just LOOCV; when $k = 2$, this amounts to implementing the Validation Set Approach twice by flipping the validation and training sets.

Example: Taiwan Traffic Data

Recall that we have three competing models in this example:

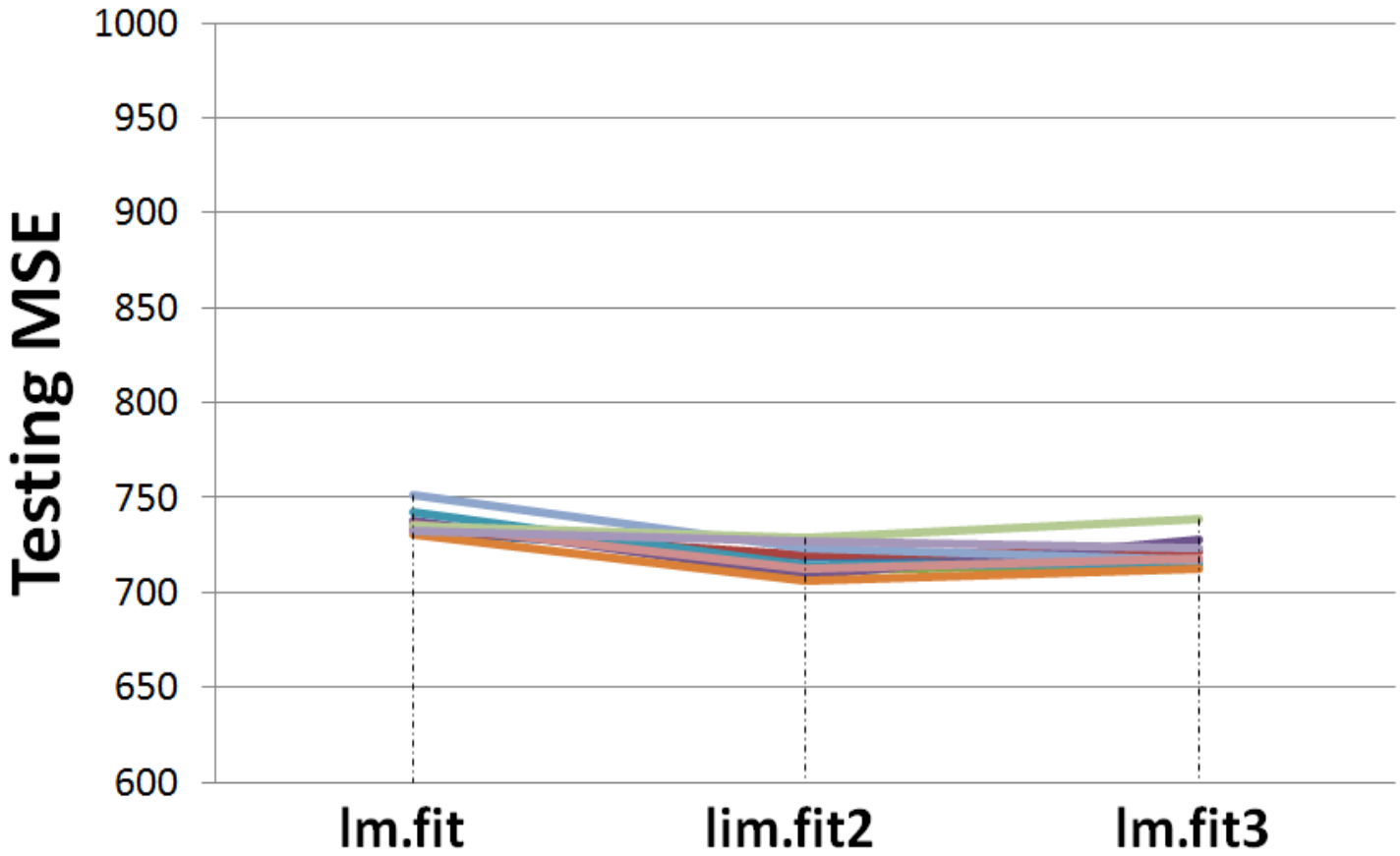
$$\texttt{lm.fit:} \quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + u,$$

$$\texttt{lm.fit2:} \quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + \beta_2 \texttt{AutoIncr} + u,$$

$$\texttt{lm.fit3:} \quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + \beta_2 \texttt{AutoIncr} + \beta_3 \texttt{MotorIncr} + u.$$

- $\text{MSE}_{\text{LOOCV}}$ for `lm.fit`, `lm.fit2`, and `lm.fit3` are, respectively, 738.3419, 718.1488, and 718.6004,
- $\text{MSE}_{\text{10-fold}}$ for `lm.fit`, `lm.fit2`, and `lm.fit3` are, respectively, 733.5457, 719.6191, and 721.1696

Consider 10 different random partitions. The plot below shows that the resulting $\text{MSE}_{10\text{-fold}}$ are quite stable (MSE between 700 and 750), in contrast with those based on the Validation Set Approach.



