Pak.j.stat.oper.res. Vol.21 No. 2 2025 pp 163-176 DOI: https://doi.org/10.18187/pjsor.v21i2.4872

Pakistan Journal of Statistics and Operation Research

New highly efficient one and two-stage ranked set sampling variations

Mohammed Obeidat[1*], Rahaf Na'amneh[2], Ahmad Hanandeh[3], Aymen Rawashdeh[4]

* Corresponding Author

1. Department of Statistics, Yarmouk University, Irbid, Jordan. E-mail: mohammad.obidat@yu.edu.jo
2. Department of Statistics, Yarmouk University, Irbid, Jordan. E-mail: 2023117002@ses.yu.edu.jo
3. Department of Mathematics, Faculty of Science, Islamic University of Madinah, Saudi Arabia, Email: hanandeh@iu.edu.sa
4. Department of Statistics, Yarmouk University, Irbid, Jordan. E-mail: ayman.r@yu.edu.jo

Abstract

In this paper, we proposed highly efficient ranked set sampling schemes to estimate the population mean. First, we proposed a new single-stage sampling scheme which we called new neoteric ranked set sampling. Second, we proposed a two-stage methods based on the systematic ranked set sampling and the new neoteric ranked set sampling. The performance of the proposed methods is compared with that of competitive two-stage methods through a Monte Carlo simulation study using various popular symmetric and asymmetric statistical distributions. The results show that the newly proposed methods are more efficient in estimating the population mean than the existing methods. The proposed methods are illustrated on data of the diameter and height of pine trees.

Key Words: Two-stage, ranked set sampling, double ranked set sampling, median ranked set sampling, extreme ranked set sampling, neoteric ranked set sampling, Monte Carlo simulation.

Mathematical Subject Classification: 62D05, 62G32, 62G0765C05

1. Introduction

Ranked set sampling (RSS) and any of its variations are sampling techniques that are useful when ranking of units can be done easily and precisely, either visually or at negligible cost using the character of interest. RSS was first proposed by McIntyre (1952) as a sampling method to estimate mean pasture and forage yields in agricultural experimentation. Statistical theory of RSS was then established by Takahasi and Wakimoto (1968) and Stokes (1980). The procedure of RSS can be summarized as follows: m sets of size m units each are drawn from the population. The m units of each set are ranked visually without actually quantifying them. From the i^{th} set select the i^{th} ordered unit for actual measurement (i=1,2,…,m). This process may be repeated r times to obtain a RSS of size $n=mr$. See Chen et al. (2003) for more information on the theory and applications of RSS and related sampling schemes.

Several variations of the RSS that rely on the concept of ranking without actual measurement were proposed in literature for estimating the population mean. Muttlak (1996) introduced pair ranked set sampling (*PRSS*), the extreme ranked set sampling (*ERSS*) was proposed by Samawi et al. (1996), and Muttlak (1997) suggested the median ranked set sampling (*MRSS*). Recently Zamanzade and Al-Omari (2016) proposed neoteric ranked set sampling (NRSS). In a similar fashion to Zamanzade and Al-Omari, Khan et al. (2019) introduced two new sampling schemes, the centralized ranked set sampling (CRSS) and the systematic ranked set sampling (SRSS). Recently, Taconeli (2024) proposed Dual-rank ranked set sampling.

The idea of two-stage sampling methods which combines two RSS variations was first introduced by Al-Saleh and Al-Kaddiri (2000) who introduced the concept of Double Ranked Set Sampling (*DRSS*). Samawi (2002) proposed double ERSS (DERSS), and Samawi and Tawalbeh (2002) introduced the double MRSS (DMRSS). Al-Nasser (2007) introduced a generalized robust sampling technique for the RSS, MRSS, and PRSS called LRSS. Taconeli and Cabral (2019) proposed different two-stage schemes; one is double NRSS (DNRSS) in which the first stage RSS is applied, while the NRSS procedure should be applied in the second stage. The second is neoteric DRSS (NDRSS) in which the first stage NRSS is applied, while the RSS procedure should be applied in the second stage, and the third is the neoteric-neoteric RSS (NNRSS) on which NRSS is applied in the two stages. They also proposed a single stage scheme that require ranking m^3which they call extended NRSS (ENRSS). Recently, Samuh et al. (2021) proposed a two-stage sampling scheme by combining RSS with MRSS. The proposed scheme suggest applying RSS in the first stage and MRSS in the second stage, which we shall denote by MRSS(RSS). Similarly, Hanandeh et al. (2022) proposed different two-stage schemes, among of the proposed schemes the one that combine ERSS in the first stage with MRSS in the second stage, denoted by MRSS(ERSS), was found to be the most efficient.

The aim of this paper is two-folded. First, we proposed a new sampling scheme which we called the new neoteric ranked set sampling (N-NRSS). Second, we proposed several two-stage ranked set sampling schemes based on SRSS and N-NRSS. These three methods differ from other RSS variations in the sense that they require ranking sets of sizes m^2 instead of ranking *m* sets of size *m* each.

