Supplementary Materials for Fast Estimation and Choice of Confidence Interval Methods for Step Regression

Shuangcheng Hua, Youyi Fong, Jarrod Kath

A Data generating models used in the simulation studies

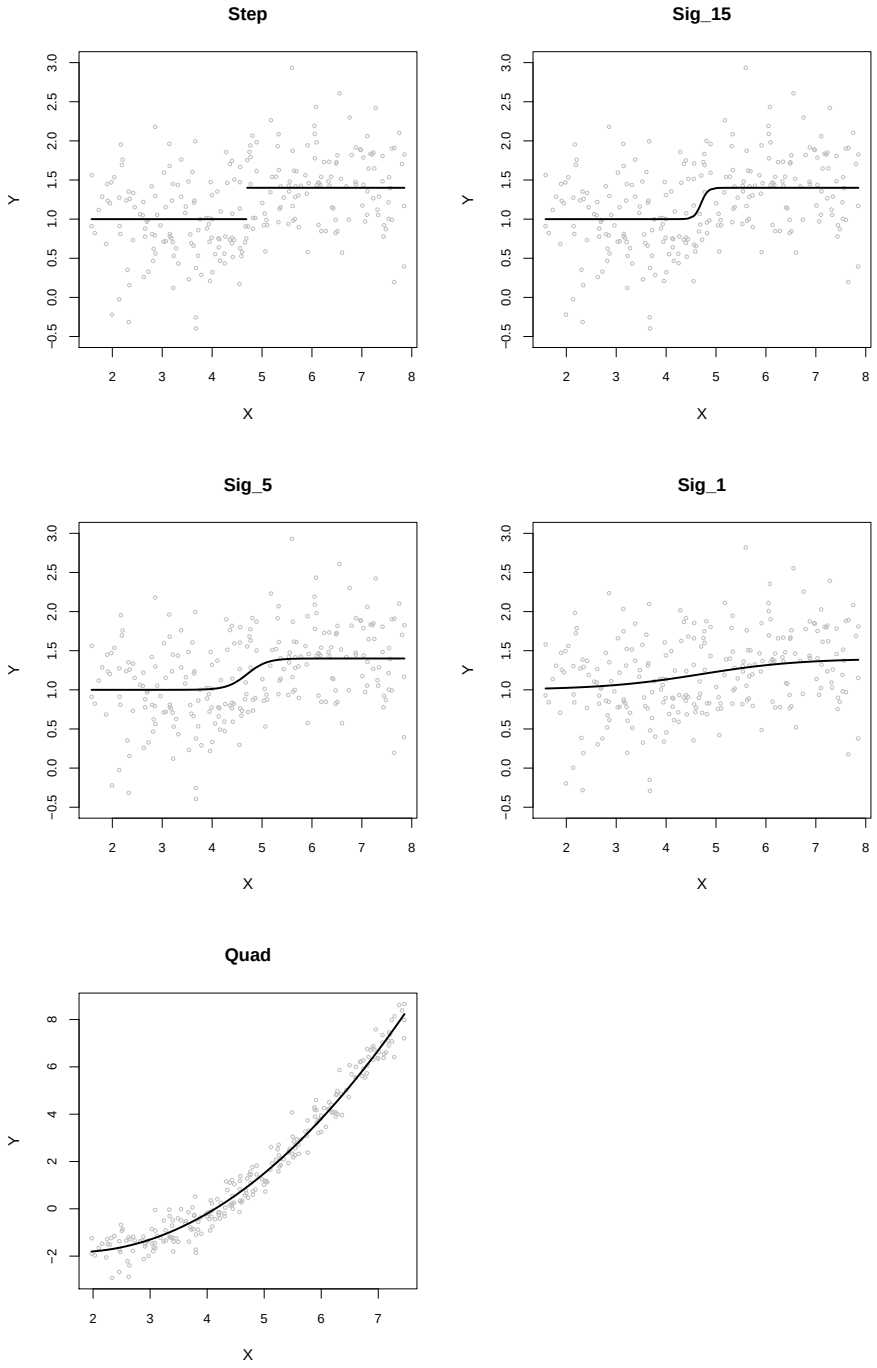

Figure A.1: Datasets of sample size 250 simulated from the step model, sigmoid models and quadratic model. Solid lines: mean functions without the predictor Z and error term. Gray points: data simulated with the predictor Z and the error term.

Step model The step linear regression data generating model is:

$$Y = \alpha + \alpha_z Z + \beta I(X > e) + \epsilon,$$

where $X \sim Unif(1.5, 7.9)$, $Z \sim N(0, \sigma = 1)$, $\epsilon \sim N(0, \sigma = 0.3)$ and is independent of the predictors. $\boldsymbol{\theta}_0 = \{e = 4.7, \alpha = 1, \alpha_z = log(1.4), \beta = -log(.67)\}$.

Sigmoid models The sigmoid models are of the form

$$Y = \alpha + \alpha_z Z + \beta \frac{\gamma e^{(X-4.7)}}{1 + \gamma e^{(X-4.7)}} + \epsilon,$$

where all the covariates and coefficients follow the same setting as the step linear regression model above. Here, γ is the steepness parameter, which is 1, 5 or 15. These S-shape functions approximate and smooth the jump of the step linear regression model.

Sig_1	β		α_z		α	
seed	n=10^5	n=10^6	n=10^5	n=10^6	n=10^5	n=10^6
1	0.236	0.238	0.336	0.336	1.087	1.078
2	0.240	0.237	0.337	0.337	1.078	1.082
3	0.242	0.238	0.337	0.336	1.072	1.082
4	0.238	0.237	0.336	0.336	1.079	1.083
5	0.241	0.237	0.336	0.337	1.079	1.084
6	0.237	0.237	0.336	0.336	1.082	1.080
7	0.238	0.237	0.338	0.336	1.077	1.080
8	0.240	0.238	0.336	0.336	1.076	1.080
9	0.236	0.237	0.335	0.336	1.084	1.079
10	0.239	0.237	0.338	0.336	1.083	1.081
Average	0.239	0.237	0.337	0.336	1.080	1.081
$\hat{\boldsymbol{\theta}}_0$		0.237		0.336		1.081

Table A.1: Approximating $\boldsymbol{\theta}_0$ using the estimated step regression model parameters when the sample size is very large. Data generated from sigmoid models with $\gamma = 1$.

