Supplementary material for the paper "Identifiability and bias reduction in the skew-probit model for a binary response"

DongHyuk Lee and Samiran Sinha
Department of Statistics, Texas A&M University, College Station, TX 77843-3143, USA
emails: dhyuklee@stat.tamu.edu, sinha@stat.tamu.edu

S.1 Figures and tables of the simulation study

In this section, we provide the simulation results for scenarios 5-12. Figures S.1 $-$ S.8 present boxplots for each parameters corresponding to the simulation scenarios $5-12$, respectively. Tables S.1 $-$ S.3 contain the mean and the standard deviation of computation time for simulation scenarios 5-16. Detailed discussion of the simulation results can be found in Section 4 of the manuscript.

S.2 Comparison between the optimization algorithms

Here we have compared different optimization algorithms for estimating parameters under the five methods. Particularly, we compared algorithms Nelder-Mead, BFGS, L-BFGS-B, nlm, nlminb, ucminf, newuoa, bobyqa, nmkb that are available in the `R` package `optimx`, in terms of mean and standard deviation of the estimators (Tables S.4, S.5, S.6), computation time (Table S.7), and the number of non-converging datasets out of 1000 replications (Table S.8). Here we present the results for scenario 6 ($X \sim \text{Normal}(0, (\sqrt{4/3})^2)$, $\beta_1 = 1$, $\delta = 4$, $\beta_0 = 0.42$, $p_m = 40\%$) only. However, based on our short and limited simulation study, the results for other scenarios follow the same trend as of scenario 6. The simulation results can be summarized as follows:

- When sample size is large, resulting estimates are quite close regardless of the algorithm and method of estimation for β_0, β_1 and δ.
- With large sample sizes, ucminf computes estimates much faster than others.
- For methods J and C, algorithms except BFGS produce very close results for β_0, β_1 and δ. Also, the mean bias from algorithm BFGS is slightly larger than that from other algorithm when the sample size is small.
- For method N, estimates seem to differ across algorithms.
- Nelder-Mead, BFGS, L-BFGS-B, nlm and nlminb are suffering from the non-convergence issue especially for method N.

Based on these findings, by far `ucminf` seems to be the best algorithm among the algorithms we consider.

S.3 R codes

```
## Necessary libraries
library(sn)
library(ucminf)

## Method N
loglik <- function(paras, X, y){
  delta <- paras[length(paras)]
  eta1 <- as.vector(X %*% paras[-length(paras)])
  mu1 <- psn(as.vector(eta1), alpha = delta)
  mu1[which(mu1==0)] <- min(mu1[mu1 != 0])
  mu1[which(mu1==1)] <- max(mu1[mu1 != 1])
  re <- sum(y*log(mu1) + (1-y)*log(1-mu1))
  return(-re)
}

## Method B
iMat <- function(y, X, paras){
  inf.mat=matrix(0, nrow=length(paras), ncol=length(paras))
  delta=paras[length(paras)]
  eta1 <- as.vector(X %*% paras[-length(paras)])
  mu1 <- psn(as.vector(eta1), alpha = delta)
  mu1 <- psn(as.vector(eta1), alpha = paras[length(paras)])
  #mu1[which(mu1==0)] <- min(mu1[mu1 != 0])
  mu1[which(mu1==0)] <- 10e-10
  #mu1[which(mu1==1)] <- max(mu1[mu1 != 1])
  mu1[which(mu1==1)] <- 1-10e-10
  term0= dnorm(eta1)*pnorm(delta*eta1)
  term1=mu1*(1-mu1)
  term2=term0*term0/term1
  term3=exp(-0.5*eta1^2*(1+delta^2))
  term4=term3*term3
  inf.mat.b <- 4*t(term2*X) %*% X
  inf.mat.bd <- -2*colSums((term0*term3/term1)*X)/(pi*(1+delta^2))
  inf.mat.d <- sum(term4/term1)/(pi*(1+delta^2))^2
  inf.mat[-length(paras), -length(paras)] <- inf.mat.b
  inf.mat[length(paras), length(paras)] <- inf.mat.d
  inf.mat[length(paras), -length(paras)] <- inf.mat.bd
  inf.mat[-length(paras), length(paras)] <- inf.mat.bd
  return(inf.mat)
}

## Method J
Jloglikp <- function(paras, X, y){
  inf.mat=matrix(0, nrow=length(paras), ncol=length(paras))
  delta=paras[length(paras)]
  eta1 <- as.vector(X %*% paras[-length(paras)])
  mu1 <- psn(as.vector(eta1), alpha = delta)
  mu1 <- psn(as.vector(eta1), alpha = paras[length(paras)])
  #mu1[which(mu1==0)] <- min(mu1[mu1 != 0])
  mu1[which(mu1==0)] <- 10e-10
  #mu1[which(mu1==1)] <- max(mu1[mu1 != 1])
  mu1[which(mu1==1)] <- 1-10e-10
  term0= dnorm(eta1)*pnorm(delta*eta1)
  term1=mu1*(1-mu1)
  term2=term0*term0/term1
  term3=exp(-0.5*eta1^2*(1+delta^2))
  term4=term3*term3
  inf.mat.b <- 4*t(term2*X) %*% X
  inf.mat.bd <- -2*colSums((term0*term3/term1)*X)/(pi*(1+delta^2))
  inf.mat.d <- sum(term4/term1)/(pi*(1+delta^2))^2
  inf.mat[-length(paras), -length(paras)] <- inf.mat.b
  inf.mat[length(paras), length(paras)] <- inf.mat.d
  inf.mat[length(paras), -length(paras)] <- inf.mat.bd
  inf.mat[-length(paras), length(paras)] <- inf.mat.bd
  if(det(inf.mat) < 0) qnty1=0 else qnty1=0.5*log(det(inf.mat))
```

```
  #######
  re <- sum(y*log(mu1) + (1-y)*log(1-mu1)) + qnty1
  return(-re)
}

## Method C
Cloglikp <- function(paras, X, y){
  delta=paras[length(paras)]
  eta1 <- as.vector(X %*% paras[-length(paras)])
  mu1 <- psn(as.vector(eta1), alpha = delta)
  mu1[which(mu1==0)] <- min(mu1[mu1 != 0])
  mu1[which(mu1==1)] <- max(mu1[mu1 != 1])
    re <- sum(y*log(mu1) + (1-y)*log(1-mu1)) - sum(log(1+paras^2/2.5^2))
  return(-re)
}

## Method G
GJloglikp <- function(paras, X, y){
  inf.mat=matrix(0, nrow=length(paras), ncol=length(paras))
  delta=paras[length(paras)]
  eta1 <- as.vector(X %*% paras[-length(paras)])
  mu1 <- psn(as.vector(eta1), alpha = delta)
  mu1 <- psn(as.vector(eta1), alpha = paras[length(paras)])
  #mu1[which(mu1==0)] <- min(mu1[mu1 != 0])
  mu1[which(mu1==0)] <- 10e-10
  #mu1[which(mu1==1)] <- max(mu1[mu1 != 1])
  mu1[which(mu1==1)] <- 1-10e-10
  term0= dnorm(eta1)*pnorm(delta*eta1)
  term1=mu1*(1-mu1)
  term2=term0*term0/term1
  term3=exp(-0.5*eta1^2*(1+delta^2))
  term4=term3*term3
  inf.mat.b <- 4*t(term2*X) %*% X
  inf.mat.bd <- -2*colSums((term0*term3/term1)*X)/(pi*(1+delta^2))
  inf.mat.d <- sum(term4/term1)/(pi*(1+delta^2))^2
  inf.mat[-length(paras), -length(paras)] <- inf.mat.b
  inf.mat[length(paras), length(paras)] <- inf.mat.d
  inf.mat[length(paras), -length(paras)] <- inf.mat.bd
  inf.mat[-length(paras), length(paras)] <- inf.mat.bd
  if(det(inf.mat) < 0) qnty1=0 else qnty1=0.5*log(det(inf.mat))
  #######
  re <- sum(y*log(mu1) + (1-y)*log(1-mu1)) + qnty1 - as.numeric(0.5*t(paras)%*%inf.mat%*%paras)
  return(-re)
}

##########################################################################################
## Data generation
set.seed(101)
n <- 200
b0 <- 0.37
b1 <- 1
delta <- 4
x <- runif(n, -2, 2)
X <- cbind(1, x)
eta <- as.numeric(b0 + b1*x)
p <- psn(eta, alpha = delta)
y <- rbinom(n, 1, p)

## Probit regression
PR <- glm(y ~ x, family = binomial(link = "probit"))

## Initial value for beta parameters
beta0 <- coef(PR)
delta0 <- runif(1, 0, 10)

## Method N
fit_naive <- ucminf(c(beta0, delta0), fn = loglik, X = X, y = y, hessian = 2)
Nest <- fit_naive$par
```

```
Nse <- sqrt(diag(fit_naive$invhessian))
coef_naive <- cbind(Nest, Nse, Nest/Nse, 2*(1-pnorm(abs(Nest/Nse))), Nest + qnorm(0.025)*Nse, Nest + qnorm(0.975)*Nse)

## Method B
store_boot <- matrix(0, nrow = 10, ncol = 3)
k <- 0
total.boot <- 0
n <- nrow(X)
while(1){
  total.boot <- total.boot + 1
  cat(k, '')
  if(!(k%%100)) cat('\n')
  idx.boot <- sample(1:n, n, replace = TRUE)
  beta0.boot <- coef(glm(y[idx.boot] ~ x[idx.boot], family = binomial(link = "probit")))
  fit_boot <- ucminf(c(beta0.boot, delta0), fn = loglik, X = X[idx.boot,], y = y[idx.boot], hessian = 0)
  k <- k+1
  store_boot[k, ] <- fit_boot$par

  if(k == 10) {
    cat('\n')
    break
  }
}
Bmle <- 2*Nest - apply(store_boot, 2, mean)
se <- sqrt(diag(solve(iMat(y, X, Bmle))))
coef_BC <- cbind(Bmle, se, Bmle/se, 2*(1-pnorm(abs(Bmle/se))), Bmle + qnorm(0.025)*se, Bmle + qnorm(0.975)*se)

## Method J
fit_Jeff <- ucminf(c(beta0, delta0), fn = Jloglikp, X = X, y = y, hessian = 2)
Jest <- fit_Jeff$par
Jse <- sqrt(diag(fit_Jeff$invhessian))
coef_Jeff <- cbind(Jest, Jse, Jest/Jse, 2*(1-pnorm(abs(Jest/Jse))), Jest + qnorm(0.025)*Jse, Jest + qnorm(0.975)*Jse)

## Method G
fit_GJ <- ucminf(c(beta0, delta0), fn = GJloglikp, X = X, y = y, hessian = 2)
Gest <- fit_GJ$par
Gse <- sqrt(diag(fit_GJ$invhessian))
coef_GJ <- cbind(Gest, Gse, Gest/Gse, 2*(1-pnorm(abs(Gest/Gse))), Gest + qnorm(0.025)*Gse,  Gest + qnorm(0.975)*Gse)

## Method C
fit_Cauchy <- ucminf(c(beta0, delta0), fn = Cloglikp, X = X, y = y, hessian = 2)
Cest <- fit_Cauchy$par
Cse <- sqrt(diag(fit_Cauchy$invhessian))
coef_Cauchy <- cbind(Cest, Cse, Cest/Cse, 2*(1-pnorm(abs(Cest/Cse))), Cest + qnorm(0.025)*Cse, Cest + qnorm(0.975)*Cse)
```