To select a sample of size *n=mr* using RSS, we follow the following steps:

(1) Randomly select m^2 units from the population.
(2) Divide the m^2 units into *m* sets each of size *m*.
(3) Rank each set separately according to the variable of interest visually or using cheap method.
(4) Select the i^{th} ranked unit from the i^{th} set for actual measurement; where *i=1,2,…,m*.
(5) Repeat steps (1) through (4) *r* times to obtain a RSS of size *n=mr.*

The following steps summarizes the SRSS sampling scheme for obtaining a sample of size *n=mr*:

(1) Randomly select m^2 units from the population.
(2) Rank the m^2 units according to the variable of interest visually or using cheap method.
(3) Select the *(m+(m-1)*(i-1))* th ranked units for actual measurement; where *i=1,2,…,m*.
(4) Repeat steps (1) through (3) *r* times to obtain a SRSS of size *n=mr.*

NRSS differs from SRSS only in step (3). In NRSS the units selected for actual measurements are $\left(\frac{m+1}{2}+m*(i-1)\right)$th ranked units when *m* is odd, for *i=1,2,..,m*, and $(l+m*(i-1))$th ranked unit when *m* is even, where $l=\frac{m}{2}+1$ if i is odd and $l=\frac{m}{2}$ if i is even, and, for *i=1,2,..,m.*

To illustrate the SRSS and NRSS methods, let us consider two special cases; case 1 (*m=3, r=1)* and case 2 (*m=4,r=1)*. Let $Y_i; i=1,2,\dots,m^2$ be the selected as in step (1) above and let $Y_{[i]}; i=1,2,\dots,m^2$ be the their order statistics. Table 1 and Table 2 show the resulted samples for the two cases mentioned above. From these tables we can clearly observe that the NRSS is more spread than the SRSS.

Table 1: NRSS and SRSS when *m*=3 and *r*=1

The selected sample	*The order statistics*	*NRSS*	*SRSS*
$Y_1,Y_2,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8,Y_9$	$Y_{[1]},Y_{[2]},Y_{[3]},Y_{[4]},Y_{[5]},Y_{[6]},Y_{[7]},Y_{[8]},Y_{[9]}$	$Y_{[2]},Y_{[5]},Y_{[8]}$	$Y_{[3]},Y_{[5]},Y_{[7]}$

Table 2: NRSS and SRSS when *m*=4 and *r*=1

The selected sample	*The order statistics*	*NRSS*	*SRSS*
$Y_1,Y_2,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8,Y_9,$ $Y_{10},Y_{11},Y_{12},Y_{13},Y_{14},Y_{15},Y_{16}$	$Y_{[1]},Y_{[2]},Y_{[3]},Y_{[4]},Y_{[5]},Y_{[6]},Y_{[7]},Y_{[8]},Y_{[9]},$ $Y_{[10]},Y_{[11]},Y_{[12]},Y_{[13]},Y_{[14]},Y_{[15]},Y_{[16]}$	$Y_{[3]},Y_{[6]},Y_{[11]},Y_{[14]}$	$Y_{[4]},Y_{[7]},Y_{[10]},Y_{[13]}$

The remainder of this paper is organized as follows. Section 2 introduces the newly proposed sampling schemes. An extensive Monte Carlo simulation study to compare of the newly proposed sampling schemes with their competitors are presented in Section 3. The proposed sampling schemes are illustrated and discussed using real data in Section 4. Section 5 concludes with discussion and findings

2. The Proposed Sampling Schemes

In this work, we proposed two single-stage sampling schemes which forms alternatives to the already available single-stage schemes. One of the proposed single-stage schemes requires the initial selection and ranking of m^2 units which makes it comparable with SRSS and NRSS schemes that are already discussed earlier. The other one require the initial selection and ranking of m^3 units which makes it comparable with ENRSS. Two two-stage sampling schemes are also proposed and studied.

2.1 The New NRSS

The new NRSS (N-NRSS) scheme is single-stage scheme that is built based on NRSS, but the selected order statistics in step (3) in NRSS has different ordering especially when the set size m is even. However, when m is odd, both NRSS and N-NRSS select the same order statistics. The N-NRSS scheme can be described as follows:

(1) Randomly select m^2 units from the population.
(2) Rank the m^2 units according to the variable of interest visually or using cheap method.
(3) Select the $\left(\frac{m+1}{2}+m*(i-1)\right)$th ranked units if m is odd and $(l+m*(i-1))$th ranked unit when m is even, where $l=\frac{m}{2}$ if $i\le\frac{m}{2}$and $l=\frac{m}{2}+1$ if $i>\frac{m}{2}$; where $i=1,2,..,m$.
(4) Repeat steps (1) through (3) r times to obtain a N-NRSS of size $n=mr$.

The following two cases compare the selected units in N-NRSS with NRSS. It is clear that N-NRSS are even more spread than NRSS when m is even (Table 4) and, indeed, they are identical when m is odd (Table 3).

Table 3: NRSS and N-NRSS when m=3 and r=1:

The selected sample	*The order statistics*	*NRSS*	*N-NRSS*
$Y_1,Y_2,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8,Y_9$	$Y_{[1]},Y_{[2]},Y_{[3]},Y_{[4]},Y_{[5]},Y_{[6]},Y_{[7]},Y_{[8]},Y_{[9]}$	$Y_{[2]},Y_{[5]},Y_{[8]}$	$Y_{[2]},Y_{[5]},Y_{[8]}$

Table 4: NRSS and N-NRSS when m=4 and r=1:

The selected sample	*The order statistics*	*NRSS*	*N-NRSS*
$Y_1,Y_2,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8,Y_9,$ $Y_{10},Y_{11},Y_{12},Y_{13},Y_{14},Y_{15},Y_{16}$	$Y_{[1]},Y_{[2]},Y_{[3]},Y_{[4]},Y_{[5]},Y_{[6]},Y_{[7]},Y_{[8]},Y_{[9]},$ $Y_{[10]},Y_{[11]},Y_{[12]},Y_{[13]},Y_{[14]},Y_{[15]},Y_{[16]}$	$Y_{[3]},Y_{[6]},Y_{[11]},Y_{[14]}$	$Y_{[2]},Y_{[6]},Y_{[11]},Y_{[15]}$

2.2 Extended N-NRSS

The extended N-NRSS (EN-NRSS) scheme is single-stage scheme that is built based on N-NRSS. The EN-NRSS requires the selection and ranking of m^3 units. The EN-NRSS scheme can be described as follows:

(1) Randomly select m^3 units from the population.
(2) Rank the m^3 units according to the variable of interest visually or using cheap method.
(3) Select the $\left(\frac{m^2+1}{2}+m^2*(i-1)\right)$th ranked units if m is odd and $(l+m^2*(i-1))$th ranked unit when m is even, where $l=\frac{m^2}{2}$ if $i\le\frac{m}{2}$and $l=\frac{m^2}{2}+1$ if $i>\frac{m}{2}$; where $i=1,2,..,m$.
(4) Repeat steps (1) through (3) r times to obtain an EN-NRSS of size $n=mr$.