Note that we know by symmetry that $e_0 = 4.7$. The results also show that $\hat{e}$ are more variable than the other parameters.

Sig_5	β		α_z		α	
seed	n=10^5	n=10^6	n=10^5	n=10^6	n=10^5	n=10^6
1	0.363	0.366	0.336	0.336	1.017	1.017
2	0.367	0.366	0.337	0.337	1.014	1.018
3	0.370	0.366	0.338	0.336	1.011	1.018
4	0.366	0.366	0.336	0.336	1.015	1.018
5	0.369	0.366	0.336	0.337	1.015	1.017
6	0.366	0.366	0.336	0.336	1.017	1.018
7	0.366	0.366	0.338	0.336	1.015	1.016
8	0.368	0.366	0.336	0.336	1.016	1.016
9	0.365	0.366	0.335	0.336	1.016	1.018
10	0.367	0.366	0.338	0.336	1.017	1.018
Average	0.367	0.366	0.340	0.336	1.015	1.017
$\hat{\boldsymbol{\theta}}_0$		0.366		0.336		1.017

Table A.2: Estimated coefficients of step regression models when data generating models are sigmoid models with steepness equals to 5 of size 10^5 and 10^6. By symmetry, we know $e_0 = 4.7$.

Sig_15	β		α_z		α	
seed	n=10^5	n=10^6	n=10^5	n=10^6	n=10^5	n=10^6
1	0.386	0.389	0.336	0.336	1.005	1.006
2	0.390	0.389	0.337	0.337	1.005	1.005
3	0.393	0.389	0.338	0.336	1.002	1.006
4	0.389	0.389	0.336	0.336	1.004	1.006
5	0.391	0.389	0.336	0.337	1.005	1.006
6	0.389	0.389	0.335	0.336	1.008	1.006
7	0.389	0.389	0.337	0.336	1.006	1.005
8	0.391	0.389	0.336	0.336	1.005	1.005
9	0.388	0.389	0.335	0.336	1.005	1.005
10	0.390	0.389	0.338	0.336	1.006	1.006
Average	0.390	0.389	0.336	0.336	1.005	1.006
$\hat{\boldsymbol{\theta}}_0$		0.389		0.336		1.006

Table A.3: Estimated coefficients of step regression models when data generating models are sigmoid models with steepness equals to 15 of size 10^5 and 10^6. By symmetry, we know $e_0 = 4.7$.

Quadratic model The quadratic model is:

$$Y = \alpha + \alpha_z Z + \gamma_1 X + \gamma_2 X^2 + \epsilon,$$

where $X \sim Unif(1.9, 7.5)$, $Z \sim N(0, \sigma = 1)$, $\epsilon \sim N(0, \sigma = 0.3)$ and is independent of other predictors, α is set to be -1, α_z is set to be $log(1.4)$, γ_1 is set to be -1 and γ_2 is set to be 0.3.

Qua	e		β		α_z		α	
seed	n=10^5	n=10^6	n=10^5	n=10^6	n=10^5	n=10^6	n=10^5	n=10^6
1	5.437	5.437	5.516	5.506	0.336	0.337	-0.322	-0.318
2	5.438	5.450	5.522	5.518	0.331	0.335	-0.321	-0.309
3	5.435	5.436	5.509	5.510	0.337	0.337	-0.322	-0.320
4	5.438	5.441	5.507	5.508	0.331	0.337	-0.320	-0.317
5	5.418	5.448	5.533	5.522	0.338	0.337	-0.345	-0.314
6	5.461	5.452	5.530	5.519	0.340	0.335	-0.300	-0.307
7	5.463	5.431	5.526	5.505	0.336	0.337	-0.300	-0.323
8	5.425	5.441	5.506	5.515	0.339	0.336	-0.327	-0.318
9	5.430	5.441	5.489	5.510	0.337	0.336	-0.321	-0.315
10	5.428	5.436	5.503	5.508	0.339	0.334	-0.333	-0.319
Average	5.437	5.441	5.514	5.512	0.336	0.336	-0.321	-0.316
$\hat{\boldsymbol{\theta}}_0$		5.441		5.512		0.336		-0.316

Table A.4: Estimated coefficients of step regression models when simulating quadratic models of size 10^5 and 10^6.

B Additional info for Section 2 Fast Grid Search

Matrix derivation of the fast grid search algorithm for step linear regression. Let $\boldsymbol{X}_e \equiv [\mathbf{1}, \boldsymbol{Z}, \boldsymbol{v}_e] \equiv [\boldsymbol{X}, \boldsymbol{v}_e]$, where $\mathbf{1}$ is a n-dimension vector of ones, $\boldsymbol{Z}$ is a $n \times p$ matrix, $\boldsymbol{v}_e \equiv \boldsymbol{I}(x > e)$ is a n-dimensional vector and equals 1 when $x_i > e$ and 0 otherwise.

$$(\boldsymbol{X}_e^T \boldsymbol{X}_e)^{-1} = \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{v}_e \\ \boldsymbol{v}_e^T\boldsymbol{X} & \boldsymbol{v}_e^T\boldsymbol{v}_e \end{bmatrix}^{-1} = c_e^{-1} \begin{bmatrix} c_e\boldsymbol{A} + \boldsymbol{A}\boldsymbol{X}^T\boldsymbol{v}_e\boldsymbol{v}_e^T\boldsymbol{X}\boldsymbol{A}^T & -\boldsymbol{A}\boldsymbol{X}^T\boldsymbol{v}_e \\ -\boldsymbol{v}_e^T\boldsymbol{X}\boldsymbol{A}^T & 1 \end{bmatrix}$$

where $\boldsymbol{A} \equiv (\boldsymbol{X}^T\boldsymbol{X})^{-1}$, $c_e \equiv \boldsymbol{v}_e^T\boldsymbol{v}_e - \boldsymbol{v}_e^T\boldsymbol{H}\boldsymbol{v}_e$, and $\boldsymbol{H} \equiv \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$.

