Supplementary material

A Skew-Normal Exponential Stochastic Frontier Model

A.1 Integral

Before deriving the $f(\epsilon)$, consider the following integral due to Owen (1980),

$$\begin{aligned}\int \phi\left(\boldsymbol{x}\right)\Phi\left(a+b\boldsymbol{x}\right)d\boldsymbol{x} = & T\left(\boldsymbol{x}, \frac{a}{\boldsymbol{x}\sqrt{1+b^2}}\right) + T\left(\frac{a}{\sqrt{1+b^2}}, \frac{\boldsymbol{x}\sqrt{1+b^2}}{a}\right) \\ & - T\left(\boldsymbol{x}, \frac{a+b\boldsymbol{x}}{\boldsymbol{x}}\right) - T\left(\frac{a}{\sqrt{1+b^2}}, \frac{ab+\boldsymbol{x}(1+b^2)}{a}\right) \\ & + \Phi\left(\boldsymbol{x}\right)\Phi\left(\frac{a}{\sqrt{1+b^2}}\right) \end{aligned} \tag{A1}$$

Denote $a_2 = \dfrac{a}{\sqrt{1+b^2}}$, then (A1) becomes

$$\underbrace{T\left(\boldsymbol{x}, \frac{a_2}{\boldsymbol{x}}\right)}_{\text{term 1}} + \underbrace{T\left(a_2, \frac{\boldsymbol{x}}{a_2}\right)}_{\text{term 2}} - \underbrace{T\left(\boldsymbol{x}, b+\frac{a}{\boldsymbol{x}}\right)}_{\text{term 3}} - \underbrace{T\left(a_2, b+\frac{\boldsymbol{x}(1+b^2)}{a}\right)}_{\text{term 4}} + \underbrace{\Phi\left(\boldsymbol{x}\right)\Phi\left(a_2\right)}_{\text{term 5}}. \tag{A2}$$

Integral (A2) is needed on the domain $[x_1, +\infty)$. We will use three properties of Owen's T function (see Owen, 1956) to simplify the integral in (A2). First,

$$T(H,A) + T\left(AH, \frac{1}{A}\right) = \begin{cases} \frac{1}{2}\left(\Phi(H)+\Phi(AH)\right) - \Phi(H)\Phi(AH) & \text{if} \quad A \geq 0 \\ \frac{1}{2}\left(\Phi(H)+\Phi(AH)\right) - \Phi(H)\Phi(AH) - \frac{1}{2} & \text{if} \quad A < 0 \end{cases}. \tag{A3}$$

Second,

$$T(H,0) = 0. \tag{A4}$$

Third,

$$T(H,+\infty) = \frac{1}{2}\Phi\left(-\left|H\right|\right). \tag{A5}$$

Integrate individual terms in (A2)

$$
\begin{aligned}
(\text{term 1})\Big|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} &= \underbrace{T\left(\boldsymbol{x}, \frac{a_2}{\boldsymbol{x}}\right)}_{\text{term 1}}\Bigg|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} && \text{(A6)} \\
&= T\left(+\infty, \frac{a_2}{+\infty}\right) - T\left(x_1, \frac{a_2}{x_1}\right) = -T\left(x_1, \frac{a_2}{x_1}\right), \\
(\text{term 2})\Big|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} &= \underbrace{T\left(a_2, \frac{\boldsymbol{x}}{a_2}\right)}_{\text{term 2}}\Bigg|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} && \text{(A7)} \\
&= T\left(a_2, +\infty\right) - T\left(a_2, \frac{x_1}{a_2}\right), \\
(\text{term 3})\Big|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} &= \underbrace{-T\left(\boldsymbol{x}, b + \frac{a}{\boldsymbol{x}}\right)}_{\text{term 3}}\Bigg|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} && \text{(A8)} \\
&= -T\left(+\infty, b + \frac{a}{+\infty}\right) + T\left(x_1, b + \frac{a}{x_1}\right) = T\left(x_1, b + \frac{a}{x_1}\right), \\
(\text{term 4})\Big|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} &= \underbrace{-T\left(a_2, b + \frac{\boldsymbol{x}(1+b^2)}{a}\right)}_{\text{term 4}}\Bigg|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} && \text{(A9)} \\
&= -T\left(a_2, +\infty\right) + T\left(a_2, b + \frac{x_1(1+b^2)}{a}\right), \\
(\text{term 5})\Big|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} &= \underbrace{\Phi\left(\boldsymbol{x}\right)\Phi\left(a_2\right)}_{\text{term 5}}\Bigg|_{\boldsymbol{x}=x_1}^{\boldsymbol{x}=+\infty} && \text{(A10)} \\
&= \Phi\left(+\infty\right)\Phi\left(a_2\right) - \Phi\left(x_1\right)\Phi\left(a_2\right) = \Phi\left(a_2\right) - \Phi\left(x_1\right)\Phi\left(a_2\right).
\end{aligned}
$$

Adding these 5 terms yields the closed form solution to the integral in (A2):

$$
\begin{aligned}
\int_{x_1}^{\infty} \phi\left(\boldsymbol{x}\right)\Phi\left(a + b\boldsymbol{x}\right) d\boldsymbol{x} = {} & \underbrace{-T\left(x_1, \frac{a_2}{x_1}\right)}_{\text{term 1}} \underbrace{-T\left(a_2, \frac{x_1}{a_2}\right)}_{\text{term 2}} + \underbrace{T\left(x_1, b + \frac{a}{x_1}\right)}_{\text{term 3}} + \\
& \underbrace{T\left(a_2, b + \frac{x_1(1+b^2)}{a}\right)}_{\text{term 4}} + \underbrace{\Phi\left(a_2\right)\Phi\left(-x_1\right)}_{\text{term 5}}. \qquad \text{(A11)}
\end{aligned}
$$

We can leave it as is or we can simplify it further by using the first property (A3) of Owen's

T function. Denote $H = x_1$ and $A = \dfrac{a_2}{x_1}$ and consider the first two terms in (A11). Then

$$\underbrace{-T\left(x_1, \frac{a_2}{x_1}\right)}_{\text{term 1}} - \underbrace{T\left(a_2, \frac{x_1}{a_2}\right)}_{\text{term 2}} \tag{A12}$$

$$= -\frac{1}{2}\left[\Phi(H) + \Phi(AH)\right] + \Phi(H)\Phi(AH) + \frac{1}{2}I\left(A < 0\right)$$

$$= -\frac{1}{2}\left[\Phi(x_1) + \Phi\left(a_2\right)\right] + \Phi(x_1)\Phi\left(a_2\right) + \frac{1}{2}I\left(\frac{a_2}{x_1} < 0\right). \tag{A13}$$

Then the sum of term 1, term 2, and term 5 in (A11) is given by

$$\underbrace{-T\left(x_1, \frac{a_2}{x_1}\right)}_{\text{term 1}} - \underbrace{T\left(a_2, \frac{x_1}{a_2}\right)}_{\text{term 2}} + \underbrace{\Phi\left(a_2\right) - \Phi\left(x_1\right)\Phi\left(a_2\right)}_{\text{term 5}} \tag{A14}$$

$$= -\frac{1}{2}\Phi(x_1) - \frac{1}{2}\Phi\left(a_2\right) + \Phi(x_1)\Phi\left(a_2\right) + \frac{1}{2}I\left(\frac{a_2}{x_1} < 0\right) + \Phi\left(a_2\right) - \Phi\left(x_1\right)\Phi\left(a_2\right)$$

$$= -\frac{1}{2}\Phi(x_1) + \frac{1}{2}\Phi\left(a_2\right) + \frac{1}{2}I\left(\frac{a_2}{x_1} < 0\right). \tag{A15}$$

Thus,

$$\int_{x_1}^{\infty} \phi\left(\boldsymbol{x}\right)\Phi\left(a + b\boldsymbol{x}\right)d\boldsymbol{x} = \underbrace{T\left(x_1, b + \frac{a}{x_1}\right)}_{\text{term 3}} + \underbrace{T\left(a_2, b + \frac{x_1(1+b^2)}{a}\right)}_{\text{term 4}}$$
$$+ \underbrace{\frac{1}{2}\left[\Phi\left(a_2\right) - \Phi\left(x_1\right) + I\left(\frac{a_2}{x_1} < 0\right)\right]}_{\text{term 1 + term 2 + term 5}}. \tag{A16}$$

A.2 Derivation of $f(\epsilon)$ and the Log-Likelihood Function

The model is

$$y = f(\boldsymbol{x}; \boldsymbol{\beta}) + v - \mathsf{p}u = f(\boldsymbol{x}; \boldsymbol{\beta}) + \epsilon, \tag{A17}$$

in which the inefficiency term is exponentially distributed, so its density is given by

$$f_u(u) = \lambda \exp\left(-\lambda u\right),$$

where $\lambda = \dfrac{1}{\sigma_u}$. The SN distribution where the location is not 0 is given by

$$v \sim SN(\xi, \sigma_v^2, \alpha). \tag{A18}$$

The mean is of the random variable v is

$$E(v) = \xi + \sigma_v \sqrt{\frac{2}{\pi}} \frac{\alpha}{\sqrt{1+\alpha^2}}. \tag{A19}$$

The assumption $E(v) = 0$ is fulfilled when

$$\xi = -\sigma_v \sqrt{\frac{2}{\pi}} \frac{\alpha}{\sqrt{1+\alpha^2}}. \tag{A20}$$

Thus,

$$E\left(v + \sigma_v \sqrt{\frac{2}{\pi}} \frac{\alpha}{\sqrt{1+\alpha^2}}\right) = 0.$$

We keep ξ for tractability. The density of v with a non-zero location is

$$f_v(v) = \frac{2}{\sigma_v} \phi\left(\frac{v-\xi}{\sigma_v}\right) \Phi\left(\alpha \frac{v-\xi}{\sigma_v}\right). \tag{A21}$$

Given that in our SF model, $\epsilon = v - \mathsf{p}u$, then $v - \xi = \epsilon + pu - \xi = \epsilon_r + \mathsf{p}u$, where $\epsilon_r = \epsilon - \xi$, the joint density of u and ϵ is

$$\begin{aligned} f(\epsilon, u) &= \underbrace{\frac{2}{\sigma_v} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right)^2\right] \Phi\left(\alpha \frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right)}_{f_v(v)} \underbrace{\lambda \exp(-\lambda u)}_{f_u(u)} \\ &= \frac{2\lambda}{\sigma_v} \frac{1}{\sqrt{2\pi}} \Phi\left(\alpha \frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) \exp\left[-\frac{1}{2}\left\{\left(\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right)^2 + 2\lambda u\right\}\right]. \end{aligned} \tag{A22}$$

A.2.1 Simplification: Sum of Powers of Exponents

Consider the expression in curly parentheses in (A22). Keeping in mind that that $\mathsf{p}^2 = 1$,

$$
\begin{aligned}
&\left(\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right)^2 + 2\lambda u = \frac{1}{\sigma_v^2}\left[\epsilon_r^2 + 2\mathsf{p}u\epsilon_r + \mathsf{p}^2u^2 + 2\lambda u\sigma_v^2\right] \\
&= \frac{1}{\sigma_v^2}\left[u^2 + 2u(\mathsf{p}\epsilon_r + \lambda\sigma_v^2) + \epsilon_r^2\right] \\
&= \frac{1}{\sigma_v^2}\left[u^2 + 2u(\mathsf{p}\epsilon_r + \lambda\sigma_v^2) + \epsilon_r^2 + (\mathsf{p}\epsilon_r + \lambda\sigma_v^2)^2 - (\mathsf{p}\epsilon_r + \lambda\sigma_v^2)^2\right] \\
&= \frac{1}{\sigma_v^2}\left[(u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2)^2 + \epsilon_r^2 - (\mathsf{p}\epsilon_r + \lambda\sigma_v^2)^2\right] \\
&= \frac{1}{\sigma_v^2}\left[(u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2)^2 + \epsilon_r^2 - \mathsf{p}^2\epsilon_r^2 - 2\mathsf{p}\epsilon_r\lambda\sigma_v^2 - \lambda^2\sigma_v^4\right] \\
&= \frac{1}{\sigma_v^2}\left[(u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2)^2 - \sigma_v^2\left(2\mathsf{p}\epsilon_r\lambda + \lambda^2\sigma_v^2\right)\right] \\
&= \frac{1}{\sigma_v^2}\left[(u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2)^2\right] - \left(2\mathsf{p}\epsilon_r\lambda + \lambda^2\sigma_v^2\right). \qquad \text{(A23)}
\end{aligned}
$$

A.2.2 Joint Density of u and ϵ Rewritten

Using (A23), the expression in (A22) becomes

$$
\begin{aligned}
&f(\epsilon, u) = \\
&= \frac{2\lambda}{\sigma_v}\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left\{\left(\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right)^2 + 2\lambda u\right\}\right]\Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) \\
&= \frac{2\lambda}{\sigma_v}\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left\{\frac{1}{\sigma_v^2}\left[(u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2)^2\right] - \left(2\mathsf{p}\epsilon_r\lambda + \lambda^2\sigma_v^2\right)\right\}\right]\Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) \\
&= \frac{2\lambda}{\sigma_v}\exp\left(\mathsf{p}\epsilon_r\lambda + \frac{\lambda^2\sigma_v^2}{2}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2}{\sigma_v}\right)^2\right]\Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) \\
&= \frac{2\lambda}{\sigma_v}\exp\left(\mathsf{p}\epsilon_r\lambda + \frac{\lambda^2\sigma_v^2}{2}\right)\phi\left(\frac{u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2}{\sigma_v}\right)\Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right). \qquad \text{(A24)}
\end{aligned}
$$

A.2.3 Marginal Density and Log-Likelihood

The marginal density of ϵ is obtained by integrating u out of (A24),

$$
\begin{aligned}
f(\epsilon) &= \int_0^\infty f(\epsilon, u) du \\
&= \int_0^\infty \frac{2\lambda}{\sigma_v} \exp\left(\mathsf{p}\epsilon_r\lambda + \frac{\lambda^2\sigma_v^2}{2}\right) \phi\left(\frac{u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2}{\sigma_v}\right) \Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) du \\
&= \frac{2\lambda}{\sigma_v} \exp\left(\mathsf{p}\epsilon_r\lambda + \frac{\lambda^2\sigma_v^2}{2}\right) \int_0^\infty \phi\left(\frac{u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2}{\sigma_v}\right) \Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) du.
\end{aligned}
$$

To integrate

$$
\int_0^\infty \phi\left(\frac{u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2}{\sigma_v}\right) \Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) du, \tag{A25}
$$

denote the rescaled and shifted u as

$$
u_{rs} = \frac{u + \mathsf{p}\epsilon_r + \lambda\sigma_v^2}{\sigma_v}. \tag{A26}
$$

Then $u = u_{rs}\sigma_v - \mathsf{p}\epsilon_r - \lambda\sigma_v^2$, $du = du_{rs}\sigma_v$ and (keeping in mind that $\mathsf{p}^2 = 1$)

$$
\begin{aligned}
\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v} &= \alpha\frac{\epsilon_r + \mathsf{p}(u_{rs}\sigma_v - \mathsf{p}\epsilon_r - \lambda\sigma_v^2)}{\sigma_v} \\
&= \alpha\frac{\epsilon_r + \mathsf{p}u_{rs}\sigma_v - \mathsf{p}^2\epsilon_r - \mathsf{p}\lambda\sigma_v^2}{\sigma_v} \\
&= \underbrace{-\alpha\mathsf{p}\lambda\sigma_v}_{a} + \underbrace{\alpha\mathsf{p}}_{b} u_{rs} \\
&= a + bu_{rs}.
\end{aligned} \tag{A27}
$$

Denote

$$
u_1 = \frac{\mathsf{p}\epsilon_r + \lambda\sigma_v^2}{\sigma_v},
$$

which is $u_{rs}|\{u = 0\}$. Then the integral in (A25) can be written as

$$
\sigma_v \int_{u_1}^\infty \phi(u_{rs}) \Phi(a + bu_{rs}) du_{rs} \tag{A28}
$$

and

$$f(\epsilon) = 2\lambda \exp\left(\mathsf{p}\epsilon_r\lambda + \frac{\lambda^2\sigma_v^2}{2}\right)\int_{u_1}^{\infty}\phi\left(u_{rs}\right)\Phi\left(a+bu_{rs}\right)du_{rs}. \tag{A29}$$

Using the result in (A11)[23]

$$\begin{aligned}\mathcal{A} &= \int_{u_1}^{\infty}\phi\left(u_{rs}\right)\Phi\left(a+bu_{rs}\right)du_{rs}\\ &= -T\left(u_1,\frac{a_2}{u_1}\right) - T\left(a_2,\frac{u_1}{a_2}\right) + T\left(u_1, b+\frac{a}{u_1}\right)\\ &\quad + T\left(a_2, b+\frac{u_1(1+b^2)}{a}\right) + \Phi\left(a_2\right)\Phi\left(-u_1\right). \end{aligned}\tag{A30}$$

Then,

$$\begin{aligned} f(\epsilon) &= 2\lambda\exp\left(\mathsf{p}\epsilon_r\lambda + \frac{\lambda^2\sigma_v^2}{2}\right)\times\mathcal{A}\\ &= 2\lambda\exp\left(\mathsf{p}\epsilon_r\lambda + \frac{\lambda^2\sigma_v^2}{2}\right)\left[\begin{array}{l} -T\left(u_1,\frac{a_2}{u_1}\right) - T\left(a_2,\frac{u_1}{a_2}\right) + T\left(u_1, b+\frac{a}{u_1}\right)\\ +T\left(a_2, b+\frac{u_1(1+b^2)}{a}\right) + \Phi\left(a_2\right)\Phi\left(-u_1\right)\end{array}\right], \end{aligned}\tag{A31}$$

where $a = -\alpha\mathsf{p}\lambda\sigma_v$, $b = \alpha\mathsf{p}$, $a_2 = a/\sqrt{1+b^2}$, and $u_1 = \mathsf{p}\epsilon_r/\sigma_v + \lambda\sigma_v$. The log-likelihood based on (A31) is therefore

$$logl = \ln\left(2\lambda\right) + \mathsf{p}\epsilon_r\lambda + \frac{\lambda^2\sigma_v^2}{2} + \ln\mathcal{A}. \tag{A32}$$

A.2.4 Gradient

To derive the gradient, note that

$$\frac{\partial T(h,a)}{\partial\omega} = T_1'(h,a)\frac{\partial h}{\partial\omega} + T_2'(h,a)\frac{\partial a}{\partial\omega}, \tag{A33}$$

[23] The result (A16) can also be used.

where

$$T_1'(h,a) = \frac{\partial T(h,a)}{\partial h} = -\frac{1}{2\sqrt{2\pi}}(2\Phi(ha)-1)\exp\left(-\frac{h^2}{2}\right) = -\phi(h)(\Phi(ha)-0.5)$$

and

$$T_2'(h,a) = \frac{\partial T(h,a)}{\partial a} = \frac{1}{2\pi(1+a^2)}\exp\left(-\frac{h^2}{2}\left(1+a^2\right)\right).$$

Further denote,

$$A = \alpha/\sqrt{1+\alpha^2}$$
$$SVU = \lambda\sigma_v.$$

Given our specifications, the following notations for derivatives are needed

$$\epsilon_\beta = -\boldsymbol{x}$$
$$\alpha_s = -\boldsymbol{z}_s$$
$$A_s = \frac{\partial A}{\partial \boldsymbol{\gamma}_s} = (1.0+\alpha^2)^{-3/2}\boldsymbol{z}_s$$
$$SV1 = \frac{\partial(1/\sigma_v)}{\partial \boldsymbol{\gamma}_v} = -0.5\frac{\boldsymbol{z}_v}{\sigma_v}$$
$$SU1 = \frac{\partial(1/\sigma_u)}{\partial \boldsymbol{\gamma}_u} = -0.5\frac{\boldsymbol{z}_u}{\sigma_u}$$
$$SVU_u = -0.5 SVU\boldsymbol{z}_u$$
$$SVU_v = 0.5 SVU\boldsymbol{z}_v.$$

The partial derivatives of u_1 are:

$$\frac{\partial u_1}{\partial \boldsymbol{\beta}} = \frac{\mathsf{p}\epsilon_\beta}{\sigma_v}$$
$$\frac{\partial u_1}{\partial \boldsymbol{\gamma}_v} = \mathsf{p}\epsilon SV1 + SVU_v$$
$$\frac{\partial u_1}{\partial \boldsymbol{\gamma}_s} = \sqrt{\frac{2}{\pi}}A_s$$

$$\frac{\partial u_1}{\partial \boldsymbol{\gamma}_u} = SVU_u.$$

The partial derivatives of a_2 are:

$$\begin{aligned}
\frac{\partial a_2}{\partial \boldsymbol{\beta}} &= \mathbf{0} \\
\frac{\partial a_2}{\partial \boldsymbol{\gamma}_v} &= -\mathsf{p}A \times SVU_v \\
\frac{\partial a_2}{\partial \boldsymbol{\gamma}_s} &= -\mathsf{p}A_s \times SVU \\
\frac{\partial a_2}{\partial \boldsymbol{\gamma}_u} &= -\mathsf{p}A \times SVU_u.
\end{aligned}$$

