

Evaluating Grant Applications with Generalized Chain Block Designs in R

Dedicated to the Memory of John Mandel

Giles Crane, Cynthia Collins, and Karin Mille
NJ Department of Health and Senior Services

Grant Application Review via Experimental Design involves many aspects of public management and modeling.

- Many applications for few grants
- Reliable reviewers are few, time limited
- Review criteria prescribed in RFA
- Experimental design involving reviewers, applications, order of review
- Desired: adjust for reviewers, order
- Objectives---be fair, thorough, efficient, & defensible.

Grant Application Reviews were held, utilizing experimental designs: large grants & mini-grants

✓ Agency Health Grants ---

10 large grants, 26 applications

PBIB, modified two-way design

✓ Mini-grants for exercise/nutrition ---

20 Small grants, 25 Applications

Lattice , one-way design

✓ More Mini-grants for exercise/nutrition---

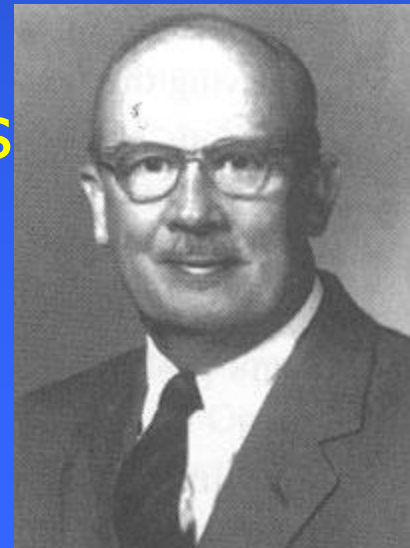
10-20 small grants, 32 Applications

Generalized chain block, Two-way design

Statisticians have provided an array of Experimental Designs and models.

- ❖ Completely randomized
- ❖ Randomized blocks
- ❖ Latin squares
- ❖ Youden squares (incomplete Latin squares)
- ❖ Williams squares (carry-over effect)
- ❖ Balanced incomplete blocks (BIB)
- ❖ Partially balanced incomplete blocks
- ❖ Chain block designs
- ❖ Lattice

Jack
Youden



Experimental Design SI.4 *



Cynthia
Collins

Periods (= Replicates) 1 to 3						
Reviewers	Replicate 1		Replicate 2		Replicate 3	
Reviewer 1	1	22	6	23	12	26
Reviewer 2	2	23	7	24	13	14
Reviewer 3	3	24	8	25	1	15
Reviewer 4	4	25	9	26	2	16
Reviewer 5	5	26	10	14	3	17
Reviewer 6	6	14	11	15	4	18
Reviewer 7	7	15	12	16	5	19
Reviewer 8	8	16	13	17	6	20
Reviewer 9	9	17	1	18	7	21
Reviewer 10	10	18	2	19	8	22
Reviewer 11	11	19	3	20	9	23
Reviewer 12	12	20	4	21	10	24
Reviewer 13	13	21	5	22	11	25

Applications are numbered 1,2, ... 26 Each reviewer scores 6 applications.
 For example: Reviewer 3 scores applications 3 and 24 first,
 applications 8 and 25 second, and applications 1 and 15 third.

The Request for Application (RFA) provides the basis for scoring (minimum of 70 points needed)

Scoring was not "blind" -- the applicant agency could be identified due to location and nature of applic.

No regional divisions. Statewide.