Comparison between Different Methods

The discussion below is based on p. 183 of JWHT (2013).

- While LOOCV averages n fitted models that are based on highly positively correlated training samples, k-fold CV averages only k fitted models that are based on less correlated training samples (with less overlapping observations). As such, k-fold CV tends to have a smaller variance than does LOOCV.
- While the Validation Set Approach utilizes only half of the sample to train models, k-fold CV makes use of $(k-1)n/k$ observations for model training. Thus, k-fold CV yields a smaller bias in estimating the testing MSE (LOOCV gives approximately unbiased estimate).
- Different choices of k lead to different bias-variance trade-off. In practice, it is common to set $k = 5$ or $k = 10$.

Comparison Based on Simulations

We conduct simulations to compare the performance of different methods. We generate the data x_i and ϵ_i as i.i.d. $\mathcal{N}(0,1)$ and

$$y_i = 1 - 2x_i + 1\, x_i^2 + \epsilon_i,$$

and train the following models with the sample of 10 million observations:

$$\begin{aligned}
\mathcal{M}_1 &: y_i = \beta_0 + \beta_1 x_i + u_i, \\
\mathcal{M}_2 &: y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + u_i, \\
\mathcal{M}_3 &: y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + u_i.
\end{aligned}$$

The "true" testing MSEs are computed from the test sample with 10 million observations: 3, 1, and 1.

Simulation Results

The simulations are based on the training sample of 100 observations with $5,000$ replications.

		LOOCV	10-Fold CV	5-Fold CV	Validation Set
$\widehat{\mathcal{M}}_1$	Bias	0.1418	0.1537	0.1754	0.2819
	Variance	0.8646	0.8815	0.8915	1.6860
$\widehat{\mathcal{M}}_2$	Bias	0.0306	0.0348	0.0427	0.0711
	Variance	0.0217	0.0223	0.0237	0.0539
$\widehat{\mathcal{M}}_3$	Bias	0.0595	0.0689	0.0810	0.1650
	Variance	0.0276	0.0298	0.0327	0.1887

The bias is smaller when k is larger: LOOCV (Validation Set Approach) has the smallest (largest) bias. Yet, we find the variance is also smaller when k is larger: LOOCV (Validation Set Approach) has the smallest (largest) variance; this is different from the discussion of JWHT (2013).

Bootstrap

- In conventional statistical analysis, we rely on a given sample to estimate unknown properties of the population and to test the performance of the estimation results. These tests depend on strong assumptions on data (e.g. normality) or asymptotic approximation.
- The conventional approach may fail when a test statistic is difficult to construct or lack an analytic form, or when its asymptotic distribution is a poor approximation to the exact distribution.
- Efron (1979) introduces the idea of Bootstrap. This approach treats the original sample as the "population", from which many "new" samples can be drawn randomly. These sample information can be used to assess the performance of the original estimation results. As such, bootstrap utilizes the sample information more than once, in contrast with the conventional approach.

Bootstrapping the Standard Errors

Example: Given two assets with returns x and y (with means μ_x and μ_y, variances σ_x^2 and σ_y^2, and covariance σ_{xy}), these two assets can be allocated using the optimal fraction α_o that minimizes $\mathrm{var}(\alpha x + (1-\alpha)y)$:

$$\alpha_o = \frac{\sigma_y^2 - \sigma_{xy}}{\sigma_x^2 + \sigma_y^2 - 2\sigma_{xy}}.$$

Given the sample $\mathcal{S} = \{(y_i, x_i), i = 1, \ldots, n\}$, we can easily calculate the estimators $\hat{\sigma}_x^2$, $\hat{\sigma}_y^2$, $\hat{\sigma}_{xy}$, and hence

$$\hat{\alpha} = \frac{\hat{\sigma}_y^2 - \hat{\sigma}_{xy}}{\hat{\sigma}_x^2 + \hat{\sigma}_y^2 - 2\hat{\sigma}_{xy}}.$$

Yet, computing the standard error of $\hat{\alpha}$ is not straightforward.