Table S.1: Mean and standard deviation of the computation time in seconds for simulation scenarios 5-8.

Scenario	n	Method N	Method B	Method J	Method G	Method C
5	200	3.822	803.802	3.130	3.685	1.352
		(1.988)	(133.671)	(0.827)	(1.144)	(0.291)
	500	7.292	1674.464	8.304	10.261	3.745
		(4.564)	(473.796)	(2.334)	(2.736)	(0.722)
	1000	11.497	2718.021	17.068	22.494	7.986
		(7.168)	(915.013)	(3.714)	(4.773)	(1.445)
	2000	18.554	4183.919	34.314	47.367	16.470
		(6.071)	(1092.610)	(6.415)	(8.845)	(2.784)
	5000	44.104	9024.047	86.919	122.305	42.522
		(5.091)	(793.901)	(15.342)	(21.602)	(7.002)
6	200	3.167	681.866	2.584	2.134	1.170
		(1.837)	(166.953)	(0.680)	(0.629)	(0.339)
	500	5.970	1371.743	7.188	5.511	3.524
		(3.932)	(470.717)	(1.525)	(1.350)	(0.725)
	1000	9.017	2167.912	14.773	10.824	7.316
		(4.772)	(790.979)	(2.872)	(2.059)	(1.342)
	2000	16.309	3556.287	30.591	22.632	15.096
		(4.409)	(882.930)	(6.077)	(4.072)	(2.750)
	5000	39.768	8025.889	78.958	52.011	38.814
		(5.111)	(615.991)	(13.735)	(10.190)	(6.579)
7	200	4.195	848.087	3.170	3.600	1.370
		(1.816)	(116.720)	(0.819)	(1.093)	(0.287)
	500	9.803	2016.009	8.440	10.126	3.925
		(4.895)	(479.681)	(1.742)	(2.707)	(0.731)
	1000	17.635	3791.818	18.035	22.730	8.666
		(10.038)	(1114.366)	(3.144)	(4.451)	(1.433)
	2000	28.638	6522.114	37.812	49.211	18.354
		(17.100)	(2194.045)	(6.338)	(9.514)	(2.892)
	5000	55.078	12451.953	100.984	129.091	47.903
		(23.217)	(3832.213)	(20.790)	(25.326)	(7.092)
8	200	3.731	753.511	2.723	2.206	1.273
		(1.760)	(150.036)	(0.656)	(0.667)	(0.313)
	500	8.760	1795.484	7.529	5.658	3.723
		(4.571)	(491.210)	(1.511)	(1.456)	(0.743)
	1000	15.011	3273.005	16.157	11.399	7.991
		(9.131)	(1094.990)	(2.822)	(2.794)	(1.367)
	2000	25.158	5697.765	34.160	22.626	16.969
		(15.048)	(2027.677)	(5.274)	(4.457)	(2.564)
	5000	48.360	10796.577	89.850	52.639	44.345
		(15.035)	(2990.052)	(12.900)	(9.996)	(5.915)

Table S.2: Mean and standard deviation of the computation time in seconds for simulation scenarios 9-12.

Scenario	n	Method				
		N	B	J	G	C
9	200	3.928	799.686	3.266	3.553	1.367
		(1.979)	(150.987)	(0.822)	(1.166)	(0.319)
	1000	7.855	1742.307	9.100	9.444	4.128
		(4.621)	(488.489)	(2.048)	(3.198)	(0.827)
	1000	12.585	2889.193	18.693	21.047	8.622
		(7.449)	(945.755)	(3.834)	(6.379)	(1.535)
	1000	21.253	4743.465	38.893	43.665	18.617
		(7.352)	(1298.788)	(6.389)	(14.482)	(3.145)
	1000	50.869	10281.374	99.519	109.065	48.856
		(5.859)	(1064.178)	(15.539)	(43.728)	(7.189)
10	200	3.190	678.125	2.054	2.181	0.925
		(1.877)	(162.614)	(0.732)	(0.627)	(0.343)
	1000	6.399	1437.032	6.591	5.400	3.136
		(4.501)	(536.712)	(1.657)	(1.207)	(0.810)
	1000	9.811	2342.970	13.954	10.666	6.895
		(6.506)	(955.549)	(2.810)	(1.840)	(1.418)
	1000	15.833	3594.166	28.660	21.818	14.366
		(6.024)	(1085.355)	(5.203)	(3.511)	(2.605)
	1000	37.206	7550.675	73.250	52.955	36.762
		(4.404)	(687.028)	(12.032)	(9.474)	(6.142)
11	200	4.316	847.786	3.271	3.540	1.404
		(1.874)	(138.402)	(0.776)	(1.154)	(0.275)
	1000	10.319	2072.804	9.375	9.354	4.418
		(5.030)	(498.986)	(1.742)	(3.361)	(0.804)
	1000	17.796	3836.815	19.396	20.454	9.336
		(9.738)	(1162.340)	(3.378)	(7.452)	(1.652)
	1000	29.204	6702.102	41.068	43.589	19.964
		(16.359)	(2175.074)	(6.767)	(16.474)	(3.076)
	1000	58.473	13090.967	106.564	115.615	52.310
		(21.303)	(3482.606)	(17.152)	(44.928)	(7.105)
12	200	3.779	740.965	2.280	2.213	1.040
		(1.686)	(136.503)	(0.674)	(0.608)	(0.334)
	1000	9.370	1854.994	7.026	5.565	3.411
		(4.567)	(479.907)	(1.472)	(1.217)	(0.695)
	1000	17.033	3579.004	15.257	10.878	7.455
		(9.285)	(1063.782)	(2.757)	(2.257)	(1.308)
	1000	27.602	6182.438	32.329	21.671	16.086
		(17.852)	(2206.682)	(5.291)	(3.438)	(2.567)
	1000	48.855	11447.403	84.990	52.469	42.309
		(21.553)	(3785.802)	(13.341)	(9.577)	(6.519)

Table S.3: Mean and standard deviation of the computation time in seconds for simulation scenarios 13-16.