$$\begin{aligned}
\boldsymbol{H_e} &= \begin{bmatrix} \boldsymbol{X} & \boldsymbol{v}_e \end{bmatrix} c_e^{-1} \begin{bmatrix} c_e\boldsymbol{A} + \boldsymbol{A}\boldsymbol{X}^T\boldsymbol{v}_e\boldsymbol{v}_e^T\boldsymbol{X}\boldsymbol{A}^T & -\boldsymbol{A}\boldsymbol{X}^T\boldsymbol{v}_e \\ -\boldsymbol{v}_e^T\boldsymbol{X}\boldsymbol{A}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}^T \\ \boldsymbol{v}_e^T \end{bmatrix} \\
&= \boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^T + c_e^{-1}\left[\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^T\boldsymbol{v}_e\boldsymbol{v}_e^T\boldsymbol{X}\boldsymbol{A}^T\boldsymbol{X}^T - \boldsymbol{v}_e\boldsymbol{v}_e^T\boldsymbol{X}\boldsymbol{A}^T\boldsymbol{X}^T - \boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^T\boldsymbol{v}_e\boldsymbol{v}_e^T + \boldsymbol{v}_e\boldsymbol{v}_e^T\right] \\
&= \boldsymbol{H} + c_e^{-1}\left[\left[\boldsymbol{H}\boldsymbol{v}_e\right]\left[\boldsymbol{H}\boldsymbol{v}_e\right]^T - \boldsymbol{v}_e\boldsymbol{v}_e^T\boldsymbol{H} - \boldsymbol{H}\boldsymbol{v}_e\boldsymbol{v}_e^T + \boldsymbol{v}_e\boldsymbol{v}_e^T\right] \\
&= \boldsymbol{H} + c_e^{-1}\left[\left(\boldsymbol{H} - \boldsymbol{I}\right)\boldsymbol{v}_e\right]\left[\left(\boldsymbol{H} - \boldsymbol{I}\right)\boldsymbol{v}_e\right]^T
\end{aligned}$$

Hence,

$$\begin{aligned}
\boldsymbol{Y}^T\boldsymbol{H_e}\boldsymbol{Y} &= \boldsymbol{Y}^T\left\{\boldsymbol{H} + c_e^{-1}\left[\left(\boldsymbol{H} - \boldsymbol{I}\right)\boldsymbol{v}_e\right]\left[\left(\boldsymbol{H} - \boldsymbol{I}\right)\boldsymbol{v}_e\right]^T\right\}\boldsymbol{Y} \\
&= \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + c_e^{-1}\boldsymbol{Y}^T\left[\left(\boldsymbol{H} - \boldsymbol{I}\right)\boldsymbol{v}_e\right]\left[\left(\boldsymbol{H} - \boldsymbol{I}\right)\boldsymbol{v}_e\right]^T\boldsymbol{Y} \\
&= \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + c_e^{-1}\left[\boldsymbol{v}_e^T\left(\boldsymbol{H} - \boldsymbol{I}\right)\boldsymbol{Y}\right]^2 \\
&\equiv \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + c_e^{-1}\left(\boldsymbol{v}_e^T\boldsymbol{r}\right)^2, \quad \text{where } \boldsymbol{r} \equiv (\boldsymbol{H} - \boldsymbol{I})\boldsymbol{Y}. \qquad (1)
\end{aligned}$$

C Additional info for Section 3 Monte Carlo studies

n	Step	Sig_15	Sig_5	Sig_1	Quad
1000	355	550	565	575	575
1500	515	825	830	830	830
2000	680	1100	1120	1120	1100
4000	1328	2238	2219	2204	2148
8000	2656	4402	4425	4474	4315
16000	5374	8857	8937	8947	8832
32000	10366	17681	17606	17805	17506

Table C.1: Monte Carlo-derived optimal block size for subsampling.