The partial derivatives of a_2/u_1 are:

$$\begin{aligned}
\frac{\partial(a_2/u_1)}{\partial \boldsymbol{\beta}} &= \frac{-a_2 \frac{\partial u_1}{\partial \boldsymbol{\beta}}}{u_1^2} \\
\frac{\partial(a_2/u_1)}{\partial \boldsymbol{\gamma}_v} &= \frac{u_1 \frac{\partial a_2}{\partial \boldsymbol{\gamma}_v} - a_2 \frac{\partial u_1}{\partial \boldsymbol{\gamma}_v}}{u_1^2} \\
\frac{\partial(a_2/u_1)}{\partial \boldsymbol{\gamma}_s} &= \frac{u_1 \frac{\partial a_2}{\partial \boldsymbol{\gamma}_s} - a_2 \frac{\partial u_1}{\partial \boldsymbol{\gamma}_s}}{u_1^2} \\
\frac{\partial(a_2/u_1)}{\partial \boldsymbol{\gamma}_u} &= \frac{u_1 \frac{\partial a_2}{\partial \boldsymbol{\gamma}_u} - a_2 \frac{\partial u_1}{\partial \boldsymbol{\gamma}_u}}{u_1^2}.
\end{aligned}$$

The partial derivatives of u_1/a_2 are:

$$\begin{aligned}
\frac{\partial(u_1/a_2)}{\partial \boldsymbol{\beta}} &= \frac{-a_2 \frac{\partial u_1}{\partial \boldsymbol{\beta}}}{a_2^2} = \frac{-\frac{\partial u_1}{\partial \boldsymbol{\beta}}}{a_2} \\
\frac{\partial(u_1/a_2)}{\partial \boldsymbol{\gamma}_v} &= \frac{a_2 \frac{\partial u_1}{\partial \boldsymbol{\gamma}_v} - u_1 \frac{\partial a_2}{\partial \boldsymbol{\gamma}_v}}{a_2^2} \\
\frac{\partial(u_1/a_2)}{\partial \boldsymbol{\gamma}_s} &= \frac{a_2 \frac{\partial u_1}{\partial \boldsymbol{\gamma}_s} - u_1 \frac{\partial a_2}{\partial \boldsymbol{\gamma}_s}}{a_2^2} \\
\frac{\partial(u_1/a_2)}{\partial \boldsymbol{\gamma}_u} &= \frac{a_2 \frac{\partial u_1}{\partial \boldsymbol{\gamma}_u} - u_1 \frac{\partial a_2}{\partial \boldsymbol{\gamma}_u}}{a_2^2}.
\end{aligned}$$

The partial derivatives of a are:

$$\begin{aligned}
\frac{\partial a}{\partial \boldsymbol{\beta}} &= \mathbf{0} \\
\frac{\partial a}{\partial \boldsymbol{\gamma}_v} &= -\mathsf{p}\alpha \times SVU_v \\
\frac{\partial a}{\partial \boldsymbol{\gamma}_s} &= -\mathsf{p}\alpha_s \times SVU \\
\frac{\partial a}{\partial \boldsymbol{\gamma}_u} &= -\mathsf{p}\alpha \times SVU_u.
\end{aligned}$$

The partial derivatives of $b + a/u_1$ are:

$$\begin{aligned}
\frac{\partial(b + a/u_1)}{\partial \boldsymbol{\beta}} &= \frac{-a\frac{\partial u_1}{\partial \boldsymbol{\beta}}}{u_1^2} \\
\frac{\partial(b + a/u_1)}{\partial \boldsymbol{\gamma}_v} &= \frac{u_1\frac{\partial a}{\partial \boldsymbol{\gamma}_v} - a\frac{\partial u_1}{\partial \boldsymbol{\gamma}_v}}{u_1^2} \\
\frac{\partial(b + a/u_1)}{\partial \boldsymbol{\gamma}_s} &= \frac{u_1\frac{\partial a}{\partial \boldsymbol{\gamma}_s} - a\frac{\partial u_1}{\partial \boldsymbol{\gamma}_s}}{u_1^2} + \mathsf{p}\alpha_s \\
\frac{\partial(b + a/u_1)}{\partial \boldsymbol{\gamma}_u} &= \frac{u_1\frac{\partial a}{\partial \boldsymbol{\gamma}_u} - a\frac{\partial u_1}{\partial \boldsymbol{\gamma}_u}}{u_1^2}.
\end{aligned}$$

The partial derivatives of $b + u_1(1 + b^2)/a$ are:

$$\begin{aligned}
\frac{\partial(b + u_1(1 + b^2)/a)}{\partial \boldsymbol{\beta}} &= \frac{(1 + b^2)\frac{\partial u_1}{\partial \boldsymbol{\beta}}}{a} \\
\frac{\partial(b + u_1(1 + b^2)/a)}{\partial \boldsymbol{\gamma}_v} &= (1 + b^2)\frac{a\frac{\partial u_1}{\partial \boldsymbol{\gamma}_v} - u_1\frac{\partial a}{\partial \boldsymbol{\gamma}_v}}{a^2} \\
\frac{\partial(b + u_1(1 + b^2)/a)}{\partial \boldsymbol{\gamma}_s} &= \frac{a(1 + b^2)\frac{\partial u_1}{\partial \boldsymbol{\gamma}_s} - u1/SVU^2(1/b^2 - 1)\frac{\partial a}{\partial \boldsymbol{\gamma}_s}}{a^2} + \mathsf{p}\alpha_s \\
\frac{\partial(b + u_1(1 + b^2)/a)}{\partial \boldsymbol{\gamma}_u} &= (1 + b^2)\frac{a\frac{\partial u_1}{\partial \boldsymbol{\gamma}_u} - u_1\frac{\partial a}{\partial \boldsymbol{\gamma}_u}}{a^2}.
\end{aligned}$$

The partial derivatives of the first three terms (denoted by $logl_{-\ln \mathcal{A}}$) in (A32) are:

$$\frac{\partial logl_{-\ln \mathcal{A}}}{\partial \boldsymbol{\beta}} = \frac{\mathsf{p}\epsilon_\beta}{\sigma_u}$$

$$\frac{\partial logl_{-\ln\mathcal{A}}}{\partial\boldsymbol{\gamma}_v} = \left(\mathsf{p}\sqrt{\frac{2}{\pi}}A + SVU\right)SVU_v$$
$$\frac{\partial logl_{-\ln\mathcal{A}}}{\partial\boldsymbol{\gamma}_s} = \sqrt{\frac{2}{\pi}}A_s$$
$$\frac{\partial logl_{-\ln\mathcal{A}}}{\partial\boldsymbol{\gamma}_u} = \mathsf{p}\epsilon SU1 + \left(\mathsf{p}\sqrt{\frac{2}{\pi}}A + SVU\right)SVU_u - 0.5\boldsymbol{z}_u.$$

Note that

$$\begin{aligned}\frac{\partial[\Phi(a_2)\Phi(-u_1)]}{\partial\omega} &= \Phi(a_2)\frac{\partial\Phi(-u_1)}{\partial\omega} + \Phi(-u_1)\frac{\partial\Phi(a_2)}{\partial\omega}\\ &= -\Phi(a_2)\phi(-u_1)\frac{\partial u_1}{\partial\omega} + \Phi(-u_1)\phi(a_2)\frac{\partial a_2}{\partial\omega}.\end{aligned}$$

Denoting $\Phi_1 = -\Phi(a_2)\phi(-u_1)$ and $\Phi_2 = \Phi(-u_1)\phi(a_2)$, the partial derivatives of $\Phi(a_2)\Phi(-u_1)$ are:

$$\frac{\partial(\Phi(a_2)\Phi(-u_1))}{\partial\boldsymbol{\beta}} = \Phi_1\frac{\partial u_1}{\partial\boldsymbol{\beta}} + \Phi_2\frac{\partial a_2}{\partial\boldsymbol{\beta}}$$
$$\frac{\partial(\Phi(a_2)\Phi(-u_1))}{\partial\boldsymbol{\gamma}_v} = \Phi_1\frac{\partial u_1}{\partial\boldsymbol{\gamma}_v} + \Phi_2\frac{\partial a_2}{\partial\boldsymbol{\gamma}_v}$$
$$\frac{\partial(\Phi(a_2)\Phi(-u_1))}{\partial\boldsymbol{\gamma}_s} = \Phi_1\frac{\partial u_1}{\partial\boldsymbol{\gamma}_s} + \Phi_2\frac{\partial a_2}{\partial\boldsymbol{\gamma}_s}$$
$$\frac{\partial(\Phi(a_2)\Phi(-u_1))}{\partial\boldsymbol{\gamma}_u} = \Phi_1\frac{\partial u_1}{\partial\boldsymbol{\gamma}_u} + \Phi_2\frac{\partial a_2}{\partial\boldsymbol{\gamma}_u}.$$

Now we can collect the terms recalling (A33) to obtain the gradient in the direction of $\boldsymbol{\beta}$:

$$\frac{\partial logl}{\partial\boldsymbol{\beta}} = \frac{\partial logl_{-\ln\mathcal{A}}}{\partial\boldsymbol{\beta}}$$

$$
+\frac{1}{\mathcal{A}}\left[\begin{array}{l}
-\left[T_1'\left(u_1,\frac{a_2}{u_1}\right)\frac{\partial u_1}{\partial\boldsymbol{\beta}}+T_2'\left(u_1,\frac{a_2}{u_1}\right)\frac{\partial\left(a_2/u_1\right)}{\partial\boldsymbol{\beta}}\right]\\
-\left[T_1'\left(a_2,\frac{u_1}{a_2}\right)\frac{\partial a_2}{\partial\boldsymbol{\beta}}+T_2'\left(a_2,\frac{u_1}{a_2}\right)\frac{\partial\left(u_1/a_2\right)}{\partial\boldsymbol{\beta}}\right]\\
+\left[T_1'\left(u_1,b+\frac{a}{u_1}\right)\frac{\partial u_1}{\partial\boldsymbol{\beta}}+T_2'\left(u_1,b+\frac{a}{u_1}\right)\frac{\partial\left(b+\frac{a}{u_1}\right)}{\partial\boldsymbol{\beta}}\right]\\
+\left[T_1'\left(a_2,b+\frac{u_1(1+b^2)}{a}\right)\frac{\partial a_2}{\partial\boldsymbol{\beta}}\right.\\
\qquad\left.+T_2'\left(a_2,b+\frac{u_1(1+b^2)}{a}\right)\frac{\partial\left(b+\frac{u_1(1+b^2)}{a}\right)}{\partial\boldsymbol{\beta}}\right]\\
+\left[\Phi_1\frac{\partial u_1}{\partial\boldsymbol{\beta}}+\Phi_2\frac{\partial a_2}{\partial\boldsymbol{\beta}}\right]
\end{array}\right]. \tag{A34}
$$

The gradient in the direction of $\boldsymbol{\gamma}_v$, $\boldsymbol{\gamma}_s$, and $\boldsymbol{\gamma}_u$ is obtained by replacing $\boldsymbol{\beta}$ with $\boldsymbol{\gamma}_v$, $\boldsymbol{\gamma}_s$, and $\boldsymbol{\gamma}_u$ in (A34).

A.3 Efficiency Estimation

To obtain observation-specific estimates of inefficiency u, we follow Jondrow et al. (1982) and first obtain the conditional distribution of u given ϵ.

$$
\begin{aligned}
f(u|\epsilon) &= \frac{f(u,\epsilon)}{f(\epsilon)}\\
&= \frac{\frac{2\lambda}{\sigma_v}\exp\left(\mathsf{p}\epsilon_r\lambda+\frac{\lambda^2\sigma_v^2}{2}\right)\phi\left(\frac{u+\mathsf{p}\epsilon_r+\lambda\sigma_v^2}{\sigma_v}\right)\Phi\left(\alpha\frac{\epsilon_r+\mathsf{p}u}{\sigma_v}\right)}{2\lambda\exp\left(\mathsf{p}\epsilon_r\lambda+\frac{\lambda^2\sigma_v^2}{2}\right)\mathcal{A}}\\
&= \frac{\frac{1}{\sigma_v}\phi\left(\frac{u+\mathsf{p}\epsilon_r+\lambda\sigma_v^2}{\sigma_v}\right)\Phi\left(\alpha\frac{\epsilon_r+\mathsf{p}u}{\sigma_v}\right)}{\mathcal{A}}.
\end{aligned} \tag{A35}
$$

We then obtain the point estimator for u by finding the mean value of the conditional distribution in (A35)

$$
E(u|\epsilon)=\int_0^{+\infty}\frac{u\times\frac{1}{\sigma_v}\phi\left(\frac{u+\mathsf{p}\epsilon_r+\lambda\sigma_v^2}{\sigma_v}\right)\Phi\left(\alpha\frac{\epsilon_r+\mathsf{p}u}{\sigma_v}\right)}{\mathcal{A}}du. \tag{A36}
$$

Using the earlier notation on a, b, u_1, denoting $u/\sigma_v + u_1 = u_{rs}$, which implies $du/\sigma_v = du_{rs}$ and noting that $\mathcal{A}$ does not depend on u, the integral in (A36) can be rewritten as

$$E(u|\epsilon) = \frac{1}{\mathcal{A}} \int_0^{+\infty} u \times \frac{1}{\sigma_v}\phi\left(\frac{u}{\sigma_v} + u_1\right)\Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) du \tag{A37}$$

$$= \frac{1}{\mathcal{A}} \int_{u_1}^{+\infty} \left(u_{rs}\sigma_v - u_1\sigma_v\right) \times \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs}. \tag{A38}$$

The integral in (A38) is split into 2 parts

$$\begin{aligned}
E(u|\epsilon) &= \frac{1}{\mathcal{A}}\left(\int_{u_1}^{+\infty} -u_1\sigma_v \times \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs}\right. \\
&\qquad \left. + \int_{u_1}^{+\infty} u_{rs}\sigma_v \times \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs}\right) \\
&= \frac{1}{\mathcal{A}}\left(-u_1\sigma_v \times \int_{u_1}^{+\infty} \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs}\right. \\
&\qquad \left. + \sigma_v \times \int_{u_1}^{+\infty} u_{rs} \times \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs}\right) \\
&= \frac{1}{\mathcal{A}}\left(-u_1\sigma_v \times \mathcal{A}\right. \\
&\qquad \left. + \sigma_v \times \int_{u_1}^{+\infty} u_{rs} \times \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs}\right) \\
&= -u_1\sigma_v + \frac{\sigma_v}{\mathcal{A}} \times \int_{u_1}^{+\infty} u_{rs} \times \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs} \\
&= \sigma_v\left(\left(\frac{1}{\mathcal{A}} \times \int_{u_1}^{+\infty} u_{rs} \times \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs}\right) - u_1\right).
\end{aligned} \tag{A39}$$

Using the result in Owen (1980),

$$\begin{aligned}
\int \boldsymbol{x}\phi\left(\boldsymbol{x}\right)\Phi\left(\boldsymbol{a} + \boldsymbol{b}\boldsymbol{x}\right) d\boldsymbol{x} &= \frac{\boldsymbol{b}}{\sqrt{1+\boldsymbol{b}^2}}\phi\left(\frac{\boldsymbol{a}}{\sqrt{1+\boldsymbol{b}^2}}\right)\Phi\left(\boldsymbol{x}\sqrt{1+\boldsymbol{b}^2} + \frac{\boldsymbol{a}\boldsymbol{b}}{\sqrt{1+\boldsymbol{b}^2}}\right) \\
&\quad - \phi\left(\boldsymbol{x}\right)\Phi\left(\boldsymbol{a} + \boldsymbol{b}\boldsymbol{x}\right),
\end{aligned} \tag{A40}$$

the integral in the first part of equation (A39) can be written as

$$\int_{u_1}^{+\infty} u_{rs} \times \phi\left(u_{rs}\right)\Phi\left(a + bu_{rs}\right) du_{rs}$$

$$
= \left[\frac{b}{\sqrt{1+b^2}}\phi\left(\frac{a}{\sqrt{1+b^2}}\right)\Phi\left(u_{rs}\sqrt{1+b^2}+\frac{ab}{\sqrt{1+b^2}}\right)-\phi\left(u_{rs}\right)\Phi\left(a+bu_{rs}\right)\right]\Bigg|_{u_{rs}=u_1}^{u_{rs}=+\infty}
$$

$$
= \frac{b}{\sqrt{1+b^2}}\phi\left(\frac{a}{\sqrt{1+b^2}}\right)\left\{1-\Phi\left(u_1\sqrt{1+b^2}+\frac{ab}{\sqrt{1+b^2}}\right)\right\}
$$

$$
- \left\{0-\phi\left(u_1\right)\Phi\left(a+bu_1\right)\right\}. \tag{A41}
$$

Therefore,

$$
E(u|\epsilon) = \tag{A42}
$$

$$
= \sigma_v\left(\frac{\text{expression in } (A41)}{\mathcal{A}}-u_1\right)
$$

$$
= \sigma_v\left(\frac{\left[\begin{array}{l}\frac{b}{\sqrt{1+b^2}}\phi\left(\frac{a}{\sqrt{1+b^2}}\right)\Phi\left(-u_1\sqrt{1+b^2}-\frac{ab}{\sqrt{1+b^2}}\right)+\\ \phi\left(u_1\right)\Phi\left(a+bu_1\right)\end{array}\right]}{\left[\begin{array}{l}-T\left(u_1,\frac{a_2}{u_1}\right)-T\left(a_2,\frac{u_1}{a_2}\right)+T\left(u_1,b+\frac{a}{u_1}\right)\\ +T\left(a_2,b+\frac{u_1(1+b^2)}{a}\right)+\Phi\left(a_2\right)\Phi\left(-u_1\right)\end{array}\right]}-u_1\right)
$$

$$
= -\mathsf{p}\epsilon_r-\lambda\sigma_v^2+\sigma_v\frac{\left[\begin{array}{l}\frac{b}{\sqrt{1+b^2}}\phi\left(\frac{a}{\sqrt{1+b^2}}\right)\Phi\left(-u_1\sqrt{1+b^2}-\frac{ab}{\sqrt{1+b^2}}\right)+\\ \phi\left(u_1\right)\Phi\left(a+bu_1\right)\end{array}\right]}{\left[\begin{array}{l}-T\left(u_1,\frac{a_2}{u_1}\right)-T\left(a_2,\frac{u_1}{a_2}\right)+T\left(u_1,b+\frac{a}{u_1}\right)\\ +T\left(a_2,b+\frac{u_1(1+b^2)}{a}\right)+\Phi\left(a_2\right)\Phi\left(-u_1\right)\end{array}\right]}.
$$

B Skew-Normal Truncated Normal Stochastic Frontier Model

Consider the model is described by (A17), but now assume that u is truncated normally distributed on a non-negative space,

$$
u \sim TN(\mu,\sigma_u^2,0,\infty). \tag{B1}
$$

B.1 Derivation of $f(\epsilon)$ and the Log-Likelihood Function

It still holds that $\epsilon = v - \mathsf{p}u$, then $v - \xi = \epsilon + pu - \xi$, where ξ is defined in (A20). The joint density of u and ϵ can be written as

$$f(\epsilon, u) = \Phi\left(\alpha\frac{\epsilon + \mathsf{p}u - \xi}{\sigma_v}\right)\frac{1}{\Phi(\mu/\sigma_u)} \times \frac{2}{\sigma_v\sigma_u}\frac{1}{2\pi}\exp\left(-\frac{1}{2}\left\{\left(\frac{\epsilon + \mathsf{p}u - \xi}{\sigma_v}\right)^2 + \left(\frac{u-\mu}{\sigma_u}\right)^2\right\}\right). \tag{B2}$$

B.1.1 Simplification: Sum of Powers of Exponents

First, consider the sum of exponents in (B2):

$$\left(\frac{\epsilon + \mathsf{p}u - \xi}{\sigma_v}\right)^2 + \left(\frac{u-\mu}{\sigma_u}\right)^2 = \frac{1}{\sigma_u^2\sigma_u^2}\begin{bmatrix} \sigma_u^2(\epsilon^2 + \mathsf{p}^2u^2 + \xi^2 + \\ 2\epsilon\mathsf{p}u - 2\epsilon\xi - 2pu\xi) + \\ \sigma_v^2(u^2 - 2u\mu + \mu^2) \end{bmatrix}. \tag{B3}$$

Now consider only the expression in squared brackets, where we need to complete the square

$$\begin{aligned}&\epsilon^2\sigma_u^2 + u^2\sigma_u^2 + \xi^2\sigma_u^2 + 2\epsilon\mathsf{p}u\sigma_u^2 - 2\epsilon\xi\sigma_u^2 - 2\mathsf{p}u\xi\sigma_u^2 + u^2\sigma_v^2 - 2u\mu\sigma_v^2 + \mu^2\sigma_v^2 + \\ &u^2(\sigma_u^2 + \sigma_v^2) + 2u(\epsilon\mathsf{p}\sigma_u^2 - \mathsf{p}\xi\sigma_u^2 - \mu\sigma_v^2) + \sigma_u^2(\epsilon^2 + \xi^2 - 2\epsilon\xi) + \mu^2\sigma_v^2.\end{aligned} \tag{B4}$$