SCORING CRITERIA FORM

NJ Department of Health & Senior Services / NJCPFS Mini - Grant Applications
Healthy Community Development RFA - Nov. 13, 2009

Community - Municipality Name _____

Reviewer Signature _____

Date _____

Disqualify? YES NO

Fund? YES NO

Funded previously for HCD YES NO

Amt Requested? _____

Amt Recommended? _____

ASSESSMENT OF NEED

Total points - 15/5

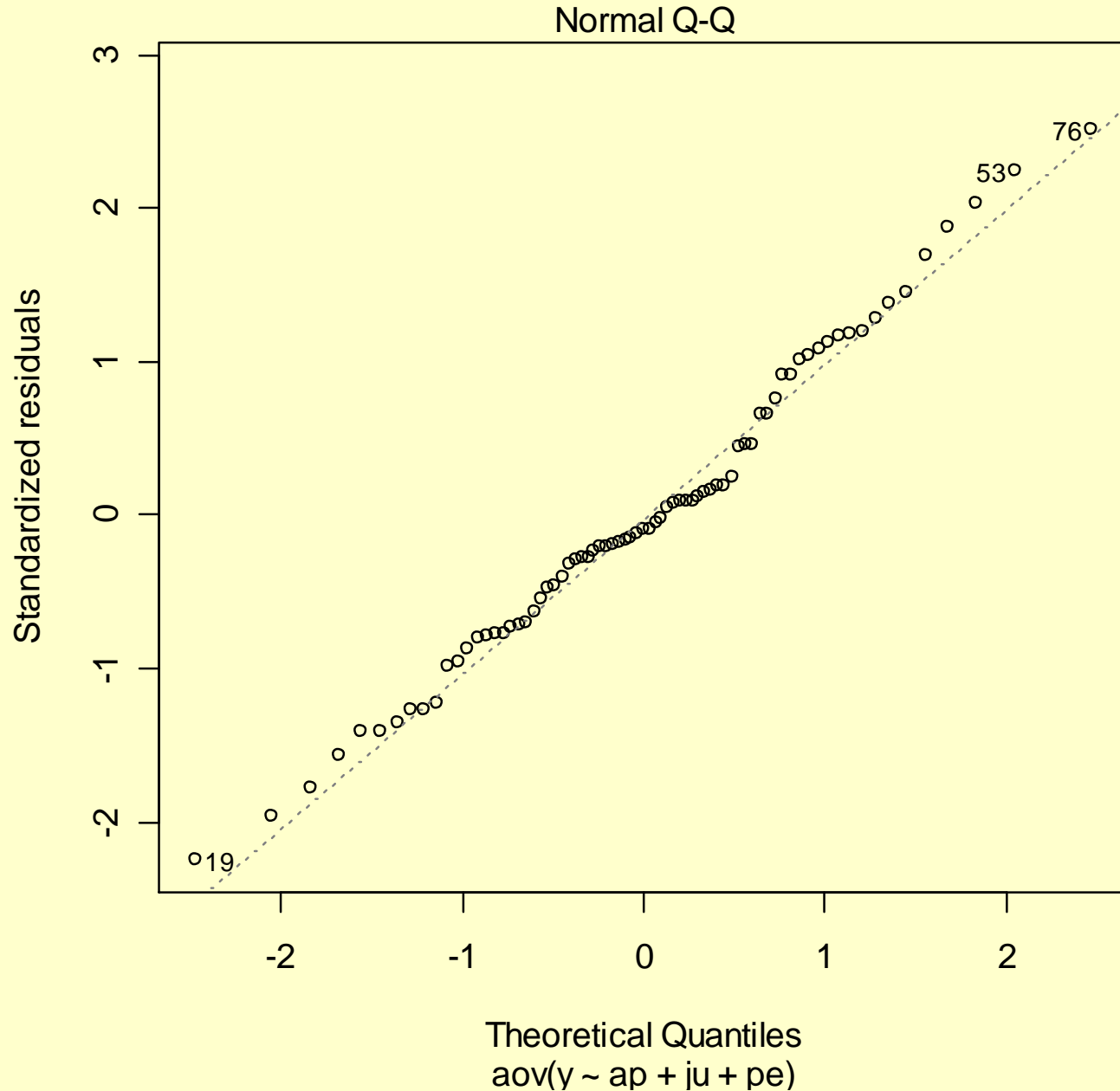
_____ Describe **nature and character of community** related to the proposed project including relevant County, population, median income, health information, and other relevant data. **(4 points)**

_____ Assets - community capacity to support the project. **(4 points)**

Potential Examples:

- any existing physical activity or wellness groups and their activities.
(i.e. local conditions that address wellness, active living, food access and

QQ plot of evaluation design with many df for error revealed approximate normality of scores.



