

Exercise – Understanding the method of paired bootstrap

1. Aim of the exercise

The aim of this exercise is to understand how to correctly estimate standard errors of regression coefficients when the error variance is not constant (i.e., heteroskedasticity is present). Standard inference methods can be misleading in such cases. We use the paired bootstrap method, which resamples observations as (y, \mathbf{X}) pairs, to account for heteroskedasticity and compare its performance to sampling directly from the population.

2. Theory

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where \mathbf{y} is an $n \times 1$ vector of outcome, and \mathbf{X} is an $n \times K$ matrix of regressors. The parameter vector $\boldsymbol{\beta}$ is $K \times 1$ and unknown. Assume the zero conditional mean assumption holds:

$$\mathbb{E}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = 0,$$

but allow for heteroskedastic errors such that

$$\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}' \mid \mathbf{X}] = \boldsymbol{\Omega},$$

where $\boldsymbol{\Omega} = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$. The ordinary least squares (OLS) estimator of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

To account for heteroskedasticity, the variance-covariance matrix of $\hat{\boldsymbol{\beta}}$ can be estimated using the heteroskedasticity-consistent estimator (often called White's estimator):

$$\widehat{\text{Var}}[\hat{\boldsymbol{\beta}}] = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\hat{\boldsymbol{\Omega}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1},$$

where $\hat{\boldsymbol{\Omega}} = \text{diag}(\hat{u}_1^2, \hat{u}_2^2, \dots, \hat{u}_n^2)$, and \hat{u}_i are the OLS residuals.

While the heteroskedasticity-consistent estimator is asymptotically valid, it can suffer from finite-sample biases, particularly in the presence of high-leverage observations. In such cases, the paired bootstrap offers an alternative by resampling entire (\mathbf{x}_i, y_i) observation pairs and approximating the finite-sample distribution of the OLS estimator without the need to specify or estimate the heteroskedastic error-covariance matrix $\boldsymbol{\Omega}$.

Here, a 'pair' refers to the combined observation (y_i, \mathbf{x}_i) , where y_i is the response variable and \mathbf{x}_i is a vector of K covariates for the i th observation. Let (y_i, \mathbf{x}_i) , for $i = 1, \dots, n$, be an i.i.d. sample from the linear model described above. To implement the paired bootstrap, we treat the dataset as a collection of observation pairs $\{(y_1, \mathbf{x}_1), (y_2, \mathbf{x}_2), \dots, (y_n, \mathbf{x}_n)\}$. For each bootstrap replication $b = 1, \dots, B$, we draw n pairs with replacement from this set to form a bootstrap sample $\{(y_1, \mathbf{x}_1)^b, (y_2, \mathbf{x}_2)^b, \dots, (y_n, \mathbf{x}_n)^b\}$. Using this resampled dataset, we compute the OLS estimator

$$\hat{\boldsymbol{\beta}}^b = (\mathbf{X}^{b'}\mathbf{X}^b)^{-1}\mathbf{X}^{b'}\mathbf{y}^b$$

and store the result. After B replications, we obtain the collection $\{\hat{\beta}^b\}_{b=1}^B$. The empirical distribution of these bootstrap estimates can then be used to assess key statistical properties of the estimator, including its standard error, bias, and to construct confidence intervals.

As discussed in the exercise on bootstrap theory, under suitable regularity conditions, (i) as the number of bootstrap samples increase:

$$\hat{G}_n(\tau, F_n) \xrightarrow{p} G_n(\tau, F_n)$$

and (ii) as the size of the initial sample increase:

$$G_n(\tau, F_n) \xrightarrow{p} G_n(\tau, F)$$

meaning that the distribution of a statistic under the empirical distribution F_n converges in probability to its true sampling distribution under the population distribution F . Equivalently, for the OLS estimator:

$$\sqrt{n}(\hat{\beta}^b - \hat{\beta}) \xrightarrow{d^*} \sqrt{n}(\hat{\beta} - \beta)$$

where d^* denotes convergence of distributions under the bootstrap measure. That is, the distribution of the OLS estimator computed from bootstrap resamples $\hat{\beta}^b$ converges to the sampling distribution of the original estimator $\hat{\beta}$. Moreover, the bootstrap estimates can be used to compute the multivariate sample variance-covariance matrix of the OLS estimator:

$$\widehat{\text{Var}}[\hat{\beta}^b] = \frac{1}{B-1} \sum_{b=1}^B (\hat{\beta}^b - \bar{\beta})(\hat{\beta}^b - \bar{\beta})',$$

where

$$\bar{\beta} = \frac{1}{B} \sum_{b=1}^B \hat{\beta}^b.$$

This estimator reflects both the variances of the individual coefficients and their covariances, thereby providing a comprehensive characterization of the estimator's sampling uncertainty. The key is that this estimator does not rely on specific assumptions about the error structure like the standard variance estimator of the OLS estimator which is given by

$$\widehat{\text{Var}}[\hat{\beta}] = s^2(X'X)^{-1}.$$

It also circumvents asymptotic approximations, which can perform poorly in finite samples or in the presence of complex forms of heteroskedasticity.