Denote relocated ϵ by $\epsilon_r = \epsilon - \xi$ and rcall that $\mathsf{p}^2 = 1$, then (B4) can be written as

$$u^2(\sigma_u^2 + \sigma_v^2) + 2u(\mathsf{p}\sigma_u^2(\epsilon - \xi) - \mu\sigma_v^2) + (\epsilon - \xi)^2\sigma_u^2 + \mu^2\sigma_v^2$$

or

$$u^2(\sigma_u^2 + \sigma_v^2) + 2u(\epsilon_r\mathsf{p}\sigma_u^2 - \mu\sigma_v^2) + \epsilon_r^2\sigma_u^2 + \mu^2\sigma_v^2. \tag{B5}$$

B.1.2 Joint Density of u and ϵ

We can now rewrite the expression in (B2)

$$f(\epsilon, u) = \left[\frac{2}{\Phi(\mu/\sigma_u)} \frac{1}{\sigma\sigma_*} \phi\left(\frac{\mu + \mathsf{p}\epsilon_r}{\sigma}\right)\right] \phi\left(\frac{u - \mu_{1,r}}{\sigma_*}\right) \Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right), \tag{B6}$$

where

$$\mu_{1,r} = \frac{\mu\sigma_v^2 - \epsilon_r \mathsf{p}\sigma_u^2}{\sigma^2}, \tag{B7}$$

and

$$\epsilon_r = \epsilon + \sigma_v\sqrt{\frac{2}{\pi}}\frac{\alpha}{\sqrt{1+\alpha^2}} = y - f(\boldsymbol{x};\boldsymbol{\beta}) + \sigma_v\sqrt{\frac{2}{\pi}}\frac{\alpha}{\sqrt{1+\alpha^2}}. \tag{B8}$$

B.1.3 Marginal Density and Log-Likelihood

The marginal density of ϵ is obtained by integrating u out (B6) over the domain of u

$$\int_0^\infty f(\epsilon, u)du = \frac{2}{\Phi(\mu/\sigma_u)} \frac{1}{\sigma\sigma_*} \phi\left(\frac{\mu + \mathsf{p}\epsilon_r}{\sigma}\right) \int_0^\infty \phi\left(\frac{u - \mu_{1,r}}{\sigma_*}\right) \Phi\left(\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right) du. \tag{B9}$$

Denote the relocated and scaled variant of u

$$u_{rs} = \frac{u - \mu_{1,r}}{\sigma_*}. \tag{B10}$$

Then $u = \mu_{1,r} + u_{rs}\sigma_*$ and

$$\alpha\frac{\epsilon_r + \mathsf{p}u}{\sigma_v} = \alpha\frac{\epsilon_r + \mathsf{p}(\mu_{1,r} + u_{rs}\sigma_*)}{\sigma_v} = \alpha\frac{\epsilon_r + \mathsf{p}\mu_{1,r}}{\sigma_v} + \alpha\frac{\mathsf{p}\sigma_*}{\sigma_v}u_{rs} = a + bu_{rs}. \tag{B11}$$

Then the marginal density of ϵ given in (B9) is

$$\begin{aligned} f(\epsilon) &= \frac{2}{\Phi(\mu/\sigma_u)} \frac{1}{\sigma\sigma_*} \phi\left(\frac{\mu + \mathsf{p}\epsilon_r}{\sigma}\right) \int_{-\frac{\mu_1}{\sigma_*}}^\infty \phi\left(u_{rs}\right) \Phi\left(a + bu_{rs}\right) \sigma_* du_{rs} && \text{(B12)} \\ &= \frac{2}{\Phi(\mu/\sigma_u)} \frac{1}{\sigma} \phi\left(\frac{\mu + \mathsf{p}\epsilon_r}{\sigma}\right) \int_{-\frac{\mu_1}{\sigma_*}}^\infty \phi\left(u_{rs}\right) \Phi\left(a + bu_{rs}\right) du_{rs}. && \text{(B13)} \end{aligned}$$

Denoting $u_1 = -\mu_1/\sigma_*$ and using the result in (A11)[24]

$$\begin{aligned}
\mathcal{B} &= \int_{u_1}^{\infty} \phi\left(u_{rs}\right) \Phi\left(a + b u_{rs}\right) du_{rs} \\
&= -T\left(u_1, \frac{a_2}{u_1}\right) - T\left(a_2, \frac{u_1}{a_2}\right) + T\left(u_1, b + \frac{a}{u_1}\right) \\
&\quad + T\left(a_2, b + \frac{u_1(1+b^2)}{a}\right) + \Phi\left(a_2\right) \Phi\left(-u_1\right),
\end{aligned} \tag{B14}$$

where $a = \alpha(\epsilon_r + \mathsf{p}\mu_{1,r})/\sigma_v$ and $b = \alpha \mathsf{p} \sigma_*/\sigma_v$. Then

$$f(\epsilon) = \frac{2}{\Phi(\mu/\sigma_u)} \frac{1}{\sigma} \phi\left(\frac{\mu + \mathsf{p}\epsilon_r}{\sigma}\right) \times \mathcal{B}. \tag{B15}$$

The log-likelihood based on (B15) is therefore

$$\ln(2) - \ln \Phi\left(\frac{\mu}{\sigma_u}\right) - \ln \sigma + \ln \phi\left(\frac{\mu + \mathsf{p}\epsilon_r}{\sigma}\right) + \ln \mathcal{B}. \tag{B16}$$

The gradients in case of the truncated normal u are derived analogously to the case of exponential u, shown in A.2.4.

B.2 Efficiency Estimation

To obtain observation-specific estimates of inefficiency u, we follow Jondrow et al. (1982) and first obtain the conditional distribution of u given ϵ

$$\begin{aligned}
f(u|\epsilon) &= \frac{f(\epsilon, x)}{f(\epsilon)} \\
&= \frac{\left[\frac{2}{\Phi(\mu/\sigma_u)} \frac{1}{\sigma\sigma_*} \phi\left(\frac{\mu + \mathsf{p}\epsilon_r}{\sigma}\right)\right] \phi\left(\frac{u - \mu_{1,r}}{\sigma_*}\right) \Phi\left(\alpha \frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right)}{\frac{2}{\Phi(\mu/\sigma_u)} \frac{1}{\sigma} \phi\left(\frac{\mu + \mathsf{p}\epsilon_r}{\sigma}\right) \times \mathcal{B}} \\
&= \frac{\frac{1}{\sigma_*} \phi\left(\frac{u - \mu_{1,r}}{\sigma_*}\right) \Phi\left(\alpha \frac{\epsilon_r + \mathsf{p}u}{\sigma_v}\right)}{\mathcal{B}.}
\end{aligned} \tag{B17}$$

[24] The result (A16) can also be used.

We then obtain the point estimator for u by finding the mean value of the conditional distribution in (B17)

$$E(u|\epsilon) = \int_0^{+\infty} \frac{u \times \frac{1}{\sigma_*}\phi\left(\frac{u-\mu_{1,r}}{\sigma_*}\right)\Phi\left(\alpha\frac{\epsilon_r+\mathsf{p}u}{\sigma_v}\right)}{\mathcal{B}} du. \tag{B18}$$

Since $\mathcal{B}$ do not depend on u,

$$E(u|\epsilon) = \frac{1}{\mathcal{B}\sigma_*}\int_0^{+\infty} u \times \phi\left(\frac{u-\mu_{1,r}}{\sigma_*}\right)\Phi\left(\alpha\frac{\epsilon_r+\mathsf{p}u}{\sigma_v}\right) du. \tag{B19}$$

To make use of the (A40), recall the notation (B10) and (B11). Then the integral in (B19) becomes

$$\begin{aligned} E(u|\epsilon) &= \frac{1}{\mathcal{B}\sigma_*}\int_{u_1}^{+\infty} (\mu_{1,r}+u_{rs}\sigma_*) \times \phi(u_{rs})\Phi(a+bu_{rs})\sigma_* du_{rs} \\ &= \frac{1}{\mathcal{B}}\int_{u_1}^{+\infty} (\mu_{1,r}+u_{rs}\sigma_*) \times \phi(u_{rs})\Phi(a+bu_{rs}) du_{rs}, \end{aligned} \tag{B20}$$

where $u_1 = -\mu_{1,r}/\sigma_*$, as before by setting $u = 0$ into (B10). The integral in (B20) is split into 2 parts

$$\begin{aligned} E(u|\epsilon) &= \frac{1}{\mathcal{B}}\Bigg(\int_{u_1}^{+\infty} \mu_{1,r} \times \phi(u_{rs})\Phi(a+bu_{rs}) du_{rs} \\ &\qquad + \int_{u_1}^{+\infty} u_{rs}\sigma_* \times \phi(u_{rs})\Phi(a+bu_{rs}) du_{rs}\Bigg) \\ &= \frac{1}{\mathcal{B}}\Bigg(\mu_{1,r} \times \int_{u_1}^{+\infty} \phi(u_{rs})\Phi(a+bu_{rs}) du_{rs} \\ &\qquad + \sigma_* \times \int_{u_1}^{+\infty} u_{rs} \times \phi(u_{rs})\Phi(a+bu_{rs}) du_{rs}\Bigg) \\ &= \frac{1}{\mathcal{B}}\Bigg(\mu_{1,r} \times \mathcal{B} \\ &\qquad + \sigma_* \times \int_{u_1}^{+\infty} u_{rs} \times \phi(u_{rs})\Phi(a+bu_{rs}) du_{rs}\Bigg) \\ &= \mu_{1,r} + \frac{\sigma_*}{\mathcal{B}} \times \int_{u_1}^{+\infty} u_{rs} \times \phi(u_{rs})\Phi(a+bu_{rs}) du_{rs}. \end{aligned} \tag{B21}$$

Using (A40),

$$
\begin{aligned}
&\int_{u_1}^{+\infty} u_{rs} \times \phi\left(u_{rs}\right) \Phi\left(a+b u_{rs}\right) d u_{rs} \\
&\quad=\left.\left[\frac{b}{\sqrt{1+b^2}} \phi\left(\frac{a}{\sqrt{1+b^2}}\right) \Phi\left(u_{rs} \sqrt{1+b^2}+\frac{a b}{\sqrt{1+b^2}}\right)-\phi\left(u_{rs}\right) \Phi\left(a+b u_{rs}\right)\right]\right|_{u_{rs}=u_1} ^{u_{rs}=+\infty} \\
&\quad=\frac{b}{\sqrt{1+b^2}} \phi\left(\frac{a}{\sqrt{1+b^2}}\right)\left\{1-\Phi\left(u_1 \sqrt{1+b^2}+\frac{a b}{\sqrt{1+b^2}}\right)\right\} \\
&\quad-\left\{0-\phi\left(u_1\right) \Phi\left(a+b u_1\right)\right\} .
\end{aligned} \tag{B22}
$$

Therefore,

$$
\begin{aligned}
E(u|\epsilon) &= \mu_{1,r} + \frac{\sigma_*}{\mathcal{B}} \times (\text{expression in } (B22)) \\
&= \mu_{1,r} + \sigma_* \frac{\left[\begin{array}{l} \frac{b}{\sqrt{1+b^2}} \phi\left(\frac{a}{\sqrt{1+b^2}}\right) \Phi\left(-u_1 \sqrt{1+b^2}-\frac{a b}{\sqrt{1+b^2}}\right)+ \\ \phi\left(u_1\right) \Phi\left(a+b u_1\right) \end{array}\right]}{\left[\begin{array}{l} -T\left(u_1, \frac{a_2}{u_1}\right)-T\left(a_2, \frac{u_1}{a_2}\right)+T\left(u_1, \frac{a}{u_1}+b\right)+ \\ T\left(a_2, \frac{u_1 a}{a_2^2}+b\right)+\Phi\left(a_2\right) \Phi\left(-u_1\right) \end{array}\right]} .
\end{aligned} \tag{B23}
$$

C Simulation study and likelihood profile analysis

C.1 SN regression

Table C1: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$. Bias and MSE, $\sigma_v = 0.5$, and three values of α, 0.5, 1, and 2

N	β_0		β_1		σ_v		α	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$\sigma_v = 0.5$, $\alpha = 0.5$								
250	0.0011	0.0023	−0.0009	0.0029	0.0466	0.0024	−0.0607	0.6667
500	−0.0014	0.0012	0.0017	0.0014	0.0340	0.0013	0.0152	0.3450
1000	−0.0003	0.0005	0.0004	0.0007	0.0206	0.0007	−0.0409	0.1960
2000	0.0014	0.0003	−0.0007	0.0003	0.0131	0.0004	−0.0173	0.0916
$\sigma_v = 0.5$, $\alpha = 1$								
250	−0.0006	0.0018	−0.0008	0.0022	0.0008	0.0014	0.0005	0.1417
500	−0.0003	0.0009	0.0006	0.0010	−0.0003	0.0009	0.0028	0.0637
1000	0.0003	0.0004	−0.0005	0.0005	−0.0029	0.0005	−0.0110	0.0348
2000	0.0006	0.0002	−0.0006	0.0003	−0.0014	0.0003	−0.0110	0.0162
$\sigma_v = 0.5$, $\alpha = 2$								
250	−0.0006	0.0012	0.0012	0.0016	−0.0043	0.0008	−0.0044	0.1289
500	0.0001	0.0007	−0.0004	0.0007	−0.0024	0.0004	−0.0002	0.0567
1000	0.0002	0.0003	−0.0001	0.0003	−0.0004	0.0002	−0.0029	0.0274
2000	−0.0000	0.0001	−0.0004	0.0002	−0.0009	0.0001	−0.0086	0.0148

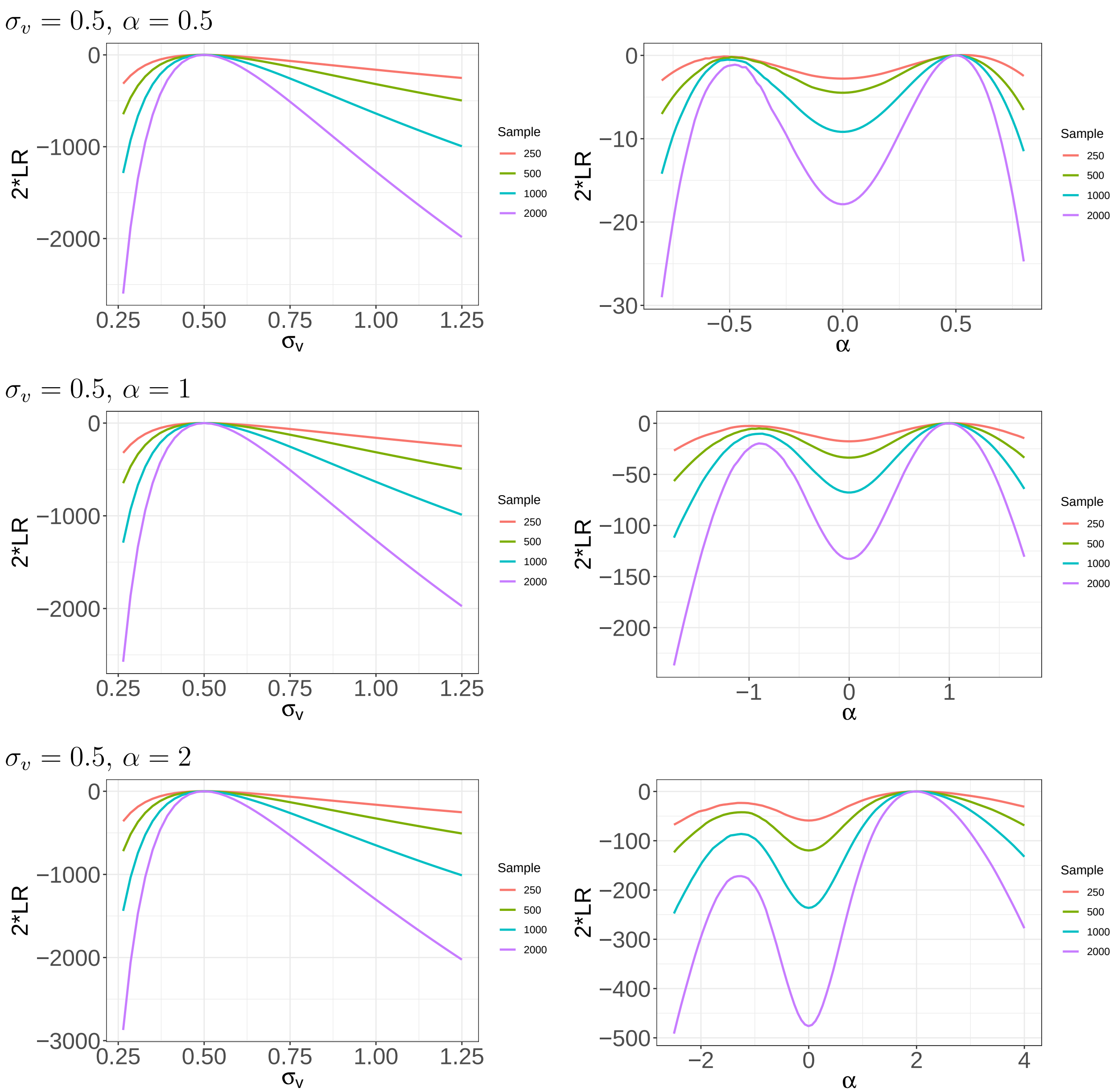

Figure C1: Median over 99 replications. Double difference between likelihood calculated at true values and likelihood calculated varying one parameter.

Table C2: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$. Bias and MSE, $\sigma_v = 1$, and three values of α, 0.5, 1, and 2

N	β_0		β_1		σ_v		α	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$\sigma_v = 1$, $\alpha = 0.5$								
250	0.0023	0.0093	−0.0019	0.0114	0.0930	0.0094	−0.0619	0.6683
500	−0.0027	0.0047	0.0034	0.0054	0.0681	0.0054	0.0146	0.3466
1000	−0.0006	0.0021	0.0009	0.0026	0.0411	0.0026	−0.0401	0.1961
2000	0.0029	0.0012	−0.0015	0.0014	0.0263	0.0016	−0.0176	0.0914
$\sigma_v = 1$, $\alpha = 1$								
250	−0.0012	0.0071	−0.0015	0.0090	0.0015	0.0057	0.0001	0.1417
500	−0.0007	0.0038	0.0012	0.0039	−0.0005	0.0036	0.0024	0.0639
1000	0.0005	0.0017	−0.0009	0.0021	−0.0059	0.0022	−0.0111	0.0349
2000	0.0012	0.0009	−0.0011	0.0011	−0.0029	0.0011	−0.0108	0.0162
$\sigma_v = 1$, $\alpha = 2$								
250	−0.0011	0.0049	0.0023	0.0062	−0.0087	0.0034	−0.0044	0.1295
500	0.0001	0.0027	−0.0008	0.0028	−0.0047	0.0016	−0.0000	0.0567
1000	0.0004	0.0011	−0.0003	0.0013	−0.0008	0.0007	−0.0033	0.0274
2000	−0.0001	0.0006	−0.0008	0.0007	−0.0019	0.0004	−0.0086	0.0148

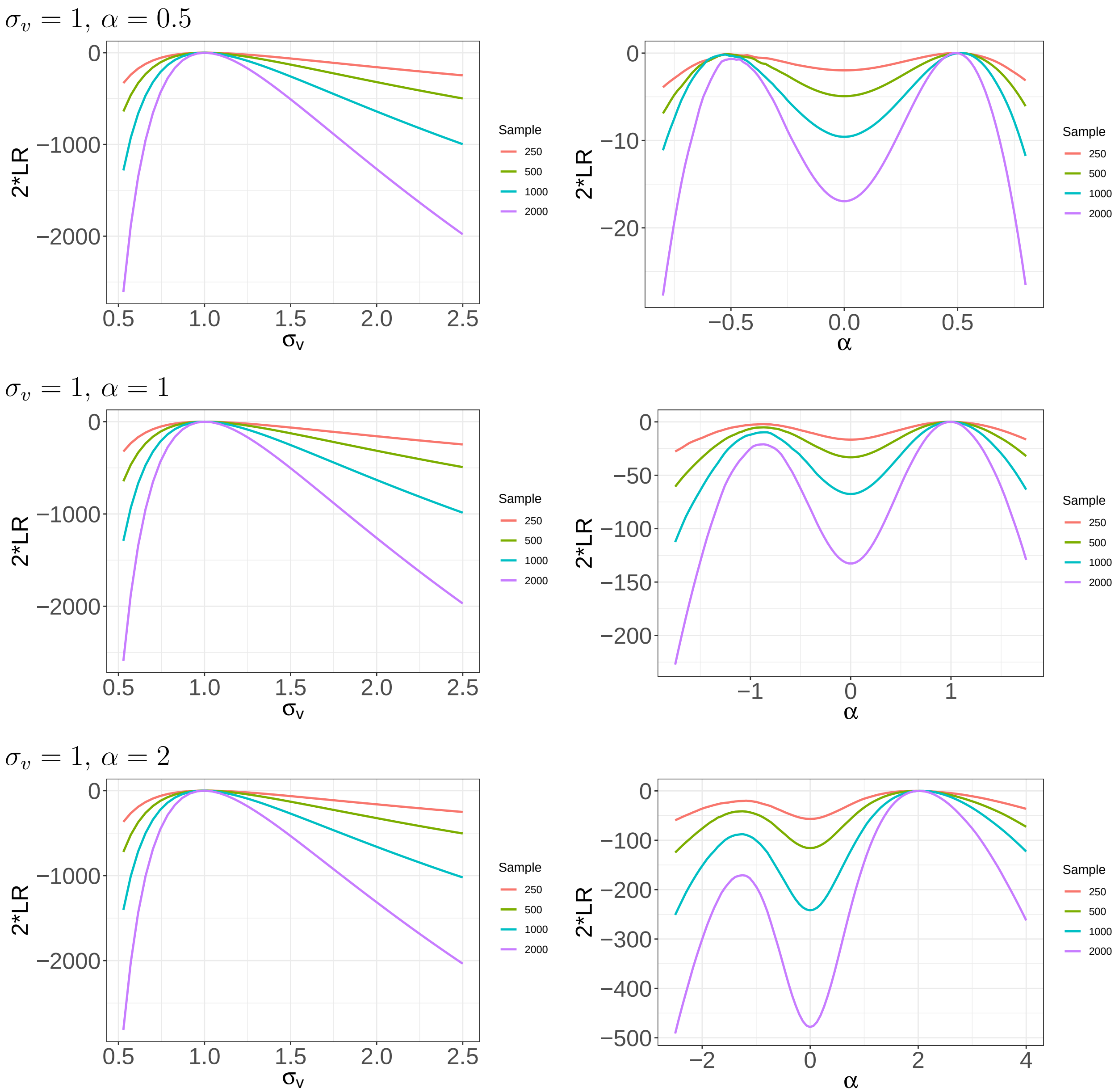

Figure C2: Median over 99 replications. Double difference between likelihood calculated at true values and likelihood calculated varying one parameter.