2009 Mini-Grants for Community Fitness & Sports were assessed in a Generalized Chain Blocks design.

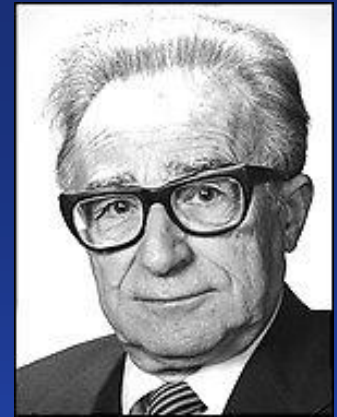
Karin
Mille



- Grants from \$2500 to \$10,000 were to be awarded.
- Grant applicants required to attend Nutrition and Fitness Leadership Conference.
- After screening, 32 grant applications to be reviewed, scored, and ranked.
- 8 Reviewers agreed to each review 8 grants

GCB designs might be termed
“designs of even numbers”.

John
Mandel



v = #treatments must be even

k = #rows (treatments/ block) must be even

$r = 2$ = #replicates of each treatment

(v, k determine design since $b*k = vr = v*2$)

b = #blocks is even

$n = b*k$ = #measurements is even

Any v , GCB for all k that are even divisors of v

Single method to generate all GCB designs →

G.C.B.

$v = 32$

$k = 8$

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	29	29	30	31	32

G.C.B. $v = 32$ $k = 8$

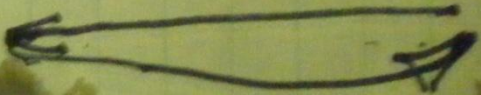
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	29	29	30	31	32

13	14	15	16
21	22	23	24
29	30	31	32
5	6	7	8

G.C.B. $v = 32$ $k = 8$

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

13	14	15	16	2	3	4	1
21	22	23	24	10	11	12	13
29	30	31	32	18	19	20	17
5	6	7	8	26	27	28	25





```
> gcbdesign(32, 8)      # generation order
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]     1     2     3     4     5     6     7     8
[2,]     9    10    11    12    13    14    15    16
[3,]    17    18    19    20    21    22    23    24
[4,]    25    26    27    28    29    30    31    32
[5,]    13    14    15    16     2     3     4     1
[6,]    21    22    23    24    10    11    12     9
[7,]    29    30    31    32    18    19    20    17
[8,]     5     6     7     8    26    27    28    25
> |
```

```
> gcbdesign(v=32, k=8, chain="col")
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	1	5	2	6	3	7	4	8
[2,]	9	13	10	14	11	15	12	16
[3,]	17	21	18	22	19	23	20	24
[4,]	25	29	26	30	27	31	28	32
[5,]	13	2	14	3	15	4	16	1
[6,]	21	10	22	11	23	12	24	9
[7,]	29	18	30	19	31	20	32	17
[8,]	5	26	6	27	7	28	8	25

```
> gcbdesign(v=32, k=8, chain="row")
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	1	2	3	4	5	6	7	8
[2,]	13	14	15	16	2	3	4	1
[3,]	9	10	11	12	13	14	15	16
[4,]	21	22	23	24	10	11	12	9
[5,]	17	18	19	20	21	22	23	24
[6,]	29	30	31	32	18	19	20	17
[7,]	25	26	27	28	29	30	31	32
[8,]	5	6	7	8	26	27	28	25

```
> |
```