	e		β		α		α_z	
n	DBL	Rule	DBL	Rule	DBL	Rule	DBL	Rule
Step								
250	98.0 (.77)	95.4 (.89)	96.0 (.17)	99.1 (.20)	96.2 (.13)	99.2 (.15)	95.9 (.09)	99.2 (.10)
500	98.2 (.39)	94.9 (.43)	96.0 (.12)	99.2 (.14)	96.3 (.09)	99.0 (.10)	96.0 (.06)	99.1 (.07)
1000	98.2 (.20)	95.3 (.21)	96.1 (.09)	99.2 (.10)	96.5 (.06)	99.4 (.07)	96.5 (.05)	99.3 (.05)
1500	98.2 (.13)	95.0 (.14)	96.2 (.07)	99.2 (.08)	96.5 (.05)	99.2 (.06)	96.5 (.04)	99.3 (.04)
2000	98.4 (.10)	94.6 (.10)	96.9 (.06)	99.4 (.07)	96.8 (.05)	99.3 (.05)	96.2 (.03)	99.3 (.04)
Sig_15								
250	97.4 (.76)	91.3 (.52)	95.0 (.15)	90.7 (.13)	95.7 (.12)	92.1 (.10)	95.3 (.08)	92.3 (.07)
500	97.6 (.43)	93.3 (.32)	94.5 (.11)	91.0 (.09)	95.0 (.08)	91.6 (.07)	95.3 (.05)	91.6 (.05)
1000	97.1 (.27)	94.7 (.22)	93.6 (.07)	90.4 (.07)	94.7 (.05)	92.3 (.05)	94.7 (.04)	92.0 (.03)
1500	96.9 (.21)	94.8 (.18)	93.9 (.06)	91.1 (.05)	94.7 (.04)	91.9 (.04)	94.6 (.03)	92.1 (.03)
2000	96.7 (.18)	95.0 (.16)	93.5 (.05)	90.9 (.05)	94.3 (.04)	92.6 (.03)	94.5 (.03)	92.3 (.02)
Sig_5								
250	96.6 (.99)	93.7 (.80)	92.7 (.14)	89.2 (.13)	94.8 (.12)	92.5 (.11)	94.6 (.08)	92.7 (.07)
500	96.8 (.67)	94.8 (.57)	92.8 (.10)	89.3 (.09)	94.5 (.08)	92.3 (.07)	94.4 (.05)	92.0 (.05)
1000	96.1 (.47)	95.0 (.42)	91.9 (.07)	89.1 (.07)	94.8 (.06)	93.2 (.05)	94.4 (.04)	92.2 (.03)
1500	96.2 (.40)	94.9 (.36)	92.5 (.06)	90.1 (.05)	94.8 (.05)	93.4 (.04)	94.1 (.03)	92.2 (.03)
2000	96.1 (.36)	95.3 (.32)	92.4 (.05)	90.5 (.05)	95.0 (.04)	93.8 (.04)	93.8 (.03)	92.1 (.02)
Sig_1								
250	95.3 (2.62)	95.0 (2.40)	77.0 (.15)	73.6 (.14)	94.3 (.16)	93.1 (.15)	93.9 (.07)	92.8 (.07)
500	95.3 (1.99)	95.2 (1.83)	80.8 (.10)	77.7 (.09)	94.1 (.11)	93.6 (.10)	93.4 (.05)	92.3 (.05)
1000	96.2 (1.54)	95.6 (1.43)	83.5 (.07)	80.5 (.07)	95.6 (.08)	94.7 (.08)	94.0 (.04)	92.4 (.03)
1500	95.5 (1.34)	94.9 (1.24)	84.5 (.06)	81.7 (.05)	95.7 (.07)	95.0 (.06)	93.8 (.03)	92.3 (.03)
2000	95.9 (1.21)	95.4 (1.11)	85.4 (.05)	82.6 (.05)	96.0 (.06)	95.6 (.06)	93.8 (.03)	92.4 (.02)
Quad								
250	96.8 (.56)	93.1 (.45)	95.6 (.99)	92.6 (.84)	96.3 (.66)	93.5 (.53)	96.9 (.49)	94.6 (.41)
500	96.4 (.38)	93.8 (.33)	94.7 (.66)	91.7 (.58)	95.5 (.45)	93.3 (.39)	96.2 (.31)	94.3 (.27)
1000	96.1 (.28)	94.8 (.25)	94.7 (.47)	94.0 (.44)	95.2 (.33)	94.5 (.30)	95.5 (.21)	94.0 (.18)
1500	96.1 (.23)	94.4 (.21)	94.6 (.38)	93.0 (.34)	95.8 (.27)	94.3 (.24)	95.2 (.16)	93.3 (.15)
2000	95.6 (.20)	94.3 (.19)	94.3 (.33)	93.1 (.30)	95.4 (.24)	94.5 (.22)	95.1 (.14)	93.7 (.13)

Table C.2: Coverage and median width of subsampling confidence intervals (in parentheses) for all model parameters based on 10,000 Monte Carlo runs. DBL and Rule use the double bootstrap-like procedure and the rule-of-thumb to choose block size, respectively.

	Step		Sig_5		Quad	
n	m	cvg (width)	m	cvg (width)	m	cvg (width)
$B_1 = 800, B_2 = 50$						
250	102	98.7 (.87)	114	98.1 (1.13)	103	98.6 (.67)
500	188	98.8 (.45)	226	98.3 (.75)	222	98.4 (.44)
1000	364	99.0 (.23)	457	98.2 (.53)	476	98.0 (.31)
2000	725	99.0 (.11)	922	98.0 (.40)	919	97.7 (.23)
$B_1 = 400, B_2 = 100$						
250	108	98.2 (.78)	125	97.1 (1.03)	121	97.4 (.58)
500	207	98.4 (.40)	248	97.4 (.69)	252	97.0 (.40)
1000	396	98.5 (.20)	498	97.0 (.49)	497	97.1 (.29)
2000	778	98.6 (.10)	1013	96.8 (.37)	999	96.4 (.21)
$B_1 = 200, B_2 = 200$						
250	109	98.0 (.77)	129	96.6 (.99)	126	96.8 (.56)
500	212	98.2 (.39)	255	96.8 (.67)	259	96.4 (.38)
1000	404	98.2 (.20)	518	96.1 (.47)	511	96.1 (.28)
2000	787	98.4 (.10)	1047	96.1 (.36)	1037	95.6 (.20)
$B_1 = 100, B_2 = 400$						
250	111	97.7 (.73)	129	96.1 (.97)	129	96.2 (.55)
500	212	97.8 (.37)	259	96.2 (.65)	265	95.7 (.37)
1000	425	97.8 (.19)	528	95.6 (.46)	526	95.6 (.27)
2000	818	98.1 (.10)	1068	95.4 (.35)	1068	95.1 (.20)
$B_1 = 50, B_2 = 800$						
250	110	97.4 (.73)	127	95.5 (.97)	125	96.0 (.55)
500	216	97.5 (.37)	261	95.5 (.65)	267	95.4 (.37)
1000	417	97.4 (.19)	534	95.1 (.46)	534	95.0 (.27)
2000	818	97.7 (.09)	1068	95.0 (.35)	1068	94.4 (.20)

Table C.3: Subsampling coverage of e using a double bootstrap-like procedure with different combinations of B_1 and B_2. m: selected block size; cvg: coverage probabilities of e estimated from 10^4 Monte Carlo replicates; width: average width of subsampling confidence intervals.

D Additional info for Section 4 Real Data Example

Variable	Unit
Potential evapotranspiration (PET) anomaly (atmospheric demand)	mm
Top soil moisture layer anomaly (0-10 cm)	mm
Deep soil moisture layer anomaly (100- 500 cm)	mm
Mean cattle density	
Top Soil moisture layer trend	mm month^{-1}
Deep Soil moisture layer trend	mm month^{-1}
Potential evapotranspiration (PET) trend (atmospheric demand)	mm month^{-1}
Ratio of C3 to C4 grasses	
Woody vegetation cover proportion	
Dryland agriculture proportion	
Irrigated agriculture proportion	

Table D.1: Variables that are adjusted in the real data example.