3. Set values for the parameters of the simulation

Clear all variables from memory, set a seed for reproducible results, and set the total number of simulation or bootstrap runs.

```

14 %% 3. Set values for the parameters of the simulation
15
16 % 3.1. Set seed reproducible results
17 rng(1)
18
19 % 3.2. Clear the memory

```

```

20 clear;
21
22 % 3.3. Set the number of simulations as the number of bootstrap samples
23 N_sim = 5000;

```

4. Generate population data

We consider a simple linear regression model with one predictor, drawn from a uniform distribution. The error term is heteroskedastic, with its variance following an exponential pattern governed by a parameter we set to 1.5. This parameter controls the degree of heteroskedasticity: higher values amplify the variance pattern, allowing us to evaluate how effectively the paired bootstrap deals with non-constant error variance. In our setup, the error variance changes multiplicatively with the value of the predictor. The exponential form ensures that the variance is always positive, reflecting the log-linear variance structures often seen in applied research. Given the assumed true value of the coefficient and the observed predictor values, we generate the outcome variable.

Each pair, comprising an outcome and its corresponding predictor, represents a single observation in the population. Resampling these pairs together preserves their joint distribution and any heteroskedasticity linking them. By treating each pair as the fundamental unit and resampling with replacement, the paired bootstrap retains the original relationship between predictor and response, including variance differences across predictor values, leverage effects, and other forms of dependence.

```

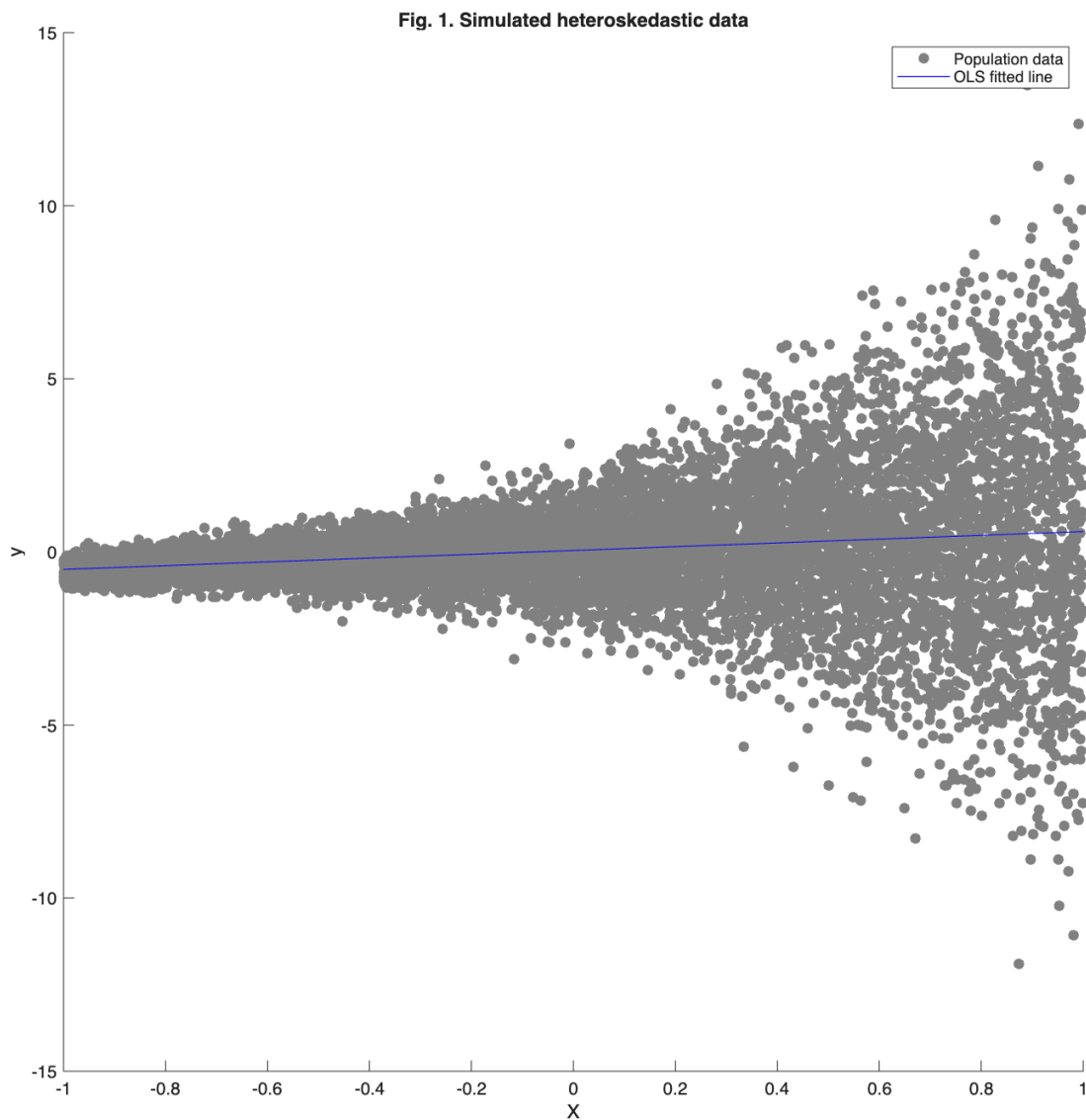
25 %% 4. Generate population data
26
27 % 4.1. Set the population size
28 N_obs_pop = 1000;
29
30 % 4.2. Generate data for the independent variable
31 X_pop = [random('Uniform', -1, 1, [N_obs_pop, 1])];
32
33 % 4.3. Heteroskedasticity parameter
34 Gamma = 1.5;
35
36 % 4.4. Generate error
37 u_pop = random('Normal', 0, exp(X_pop*Gamma), [N_obs_pop, 1]);
38
39 % 4.5. Set true beta of the model
40 B_true = 0.5;
41
42 % 4.6. Generate y
43 y_pop = X_pop*B_true+u_pop;
44
45 % 4.7. Create the (paired) population data
46 data_pop = [y_pop X_pop];