As a check, we generate all GCB designs given in J Mandel's 1954 Biometrics paper

```
gcbMandel = function() {  
  cat("John Mandel, 1954, page 256\n\n")  
  cat("Interchanging Rows and columns is also GCB\n\n")  
  print( gcbdesign(8,4) )  
  print( gcbdesign(12,4) )  
  print( gcbdesign(18,6) )  
  print( gcbdesign(24,6) )  
  print( gcbdesign(20,4) )  
  print( gcbdesign(30,6) )  
} #End function gcbMandel
```


	[,1]	[,2]	[,3]	[,4]
[1,]	1	2	3	4
[2,]	5	6	7	8
[3,]	7	8	2	1
[4,]	3	4	6	5

Mandel 1954

First 3 designs

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	1	2	3	4	5	6
[2,]	7	8	9	10	11	12
[3,]	10	11	12	2	3	1
[4,]	4	5	6	8	9	7

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	1	2	3	4	5	6
[2,]	7	8	9	10	11	12
[3,]	13	14	15	16	17	18
[4,]	10	11	12	2	3	1
[5,]	16	17	18	8	9	7
[6,]	4	5	6	14	15	13

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	1	2	3	4	5	6	7	8
[2,]	9	10	11	12	13	14	15	16
[3,]	17	18	19	20	21	22	23	24
[4,]	13	14	15	16	2	3	4	1
[5,]	21	22	23	24	10	11	12	9
[6,]	5	6	7	8	18	19	20	17

Mandel 1954
 Designs 4-6

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	1	2	3	4	5	6	7	8	9	10
[2,]	11	12	13	14	15	16	17	18	19	20
[3,]	16	17	18	19	20	2	3	4	5	1
[4,]	6	7	8	9	10	12	13	14	15	11

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	1	2	3	4	5	6	7	8	9	10
[2,]	11	12	13	14	15	16	17	18	19	20
[3,]	21	22	23	24	25	26	27	28	29	30
[4,]	16	17	18	19	20	2	3	4	5	1
[5,]	26	27	28	29	30	12	13	14	15	11
[6,]	6	7	8	9	10	22	23	24	25	21

Correctness was checked with two GCB examples and three computational methods

J. Mandel 1954 treadwear ($v=8, k=4$) and most recent mini-grant data ($v=32, k=8$).

R function to compute J. Mandel's analysis of GCB

R analysis tools: `lm`, `glm`, `aov`,
with `allEffects()` from `effects` package (J. Fox)
(Note: `model.tables()` does not give correct adjusted means for GCB, lack of balance?.)

OpenBUGS/WinBUGS Bayesian model

This GCB design was effective in adjusting for differences among reviewers (blocks).

```
> lm1 <- lm(y ~ blocks + rows + trt)
```

```
> anova(lm1)
```

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
blocks	7	4034.3	576.32	4.9517	0.0028931	**
rows	7	2421.3	345.89	2.9719	0.0295168	*
trt	31	16707.2	538.94	4.6305	0.0006009	***
Residuals	18	2095.0	116.39			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
0.1 ' ' 1

The “correct” adjusted treatment means (fit) were computed *via* the effects package.

```
> m <- data.frame(allEffects(lm1)$trt)

> m <- m[ order(m$fit,decreasing=TRUE), ]
> print(d,row.names=F)
```

trt	fit	se	lower	upper
4	99.1875	10.44579	77.24170	121.1333
13	99.0625	10.44579	77.11670	121.0083
27	96.3125	10.44579	74.36670	118.2583
25	96.1875	10.44579	74.24170	118.1333
5	94.9375	10.44579	72.99170	116.8833
1	93.5625	10.44579	71.61670	115.5083
.
9	51.1875	10.44579	29.24170	73.1333
20	50.1875	10.44579	28.24170	72.1333
3	48.6875	10.44579	26.74170	70.6333
22	44.6875	10.44579	22.74170	66.6333
2	39.8125	10.44579	17.86670	61.7583
6	39.6875	10.44579	17.74170	61.6333