Let $\mathcal{S} = \{(y_i, x_i), i = 1, \ldots, n\}$ denote the original sample. Below is the procedure for bootstrapping the standard error.

1. For each $b = 1, 2, \ldots, B$, randomly draw n observations from $\mathcal{S}$ with replacement and obtain the b^{th} bootstrapped sample:

$$\mathcal{S}_b = \{(y^*_{b,1}, x^*_{b,1}), \ldots, (y^*_{b,n}, x^*_{b,n})\},$$

   This is also known as case resampling.
2. For each bootstrapped sample $\mathcal{S}_b$, compute $\hat{\alpha}^*_b$, $b = 1, 2, \ldots, B$.
3. The resulting $\hat{\alpha}^*_1, \ldots, \hat{\alpha}^*_B$ constitute an empirical distribution of $\hat{\alpha}$. We can then compute its sample mean: $\bar{\hat{\alpha}}^* = B^{-1} \sum_{b=1}^{B} \hat{\alpha}^*_b$, and its standard error:

$$\text{SE}^*(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\alpha}^*_b - \bar{\hat{\alpha}}^*\right)^2}.$$

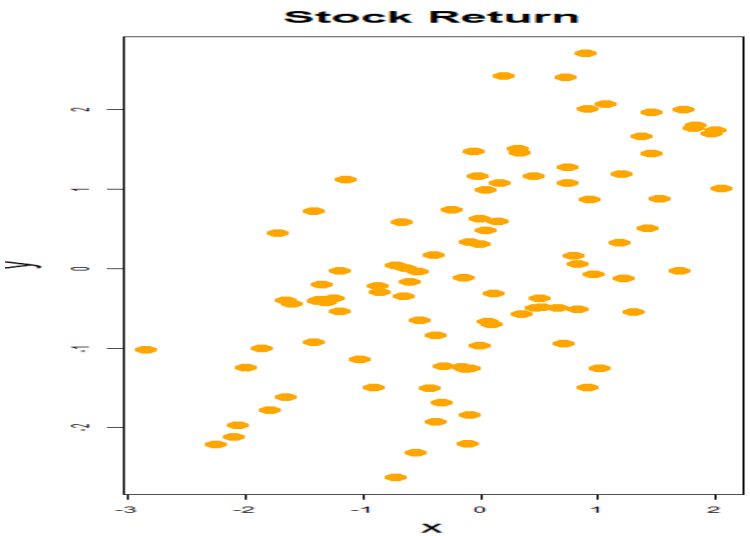
Remarks:

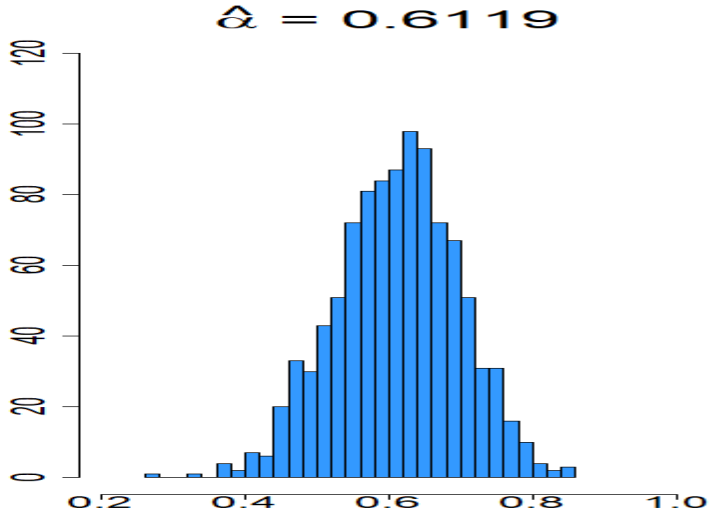
- In bootstrap, random sampling an i.i.d. sample creates another i.i.d. sample. Note that in each sampling, some observations in $\mathcal{S}$ may not be drawn, while some observations in $\mathcal{S}$ may be drawn more than once. Thus, bootstrapped samples are different in general.
- The statistics computed using the bootstrapped samples form an empirical distribution of the statistic from the original sample. Thus, it is typical to choose a large B in bootstrap.
- Bootstrap can be applied to many difficult estimation problems and does not require strong assumptions on data or model. It has been found that bootstrap usually leads to more reliable inferences than those based on asymptotic approximation.