```

5. Plot the scatter diagram and the OLS fitted line

Figure 1 plots the generated data along with the corresponding OLS regression line. It illustrates the presence of heteroskedasticity: the dispersion of the outcome variable increases with the predictor. The OLS line demonstrates that the conditional mean remains linear, even though the variance is non-constant.

```
48 %% 5. Plot the scatter diagram and the OLS fitted line
49 figure
50 hold on
51 scatter(X_pop,y_pop,'filled','MarkerFaceColor',[0.5 0.5 0.5]);
52 h = lsline;
53 set(h,'Color','blue');
54 title('Fig. 1. Simulated heteroskedastic data');
55 xlabel('X');
56 ylabel('y');
57 legend('Population data','OLS fitted line');
58 hold off
```



6. Draw samples from the population

Here we run a Monte Carlo experiment by repeatedly drawing i.i.d. random samples from the population dataset using the `datasample` function of MATLAB to study how a regression coefficient behaves across samples. For each of `N_sim` samples, the code selects `N_obs_sample` paired observations (y_i, X_i) without replacement and stores them in the three-dimensional matrix `data_samples_pop` of size $100 \times 2 \times N_{sim}$. In this structure, rows represent individual observations, columns hold the two variables, and the third dimension indexes the simulation run, so each slice `(:,:,i)` contains one complete sample. The external OLS estimation function is then applied to each sample, and the resulting coefficient estimate from that run is saved in the vector `B_hats_data_samples_pop` for later analysis.

```
60 %% 6. Draw samples from the population
61
62 % 6.1. Set the sample size
63 N_obs_sample = 100;
64
65 % 6.2. Preallocate matrix to save (paired) samples
66 data_samples_pop = NaN(N_obs_sample,2,N_sim);
67
68 % 6.3. Preallocate vector to store coefficient estimates
69 B_hats_data_samples_pop = NaN(N_sim,1);
70
71 % 6.4. Monte Carlo sampling
72 for i = 1:N_sim
73     sample_i = datasample(data_pop,N_obs_sample,'Replace',false);
74     data_samples_pop(:,:,i) = sample_i; % Save samples in 3rd dimension
75     y = sample_i(:,1);
76     X = sample_i(:,2);
77     LSS = exercisefunctionlss(y, X);
78     B_hats_data_samples_pop(i) = LSS.B_hat(1,1);
79 end
```