Table 5 describe EN-NRSS for $m=2$ and $r=1$.

Table 5: EN-NRSS when m=2 and r=1:

The selected sample	*The order statistics*	*EN-NRSS*
$Y_1,Y_2,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8$	$Y_{[1]},Y_{[2]},Y_{[3]},Y_{[4]},Y_{[5]},Y_{[6]},Y_{[7]},Y_{[8]}$	$Y_{[2]},Y_{[7]}$

2.3 Double new neoteric ranked set sampling

Double new neoteric ranked set sampling (DN-NRSS) is a two-stage design in which N-NRSS is applied in the two stages. To draw a DN-NRSS sample, apply the following steps:

(1) Randomly select m^3 units from the population and divide them into m sets of size m^2 each.
(2) Apply the N-NRSS on each set to obtain m sets of size m each.
(3) Merge the sets obtained in step 2 into one set of size m^2 and apply N-NRSS on this set to obtain a set of size m. This final set will be actually measured.
(4) Repeat steps 1 through 3 r times to obtain a DN-NRSS sample of size $n=mr$.

The DN-NRR is illustrated in Table 6 below for m=3.

Table 6: DN-NRSS for m=3 and r=1

Set	*The selected sample*	*The order statistics*	*N-NRSS*	*merge*	*DN-NRSS*
1	$Y_1^1, Y_2^1, Y_3^1, Y_4^1, Y_5^1, Y_6^1, Y_7^1, Y_8^1, Y_9^1$	$Y_{[1]}^1, Y_{[2]}^1, Y_{[3]}^1, Y_{[4]}^1, Y_{[5]}^1, Y_{[6]}^1, Y_{[7]}^1, Y_{[8]}^1, Y_{[9]}^1$	$(X_1, X_2, X_3) = (Y_{[2]}^1, Y_{[5]}^1, Y_{[8]}^1)$	$X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9$	$X_{[2]}, X_{[5]}, X_{[8]}$
2	$Y_1^2, Y_2^2, Y_3^2, Y_4^2, Y_5^2, Y_6^2, Y_7^2, Y_8^2, Y_9^2$	$Y_{[1]}^2, Y_{[2]}^2, Y_{[3]}^2, Y_{[4]}^2, Y_{[5]}^2, Y_{[6]}^2, Y_{[7]}^2, Y_{[8]}^2, Y_{[9]}^2$	$(X_4, X_5, X_6) = Y_{[2]}^2, Y_{[5]}^2, Y_{[8]}^2$		
3	$Y_1^3, Y_2^3, Y_3^3, Y_4^3, Y_5^3, Y_6^3, Y_7^3, Y_8^3, Y_9^3$	$Y_{[1]}^3, Y_{[2]}^3, Y_{[3]}^3, Y_{[4]}^3, Y_{[5]}^3, Y_{[6]}^3, Y_{[7]}^3, Y_{[8]}^3, Y_{[9]}^3$	$(X_7, X_8, X_9) = Y_{[2]}^3, Y_{[5]}^3, Y_{[8]}^3$		

2.4 Double systematic ranked set sampling

In a similar fashion of DN-NRSS, the Double systematic ranked set sampling (DSRSS) is a two-stage scheme in which SRSS is applied in the two stages. The following steps summarizes DSRSS:

(1) Randomly select m^3 units from the population and divide them into m sets of size m^2 each.
(2) Apply the SRSS on each set to obtain m sets of size m each.
(3) Merge the sets obtained in step 2 into one set of size m^2 and apply SRSS on this set to obtain a set of size m. This final set will be actually measured.
(4) Repeat steps 1 through 3 r times to obtain a DSRSS sample of size $n=mr$.

The DN-NRR is illustrated in the Table 7 for m=3 and $r=1$.

Table 6: DSRSS for m=3 and r=1

Set	*The selected sample*	*The order statistics*	*SRSS*	*merge*	*DSRSS*
1	$Y_1^1, Y_2^1, Y_3^1, Y_4^1, Y_5^1, Y_6^1, Y_7^1, Y_8^1, Y_9^1$	$Y_{[1]}^1, Y_{[2]}^1, Y_{[3]}^1, Y_{[4]}^1, Y_{[5]}^1, Y_{[6]}^1, Y_{[7]}^1, Y_{[8]}^1, Y_{[9]}^1$	$(X_1, X_2, X_3) = (Y_{[3]}^1, Y_{[5]}^1, Y_{[7]}^1)$	$X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9$	$X_{[3]}, X_{[5]}, X_{[7]}$
2	$Y_1^2, Y_2^2, Y_3^2, Y_4^2, Y_5^2, Y_6^2, Y_7^2, Y_8^2, Y_9^2$	$Y_{[1]}^2, Y_{[2]}^2, Y_{[3]}^2, Y_{[4]}^2, Y_{[5]}^2, Y_{[6]}^2, Y_{[7]}^2, Y_{[8]}^2, Y_{[9]}^2$	$(X_4, X_5, X_6) = Y_{[3]}^2, Y_{[5]}^2, Y_{[7]}^2$		
3	$Y_1^3, Y_2^3, Y_3^3, Y_4^3, Y_5^3, Y_6^3, Y_7^3, Y_8^3, Y_9^3$	$Y_{[1]}^3, Y_{[2]}^3, Y_{[3]}^3, Y_{[4]}^3, Y_{[5]}^3, Y_{[6]}^3, Y_{[7]}^3, Y_{[8]}^3, Y_{[9]}^3$	$(X_7, X_8, X_9) = Y_{[3]}^3, Y_{[5]}^3, Y_{[7]}^3$		