C.2 SN-Exp regression

Table C3: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 0.5$, $\sigma_u = 0.5$, and three values of α, 0.5, 1, and 2

N	β_0		β_1		σ_u		σ_v		α	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$\sigma_v = 0.5$, $\alpha = 0.5$, $\sigma_u = 0.5$										
250	−0.0252	0.0083	0.0012	0.0052	−0.0214	0.0039	0.1066	0.0115	−0.9240	2.3132
500	−0.0155	0.0041	−0.0028	0.0026	−0.0129	0.0018	0.0876	0.0077	−0.5232	1.6280
1000	−0.0084	0.0019	0.0004	0.0012	−0.0047	0.0010	0.0626	0.0040	−0.2449	1.0230
2000	−0.0029	0.0009	−0.0005	0.0006	−0.0040	0.0005	0.0478	0.0025	−0.5033	0.6415
$\sigma_v = 0.5$, $\alpha = 1$, $\sigma_u = 0.5$										
250	−0.0257	0.0062	−0.0015	0.0044	−0.0190	0.0027	0.0506	0.0047	−0.3371	2.0782
500	−0.0086	0.0026	0.0007	0.0019	−0.0050	0.0013	0.0307	0.0024	−0.0945	1.0007
1000	−0.0024	0.0014	−0.0012	0.0010	−0.0025	0.0007	0.0123	0.0012	−0.0613	0.2820
2000	−0.0029	0.0007	−0.0002	0.0005	−0.0029	0.0003	0.0041	0.0009	−0.0530	0.1628
$\sigma_v = 0.5$, $\alpha = 2$, $\sigma_u = 0.5$										
250	−0.0238	0.0045	0.0041	0.0033	−0.0207	0.0020	−0.0053	0.0024	−0.4419	1.7232
500	−0.0101	0.0020	0.0009	0.0017	−0.0103	0.0009	−0.0052	0.0013	−0.1619	0.8080
1000	−0.0056	0.0011	−0.0001	0.0009	−0.0052	0.0004	−0.0047	0.0007	−0.1032	0.4184
2000	−0.0038	0.0006	−0.0002	0.0005	−0.0046	0.0003	−0.0037	0.0004	−0.0893	0.2173

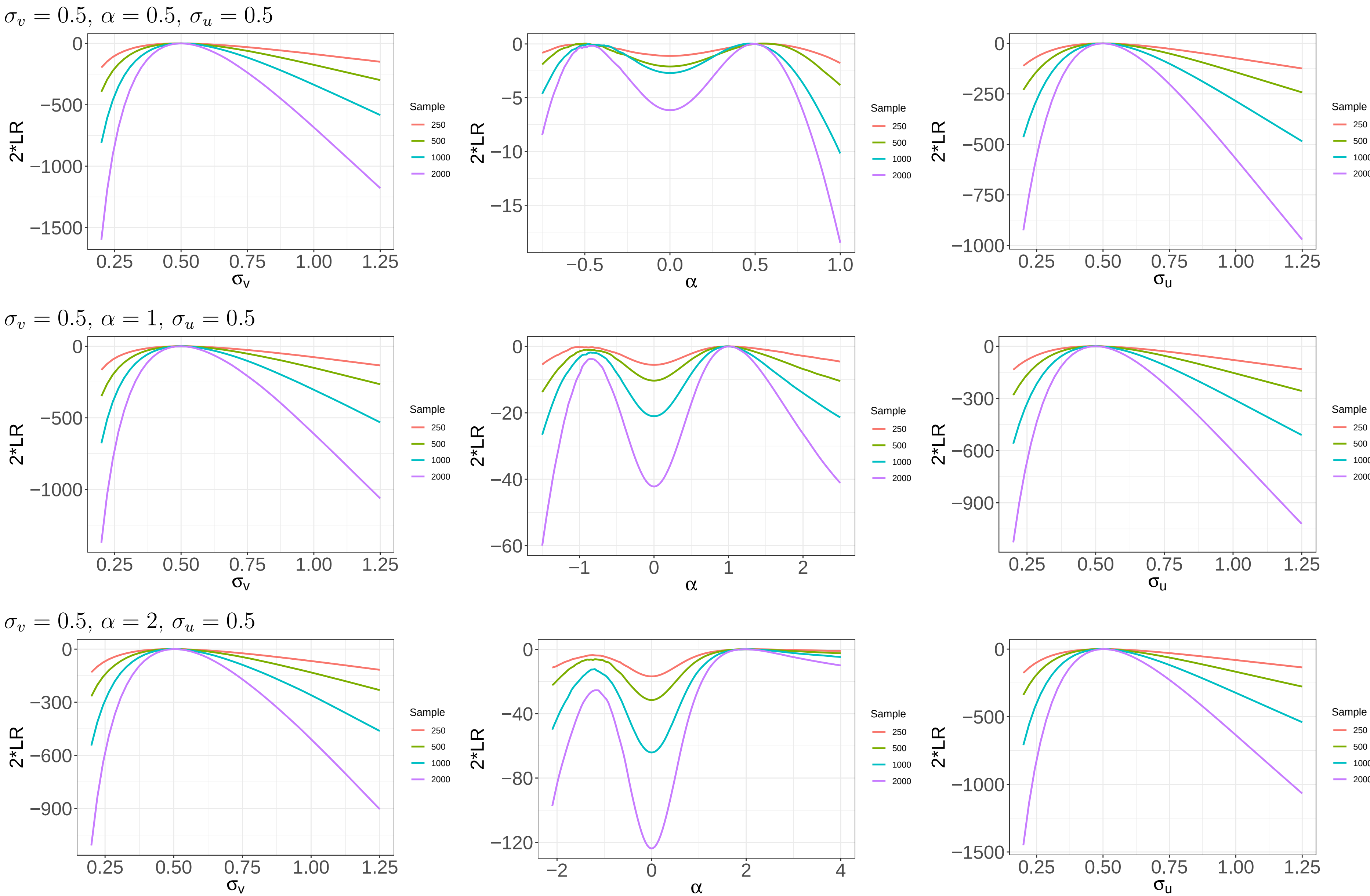

Figure C3: Median over 99 replications. Double difference between likelihood calculated at true values and likelihood calculated varying one parameter.

Table C4: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 0.5$, $\sigma_u = 1$, and three values of α, 0.5, 1, and 2

N	β_0		β_1		σ_u		σ_v		α	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$\sigma_v = 0.5$, $\alpha = 0.5$, $\sigma_u = 1$										
250	−0.0275	0.0117	0.0034	0.0087	−0.0221	0.0056	0.1375	0.0189	−1.2614	3.6717
500	−0.0117	0.0049	0.0009	0.0043	−0.0116	0.0025	0.1081	0.0117	−0.9804	2.4087
1000	−0.0042	0.0023	0.0003	0.0020	−0.0057	0.0013	0.0811	0.0066	−0.7277	1.6318
2000	−0.0005	0.0012	−0.0018	0.0010	−0.0022	0.0007	0.0610	0.0038	−0.2050	0.9351
$\sigma_v = 0.5$, $\alpha = 1$, $\sigma_u = 1$										
250	−0.0272	0.0087	0.0021	0.0076	−0.0253	0.0049	0.0701	0.0079	−1.3022	4.8366
500	−0.0128	0.0040	0.0039	0.0037	−0.0110	0.0023	0.0472	0.0041	−0.3316	2.7463
1000	−0.0056	0.0019	0.0027	0.0016	−0.0039	0.0012	0.0293	0.0021	−0.0842	0.9196
2000	−0.0015	0.0010	−0.0007	0.0009	−0.0014	0.0006	0.0144	0.0011	−0.0029	0.2628
$\sigma_v = 0.5$, $\alpha = 2$, $\sigma_u = 1$										
250	−0.0240	0.0069	−0.0063	0.0060	−0.0274	0.0044	0.0064	0.0044	−0.9107	4.5340
500	−0.0158	0.0034	0.0034	0.0033	−0.0153	0.0018	−0.0055	0.0024	−0.4427	1.6533
1000	−0.0055	0.0015	0.0024	0.0015	−0.0052	0.0009	−0.0077	0.0011	−0.2135	0.8287
2000	−0.0046	0.0008	0.0005	0.0008	−0.0027	0.0004	−0.0038	0.0006	−0.0468	0.4180

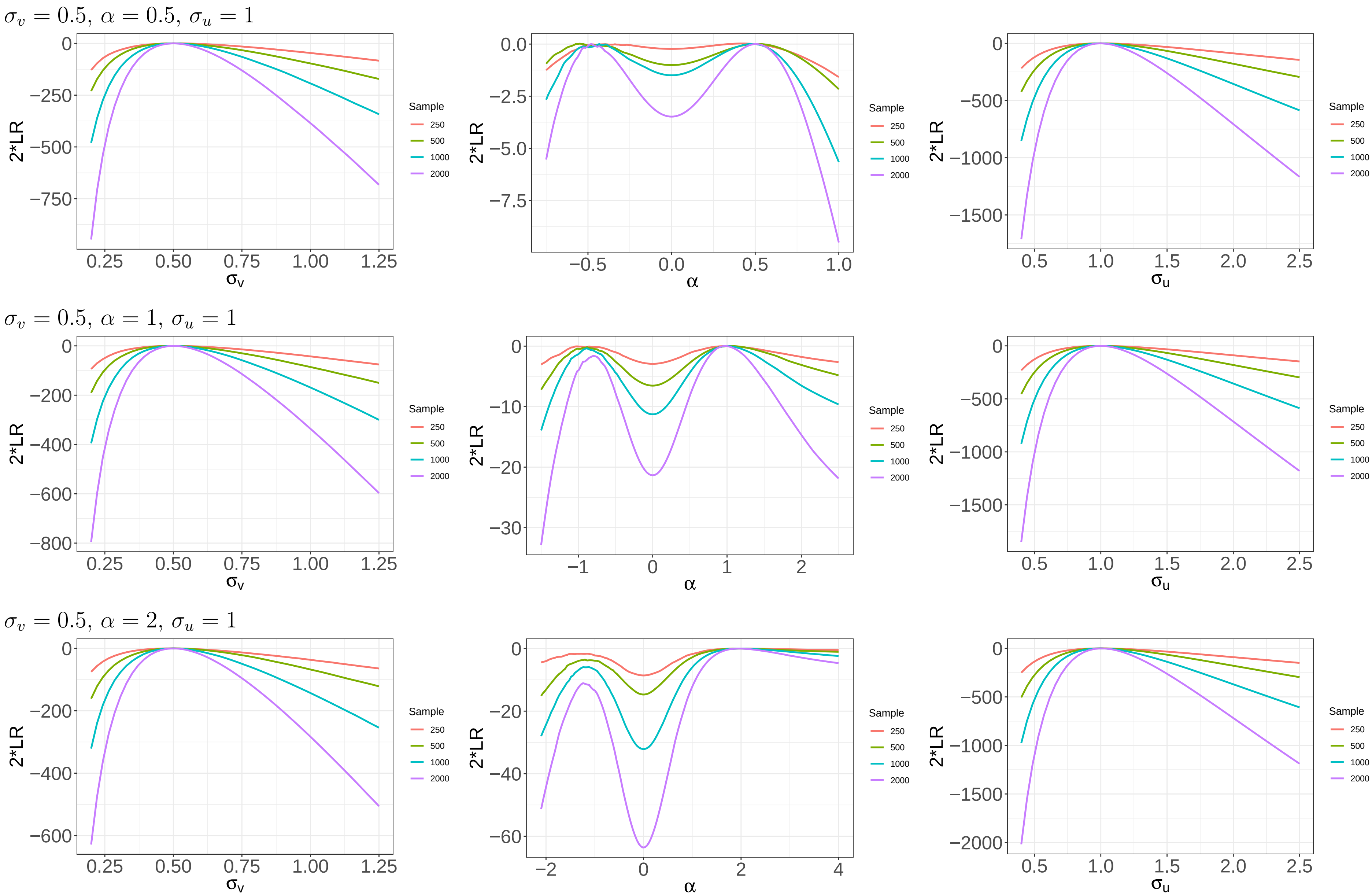

Figure C4: Median over 99 replications. Double difference between likelihood calculated at true values and likelihood calculated varying one parameter.

Table C5: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 1$, $\sigma_u = 0.5$, and three values of α, 0.5, 1, and 2

N	β_0		β_1		σ_u		σ_v		α	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$\sigma_v = 1$, $\alpha = 0.5$, $\sigma_u = 0.5$										
250	−0.0211	0.0292	−0.0021	0.0152	0.0010	0.0185	0.1120	0.0162	−0.4927	0.9480
500	0.0004	0.0144	−0.0072	0.0065	0.0110	0.0084	0.0771	0.0090	−0.4190	0.4520
1000	−0.0102	0.0081	0.0004	0.0035	−0.0082	0.0047	0.0688	0.0063	−0.3222	0.2884
2000	−0.0111	0.0042	−0.0026	0.0019	−0.0078	0.0024	0.0539	0.0044	−0.0981	0.2498
$\sigma_v = 1$, $\alpha = 1$, $\sigma_u = 0.5$										
250	−0.0263	0.0200	0.0101	0.0128	0.0043	0.0115	0.0190	0.0100	−0.2292	0.9933
500	−0.0060	0.0107	0.0015	0.0060	0.0007	0.0050	0.0018	0.0056	−0.1091	0.4430
1000	−0.0091	0.0058	0.0001	0.0029	−0.0078	0.0030	0.0003	0.0038	−0.0616	0.1496
2000	−0.0053	0.0026	−0.0006	0.0014	−0.0065	0.0015	0.0026	0.0020	−0.0164	0.0731
$\sigma_v = 1$, $\alpha = 2$, $\sigma_u = 0.5$										
250	−0.0103	0.0122	−0.0039	0.0098	0.0090	0.0047	−0.0505	0.0078	−0.3012	1.0124
500	−0.0070	0.0070	0.0009	0.0046	0.0039	0.0021	−0.0243	0.0033	−0.1340	0.4523
1000	0.0033	0.0033	−0.0011	0.0025	0.0051	0.0013	−0.0123	0.0013	−0.0337	0.2051
2000	0.0002	0.0015	−0.0023	0.0012	0.0015	0.0006	−0.0050	0.0007	−0.0077	0.0955

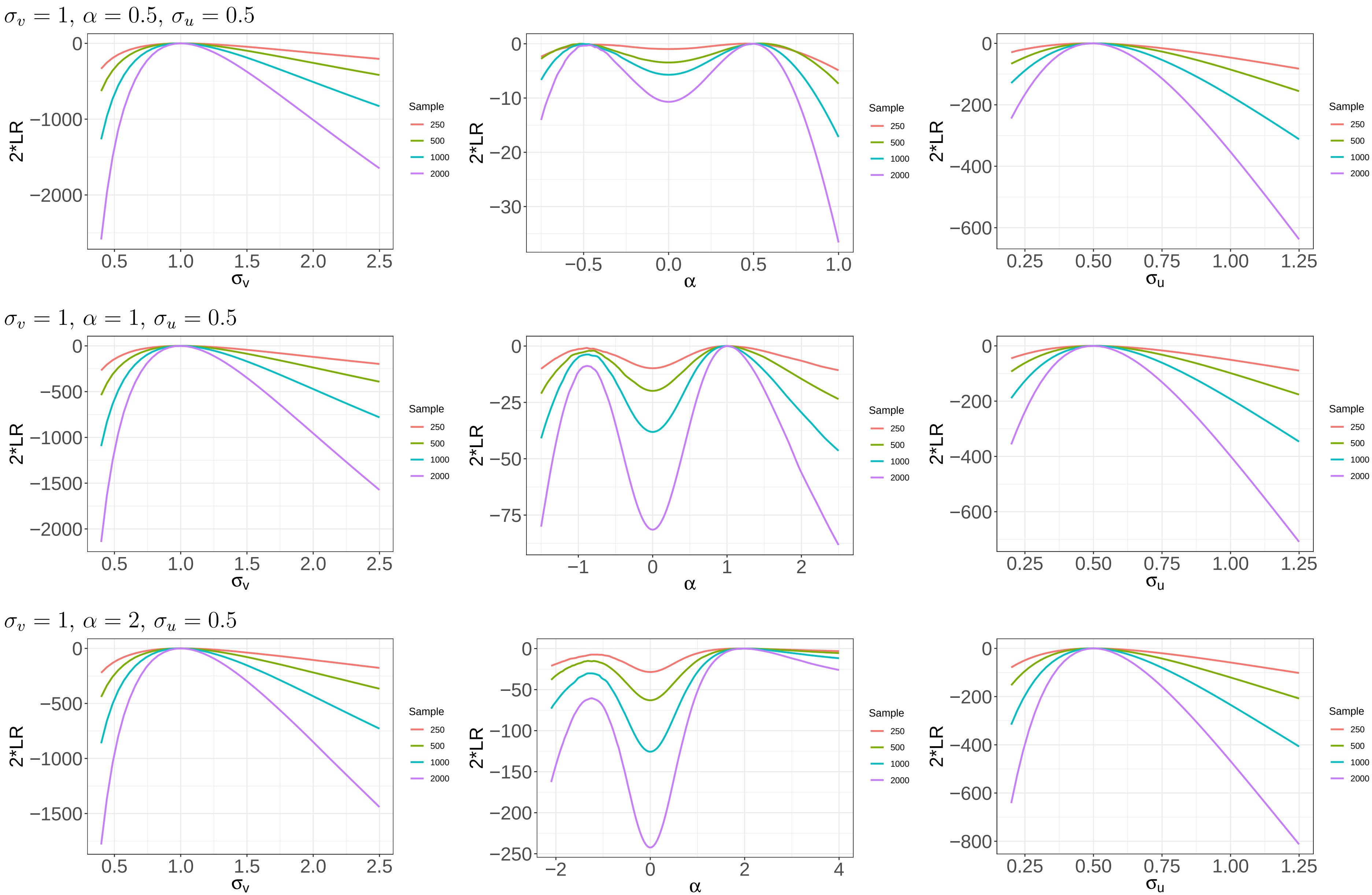

Figure C5: Median over 99 replications. Double difference between likelihood calculated at true values and likelihood calculated varying one parameter.