7. Pick an "initial" sample

We replicate the empirical setting by drawing a single sample from the population without replacement. This initial sample serves as the source for all subsequent bootstrap resamples. Note that the data comprise paired observations of the outcome and predictor, preserving their joint empirical distribution in accordance with paired-bootstrap theory.

```
81 %% 7. Pick an "initial" sample
82
83 % Pick a sample from the samples drawn from the population
84 data_sample = datasample(data_samples_pop(:,:,i),N_obs_sample, ...
85     'Replace',false);
```

8. Draw (bootstrap) samples from the initial sample

Begin by setting up an empty vector to store coefficient estimates from each simulation. For

every run, draw a bootstrap sample with replacement from the initial dataset, then extract the response and predictor variables. Estimate the model using OLS and append the resulting coefficient to the vector. After `N_sim` repetitions, the collection of coefficients forms a bootstrap-based empirical approximation to the sampling distribution of the estimator under heteroskedastic errors.

```

87 %% 8. Draw (bootstrap) samples from the initial sample
88
89 % 8.1. Preallocate vector to store coefficient estimates
90 B_hats_data_samples_boot = NaN(N_sim,1);
91
92 % 8.2. Resample from initial sample and estimate coefficients each time
93 for i = 1:N_sim
94     data_samples_boot = datasample(data_sample,N_obs_sample, ...
95         'Replace',true);
96     y = data_samples_boot(:,1);
97     X = data_samples_boot(:,2);
98     LSS = exercisefunctionlss(y,X);
99     B_hats_data_samples_boot(i) = LSS.B_hat(1,1);
100 end