3. Simulation Study

In this section we performed an extensive Monte Carlo simulation study to assess the performance of the proposed sampling schemes for estimating the population mean for several symmetric and asymmetric popular statistical distributions. The selected distributions were considered by many authors; see for example Hanandeh et al., (2022) and the references therein. The proposed schemes are compared with their counterparts; namely NNRSS, ENRSS (Taconeli and Cabral, 2019), MRSS(RSS) (Samuh et al., 2021), and MRSS(ERSS) (Hanandeh et al., 2022).

To assess the effect of the set size, m, we considered different set size values ranging from 3 to 6. Note that for larger set sizes, the proposed schemes are not feasible in practice especially those requiring ranking m^2 units at each stage, therefore larger m values were not considered. To increase sample size one may choose a larger r, however, the efficiency of the proposed methods with respect to SRS will not be affected by the value of r. Therefore, we set $r = 1$ in all of our simulations. Please note that all samplings are done from infinite populations

Different scenarios were considered by combining each set size with each sampling method. For each scenario, N=100,000 datasets were selected from each distribution and the mean square error (MSE) of the mean estimate were calculated. sampling schemes were then compared using the relative efficiency (RE) of the mean estimators based on the proposed sampling schemes when compared with the mean estimator based on SRS. Let $\bar{X}_{Mi}$, denote the sample mean based on sampling scheme M and dataset i and let μ be the true population mean, the MSE of $\bar{X}_M$ is defined as

$$MSE(\bar{X}_M) = \frac{\sum_{i=1}^{N}(\bar{X}_{Mi}-\mu)^2}{N},$$

and the RE of $\bar{X}_M$ with respect to $\bar{X}_{SRS}$ is defined as

$$RE(\bar{X}_M) = \frac{MSE(\bar{X}_{SRS})}{MSE(\bar{X}_M)}.$$

The results of the simulation studies are given in Tables 7 to 10. The following summarizes the findings

- For logistic and student-t distributions, DSRSS is the most efficient method, followed by EN-NRSS and ENRSS.
- For other symmetric distributions, EN-NRSS and ENRSS perform almost the same and are the most efficient methods compared to the other sampling schemes, followed by DN-NRSS and NNRSS.
- For asymmetric distributions, EN-NRSS is the most efficient scheme followed by ENRSS and DN-NRSS especially as for even *m*.

In general, we recommend EN-NRSS for asymmetric distributions, DRSS for logistic and student-t distributions, and either EN-NRSS or ENRSS for any other symmetric distributions.

Table 7: Relative efficiency for RSS-based estimators under perfect ranking when *m*=3

Distribution	DRSS	MRSS(RSS)	MRSS(ERSS)	NNRSS	ENRSS	EN-NRSS	DN-NRSS	DSRSS
U(0,1)	3.043	2.408	2.406	6.592	7.589	7.589	6.534	4.168
N(0,1)	2.652	3.628	3.653	7.113	8.047	8.047	7.123	6.925
logistic(0,1)	2.424	4.329	4.33	7.645	8.613	8.613	7.636	8.559
student-t(4)	2.078	5.574	5.59	9.317	10.325	10.325	9.3	11.467
Beta(3,3)	2.827	3.06	3.063	6.66	7.636	7.636	6.668	5.681
ArcSin(0,1)	3.022	1.903	1.896	6.883	8.114	8.114	6.862	3.102
Beta(5,2)	2.676	2.818	2.799	6.569	7.544	7.544	6.586	4.976
Rayleigh(1)	2.63	3.029	3.032	6.754	7.607	7.607	6.8	5.523
Half Normal(2)	2.489	2.403	2.404	6.543	7.327	7.327	6.559	4.207
Exponential(1)	2.032	2.032	2.035	6.405	7.238	7.238	6.368	3.422
Gamma(2,3)	2.261	2.526	2.506	6.612	7.502	7.502	6.595	4.338
ChiSquare(3)	2.192	2.304	2.305	6.585	7.287	7.287	6.557	3.947
LogNormal(0,1)	1.488	2.448	2.438	7.629	8.791	8.791	7.644	4.038
Pareto(1,3)	1.486	3.052	3.047	8.806	10.849	10.849	8.841	4.507
Weibull(0.5,1)	1.297	2.118	2.125	7.309	8.297	8.297	7.304	3.483
Gamma(0.5,1)	1.706	1.635	1.637	6.11	6.977	6.977	6.14	2.702

Table 8: Relative efficiency for RSS-based estimators under perfect ranking when *m*=4

Distribution	DRSS	MRSS(RSS)	MRSS(ERSS)	NNRSS	ENRSS	EN-NRSS	DN-NRSS	DSRSS
U(0,1)	4.295	3.484	3.346	11.457	13.969	14.852	16.447	8.019
N(0,1)	3.541	5.095	3.554	13.207	14.602	14.557	12.126	12.836
logistic(0,1)	3.096	5.94	3.776	14.609	15.814	15.342	11.329	15.652
student-t(4)	2.541	7.684	4.381	18.16	19.231	18.283	11.775	20.767
Beta(3,3)	3.897	4.352	3.422	12.179	14.023	14.342	13.23	10.65
ArcSin(0,1)	4.313	2.792	3.31	12.888	15.019	16.063	20.789	6.288
Beta(5,2)	3.649	3.558	3.306	11.085	13.437	14.041	12.292	7.712
Rayleigh(1)	3.46	3.802	3.362	11.343	13.553	14.071	12.156	8.586