OpenBUGS, with R package Brugs, provided a check on the adj. treatment means.

```
model  #Main techniques in declarative script
  {for( i in 1 : N ) {
    y[i] ~ dnorm(mu[i],tau)

    mu[i] <- gm + block[ blockno[i]] +
              row[rowno[i]] + treat[treatno[i]]
  }

# Parameter constraints
  block[b] <- -sum( block[1:(b-1)] )
  row[k] <- -sum( row[1:(k-1)] )
  treat[v] <- -sum( treat[1:(v-1)] )

  for (i in 1:v) {
    adjmean[i] <- gm + treat[i]
  }
}
```

BRugs runs OpenBUGS within R by referencing files of model, data, and inits (initial values).

```
library(BRugs)
```

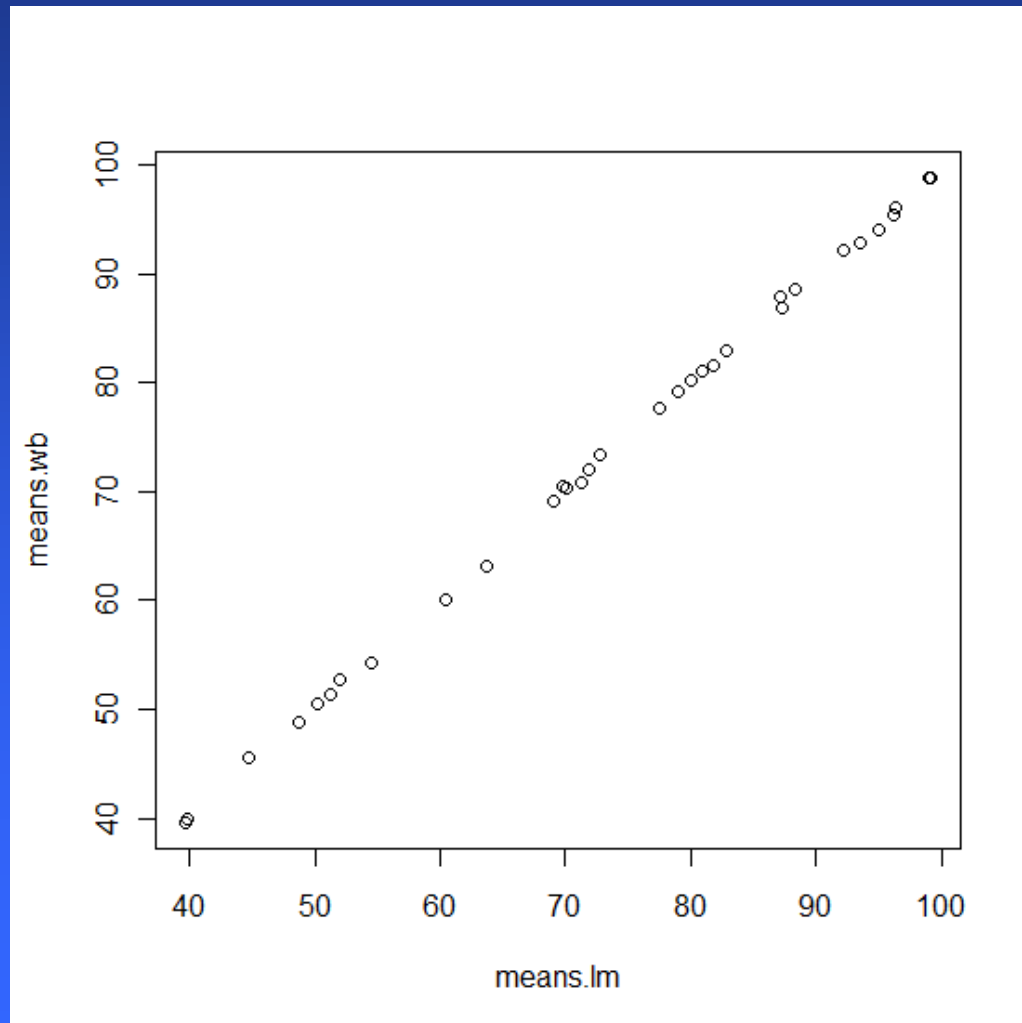
```
modelCheck("wbmodel.txt") # check model  
modelData("wbdata.txt")  # load data  
modelCompile(numChains=1) # compile model
```

```
modelInits("wbinit.txt")  
modelGenInits()          # Any var not in inits.
```

```
modelUpdate(1000)        # burn in  
samplesSet("adjmean")    # set vars to monitor  
modelUpdate(5000)        # Gibbs sampling ....
```