Now suppose the return pairs $(y, x)'$ are generated according to:

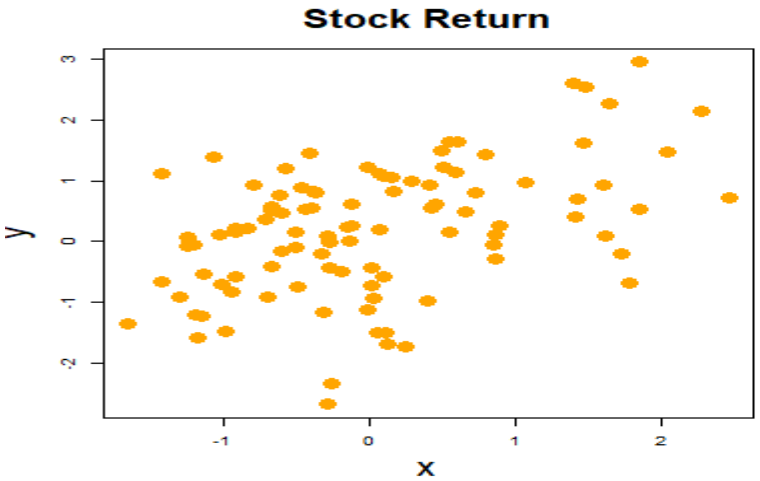
$$\begin{pmatrix} y \\ x \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.25 & 0.5 \\ 0.5 & 1 \end{pmatrix} \right).$$

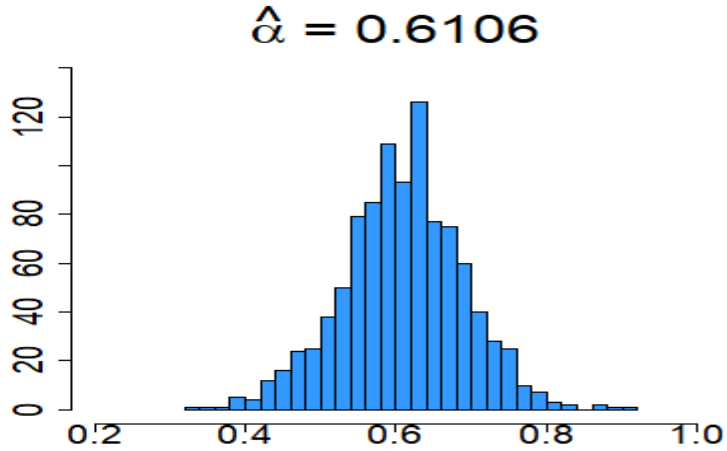
Here, $\alpha_o = 0.6$. Below is the scatter plot of a random sample of 100 observations generated from this distribution (left), for which $\hat{\alpha} = 0.6119$, and the bootstrapped distribution of $\hat{\alpha}$ (right) with $\bar{\hat{\alpha}}^* = 0.6113$ and $\text{SE}^*(\hat{\alpha}) = 0.0853$. This is close to the "true" SE (0.081) which is calculated from $100,000$ random samples.