Scenario	n	Method N	Method B	Method J	Method G	Method C
13	200	5.507	1121.912	4.273	5.080	1.857
		(2.324)	(130.245)	(0.924)	(1.571)	(0.352)
	500	11.793	2515.691	11.912	12.929	5.502
		(6.347)	(570.718)	(2.493)	(4.855)	(0.956)
	1000	18.154	4209.724	24.966	27.752	11.876
		(10.464)	(1278.700)	(4.661)	(10.013)	(1.931)
	2000	30.402	6934.594	51.601	61.736	24.851
		(14.016)	(2178.962)	(9.357)	(19.773)	(3.862)
	5000	65.907	13620.114	127.566	170.774	63.108
		(7.435)	(1695.310)	(19.217)	(40.900)	(9.270)
14	200	4.693	992.536	3.811	3.323	1.755
		(2.292)	(179.904)	(0.836)	(0.925)	(0.427)
	500	9.200	2033.015	10.231	7.886	4.961
		(5.720)	(615.024)	(1.882)	(1.899)	(0.910)
	1000	14.267	3300.014	21.488	16.089	10.264
		(8.820)	(1161.112)	(3.771)	(2.727)	(1.762)
	2000	23.806	5368.852	43.440	32.736	21.371
		(9.888)	(1615.853)	(7.289)	(5.207)	(3.396)
	5000	54.881	11112.923	108.208	78.981	53.971
		(6.420)	(1067.874)	(15.308)	(13.124)	(7.959)
15	200	5.806	1152.649	4.250	5.207	1.889
		(2.094)	(118.483)	(0.963)	(1.471)	(0.348)
	500	14.336	2788.150	12.017	13.311	5.646
		(6.088)	(525.498)	(2.313)	(4.565)	(0.988)
	1000	26.980	5482.462	26.334	29.472	12.712
		(12.239)	(1302.016)	(4.836)	(8.758)	(1.995)
	2000	46.323	10059.626	56.555	65.201	26.470
		(24.921)	(2893.054)	(11.188)	(16.012)	(3.970)
	5000	84.802	19340.946	146.594	178.735	68.926
		(43.029)	(5942.179)	(29.312)	(34.232)	(9.475)
16	200	5.365	1048.559	3.846	3.437	1.840
		(2.159)	(150.968)	(0.805)	(1.003)	(0.407)
	500	12.492	2501.348	10.530	8.419	5.157
		(5.874)	(569.516)	(1.929)	(2.373)	(0.911)
	1000	23.116	4763.111	22.318	16.896	10.821
		(12.847)	(1315.694)	(3.835)	(3.745)	(1.769)
	2000	38.102	8567.409	47.326	33.447	22.814
		(22.548)	(2828.916)	(8.367)	(5.933)	(3.581)
	5000	70.070	16213.098	121.829	80.596	59.529
		(33.355)	(5296.358)	(21.552)	(13.490)	(8.610)

Table S.4: The mean (standard deviation) different estimators of the intercept parameter for scenario 6. The true value of β_0 was 0.42.

Method	n	Algorithms								
		Nelder-Mead	BFGS	L-BFGS-B	nlm	nlminb	ucminf	newuoa	bobyqa	nmkb
N	200	0.297	0.160	0.284	0.263	0.260	0.268	0.294	0.294	0.298
		(0.335)	(0.456)	(0.345)	(0.436)	(0.361)	(0.420)	(0.339)	(0.338)	(0.346)
	500	0.390	0.390	0.388	0.386	0.378	0.388	0.389	0.389	0.390
		(0.155)	(0.153)	(0.156)	(0.175)	(0.162)	(0.167)	(0.157)	(0.156)	(0.155)
	1000	0.412	0.381	0.412	0.412	0.409	0.412	0.412	0.412	0.412
		(0.063)	(0.188)	(0.063)	(0.063)	(0.063)	(0.063)	(0.063)	(0.063)	(0.063)
	2000	0.416	0.416	0.416	0.416	0.416	0.416	0.416	0.416	0.416
		(0.040)	(0.040)	(0.040)	(0.040)	(0.040)	(0.040)	(0.040)	(0.040)	(0.040)
	5000	0.419	0.419	0.419	0.419	0.419	0.419	0.419	0.419	0.419
		(0.024)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)
J	200	0.359	0.157	0.359	0.359	0.359	0.359	0.359	0.359	0.356
		(0.111)	(0.503)	(0.111)	(0.111)	(0.111)	(0.111)	(0.111)	(0.111)	(0.128)
	500	0.386	0.386	0.386	0.386	0.386	0.386	0.386	0.386	0.386
		(0.077)	(0.077)	(0.077)	(0.077)	(0.077)	(0.077)	(0.077)	(0.077)	(0.077)
	1000	0.400	0.338	0.400	0.400	0.400	0.400	0.400	0.400	0.400
		(0.055)	(0.289)	(0.055)	(0.055)	(0.055)	(0.055)	(0.055)	(0.055)	(0.055)
	2000	0.409	0.410	0.410	0.410	0.410	0.410	0.410	0.410	0.410
		(0.038)	(0.038)	(0.038)	(0.038)	(0.038)	(0.038)	(0.038)	(0.038)	(0.038)
	5000	0.416	0.416	0.416	0.416	0.416	0.416	0.416	0.416	0.416
		(0.023)	(0.023)	(0.023)	(0.023)	(0.023)	(0.023)	(0.023)	(0.023)	(0.023)
G	200	−0.372	−0.568	−0.370	−0.375	−0.413	−0.463	−0.405	−0.400	−0.334
		(0.392)	(0.292)	(0.413)	(0.410)	(0.387)	(0.365)	(0.365)	(0.372)	(0.418)
	500	−0.539	−0.576	−0.538	−0.549	−0.575	−0.595	−0.559	−0.554	−0.524
		(0.296)	(0.325)	(0.302)	(0.302)	(0.287)	(0.244)	(0.256)	(0.264)	(0.333)
	1000	−0.634	−0.692	−0.623	−0.634	−0.654	−0.653	−0.636	−0.632	−0.626
		(0.189)	(0.096)	(0.203)	(0.203)	(0.198)	(0.137)	(0.151)	(0.160)	(0.235)
	2000	−0.690	−0.694	−0.674	−0.685	−0.698	−0.685	−0.671	−0.671	−0.696
		(0.076)	(0.095)	(0.082)	(0.083)	(0.083)	(0.075)	(0.062)	(0.062)	(0.105)
	5000	−0.710	−0.708	−0.695	−0.689	−0.707	−0.696	−0.690	−0.690	−0.712
		(0.048)	(0.049)	(0.044)	(0.041)	(0.048)	(0.070)	(0.041)	(0.042)	(0.047)
C	200	0.220	0.219	0.220	0.220	0.220	0.220	0.220	0.220	0.220
		(0.238)	(0.241)	(0.238)	(0.238)	(0.238)	(0.238)	(0.238)	(0.238)	(0.238)
	500	0.339	0.340	0.339	0.339	0.339	0.339	0.339	0.339	0.339
		(0.142)	(0.141)	(0.142)	(0.142)	(0.142)	(0.142)	(0.142)	(0.142)	(0.142)
	1000	0.386	0.382	0.386	0.386	0.386	0.386	0.386	0.386	0.386
		(0.068)	(0.105)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)
	2000	0.404	0.404	0.404	0.404	0.404	0.404	0.404	0.404	0.404
		(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)
	5000	0.414	0.414	0.414	0.414	0.414	0.414	0.414	0.414	0.414
		(0.024)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)

Table S.5: The mean (standard deviation) of different estimators of the slope parameter for scenario 6. The true value of β_1 was 1.

Method	n	Algorithms								
		Nelder-Mead	BFGS	L-BFGS-B	nlm	nlminb	ucminf	newuoa	bobyqa	nmkb
N	200	1.131	1.206	1.144	1.100	1.168	1.106	1.135	1.135	1.121
		(0.337)	(0.353)	(0.350)	(0.281)	(0.360)	(0.291)	(0.343)	(0.342)	(0.325)
	500	1.040	1.040	1.041	1.040	1.055	1.039	1.040	1.041	1.039
		(0.183)	(0.182)	(0.183)	(0.177)	(0.187)	(0.178)	(0.182)	(0.182)	(0.181)
	1000	1.011	1.028	1.011	1.011	1.015	1.011	1.011	1.011	1.011
		(0.100)	(0.129)	(0.100)	(0.100)	(0.099)	(0.100)	(0.100)	(0.099)	(0.100)
	2000	1.006	1.006	1.006	1.006	1.007	1.006	1.006	1.006	1.006
		(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)
	5000	1.002	1.002	1.002	1.002	1.002	1.002	1.002	1.002	1.002
		(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)
J	200	1.095	1.127	1.095	1.095	1.095	1.095	1.095	1.095	1.095
		(0.196)	(0.191)	(0.196)	(0.196)	(0.196)	(0.196)	(0.196)	(0.196)	(0.196)
	500	1.054	1.054	1.054	1.054	1.053	1.054	1.054	1.054	1.054
		(0.126)	(0.126)	(0.126)	(0.126)	(0.125)	(0.126)	(0.126)	(0.126)	(0.126)
	1000	1.028	1.044	1.027	1.027	1.027	1.027	1.027	1.027	1.027
		(0.089)	(0.105)	(0.089)	(0.089)	(0.089)	(0.089)	(0.089)	(0.089)	(0.089)
	2000	1.016	1.016	1.016	1.016	1.016	1.016	1.016	1.016	1.016
		(0.065)	(0.065)	(0.065)	(0.065)	(0.065)	(0.065)	(0.065)	(0.065)	(0.065)
	5000	1.006	1.006	1.005	1.005	1.005	1.005	1.005	1.005	1.005
		(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)
G	200	1.883	1.998	1.869	1.873	1.900	1.932	1.911	1.909	1.839
		(0.578)	(0.471)	(0.591)	(0.586)	(0.562)	(0.531)	(0.553)	(0.559)	(0.622)
	500	1.975	1.965	1.969	1.974	1.985	2.007	1.995	1.991	1.943
		(0.355)	(0.372)	(0.357)	(0.354)	(0.335)	(0.296)	(0.325)	(0.330)	(0.415)
	1000	2.011	2.033	2.005	2.007	2.010	2.025	2.020	2.018	1.991
		(0.224)	(0.167)	(0.235)	(0.232)	(0.227)	(0.184)	(0.202)	(0.208)	(0.278)
	2000	2.036	2.034	2.036	2.035	2.036	2.036	2.037	2.036	2.032
		(0.124)	(0.129)	(0.127)	(0.127)	(0.126)	(0.123)	(0.122)	(0.121)	(0.141)
	5000	2.039	2.039	2.039	2.039	2.039	2.036	2.039	2.039	2.039
		(0.078)	(0.078)	(0.079)	(0.078)	(0.078)	(0.083)	(0.079)	(0.078)	(0.078)
C	200	1.259	1.259	1.259	1.259	1.259	1.259	1.259	1.259	1.259
		(0.293)	(0.292)	(0.293)	(0.293)	(0.293)	(0.293)	(0.293)	(0.293)	(0.293)
	500	1.117	1.117	1.117	1.117	1.117	1.117	1.117	1.117	1.117
		(0.176)	(0.176)	(0.176)	(0.176)	(0.176)	(0.176)	(0.176)	(0.176)	(0.176)
	1000	1.051	1.052	1.051	1.051	1.051	1.051	1.051	1.051	1.051
		(0.101)	(0.101)	(0.101)	(0.101)	(0.101)	(0.101)	(0.101)	(0.101)	(0.101)
	2000	1.027	1.026	1.026	1.026	1.026	1.026	1.026	1.026	1.026
		(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)	(0.068)
	5000	1.009	1.010	1.009	1.009	1.009	1.009	1.009	1.009	1.009
		(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)	(0.041)