Half Normal(2)	3.296	2.742	3.149	9.441	12.403	13.509	11.38	5.375
Exponential(1)	2.525	1.958	2.935	7.457	10.831	12.55	9.271	3.43
Gamma(2,3)	2.893	2.675	3.145	9.222	12.19	13.341	10.492	5.091
ChiSquare(3)	2.76	2.35	3.062	8.401	11.704	13.085	9.983	4.302
LogNormal(0,1)	1.667	2.113	3.359	6.95	10.982	13.279	10.558	3.5
Pareto(1,3)	1.532	2.49	4.054	7.607	11.544	14.144	12.974	3.878
Weibull(0.5,1)	1.45	1.706	3.018	5.841	9.08	11.4	9.67	2.87
Gamma(0.5,1)	2.09	1.465	2.641	6.064	9.607	11.728	7.985	2.45

Table 9: Relative efficiency for RSS-based estimators under perfect ranking when *m*=5

Distribution	DRSS	MRSS(RSS)	MRSS(ERSS)	NNRSS	ENRSS	EN-NRSS	DN-NRSS	DSRSS
U(0,1)	5.679	4.365	3.257	20.427	23.212	23.212	20.291	13.599
N(0,1)	4.447	7.328	5.256	20.578	23.361	23.361	20.64	20.673
logistic(0,1)	3.845	9.107	6.432	21.764	24.578	24.578	21.629	25.249
student-t(4)	2.891	12.197	8.631	25.187	28.798	28.798	25.184	32.517
Beta(3,3)	5.101	6.037	4.374	19.993	22.655	22.655	20.029	17.568
ArcSin(0,1)	5.747	3.144	2.432	21.189	23.993	23.993	21.036	11.089
Beta(5,2)	4.641	3.547	3.084	19.068	21.845	21.845	19.211	10.084
Rayleigh(1)	4.453	4.13	3.472	18.817	21.906	21.906	18.931	11.263
Half Normal(2)	4.144	2.214	2.132	17.354	20.08	20.08	17.259	6.114
Exponential(1)	2.992	1.267	1.345	14.263	16.629	16.629	14.177	3.436
Gamma(2,3)	3.508	1.994	2.001	16.308	18.89	18.89	16.333	5.409
ChiSquare(3)	3.329	1.661	1.698	15.551	17.811	17.811	15.482	4.503
LogNormal(0,1)	1.801	1.331	1.432	12.418	14.024	14.024	12.371	3.36
Pareto(1,3)	1.666	1.574	1.688	15.99	15.955	15.955	16.012	4.439
Weibull(0.5,1)	1.556	1.036	1.128	10.306	11.406	11.406	10.231	2.59
Gamma(0.5,1)	2.427	0.864	0.953	12.029	14.233	14.233	12.141	2.33

Table 10: Relative efficiency for RSS-based estimators under perfect ranking when *m*=6

Distribution	DRSS	MRSS(RSS)	MRSS(ERSS)	NNRSS	ENRSS	EN-NRSS	DN-NRSS	DSRSS
U(0,1)	7.24	5.598	5.039	29.718	33.957	34.553	34.676	21.037
N(0,1)	5.387	9.123	5.361	30.618	34.45	34.39	29.453	29.995
logistic(0,1)	4.51	11.392	5.724	32.522	35.56	35.195	28.615	36.032
student-t(4)	3.202	15.095	6.694	38.995	41.332	40.566	30.7	46.847
Beta(3,3)	6.308	7.614	5.148	29.29	33.533	33.775	30.737	25.757
ArcSin(0,1)	7.349	4.091	5.57	31.542	35.345	36.029	38.408	17.849
Beta(5,2)	5.698	3.71	4.811	25.366	30.982	31.868	29.671	11.981
Rayleigh(1)	5.434	4.373	4.932	25.254	30.604	31.458	29.882	13.584
Half Normal(2)	5.012	2.129	4.401	20.223	27.507	29.278	29.251	6.617
Exponential(1)	3.508	1.148	3.811	13.831	21.109	23.693	26.692	3.416
Gamma(2,3)	4.144	1.867	4.302	17.871	24.996	27.014	27.81	5.601
ChiSquare(3)	3.921	1.527	4.106	16.287	23.756	26.077	27.903	4.62
LogNormal(0,1)	1.923	1.152	3.641	10.032	15.492	17.749	27.804	3.155
Pareto(1,3)	1.793	1.416	4.476	10.617	15.917	18.209	31.484	3.499
Weibull(0.5,1)	1.765	0.921	3.168	7.916	12.334	14.35	25.449	2.451
Gamma(0.5,1)	2.767	0.767	3.258	10.337	17.272	20.084	24.325	2.24

4. Real Data Example

In this section we will analyze the *spati2* dataset that is available in the R-package *`Imfor'* (Mehtatalo, 2018). The dataset was collected by Pukkala (1989) and consists of the heights and diameters of 1678 Scots pine trees in Ilomantsi, Finland. The variable of interest is the height of trees and the aim is to estimate the mean of the trees heights. Figure 1 shows the distribution of trees heights and the scatter plot of diameter vs heights. It can be seen that the heights are unimodal, skewed to the right, and have a strong positive correlation with diameter (Pearson's correlation =0.866). The summary statistics of heights are presented in Table 11.