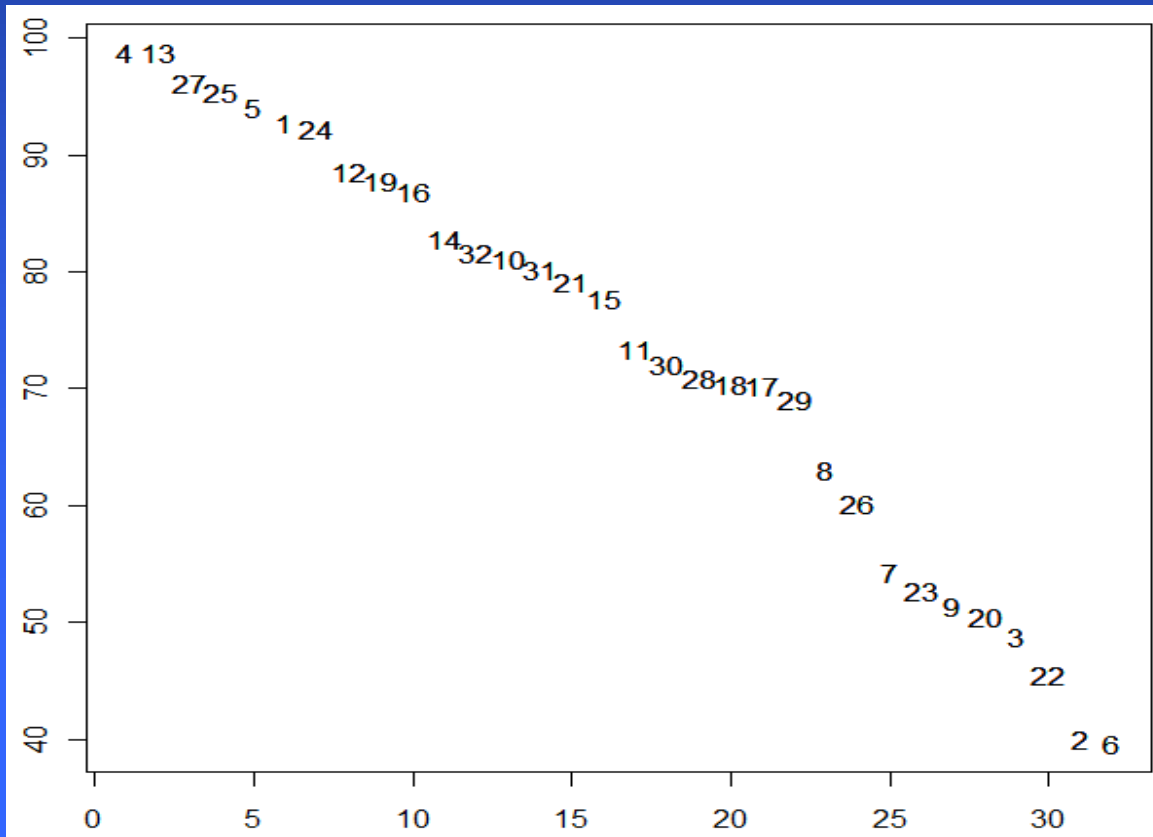
```
stats<- samplesStats("adjmean")
```

```
> means.wb = as.vector(stats$mean)
> means.lm = as.numeric(effect("trt", lm1)$fit)
> plot(means.lm, means.wb)
```




```
trtno    <- 1:length(stats$mean)
results  <- data.frame(trtno, stats$mean)
names(results) <- c("trtno", "adjm")
downorder <- order(results$adjm, decreasing=TRUE)
results   <- results[downorder, ]
```

```
plot(results$adjm, pch="")
text(1:32, results$adjm, results$trtno)
```



The grant review design was effective, organizationally and statistically.