```

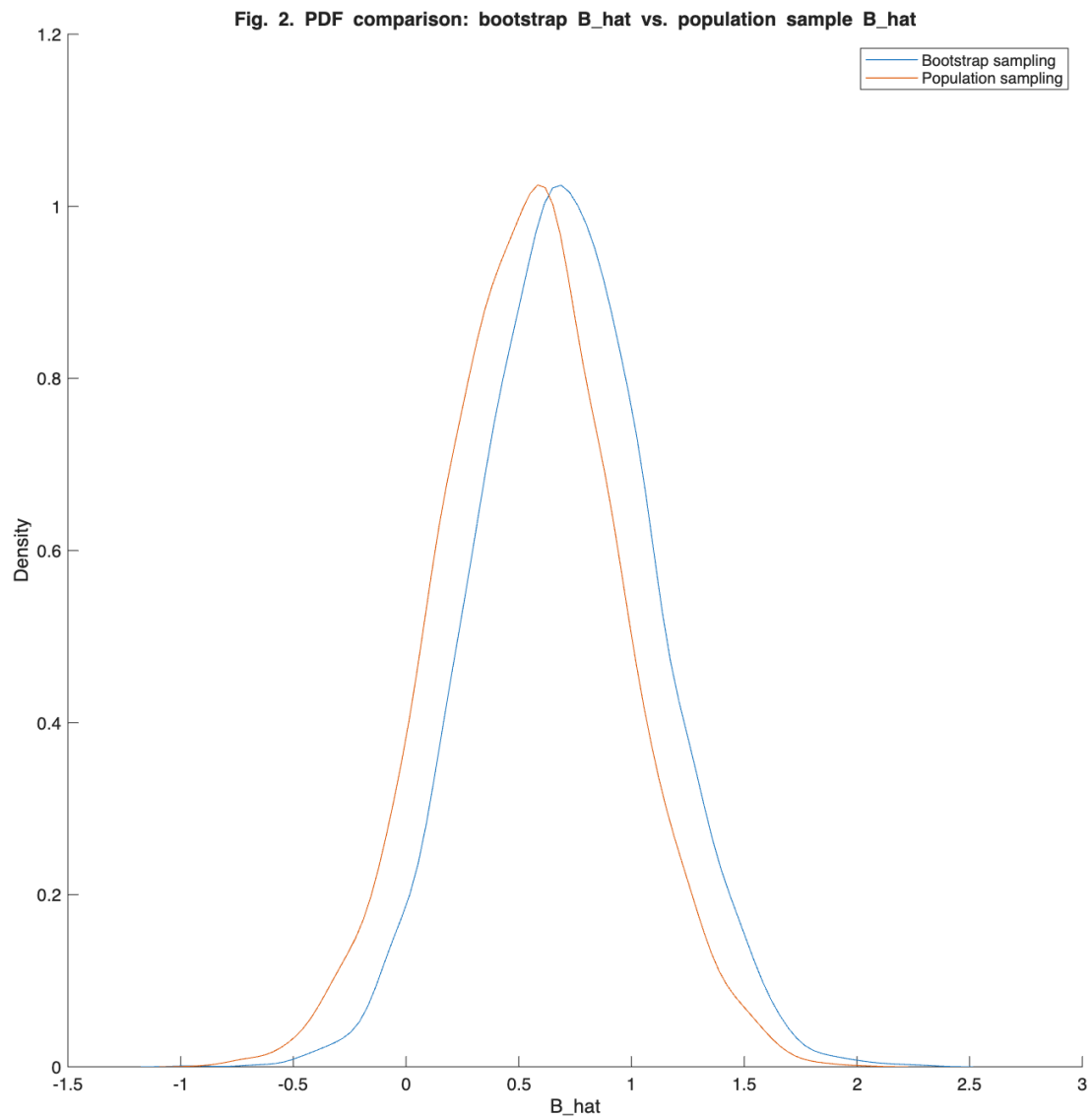
9. Plot the estimated PDFs

In Figure 2, we compare the sampling distribution of slope estimates obtained from paired-bootstrap resamples of the original dataset with the sampling distribution from repeated draws of the entire population. Each distribution is smoothed into a PDF using `ksdensity`. The two PDFs have nearly identical shape and spread, showing that the bootstrap closely reproduces the form of the true sampling distribution. A small horizontal offset between them occurs because the bootstrap is generated from the finite original dataset; if that dataset's mean slope differs slightly from the population's mean slope, the bootstrap distribution will reflect the same shift. This demonstrates that the bootstrap carries over any bias present in the original sample while still accurately representing sampling variability, even with heteroskedastic errors.

```

102 %% 9. Plot the estimated PDFs
103 figure
104 hold on
105 ksdensity(B_hats_data_samples_boot,'function','pdf');
106 ksdensity(B_hats_data_samples_pop,'function','pdf');
107 title(['Fig. 2. PDF comparison: bootstrap B\_hat vs. population ' ...
108     ' sample B\_hat']);
109 xlabel('B\_hat');
110 ylabel('Density');
111 legend('Bootstrap sampling','Population sampling');
112 hold off