Table C6: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 1$, $\sigma_u = 1$, and three values of α, 0.5, 1, and 2

N	β_0		β_1		σ_u		σ_v		α	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$\sigma_v = 1$, $\alpha = 0.5$, $\sigma_u = 1$										
250	−0.0639	0.0348	0.0034	0.0189	−0.0410	0.0157	0.2055	0.0430	−0.6357	2.2089
500	−0.0199	0.0152	0.0006	0.0102	−0.0216	0.0069	0.1611	0.0262	−0.4993	1.5535
1000	−0.0205	0.0072	0.0020	0.0049	−0.0136	0.0034	0.1260	0.0163	−0.5048	1.0149
2000	−0.0106	0.0040	−0.0012	0.0027	−0.0143	0.0018	0.0772	0.0081	−0.5175	0.5326
$\sigma_v = 1$, $\alpha = 1$, $\sigma_u = 1$										
250	−0.0479	0.0263	0.0128	0.0168	−0.0347	0.0109	0.0935	0.0175	−0.4009	2.5465
500	−0.0189	0.0116	−0.0010	0.0081	−0.0168	0.0052	0.0538	0.0091	−0.1998	1.0067
1000	−0.0170	0.0054	0.0046	0.0042	−0.0085	0.0027	0.0286	0.0055	−0.0846	0.3813
2000	−0.0094	0.0030	0.0000	0.0024	−0.0105	0.0016	−0.0168	0.0057	−0.2408	0.4031
$\sigma_v = 1$, $\alpha = 2$, $\sigma_u = 1$										
250	−0.0516	0.0187	0.0106	0.0161	−0.0388	0.0081	−0.0128	0.0104	−0.4755	1.6486
500	−0.0246	0.0086	0.0050	0.0071	−0.0218	0.0037	−0.0107	0.0051	−0.2244	0.9788
1000	−0.0164	0.0045	0.0014	0.0036	−0.0135	0.0019	−0.0159	0.0036	−0.1565	0.5443
2000	−0.0144	0.0023	0.0016	0.0018	−0.0116	0.0012	−0.0168	0.0020	−0.1553	0.3266

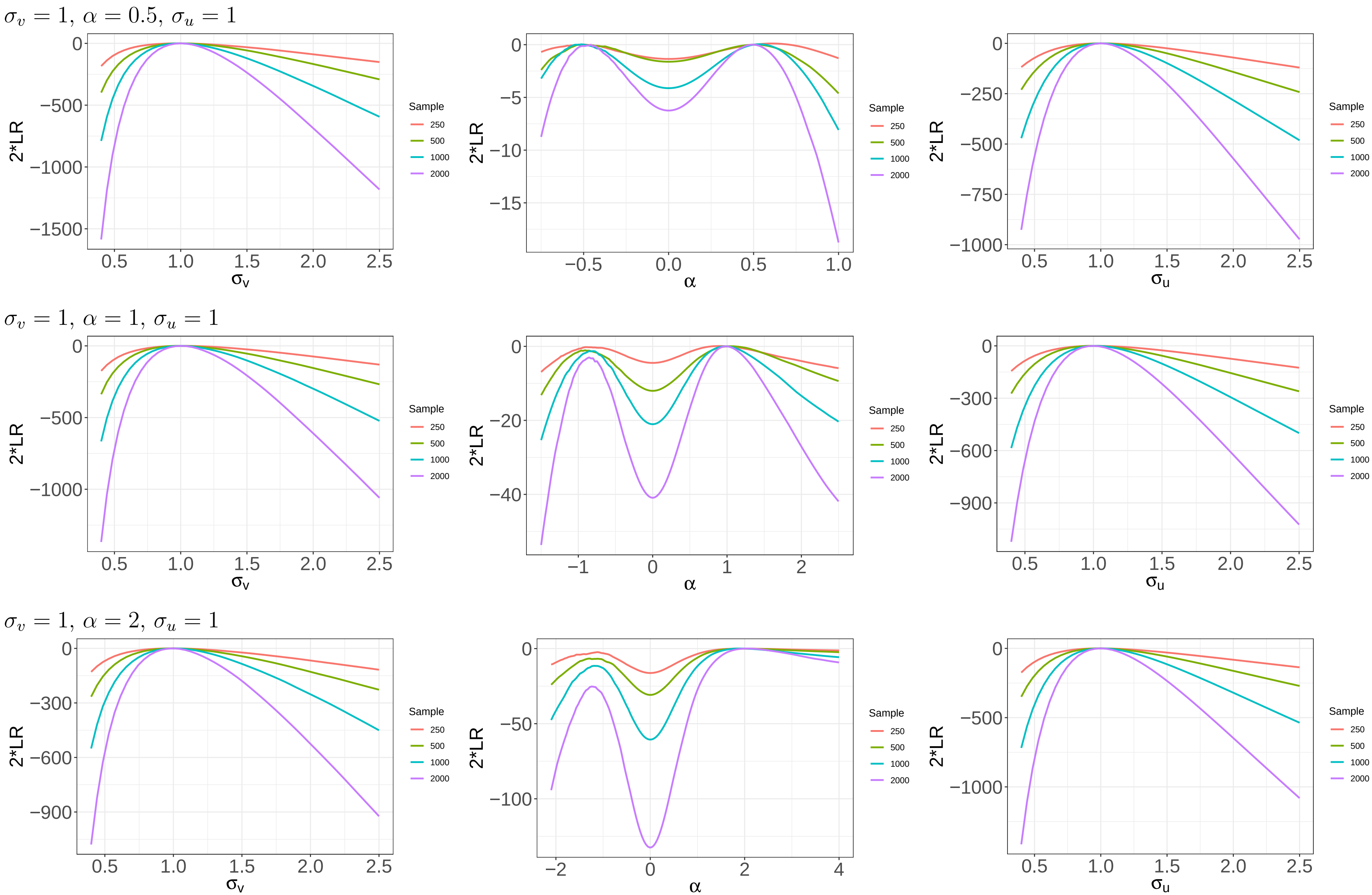

Figure C6: Median over 99 replications. Double difference between likelihood calculated at true values and likelihood calculated varying one parameter.

D Simulation Study on the Comparison of the New Model and the N-Exp SF Model

D.1 The True Noise Originates from the Skew-Normal Distribution

Table D1: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 0.5$, $\sigma_u = 0.5$, and three values of α, 0.5, 1, and 2. Estimated by SN-Exp

N	β_0		β_1		σ_u		σ_v		α		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
$\sigma_v = 0.5$, $\alpha = 0.5$, $\sigma_u = 0.5$														
250	−0.0357	0.0081	0.0042	0.0051	−0.0229	0.0037	0.1029	0.0110	−0.5167	2.1292	−0.0193	0.0153	0.4560	0.4442
500	−0.0131	0.0039	0.0011	0.0026	−0.0111	0.0019	0.0840	0.0072	−0.5161	1.5634	−0.0261	0.0148	0.4420	0.4462
1000	−0.0052	0.0019	−0.0019	0.0013	−0.0063	0.0009	0.0621	0.0040	−0.5133	1.0004	−0.0279	0.0146	0.4380	0.4454
2000	−0.0049	0.0010	−0.0009	0.0007	−0.0049	0.0005	0.0405	0.0021	−0.5071	0.5795	−0.0287	0.0144	0.4360	0.4457
$\sigma_v = 0.5$, $\alpha = 1$, $\sigma_u = 0.5$														
250	−0.0245	0.0064	0.0023	0.0045	−0.0218	0.0028	0.0442	0.0047	−0.9871	2.4043	−0.0182	0.0131	0.4560	0.4796
500	−0.0060	0.0029	−0.0002	0.0021	−0.0077	0.0013	0.0250	0.0024	−0.1553	1.0070	−0.0230	0.0126	0.4440	0.4803
1000	−0.0060	0.0016	−0.0008	0.0011	−0.0048	0.0007	0.0126	0.0015	−0.1250	0.5910	−0.0249	0.0125	0.4390	0.4804
2000	−0.0047	0.0007	−0.0004	0.0006	−0.0052	0.0003	−0.0001	0.0009	−0.1045	0.1885	−0.0250	0.0124	0.4385	0.4808
$\sigma_v = 0.5$, $\alpha = 2$, $\sigma_u = 0.5$														
250	−0.0214	0.0041	−0.0031	0.0034	−0.0159	0.0016	−0.0078	0.0022	−0.3942	1.6090	−0.0169	0.0100	0.4520	0.5310
500	−0.0094	0.0020	−0.0007	0.0019	−0.0075	0.0008	−0.0056	0.0012	−0.2114	0.7209	−0.0200	0.0099	0.4420	0.5304
1000	−0.0051	0.0012	−0.0005	0.0010	−0.0038	0.0004	−0.0035	0.0006	−0.0744	0.4009	−0.0228	0.0098	0.4340	0.5300
2000	−0.0018	0.0006	−0.0020	0.0005	−0.0027	0.0002	−0.0039	0.0004	−0.0589	0.2065	−0.0228	0.0097	0.4335	0.5294

Table D2: The DGP and setup are as in Table D1. Estimated by N-Exp

N	β_0		β_1		σ_u		σ_v		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
$\sigma_v = 0.5$, $\alpha = 0.5$, $\sigma_u = 0.5$												
250	−0.0164	0.0059	0.0051	0.0048	−0.0106	0.0023	−0.0323	0.0016	−0.0258	0.0149	0.4440	0.4447
500	−0.0051	0.0029	0.0024	0.0025	−0.0064	0.0011	−0.0328	0.0012	−0.0282	0.0145	0.4360	0.4463
1000	−0.0030	0.0016	−0.0014	0.0012	−0.0033	0.0006	−0.0320	0.0010	−0.0295	0.0144	0.4340	0.4455
2000	−0.0025	0.0007	−0.0005	0.0007	−0.0033	0.0003	−0.0314	0.0010	−0.0293	0.0144	0.4340	0.4460
$\sigma_v = 0.5$, $\alpha = 1$, $\sigma_u = 0.5$												
250	−0.0200	0.0050	0.0035	0.0043	−0.0212	0.0021	−0.0779	0.0061	−0.0182	0.0128	0.4560	0.4793
500	−0.0135	0.0025	0.0004	0.0021	−0.0137	0.0010	−0.0774	0.0060	−0.0202	0.0125	0.4520	0.4804
1000	−0.0157	0.0013	−0.0002	0.0011	−0.0134	0.0005	−0.0754	0.0057	−0.0203	0.0124	0.4510	0.4804
2000	−0.0135	0.0007	−0.0002	0.0005	−0.0138	0.0003	−0.0752	0.0057	−0.0210	0.0124	0.4490	0.4810
$\sigma_v = 0.5$, $\alpha = 2$, $\sigma_u = 0.5$												
250	−0.0381	0.0042	−0.0001	0.0036	−0.0354	0.0022	−0.1213	0.0147	−0.0063	0.0102	0.4800	0.5310
500	−0.0336	0.0026	−0.0014	0.0018	−0.0325	0.0014	−0.1184	0.0140	−0.0080	0.0100	0.4780	0.5304
1000	−0.0318	0.0016	0.0001	0.0010	−0.0328	0.0011	−0.1172	0.0137	−0.0084	0.0099	0.4750	0.5301
2000	−0.0311	0.0011	−0.0015	0.0005	−0.0311	0.0010	−0.1182	0.0140	−0.0090	0.0099	0.4750	0.5297

Table D3: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 0.5$, $\sigma_u = 1$, and three values of α, 0.5, 1, and 2. Estimated by SN-Exp

N	β_0		β_1		σ_u		σ_v		α		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
$\sigma_v = 0.5$, $\alpha = 0.5$, $\sigma_u = 1$														
250	−0.0269	0.0106	0.0016	0.0083	−0.0304	0.0053	0.1427	0.0205	−1.4031	3.8953	−0.0035	0.0104	0.4840	0.6329
500	−0.0074	0.0055	−0.0033	0.0042	−0.0137	0.0026	0.1074	0.0117	−1.0346	2.4184	−0.0078	0.0100	0.4680	0.6337
1000	−0.0035	0.0024	0.0002	0.0020	−0.0051	0.0013	0.0837	0.0071	−0.5029	1.5036	−0.0090	0.0097	0.4630	0.6341
2000	−0.0065	0.0012	0.0011	0.0009	−0.0053	0.0006	0.0598	0.0037	−0.5010	0.8909	−0.0090	0.0097	0.4635	0.6330
$\sigma_v = 0.5$, $\alpha = 1$, $\sigma_u = 1$														
250	−0.0250	0.0081	−0.0010	0.0069	−0.0300	0.0051	0.0699	0.0078	−1.2792	4.3265	−0.0027	0.0085	0.4840	0.6641
500	−0.0156	0.0046	0.0015	0.0035	−0.0152	0.0024	0.0488	0.0042	−0.9502	2.7484	−0.0055	0.0082	0.4720	0.6636
1000	−0.0083	0.0021	0.0014	0.0017	−0.0065	0.0011	0.0260	0.0022	−0.1918	1.0044	−0.0078	0.0080	0.4620	0.6635
2000	−0.0006	0.0010	−0.0006	0.0008	−0.0019	0.0006	0.0094	0.0013	−0.0872	0.3694	−0.0082	0.0079	0.4600	0.6634
$\sigma_v = 0.5$, $\alpha = 2$, $\sigma_u = 1$														
250	−0.0293	0.0070	−0.0042	0.0058	−0.0267	0.0045	0.0093	0.0053	−0.8975	4.5767	−0.0037	0.0063	0.4800	0.7044
500	−0.0160	0.0035	0.0038	0.0033	−0.0153	0.0019	−0.0024	0.0021	−0.4370	1.6964	−0.0064	0.0061	0.4620	0.7045
1000	−0.0052	0.0017	−0.0014	0.0016	−0.0088	0.0009	−0.0069	0.0011	−0.2023	0.6871	−0.0080	0.0059	0.4540	0.7041
2000	−0.0028	0.0008	−0.0007	0.0008	−0.0031	0.0005	−0.0045	0.0006	−0.0853	0.4044	−0.0086	0.0059	0.4495	0.7040

Table D4: The DGP and setup are as in Table D3. Estimated by N-Exp

N	β_0		β_1		σ_u		σ_v		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
$\sigma_v = 0.5,\ \alpha = 0.5,\ \sigma_u = 1$												
250	−0.0072	0.0094	−0.0001	0.0090	0.0009	0.0049	−0.0380	0.0026	−0.0078	0.0099	0.4680	0.6331
500	−0.0016	0.0045	−0.0006	0.0042	0.0028	0.0022	−0.0364	0.0016	−0.0097	0.0098	0.4600	0.6338
1000	−0.0023	0.0021	0.0002	0.0021	−0.0020	0.0011	−0.0330	0.0012	−0.0093	0.0096	0.4610	0.6341
2000	−0.0051	0.0010	0.0012	0.0009	−0.0033	0.0005	−0.0312	0.0010	−0.0093	0.0096	0.4615	0.6330
$\sigma_v = 0.5,\ \alpha = 1,\ \sigma_u = 1$												
250	−0.0190	0.0072	0.0019	0.0073	−0.0086	0.0053	−0.0874	0.0078	−0.0045	0.0082	0.4760	0.6642
500	−0.0179	0.0039	0.0007	0.0037	−0.0076	0.0023	−0.0768	0.0059	−0.0065	0.0080	0.4700	0.6636
1000	−0.0137	0.0020	0.0012	0.0018	−0.0094	0.0010	−0.0758	0.0058	−0.0062	0.0080	0.4690	0.6636
2000	−0.0083	0.0009	−0.0012	0.0008	−0.0088	0.0005	−0.0758	0.0058	−0.0065	0.0080	0.4680	0.6635
$\sigma_v = 0.5,\ \alpha = 2,\ \sigma_u = 1$												
250	−0.0315	0.0072	−0.0046	0.0070	−0.0130	0.0054	−0.1282	0.0165	−0.0015	0.0062	0.4880	0.7045
500	−0.0334	0.0041	0.0026	0.0033	−0.0184	0.0026	−0.1232	0.0152	−0.0027	0.0061	0.4820	0.7045
1000	−0.0298	0.0021	−0.0023	0.0017	−0.0226	0.0015	−0.1182	0.0141	−0.0034	0.0060	0.4780	0.7042
2000	−0.0266	0.0012	−0.0008	0.0008	−0.0244	0.0009	−0.1166	0.0136	−0.0032	0.0060	0.4785	0.7040

Table D5: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 1$, $\sigma_u = 0.5$, and three values of α, 0.5, 1, and 2. Estimated by SN-Exp

N	β_0		β_1		σ_u		σ_v		α		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
$\sigma_v = 1$, $\alpha = 0.5$, $\sigma_u = 0.5$														
250	0.0063	0.0483	−0.0756	0.0148	0.0401	0.0263	0.0547	0.0135	−0.4843	0.8307	−0.0646	0.0291	0.4000	0.2582
500	−0.0097	0.0174	−0.0074	0.0076	−0.0088	0.0092	0.0787	0.0101	−0.4905	0.5613	−0.0582	0.0272	0.4080	0.2688
1000	−0.0095	0.0084	−0.0037	0.0035	−0.0075	0.0043	0.0675	0.0069	−0.4890	0.3207	−0.0593	0.0271	0.4060	0.2693
2000	−0.0141	0.0044	−0.0005	0.0020	−0.0078	0.0027	0.0577	0.0049	−0.3974	0.2549	−0.0571	0.0270	0.4080	0.2706
$\sigma_v = 1$, $\alpha = 1$, $\sigma_u = 0.5$														
250	−0.0255	0.0202	−0.0068	0.0124	0.0044	0.0108	0.0163	0.0121	−0.3387	1.0042	−0.0551	0.0255	0.4080	0.3000
500	−0.0208	0.0127	0.0029	0.0066	−0.0016	0.0053	0.0039	0.0065	−0.1604	0.6871	−0.0532	0.0249	0.4110	0.2998
1000	−0.0081	0.0051	−0.0026	0.0030	−0.0009	0.0031	−0.0013	0.0038	−0.0422	0.1697	−0.0551	0.0249	0.4080	0.2990
2000	−0.0063	0.0026	−0.0011	0.0014	−0.0024	0.0014	0.0013	0.0021	−0.0075	0.0685	−0.0546	0.0246	0.4080	0.2988
$\sigma_v = 1$, $\alpha = 2$, $\sigma_u = 0.5$														
250	−0.0215	0.0148	−0.0103	0.0134	−0.0013	0.0050	−0.0490	0.0058	−0.7270	2.0347	−0.0409	0.0217	0.4240	0.3482
500	−0.0034	0.0066	−0.0055	0.0048	0.0029	0.0020	−0.0304	0.0033	−0.1782	0.3949	−0.0489	0.0213	0.4080	0.3442
1000	0.0012	0.0029	−0.0039	0.0023	0.0024	0.0012	−0.0124	0.0015	−0.0930	0.2055	−0.0494	0.0211	0.4080	0.3432
2000	−0.0016	0.0014	0.0030	0.0011	0.0005	0.0006	−0.0049	0.0007	−0.0301	0.0928	−0.0491	0.0209	0.4075	0.3422

Table D6: The DGP and setup are as in Table D5. Estimated by N-Exp

N	β_0		β_1		σ_u		σ_v		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
$\sigma_v = 1$, $\alpha = 0.5$, $\sigma_u = 0.5$												
250	−0.0863	0.0471	−0.0438	0.0120	−0.0727	0.0253	−0.0357	0.0039	−0.0321	0.0276	0.4600	0.2587
500	−0.0299	0.0150	−0.0044	0.0070	−0.0290	0.0070	−0.0583	0.0039	−0.0517	0.0268	0.4180	0.2687
1000	−0.0198	0.0063	0.0002	0.0031	−0.0177	0.0028	−0.0586	0.0036	−0.0543	0.0267	0.4130	0.2697
2000	−0.0192	0.0031	0.0015	0.0019	−0.0174	0.0017	−0.0575	0.0033	−0.0539	0.0267	0.4145	0.2709
$\sigma_v = 1$, $\alpha = 1$, $\sigma_u = 0.5$												
250	−0.1057	0.0276	−0.0022	0.0115	−0.0804	0.0118	−0.1410	0.0199	−0.0247	0.0249	0.4600	0.2981
500	−0.0740	0.0137	0.0075	0.0059	−0.0554	0.0056	−0.1457	0.0213	−0.0304	0.0241	0.4500	0.2992
1000	−0.0569	0.0063	−0.0018	0.0029	−0.0517	0.0036	−0.1454	0.0211	−0.0314	0.0240	0.4480	0.2995
2000	−0.0532	0.0040	−0.0003	0.0013	−0.0522	0.0029	−0.1478	0.0218	−0.0320	0.0240	0.4475	0.2990
$\sigma_v = 1$, $\alpha = 2$, $\sigma_u = 0.5$												
250	−0.1462	0.0255	−0.0060	0.0115	−0.1138	0.0130	−0.2291	0.0525	−0.0025	0.0213	0.4960	0.3470
500	−0.1324	0.0191	−0.0054	0.0047	−0.1236	0.0153	−0.2332	0.0544	0.0068	0.0214	0.5120	0.3432
1000	−0.1132	0.0130	−0.0027	0.0023	−0.1134	0.0129	−0.2343	0.0549	0.0044	0.0212	0.5075	0.3428
2000	−0.1154	0.0134	0.0037	0.0011	−0.1127	0.0127	−0.2350	0.0552	0.0057	0.0212	0.5100	0.3421

Table D7: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 1$, $\sigma_u = 1$, and three values of α, 0.5, 1, and 2. Estimated by SN-Exp