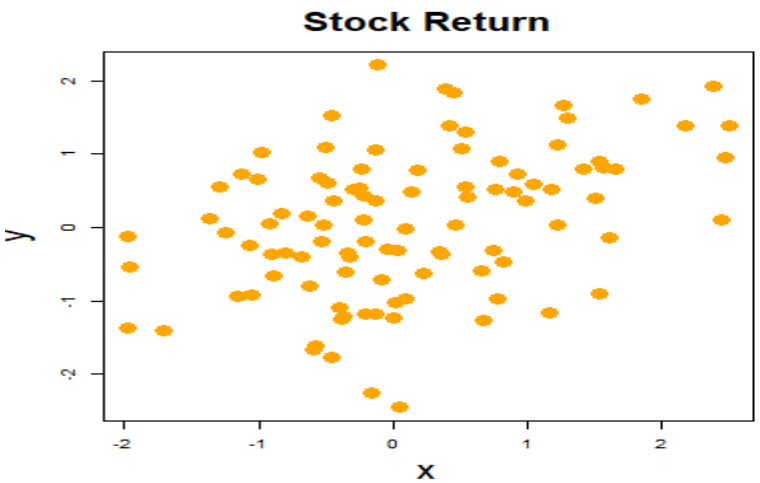


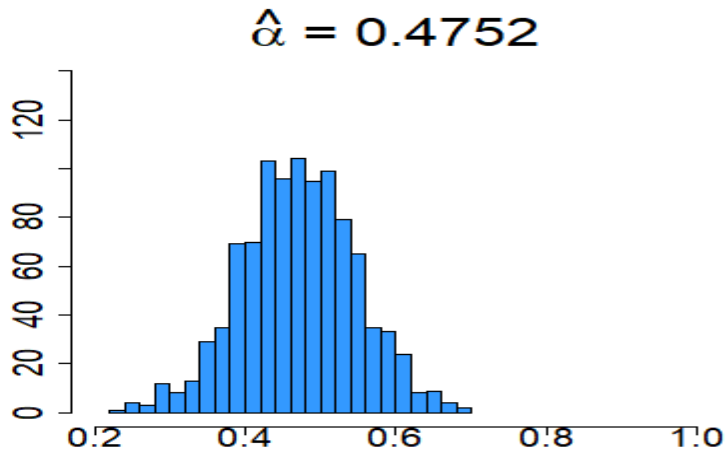


Below are the scatter plots of another 2 random samples for which $\hat{\alpha}$ are 0.6106 and 0.4752. Bootstrap then yields $\bar{\hat{\alpha}}^*$: 0.6089 and 0.4711, and $\text{SE}^*(\hat{\alpha})$: 0.0806 and 0.0762, which are are close to the "true" SE (0.081), even when $\hat{\alpha}$ is far away from 0.6.









Standard Errors in Regressions

Given the sample $\mathcal{S} = \{(y_i, \boldsymbol{x}_i'), i = 1, \ldots, n\}$, regressing y_i on $\boldsymbol{x}_i$ yields

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_k x_{ik} + \hat{u}_i, \quad i = 1, \ldots, n,$$

where $\hat{\boldsymbol{\beta}}$ is the vector of the OLS estimates and $\hat{u}_i$ the residuals. We have learned that the standard error of the coefficient estimate: $\hat{\beta}_j$ is the square root of the $(j+1)$ th diagonal element of the classical estimator: $\hat{\sigma}^2 (\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i')^{-1}$, or the Eicker-White-type estimator:

$$\left(\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i' \right)^{-1} \left(\sum_{i=1}^n \hat{u}_i^2 \boldsymbol{x}_i \boldsymbol{x}_i' \right) \left(\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i' \right)^{-1}.$$

Instead, we may compute standard errors via bootstrap.

Paired Bootstrap

Given the sample $\mathcal{S}$ of n observations, the procedure for bootstrapping the regression standard error is:

1. For each $b = 1, 2, \ldots, B$, randomly draw n observations from $\mathcal{S}$ with replacement and obtain the b^{th} bootstrapped sample: $\mathcal{S}_b = \{(y^*_{b,i}, \boldsymbol{x}^{*\prime}_{b,i}), i = 1, \ldots, n\}$.
2. For each $\mathcal{S}_b$, regress $y^*_{b,i}$ on $\boldsymbol{x}^*_{b,i}$ to obtain $\hat{\boldsymbol{\beta}}^*_b$.
3. The bootstrapped standard error of $\hat{\beta}_j$ is:

$$\text{SE}^*(\hat{\beta}_j) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\beta}^*_{j,b} - \bar{\hat{\beta}}^*_j)^2},$$

where $\bar{\hat{\beta}}^*_j = B^{-1}\sum_{b=1}^{B}\hat{\beta}^*_{j,b}$

Residual Bootstrap

The residual bootstrap suggests bootstrapping the original residuals $\hat{u}_i = y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}$ while keeping the regressors $\boldsymbol{x}_i$ fixed.

1. For each $b = 1, 2, \ldots, B$, randomly draw n observations from the residuals $\hat{u}_i$ with replacement, denoted as $\hat{u}^*_{b,i}$, and compute $y^*_{b,i}$ as

$$y^*_{b,i} = \boldsymbol{x}_i'\hat{\boldsymbol{\beta}} + \hat{u}^*_{b,i}.$$

   The b th bootstrapped sample is $\mathcal{S}_b = \{(y^*_{b,i}, \boldsymbol{x}_i'), i = 1, \ldots, n\}$.
2. For each $\mathcal{S}_b$, regress $y^*_{b,i}$ on $\boldsymbol{x}_i$ to obtain $\hat{\boldsymbol{\beta}}^*_b$.
3. The bootstrapped standard error of $\hat{\beta}_j$ is:

$$\mathsf{SE}^*(\hat{\beta}_j) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\beta}^*_{j,b} - \bar{\hat{\beta}}^*_j\right)^2}.$$