Since the diameter of the trees is more accessible and easy to rank visually, the data will be analyzed twice: First, ranking will be done according to the height variable which is considered perfect ranking. Second, ranking will be done according to the diameter variable, that is imperfect ranking. We considered the same set sizes (m=3,4,5, and 6). Foe each set size, 100,000 datasets were samples from the heights data using the sampling schemes DRSS, MRSS(RSS), MRSS(ERSS), NNRSS, ENRSS, EN-NRSS, DN-NRSS, and DSRSS. The relative efficiency of the proposed schemes with respect to SRS are presented in Table 12 for perfect ranking and Table 13 for imperfect ranking. The result agrees with the simulation study for asymmetric distributions. The newly proposed EN-NRSS slightly outperforms other sampling schemes, regardless whether the ranking is perfect or imperfect, followed by ENRSS and DN-NRSS.

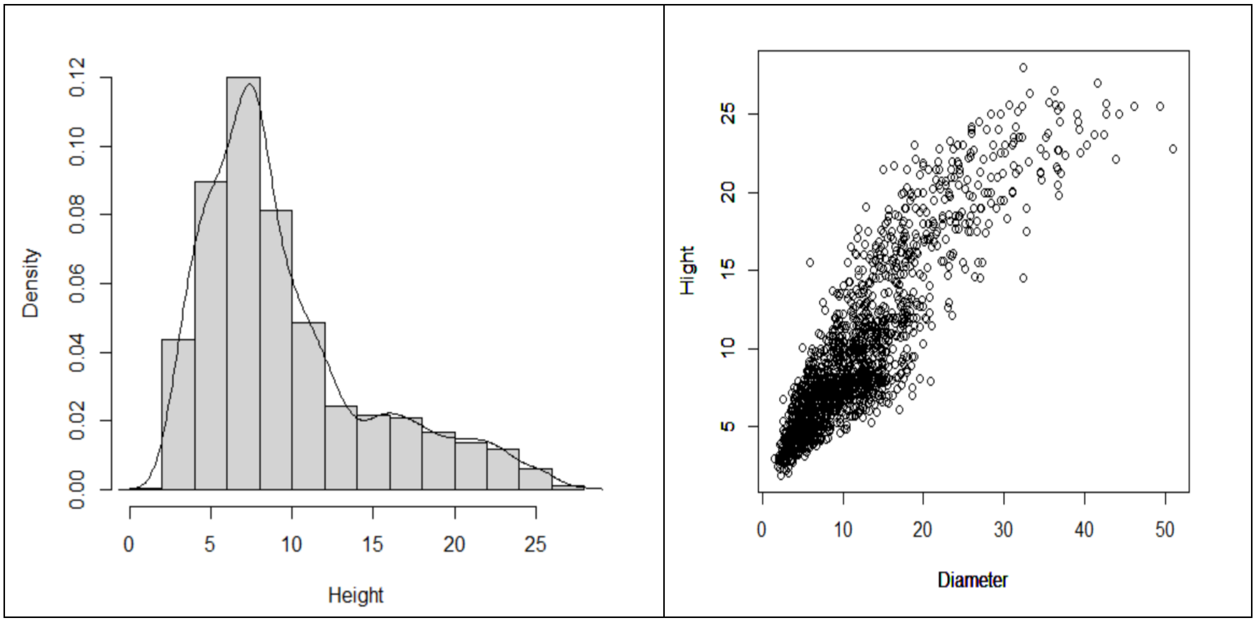

Figure 1: Histogram of trees heights (left) and scatter plot of diameter vs heights (right)

Table 11: Descriptive statistics of trees heights (in meter)

N	Mean	Min	Q1	Median	Q3	Max
1678	9.566	1.9	6	8	11.7	28

Table 12: Relative efficiency of different sampling schemes with perfect ranking

m	DRSS	MRSS(RSS)	MRSS(ERSS)	NNRSS	ENRSS	EN-NRSS	DN-NRSS	DSRSS
3	2.3329	2.8234	2.8234	5.336	6.0278	6.0278	5.336	3.668
4	3.0726	3.0326	2.7619	7.209	11.16	12.2737	8.2034	3.8033
5	3.9119	2.4917	2.4649	17.9963	21.7443	21.7443	17.9963	3.8618
6	4.7117	2.3047	3.616	19.4607	28.8618	30.2444	21.9428	3.9429

Table 13: Relative efficiency of different sampling schemes with imperfect ranking

m	DRSS	MRSS(RSS)	MRSS(ERSS)	NNRSS	ENRSS	EN-NRSS	DN-NRSS	DSRSS
3	1.8102	2.2993	2.3118	2.6482	2.7391	2.7537	2.6368	2.8492
4	2.1158	2.393	2.0075	2.8656	3.3838	3.5158	3.3538	2.7698
5	2.4082	2.4787	2.3124	3.8867	4.1161	4.1186	3.8775	2.6496
6	2.659	2.3647	2.2589	3.6645	4.3791	4.4526	4.2859	2.6515

5. Discussion and Conclusion

In this paper we proposed four sampling schemes based on NRSS and SRSS, to estimate the population mean. These sampling schemes are compared with the old sampling schemes through an extensive Monte Carlo simulation study as well as a real data example on heights of pine trees.

It is observed that some of the newly proposed methods, namely EN-NRSS and DN-NRSS outperform all previously proposed methods especially for asymmetric distributions. This also can be observed from the real data example in both cases when ranking is done perfectly or with error. EN-NRSS is followed by ENRSS in the real data example.

For symmetric distributions, the newly proposed schemes EN-NRSS and DSRSS outperform other schemes followed by NNRSS and ENRSS. This can be clearly seen for logistic and student's t distributions.