N	β_0		β_1		σ_u		σ_v		α		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
$\sigma_v = 1$, $\alpha = 0.5$, $\sigma_u = 1$														
250	−0.0461	0.0364	−0.0055	0.0206	−0.0342	0.0150	0.1908	0.0375	−0.5165	2.0232	−0.0265	0.0255	0.4480	0.4457
500	−0.0197	0.0143	−0.0098	0.0105	−0.0220	0.0075	0.1732	0.0306	−0.5184	1.4830	−0.0303	0.0247	0.4440	0.4457
1000	−0.0230	0.0080	−0.0019	0.0053	−0.0169	0.0039	0.1166	0.0152	−0.5276	0.9964	−0.0314	0.0243	0.4410	0.4458
2000	−0.0139	0.0034	0.0049	0.0027	−0.0107	0.0018	0.0627	0.0073	−0.5126	0.4259	−0.0327	0.0239	0.4370	0.4461
$\sigma_v = 1$, $\alpha = 1$, $\sigma_u = 1$														
250	−0.0418	0.0264	−0.0024	0.0176	−0.0324	0.0110	0.0834	0.0179	−0.8961	2.2313	−0.0221	0.0223	0.4520	0.4816
500	−0.0226	0.0125	−0.0013	0.0081	−0.0247	0.0055	0.0593	0.0108	−0.2833	1.0515	−0.0239	0.0214	0.4480	0.4815
1000	−0.0145	0.0056	−0.0013	0.0041	−0.0189	0.0030	0.0079	0.0084	−0.5414	0.9996	−0.0255	0.0212	0.4450	0.4812
2000	−0.0138	0.0033	0.0058	0.0023	−0.0138	0.0016	−0.0193	0.0073	−0.4290	0.7083	−0.0267	0.0209	0.4420	0.4808
$\sigma_v = 1$, $\alpha = 2$, $\sigma_u = 1$														
250	−0.0446	0.0181	−0.0038	0.0134	−0.0356	0.0072	−0.0130	0.0102	−0.4742	2.0877	−0.0161	0.0174	0.4560	0.5296
500	−0.0187	0.0084	−0.0007	0.0070	−0.0183	0.0035	−0.0193	0.0058	−0.2688	0.9383	−0.0205	0.0171	0.4440	0.5298
1000	−0.0148	0.0045	0.0010	0.0036	−0.0134	0.0022	−0.0166	0.0035	−0.1631	0.5327	−0.0223	0.0169	0.4390	0.5294
2000	−0.0083	0.0024	0.0019	0.0020	−0.0086	0.0011	−0.0154	0.0019	−0.1309	0.2683	−0.0235	0.0167	0.4345	0.5298

Table D8: The DGP and setup are as in Table D7. Estimated by N-Exp

N	β_0		β_1		σ_u		σ_v		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
$\sigma_v = 1,\ \alpha = 0.5,\ \sigma_u = 1$												
250	−0.0127	0.0250	−0.0031	0.0195	−0.0102	0.0100	−0.0692	0.0070	−0.0315	0.0246	0.4400	0.4461
500	−0.0068	0.0115	−0.0084	0.0103	−0.0114	0.0049	−0.0637	0.0047	−0.0331	0.0242	0.4360	0.4458
1000	−0.0059	0.0058	−0.0014	0.0050	−0.0058	0.0024	−0.0654	0.0043	−0.0339	0.0239	0.4350	0.4464
2000	−0.0077	0.0029	0.0029	0.0026	−0.0055	0.0013	−0.0612	0.0037	−0.0335	0.0238	0.4355	0.4461
$\sigma_v = 1,\ \alpha = 1,\ \sigma_u = 1$												
250	−0.0417	0.0210	0.0020	0.0168	−0.0302	0.0085	−0.1566	0.0246	−0.0208	0.0216	0.4560	0.4816
500	−0.0302	0.0109	−0.0010	0.0077	−0.0315	0.0043	−0.1491	0.0222	−0.0211	0.0211	0.4560	0.4818
1000	−0.0291	0.0047	−0.0004	0.0039	−0.0297	0.0021	−0.1500	0.0225	−0.0223	0.0210	0.4510	0.4812
2000	−0.0304	0.0029	0.0040	0.0022	−0.0299	0.0015	−0.1485	0.0220	−0.0230	0.0209	0.4500	0.4812
$\sigma_v = 1,\ \alpha = 2,\ \sigma_u = 1$												
250	−0.0706	0.0184	−0.0041	0.0132	−0.0736	0.0095	−0.2400	0.0576	−0.0061	0.0176	0.4800	0.5300
500	−0.0636	0.0094	−0.0020	0.0074	−0.0670	0.0055	−0.2382	0.0568	−0.0083	0.0172	0.4760	0.5299
1000	−0.0646	0.0058	0.0014	0.0033	−0.0637	0.0044	−0.2371	0.0562	−0.0090	0.0172	0.4750	0.5296
2000	−0.0614	0.0042	−0.0001	0.0018	−0.0638	0.0041	−0.2357	0.0556	−0.0095	0.0170	0.4740	0.5301

D.2 The True Noise Originates from the Normal Distribution

Table D9: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim N(0, \sigma_v^2)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 0.5$ and $\sigma_u = 0.5$.

N	β_0		β_1		σ_u		σ_v		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
Estimated by SN-Exp												
250	−0.0253	0.0104	−0.0043	0.0059	−0.0154	0.0040	0.1390	0.0193	−0.0237	0.0165	0.4480	0.4286
500	−0.0102	0.0040	−0.0006	0.0026	−0.0086	0.0021	0.1183	0.0140	−0.0290	0.0159	0.4380	0.4274
1000	−0.0106	0.0023	0.0004	0.0015	−0.0068	0.0011	0.0998	0.0100	−0.0297	0.0158	0.4360	0.4270
2000	−0.0067	0.0011	−0.0008	0.0007	−0.0060	0.0005	0.0794	0.0063	−0.0304	0.0156	0.4345	0.4271
Estimated by N-Exp												
250	−0.0044	0.0070	−0.0057	0.0061	−0.0035	0.0026	−0.0032	0.0011	−0.0298	0.0159	0.4360	0.4286
500	−0.0005	0.0029	−0.0008	0.0026	−0.0022	0.0012	−0.0031	0.0006	−0.0336	0.0156	0.4280	0.4279
1000	−0.0031	0.0016	0.0001	0.0014	−0.0002	0.0006	−0.0013	0.0003	−0.0329	0.0156	0.4300	0.4271
2000	−0.0006	0.0007	−0.0005	0.0007	−0.0014	0.0003	−0.0004	0.0001	−0.0325	0.0155	0.4300	0.4272

Table D10: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim N(0, \sigma_v^2)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 0.5$ and $\sigma_u = 1$.

N	β_0		β_1		σ_u		σ_v		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
Estimated by SN-Exp												
250	−0.0400	0.0124	0.0134	0.0098	−0.0299	0.0061	0.1857	0.0345	−0.0045	0.0114	0.4800	0.6152
500	−0.0140	0.0052	0.0014	0.0046	−0.0171	0.0028	0.1464	0.0214	−0.0080	0.0110	0.4700	0.6159
1000	−0.0068	0.0028	0.0012	0.0022	−0.0086	0.0013	0.1186	0.0141	−0.0101	0.0108	0.4620	0.6154
2000	−0.0060	0.0014	0.0002	0.0012	−0.0045	0.0007	0.0954	0.0091	−0.0100	0.0107	0.4610	0.6156
Estimated by N-Exp												
250	−0.0160	0.0106	0.0129	0.0101	0.0003	0.0052	−0.0056	0.0021	−0.0079	0.0110	0.4680	0.6153
500	−0.0034	0.0044	0.0013	0.0045	−0.0058	0.0024	−0.0022	0.0009	−0.0100	0.0108	0.4600	0.6160
1000	−0.0013	0.0024	0.0029	0.0021	−0.0021	0.0011	−0.0020	0.0004	−0.0109	0.0107	0.4580	0.6155
2000	−0.0016	0.0012	0.0000	0.0012	−0.0011	0.0005	−0.0010	0.0002	−0.0111	0.0107	0.4580	0.6157

Table D11: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim N(0, \sigma_v^2)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 1$ and $\sigma_u = 0.5$.

N	β_0		β_1		σ_u		σ_v		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
Estimated by SN-Exp												
250	0.0043	0.0320	−0.0108	0.0174	0.0182	0.0253	0.1689	0.0298	−0.0705	0.0297	0.3920	0.2535
500	−0.0020	0.0179	−0.0045	0.0081	0.0099	0.0106	0.1362	0.0194	−0.0686	0.0290	0.3930	0.2568
1000	0.0015	0.0093	−0.0026	0.0047	0.0035	0.0058	0.1165	0.0137	−0.0666	0.0287	0.3960	0.2552
2000	−0.0150	0.0054	0.0001	0.0023	−0.0129	0.0034	0.1105	0.0123	−0.0581	0.0281	0.4085	0.2547
Estimated by N-Exp												
250	−0.0392	0.0364	−0.0076	0.0155	−0.0207	0.0175	0.0028	0.0040	−0.0602	0.0298	0.4040	0.2519
500	−0.0202	0.0150	0.0009	0.0069	−0.0072	0.0071	0.0012	0.0019	−0.0641	0.0287	0.3980	0.2564
1000	−0.0079	0.0069	−0.0009	0.0040	−0.0027	0.0036	−0.0016	0.0009	−0.0628	0.0281	0.4010	0.2556
2000	−0.0094	0.0035	0.0017	0.0020	−0.0043	0.0019	0.0013	0.0005	−0.0623	0.0281	0.4025	0.2553

Table D12: The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim N(0, \sigma_v^2)$, and $u \sim Exp(1/\sigma_u)$. Bias and MSE, $\sigma_v = 1$ and $\sigma_u = 1$.

N	β_0		β_1		σ_u		σ_v		Efficiency			
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Upward Bias	Correlation
Estimated by SN-Exp												
250	−0.0685	0.0406	0.0127	0.0233	−0.0392	0.0178	0.2789	0.0779	−0.0269	0.0276	0.4520	0.4291
500	−0.0375	0.0174	0.0005	0.0112	−0.0216	0.0083	0.2364	0.0559	−0.0322	0.0265	0.4420	0.4278
1000	−0.0269	0.0095	−0.0005	0.0059	−0.0195	0.0047	0.1959	0.0385	−0.0341	0.0261	0.4390	0.4273
2000	−0.0190	0.0045	0.0008	0.0027	−0.0174	0.0023	0.1469	0.0216	−0.0350	0.0258	0.4375	0.4278
Estimated by N-Exp												
250	−0.0212	0.0285	0.0107	0.0220	−0.0128	0.0123	−0.0030	0.0042	−0.0346	0.0264	0.4360	0.4297
500	−0.0069	0.0129	0.0003	0.0104	−0.0020	0.0049	−0.0025	0.0019	−0.0374	0.0258	0.4320	0.4284
1000	−0.0058	0.0067	−0.0004	0.0056	−0.0070	0.0025	−0.0010	0.0010	−0.0373	0.0257	0.4320	0.4274
2000	−0.0043	0.0033	0.0011	0.0025	−0.0015	0.0014	0.0002	0.0005	−0.0384	0.0256	0.4305	0.4279

E Simulations and applications

E.1 The R package

We have developed the R package snreg which is briefly discussed in the Section 3.3. It is not yet available on CRAN. The binary package is available in three flavors:

- https://www.dropbox.com/s/sd96aco4cpnw4r4/snreg_0.1_R_x86_64-pc-linux-gnu.tar.gz?dl=0 – Debian
- https://www.dropbox.com/s/u5zp975wrk8krnq/snreg_0.1.zip?dl=0 – Windows
- https://www.dropbox.com/s/sqdo9g2r1vo7o7w/snreg_0.1.tgz?dl=0 – Mac

All three versions were compiled under version 4.1.0 of R. All R codes to replicate 3 applications as well as perform the MC study and produce tables and likelihood profile figure can be downloaded at https://www.dropbox.com/s/4wzhj5kgfzgoefi/R-Codes-replicate.zip?dl=0.

E.2 Banking application

E.2.1 Obtaining the sample of 500 Banks

This code assumes that the file 2b_QLH.txt is downloaded from http://qed.econ.queensu.ca/jae/datasets/restrepo-tobon001/ and the working directory is set to where this file is placed. R codes can be downloaded at https://www.dropbox.com/s/4wzhj5kgfzgoefi/R-Codes-replicate.zip?dl=0.

```
require(dplyr)
require(readr)

banks07 <- read_delim("2b_QLH.txt", "\t", escape_double = FALSE,
                      trim_ws = TRUE)

# rename 'entity' to 'id'
colnames(banks07) [colnames(banks07) == "entity"] <- "id"

# generate required variables
banks07 <- banks07 %>%
  mutate(
```

```
    banks07,
    lnzscore = log(ZSCORE),
    lnsdroa = log(SDROA),
    TC = TOC,
    ER = Z / TA,
    LA = Y2 / TA)

# keep if 2007
banks07 <- banks07 %>% filter(year >= 2007 & year <= 2007)

# SCOPE "HHI index across five loan categories"
banks07 <- banks07 %>%
  mutate(
    # banks07,
    OL  = ToLoLe - (ifelse(is.na(REL),0,REL) +
                    ifelse(is.na(AL),0,AL) +
                    ifelse(is.na(CIL),0,CIL) +
                    ifelse(is.na(IL),0,IL) ),
    SY1 = (REL/ToLoLe)^2,
    SY2 = (AL/ToLoLe)^2,
    SY3 = (CIL/ToLoLe)^2,
    SY4 = (IL/ToLoLe)^2,
    SY5 = (OL/ToLoLe)^2,
    scope = ifelse(is.na(SY1),0,SY1) +
      ifelse(is.na(SY2),0,SY2) +
      ifelse(is.na(SY3),0,SY3)+
      ifelse(is.na(SY4),0,SY4) +
      ifelse(is.na(SY5),0,SY5) )

# Consider banks between the 10th and 90th percentiles
# in terms of TA and TC
lo <- 0.1
up <- 0.9
TAq1q3 <- quantile(banks07$TA, probs = c(lo,up))
TCq1q3 <- quantile(banks07$TC, probs = c(lo,up))

TA_constr <- banks07$TA >= TAq1q3[1] & banks07$TA <= TAq1q3[2]
TC_constr <- banks07$TC >= TCq1q3[1] & banks07$TC <= TCq1q3[2]
SDROA_constr <- banks07$SDROA < 4

banks07 <- banks07[(TA_constr & TC_constr & SDROA_constr),]

# Only required variables
banks07 <-
  banks07[, c("id", "year", "TC", "Y1", "Y2", "W1","W2","W3",
              "ER", "LA", "TA", "ZSCORE", "ZSCORE3", "SDROA", "LLP",
              "lnzscore","lnsdroa",'scope')]

# complete observations
banks07 <- banks07[complete.cases(banks07),]

# 500 randomly selected banks
id_names <- unique(banks07$id)
```

```
N_total <- length(id_names)
set.seed(173451)
ids_n2choose <- sample(1:N_total, 500)
ids2choose <- id_names[ids_n2choose]
banks07 <- banks07[banks07$id %in% ids2choose,]

# sort
banks07 <- banks07[order(banks07$id, banks07$year),]
```

E.2.2 Replication Code

The replication code assumes that the R package **snreg** is installed.

```
data(banks07, package = "snreg")
dat1 <- banks07
head(dat1)
dat1$mysample <- TRUE

myprod <- FALSE

spe.tl <-
  log(TC/W3) ~ (log(Y1) + log(Y2) + log(W1/W3) + log(W2/W3))^2 +
  I(0.5*log(Y1)^2) + I(0.5*log(Y2)^2) +
  I(0.5*log(W1/W3)^2) + I(0.5*log(W2/W3)^2)

# NO ---------------------------------------

m.NO <-
  snreg::lm.mle  (
    formula = spe.tl, data = dat1, subset = mysample,
    ln.var.v  = NULL,
    technique = c('bfgs')
  )

m.NO$value
m.NO$coef
sqrt(diag(m.NO$vcov))
cbind(coef = m.NO$coef, z = m.NO$coef/sqrt(diag(m.NO$vcov)))


# SNO --------------------------------------

formSK <- NULL
formSV <- NULL


init.skY <- seq(0,3,.07)
init.skY[1] <- 0.01
init.skY <- sort(c(-init.skY,init.skY))

tymch1 <- NULL
```

```
for (in.sk in init.skY) {
  m0.sn.tl <-
    snreg::snreg  (
      formula = spe.tl, data = dat1, subset = mysample,
      start.sk  = in.sk,
      ln.var.v  = formSV,
      skew.v    = formSK,
      start.val = NULL,
      technique = c('bfgs'),
      lmtol     = 1e-5,
      reltol    = 1e-15,
      maxit     = 399,
      report    = 1,
      trace     = 1,
      print.level = 4,
      digits    = 4)
  tymch1 <- rbind(tymch1, c(in.sk, m0.sn.tl$value))
}

tymch1[order(tymch1[,2]),]

m.SN0 <-
  snreg::snreg  (
    formula = spe.tl, data = dat1, subset = mysample,
    start.sk  = tymch1[order(tymch1[,2]),][ nrow(tymch1),1],
    ln.var.v  = formSV,
    skew.v    = formSK,
    start.val = NULL,
    technique = c('nr')
  )

m.SN0$ll
m.SN0$coef
sqrt(diag(m.SN0$vcov))
cbind(coef = m.SN0$coef, z = m.SN0$coef/sqrt(diag(m.SN0$vcov)))


# SF0 ---------------------------------------

formSU <- NULL
formSK <- NULL
formSV <- NULL


m.SF0 <-
  npsf::sf(
    spe.tl, data = dat1,
    subset = mysample, prod = myprod,
    ln.var.u.0i = formSU, ln.var.v.0i = formSV,
    distribution = "e",
    lmtol = 1e-5, print.level = 4,
    cost.eff.less.one = TRUE)
```

```

m.SF0$loglik
m.SF0$coef
sqrt(diag(m.SF0$vcov))
cbind(coef = m.SF0$coef, z = m.SF0$coef/sqrt(diag(m.SF0$vcov)))


# SF1 --------------------------------------


formSK <- NULL
formSU <- NULL
formSV <- NULL

m.SF1 <-
  snreg::snsf  (
    formula = spe.tl,
    data = dat1, subset = mysample,
    distribution = "e",
    prod      = myprod,
    start.sk  = -1,
    ln.var.u  = formSU,
    ln.var.v  = formSV,
    skew.v    = formSK,
    mean.u    = NULL,
    start.val = NULL,
    technique = c('bfgs')
  )

m.SF1$ll
m.SF1$coef
sqrt(diag(m.SF1$vcov))
cbind(coef = m.SF1$coef, z = m.SF1$coef/sqrt(diag(m.SF1$vcov)))




# SF2 --------------------------------------


formSK <- ~ SDROA
formSV <- ~ log(TA)
formSU <- ~ scope

m.SF2 <-
  snreg::snsf  (
    formula = spe.tl, data = dat1, subset = mysample == TRUE,
    distribution = "e",
    prod      = myprod,
    start.sk  = -0.5,
    ln.var.u  = formSU,
    ln.var.v  = formSV,
    skew.v    = formSK,
```

```
    mean.u    = NULL,
    start.val = NULL,
    technique = c(’bfgs’),
    # vcetype   = ’opg’,
    lmtol     = 1e-5,
    reltol    = 1e-15,
    maxit     = 999,
    report    = 1,
    trace     = 1,
    print.level = 5,
    digits    = 4,
    threads   = 1)


m.SF2$ll
m.SF2$coef
sqrt(diag(m.SF2$vcov))
cbind(coef = m.SF2$coef, z = m.SF2$coef/sqrt(diag(m.SF2$vcov)))
```

E.3 Beltway Sniper application

The replication code assumes that the R package snreg is installed. The dataset dat2.csv can be downloaded from https://www.dropbox.com/s/f12eh41bfxbk31o/dat2.csv?dl=0. The results in Table 2 can be reproduced using the following code. R codes can be downloaded at https://www.dropbox.com/s/4wzhj5kgfzgoefi/R-Codes-replicate.zip?dl=0.

```
sniper <- read_csv("dat2.csv")
sniper$mysample <- TRUE
head(sniper)

myprod <- TRUE

used.spe <-
  log(mathpass) ~
  log(fte) + log(ratio) +
  I(log(fte)^2) + I(log(ratio)^2) +
  p_black + p_hisp + log(fte):log(ratio)

# NO ---------------------------------------

m.NO <-
  snreg::lm.mle  (
    formula = used.spe,
    data = sniper, subset = mysample,
    ln.var.v  = NULL,
    technique = c(’bfgs’),
```

```
    lmtol     = 1e-5,
    reltol    = 1e-15,
    maxit     = 399,
    report    = 1,
    trace     = 1,
    print.level = 4,
    digits    = 4,
    threads   = Nthreads)

m.NO$value
m.NO$coef
sqrt(diag(m.NO$vcov))


# SNO ---------------------------------------

formSK <- NULL
formSV <- NULL


init.skY <- seq(0,3,.1)
init.skY[1] <- 0.01
init.skY <- sort(c(-init.skY,init.skY))

tymch1 <- NULL

for (in.sk in init.skY) {
  m.SNO <-
    snreg::snreg  (
      formula = used.spe,
      data = sniper, subset = mysample,
      start.sk  = in.sk,
      ln.var.v  = formSV,
      skew.v    = formSK,
      start.val = NULL,
      technique = c('bfgs'),
      lmtol     = 1e-5,
      reltol    = 1e-15,
      maxit     = 399,
      report    = 1,
      trace     = 1,
      print.level = 4,
      digits    = 4,
      threads   = Nthreads)
  tymch1 <- rbind(tymch1, c(in.sk, m.SNO$value))
}

m.SNO <-
  snreg::snreg  (
    formula = used.spe,
    data = sniper, subset = mysample,
    start.sk  = tymch1[order(tymch1[,2]),][ nrow(tymch1)-0,1],
    ln.var.v  = formSV,
```

```
    skew.v    = formSK,
    start.val = NULL,
    technique = c('bfgs'),
    vcetype = 'opg',
    lmtol     = 1e-5,
    reltol    = 1e-15,
    maxit     = 399,
    report    = 1,
    trace     = 1,
    print.level = 4,
    digits    = 4,
    threads   = Nthreads)

m.SN0$value
m.SN0$coef
sqrt(diag(m.SN0$vcov))

# SF0 ---------------------------------------

formSU <- NULL
formSK <- NULL
formSV <- NULL

m.SF0 <-
  npsf::sf(
    formula = used.spe,
    data = sniper,
    subset = mysample, prod = myprod,
    ln.var.u.0i = formSU, ln.var.v.0i = formSV,
    distribution = "e",
    lmtol = 1e-5, print.level = 4,
    cost.eff.less.one = TRUE)

m.SF0$loglik
m.SF0$coef
sqrt(diag(m.SF0$vcov))

# SF1 ---------------------------------------

formSK <- NULL
formSU <- NULL
formSV <- NULL


m.SF1 <-
  snreg::snsf  (
    formula = used.spe,
    data = sniper, subset = mysample,
    distribution = "e",
    prod      = myprod,
    start.sk  = .5,
    ln.var.u  = formSU,
    ln.var.v  = formSV,
```

```
    skew.v    = formSK,
    mean.u    = NULL,
    start.val = NULL,
    technique = c('nr')
  )

m.SF1$ll
m.SF1$coef
sqrt(diag(m.SF1$vcov))


# SF2 ----------------------------------------

formSV <- ~ enroll
formSK <- ~log(closeness)
formSU <- ~ p_frl

m.SF2 <-
  snreg::snsf  (
    formula = used.spe,
    data = sniper, subset = mysample,
    distribution = "e",
    prod      = myprod,
    start.sk  = -0.7,
    ln.var.u  = formSU,
    ln.var.v  = formSV,
    skew.v    = formSK,
    mean.u    = NULL,
    start.val = NULL,
    technique = c('bfgs')
    )

m.SF2$ll
m.SF2$coef
sqrt(diag(m.SF2$vcov))
```