E Additional Monte Carlo results

E.1 Non-central threshold

In this subsection we conduct simulation studies to examine the performance of different methods when the threshold e is not located in the middle of the distribution of X. We keep the mean model the same as before, but let $X \sim Unif(3.1, 9.5)$, which is shifted 1.6 to the right compared to the original distribution of $\sim Unif(1.5, 7.9)$.

We perform simulation studies under three data generation models: Step, Sig_5, and Quad. Under Step, e_0 remains 4.7, and 25% of the probability mass of X lies to its left. Under Sig_5, e_0 changes to 4.72 (based on the average of 10 Monte Carlo replicates with sample size 10^6) and 25.3% of the probability mass of X lies to its left. Under Quad, e_0 changes to 6.82 (based on the average of 10 Monte Carlo replicates with sample size 10^6) and 58% of the probability mass of X lies to its left.

The performance of percentile and symmetric Efron bootstrap confidence intervals for all model parameters is shown in Table E.1. The coverage probabilities are similar to the original Monte Carlo study. In general, the percentile bootstrap confidence intervals seriously undercover; the symmetric bootstrap confidence intervals have some over-coverage for e, but provide good coverage for all the slope parameters. Also, as might be expected, the precision of the parameter estimates decreases when the threshold is not located in the center of the X's distribution. For example, when $n = 250$, the symmetric bootstrap confidence interval for β has a coverage of 95.1% and 95.0% in the original Monte Carlo study and the current study, respectively, with mean widths of 0.18 and 0.15, respectively.

	e		β		α		α_z	
n	Percentile	Symmetric	Percentile	Symmetric	Percentile	Symmetric	Percentile	Symmetric
Step								
250	85.3 (.54)	96.8 (.54)	94.8 (.18)	95.0 (.18)	95.3 (.16)	95.9 (.16)	94.6 (.08)	94.6 (.08)
500	83.1 (.24)	96.5 (.24)	94.8 (.12)	95.1 (.12)	95.2 (.11)	95.3 (.11)	94.4 (.05)	94.6 (.05)
1000	83.1 (.12)	96.8 (.12)	94.8 (.09)	94.9 (.09)	95.1 (.08)	95.2 (.08)	95.0 (.04)	95.1 (.04)
2000	82.2 (.06)	96.4 (.06)	94.8 (.06)	95.0 (.06)	95.1 (.05)	95.1 (.05)	94.7 (.03)	94.9 (.03)
Sig_5								
250	95.2 (1.00)	97.0 (1.00)	92.6 (.19)	94.3 (.19)	95.2 (.19)	95.5 (.19)	94.8 (.08)	94.8 (.08)
500	95.6 (.66)	96.6 (.66)	93.3 (.13)	94.5 (.13)	95.4 (.13)	95.6 (.13)	94.7 (.05)	94.8 (.05)
1000	96.5 (.48)	97.1 (.48)	93.9 (.09)	95.0 (.09)	96.2 (.09)	96.0 (.09)	95.2 (.04)	95.4 (.04)
2000	96.5 (.36)	96.5 (.36)	94.2 (.07)	94.8 (.07)	96.5 (.07)	96.2 (.07)	95.0 (.03)	95.0 (.03)
Quad								
250	95.3 (.48)	96.7 (.48)	95.2 (1.30)	94.9 (1.30)	95.6 (1.01)	95.0 (1.01)	96.2 (.71)	95.7 (.71)
500	95.7 (.34)	96.9 (.34)	95.6 (.91)	95.3 (.91)	96.3 (.75)	95.9 (.75)	96.4 (.48)	96.1 (.48)
1000	96.0 (.25)	96.7 (.25)	95.9 (.66)	95.7 (.66)	96.4 (.54)	95.8 (.54)	96.1 (.32)	96.2 (.32)
2000	95.8 (.19)	96.7 (.19)	95.6 (.45)	95.6 (.45)	96.1 (.38)	95.6 (.38)	95.8 (.22)	96.0 (.22)

Table E.1: Simulation results from 10,000 Monte Carlo runs when $X \sim Unif(3.1, 9.5)$. Estimated coverage and mean width of percentile and symmetric Efron bootstrap CIs.

The performance of subsampling confidence intervals for the threshold parameter e, using either a double bootstrap-like procedure or a rule-of-thumb to select block size, is shown in Table E.2. The results are again similar to those from the original Monte Carlo study. It is especially reassuring to see that selecting block size by the proposed rule-of-thumb performs similarly as in the original Monte Carlo study.

	Step		Sig_5		Quad	
n	m	cvg (width)	m	cvg (width)	m	cvg (width)
Efron bootstrap - symmetric						
250		96.8 (.54)		97.0 (1.00)		96.7 (.48)
500		96.5 (.24)		96.6 (.66)		96.9 (.34)
1000		96.8 (.12)		97.1 (.48)		96.7 (.25)
2000		96.4 (.06)		96.5 (.36)		96.7 (.19)
Subsampling - DBL						
250	111	98.1 (.75)	127	97.3 (1.03)	129	96.6 (.54)
500	208	98.2 (.39)	259	96.6 (.68)	261	96.5 (.37)
1000	425	98.6 (.20)	518	96.7 (.49)	511	96.3 (.26)
2000	818	98.5 (.10)	1058	95.8 (.36)	1037	95.6 (.19)
Subsampling - rule of thumb						
250	89	96.5 (.96)	140	94.5 (.84)	140	93.6 (.45)
500	176	95.0 (.43)	279	94.9 (.59)	279	94.1 (.32)
1000	347	95.3 (.22)	557	95.2 (.44)	557	95.0 (.24)
2000	686	95.5 (.11)	1112	95.0 (.33)	1112	94.0 (.17)

Table E.2: Coverage of e when $X \sim Unif(3.1, 9.5)$. m: selected block size; cvg: coverage probabilities of e estimated from 10^4 Monte Carlo replicates; width: average width of subsampling confidence intervals.