It is worth mentioning that EN-NRSS and ENRSS requires ranking m^3 units which might not be an easy task in practice especially for moderate and large m, therefore, it might be easier to consider applying DN-NRSS or NNRSS which provide a competitive performance to EN-NRSS.

As a future work, the efficiency of the proposed methods in estimating other population parameters, such as the variance, is to be considered. To the best of our knowledge, most work has focused on the estimation of the population mean.

Funding: No funding was received for conducting this study.

Data availability: The data used in the simulation study was self-generated Using the R statistical software. Real data example was obtained from *spati2* dataset that is available in the R-package *`Imfor'* through Datacamp website https://www.datacamp.com/datalab/w/8d5d8d48-b693-4549-a3f1-76b1b6be887d/edit. The data file can also be obtained from the corresponding author upon request.

Code availability: The codes in this paper represent a new development on R statistical software and can be obtained from the corresponding author upon request.

Declarations Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical statements: We hereby declare that, this manuscript is the result of our independent creation under the reviewers' comments. Except the quoted contents, this manuscript does not contain any research achievements by other individuals or groups. We are the only authors of this manuscript, that have been published or written. The legal responsibility of this statement shall be borne by us.

References

1- Al-Nasser, A. D. (2007). L ranked set sampling: A generalization procedure for robust visual sampling. *Communications in Statistics—Simulation and Computation*, 36(1), 33-43.
2- A-Saleh, M. F. and Al-Kadiri, M. A. (2000). Double-ranked set sampling. *Statistics & Probability Letters*, 48(2):205-212.
3- Chen Z, Bai Z, Sinha B. (2003). Ranked set sampling: theory and applications. New York: *Springer Science & Business Media.*
4- Hanandeh, A. A., Al-Nasser, A. D., & Al-Omari, A. I. (2022). New mixed ranked set sampling variations. *Stat*, 11(1), e417.
5- Khan, L., Shabbir, J., & Khalil, A. (2019). Modified ranked set sampling schemes for estimating the population mean. *Journal of statistics and management systems*, 22(8), 1481-1497.
6- Mehtatalo L. (2018). lmfor: Functions for forest biometrics. R package version 1.2; 2017. Available from: https://CRAN.R-project.org/package=lmfor.
7- [20] Pukkala T. Prediction of tree diameter and height in a Scots pine stand as a function
8- McIntyre, G., (1952). A method for using selective sampling using ranked set. *Austral. J. Agric. Res*. 3, 385--890.

9- Muttlak, H. A. (1996). Pair ranked set sampling. *Biometrical Journal*, 38(7):879-885.
10- Muttlak, H. (1997). Median ranked set sampling. *Journal of Applied Statistical Sciences*, 6, 245-255.
11- Pukkala, T. 1989. Prediction of tree diameter and height in a Scots pine stand as a function of the spatial pattern of trees. Silva Fennica 23(2): 83-99.
12- Samawi, H. M. (2002). On double extreme rank set sample with application to regression estimator. *Metron—International Journal of Statistics,* 60, 50-63.
13- Samawi, H. M. Abu-Dayyeh, W. and Ahmed, M. S. (1996). Estimating the population mean using extreme ranked set sampling. *Biometrical Journal*, 38:577-586.
14- Samawi, H.M. and Tawalbeh, E.M. (2002). Double median ranked set sample: comparing to other double ranked samples for mean and ratio estimators, *Journal of Modern Applied Statistical Methods*, 1(2), 428–442.
15- Samuh, M. H., Omar, M. H., & Hossain, M. P. (2021). Mixed double-ranked set sampling: A more efficient and practical approach. *REVSTAT-Statistical Journal*, 19(1), 145-160.
16- Stokes, L., (1980). Estimation of variance using judgment ordered ranked set samples. *Biometrics* 36, 35--42.
17- Taconeli, C. A. (2024). Dual-rank ranked set sampling. *Journal of Statistical Computation and Simulation*, 94(1), 29-49.
18- Taconeli, C. A., & Cabral, A. D. S. (2019). New two-stage sampling designs based on neoteric ranked set sampling. *Journal of Statistical Computation and Simulation*, 89(2), 232-248.
19- Takahasi, K., Wakimoto, K., (1968). On unbiased estimates of the population mean based on the sample stratified by means of ordering. *Ann. Inst. Statist. Math*. 20, 421-- 428.
20- Zamanzade, E. and Al-Omari, A. I. (2016). New ranked set sampling for estimating the population mean and variance. *Hacettepe journal of Mathematics and Statistics*, 46(6): 1891-1905.

APPENDIX: **R code for the simulation Study**

```
rnd=function(ss) rnorm(ss)
mu=0

#runif(ss)
#mu=0.5

#rnorm(ss)
#mu=0

#rlogis(ss)
#mu=0

#rt(ss,df=4)
#mu=0

#rbeta(ss,3,3)
#mu=0.5

# (sin(0.5*pi*runif(ss)))^2
#mu=0.5

#rbeta(ss,5,2)
#mu=5/7

#sqrt(rexp(ss,1))
#mu=gamma(1.5)

#abs(rnorm(ss,mean=0,sd=2))
#mu=2*sqrt(2/pi)
```