E.4 NBER Data Application

The replication code assumes that the R package snreg is installed. The dataset dat3.csv can be downloaded from https://www.dropbox.com/s/2vd13gw0dp72nvh/dat3.csv?dl=0.

```
require(readr)
require(dplyr)
require(ggplot2)
require(reshape2)

dat0 <- read_csv("dat3.csv")
head(dat0)

myprodLogic <- TRUE
```

```
myprod <- TRUE

# specification ---------------------------------------------------------

formTL  <- log(y) ~ (log(K) + log(L) + log(M))^2 +
  I(0.5*log(K)^2) +  I(0.5*log(L)^2) +  I(0.5*log(M)^2)

formSV <- NULL
formSK <- NULL
formSU <- NULL
formMU <- NULL

# Normal-Exponential SF model --------------------------------------------

my.years <- seq(from = min(dat0$year), to = max(dat0$year))

## initialize matrices and lists to be filled ----

ols.skewness <- NULL
n.exp.rez <-
  data.frame(year = my.years, logl_n_exp = NA, effNexp = NA, m3t = NA,
             p_m3t = NA, skewness = NA, sk_wrong = NA)
n.exp.eff <- n.exp.coef <- list()

curr.year <- 1982

for(curr.year in my.years){
  dat1 <- dat0 %>% mutate(mysubset = year == curr.year & textile)
  tymch <- npsf::sf(
    formula = formTL, data = dat1, subset = mysubset, prod = myprodLogic,
    lmtol = 1e-4,  maxit = 199,
    distribution = 'e',
    mean.u.0i.zero  = TRUE,
    reltol       = 1e-13,
    mean.u.0i       = formMU,
    ln.var.u.0i     = formSU,
    ln.var.v.0i     = formSV,
    print.level  = 5)
  ols.skewness <- c(ols.skewness, tymch$olsSkewness)
  effAve <- ifelse(tymch$olsSkewness > 0, 1,
                   mean(tymch$efficiencies$JLMS, na.rm = TRUE))
  n.exp.rez[n.exp.rez[,1] == curr.year, 2:6] <-
    c(ifelse(tymch$loglik > 1e4, NA, tymch$loglik), effAve,
      ifelse(is.null( tymch$m3t), NA,  tymch$m3t),
      ifelse(is.null(tymch$p_m3t), NA, tymch$p_m3t), tymch$olsSkewness)
  n.exp.rez[n.exp.rez[,1] == curr.year, 7] <- tymch$olsSkewness > 0
  n.exp.eff[[curr.year - min(dat0$year) + 1]] <- tymch$efficiencies$JLMS
  n.exp.coef[[curr.year - min(dat0$year) + 1]] <- tymch$coef
  if(tymch$loglik == 1e4){
    tymch <- m0.NBER.n.exp <- npsf::sf(
      formula = formTL, data = dat1, subset = mysubset, prod = myprodLogic,
      lmtol = 5e-3,  maxit = 199,
      distribution = 'e',
```

```
      mean.u.0i.zero  = TRUE,
      reltol       = 1e-13,
      mean.u.0i       = formMU,
      ln.var.u.0i     = formSU,
      ln.var.v.0i     = formSV,
      print.level  = 5)
    ols.skewness <- c(ols.skewness, tymch$olsSkewness)
    effAve <- ifelse(tymch$olsSkewness > 0, 1,
                     mean(tymch$efficiencies$JLMS, na.rm = TRUE))
    n.exp.rez[n.exp.rez[,1] == curr.year, 2:6] <-
      c(ifelse(tymch$loglik > 1e4, NA, tymch$loglik),
        effAve, tymch$m3t, tymch$p_m3t, tymch$olsSkewness)
    n.exp.rez[n.exp.rez[,1] == curr.year, 7] <- tymch$olsSkewness > 0
    n.exp.eff[[curr.year - min(dat0$year) + 1]] <- tymch$efficiencies$JLMS
    n.exp.coef[[curr.year - min(dat0$year) + 1]] <- tymch$coef
  }
}

## Normal-Exponential SF model done ---------------------------------------


# Skew-Normal-Exponential SF model -----------------------------------------

## multistart in each year -------------------------------------------------
## multistart using different "init.sks"

my.seq <- seq(0, 3, by = .2)
my.seq[1] <- 0.02
init.sks <- sort( c(my.seq,-my.seq) )
B <- length(init.sks)
# my.years <- 1962
my.years <- seq(from = min(dat0$year), to = max(dat0$year))

## initialize matrices and lists to be filled ----

sn.exp.rez.all <- list()

time0 <- proc.time()
for(curr.year in my.years){
  cat('Current year is',curr.year,'\n')
  time3 <- proc.time()
  # get the value of N-Exp
  ll.n.exp <- n.exp.rez[n.exp.rez[,1]==curr.year,2]
  # prepare the data frame for th eresults
  rez <- data.frame(initSk = init.sks,
                    ll_n_exp = ll.n.exp,
                    ll = -999, diff = NA,
                    sk = NA, skSEaim = NA, skSEopg = NA,
                    zSEaim = NA, zSEopg = NA,
                    effSNexp = NA, effNexp = NA, skNexp = NA)
  # get this year only
  dat1 <- dat0 %>% mutate(mysubset = year == curr.year & textile)
  for(i in 1:B){
```

```
    # try to run ‘aim‘
    tymch <- tryCatch( snreg::snsf(
      formula = formTL, data = dat1, subset = mysubset, prod = myprod,
      distribution = ’e’,
      technique    = ’bfgs’,
      vcetype      = ’aim’,
      start.sk      = init.sks[i],
      mean.u.zero  = TRUE,
      reltol       = 1e-12,
      lmtol        = 1e-5,
      maxit        = 99,
      mean.u       = formMU,
      ln.var.u     = formSU,
      ln.var.v     = formSV,
      v.skew       = formSK,
      print.level  = 3,
      threads      = 1), error = function(e) e )
    # if it worked: see conditions
    max.increase <- 20
    if(!inherits(tymch, "error")){
      if(tymch$convergence == 0 & !is.na(tymch$ll) &
         tymch$ll < ll.n.exp + max.increase & tymch$ll > ll.n.exp &
         mean(tymch$eff, na.rm = TRUE) < .99 &
         abs(tymch$par[names(tymch$par) == ’Skew_v0i_(Intercept)’]) < 15 ){
        rez[i,3] <- tymch$ll
        rez[i,4] <- tymch$ll - ll.n.exp
        rez[i,5] <- tymch$par[names(tymch$par) == ’Skew_v0i_(Intercept)’]
        if(!is.null(tymch$sds[names(tymch$par) == ’Skew_v0i_(Intercept)’])){
          rez[i,6] <- tymch$sds[names(tymch$par) == ’Skew_v0i_(Intercept)’]
        }
        # get ‘opg‘ sd
        tymch <- tryCatch( snreg::snsf(
          formula = formTL, data = dat1, subset = mysubset, prod = myprod,
          distribution = ’e’,
          technique    = ’bfgs’,
          vcetype      = ’opg’,
          start.sk      = init.sks[i],
          # start.val    = theta00,
          mean.u.zero  = TRUE,
          reltol       = 1e-12,
          lmtol        = 1e-5,
          maxit        = 99,
          mean.u       = formMU,
          ln.var.u     = formSU,
          ln.var.v     = formSV,
          v.skew       = formSK,
          print.level  = 0,
          threads      = 1), error = function(e) e )
        # if it worked: see conditions (to be honest, it should have in this ‘if‘)
        if(!inherits(tymch, "error")){
          if(tymch$convergence == 0 & !is.na(tymch$ll) &
             tymch$ll < ll.n.exp + max.increase & tymch$ll > ll.n.exp &
             mean(tymch$eff, na.rm = TRUE) < .99 &
```

```
       abs(tymch$par[names(tymch$par) == ’Skew_v0i_(Intercept)’]) < 15 ){
      if(!is.null(tymch$sds[names(tymch$par) == ’Skew_v0i_(Intercept)’])){
       rez[i,7] <- tymch$sds[names(tymch$par) == ’Skew_v0i_(Intercept)’]
      }
     }
    }
    # no good if both ‘opg‘ and ‘aim‘ se are NA: toss it
    if(is.na(rez[i,6]) & is.na(rez[i,7])) rez[i,3] <- -999
    # calculate z-val for ‘aim‘
    if(!is.na(rez[i,6])){
     rez[i,8] <- rez[i,5] / rez[i,6]
    }
    # calculate z-val for ‘opg‘
    if(!is.na(rez[i,7])){
     rez[i,9] <- rez[i,5] / rez[i,7]
    }
   }
  }
  rez[i,10] <- mean(tymch$eff, na.rm = TRUE)
  # rez[i,12] <- round(n.exp.rez[curr.year - min(dat0$year) + 1,6], 2)
  if(n.exp.rez[curr.year - min(dat0$year) + 1,6] > 0){
   rez[i,11] <- 1
  } else {
   rez[i,11] <- mean(n.exp.eff[[curr.year - min(dat0$year) + 1]], na.rm = TRUE)
  }
  # which(lls == max(lls))
 }
 # save into the list
 # sn.exp.rez.all[[curr.year - min(my.years) + 1]] <- rez
 toShow <- 17
 tymch2 <- rez %>% filter(ll != -999) %>% arrange(-ll)
 if(nrow(tymch2) > toShow){
  sn.exp.rez.all[[curr.year - min(dat0$year) + 1]] <- tymch2[1:toShow,]
 } else {
  sn.exp.rez.all[[curr.year - min(dat0$year) + 1]] <- tymch2
 }
 time4 <- proc.time()
 timing((time4 - time3)[1], wording = paste0(’Year ’,curr.year,’ finished in ’))
 print(sn.exp.rez.all[[curr.year - min(dat0$year) + 1]], digits = 4)
}
time1 <- proc.time()
(time1 - time0)

## choose par SE ----------------------------------------------------------------


choice.par <- data.frame(year = my.years, row = NA, vcetype = NA)
for(j in 1:length(my.years)){
 if(nrow(sn.exp.rez.all[[j]]) > 0){
  tymch1 <- sn.exp.rez.all[[j]][,c(’zSEaim’,’zSEopg’)]
  tymch1$useSEaim <-
   ifelse(!is.na(tymch1$zSEaim) && abs(tymch1$zSEaim) < 5e5, 1, 0)
  tymch1$useSEopg <-
```

```
      ifelse(!is.na(tymch1$zSEopg), 2, 0)
    tymch1$rezSE <- tymch1$useSEaim + tymch1$useSEopg
    tymch2 <- list()
    # choose the first where SE is available
    for(i in 1: nrow(tymch1)){
      if(i == 1 & tymch1$rezSE[i] > 0){
        tymch2[[1]] <- 1
        break
      }
      if(tymch1$rezSE[i] > 0){
        tymch2[[1]] <- i
        break
      }
    }
    # which: `aim` or `opg`
    if(tymch1$rezSE[tymch2[[1]]] == 1 | tymch1$rezSE[tymch2[[1]]] == 3){
      tymch2[[2]] <- 'aim'
    } else {
      tymch2[[2]] <- 'opg'
    }
    tymch2
    choice.par[j,2] <- tymch2[[1]]
    choice.par[j,3] <- tymch2[[2]]
  }
}
choice.par

# display rez with the choices
choice.rez <- data.frame(year = my.years)
tymch2 <- NULL
for(j in 1:length(my.years)){
  if(nrow(sn.exp.rez.all[[j]]) > 0){
    tymch1 <- sn.exp.rez.all[[j]][choice.par[j,2],]
    tymch2 <- rbind(tymch2, tymch1)
  }
}
choice.rez <- cbind(my.years, tymch2)
colnames(choice.rez) <- c('year', colnames(tymch1))

## Now use the chosen values from multistart ----

sn.exp.rez <- data.frame(year = my.years, logl_sn_exp = NA,
                         sk_coef = NA, sk_se = NA, sk_lb = NA, sk_ub = NA)
sn.exp.eff <- list()

my.years <- seq(from = min(dat0$year), to = max(dat0$year))

curr.year <- 1965
# my.years <- 1964

sn.exp.rez.all[[i]][choice.par[i,2],1]

for(curr.year in my.years){
```

```
  i <- which(curr.year == my.years)
  dat1 <- dat0 %>% mutate(mysubset = year == curr.year & textile)
  m0.NBER.sn.exp <- snreg::snsf(
    formula = formTL, data = dat1, subset = mysubset, prod = myprod,
    distribution = 'e',
    technique    = 'bfgs',
    # vcetype      = optns[i,'se'],
    vcetype      = choice.par[i,3],
    start.sk     = sn.exp.rez.all[[i]][choice.par[i,2],1],
    # start.val    = n.exp.coef[[i]],
    mean.u.zero  = TRUE,
    reltol       = 1e-12,
    lmtol        = 1e-5,
    maxit        = 99,
    mean.u       = formMU,
    ln.var.u     = formSU,
    ln.var.v     = formSV,
    v.skew       = formSK,
    print.level  = 5,
    threads      = 1)
  sn.exp.rez[sn.exp.rez[,1] == curr.year, 2] <-
    c(ifelse(m0.NBER.sn.exp$ll > 1e2, NA, m0.NBER.sn.exp$ll))
  sn.exp.rez[sn.exp.rez[,1] == curr.year, 2] <-
    c(m0.NBER.sn.exp$ll)
  sn.exp.eff[[curr.year - min(dat0$year) + 1]] <-
    m0.NBER.sn.exp$eff
  tymch1 <- m0.NBER.sn.exp$ctab[nrow(m0.NBER.sn.exp$ctab)-1,1:2]
  # qt(.975, m0.NBER.sn.exp$n - nrow(m0.NBER.sn.exp$ctab))
  qnorm(.975)
  sn.exp.rez[sn.exp.rez[,1] == curr.year, 3:6] <-
    c(tymch1, tymch1[1] - qnorm(.975)*tymch1[2], tymch1[1] +
    qnorm(.975)*tymch1[2])
}

## Skew-Normal-Exponential SF model done ----------------------------


# Figure 8 -------------------------------------------------------------


sn.exp.rez$skSig <-
  (sn.exp.rez$sk_coef < 0 & sn.exp.rez$sk_lb < 0 & sn.exp.rez$sk_ub < 0) |
  (sn.exp.rez$sk_coef > 0 & sn.exp.rez$sk_lb > 0 & sn.exp.rez$sk_ub > 0)
sn.exp.rez$zval <-
  paste0('z-value=', round(sn.exp.rez$sk_coef/sn.exp.rez$sk_se, 2))



# table with efficiencies
eff.ave <- data.frame(year = my.years, eff_sn = NA, eff_n = NA)
for(i in 1:length(my.years)){
  eff.ave[i,2] <- mean(sn.exp.eff[[i]])
  if(n.exp.rez$skewness[i] > 0){
```

```
    eff.ave[i,3] <- 1
  } else {
    eff.ave[i,3] <- mean(n.exp.eff[[i]])
  }
}
head(eff.ave)


# skewness SN-EXP significant -----------------------------------------------

eff.ave$sk <- sn.exp.rez$sk_coef
eff.ave$skSig <- sn.exp.rez$skSig
eff.ave$zval <- sn.exp.rez$zval
head(eff.ave)


# sc 2 categories -----------------------------------------------------------



eff.ave$Coefficient <- ifelse(eff.ave$skSig, "Significant", "Insignificant")
mycolors <- c("Significant" = "red", "Insignificant" = "blue")

ggplot(eff.ave, aes(x=year, y=sk)) +
  geom_point(size = 2, aes(color = Coefficient, shape = Coefficient)) +
  # labs(title = 'Significance of the skewness coefficient by year') +
  geom_label(
    data = eff.ave %>%
      filter(year == 1959 | year == 1963 | year == 1964 | year == 1968 |
               year == 1983 | year == 1984 | year == 1985 | year == 1996
             | year == 1997), # Filter data first
    aes(label=zval),
    # nudge_x = 0.0025, nudge_y = -0.77,
    nudge_x = 4.85,
    nudge_y = 0,#-0.77,
  ) +
  scale_color_manual(values = mycolors) +
  ylab(expression(SN-Exp~Model: hat(alpha))) +
  xlab("Year") +
  # theme_gray() +
  theme(legend.position="bottom") +
  theme(legend.title = element_blank())
ggsave(paste0("nber_sk_estimate_sig.pdf"), width = 19,
       height = 12, units = c("cm"))


eff.ave$Coefficient <-
  ifelse( n.exp.rez$skewness < 0, 'N-Exp: correct skewness',
          ifelse(eff.ave$skSig, 'SN-Exp: significant skewness coef.',
                 'SN-Exp: insignificant skewness coef.'))



```

```
# Figure 9 ---------------------------------------------------------------
# sc 4 categories --------------------------------------------------------

eff.ave$Coefficient <- paste(
  ifelse( n.exp.rez$skewness < 0, 'N-Exp: Correct', 'N-Exp: Wrong'),
  ifelse(eff.ave$skSig, "SN-Exp: Significant",
         "SN-Exp: Insignificant"), sep = "; " )


ggplot(eff.ave, aes(x=year, y=sk)) +
  geom_point(size = 2, aes(color = Coefficient, shape = Coefficient)) +
  theme_gray() +geom_label(
    data = eff.ave %>%
      filter(year == 1959 | year == 1963 | year == 1964 | year == 1968 |
               year == 1983 | year == 1984 | year == 1985 | year == 1996
             | year == 1997), # Filter data first
    aes(label=zval),
    nudge_x = 4.85,
    nudge_y = 0,#-0.77,
  ) +
  ylab(expression(SN-Exp~Model: hat(alpha))) +
  xlab("Year") +
  theme(legend.position = "bottom") +
  theme(legend.title = element_blank()) +
  guides(col = guide_legend(nrow = 2))
ggsave(paste0("nber_sk_estimate_sig2.pdf"), width = 19,
       height = 12.5, units = c("cm"))

# Figure 10 ---------------------------------------------------------------

eff.ave2 <- melt(eff.ave[,1:3], id.vars = 'year',
                 variable.name = "Model", value.name = "Efficiency")
head(eff.ave2)
levels(eff.ave2$Model)[levels(eff.ave2$Model)=='eff_sn'] <- 'SN-Exp'
levels(eff.ave2$Model)[levels(eff.ave2$Model)=='eff_n'] <- 'N-Exp'
eff.ave2$marker_size <- ifelse(eff.ave2$Model == 'SN-Exp', 1.5, 2.5)
eff.ave2$marker_shape <- factor( ifelse(eff.ave2$Model == 'SN-Exp', 6, 5) )

ggplot(eff.ave2, aes(x=year, y=Efficiency, group = Model)) +
  geom_point(aes(shape=Model, color=Model, size=Model)) +
  scale_shape_manual(values=c(1, 17))+
  scale_size_manual(values=c(5,3))+
  theme_gray() +
  theme(legend.position = "bottom") #+
theme(legend.title = element_blank())
ggsave(paste0("nber_ave_effs.pdf"), width = 19, height = 9, units = c("cm"))
```

E.5 Simulation study

In this section as well as in the section where we give the code to produce figures for the profile analysis, we show only the code for Model (5). The analysis for Model (3) is easier as it does not have the u component. This is done to conserve space.