E.2 Non-Gaussian noise

In this subsection we conduct simulation studies to examine the performance of different methods when the noise term in the data generation model is not normally distributed. Instead of having $\epsilon \sim N(0, \sigma = 0.3)$, we let $\epsilon \sim t(df = 4) * 0.3/\sqrt{2}$, whose standard deviation is also 0.3.

The performance of percentile and symmetric Efron bootstrap confidence intervals for all model parameters is shown in Table E.3. The coverage probabilities are similar to the original Monte Carlo study. In general, the percentile bootstrap confidence intervals seriously undercover, while the symmetric bootstrap confidence intervals provide good coverage for all the parameters.

	e		β		α		α_z	
n	Percentile	Symmetric	Percentile	Symmetric	Percentile	Symmetric	Percentile	Symmetric
Step								
250	77.9 (.48)	95.5 (.48)	94.4 (.15)	94.8 (.15)	95.0 (.11)	95.5 (.11)	94.6 (.07)	95.1 (.07)
500	76.4 (.23)	95.4 (.23)	94.8 (.10)	95.1 (.10)	94.9 (.07)	95.2 (.07)	94.4 (.05)	94.9 (.05)
1000	76.4 (.11)	95.4 (.11)	95.0 (.07)	95.2 (.07)	94.8 (.05)	95.0 (.05)	94.3 (.04)	94.8 (.04)
2000	75.3 (.06)	95.4 (.06)	95.0 (.05)	95.1 (.05)	94.5 (.04)	94.7 (.04)	94.8 (.03)	94.9 (.03)
Sig_5								
250	94.4 (.89)	96.4 (.89)	92.2 (.15)	94.1 (.15)	95.3 (.12)	95.9 (.12)	94.8 (.08)	95.3 (.08)
500	95.6 (.61)	96.1 (.61)	92.6 (.10)	94.2 (.10)	95.1 (.08)	95.6 (.08)	94.7 (.05)	94.9 (.05)
1000	95.5 (.45)	96.0 (.45)	93.3 (.07)	94.4 (.07)	95.4 (.06)	95.6 (.06)	94.5 (.04)	94.8 (.04)
2000	96.2 (.34)	96.5 (.34)	93.4 (.05)	94.4 (.05)	95.5 (.04)	95.5 (.04)	95.0 (.03)	95.2 (.03)
Quad								
250	94.6 (.46)	96.5 (.46)	95.4 (.96)	95.0 (.96)	96.0 (.60)	95.2 (.60)	96.3 (.46)	95.8 (.46)
500	95.2 (.34)	96.2 (.34)	95.5 (.68)	95.1 (.68)	96.2 (.45)	95.3 (.45)	96.3 (.31)	96.0 (.31)
1000	95.2 (.25)	95.9 (.25)	95.5 (.49)	95.4 (.49)	96.2 (.32)	95.9 (.32)	96.5 (.21)	96.2 (.21)
2000	95.3 (.20)	96.1 (.20)	95.8 (.34)	95.6 (.34)	96.0 (.24)	95.8 (.24)	96.1 (.14)	96.2 (.14)

Table E.3: Simulation results from 10,000 Monte Carlo runs when X follows a Student's t distribution. Estimated coverage and mean width of percentile and symmetric Efron bootstrap CIs.

The performance of subsampling confidence intervals for the threshold parameter e, using either a double bootstrap-like procedure or a rule-of-thumb to select block size, is shown in Table E.4. Under Sig_5 and Quad, the results are similar to those from the original Monte Carlo study. Under Step, we see some interesting differences. With the double bootstrap-like procedure, the selected block sizes become smaller when the noise distribution follows a Student's t distribution instead of a normal distribution; this leads to wider confidence intervals and a similar level of coverage. These results make sense considering that the Student's t distribution with four degrees of freedom has a longer tail than the normal distribution. With the proposed rule-of-thumb, the selected block sizes are the same regardless of the noise distribution in the data generating model because the rule only depends on the sample size. The results show that the rules developed under a normal noise distribution lead to under-coverage when the noise distribution is a Student's t distribution.

	Step		Sig_5		Quad	
n	m	cvg (width)	m	cvg (width)	m	cvg (width)
Efron bootstrap - symmetric						
250		95.5 (.48)		96.4 (.89)		96.5 (.46)
500		95.4 (.23)		96.1 (.61)		96.2 (.34)
1000		95.4 (.11)		96.0 (.45)		95.9 (.25)
2000		95.4 (.06)		96.5 (.34)		96.1 (.20)
Subsampling - DBL						
250	92	97.4 (.83)	125	96.9 (.96)	125	96.6 (.56)
500	178	97.7 (.43)	252	96.6 (.65)	259	96.2 (.37)
1000	346	97.5 (.22)	505	96.4 (.47)	507	96.3 (.27)
2000	680	97.9 (.11)	1037	96.2 (.35)	1037	95.5 (.20)
Subsampling - rule of thumb						
250	89	92.3 (.80)	140	93.9 (.75)	140	92.9 (.44)
500	176	91.4 (.39)	279	95.2 (.54)	279	93.9 (.32)
1000	347	91.0 (.19)	557	94.5 (.41)	557	94.3 (.24)
2000	686	90.4 (.10)	1112	94.8 (.32)	1112	94.1 (.18)

Table E.4: Coverage of e when X follows a Student's t distribution. m: selected block size; cvg: coverage probabilities of e estimated from 10^4 Monte Carlo replicates; width: average width of subsampling confidence intervals.

F Fast Grid Search with two thresholds

Consider the following model for a step linear regression model with 2 threshold parameters:

$$Y = \alpha + \alpha_z^T z + \beta_1 I(x_1 > e) + \beta_2 I(x_2 > f) + \epsilon,$$

where e and f denote the threshold parameters ($e < f$); Y is the response; x_1 and x_2 are the predictors with threshold effects; z denotes p additional predictors (a $n \times p$ matrix) and ϵ is a normally distributed error term independent of the predictors.