Example

- Generate x_i as i.i.d. $\mathcal{N}(0,1)$, u_i as i.i.d. $\mathcal{N}(0,1)$ or $t(4)$, $i = 1, \ldots, 30$. Then, generate y_i according to:

$$y_i = \beta_0 + \beta_1 x_i + u_i, \quad \beta_0 = 2, \beta_1 = 4.$$

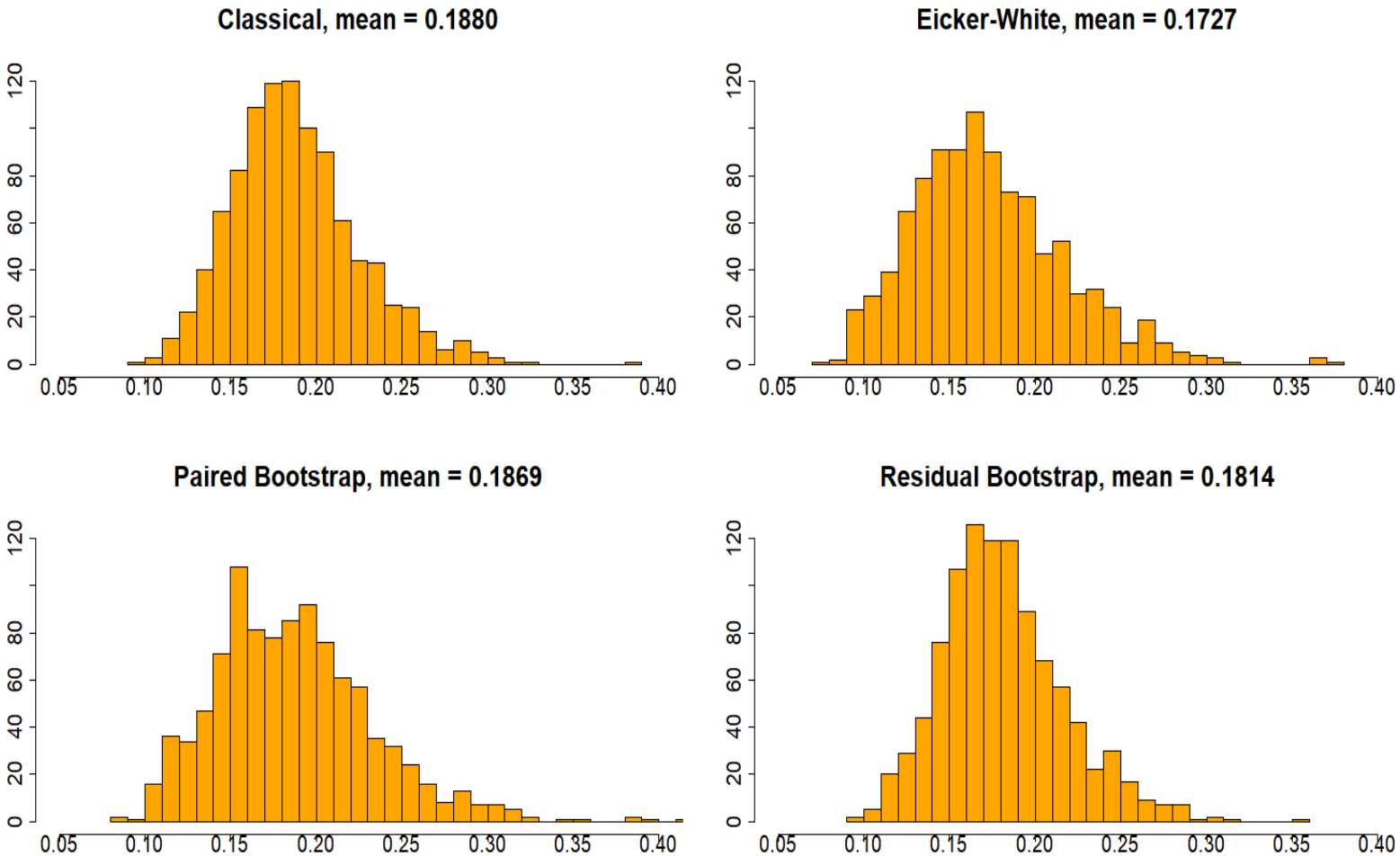
  This gives the sample $\mathcal{S} = \{(y_i, x_i), i = 1, \ldots, 30\}$.