Table S.6: The mean (standard deviation) of different estimators for the skewness parameter for scenario 6. The true value of δ was 4.

Method	n	Nelder-Mead	BFGS	L-BFGS-B	nlm	nlminb	ucminf	newuoa	bobyqa	nmkb
		Algorithms								
N	200	957.4	14.95	617.4	1501.5	410.8	1435.9	23.1	13.6	1201.8
		(2663.5)	(18.56)	(1378.1)	(6234.9)	(988.2)	(4619.1)	(23.2)	(13.3)	(3780.9)
	500	621.6	13.31	495.1	1234.0	251.5	1680.3	12.40	8.161	1273.1
		(1982.5)	(17.77)	(1360.9)	(4122.0)	(904.4)	(6757.5)	(16.04)	(8.153)	(4749.7)
	1000	166.9	7.179	171.2	433.5	73.91	634.7	6.826	5.671	470.9
		(802.0)	(11.02)	(851.6)	(2641.5)	(650.1)	(3935.1)	(8.663)	(4.659)	(2983.8)
	2000	23.38	4.857	22.71	92.55	4.467	106.1	4.758	4.580	88.73
		(219.6)	(4.875)	(214.1)	(1390.3)	(1.847)	(1445.1)	(3.722)	(2.277)	(1313.0)
	5000	4.181	4.181	4.185	4.185	4.185	4.185	4.185	4.185	4.185
		(0.868)	(0.866)	(0.866)	(0.866)	(0.866)	(0.866)	(0.866)	(0.866)	(0.866)
J	200	3.014	2.367	3.014	3.014	3.014	3.014	3.014	3.014	3.007
		(1.109)	(1.778)	(1.109)	(1.109)	(1.109)	(1.109)	(1.109)	(1.109)	(1.127)
	500	3.628	3.626	3.629	3.629	3.631	3.629	3.629	3.629	3.629
		(1.432)	(1.423)	(1.431)	(1.431)	(1.431)	(1.431)	(1.431)	(1.431)	(1.431)
	1000	3.916	3.667	3.918	3.918	3.918	3.918	3.918	3.918	3.918
		(1.484)	(1.922)	(1.483)	(1.483)	(1.483)	(1.483)	(1.483)	(1.483)	(1.483)
	2000	4.012	4.015	4.015	4.015	4.015	4.015	4.015	4.015	4.015
		(1.269)	(1.265)	(1.267)	(1.268)	(1.268)	(1.268)	(1.268)	(1.265)	(1.268)
	5000	4.034	4.034	4.037	4.037	4.037	4.037	4.037	4.037	4.037
		(0.771)	(0.770)	(0.769)	(0.769)	(0.769)	(0.769)	(0.769)	(0.769)	(0.769)
G	200	1.269	0.273	1.365	1.308	1.013	0.760	1.028	1.067	1.675
		(2.381)	(0.907)	(2.510)	(2.434)	(2.089)	(1.802)	(2.149)	(2.186)	(2.864)
	500	0.614	0.606	0.629	0.592	0.474	0.283	0.450	0.472	0.913
		(1.943)	(1.978)	(1.954)	(1.881)	(1.654)	(1.133)	(1.568)	(1.581)	(2.650)
	1000	0.214	0.025	0.249	0.229	0.189	0.087	0.167	0.177	0.367
		(0.996)	(0.072)	(1.062)	(1.048)	(0.993)	(0.434)	(0.839)	(0.793)	(1.628)
	2000	0.030	0.031	0.055	0.039	0.022	0.032	0.051	0.052	0.042
		(0.131)	(0.230)	(0.183)	(0.185)	(0.166)	(0.043)	(0.018)	(0.015)	(0.408)
	5000	0.005	0.007	0.023	0.031	0.009	0.015	0.030	0.029	0.002
		(0.034)	(0.033)	(0.025)	(0.015)	(0.033)	(0.030)	(0.017)	(0.017)	(0.034)
C	200	2.121	2.116	2.120	2.120	2.120	2.120	2.120	2.120	2.120
		(1.170)	(1.179)	(1.170)	(1.170)	(1.170)	(1.170)	(1.170)	(1.170)	(1.170)
	500	3.049	3.050	3.050	3.050	3.050	3.050	3.050	3.050	3.050
		(1.349)	(1.346)	(1.350)	(1.350)	(1.350)	(1.350)	(1.350)	(1.350)	(1.350)
	1000	3.573	3.544	3.575	3.575	3.575	3.575	3.575	3.575	3.575
		(1.336)	(1.508)	(1.337)	(1.337)	(1.337)	(1.337)	(1.337)	(1.337)	(1.337)
	2000	3.829	3.832	3.832	3.832	3.832	3.832	3.832	3.832	3.832
		(1.181)	(1.175)	(1.178)	(1.179)	(1.179)	(1.179)	(1.179)	(1.178)	(1.179)
	5000	3.964	3.962	3.966	3.966	3.966	3.966	3.966	3.966	3.966
		(0.750)	(0.750)	(0.749)	(0.749)	(0.749)	(0.749)	(0.749)	(0.749)	(0.749)

Table S.7: The mean (standard deviation) computation time for different algorithms for scenario 6.