To estimate the step points e and f, maximizing the log likelihood of the step linear regression models is equivalent to minimize the sum of squares of the residuals, which is expressed as: $\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}} = (\boldsymbol{Y} - \hat{\boldsymbol{Y}})^T(\boldsymbol{Y} - \hat{\boldsymbol{Y}}) = [(\boldsymbol{I} - \boldsymbol{H_{e,f}})\boldsymbol{Y}]^T[(\boldsymbol{I} - \boldsymbol{H_{e,f}})\boldsymbol{Y}] = \boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{H_{e,f}}\boldsymbol{Y}$, where $\boldsymbol{H_{e,f}} \equiv \boldsymbol{X_{e,f}}(\boldsymbol{X_{e,f}^T}\boldsymbol{X_{e,f}})^{-1}\boldsymbol{X_{e,f}^T}$ denotes the projection matrix; $\boldsymbol{X_{e,f}}$ is the design matrix of the step linear regression models.

As $\boldsymbol{Y}^T\boldsymbol{Y}$ is independent of step points e and f, it is sufficient to compare $\boldsymbol{Y}^T\boldsymbol{H_{e,f}}\boldsymbol{Y}$ for a grid of $\{e, f\}$

$$\left\{ \begin{array}{ccc} e_1, f_1 & \dots & e_1, f_K \\ & \dots & \\ e_M, f_1 & \dots & e_M, f_K \end{array} \right\}.$$

Let $\boldsymbol{X_{e,f}} \equiv [\boldsymbol{1}, \boldsymbol{z}, \boldsymbol{I}(x_1 > e), \boldsymbol{I}(x_2 > f)] \equiv [\boldsymbol{X}, \boldsymbol{v}_{e,f}]$, where $\boldsymbol{1}$ is a n-dimension vector of ones, $\boldsymbol{z}$ is a $n \times p$ matrix, $\boldsymbol{v}_{e,f} \equiv [\boldsymbol{I}(x_1 > e), \boldsymbol{I}(x_2 > f)]$ is a $n \times 2$ matrix of ones and zeros.

$$\begin{aligned} \boldsymbol{Y}^T\boldsymbol{H_{e,f}}\boldsymbol{Y} &= \boldsymbol{Y}^T \begin{bmatrix} \boldsymbol{X} & \boldsymbol{v}_{e,f} \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{X}^T \\ \boldsymbol{v}_{e,f}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{X} & \boldsymbol{v}_{e,f} \end{bmatrix} \right)^{-1} \begin{bmatrix} \boldsymbol{X}^T \\ \boldsymbol{v}_{e,f}^T \end{bmatrix} \boldsymbol{Y} \\ &= \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + (\boldsymbol{v}_{e,f}^T\boldsymbol{r})^T(\boldsymbol{v}_{e,f}^T\boldsymbol{v}_{e,f} - \boldsymbol{v}_{e,f}^T\boldsymbol{H}\boldsymbol{v}_{e,f})^{-1}\boldsymbol{v}_{e,f}^T\boldsymbol{r} \end{aligned} \tag{2}$$

where $\boldsymbol{H} \equiv \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$; $\boldsymbol{r} \equiv (\boldsymbol{H} - \boldsymbol{I})\boldsymbol{Y}$. By QR decomposition, we can further write $\boldsymbol{v}_{e,f}^T\boldsymbol{H}\boldsymbol{v}_{e,f} = (\boldsymbol{Q}_1^T\boldsymbol{v}_{e,f})^T\boldsymbol{Q}_1^T\boldsymbol{v}_{e,f}$, where $\boldsymbol{Q}_1$ is the first p columns of the orthogonal matrix Q from the QR decomposition.

To find the recursive relationship of $\boldsymbol{Y}^T\boldsymbol{H_{e,f}}\boldsymbol{Y}$, first we consider $\boldsymbol{v}_{e_t,f}$ and $\boldsymbol{v}_{e_{t+1},f}$. Let $\boldsymbol{\delta}_{t,f} = \boldsymbol{v}_{e_t,f} - \boldsymbol{v}_{e_{t+1},f}$, where $\boldsymbol{\delta}_{t,f}$ is a $n \times 2$ matrix with first column of $m + 1^{th}$ entry equal to one and zeros otherwise and with second column of zeros.

We find that:

$$\boldsymbol{v}_{e_{t+1},f}^T\boldsymbol{v}_{e_{t+1},f} = \boldsymbol{v}_{e_t,f}^T\boldsymbol{v}_{e_t,f} - 2\boldsymbol{\delta}_{t,f}^T\boldsymbol{v}_{e_t,f} + \boldsymbol{\delta}_{t,f}^T\boldsymbol{\delta}_{t,f} \tag{3}$$

$$\boldsymbol{v}_{e_{t+1},f}^T\boldsymbol{Q}_1 = \boldsymbol{v}_{e_t,f}^T\boldsymbol{Q}_1 - \boldsymbol{\delta}_{t,f}^T\boldsymbol{Q}_1 \tag{4}$$

$$\boldsymbol{v}_{e_{t+1},f}^T\boldsymbol{r} = \boldsymbol{v}_{e_t,f}^T\boldsymbol{r} - \boldsymbol{\delta}_{t,f}^T\boldsymbol{r} \tag{5}$$

where

$$\boldsymbol{\delta}_{t,f}^T\boldsymbol{v}_{e_t,f} = \boldsymbol{\delta}_{t,f}^T\boldsymbol{\delta}_{t,f} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix};$$

$\boldsymbol{\delta}_{t,f}^T\boldsymbol{Q}_1$ is a $2 \times n$ matrix with the first row of the $m + 1^{th}$ row vector of $\boldsymbol{Q}_1$ and with the second row of zeros; $\boldsymbol{\delta}_{t,f}^T\boldsymbol{r}$ is a 2×1 matrix with the first element equal to the $m + 1^{th}$ element of $\boldsymbol{r}$ and with the second element of zero.

Second, we consider $\boldsymbol{v}_{e,f_t}$ and $\boldsymbol{v}_{e,f_{t+1}}$. Let $\boldsymbol{\delta}_{t,e} = \boldsymbol{v}_{e,f_t} - \boldsymbol{v}_{e,f_{t+1}}$, where $\boldsymbol{\delta}_{t,e}$ is a $n \times 2$ matrix with first column of zeros and with the second column of $k+1^{th}$ entry equal to one and zeros otherwise.