- Regress y_i on $\boldsymbol{x}_i = (1, x_i)'$ to obtain $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)'$. The standard error of $\hat{\beta}_1$ is the square root of the 2nd diagonal element of the classical estimator: $\hat{\sigma}^2 (\sum_{i=1}^{30} \boldsymbol{x}_i \boldsymbol{x}_i')^{-1}$, or the Eicker-White-type estimator:

$$\left(\sum_{i=1}^{30} \boldsymbol{x}_i \boldsymbol{x}_i' \right)^{-1} \left(\sum_{i=1}^{30} \hat{u}_i^2 \boldsymbol{x}_i \boldsymbol{x}_i' \right) \left(\sum_{i=1}^{30} \boldsymbol{x}_i \boldsymbol{x}_i' \right)^{-1}.$$

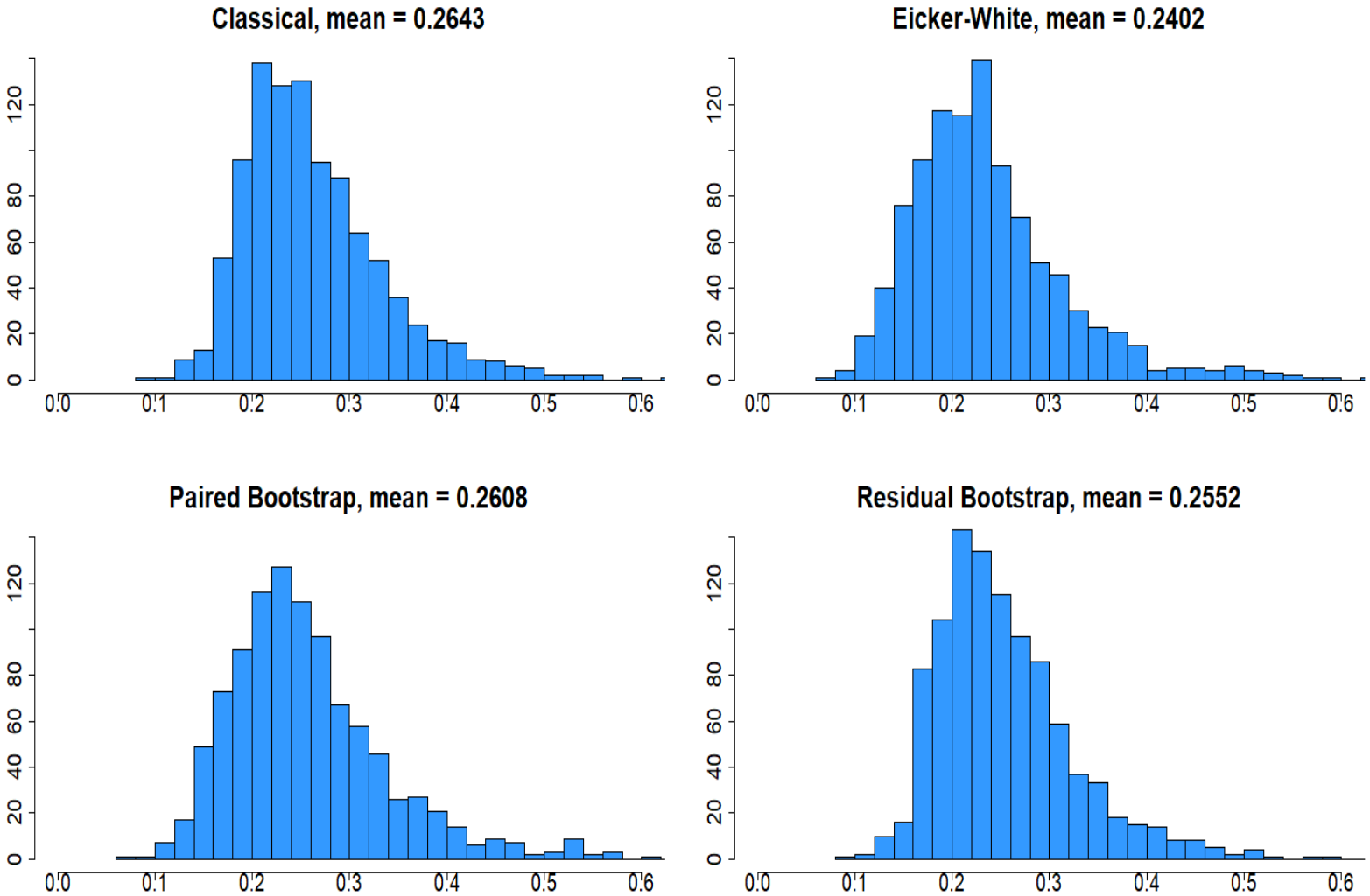
Conventional/Bootstrapped Standard Errors: $u_i \sim \mathcal{N}(0, 1)$