Method	n	Algorithms								
		Nelder-Mead	BFGS	L-BFGS-B	nlm	nlminb	ucminf	newuoa	bobyqa	nmkb
N	200	4.371	10.519	4.977	3.216	2.614	3.222	10.800	12.020	3.255
		(1.745)	(13.436)	(2.849)	(1.757)	(1.556)	(1.882)	(7.661)	(6.781)	(1.305)
	500	9.564	12.542	9.744	5.813	4.811	6.031	18.624	24.061	7.086
		(3.704)	(16.892)	(5.968)	(3.216)	(2.712)	(3.987)	(15.614)	(14.605)	(2.915)
	1000	16.574	18.559	15.070	9.401	8.275	9.162	26.292	40.406	12.392
		(5.032)	(11.699)	(6.949)	(3.713)	(2.721)	(4.799)	(20.550)	(23.810)	(3.866)
	2000	30.622	25.677	26.168	16.709	15.805	16.368	42.238	67.987	23.118
		(6.907)	(9.634)	(5.695)	(3.317)	(2.063)	(4.428)	(22.018)	(32.215)	(4.754)
	5000	75.786	76.747	62.461	40.750	41.300	39.844	95.604	155.691	56.696
		(13.079)	(8.064)	(6.073)	(2.904)	(4.198)	(5.254)	(24.941)	(48.748)	(8.146)
J	200	5.484	4.658	4.917	3.114	2.507	2.700	5.432	8.168	4.192
		(1.252)	(1.269)	(0.618)	(0.417)	(0.370)	(0.681)	(1.727)	(2.852)	(0.814)
	500	14.706	12.968	12.717	8.043	6.982	7.426	16.440	26.442	11.132
		(3.036)	(2.606)	(1.694)	(0.812)	(0.787)	(1.458)	(5.461)	(10.484)	(1.889)
	1000	29.720	28.047	25.102	16.322	14.579	15.117	34.875	57.131	22.439
		(6.065)	(5.532)	(3.272)	(1.558)	(1.945)	(2.768)	(12.482)	(24.220)	(3.737)
	2000	58.649	49.735	50.072	32.261	30.666	31.335	72.353	119.231	45.196
		(11.485)	(8.960)	(5.685)	(2.549)	(3.742)	(5.750)	(23.916)	(46.842)	(7.007)
	5000	149.114	151.127	123.431	81.225	81.807	80.770	182.776	301.441	113.208
		(25.890)	(17.377)	(11.495)	(5.605)	(8.481)	(12.189)	(46.677)	(88.544)	(15.869)
G	200	4.248	2.522	4.747	2.888	1.901	2.276	3.141	4.380	3.382
		(1.337)	(0.688)	(0.930)	(1.557)	(0.603)	(0.637)	(3.009)	(5.178)	(1.056)
	500	9.530	11.293	11.434	7.289	4.805	5.402	6.317	8.239	8.746
		(2.874)	(3.109)	(2.022)	(4.439)	(1.178)	(1.361)	(7.023)	(10.462)	(2.732)
	1000	19.699	14.950	23.800	14.567	10.621	11.777	11.657	14.560	19.836
		(4.766)	(2.316)	(3.471)	(7.786)	(2.090)	(2.140)	(7.742)	(10.265)	(5.441)
	2000	39.653	56.040	49.089	30.837	22.576	25.290	22.735	27.740	43.919
		(7.725)	(8.001)	(7.349)	(18.046)	(4.311)	(4.533)	(3.821)	(4.557)	(9.834)
	5000	94.046	84.292	120.613	77.704	59.130	57.785	56.831	66.628	116.309
		(14.596)	(12.960)	(16.258)	(17.883)	(10.587)	(10.888)	(9.852)	(10.132)	(18.497)
C	200	2.730	2.488	2.464	1.573	1.225	1.239	2.565	3.869	2.070
		(0.747)	(1.240)	(0.402)	(0.229)	(0.256)	(0.343)	(0.818)	(1.338)	(0.499)
	500	7.443	6.877	6.418	4.041	3.486	3.658	7.884	12.353	5.658
		(1.543)	(1.532)	(0.830)	(0.426)	(0.458)	(0.707)	(2.392)	(4.240)	(1.040)
	1000	14.627	14.292	12.528	8.165	7.388	7.566	16.738	27.515	11.440
		(3.117)	(4.720)	(1.536)	(0.741)	(0.935)	(1.278)	(5.502)	(10.548)	(1.903)
	2000	29.376	25.410	24.648	15.997	15.230	15.468	34.868	57.615	22.822
		(5.769)	(4.800)	(2.952)	(1.249)	(1.887)	(2.571)	(11.061)	(22.363)	(3.746)
	5000	74.426	75.470	61.695	40.610	41.032	39.939	92.239	149.836	57.068
		(12.813)	(8.667)	(5.813)	(2.885)	(4.374)	(5.981)	(23.903)	(43.749)	(8.381)

Table S.8: The number of non-convergent datasets for different algorithms for scenario 6.

Method	n	Algorithms								
		Nelder-Mead	BFGS	L-BFGS-B	nlm	nlminb	ucminf	newuoa	bobyqa	nmkb
N	200	1	3	70	3	176	0	0	0	0
	500	1	0	13	9	114	0	0	0	0
	1000	1	0	0	3	40	0	0	0	0
	2000	0	0	0	0	9	0	0	0	0
	5000	0	0	0	0	0	0	0	0	0
J	200	0	0	0	0	0	0	0	0	0
	500	0	0	0	0	1	0	0	0	0
	1000	0	0	0	0	0	0	0	0	0
	2000	0	0	0	0	0	0	0	0	0
	5000	0	0	0	0	0	0	0	0	0
G	200	0	0	18	0	0	0	0	0	3
	500	0	0	23	0	3	0	0	0	3
	1000	2	0	27	0	2	0	0	0	4
	2000	3	0	27	0	2	0	0	0	1
	5000	3	0	25	0	7	0	0	0	2
C	200	0	0	0	0	0	0	0	0	0
	500	0	0	0	0	0	0	0	0	0
	1000	0	0	0	0	0	0	0	0	0
	2000	0	0	0	0	0	0	0	0	0
	5000	0	0	0	0	0	0	0	0	0

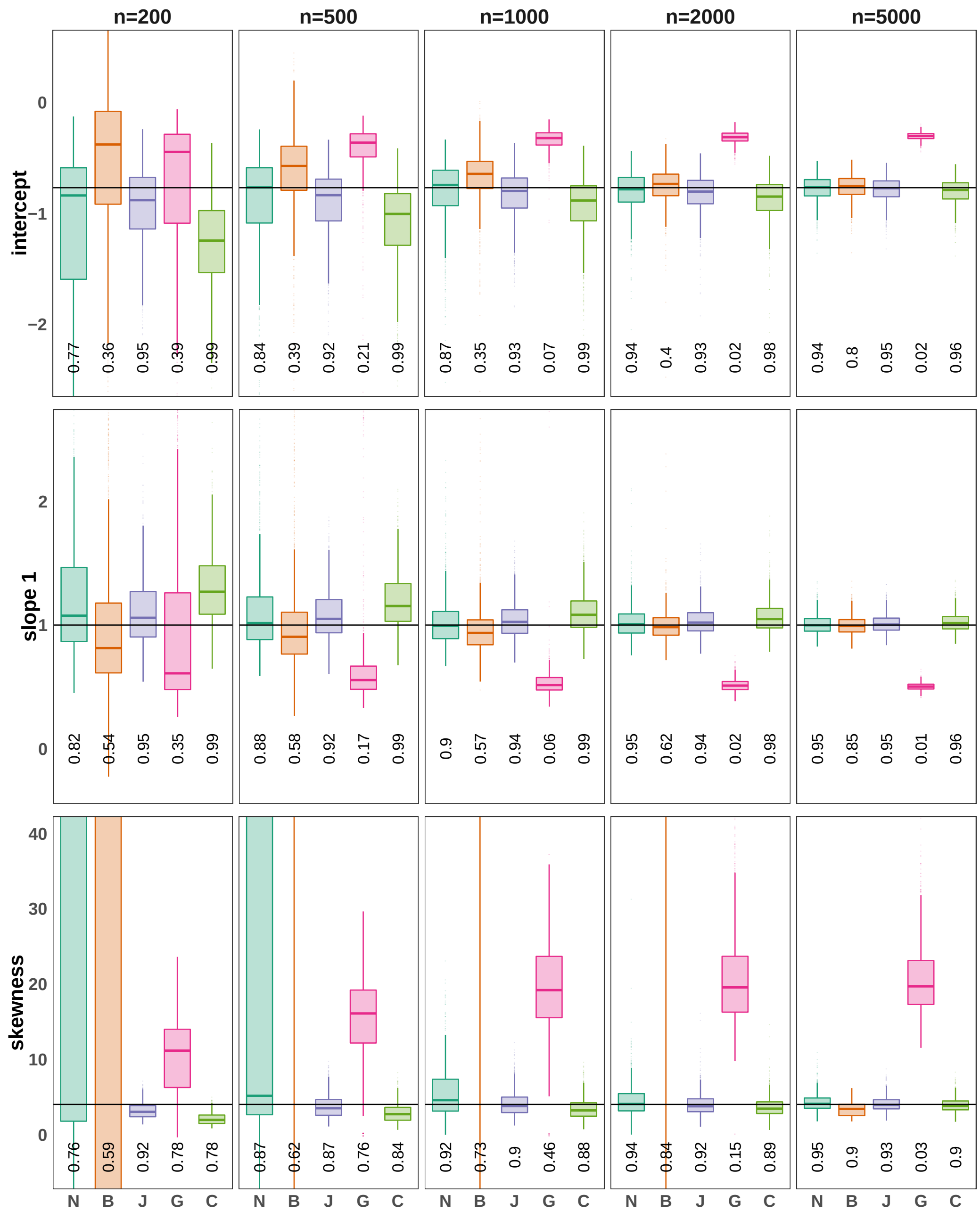

Figure S.1: Simulation results based on 1000 replications when $X \sim \text{Normal}(0, (\sqrt{4/3})^2)$, $\delta = 4$, $\beta_0 = -0.77$, $\beta_1 = 1$, and $p_m = 12\%$ The numbers in the boxplots are the empirical coverage probabilities for the nominal level 0.95 based on the standard error derived from the Fisher information matrix. The horizontal line in each figure indicates the true value of the parameter. N: Naive MLE, B: Bootstrap bias correction, J: Penalized likelihood estimation with Jeffrey's prior, G: Penalized likelihood estimation with generalized information matrix, C: Penalized likelihood estimation with Cauchy distribution.