Accordingly:

$$\boldsymbol{v}_{e,f_{t+1}}^T \boldsymbol{v}_{e,f_{t+1}} = \boldsymbol{v}_{e,f_t}^T \boldsymbol{v}_{e,f_t} - 2\boldsymbol{\delta}_{t,e}^T \boldsymbol{v}_{e,f_t} + \boldsymbol{\delta}_{t,e}^T \boldsymbol{\delta}_{t,e} \tag{6}$$

$$\boldsymbol{v}_{e,f_{t+1}}^T \boldsymbol{Q}_1 = \boldsymbol{v}_{e,f_t}^T \boldsymbol{Q}_1 - \boldsymbol{\delta}_{t,e}^T \boldsymbol{Q}_1 \tag{7}$$

$$\boldsymbol{v}_{e,f_{t+1}}^T \boldsymbol{r} = \boldsymbol{v}_{e,f_t}^T \boldsymbol{r} - \boldsymbol{\delta}_{t,e}^T \boldsymbol{r} \tag{8}$$

where

$$\boldsymbol{\delta}_{t,e}^T \boldsymbol{v}_{e,f_t} = \boldsymbol{\delta}_{t,e}^T \boldsymbol{\delta}_{t,e} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix};$$

$\boldsymbol{\delta}_{t,e}^T \boldsymbol{Q}_1$ is a $2 \times n$ matrix with the first row of zeros and with the second row of the $k+1^{th}$ row vector of $\boldsymbol{Q}_1$; $\boldsymbol{\delta}_{t,e}^T \boldsymbol{r}$ is a 2×1 matrix with the first element of zero and with the second element equal to the $k+1^{th}$ element of $\boldsymbol{r}$.

With the formulations above, we are capable of updating $\boldsymbol{Y}^T \boldsymbol{H}_{\boldsymbol{e},\boldsymbol{f}} \boldsymbol{Y}$ conditional on e_{t+1} and f_{t+1} by storing the previous values based on e_t and f_t. We write the procedures formally as follows:

Algorithm 1 *The fast grid search algorithm for step linear regression models*

1. *Sort the samples by the ascending order of* x_{1i} *and* x_{2i}
2. *Compute and store 2 copies of the initial values:* $\boldsymbol{v}_{e_1,f_1}^T \boldsymbol{v}_{e_1,f_1}$, $\boldsymbol{v}_{e_1,f_1}^T \boldsymbol{Q}_1$, $\boldsymbol{v}_{e_1,f_1}^T \boldsymbol{r}$
3. *For k in 1 to K:*
   (a) *if* $k > 1$
      i. *update* $\boldsymbol{v}_{e_1,f_k}^T \boldsymbol{v}_{e_1,f_k}$, $\boldsymbol{v}_{e_1,f_k}^T \boldsymbol{Q}_1$, $\boldsymbol{v}_{e_1,f_k}^T \boldsymbol{r}$ *based on (3), (4) and (5), and save 2 copies*
      ii. *compute and store* $\boldsymbol{Y}^T \boldsymbol{H}_{e_1,f_k} \boldsymbol{Y}^T$ *according to (2)*
   (b) *For m in 1 to M-1:*
      i. *update* $\boldsymbol{v}_{e_{m+1},f_k}^T \boldsymbol{v}_{e_{m+1},f_k}$, $\boldsymbol{v}_{e_{m+1},f_k}^T \boldsymbol{Q}_1$, $\boldsymbol{v}_{e_{m+1},f_k}^T \boldsymbol{r}$ *based on (6), (7) and (8)*
      ii. *compute and store* $\boldsymbol{Y}^T \boldsymbol{H}_{e_{m+1},f_k} \boldsymbol{Y}^T$ *according to (2)*

G Pseudocode of the fast grid search algorithm for step regression models

Algorithm 2

ord_data $\leftarrow$ *Data reordered with the smallest* $\boldsymbol{x}$ *values coming first (*$\boldsymbol{x}$ *is the step covariate)*

$H \leftarrow \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$ *where* $\boldsymbol{X}$ *is the design matrix, excluding the step covariate*

$\boldsymbol{r} \leftarrow \boldsymbol{Y}^T[H - I]$ *where* $\boldsymbol{Y}$ *is the outcome of interest, and* I *is the* n *by* n *identity matrix where* n *is the sample size*

$Q \leftarrow$ *the first* p *columns of the orthogonal matrix from the QR decomposition of* H

cand_e $\leftarrow$ *all* $\boldsymbol{x}$ *ordered values above the* 2.5% *quantile and below the* 97.5% *quantile.*

num_und $\leftarrow$ *number of values less than the smallest value in cand_e*

update_dot_dot $\leftarrow \sum_{i=1}^{num\_und} 1$

*update_dot_*$\boldsymbol{r} \leftarrow \sum_{i=1}^{num\_und} \boldsymbol{r}_i$

*update_dot_*$\boldsymbol{Q} \leftarrow \sum_{i=1}^{num\_und} \boldsymbol{Q}_i$

best_threshold $\leftarrow x_{num\_und}$

best_like_proxy $\leftarrow 0$

for $i \leftarrow num\_und + 1, num\_und + length(cand\_e)$ ***do***

    update_dot_dot $\leftarrow$ *update_dot_dot* $- 1$

    *update_dot_*$\boldsymbol{r}$ $\leftarrow$ *update_dot_*$\boldsymbol{r} - r_i$

    *update_dot_*$\boldsymbol{Q}$ $\leftarrow$ *update_dot_*$\boldsymbol{Q} - \boldsymbol{Q}_i$

    like_proxy $\leftarrow$ *update_dot_*$\boldsymbol{r}^2/($*update_dot_dot* $-$ *update_dot_*$\boldsymbol{Q}^T$*update_dot_*$\boldsymbol{Q})$

    if *like_proxy* $>$ *best_like_proxy* ***then***

        best_like_proxy $\leftarrow$ *like_proxy*

        best_threshold $\leftarrow \boldsymbol{x}_i$

    end if

end for