DOI: https://doi.org/10.18187/pjsor.v21i2.4872

```
#rexp(ss,1)
#mu=1

#rgamma(ss,2,rate=3)
#mu=2/3

#rgamma(ss,1.5,scale=2)
#mu=3

#rlnorm(ss)
#mu=exp(0.5)

#(1-runif(ss))^(-1/3)
#mu=1.5

#rweibull(ss, 0.5, 1)
#mu=gamma(3)

#rgamma(ss,0.5,scale=1)
#mu=0.5

iter=1e5
dic=3

time1=Sys.time()


for(m in c(3,4,5,6))
{
r=1
n=m*r
k=floor(m/2)
ymat=matrix(0,m,m)

xbar1=rep(NA,iter) ## SRS
xbar2=rep(NA,iter) ## DRSS
xbar3=rep(NA,iter) ## MRSS(RSS)
xbar4=rep(NA,iter) ## MRSS(ERSS)
xbar5=rep(NA,iter) ## NNRSS
xbar6=rep(NA,iter) ## ENRSS
xbar7=rep(NA,iter) ## EN-NRSS
xbar8=rep(NA,iter) ## DN-NRSS
xbar9=rep(NA,iter) ## DSRSS


for(j in 1:iter)
{

##### SRS ######
xdata=NA
xsrs=rnd(n)
xbar1[j]=mean(xsrs)

##### DRSS #######

xdata=NA
```

```
for(i in 1:r)
{
x=array(rnd(m^3),dim=c(m,m,m))
xs=apply(x,c(2,3),sort)
y=apply(xs,3,diag)
y=apply(y,2,sort)
xdata=c(xdata,diag(y))
}
xdata=xdata[-1]
xbar2[j]=mean(xdata)

##### RSS then MRSS #####

xdata=NA

for(i in 1:r)
{
x=array(rnd(m^3),dim=c(m,m,m))
xs=apply(x,c(2,3),sort)
y=apply(xs,3,diag)

if(k==(m/2))
{
   ys=apply(y,2,sort)
   data=c(ys[k,1:k],ys[k+1,(k+1):m])
   }else{
   data=apply(y,2,median)
}

xdata=c(xdata,data)

}
xdata=xdata[-1]
xbar3[j]=mean(xdata)

##### ERSS then MRSS #####

xdata=NA

for(i in 1:r)
{
x=array(rnd(m^3),dim=c(m,m,m))
a=apply(x,c(2,3),min)
b=apply(x,c(2,3),max)

if(k==(m/2))
{
   ymat[1:k,]=a[1:k,]
   ymat[(m-k+1):m,]=b[(m-k+1):m,]
   ymat=apply(ymat,2,sort)
   a=ymat[k,]
   b=ymat[k+1,]
   data=c(a[1:k],b[(k+1):m])
   }else{
   c1=apply(x,c(2,3),median)
   ymat[1:k,]=a[1:k,]
   ymat[(m-k+1):m,]=b[(m-k+1):m,]
```

```
   ymat[k+1,]=c1[k+1,]
   data=apply(ymat,2,median)
}

xdata=c(xdata,data)
}
xdata=xdata[-1]
xbar4[j]=mean(xdata)

##### NRSS then NRSS (NNRSS) #####

xdata=NA

for(i in 1:r)
{
x=matrix(rnd(m^3),m,m^2)
xs=apply(x,1,sort)

if(k==m/2)
{
ll=rep(c((m+2)/2,k),k)
id=ll+m*(0:(m-1))
y=xs[id,]
ys=sort(y)
data=ys[id]
}else{
id=(m+1)/2+m*(0:(m-1))
y=xs[id,]
ys=sort(y)
data=ys[id]
}
xdata=c(xdata,data)
}
xdata=xdata[-1]
xbar5[j]=mean(xdata)

##### ENRSS #####

x=rnd(m^3)
xs=sort(x)

if(k==m/2)
{
ll=rep(c((m^2+2)/2,m^2/2),k)
id=ll+m^2*(0:(m-1))
data=xs[id]
}else{
id=(m^2+1)/2+m^2*(0:(m-1))
data=xs[id]
}
xbar6[j]=mean(data)



##### EN-NRSS #####
```

```
if(k==m/2)
{
ll=rep(c(m^2/2,(m^2+2)/2),each=k)
id=ll+m^2*(0:(m-1))
data=xs[id]
}else{
id=(m^2+1)/2+m^2*(0:(m-1))
data=xs[id]
}

xbar7[j]=mean(data)


##### NEW NRSS then New NRSS #####

xdata=NA

for(i in 1:r)
{
x=matrix(rnd(m^3),m,m^2)
xs=apply(x,1,sort)

if(k==m/2)
{
ll=rep(c(k,k+1),each=k)
id=ll+m*(0:(m-1))
y=xs[id,]
ys=sort(y)
data=ys[id]
}else{
id=(m+1)/2+m*(0:(m-1))
y=xs[id,]
ys=sort(y)
data=ys[id]
}
xdata=c(xdata,data)
}
xdata=xdata[-1]
xbar8[j]=mean(xdata)


##### SRSS then SRSS #####

xdata=NA

for(i in 1:r)
{
x=matrix(rnd(m^3),m,m^2)
xs=apply(x,1,sort)
id=(m-1)*(0:(m-1))+m
y=xs[id,]
ys=sort(y)
data=ys[id]
xdata=c(xdata,data)
}
xdata=xdata[-1]
xbar9[j]=mean(xdata)
```

```
}

mse1=sum((xbar1-mu)^2)/iter
mse2=sum((xbar2-mu)^2)/iter
mse3=sum((xbar3-mu)^2)/iter
mse4=sum((xbar4-mu)^2)/iter
mse5=sum((xbar5-mu)^2)/iter
mse6=sum((xbar6-mu)^2)/iter
mse7=sum((xbar7-mu)^2)/iter
mse8=sum((xbar8-mu)^2)/iter
mse9=sum((xbar9-mu)^2)/iter

cat(round(mse1/mse2,dic),round(mse1/mse3,dic),round(mse1/mse4,dic),round(mse1
/mse5,dic),round(mse1/mse6,dic),round(mse1/mse7,dic),round(mse1/mse8,dic),rou
nd(mse1/mse9,dic),"\n")

}


time2=Sys.time()
print(time2-time1)
```