- The adjustment for reviewers reduced variation due to differences of scoring level among reviewers
- Period effects did not appear to be sizeable, though the analysis of variance indicated that the effects were real.
- Mini-grants were then awarded on the basis of the adjusted application mean scores, presented in decreasing order.

R functions can help the design process.

`gcbdesign(v,k)` generates design for any v, k

`gcbindex(v1,v2)` lists GCB for range of # treatments
(handout)

`gcbgroups()` lists treatment groups and distances

`gcb.compare()` multipliers of individual error var
to obtain variances of differences of treat effects

`gcb.eff()` D-efficiency of GCB

Functions for GCB may encourage writing utility functions for other design series.

`gcb.as.matrix(blocks, rows, trt)` converts to matrix
`gcb.as.df (des)` converts to data frame

`is.gcb(des)` TRUE if a generalized block design.

`gcbrandom(des)` randomizes rows,cols of GCG
`gcborder(des)` unrandomizes GCB into gen order.

`gcbenter(v, k)` columnar data entry using `fix()`
`gcbreenter(v,k,d)` correct/continue data entry

Precision of treatment comparisons depends upon separation in the chain.

```
gcb.compare(v=24, k=8)
```

	v1	v2	varfactor
1	1	2	1.416667
2	1	3	1.416667
3	1	4	1.416667
4	1	5	1.750000
5	1	6	1.416667
6	1	7	1.666667
7	1	8	2.083333
8	1	9	2.083333
9	1	10	1.416667

and so on.

D-efficiency may be helpful in comparing GCB designs (see handout).

```
gcb.eff = function(v,k)    {  
  
  des = gcbdesign(v,k) # generate design  
  desdf = gcb.as.df(des) # convert to data frame  
  
  x = model.matrix(~factor(blocks) +  
    factor(rows) + factor(trt), data=desdf )  
  xpx = t(x) %*% x  
  
  D = 100* 1/(N*det( solve(xpx) )^(1/p)) #OPTEX  
  #D = 100* det( solve(xpx) )^(1/p) / N #JMP  
  return(D)  
} #End gcb.eff
```

In awarding grants we have learned:

- Secure the most reliable reviewers
- Have 1-2 reviewers on reserve.
- Reviewers must attend Orientation
- Consider regionalizing RFA to reduce number of Grant Apps. to be ranked together (may require political savvy).
- Require only title page to have applicant name in order to blind scoring if possible.
- Pre-test the scoring methodology.
- Avoid huge meetings. Consider Exp. Design.
- DOUBLE CHECK EVERYTHING.
- **Amat victoria curam** - *Victory favors those who prepare*

Generalized Chain Block Designs

Further work

Prepare R package of gcb functions.

Simple chain block design analysis & example

Compose plots helpful for GCB experiments

Contrasts of adjusted treatment means;
combination treatments, other techniques.

Investigate Augmented designs wherein
treatments replicated more than twice.

Saving study when one response is bad.

Experimental Design References

Bose, R.C.; Clatworthy, W.H.; and Shrikhande, S.S. (1939). Partially balanced incomplete block designs. *Sankhya* 4, pp 337-372.

Cochran, W.G. and Cox, G.M. (1957). *Experimental Designs*. (John Wiley & Sons; New York).

Mandel, J. (1954). Chain block designs with two-way elimination of heterogeneity. *Biometrics* 10: 251-272.

Youden, W.J. and Connor, W.S. (1953). The chain block design. *Biometrics* 9, 127-140.