```



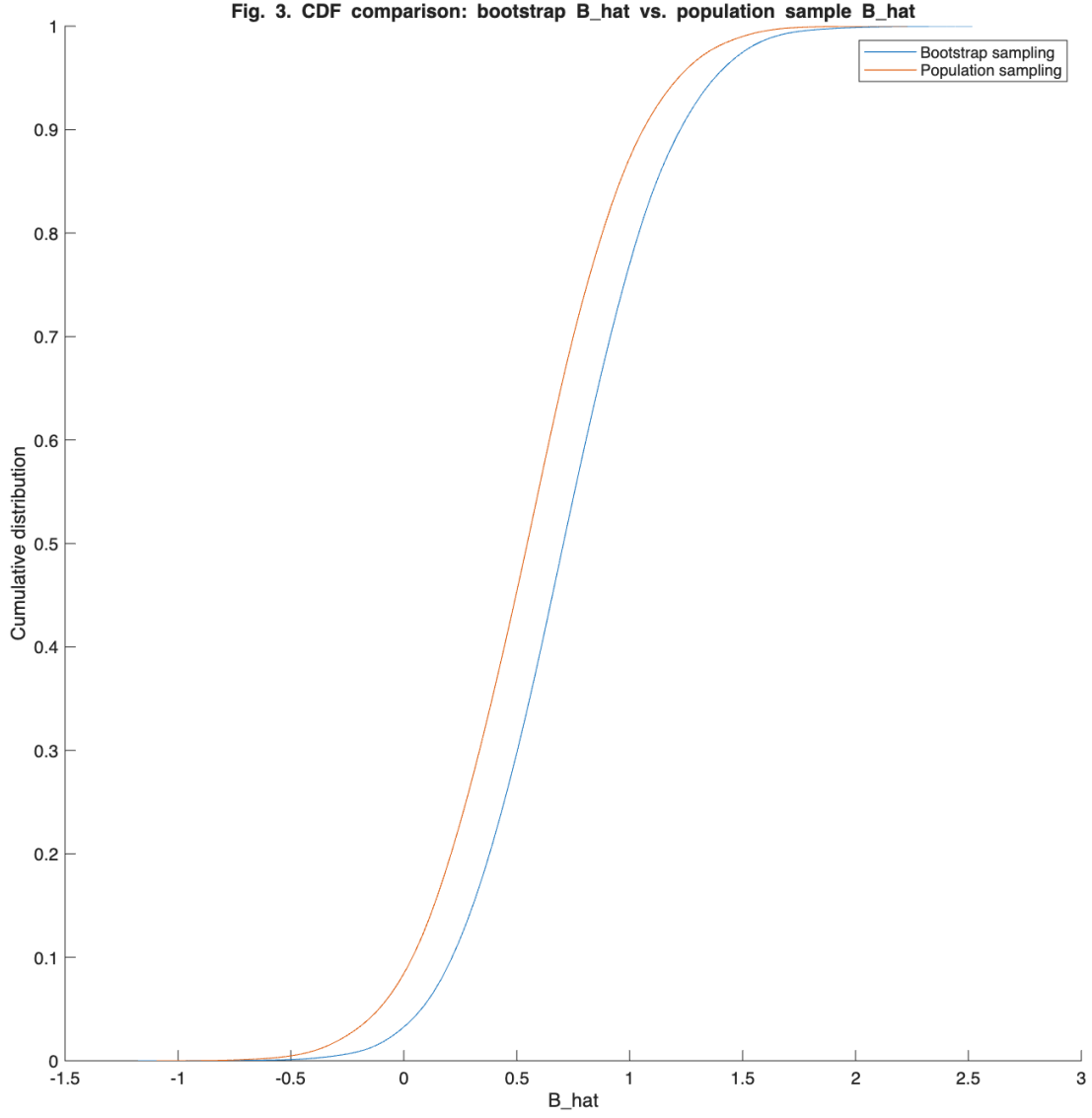
10. Plot the estimated CDFs

Figure 3 plots the CDFs of coefficient estimates from bootstrap and population sampling, plotted via kernel density estimation.

```

114 figure
115 hold on
116 ksdensity(B_hats_data_samples_boot,'function','cdf');
117 ksdensity(B_hats_data_samples_pop,'function','cdf');
118 title(['Fig. 3. CDF comparison: bootstrap  $B_{\hat{}}$  vs. population ' ...
119       'sample  $B_{\hat{}}$ ']);
120 xlabel('B\hat');
121 ylabel('Cumulative distribution');
122 legend('Bootstrap sampling','Population sampling');
123 hold off

```



11. Comparing the standard error estimators across different sample sizes

We evaluate the performance of the paired bootstrap as an estimator of the standard error of a coefficient estimate, comparing it with the heteroskedasticity consistent estimator and with the Monte Carlo estimator, which here represents the true standard error based on the full population. As described in Section 8, from a given sample drawn from the population, we generated multiple bootstrap resamples and computed the corresponding coefficient estimates, thereby producing an empirical sampling distribution. The standard deviation of this distribution is the bootstrap estimate of the standard error.

From a theoretical standpoint, in small samples the heteroskedasticity consistent estimator often exhibits downward bias, while the bootstrap, although still biased, can more closely approximate the true sampling variability. This is precisely what we observe at a sample size of 10: the bootstrap estimate is 0.8245, the Monte Carlo reference is 1.2883, and the heteroskedasticity consistent estimate is 0.7696, with both methods understating the variability. As the sample size increases, both estimators converge toward the true value. The bootstrap nearly matches it at a sample size of 100, with 0.3879 compared with the Monte Carlo value

of 0.3944, while the heteroskedasticity consistent estimator remains slightly lower at 0.3507. This pattern aligns with asymptotic theory, in which larger sample sizes reduce bias in both methods, and the bootstrap's finite sample performance is typically superior.

```
126 %% 11. Comparing the SE estimators across different sample sizes
127
128 % 11.1. True estimate based on Monte Carlo simulation
129 SE_true = std(B_hats_data_samples_pop);
130
131 % 11.2. Heteroskedasticity-consistent estimate
132 SE_HC = LSS.B_hat_SEE_robust;
133
134 % 11.3. Bootstrap estimate
135 SE_boot = std(B_hats_data_samples_boot);
```

12. Final notes

This file is prepared and copyrighted by Axel Zoons, Quinten Solomons, and Tunga Kantarci. This file and the accompanying MATLAB file are available on GitHub and can be accessed via this [link](#).