GENERATION OF NEIGHBOUR BALANCED GENERALIZED ROW COLUMN DESIGNS THROUGH SAS MACRO

Anindita Datta[1], Seema Jaggi[1], Cini Varghese[1], Eldho Varghese[2], Arpan Bhowmik[1]
and Mohd Harun[1]

[1] ICAR-Indian Agricultural Statistics Research Institute, New Delhi
[2] ICAR-Central Marine Fisheries Research Institute, Cochin

In case of a Generalized Row Column designs, there are more number of units in a cell and the treatment applied to one experimental units in a cell may affect the response on neighbouring units in the same cell. Neighbor Balanced GRC (NBGRC) designs are useful for this situation. A GRC design with v treatments in p rows and q columns is said to be neighbor balanced if within a cell every treatment has every other treatment appearing as neighbour a constant number of times (say λ times). A SAS macros have been developed for the generation of Neighbour Balanced GRC (NBGRC) designs. These SAS macros will facilitate the researchers under NARES to easily obtain a NBGRC designs by entering the value of respective parameters.
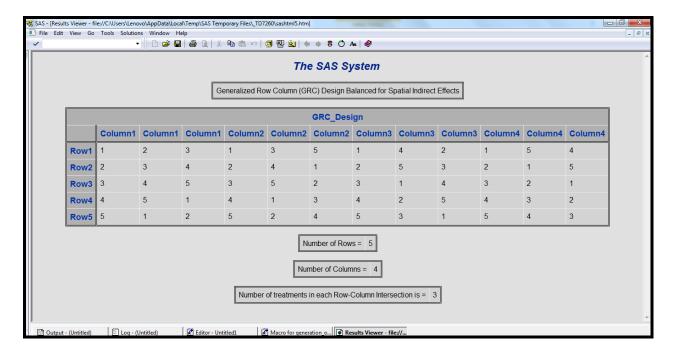
The following SAS macro has been developed to generate NBGRC designs for parameter v (prime), $p = v$, $q = v-1$, $k = s$ ($3 \leq s \leq v-1$), $r = s(v-1)$ and $\lambda=2(s-1)$. Here, user need to enter the number of treatment as v (**prime**) and the number of units per cell as k (≥ 2). If user run the macro after entering any prime number as the value of v and also as the value of k, then the SAS Macro will generate a particular NBGRC designs corresponding to the value of v and k under the heading **Neighbour Balanced Generalized Row Column (GRC) Design**. Once user run the macro, every time the SAS macro would also generate a word file containing the output. User can then save the word file.

Code

```
%let v=5;/* Enter the number of teatments (Treament number should be odd
number)*/
%let s=3;/*Enter the cell sizes(it varies from 2 to (v-1)*/
ods rtf  file= 'output.rtf' startpage=no;
proc iml;
TRT1=j(&v,&s*(&v-1),0);
k=1;
do i=1 to &s;
do j=1 to &v;
TRT1[j,i]=(j+(i-1));
if TRT1[j,i]>&v then TRT1[j,i]=TRT1[j,i]-&v;
end;
end;
```

```
kk=&s+1;
do k=1 to &v-1;
do i=1 to &s;
do j=1 to &v;
TRT1[j,kk]=TRT1[j,kk-(&s)]+(i-1);
if TRT1[j,kk]>&v then do;
TRT1[j,kk]=TRT1[j,kk]-&v;
end;
end;
kk=kk+1;
end;
end;
varNames2= "Column1":"Column"+strip(char(&v-1));
varNames3= "Row1":"Row"+strip(char(&v));
do i=1 to (&v-1);
do j=1 to &s;
columns=varNames2[ ,i];
columns1=columns1||columns;
end;
end;
GRC_Design=char(TRT1,5,0);
print 'Neighbour Balanced Generalized Row Column (GRC) Design';
print GRC_Design[rowname=varNames3 colname=columns1];
print 'Number of Rows =' &v;
print 'Number of Columns ='(&v-1);
print 'Number of treatments in each Row-Column Intersection is =' &s;
ods rtf close;
quit;
```

A screenshot of the output is as follows

SAS Macros developed under the present investigation would provide a readymade solutions to the end user under NARES. These macros would help in increasing the application potential of NBGRC designs.

References

Bailey, R. A. (1988). Semi Latin squares. *Journal of Statistical Planning and Inference*, **18**, 299-312.

Jaggi, S., Varghese, C., Varghese, E. and Sharma, V. K. (2010). Generalized incomplete Trojan-type designs. *Statistics and Probability Letters*, **80**, 706-710.

Datta, A., Jaggi, S., Varghese, C. and Varghese, E. (2014). Structurally incomplete row-column designs with multiple units per cell. *Statistics and Applications*, **12** (1&2), 71-79.

Datta, A., Jaggi, S., Varghese, C. and Varghese, E. (2015). Some series of row-column designs with multiple units per cell. *Calcutta Statistical Association Bulletin*, **67** (265-266), 89-99.

Datta, A., Jaggi, S., Varghese, C. and Varghese, E. (2016). Series of Incomplete Row-Column Designs with Two Units per Cell. *Advances in Methodology and Statistics*. **13**(1), 17-25.

Datta, A., Jaggi, S., Varghese, E. and Varghese, C. (2017). Generalized Confounded Row-Column Designs. *Communication in Statistics: Theory and Methods*. **46**(12), 6213-6221.