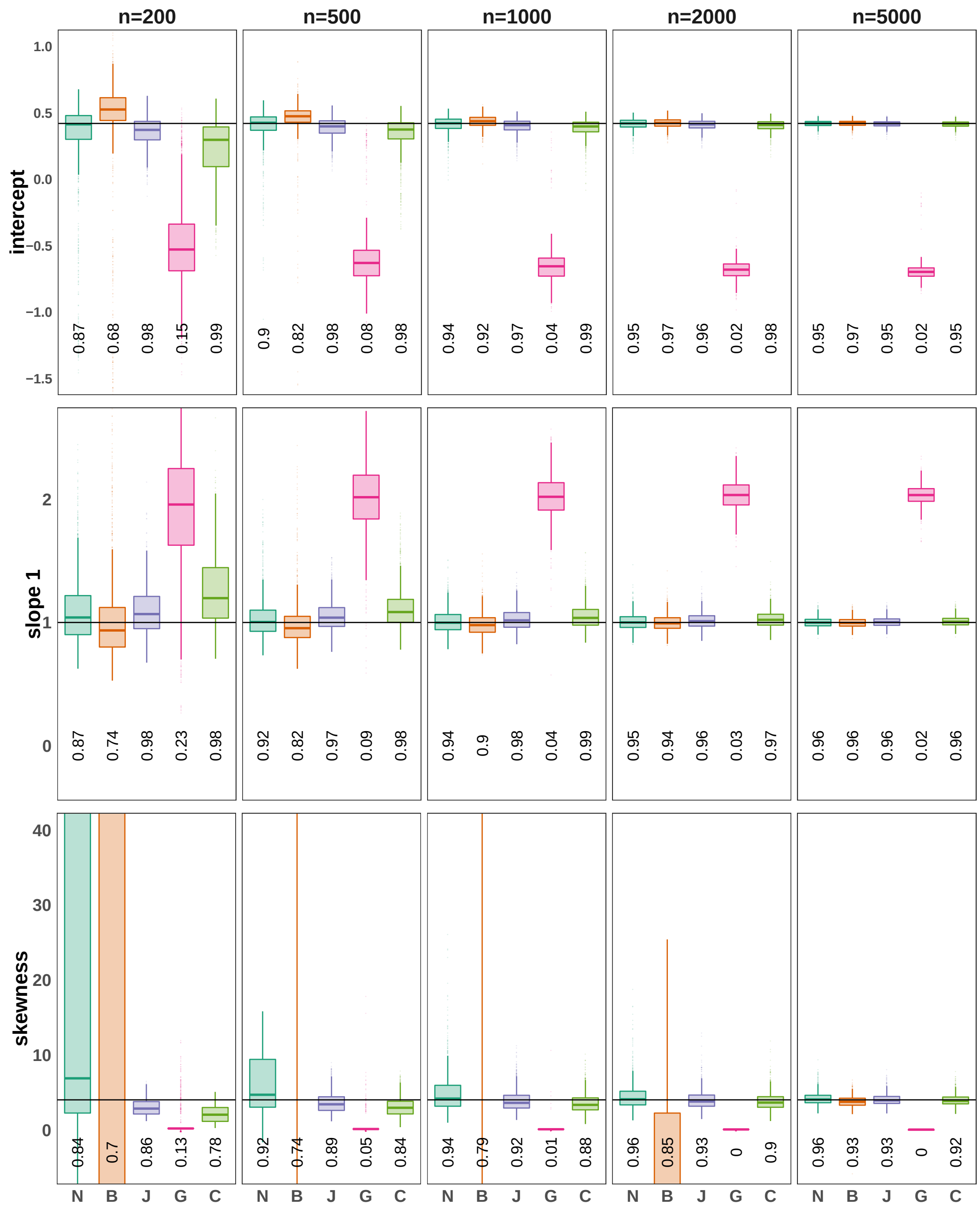

Figure S.2: Simulation results based on 1000 replications when $X \sim \text{Normal}(0, (\sqrt{4/3})^2)$, $\delta = 4$, $\beta_0 = 0.42$, $\beta_1 = 1$, and $p_m = 40\%$. The numbers in the boxplots are the empirical coverage probabilities for the nominal level 0.95 based on the standard error derived from the Fisher information matrix. The horizontal line in each figure indicates the true value of the parameter. N: Naive MLE, B: Bootstrap bias correction, J: Penalized likelihood estimation with Jeffrey's prior, G: Penalized likelihood estimation with generalized information matrix, C: Penalized likelihood estimation with Cauchy distribution.

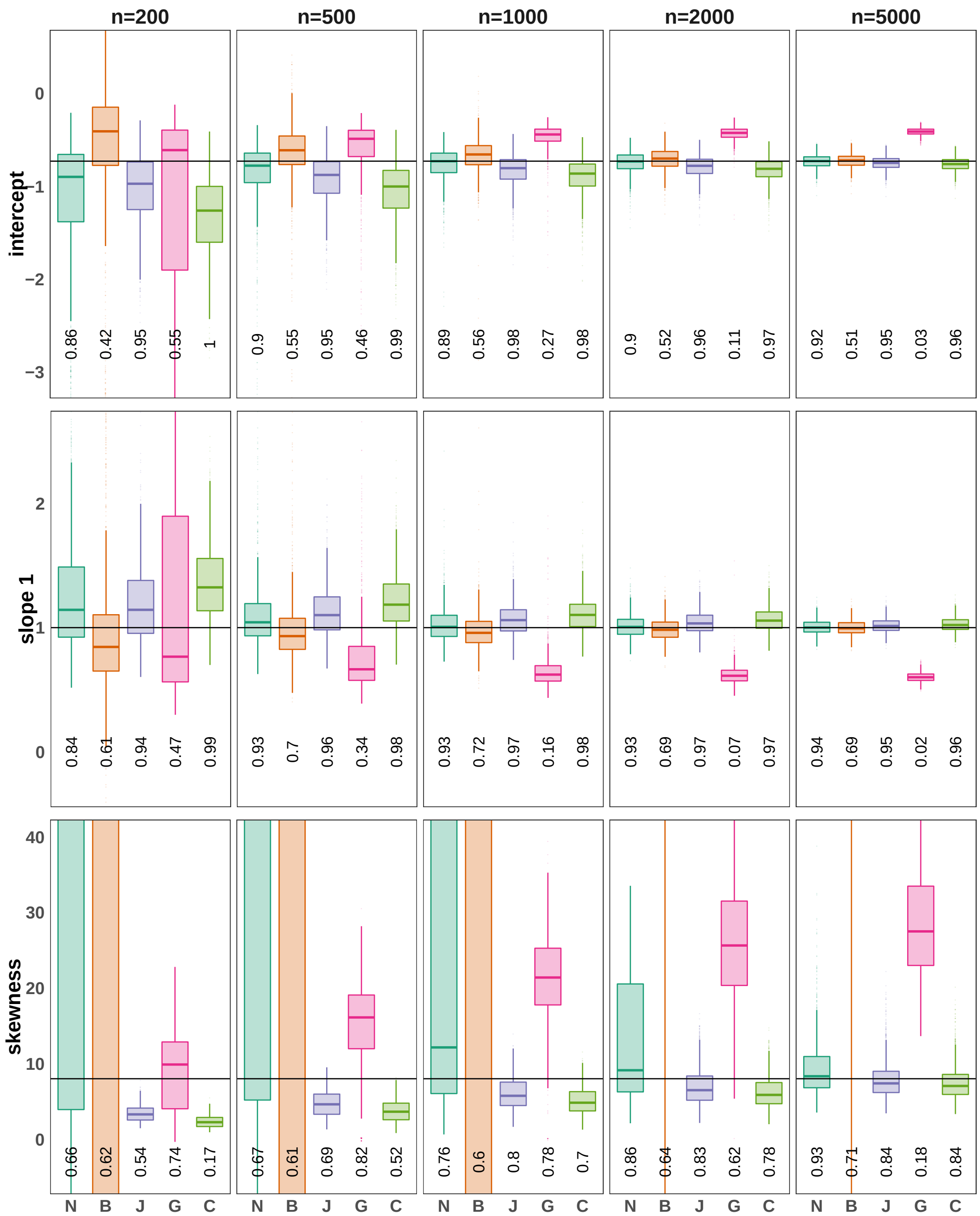

Figure S.3: Simulation results based on 1000 replications when $X \sim \text{Normal}(0, (\sqrt{4/3})^2)$, $\delta = 8$, $\beta_0 = -0.73$, $\beta_1 = 1$, and $p_m = 12\%$. The numbers in the boxplots are the empirical coverage probabilities for the nominal level 0.95 based on the standard error derived from the Fisher information matrix. The horizontal line in each figure indicates the true value of the parameter. N: Naive MLE, B: Bootstrap bias correction, J: Penalized likelihood estimation with Jeffrey's prior, G: Penalized likelihood estimation with generalized information matrix, C: Penalized likelihood estimation with Cauchy distribution.

