Scientia Iranica

A simulation study: Robust ratio double sampling estimator of finite population mean in the presence of outliers

Tolga Zaman[1*], Hasan Bulut[2]

[1]Çankırı Karatekin University, Faculty of Science, Department of Statistics, 18100 Çankırı, Turkey

[2]Ondokuz Mayıs University, Faculty of Science, Department of Statistics, 55139 Samsun, Turkey

Email Corresponding Author: zamantolga@gmail.com (Zaman, T). Cell: 903762189540

Email: hasan.bulut@omu.edu.tr (Bulut, H).

APPENDIX

We give our R codes for simulation better understanding as below:

```
library(e1071)
library(MASS)
library(leaps)
library(robust)
library(robustbase)
library(L1pack)
t1<-function(b.est,y2,x2,x1){
  num<-y2+b.est*(x1-x2)
  den<-x2
  return((num/den)*x1)}
t2<-function(b.est,y2,x2,x1,Cx){
  num<-y2+b.est*(x1-x2)
  den<-x2+Cx
  return((num/den)*(x1+Cx))}
t3<-function(b.est,y2,x2,x1,Beta2){
  num<-y2+b.est*(x1-x2)
  den<-x2+Beta2
  return((num/den)*(x1+Beta2))}
N=150 ; n1<-30 ; n2<-10
alpha=0.1
N.clean<-N*(1-alpha)
N.out<-N*alpha
MSE1.hubM<-MSE1.hubMM<-MSE1.lts<-MSE1.lms<-0
MSE2.hubM<-MSE2.hubMM<-MSE2.lts<-MSE2.lms<-0
MSE3.hubM<-MSE3.hubMM<-MSE3.lts<-MSE3.lms<-0
m=0; ITER=10000
for (h in 1:ITER) {
  X.clean<-rnorm(N.clean)
  X.out<-rnorm(N.out,mean=25, sd=1)
  X<-c(X.clean,X.out)
  eps.clean<-rnorm(N.clean)
  eps.out<-rnorm(N.out,mean=25, sd=1)
  eps<-c(eps.clean,eps.out)
  Y<-2+3*X+eps
  DATA<-cbind(Y,X)
  sample1<-c(sample(1:N.clean,n1*(1-alpha),replace=FALSE),
         sample((N.clean+1):N,n1*alpha,replace=FALSE))
  y1data<-DATA[sample1,1]
  x1data<-DATA[sample1,2]
  x1<-mean(x1data);y1<-mean(y1data)
  Cx<-sd(x1data)/x1
  Beta2<-kurtosis(x1data)
  MUy<-y1
  mod.rlmHub<-rlm(y1data~x1data,psi = psi.huber,maxit = 50)
  b.hubM<-mod.rlmHub$coefficients[2]
  mod.lmRob<-lmRob(y1data~x1data)
  b.lmRob<-mod.lmRob$coefficients[2]
  mod.lts<-lqs(y1data~x1data,method = "lts")
  b.lts<-mod.lts$coefficients[2]
  mod.lms<-lqs(y1data~x1data,method = "lms")
  b.lms<-mod.lms$coefficients[2]
  BETAS<-c(b.hubM,b.lmRob,b.lts,b.lms)
  n1.clean<-n1*(1-alpha)
  sample2<-c(sample(1:n1.clean,n2*(1-alpha),replace=FALSE),
         sample((n1.clean+1):n1,n2*alpha,replace=FALSE))
  y2data<-y1data[sample2]
  x2data<-x1data[sample2]
  y2<-mean(y2data); x2<-mean(x2data)
  T1.hubM<-t1(BETAS[1],y2,x2,x1)
  T2.hubM<-t2(BETAS[1],y2,x2,x1,Cx)
  T3.hubM<-t3(BETAS[1],y2,x2,x1,Beta2)
  T1.hubMM<-t1(BETAS[2],y2,x2,x1)
  T2.hubMM<-t2(BETAS[2],y2,x2,x1,Cx)
  T3.hubMM<-t3(BETAS[2],y2,x2,x1,Beta2)
  T1.lts<-t1(BETAS[3],y2,x2,x1)
  T2.lts<-t2(BETAS[3],y2,x2,x1,Cx)
  T3.lts<-t3(BETAS[3],y2,x2,x1,Beta2)
  T1.lms<-t1(BETAS[4],y2,x2,x1)
  T2.lms<-t2(BETAS[4],y2,x2,x1,Cx)
  T3.lms<-t3(BETAS[4],y2,x2,x1,Beta2)
  MSE1.hubM<-MSE1.hubM+(T1.hubM-MUy)^2
  MSE2.hubM<-MSE1.hubM+(T2.hubM-MUy)^2
  MSE3.hubM<-MSE1.hubM+(T3.hubM-MUy)^2
  MSE1.hubMM<-MSE1.hubMM+(T1.hubMM-MUy)^2
  MSE2.hubMM<-MSE1.hubMM+(T2.hubMM-MUy)^2
  MSE3.hubMM<-MSE1.hubMM+(T3.hubMM-MUy)^2
  MSE1.lts<-MSE1.lts+(T1.lts-MUy)^2
  MSE2.lts<-MSE1.lts+(T2.lts-MUy)^2
  MSE3.lts<-MSE1.lts+(T3.lts-MUy)^2
  MSE1.lms<-MSE1.lms+(T1.lms-MUy)^2
  MSE2.lms<-MSE1.lms+(T2.lms-MUy)^2
  MSE3.lms<-MSE1.lms+(T3.lms-MUy)^2
}
MSE.hubM<-c(MSE1.hubM,MSE2.hubM,MSE3.hubM)/ITER
MSE.hubMM<-c(MSE1.hubMM,MSE2.hubMM,MSE3.hubMM)/ITER
MSE.lts<-c(MSE1.lts,MSE2.lts,MSE3.lts)/ITER
MSE.lms<-c(MSE1.lms,MSE2.lms,MSE3.lms)/ITER
MSE.result<-cbind(MSE.hubM,MSE.hubMM,MSE.lts,MSE.lms)
colnames(MSE.result)<-c("HuberM","HuberMM","LTS","LMS")
```

```
rownames(MSE.result)<-c("Est1","Est2","Est3")
MSE.result
```