The DGP is $\ln Y = \beta_0 + \beta_1 \ln X + v - u$, where $\beta_0 = 0.2$, $\beta_1 = 0.5$, $v \sim SN(0, \sigma_v^2, \alpha)$, and $u \sim Exp(1/\sigma_u)$. The simulation study is performed for $\sigma_v = 0.5, 1$, $\sigma_u = 0.5, 1$, and three values of α, 0.5, 1, and 2. The simulation study is performed for sample sizes $250, 500, 1000, 2000$.

E.5.1 MC study

The simulation was performed parallel. In the following code `my.q` $(1, ..., 48)$ defines a scenario. The outcome of the code produces 48 files with file name as `rez_MC-SN-Exp-",my.q,".Rdata`. In the next subsection, we show the code to collect the results from these 48 files. `R` codes can be downloaded at https://www.dropbox.com/s/4wzhj5kgfzgoefi/R-Codes-replicate.zip?dl=0.

```
require(sn)
require(dplyr)

# Chosee which one is run ---------------------------------------------------

my.q <- 1 # this can be 1,...,48
cat("current q = ",my.q,"\n", sep = "")

# do the log ----------------------------------------------------------------

toSink <- TRUE
toSink <- FALSE

if(toSink){
  zz <- file(paste0(getwd(),"/R/rez_MC-SN-Exp-",my.q,".txt"), open = "wt")
  sink(zz)
  sink(zz, type = "message")
}

## parameters ---------------------------------------------------------------

sdvY <- c( 0.5, 1 )
alpY <- c( 0.5, 1, 2 )
sduY <- c( 0.5, 1 )
nY   <- c( 250, 500, 1000, 2000 )
coefY <- c(.2, .5)

```

```
set.seed(141876325)

B <- 1999
print.level0 <- 0

## loop identification --------------------------------------------------
# to collect tables later
qqq <- 1
loop.ident <- NULL
for(n.i in 1:length(nY)){
  n0 <- nY[n.i]
  for (sdv.i in 1:length(sdvY)) {
    sv0 <- sdvY[sdv.i]
    for (alp.i in 1:length(alpY)) {
      al1 <- alpY[alp.i]
      for (sdu.i in 1:length(sduY)) {
        su0 <- sduY[sdu.i]
        al2 <- su0 / sv0
        # cat("loop is: n =",n0,"; sv =",sv0,"; alpha =",al1,";
        # alpha2 =",al2," or ",n.i*100 + sdvY.i*10 + alpY.i,"\n")
        loop.ident <-
          rbind(
            loop.ident,
            c(qqq, n0, sv0, al1, al2, su0)
          )
        qqq <- qqq + 1
      }
    }
  }
}

# colnames(loop.ident, do.NULL = FALSE)
colnames(loop.ident) <- c('Counter', 'N', "sv", "alpha", "alpha2", "su")
loop.ident

# now, do just one B times ------

## par in this loop -----------------------------------------------------

n0  <- loop.ident[my.q, 2]
sv0 <- loop.ident[my.q, 3]
al1 <- loop.ident[my.q, 4]
al2 <- loop.ident[my.q, 5]
su0 <- loop.ident[my.q, 6]

time1 <- proc.time()

bias.sn2 <- matrix(ncol = length(coefY)+3, nrow = B+length(toExclude))

##  begin the loop ------------------------------------------------------

for(b in myBY) {

```

```
    # control the seed
    # record the seed
    seedmy[b] <- sample(1:1e8, 1)

    set.seed(seedmy[b])

    ### gen data --------------------------------------------------------------

    #  inputs
    x1  <- runif( n = n0, min = 1, max = 4 )

    # skewed noise
    xi0 <- -sv0 * sqrt(2/pi) * al1 / sqrt(1+al1^2)
    v0  <- sn::rsn( n = n0, xi = xi0, omega = sv0, alpha = al1 )
    u0  <- rexp( n = n0, rate = 1/su0 )

    # output
    lny <- coefY[1] + coefY[2] * log(x1) + v0 - u0

    dat1 <- data.frame(lhs = lny, x1)

    tymch0Y <- list()
    choo <- NULL
    qq <- 1 # this is a counter

    ### multi-start -----------------------------------------------------------

    # go over possible starting values of alpha
    init.skY <- seq(from = 0, to = 3, by = 0.8)
    # alpha can't be 0, so make it small
    init.skY[1] <- 0.01
    # both positive and negative starting values
    init.skY <- sort( unique( c(-init.skY[-1], init.skY[-1], 0.01, al1 ) ) )

    for (in.sk in init.skY) {
      # try to estimate
      tymch00 <- tryCatch(
        sfsnnoise::sfsnnoise(
          lhs ~ log(x1), data = dat1,
          technique = c('nr'),
          start.sk = in.sk,
          distribution = 'e' ,
          prod = TRUE,
          maxit = 119,
          lmtol     = 1e-5,
          reltol    = 1e-15,
          print.level = print.level0),
        error = function(e) e )
      # do if no error
      if(!inherits(tymch00, "error")){
        choo <- rbind(choo,
                      c(tymch00$ll, tymch00$convergence,
                        tymch00$coef['Skew_v0i_(Intercept)'],
```

```
                        tymch00$vcov['Skew_v0i_(Intercept)','Skew_v0i_(Intercept)'],
                        tymch00$RSS, qq, tymch00$coef)
      )
      tymch0Y[[qq]] <- tymch00
      qq <- qq + 1
    }
  }

  ## choose ------------------------------------------------------------------

  if(!is.null(toExclude)){
    choo <- choo[-toExclude, , drop = FALSE]
  }

  choo <- choo[!is.na(choo[,1]), , drop = FALSE]

  # check is converged
  choo <- choo[choo[,2] == 0, , drop = FALSE]


  if(!is.null(choo)){
    colnames(choo) <- c('ll','convergence','skewness','se(alpha)','RRS','counter',
                         "(Intercept)" , "log(x1)",
                         "lnVARv0i_(Intercept)",
                         "Skew_v0i_(Intercept)",
                         "lnVARu0i_(Intercept)")

    if(nrow(choo) > 0){
      # sort first
      choo <- choo[rev(order(choo[,c(1)])), , drop = FALSE]


      if(nrow(choo) > 1){
        # # if ll is greater, but RSS is not, discard that row
        # # maybe needs improvement
        choo1 <- choo[!is.na(choo[,1]),c('ll','RRS','counter')]

        RRS.imrove <- choo1[-nrow(choo1),2] < choo1[-1,2]
        which.improve <- which ( RRS.imrove)
        my.counter <- choo[min(which.improve), 6]
        if(is.na(my.counter)) my.counter <- 1 # this for nrow(choo) = 2 and RSS not de
      } else {
        my.counter <- choo[1,6]
      }

      ## the chosen ------------------------------------------------------------

      tymch0 <- tymch0Y[[ my.counter ]]

      bias.sv.sn2 <- sqrt(exp(tymch0$coef['lnVARv0i_(Intercept)'])) - sv0

      bias.su.sn2 <- sqrt(exp(tymch0$coef['lnVARu0i_(Intercept)'])) - su0

```

```
      al1h <- tymch0$coef['Skew_v0i_(Intercept)']
      gamma1 <- 0.5*(4-pi)*sign(al1h)* (al1h^2/(pi/2+(pi/2-1)*al1h^2))^1.5
      bias.al.sn2 <- al1h - al1

      bias.sn2[b,] <- c(coef(tymch0)[1:2] - coefY, bias.su.sn2, bias.sv.sn2, bias.al.s

      cat(" ================= Current b = ",b," ================= \n", sep = "")
    } else {
      cat(" ================= Current b = ",b," is skipped =================  \n", sep
      # warning("this b = ",b," is skipped")
    }

  } else {
    cat(" ================= Current b = ",b," is not successfull =================  \n
  }

  save(list=ls(), file = paste0(getwd(),"/R/rez_MC-SN-Exp-",my.q,".Rdata"))
}

time2 <- proc.time()
time2 - time1

if(toSink){
  sink()
}

rm(dat1)
rm(lny)
rm(x1)

cat("saving file 0\n")
save(list=ls(), file = paste0(getwd(),"/R/rez_MC-SN-Exp-",my.q,".Rdata"))
```

E.5.2 Collection of the results

The results now are collected from the 48 files (rez_MC-SN-Exp-",my.q,".Rdata) in order to reproduce tables in the Appendix C.

```
require(xtable)
my.sanitize.text.function <-
  function(str)
    gsub("e-0", "e-", gsub("NA", "", str, fixed = TRUE), fixed = FALSE)

# need to figure my.q from the scenario

min.N <- 250

for (sdv.i in 1:length(sdvY)) {
  sv0 <- sdvY[sdv.i]
  for (alp.i in 1:length(alpY)) {
```

```
    al1 <- alpY[alp.i]
    which.my.qY <- loop.ident[, 3] == sv0 &
      loop.ident[, 4] == al1 &
      loop.ident[, 2] >= min.N
    my.qY <- loop.ident[which.my.qY, 1]
    print(my.qY)
    my.q <- 30
    my.rezY <- NULL
    for (my.q in my.qY) {
      # load data
      load( paste0(getwd(),"/R/rez_MC-SN-Exp-",my.q,".Rdata"))

      mse.sn2 <- bias.sn2^2

      my.order <- c(seq(from = 1, by = 2, length.out = length(my.bias)),
                    seq(from = 2, by = 2, length.out = length(my.bias))
      )
      my.rez <- c(my.bias, my.mse) [ order(my.order) ]
      my.rezY <- rbind(my.rezY, my.rez)
    }
    my.rezY <- data.frame(nY[nY >= min.N], my.rezY)
    my.rezY[,1] <- as.character(my.rezY[,1])
    colnames(my.rezY) <-
      c('N','b0_bias','b0_mse','b1_bias','b1_mse','sV_bias',
        'sV_mse','alhpa_bias','alpha_mse')
    print(my.rezY)
    print( xtable(my.rezY, digits = 4),
           include.rownames = FALSE,
           include.colnames = FALSE,
           only.contents = TRUE,
           hline.after = NULL,
           booktabs = TRUE,
           math.style.negative = TRUE,
           sanitize.text.function = my.sanitize.text.function,
           file =
             paste0(getwd(),"/paper/",paper1,"/tabs/MC_SN_sv_",sv0,"_al1_",al1,".tex")
    )
  }
}
```

E.6 Profiling analysis

Here we describe how to obtain the figures for the profiling analysis that appear in Appendix C.

The following code will produce all the figures.

```
require(sn)

## parameters -------------------------------------------------------------

sdvY <- c( 0.5, 1 )
```

```
alpY <- c( 0.5, 1, 2 )
sduY <- c( 0.5, 1 )
nY   <- c( 250, 500, 1000, 2000 )
coefY <- c(.2, .5)

set.seed(141876325)

time1 <- proc.time()

B <- 99

n.alpha <- 150
n.sv <- 50
n.su <- 50

qqq <- 1
for (sdv.i in 1:length(sdvY)) {
  sv0 <- sdvY[sdv.i]
  for (alp.i in 1:length(alpY)) {
    al1 <- alpY[alp.i]
    for (sdu.i in 1:length(sduY)) {
      su0 <- sduY[sdu.i]
      al2 <- su0 / sv0

      # * grid of alpha ---------------------------------------------------------

      if(al1 == 0.5){
        alphaY <- seq(from = -0.75, to = 1.0, length.out = n.alpha)
      }
      if(al1 == 1.0){
        alphaY <- seq(from = -1.5, to = 2.5, length.out = n.alpha)
      }
      if(al1 == 1.5){
        alphaY <- seq(from = -1.9, to = 3.3, length.out = n.alpha)
      }
      if(al1 == 2.0){
        alphaY <- seq(from = -2.1, to = 4.0, length.out = n.alpha)
      }
      if(al1 == 3.0){
        alphaY <- seq(from = -2.5, to = 6.0, length.out = n.alpha)
      }
      alphaY <- alphaY[!(alphaY == 0)]

      # * grid of sigma_v -------------------------------------------------------
      svY <- seq(from = sv0/2.5, to = sv0*2.5, length.out = n.sv)

      # * grid of sigma_u -------------------------------------------------------
      suY <- seq(from = su0/2.5, to = su0*2.5, length.out = n.su)

      # medians
      profiles.al.med <- data.frame(stat = as.matrix(alphaY))
      profiles.sv.med <- data.frame(stat = as.matrix(svY))
      profiles.su.med <- data.frame(stat = as.matrix(suY))
```

```
time1 <- proc.time()

for(n.i in 1:length(nY)){

  n0 <- nY[n.i]

  if(n.i == 1) cat("\nloop is: N = ",n0,"; sv = ",sv0,"; alpha = ",al1,";
                   alpha2 = ",al2,"; su = ",su0,";
                   qqq = ",qqq,"/",nrow(loop.ident)/length(nY),"\n", sep = "")
  cat(" current N = ",n0,"\n", sep = "")

  profiles.svY <- matrix(NA, nrow = B, ncol = length(svY) )
  profiles.alY <- matrix(NA, nrow = B, ncol = length(alphaY) )
  profiles.suY <- matrix(NA, nrow = B, ncol = length(suY) )

  for (b in seq.int(B)) {

    # ** control the seed ------------------------------------------------------
    set.seed(sample(1:1e8, 1))


    # ** data ------------------------------------------------------------------

    #  input
    # x1  <- runif( n = n0, min = 1, max = 4 )

    # skewed noise
    xi0 <- -sv0 * sqrt(2/pi) * al1 / sqrt(1+al1^2)
    v0  <- sn::rsn( n = n0, xi = xi0, omega = sv0, alpha = al1 )
    v0  <- as.numeric(v0)
    u0  <- rexp( n = n0, rate = 1/su0 )

    # output
    lny <- coefY[1] + v0 - u0

    ones <- as.matrix( rep(1, n0) )
    X <- ones

    # true values
    true.theta <- c(Intercept = coefY[1],
                    lnVARv0i_Intercept = log(sv0^2),
                    Skew_v0i_Intercept = al1,
                    lnVARu0i_Intercept = log(su0^2))

    # ll at true values

    # ** ll at true values -----------------------------------------------------

    ll0 <- .ll.sn.exp(
      theta = true.theta, myprod = TRUE,
      y=lny, x=X, zsv=ones, zsk=ones, zsu=ones,
      n.ids=n0, k=ncol(X), ksv=ncol(ones), ksk=ncol(ones), ksu = ncol(ones))
```

```
        ll0

        # ** profiling alpha --------------------------------------------------

        tymch0 <- numeric(length(alphaY))
        for(i in seq_len(length(alphaY))){
          # current theta
          curr.theta <- c(Intercept = coefY[1],
                          # slope = coefY[2],
                          lnVARv0i_Intercept = log(sv0^2),
                          Skew_v0i_Intercept = alphaY[i],
                          lnVARu0i_Intercept = log(su0^2))
          # log-likelihood at current theta
          tymch0[i] <- .ll.sn.exp(
            theta = curr.theta, myprod = TRUE,
            y=lny, x=X, zsv=ones, zsk=ones, zsu=ones,
            n.ids=n0, k=ncol(X), ksv=ncol(ones), ksk=ncol(ones), ksu = ncol(ones))
        }
        profiles.alY[b,] <- 2*(tymch0-ll0)

        # ** profiling sigma_v ------------------------------------------------

        tymch0 <- numeric(length(svY))
        for(i in seq_len(length(svY))){
          # current theta
          curr.theta <- c(Intercept = coefY[1],
                          # slope = coefY[2],
                          lnVARv0i_Intercept = log(svY[i]^2),
                          Skew_v0i_Intercept = al1,
                          lnVARu0i_Intercept = log(su0^2))
          # log-likelihood at current theta
          tymch0[i] <- .ll.sn.exp(
            theta = curr.theta, myprod = TRUE,
            y=lny, x=X, zsv=ones, zsk=ones, zsu=ones,
            n.ids=n0, k=ncol(X), ksv=ncol(ones), ksk=ncol(ones), ksu = ncol(ones))

        }
        profiles.svY[b,] <- 2*(tymch0-ll0)

        # ** profiling sigma_u ------------------------------------------------

        tymch0 <- numeric(length(suY))
        for(i in seq_len(length(suY))){
          # current theta
          curr.theta <- c(Intercept = coefY[1],
                          # slope = coefY[2],
                          lnVARv0i_Intercept = log(sv0^2),
                          Skew_v0i_Intercept = al1,
                          lnVARu0i_Intercept = log(suY[i]^2))
          # log-likelihood at current theta
          tymch0[i] <- .ll.sn.exp(
            theta = curr.theta, myprod = TRUE,
```

```
            y=lny, x=X, zsv=ones, zsk=ones, zsu=ones,
            n.ids=n0, k=ncol(X), ksv=ncol(ones), ksk=ncol(ones), ksu = ncol(ones))

        }
        # plot(suY, tymch0)
        profiles.suY[b,] <- 2*(tymch0-ll0)

      }

      # ** aggregating ------------------------------------------------------------

      profiles.sv.med <-
        cbind(profiles.sv.med, apply(profiles.svY, MARGIN = 2, median, na.rm = TRUE)
      profiles.al.med <-
        cbind(profiles.al.med, apply(profiles.alY, MARGIN = 2, median, na.rm = TRUE)
      profiles.su.med <-
        cbind(profiles.su.med, apply(profiles.suY, MARGIN = 2, median, na.rm = TRUE)

    }

    time2 <- proc.time()

    timing((time2 - time1)[3])

    # graphs ----------------------------------------------------------------------

    my.width <- 17
    my.height <- 10
    my.linewd <- 0.9


    # * alpha ---------------------------------------------------------------------


    profiles.al <- profiles.al.med

    colnames(profiles.al)[-1] <- as.character(nY)
    head(profiles.al)

    profiles.al.melted <-
      reshape2::melt(data = profiles.al,
                     id.vars = 'stat',
                     variable.name = 'Sample')
    head(profiles.al.melted)

    myupper <- -mylow

    mylow <- min(alphaY)
    myupper <- max(alphaY)

    ggplot(profiles.al.melted %>% filter(stat >= mylow & stat <= myupper),
           aes(x=stat, y=value)) +
      geom_line(aes(color = Sample, group = Sample), size = my.linewd) +
```

```
    xlab(expression(alpha)) +
    ylab('2*LR') +
    # theme(legend.position = "none") +
    theme_bw() +
    theme(
      axis.title = element_text(size = 20),
      axis.text.x = element_text(angle = 0, hjust = NULL, size = 20),
      axis.text.y = element_text(angle = 0, hjust = NULL, size = 20))
  ggsave(
    paste0("profile_SNExp_al_sv_",sv0,"_al1_",al1,"_su0_",su0,".pdf"),
    width = my.width, height = my.height, units = c("cm"))


  # * sigma v ------------------------------------------------------------

  profiles.sv <- profiles.sv.med

  colnames(profiles.sv)[-1] <- as.character(nY)
  head(profiles.sv)

  profiles.sv.melted <-
    reshape2::melt(data = profiles.sv,
                   id.vars = 'stat',
                   variable.name = 'Sample')
  head(profiles.sv.melted)

  mylow <- sv0/2.5
  myupper <- sv0*2.5

  ggplot(profiles.sv.melted %>% filter(stat >= mylow & stat <= myupper),
         aes(x=stat, y=value)) +
    geom_line(aes(color = Sample, group = Sample), size = my.linewd) +
    xlab(expression(sigma[v])) +
    ylab('2*LR') +
    # theme(legend.position = "none") +
    theme_bw() +
    theme(
      axis.title = element_text(size = 20),
      axis.text.x = element_text(angle = 0, hjust = NULL, size = 20),
      axis.text.y = element_text(angle = 0, hjust = NULL, size = 20))
  ggsave(
    paste0("profile_SNExp_sv_sv_",sv0,"_al1_",al1,"_su0_",su0,".pdf"),
    width = my.width, height = my.height, units = c("cm"))

  # * sigma u ------------------------------------------------------------

  profiles.su <- profiles.su.med

  colnames(profiles.su)[-1] <- as.character(nY)
  head(profiles.su)

  profiles.su.melted <-
    reshape2::melt(data = profiles.su,
```

```
                      id.vars = 'stat',
                      variable.name = 'Sample')
      head(profiles.su.melted)

      mylow <- su0/2.5
      myupper <- su0*2.5

      ggplot(profiles.su.melted %>% filter(stat >= mylow & stat <= myupper),
             aes(x=stat, y=value)) +
        geom_line(aes(color = Sample, group = Sample), size = my.linewd) +
        xlab(expression(sigma[u])) +
        ylab('2*LR') +
        # theme(legend.position = "none") +
        theme_bw() +
        theme(
          axis.title = element_text(size = 20),
          axis.text.x = element_text(angle = 0, hjust = NULL, size = 20),
          axis.text.y = element_text(angle = 0, hjust = NULL, size = 20))
      ggsave(
        paste0("profile_SNExp_su_sv_",sv0,"_al1_",al1,"_su0_",su0,".pdf"),
        width = my.width, height = my.height, units = c("cm"))

      qqq <- qqq + 1
    }
  }
}
```