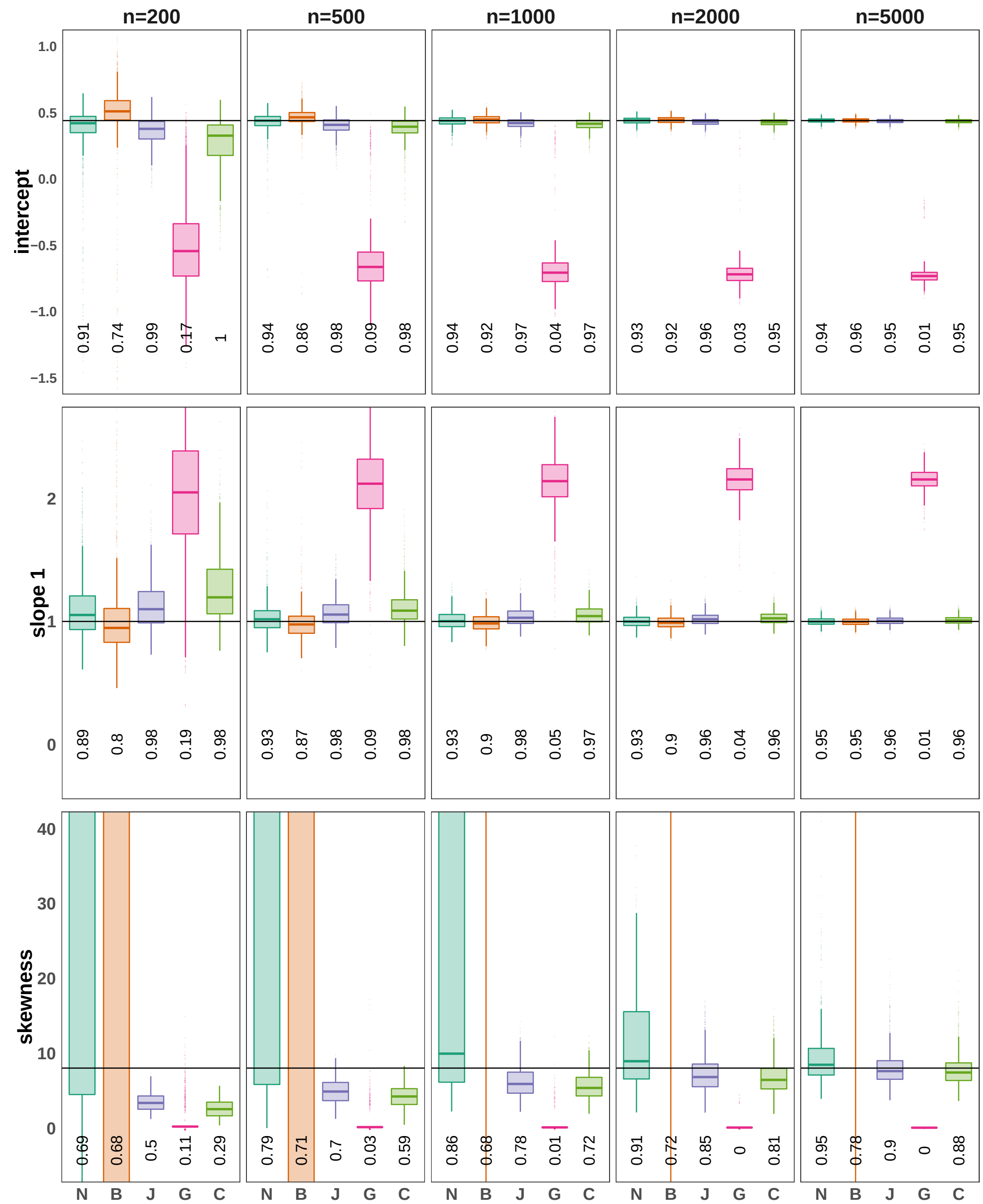

Figure S.4: Simulation results based on 1000 replications when $X \sim \text{Normal}(0, (\sqrt{4/3})^2)$, $\delta = 8$, $\beta_0 = 0.44$, $\beta_1 = 1$, and $p_m = 40\%$. The numbers in the boxplots are the empirical coverage probabilities for the nominal level 0.95 based on the standard error derived from the Fisher information matrix. The horizontal line in each figure indicates the true value of the parameter. N: Naive MLE, B: Bootstrap bias correction, J: Penalized likelihood estimation with Jeffrey's prior, G: Penalized likelihood estimation with generalized information matrix, C: Penalized likelihood estimation with Cauchy distribution.

Figure S.5: Simulation results based on 1000 replications when $X \sim 0.5\text{Normal}(-1, (\sqrt{1/3})^2) + 0.5\text{Normal}(1, (\sqrt{1/3})^2)$, $\delta = 4$, $\beta_0 = -0.85$, $\beta_1 = 1$, and $p_m = 12\%$ The numbers in the boxplots are the empirical coverage probabilities for the nominal level 0.95 based on the standard error derived from the Fisher information matrix. The horizontal line in each figure indicates the true value of the parameter. N: Naive MLE, B: Bootstrap bias correction, J: Penalized likelihood estimation with Jeffrey's prior, G: Penalized likelihood estimation with generalized information matrix, C: Penalized likelihood estimation with Cauchy distribution.

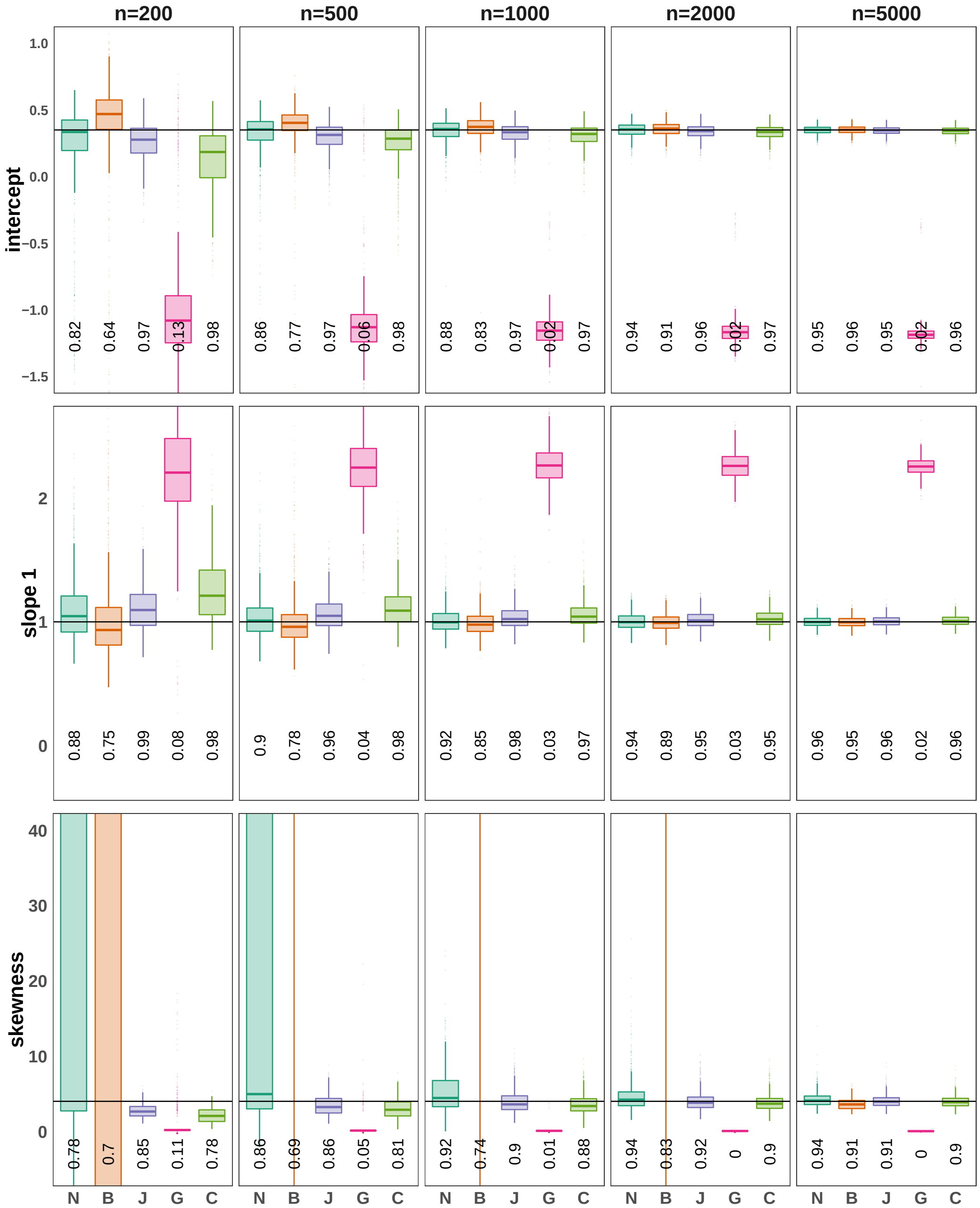

Figure S.6: Simulation results based on 1000 replications when $X \sim 0.5\text{Normal}(-1, (\sqrt{1/3})^2) + 0.5\text{Normal}(1, (\sqrt{1/3})^2)$, $\delta = 4$, $\beta_0 = 0.35$, $\beta_1 = 1$, and $p_m = 40\%$. The numbers in the boxplots are the empirical coverage probabilities for the nominal level 0.95 based on the standard error derived from the Fisher information matrix. The horizontal line in each figure indicates the true value of the parameter. N: Naive MLE, B: Bootstrap bias correction, J: Penalized likelihood estimation with Jeffrey's prior, G: Penalized likelihood estimation with generalized information matrix, C: Penalized likelihood estimation with Cauchy distribution.

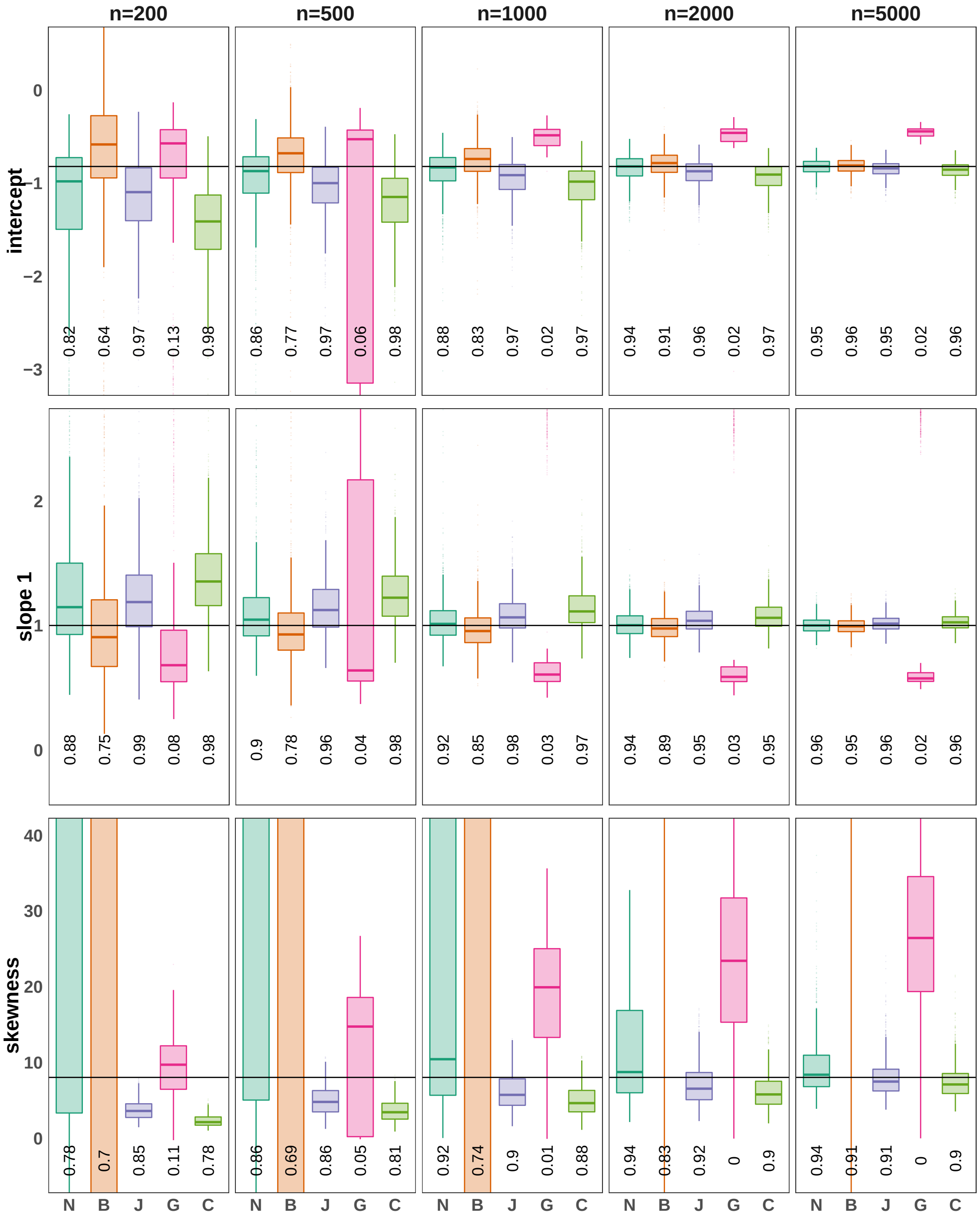

Figure S.7: Simulation results based on 1000 replications when $X \sim 0.5\text{Normal}(-1, (\sqrt{1/3})^2) + 0.5\text{Normal}(1, (\sqrt{1/3})^2)$, $\delta = 8$, $\beta_0 = -0.82$, $\beta_1 = 1$, and $p_m = 12\%$. The numbers in the boxplots are the empirical coverage probabilities for the nominal level 0.95 based on the standard error derived from the Fisher information matrix. The horizontal line in each figure indicates the true value of the parameter. N: Naive MLE, B: Bootstrap bias correction, J: Penalized likelihood estimation with Jeffrey's prior, G: Penalized likelihood estimation with generalized information matrix, C: Penalized likelihood estimation with Cauchy distribution.

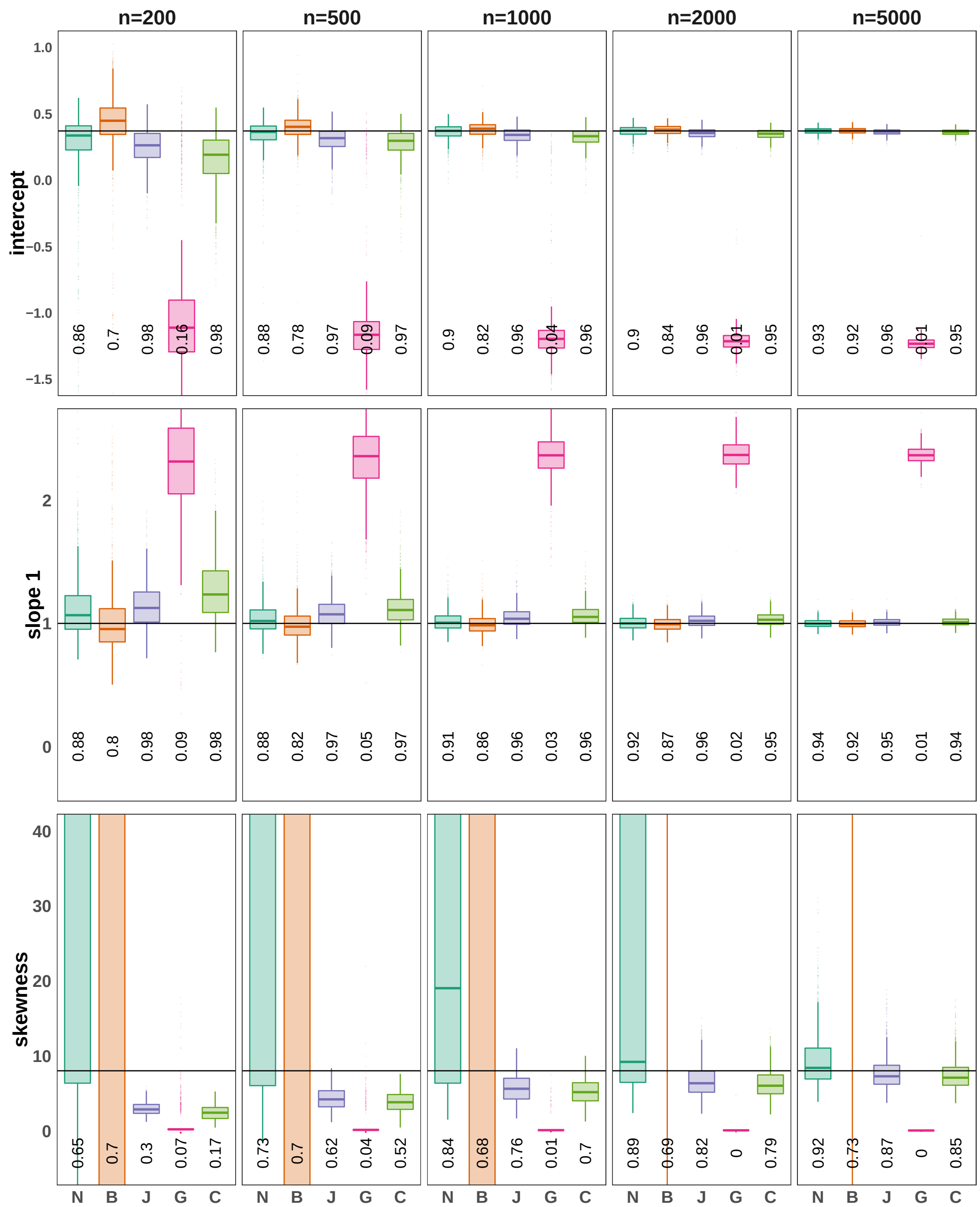

Figure S.8: Simulation results based on 1000 replications when $X \sim 0.5\text{Normal}(-1, (\sqrt{1/3})^2) + 0.5\text{Normal}(1, (\sqrt{1/3})^2)$, $\delta = 8$, $\beta_0 = 0.37$, $\beta_1 = 1$, and $p_m = 40\%$. The numbers in the boxplots are the empirical coverage probabilities for the nominal level 0.95 based on the standard error derived from the Fisher information matrix. The horizontal line in each figure indicates the true value of the parameter. N: Naive MLE, B: Bootstrap bias correction, J: Penalized likelihood estimation with Jeffrey's prior, G: Penalized likelihood estimation with generalized information matrix, C: Penalized likelihood estimation with Cauchy distribution.