Below are the simulated distributions of the SEs, based on $1,000$ simulated samples of 30 observations. Note that the "true" SE is 0.1922.



Conventional/Bootstrapped Standard Errors: $u_i \sim t(4)$

Note that the "true" SE is 0.2724.



Bootstrapping the Critical Values

If our interest is in hypothesis testing, we may bootstrap the test statistic and its critical values. We take the t statistic, $t_j = (\hat{\beta}_j - c)/\text{se}(\hat{\beta}_j)$ with c the hypothetical value, as an example.

- Using the paired bootstrap or the residual bootstrap to compute the bootstrapped statistics:

$$t^*_{j,b} = (\hat{\beta}^*_{j,b} - \hat{\beta}_j)/\text{se}^*(\hat{\beta}_j), \quad b = 1, 2, \ldots, B.$$

  Note that $t^*_{j,b}$ is centered at $\hat{\beta}_j$, rather than c.
- Order the statistics $t^*_{j,b}$ from smallest to largest. Given the significance level α, the critical value of the one-sided test is the $(1-\alpha)$ th quantile of $t^*_{j,b}$, and the critical values of the two-sided test are the $(\alpha/2)$ th and $(1-\alpha/2)$ th quantiles.

Letting $\tilde{u}^*_{b,i} = y^*_{b,i} - \boldsymbol{x}^{*\prime}_{b,i}\hat{\boldsymbol{\beta}}$, the OLS estimator from the paired bootstrap can be written as:

$$\begin{aligned}\hat{\boldsymbol{\beta}}^*_b &= \left(\sum_{i=1}^n \boldsymbol{x}^*_{b,i}\boldsymbol{x}^{*\prime}_{b,i}\right)^{-1}\left(\sum_{i=1}^n \boldsymbol{x}^*_{b,i}y^*_{b,i}\right) \\ &= \hat{\boldsymbol{\beta}} + \left(\sum_{i=1}^n \boldsymbol{x}^*_{b,i}\boldsymbol{x}^{*\prime}_{b,i}\right)^{-1}\left(\sum_{i=1}^n \boldsymbol{x}^*_{b,i}\tilde{u}^*_{b,i}\right).\end{aligned}$$

The fact that y^*_i and $\boldsymbol{x}^*_i$ are drawn jointly in paired bootstrap causes correlations between $\boldsymbol{x}^*_{b,i}$ and $\tilde{u}^*_{b,i}$, causing a "simultaneity" problem in the regression of $y^*_{b,i}$ on $\boldsymbol{x}^*_{b,i}$. As such, $\hat{\boldsymbol{\beta}}^*_b$ may differ from $\hat{\boldsymbol{\beta}}$ substantially, and their difference does not vanish in the limit. This may affect the bootstrapped critical values because the test statistic is centered at $\hat{\boldsymbol{\beta}}$.

The residual bootstrap avoids the aforementioned problem by fixing $\boldsymbol{x}_i$ and bootstrapping the residuals $\hat{u}_i$. As $\hat{u}_{b,i}^* = y_{b,i}^* - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}$, the OLS estimator based on the residual bootstrap can be written as:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_b^* &= \left(\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i'\right)^{-1}\left(\sum_{i=1}^n \boldsymbol{x}_i y_{b,i}^*\right) \\ &= \hat{\boldsymbol{\beta}} + \left(\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i'\right)^{-1}\left(\sum_{i=1}^n \boldsymbol{x}_i \hat{u}_{b,i}^*\right),\end{aligned}$$

and $\boldsymbol{x}_i$ and $\hat{u}_{b,i}^*$ are no longer determined jointly. Consequently, the second term on the right-hand side above would vanish in the limit.

References and Acknowledgement

References

1. James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An Introduction to Statistical Learning, with Applications in R*, New York: Springer. (JWHT (2013))
2. Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*, Second Edition, New York: Springer.

Some of the figures in this presentation are taken from JWHT (2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani