

Using the BsMD Package for Bayesian Screening and Model Discrimination*

Ernesto Barrios
University of Wisconsin-Madison

March, 2005

Contents

1	Introduction	2
2	Screening Designs	2
2.1	Daniel Plots	3
2.1.1	Box et al. 1986: Example 1	4
2.1.2	Box et al. 1986: Example 3	4
2.2	Lenth Plots	5
2.2.1	Box et al. 1986: Example 2	6
2.2.2	Box et al. 1986: Example 4	7
3	Bayesian Screening	8
3.1	Fractional Factorial Designs	8
3.1.1	Box et al. 1986: Example 1	9
3.1.2	Box et al. 1986: Example 4	10
3.2	Plackett-Burman Designs	12
3.2.1	Box et al. 1993: Example 1	12
3.2.2	Box et al. 1993: Example 2	15
3.3	Extra Runs	16
3.3.1	Box et al. 1993: Example 3	16
4	Model Discrimination	19
4.1	4-run Follow-Up Experiments	20
4.1.1	Meyer et al. 1996: Example 1	20
4.1.2	Meyer et al. 1996: Example 2	21
4.2	One-run-at-a-time Experiments	25
4.2.1	Meyer et al. 1996: Example 2	25
5	Summary	27

*BsMD current version: 0.6-5.

References	27
Appendix	28

1 Introduction

Screening experiments are employed at the initial stages of investigation to discriminate, among many factors, those with potential effect over the response under study. It is common in screening studies to use most of the observations estimating different contrasts, leaving only a few or even no degrees of freedom at all to estimate the experiment standard error. Under these conditions it is not possible to assess the statistical significance of the estimated contrast effects. Some procedures, for example, the analysis of the normal plot of the effects, have been developed to overcome this situation.

BsMD package includes a set of functions useful for factor screening in unreplicated factorial experiments. Some of the functions were written originally for **S**, then adapted for **S-PLUS** and now for **R**. Functions for Bayesian screening and model discrimination follow-up designs are based on Daniel Meyer's **mdopt** FORTRAN bundle (Meyer, 1996). The programs were modified and converted to subroutines to be called from **R** functions.

This document is organized in three sections: Screening Designs, Bayesian Screening, and Model Discrimination, with the references to the articles as subsections to indicate the sources of the examples presented. All the examples in Box and Meyer (1986, 1993) and Meyer, Steinberg, and Box (1996) are worked out and the code displayed in its totality to show the use of the functions in the **BsMD** package. The detailed discussion of the examples and the theory behind them is left to the original papers. Details of the **BsMD** functions are contained to their help pages.

2 Screening Designs

In screening experiments, *factor sparsity* is usually assumed. That is, from all factors considered in the experiment only a few of these will actually affect the response. (See for example, Box and Meyer (1986), sec. 1.) Based on this sparsity hypothesis various procedures have been developed to identify such active factors. Some of these procedures are included in the **BsMD** package: **DanielPlot** (Normal Plot of Effects), **LenthPlot** (based on a robust estimation of the standard error of the contrasts), and **BsProb** for Bayesian screening. See the references for details on the theory of the procedures. The data set used in the examples of this section is from Box and Meyer (1986). They represent four different experiments: *log drill advance*, *tensile strength*, *shrinkage* and *yield of isatin* with responses denoted by y_1, \dots, y_4 and different design factors. The estimable contrasts are denoted by X_1, \dots, X_{15} . The design matrix and responses are presented next.

```
> options(width = 80)
> library(BsMD)
> data(BM86.data, package = "BsMD")
> print(BM86.data)
```

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	y1	y2	y3	y4
1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1	0.23	43.7	14.0	0.08
2	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1	0.30	40.2	16.8	0.04

3	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1	0.52	42.4	15.0	0.53
4	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	0.54	44.7	15.4	0.43
5	-1	-1	1	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	0.70	42.4	27.6	0.31
6	1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	0.76	45.9	24.0	0.09
7	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1.00	42.2	27.4	0.12
8	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	0.96	40.6	22.6	0.36
9	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1	0.32	42.4	22.3	0.79
10	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1	0.39	45.5	17.1	0.68
11	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	0.61	43.6	21.5	0.73
12	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	0.66	40.6	17.5	0.08
13	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	0.89	44.0	15.9	0.77
14	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	0.97	40.2	21.9	0.38
15	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1.07	42.5	16.7	0.49
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.21	46.5	20.3	0.23

Saturated linear models for each of the responses are fitted and the estimated coefficients are presented in the table below. The `lm` calls, not displayed here, produce the `advance.lm`, ..., `yield.lm` objects used in the next subsections.

	advance	shrinkage	strength	yield
(Intercept)	0.70	42.96	19.75	0.38
X1	0.03	0.06	-0.30	-0.10
X2	0.13	-0.07	-0.20	-0.01
X3	-0.01	0.15	-0.30	0.00
X4	0.25	0.08	2.30	-0.04
X5	0.00	0.20	0.45	0.02
X6	-0.01	-0.01	-0.10	-0.03
X7	0.00	0.19	-0.15	0.07
X8	0.07	0.20	-0.60	0.14
X9	0.01	-0.03	0.35	-0.08
X10	0.00	0.21	0.05	-0.13
X11	0.01	0.06	0.15	-0.05
X12	0.02	0.06	-2.75	-0.01
X13	0.01	-0.19	1.90	0.00
X14	-0.01	1.07	0.05	0.06
X15	0.01	1.55	-0.30	0.01

For each of the experiments the 16 runs are used on the estimation of the 15 contrasts and the constant term. Thus the need of graphical aims to determine which are likely active contrasts.

2.1 Daniel Plots

Daniel plots, known as normal plot of effects, arrange the estimated factor effects in a normal probability plot; those factors “out of the straight line” are identified as potentially active factors. See for example, Daniel (1976) for different applications and interpretations.

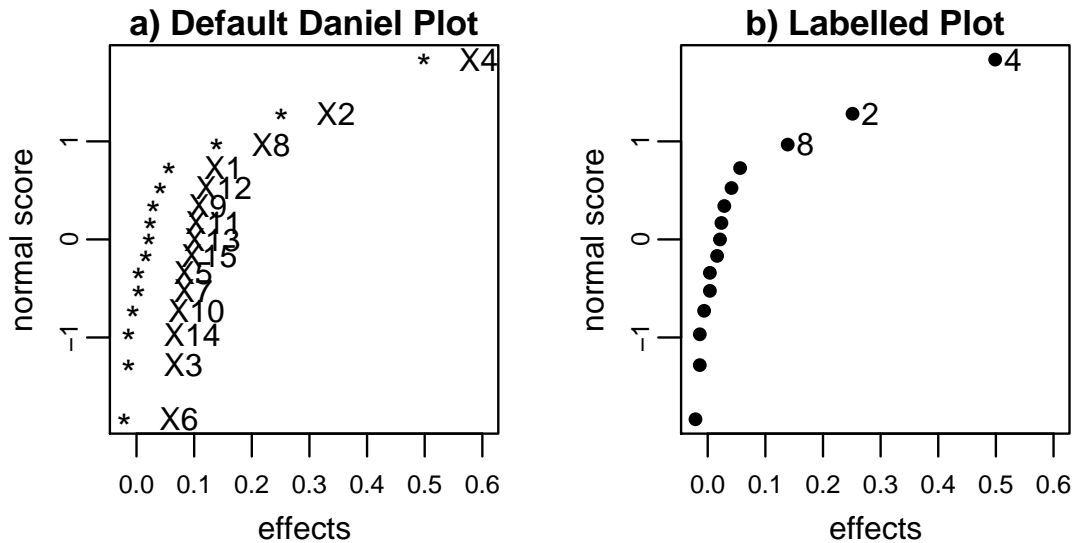
`DanielPlot` produces normal plot of effects. The main argument of the function is an `lm` object, say, `lm.obj`. The function removes the constant term (`Intercept`) if it is in the model. Factor

effects, assumed as $2 \times \text{coef}(\text{lm.obj})$ are displayed using the `qqnorm` function. See the help pages for details.

2.1.1 Box et al. 1986: Example 1

By default `DanielPlot` labels all the effects, as show in figure a). This example shows how to label only some particular factors for clarity, as exhibited in figure b). The corresponding linear model `advance.lm` was already fitted at the beginning of the section.

```
> par(mfrow = c(1, 2), mar = c(3, 3, 1, 1), mgp = c(1.5, 0.5, 0),
+     oma = c(0, 0, 0, 0), xpd = TRUE, pty = "s", cex.axis = 0.7,
+     cex.lab = 0.8, cex.main = 0.9)
> DanielPlot(advance.lm, cex.pch = 0.8, main = "a) Default Daniel Plot")
> DanielPlot(advance.lm, cex.pch = 0.8, main = "b) Labelled Plot",
+     pch = 20, faclab = list(idx = c(2, 4, 8), lab = c(" 2", " 4",
+     " 8")))
```



2.1.2 Box et al. 1986: Example 3

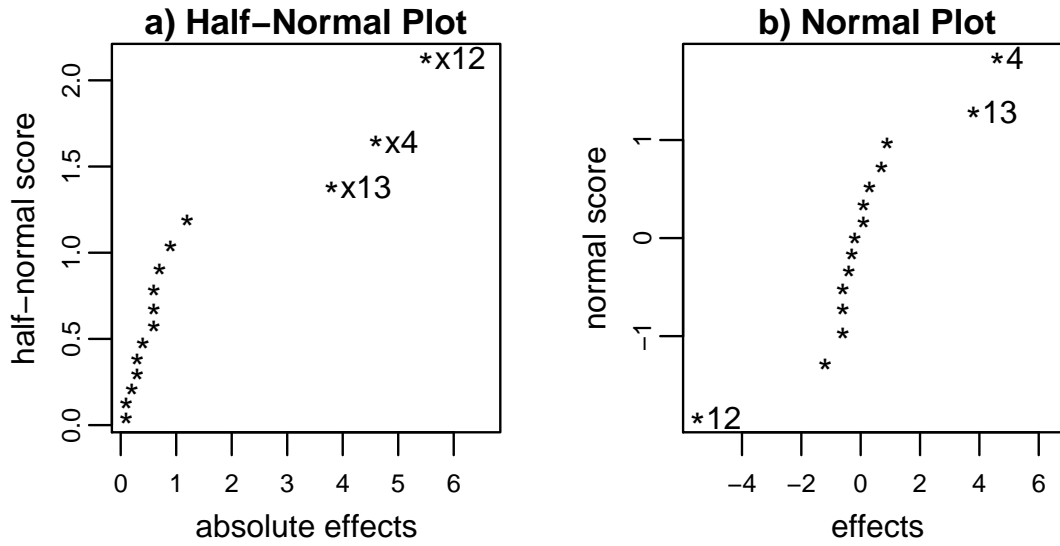
Some people prefer the use of half-normal plots. These plots are similar to the normal plots but instead of the signed effects absolute values of the effects are displayed. There are some advantages and disadvantages using one or the other. See for example, Daniel (1976, chap. 7.6).

Figure a) depicts the half-normal plot of the effects for the strength response (`y3`). `DanielPlot` has the option to generate half-normal plots (`half=TRUE`). The corresponding normal plot of signed effects is presented in figure b) below.

```

> par(mfrow = c(1, 2), mar = c(3, 3, 1, 1), mgp = c(1.5, 0.5, 0),
+     oma = c(0, 0, 0, 0), xpd = TRUE, pty = "s", cex.axis = 0.7,
+     cex.lab = 0.8, cex.main = 0.9)
> DanielPlot(strength.lm, half = TRUE, cex.pch = 0.8, main = "a) Half-Normal Plot",
+     faclab = list(idx = c(4, 12, 13), lab = c(" x4", " x12",
+     " x13")))
> DanielPlot(strength.lm, main = "b) Normal Plot", faclab = list(idx = c(4,
+     12, 13), lab = c(" 4", " 12", " 13")))

```



2.2 Lenth Plots

Lenth's method for factor effects assessment is based on factor sparsity too. For an unreplicated factorial design let c_1, \dots, c_m be the estimated contrasts and approximate the standard error by $s_0 = 1.5 \times \text{median } |c_i|$. Then the author defines the *pseudo standard error* by

$$\text{PSE} = 1.5 \times \text{median}_{|c_j| < 2.5s_0} |c_j|$$

and the 95% *margin of error* by

$$\text{ME} = t_{0.975, d} \times \text{PSE}$$

where $t_{0.975, d}$ is the .975th quantile of the t distribution with $d = m/3$ degrees of freedom. The 95% *simultaneous margin of error* (SME) is defined for simultaneous inference on all the contrast and is given by

$$\text{SME} = t_{\gamma, d} \times \text{PSE}$$

where $\gamma = (1 + 0.95^{1/m})/2$. See Lenth (1989), for details.

The `LenthPlot` function displays the factor effects and the SE and SME limits. Spikes instead of the barplot used originally by Lenth are employed to represent the factor effects. As in `DanielPlot`, the main argument for the function is a `lm` object, and `2*coef(lm.obj)` is displayed.

2.2.1 Box et al. 1986: Example 2

Figure a) below shows the default plot produced by `LenthPlot`. The SE and MSE limits at a 95% confidence level ($\alpha = 0.05$) are displayed by default. Figure b) shows Lenth's plot for the same experiment using $\alpha = 0.01$, locating the labels of SME and ME close to the vertical axis and labelling the contrast effects X_{14} and X_{15} as P and $-M$, for period and material respectively and accordingly to Lenth's paper. Note that the effects are considered as 2 times the coefficients b .

```
> par(mfrow = c(1, 2), mar = c(4, 4, 1, 1), mgp = c(1.5, 0.5, 0),
+     oma = c(0, 0, 0, 0), xpd = TRUE, pty = "s", cex.axis = 0.7,
+     cex.lab = 0.8, cex.main = 0.9)
> LenthPlot(shrinkage.lm)
```

alpha	PSE	ME	SME
0.0500000	0.2250000	0.5783809	1.1741965

```
> title("a) Default Lenth Plot")

NULL

> b <- coef(shrinkage.lm)[-1]
> LenthPlot(shrinkage.lm, alpha = 0.01, adj = 0.2)
```

alpha	PSE	ME	SME
0.0100000	0.2250000	0.9072322	1.6855749

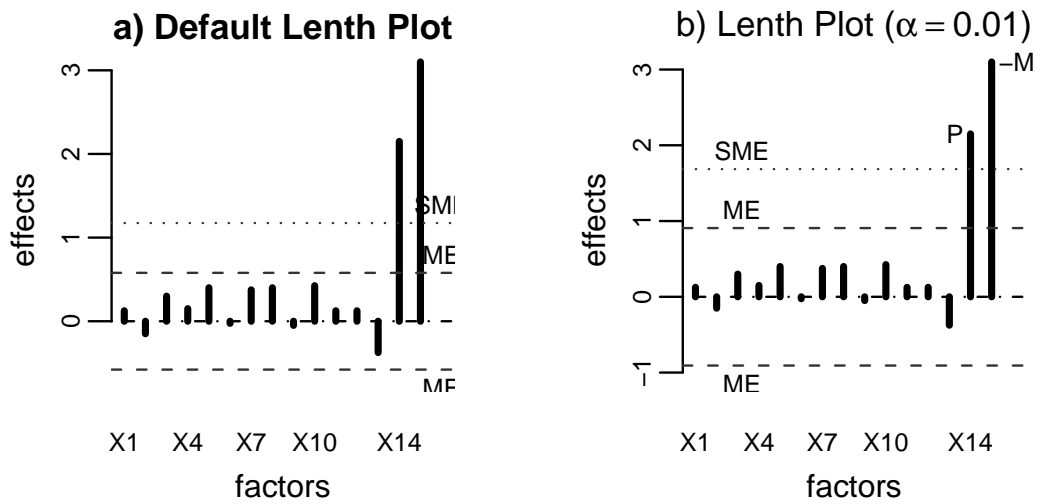
```
> title(substitute("b) Lenth Plot (" * a * ")"), list(a = quote(alpha ==
+     0.01))))

NULL

> text(14, 2 * b[14], "P ", adj = 1, cex = 0.7)

NULL

> text(15, 2 * b[15], "-M", adj = 0, cex = 0.7)
```

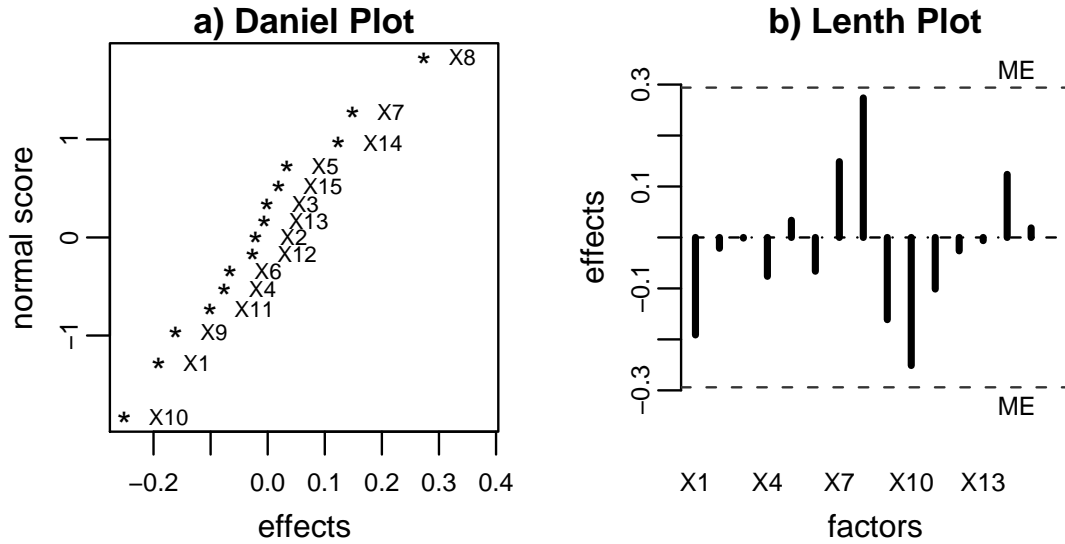


2.2.2 Box et al. 1986: Example 4

This example exhibits the Daniel and Lenth plots for the isatin data, originally presented by Davis and co-authors in 1954 and discussed in the Box and Meyer paper (p. 16–17). As can be seen in the figures below, it is not clear which contrasts may be active. For example, in Lenth's plot none of the effects goes beyond the margin of error ME, thus the SME limits are not displayed. The corresponding Bayes plot is presented in the next section.

```
> par(mfrow = c(1, 2), mar = c(3, 3, 1, 1), mgp = c(1.5, 0.5, 0),
+     oma = c(0, 0, 0, 0), pty = "s", cex.axis = 0.7, cex.lab = 0.8,
+     cex.main = 0.9)
> DanielPlot(yield.lm, cex.pch = 0.6, main = "a) Daniel Plot")
> LenthPlot(yield.lm, alpha = 0.05, xlab = "factors", adj = 0.9,
+     main = "b) Lenth Plot")
```

alpha	PSE	ME	SME
0.0500000	0.1143750	0.2940103	0.5968832



3 Bayesian Screening

Box and Meyer Bayesian screening is also based on the factor sparsity hypothesis. For the linear model $y = X\beta + \epsilon$, the procedure assigns to each of the β_i independent prior normal distributions $N(0, \gamma^2 \sigma^2)$, where σ^2 is the variance of the error and γ^2 is the magnitude of the effect relative to the experimental noise. The factor sparsity assumption is brought into the procedure assigning a prior probability π to any factor of being active, and $1 - \pi$ to the factor of being inert. Models M_l for all-subsets of factors (main effects and interactions) are constructed and their posterior probabilities calculated. Marginal factor posterior probabilities p_i are computed and displayed. Those contrasts or factor effects with higher probabilities are identified as potentially active. See Box and Meyer (1986, 1993) for explanation and details of the procedure.

The `BsProb` function computes the posterior probabilities for Bayesian screening. The function calls the `bs` FORTRAN subroutine, a modification of the `mbcqp15.f` program included in the `mdopt` bundle. The complete output of the program is saved in the working directory as `BsPrint.out`. The file is overwritten if it already exists. Thus, rename the `BsPrint.out` file after each call to `BsProb` if you want to keep the complete output. Note however, that most of the output is included in the `BsProb`'s output list. This is a list of class `BsProb` with methods functions for `print`, `plot` and `summary`.

3.1 Fractional Factorial Designs

Bayesian screening was presented by Box and Meyer in their 1986 and 1993 papers. The former refers to 2-level orthogonal designs while the latter refer to general designs. The distinction is

important since in the case of 2-level orthogonal designs some factorization is possible that allows the calculation of the marginal probabilities without summing over all-subsets models' probabilities. This situation is explained in the 1986 paper, where α and k are used instead of the π and γ described at the beginning of the section. Their correspondence is $\alpha = \pi$, and $k^2 = n\gamma^2 + 1$, where n is the number of runs in the design. The function is written for the general case and arguments \mathbf{p} and \mathbf{g} (for π and γ) should be provided. In the mentioned paper the authors estimated α and k for a number of published examples. They found $.13 \leq \hat{\alpha} \leq .27$, and $2.7 \leq \hat{k} \leq 27$. Average values of $\alpha = 0.20$ ($= \pi$) and $k = 10$ ($\gamma = 2.49$) are used in the examples.

3.1.1 Box et al. 1986: Example 1

This example exhibits most of the output of the `BsProb` function. The design matrix and response vector, the 15 contrasts and 5 models posterior probabilities are printed. As mentioned before, $\mathbf{g}=2.49$ corresponds to $k = 10$ used in the paper. Note that all possible 2^{15} factor combinations were used to construct the `totMod=32768` estimated models. Only the top `nMod=5` are displayed. See the `BsProb` help pages for details. Figures below show the Bayes plot (a) and Daniel plot (b) for the estimated effects. In this case both procedures clearly identify x_2 , x_4 , and x_8 as active contrasts.

```
> par(mfrow = c(1, 2), mar = c(3, 3, 1, 1), mgp = c(1.5, 0.5, 0),
+     oma = c(0, 0, 0, 0), pty = "s", cex.axis = 0.7, cex.lab = 0.8,
+     cex.main = 0.9)
> X <- as.matrix(BM86.data[, 1:15])
> y <- BM86.data[, 16]
> advance.BsProb <- BsProb(X = X, y = y, blk = 0, mFac = 15, mInt = 1,
+   p = 0.2, g = 2.49, ng = 1, nMod = 10)
> print(advance.BsProb, X = FALSE, resp = FALSE, nMod = 5)
```

Calculations:

nRun	nFac	nBlk	mFac	mInt	p	g	totMod
16.00	15.00	0.00	15.00	1.00	0.20	2.49	32768.00

Output file: BsPrint.out

Factor probabilities:

	Factor	Code	Prob
1	none	none	0.000
2	X1	x1	0.240
3	X2	x2	1.000
4	X3	x3	0.028
5	X4	x4	1.000
6	X5	x5	0.025
7	X6	x6	0.034
8	X7	x7	0.025
9	X8	x8	0.983
10	X9	x9	0.046
11	X10	x10	0.025

```

12   X11  x11 0.037
13   X12  x12 0.091
14   X13  x13 0.034
15   X14  x14 0.028
16   X15  x15 0.030

```

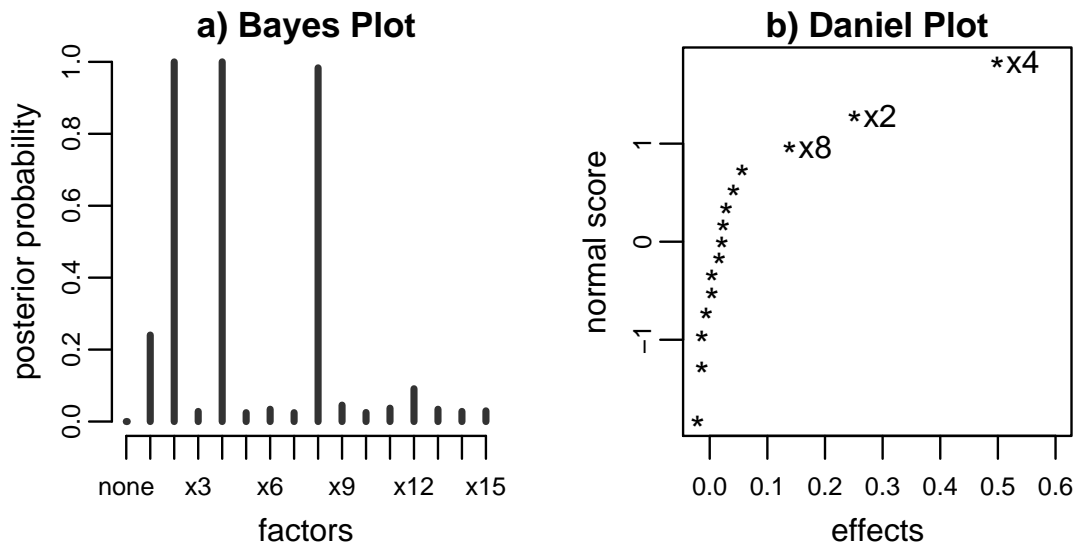
Model probabilities:

	Prob	Sigma2	NumFac	Factors
M1	0.504	0.003	3	2,4,8
M2	0.148	0.002	4	1,2,4,8
M3	0.043	0.003	4	2,4,8,12
M4	0.022	0.003	4	2,4,8,9
M5	0.022	0.002	5	1,2,4,8,12

```

> plot(advance.BsProb, main = "a) Bayes Plot")
> DanielPlot(advance.lm, cex.pch = 0.6, main = "b) Daniel Plot",
+   faclab = list(idx = c(2, 4, 8), lab = c(" x2", " x4", " x8")))

```



3.1.2 Box et al. 1986: Example 4

As mentioned in section 2.2.2, in the isatin data example active contrasts, if present, are not easily identified by Daniel or Lenth's plot. This situation is reflected in the sensitivity of the Bayes procedure to the value of γ . Different values for k (γ) can be provided to the **BsProb** function and the respective factor posterior probabilities computed. The range of such probabilities is plotted as

stacked spikes. This feature is useful in data analysis. See next subsection for further explanation. In the call of the function `BsProb`, `g=c(1.22,3.74)` and `ng=10` indicate that the calculation of the marginal posterior probabilities is done for 10 equally spaced values of γ in the range (1.22,3.74) corresponding to the range of k between 5 and 15 used in the paper. The sensitivity of the posterior probabilities to various values of γ is exhibited in figure a) below. The large ranges displayed by some of the contrasts is an indication that no reliable inference is possible to draw from the data.

```
> par(mfrow = c(1, 2), mar = c(3, 3, 1, 1), mgp = c(1.5, 0.5, 0),
+     oma = c(0, 0, 0, 0), pty = "s", cex.axis = 0.7, cex.lab = 0.8,
+     cex.main = 0.9)
> X <- as.matrix(BM86.data[, 1:15])
> y <- BM86.data[, 19]
> yield.BsProb <- BsProb(X = X, y = y, blk = 0, mFac = 15, mInt = 1,
+   p = 0.2, g = c(1.22, 3.74), ng = 10, nMod = 10)
> summary(yield.BsProb)
```

Calculations:

	nRun	nFac	nBlk	mFac	mInt	p	g[1]	g[10]
	16.00	15.00	0.00	15.00	1.00	0.20	1.22	3.74
totMod	32768.00							

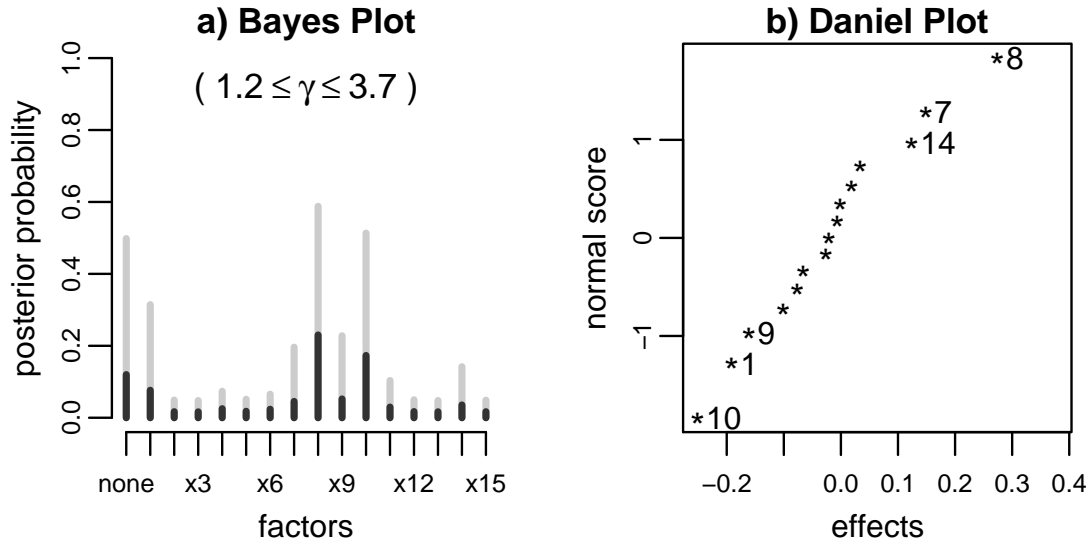
Posterior probabilities for each gamma value:

	1	2	3	4	5	6	7	8	9	10
gamma	1.220	1.500	1.780	2.060	2.340	2.620	2.900	3.180	3.460	3.740
none	0.120	0.167	0.218	0.268	0.316	0.360	0.400	0.436	0.469	0.498
x1	0.314	0.271	0.228	0.190	0.159	0.134	0.115	0.099	0.086	0.076
x2	0.049	0.041	0.035	0.030	0.027	0.024	0.022	0.020	0.018	0.017
x3	0.048	0.039	0.034	0.029	0.026	0.023	0.021	0.019	0.018	0.016
x4	0.074	0.066	0.059	0.053	0.048	0.042	0.037	0.032	0.028	0.025
x5	0.051	0.043	0.037	0.032	0.028	0.026	0.023	0.021	0.019	0.018
x6	0.066	0.057	0.051	0.047	0.042	0.038	0.034	0.030	0.027	0.024
x7	0.196	0.170	0.143	0.119	0.099	0.083	0.070	0.060	0.052	0.045
x8	0.588	0.531	0.473	0.420	0.374	0.335	0.302	0.274	0.250	0.230
x9	0.228	0.197	0.164	0.136	0.113	0.095	0.080	0.069	0.060	0.052
x10	0.513	0.456	0.399	0.348	0.304	0.267	0.237	0.212	0.191	0.173
x11	0.104	0.093	0.082	0.071	0.061	0.052	0.045	0.039	0.034	0.030
x12	0.050	0.041	0.035	0.031	0.027	0.024	0.022	0.020	0.019	0.017
x13	0.048	0.040	0.034	0.029	0.026	0.023	0.021	0.019	0.018	0.016
x14	0.142	0.125	0.107	0.091	0.076	0.064	0.055	0.047	0.041	0.035
x15	0.049	0.040	0.034	0.030	0.026	0.024	0.021	0.020	0.018	0.017

```
> plot(yield.BsProb, main = "a) Bayes Plot")
> title(substitute("( " * g1 * " " * g2 * " )", list(g1 = quote(1.2 <=
+   gamma), g2 = quote(" " <= 3.7))), line = -1)
```

NULL

```
> DanielPlot(yield.lm, cex.pch = 0.6, main = "b) Daniel Plot",
+   faclab = list(idx = c(1, 7, 8, 9, 10, 14), lab = paste(" ",
+   c(1, 7, 8, 9, 10, 14), sep = "")))
```



3.2 Plackett-Burman Designs

Simulation studies have shown Bayes screening to be robust to reasonable values of $\pi(\alpha)$. The method however is more sensitive to variation of γ values. Box and Meyer suggest the use of the γ value that minimize the posterior probability of the null model (no active factors). The rationale of this recommendation is because this value of γ also maximizes the likelihood function of γ since

$$p(\gamma|y) \propto \frac{1}{p(M_0|y, \gamma)}$$

where M_0 denotes the null model with no factors. See Box and Meyer (1993) and references therein.

3.2.1 Box et al. 1993: Example 1

This example considers a factorial design where 5 factors are allocated in a 12-run Plackett-Burman. The runs were extracted from the 2^5 factorial design in of the reactor experiment introduced by Box et al. (1978) and presented in section 4.1.2. Posterior probabilities are obtained and 3 factors are identified as potentially active, as shown in figure a) below. Then, the complete saturated design (11 orthogonal columns) is considered and marginal probabilities are calculated and displayed in figure b). None of the other contrasts x_6 - x_{11} seem to be active.

```

> par(mfrow = c(1, 2), mar = c(3, 3, 1, 1), mgp = c(1.5, 0.5, 0),
+     oma = c(0, 0, 0, 0), pty = "s", cex.axis = 0.7, cex.lab = 0.8,
+     cex.main = 0.9)
> data(BM93.e1.data, package = "BsMD")
> X <- as.matrix(BM93.e1.data[, 2:6])
> y <- BM93.e1.data[, 7]
> prob <- 0.25
> gamma <- 1.6
> reactor5.BsProb <- BsProb(X = X, y = y, blk = 0, mFac = 5, mInt = 3,
+   p = prob, g = gamma, ng = 1, nMod = 10)
> summary(reactor5.BsProb)

```

Calculations:

nRun	nFac	nBlk	mFac	mInt	p	g	totMod
12.00	5.00	0.00	5.00	3.00	0.25	1.60	32.00

Factor probabilities:

	Factor	Code	Prob
1	none	none	0.025
2	A	x1	0.011
3	B	x2	0.964
4	C	x3	0.009
5	D	x4	0.899
6	E	x5	0.577

Model probabilities:

	Prob	Sigma2	NumFac	Factors
M1	0.563	8.67	3	2,4,5
M2	0.324	39.51	2	2,4
M3	0.062	122.11	1	2
M4	0.025	240.45	0	none
M5	0.004	89.75	2	2,5
M6	0.003	211.33	1	5
M7	0.003	22.91	3	1,2,4
M8	0.002	226.88	1	4
M9	0.002	5.96	4	2,3,4,5
M10	0.002	5.99	4	1,2,4,5

```

> plot(reactor5.BsProb, main = "a) Main Effects")
> data(PB12Des, package = "BsMD")
> X <- as.matrix(PB12Des)
> reactor11.BsProb <- BsProb(X = X, y = y, blk = 0, mFac = 11,
+   mInt = 3, p = prob, g = gamma, ng = 1, nMod = 10)
> print(reactor11.BsProb, models = FALSE)

```

Design Matrix:

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
----	----	----	----	----	----	----	----	----	-----	-----

```

1  1 -1  1 -1 -1 -1  1  1  1 -1  1
2  1  1 -1  1 -1 -1 -1  1  1  1 -1
3 -1  1  1 -1  1 -1 -1 -1  1  1  1
4  1 -1  1  1 -1  1 -1 -1 -1  1  1
5  1  1 -1  1  1 -1  1 -1 -1 -1  1
6  1  1  1 -1  1  1 -1  1 -1 -1 -1
7 -1  1  1  1 -1  1  1 -1  1 -1 -1
8 -1 -1  1  1  1 -1  1  1 -1  1 -1
9 -1 -1 -1  1  1  1 -1  1  1 -1  1
10 1 -1 -1 -1  1  1  1 -1  1  1 -1
11 -1  1 -1 -1 -1  1  1  1 -1  1  1
12 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

```

Response vector:

```
56 93 67 60 77 65 95 49 44 63 63 61
```

Calculations:

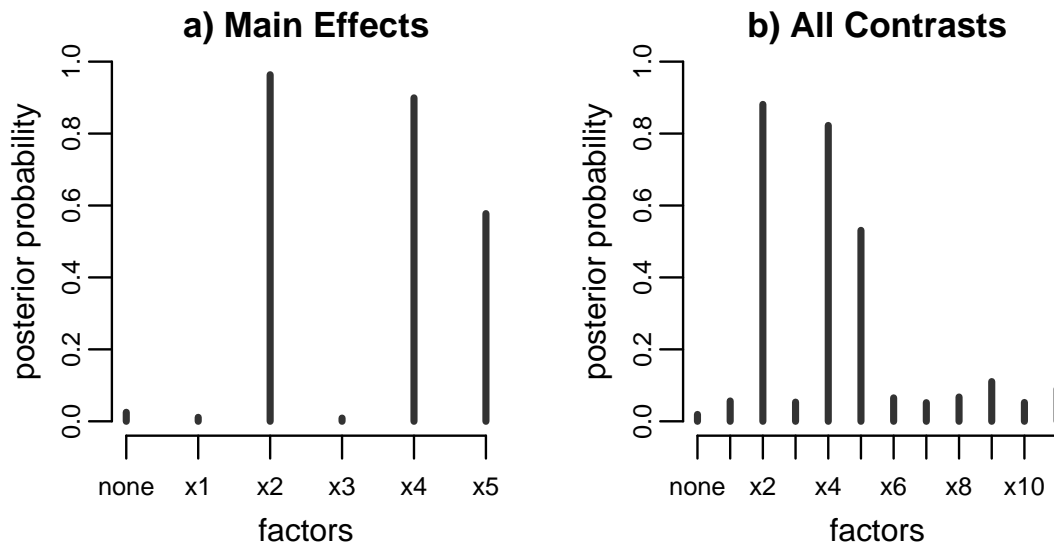
nRun	nFac	nBlk	mFac	mInt	p	g	totMod
12.00	11.00	0.00	11.00	3.00	0.25	1.60	2048.00

Output file: BsPrint.out

Factor probabilities:

	Factor	Code	Prob
1	none	none	0.019
2	x1	x1	0.056
3	x2	x2	0.881
4	x3	x3	0.053
5	x4	x4	0.823
6	x5	x5	0.531
7	x6	x6	0.065
8	x7	x7	0.052
9	x8	x8	0.067
10	x9	x9	0.110
11	x10	x10	0.052
12	x11	x11	0.090

```
> plot(reactor11.BsProb, main = "b) All Contrasts")
```



3.2.2 Box et al. 1993: Example 2

In this example again a 12-run Plackett-Burman design is analyzed. The effect of 8 factors (A, \dots, G), on the fatigue life of weld repaired castings is studied. As mentioned before, Box and Meyer suggest the use of values of γ that maximizes its likelihood (minimizes the probability of the null model). Figure a) below displays $P\{\gamma|y\}$ ($\equiv 1/P\{M_0|y\}$) as function of γ . It can be seen that the likelihood $P\{\gamma|y\}$ is maximum around $\gamma = 1.5$. In this example the maximization is carried out by calculating the marginal posterior probabilities for $1 \leq \gamma \leq 2$ and plotting the reciprocal of the probabilities of the null model. These probabilities are allocated in the first row of the probabilities matrix (`fatigueG.BsProb$prob`), where `fatigueG.BsProb` is the output of `BsProb`. A Bayes plot based on this $\gamma = 1.5$ is exhibited in figure b). Factors $F(X_6)$ and $G(X_7)$ clearly stick out from the rest. Alternatively, the unscaled γ likelihood ($P\{\gamma|y\}$) could be used since it has been already calculated by `BsProb` and assigned to `fatigueG.BsProb$pgam` element.

```
> par(mfrow = c(1, 2), mar = c(3, 3, 1, 1), mgp = c(1.5, 0.5, 0),
+     oma = c(0, 0, 1, 0), pty = "s", cex.axis = 0.7, cex.lab = 0.8,
+     cex.main = 0.9)
> data(BM93.e2.data, package = "BsMD")
> X <- as.matrix(BM93.e2.data[, 1:7])
> y <- BM93.e2.data[, 8]
> prob <- 0.25
> gamma <- c(1, 2)
> ng <- 20
> fatigueG.BsProb <- BsProb(X = X, y = y, blk = 0, mFac = 7, mInt = 2,
```

```

+   p = prob, g = gamma, ng = ng, nMod = 10)
> plot(fatigueG.BsProb$GAMMA, 1/fatigueG.BsProb$prob[1, ], type = "o",
+   xlab = expression(gamma), ylab = substitute("P{" * g * "|y}",
+   list(g = quote(gamma))))
> title(substitute("a) P{" * g * "|y}" %prop% "1/P{Null|y, " *
+   g * "}", list(g = quote(gamma))), line = +0.5, cex.main = 0.8)

```

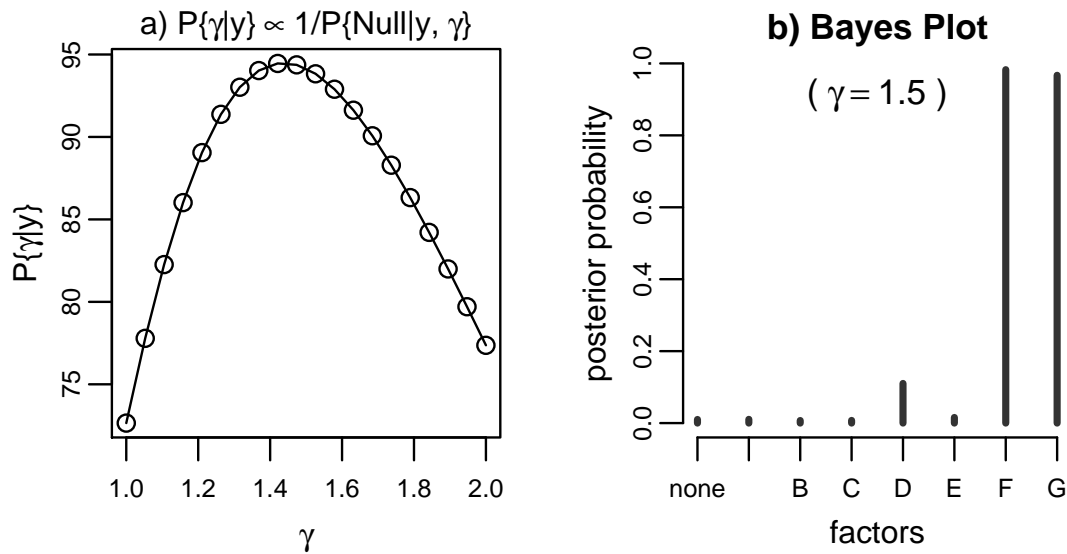
NULL

```

> gamma <- 1.5
> fatigue.BsProb <- BsProb(X = X, y = y, blk = 0, mFac = 7, mInt = 2,
+   p = prob, g = gamma, ng = 1, nMod = 10)
> plot(fatigue.BsProb, main = "b) Bayes Plot", code = FALSE)
> title(substitute("( " * g * " )", list(g = quote(gamma == 1.5))),
+   line = -1)

```

NULL



3.3 Extra Runs

3.3.1 Box et al. 1993: Example 3

This is the injection molding example from Box et al. (1978), where the analysis of the design is discussed in detail. In Box and Meyer (1993) the design is reanalyzed from the Bayesian approach. Firstly, a 16-run fractional factorial design is analyzed and the marginal posterior probabilities

are calculated and displayed in figure a) below. Factors A , C , E and H are identified as potential active factors. The 2^{8-4} factorial design collapses to a replicated 2^{4-1} design in these factors. Thus, estimates of the main effects and interactions are not all possible. Then, it is assumed that 4 extra runs are available and the full 20-run design is analyzed considering the blocking factor as another design factor. Their posterior probabilities are computed and exhibited in figure b). It is noted in the paper that the conclusions arrived there differ from those in Box et al. (1978), because the order of the interactions considered in the analysis, 3 and 2 respectively. In the `BsProb` function, the maximum interaction order to consider is declared with the argument `mInt`. For a detailed of the analysis see the source paper and reference therein.

```
> par(mfrow = c(1, 2), mar = c(4, 4, 1, 1), mgp = c(2, 0.5, 0),
+     oma = c(0, 0, 1, 0), pty = "s", cex.axis = 0.7, cex.lab = 0.8,
+     cex.main = 0.9)
> data(BM93.e3.data, package = "BsMD")
> print(BM93.e3.data)
```

	blk	A	B	C	D	E	F	G	H	y
1	-1	-1	-1	-1	1	1	1	-1	1	14.0
2	-1	1	-1	-1	-1	-1	1	1	1	16.8
3	-1	-1	1	-1	-1	1	-1	1	1	15.0
4	-1	1	1	-1	1	-1	-1	-1	1	15.4
5	-1	-1	-1	1	1	-1	-1	1	1	27.6
6	-1	1	-1	1	-1	1	-1	-1	1	24.0
7	-1	-1	1	1	-1	-1	1	-1	1	27.4
8	-1	1	1	1	1	1	1	1	1	22.6
9	-1	1	1	1	-1	-1	-1	1	-1	22.3
10	-1	-1	1	1	1	1	-1	-1	-1	17.1
11	-1	1	-1	1	1	-1	1	-1	-1	21.5
12	-1	-1	-1	1	-1	1	1	1	-1	17.5
13	-1	1	1	-1	-1	1	1	-1	-1	15.9
14	-1	-1	1	-1	1	-1	1	1	-1	21.9
15	-1	1	-1	-1	1	1	-1	1	-1	16.7
16	-1	-1	-1	-1	-1	-1	-1	-1	-1	20.3
17	1	-1	1	1	1	-1	-1	-1	1	29.4
18	1	-1	1	-1	-1	-1	1	1	1	19.7
19	1	1	1	-1	-1	1	-1	-1	1	13.6
20	1	1	1	1	1	1	1	1	1	24.7

```
> X <- as.matrix(BM93.e3.data[1:16, 2:9])
> y <- BM93.e3.data[1:16, 10]
> prob <- 0.25
> gamma <- 2
> plot(BsProb(X = X, y = y, blk = 0, mFac = 8, mInt = 3, p = prob,
+     g = gamma, ng = 1, nMod = 10), code = FALSE, main = "a) Fractional Factorial (FF)")
> X <- as.matrix(BM93.e3.data[, c(2:9, 1)])
> y <- BM93.e3.data[, 10]
> plot(BsProb(X = X, y = y, blk = 0, mFac = 9, mInt = 3, p = prob,
```

```
+      g = gamma, ng = 1, nMod = 5), code = FALSE, main = "b) FF with Extra Runs",
+      prt = TRUE, )
```

Calculations:

nRun	nFac	nBlk	mFac	mInt	p	g	totMod
20.00	9.00	0.00	9.00	3.00	0.25	2.00	512.00

Factor probabilities:

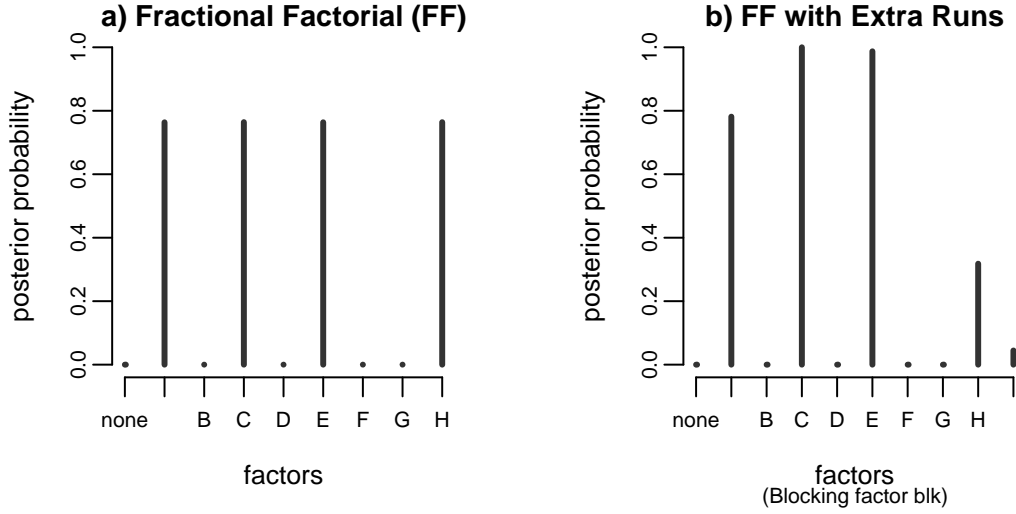
	Factor	Code	Prob
1	none	none	0.000
2	A	x1	0.781
3	B	x2	0.000
4	C	x3	1.000
5	D	x4	0.000
6	E	x5	0.987
7	F	x6	0.000
8	G	x7	0.000
9	H	x8	0.318
10	blk	x9	0.045

Model probabilities:

	Prob	Sigma2	NumFac	Factors
M1	0.672	1.012	3	1,3,5
M2	0.194	1.154	3	3,5,8
M3	0.086	0.593	4	1,3,5,8
M4	0.024	0.473	4	3,5,8,9
M5	0.010	0.519	4	1,3,5,9

```
> mtext(side = 1, "(Blocking factor blk)", cex = 0.7, line = 2.5)
```

NULL



4 Model Discrimination

Follow-up experiments for model discrimination (MD) are discussed by Meyer, Steinberg, and Box (1996). They introduce the design of follow-up experiments based on the MD criterion:

$$MD = \sum_{i \neq j} P(M_i|Y)P(M_j|Y)I(p_i, p_j)$$

where p_i denotes the predictive density of a new observation(s) conditional on the observed data Y and on model M_i being the correct model, and $I(p_i, p_j) = \int p_i \ln(p_i/p_j)$ is the Kullback-Leibler information, measuring the mean information for discriminating in favor of M_i against M_j when M_i is true. Under this criterion designs with larger MD are preferred.

The criterion combines the ideas for discrimination among models presented by Box and Hill (1967) and the Bayesian factor screening by Box and Meyer. The authors present examples for 4-run follow-up experiments but the criterion can be applied to any number of runs. In the next subsections we present the 4-run examples in the Meyer et al. (1996) paper and revisit the last of the examples from the one-run-at-a-time experimentation strategy.

The MD function is available for MD optimal follow-up designs. The function calls the `md` FORTRAN subroutine, a modification of the `MD.f` program included in the `mdopt` bundle. The output of the MD program is saved at the working directory in `MDPrint.out` file. The output of the MD function is a list of class MD with `print` and `summary` method functions.

For a given number of factors and a number of follow-up sets of runs, models are built and their MD calculated. The method employs the exchange search algorithm. See Meyer et al. (1996) and references therein. The MD function uses factor probabilities provided by `BsProb`. See the help pages for details.

4.1 4-run Follow-Up Experiments

4.1.1 Meyer et al. 1996: Example 1

The example presents the 5 best MD 4-run follow-up experiments for injection molding example, presented in section 3.3.1. In the code below note the call to the `BsProb` function before calling MD. The procedure selects the follow-up runs from a set of candidate runs `Xcand` (the original 2^{8-4} design), including the blocking factor `blk`. The best 4-run follow-up experiment, runs (9, 9, 12, 15), has a MD of 85.72, followed by (9, 12, 14, 15) with MD = 84.89. Note that these runs are different from the 4 extra runs in section 3.3.1.

```
> par(mfrow = c(1, 2), mar = c(3, 4, 1, 1), mgp = c(2, 0.5, 0),
+     oma = c(0, 0, 1, 0), pty = "s", cex.axis = 0.7, cex.lab = 0.8,
+     cex.main = 0.9)
> data(BM93.e3.data, package = "BsMD")
> X <- as.matrix(BM93.e3.data[1:16, c(1, 2, 4, 6, 9)])
> y <- BM93.e3.data[1:16, 10]
> injection16.BsProb <- BsProb(X = X, y = y, blk = 1, mFac = 4,
+   mInt = 3, p = 0.25, g = 2, ng = 1, nMod = 5)
> X <- as.matrix(BM93.e3.data[1:16, c(1, 2, 4, 6, 9)])
> p <- injection16.BsProb$ptop
> s2 <- injection16.BsProb$sigtop
> nf <- injection16.BsProb$nf top
> facs <- injection16.BsProb$jtop
> nFDes <- 4
> Xcand <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
+   1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1,
+   -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1,
+   1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1,
+   1, -1, 1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1, -1), nrow = 16,
+   dimnames = list(1:16, c("blk", "A", "C", "E", "H")))
> print(MD(X = X, y = y, nFac = 4, nBlk = 1, mInt = 3, g = 2, nMod = 5,
+   p = p, s2 = s2, nf = nf, facs = facs, nFDes = 4, Xcand = Xcand,
+   mIter = 20, nStart = 25, top = 5))
```

Base:

nRuns	nFac	nBlk	maxInt	gMain	gInter	nMod
16	4	1	3	2	2	5

Follow up:

nCand	nRuns	maxIter	nStart
16	4	20	25

Competing Models:

	Prob	Sigma2	NumFac	Factors
M1	0.236	0.582	3	1,2,3
M2	0.236	0.582	3	2,3,4
M3	0.236	0.582	3	1,3,4

```

M4 0.236 0.582 3      1,2,4
M5 0.057 0.441 4      1,2,3,4

```

Candidate runs:

```

      blk  A  C  E  H
1      1 -1 -1 -1 -1
2      1 -1 -1  1  1
3      1 -1  1 -1  1
4      1 -1  1  1 -1
5      1  1 -1 -1  1
6      1  1 -1  1 -1
7      1  1  1 -1 -1
8      1  1  1  1  1
9      1 -1 -1 -1  1
10     1 -1 -1  1 -1
11     1 -1  1 -1 -1
12     1 -1  1  1  1
13     1  1 -1 -1 -1
14     1  1 -1  1  1
15     1  1  1 -1  1
16     1  1  1  1 -1

```

Search trace output file: MDPrint.out

Top 5 runs:

```

      D r1 r2 r3 r4
1 85.726  9  9 12 15
2 84.893  9 12 14 15
3 83.684  9 11 12 15
4 77.136  9 11 12 14
5 77.111  9  9 11 12

```

4.1.2 Meyer et al. 1996: Example 2

This example is based on the 2^5 factorial reactor experiment presented initially in Box et al. (1978, chap. 12) and revisited from the MD criterion perspective in Box et al. (2005, chap. 7). The full design matrix and response is:

```

> data(Reactor.data, package = "BsMD")
> print(Reactor.data)

```

```

      A  B  C  D  E  y
1 -1 -1 -1 -1 -1 61
2  1 -1 -1 -1 -1 53
3 -1  1 -1 -1 -1 63
4  1  1 -1 -1 -1 61
5 -1 -1  1 -1 -1 53

```

```

6   1 -1  1 -1 -1 56
7  -1  1  1 -1 -1 54
8   1  1  1 -1 -1 61
9  -1 -1 -1  1 -1 69
10  1 -1 -1  1 -1 61
11 -1  1 -1  1 -1 94
12  1  1 -1  1 -1 93
13 -1 -1  1  1 -1 66
14  1 -1  1  1 -1 60
15 -1  1  1  1 -1 95
16  1  1  1  1 -1 98
17 -1 -1 -1 -1  1 56
18  1 -1 -1 -1  1 63
19 -1  1 -1 -1  1 70
20  1  1 -1 -1  1 65
21 -1 -1  1 -1  1 59
22  1 -1  1 -1  1 55
23 -1  1  1 -1  1 67
24  1  1  1 -1  1 65
25 -1 -1 -1  1  1 44
26  1 -1 -1  1  1 45
27 -1  1 -1  1  1 78
28  1  1 -1  1  1 77
29 -1 -1  1  1  1 49
30  1 -1  1  1  1 42
31 -1  1  1  1  1 81
32  1  1  1  1  1 82

```

First, it is assumed that only 8 runs (25, 2, ..., 32), from a 2^{5-2} were run. The runs are displayed in the output as `Fraction`. Bayesian screening is applied and posterior marginal probabilities are calculated and shown in figure a) below. These probabilities are used to find the MD optimal 4-run follow-up designs choosing the possible factor level combinations from the full 2^5 design. Since in this example responses for all the 32 runs are available, they are used as if the follow-up experiment was actually run and the posterior factor probabilities for the 12-run experiment determined and displayed in figure b). It is apparent how the extra runs clean up the activity of factors *B*, *D* and *E*. Note that the output of the `BsProb` function is used in the the call of `MD`. The complete output of both functions is sent to the files `BsPrint.out` and `MDPrint.out` respectively. Also, remember that method functions `print` and `summary` are available to control the amount of displayed output.

```

> par(mfrow = c(1, 2), mar = c(3, 4, 1, 1), mgp = c(2, 0.5, 0),
+     oma = c(0, 0, 0, 0), pty = "s", cex.axis = 0.7, cex.lab = 0.8,
+     cex.main = 0.9)
> fraction <- c(25, 2, 19, 12, 13, 22, 7, 32)
> cat("Fraction: ", fraction)

```

```
Fraction:  25 2 19 12 13 22 7 32
```

```
> X <- as.matrix(cbind(blk = rep(-1, 8), Reactor.data[fraction,
+   1:5]))
> y <- Reactor.data[fraction, 6]
> print(reactor8.BsProb <- BsProb(X = X, y = y, blk = 1, mFac = 5,
+   mInt = 3, p = 0.25, g = 0.4, ng = 1, nMod = 32), X = FALSE,
+   resp = FALSE, factors = TRUE, models = FALSE)
```

Calculations:

nRun	nFac	nBlk	mFac	mInt	p	g	totMod
8.00	5.00	1.00	5.00	3.00	0.25	0.40	32.00

Output file: BsPrint.out

Factor probabilities:

	Factor	Code	Prob
1	none	none	0.230
2	A	x1	0.271
3	B	x2	0.375
4	C	x3	0.172
5	D	x4	0.291
6	E	x5	0.170

```
> plot(reactor8.BsProb, code = FALSE, main = "a) Initial Design\n(8 runs)")
> p <- reactor8.BsProb$ptop
> s2 <- reactor8.BsProb$sigtop
> nf <- reactor8.BsProb$nftop
> facs <- reactor8.BsProb$jtop
> nFDes <- 4
> Xcand <- as.matrix(cbind(blk = rep(+1, 32), Reactor.data[, 1:5]))
> print(MD(X = X, y = y, nFac = 5, nBlk = 1, mInt = 3, g = 0.4,
+   nMod = 32, p = p, s2 = s2, nf = nf, facs = facs, nFDes = 4,
+   Xcand = Xcand, mIter = 20, nStart = 25, top = 5), Xcand = FALSE,
+   models = FALSE)
```

Base:

nRuns	nFac	nBlk	maxInt	gMain	gInter	nMod
8.0	5.0	1.0	3.0	0.4	0.4	32.0

Follow up:

nCand	nRuns	maxIter	nStart
32	4	20	25

Search trace output file: MDPrint.out

Top 5 runs:

	D	r1	r2	r3	r4
1	0.615	4	10	11	26

```

2 0.610  4 10 11 28
3 0.608  4 10 26 27
4 0.606  4 10 12 27
5 0.603  4 11 12 26

```

```

> new.runs <- c(4, 10, 11, 26)
> cat("Follow-up:", new.runs)

```

Follow-up: 4 10 11 26

```

> X <- rbind(X, Xcand[new.runs, ])
> y <- c(y, Reactor.data[new.runs, 6])
> print(reactor12.BsProb <- BsProb(X = X, y = y, blk = 1, mFac = 5,
+   mInt = 3, p = 0.25, g = 1.2, ng = 1, nMod = 5))

```

Design Matrix:

	blk	A	B	C	D	E
25	-1	-1	-1	-1	1	1
2	-1	1	-1	-1	-1	-1
19	-1	-1	1	-1	-1	1
12	-1	1	1	-1	1	-1
13	-1	-1	-1	1	1	-1
22	-1	1	-1	1	-1	1
7	-1	-1	1	1	-1	-1
32	-1	1	1	1	1	1
4	1	1	1	-1	-1	-1
10	1	1	-1	-1	1	-1
11	1	-1	1	-1	1	-1
26	1	1	-1	-1	1	1

Response vector:

44 53 70 93 66 55 54 82 61 61 94 45

Calculations:

nRun	nFac	nBlk	mFac	mInt	p	g	totMod
12.00	5.00	1.00	5.00	3.00	0.25	1.20	32.00

Output file: BsPrint.out

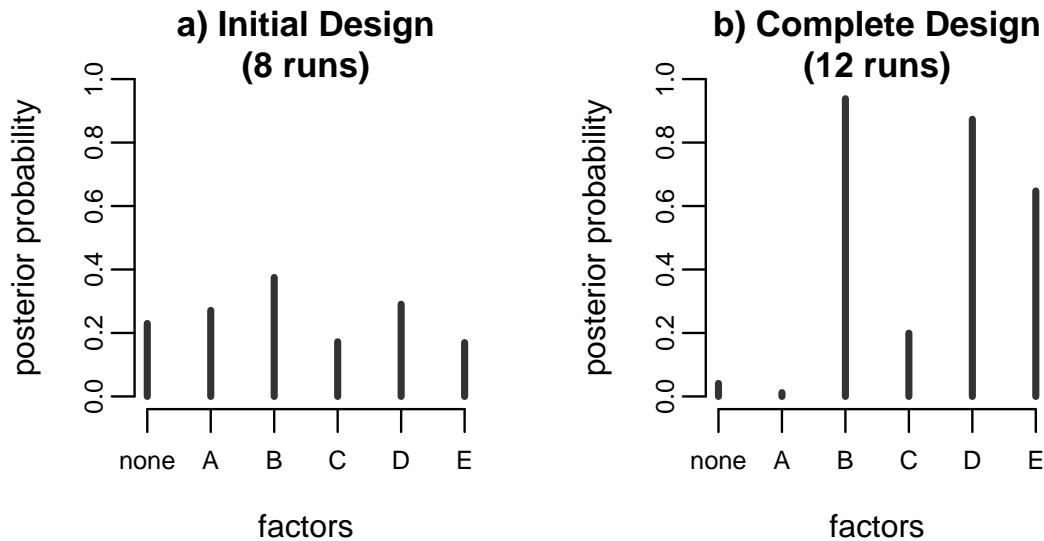
Factor probabilities:

Factor	Code	Prob
1	none	none 0.041
2	A	x1 0.012
3	B	x2 0.938
4	C	x3 0.199
5	D	x4 0.873
6	E	x5 0.647

Model probabilities:

	Prob	Sigma2	NumFac	Factors
M1	0.462	17.11	3	2,4,5
M2	0.209	66.63	2	2,4
M3	0.172	7.51	4	2,3,4,5
M4	0.064	167.76	1	2
M5	0.041	288.79	0	none

```
> plot(reactor12.BsProb, code = FALSE, main = "b) Complete Design\n(12 runs)")
```



4.2 One-run-at-a-time Experiments

4.2.1 Meyer et al. 1996: Example 2

Example 4.1.2 is considered again in this subsection. In this exercise we assume that the follow-up experimentation is in one-run-at-a-time fashion instead of the 4-run experiment discussed before. At each stage marginal posterior probabilities are computed and MD is determined, using $\gamma = 0.4, 0.7, 1.0, 1.3$. Once again, candidate runs are chosen from the 2^5 design. It can be seen there that at run 11, factors B , D and possibly E too, are cleared from the other factors. Note also that the final set of runs under the one-at-a-time approach (10, 4, 11, 15) ended being different from (4, 10, 11, 26) suggested by the 4-run follow-up strategy based on $\gamma = 0.4$. Bayes plots for each step are displayed in the figure below. See Box et al. (2005, chap. 7) for discussion of this approach. The code used in this section is included as appendix.

Design Matrix:

	blk	A	B	C	D	E
25	-1	-1	-1	-1	1	1
2	-1	1	-1	-1	-1	-1
19	-1	-1	1	-1	-1	1
12	-1	1	1	-1	1	-1
13	-1	-1	-1	1	1	-1
22	-1	1	-1	1	-1	1
7	-1	-1	1	1	-1	-1
32	-1	1	1	1	1	1
10	1	1	-1	-1	1	-1
4	1	1	1	-1	-1	-1
11	1	-1	1	-1	1	-1
15	1	-1	1	1	1	-1

Response vector:

44	53	70	93	66	55	54	82	61	61	94	95
----	----	----	----	----	----	----	----	----	----	----	----

Calculations:

nRun	nFac	nBlk	mFac	mInt	p	g	totMod
12.00	5.00	1.00	5.00	3.00	0.25	1.30	32.00

Output file: BsPrint.out

Factor probabilities:

Factor	Code	Prob	
1	none	none	0.035
2	A	x1	0.026
3	B	x2	0.944
4	C	x3	0.021
5	D	x4	0.917
6	E	x5	0.469

Model probabilities:

	Prob	Sigma2	NumFac	Factors
M1	0.441	15.24	3	2,4,5
M2	0.428	52.45	2	2,4
M3	0.036	173.18	1	2
M4	0.036	277.34	0	none
M5	0.016	8.95	4	1,2,4,5

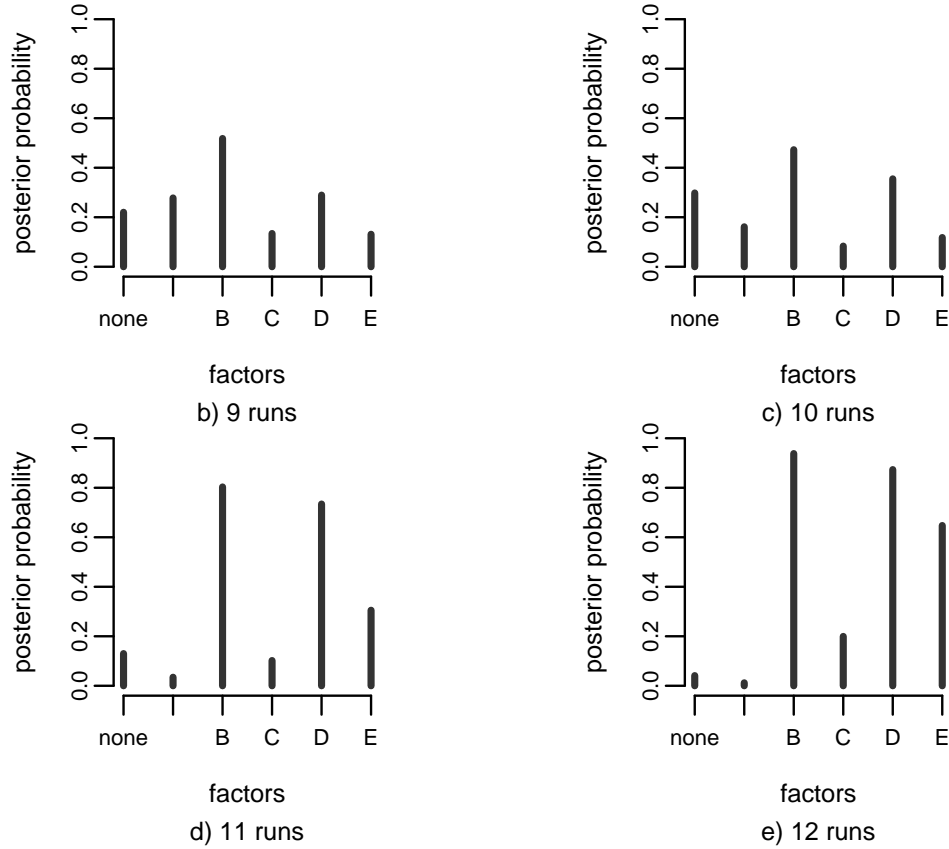
NULL

NULL

NULL

NULL

NULL

One-at-a-time Experiments

5 Summary

Various techniques are available for factor screening of unreplicated experiments. In this document we presented the functions of the BsMD package for Bayesian Screening and Model Discrimination. A number of examples were worked to show some of the features of such functions. We refer the reader to the original papers for detailed discussion of the examples and the theory behind the procedures.

Acknowledgment

Many thanks to Daniel Meyer for making his FORTRAN programs available and his permission to adapt and include them in this package. I want to thank as well Alejandro Muñoz whose detailed comments on early versions of the package and this document made BsMD easier to use and understand.

References

- G. E. P. Box and W.J. Hill. Discrimination among mechanistic models. *Technometrics*, 9(1):57–71, 1967.
- G. E. P. Box and R. D. Meyer. An analysis for unreplicated fractional factorials. *Technometrics*, 28(1):11–18, 1986.
- G. E. P. Box and R. D. Meyer. Finding the active factors in fractionated screening experiments. *Journal of Quality Technology*, 25(2):94–105, 1993.
- G. E. P. Box, W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters*. Wiley, New York, 1978.
- G. E. P. Box, J. S. Hunter, and W. G. Hunter. *Statistics for Experimenters II*. Wiley, New York, 2005.
- C. Daniel. *Applications of Statistics to Industrial Experimentation*. Wiley Sciences, New Jersey, 1976.
- R. V. Lenth. Quick and Easy Analysis of Unreplicated Factorials. *Technometrics*, 31(4):469–473, 1989.
- R. D. Meyer. mdopt: FORTRAN programs to generate MD-optimal screening and follow-up designs, and analysis of data. *Statlib*, August 1996. URL <http://lib.stat.cmu.edu>.
- R. D. Meyer, D. M. Steinberg, and G. E. P. Box. Follow-Up Designs to Resolve Confounding in Multifactor Experiments (with discussion). *Technometrics*, 38(4):303–332, 1996.

Appendix

Code used in section 4.2.1.

```
data(Reactor.data,package="BsMD")

#cat("Initial Design:\n")
X <- as.matrix(cbind(blk=rep(-1,8),Reactor.data[fraction,1:5]))
y <- Reactor.data[fraction,6]
lst <- reactor8.BsProb <- BsProb(X=X,y=y,blk=1,mFac=5,mInt=3,p=0.25,g=0.40,ng=1,nMod=32)

#cat("Follow-Up: run 1\n")
p <- lst$ptop; s2 <- lst$sigtop; nf <- lst$nftop; facs <- lst$jtop
reactor8.MD <- MD(X=X,y=y,nFac=5,nBlk=1,mInt=3,g=0.40,nMod=32,p=p,s2=s2,nf=nf,facs=facs,
  nFDes=1,Xcand=Xcand,mIter=20,nStart=25,top=3)
new.run <- 10
X <- rbind(X,Xcand[new.run,]); rownames(X)[nrow(X)] <- new.run
y <- c(y,Reactor.data[new.run,6])
lst <- reactor9.BsProb <- BsProb(X=X,y=y,blk=1,mFac=5,mInt=3,p=0.25,g=0.7,ng=1,nMod=32)
```

```

#cat("Follow-Up: run 2\n")
p <- lst$ptop; s2 <- lst$sigtop; nf <- lst$nfstop; facs <- lst$jtop
reactor9.MD <- MD(X=X,y=y,nFac=5,nBlk=1,mInt=3,g=0.7,nMod=32,p=p,s2=s2,nf=nf,facs=facs,
  nFDes=1,Xcand=Xcand,mIter=20,nStart=25,top=3)
new.run <- 4
X <- rbind(X,Xcand[new.run,]); rownames(X)[nrow(X)] <- new.run
y <- c(y,Reactor.data[new.run,6])
lst <- reactor10.BsProb <- BsProb(X=X,y=y,blk=1,mFac=5,mInt=3,p=0.25,g=1.0,ng=1,nMod=32)

#cat("Follow-Up: run 3\n")
p <- lst$ptop; s2 <- lst$sigtop; nf <- lst$nfstop; facs <- lst$jtop
reactor10.MD <- MD(X=X,y=y,nFac=5,nBlk=1,mInt=3,g=1.0,nMod=32,p=p,s2=s2,nf=nf,facs=facs,
  nFDes=1,Xcand=Xcand,mIter=20,nStart=25,top=3)
new.run <- 11
X <- rbind(X,Xcand[new.run,]); rownames(X)[nrow(X)] <- new.run
y <- c(y,Reactor.data[new.run,6])
lst <- reactor11.BsProb <- BsProb(X=X,y=y,blk=1,mFac=5,mInt=3,p=0.25,g=1.3,ng=1,nMod=32)

#cat("Follow-Up: run 4\n")
p <- lst$ptop; s2 <- lst$sigtop; nf <- lst$nfstop; facs <- lst$jtop
reactor10.MD <- MD(X=X,y=y,nFac=5,nBlk=1,mInt=3,g=1.3,nMod=32,p=p,s2=s2,nf=nf,facs=facs,
  nFDes=1,Xcand=Xcand,mIter=20,nStart=25,top=3)
new.run <- 15
X <- rbind(X,Xcand[new.run,]); rownames(X)[nrow(X)] <- new.run
y <- c(y,Reactor.data[new.run,6])
reactor12 <- BsProb(X=X,y=y,blk=1,mFac=5,mInt=3,p=0.25,g=1.30,ng=1,nMod=10)

print(reactor12,nMod=5,models=TRUE,plt=FALSE)

par(mfrow=c(2,2),mar=c(3,4,1,1),mgp=c(2,.5,0),oma=c(1,0,1,0),
  pty="s",cex.axis=0.7,cex.lab=0.8,cex.main=0.9)
plot(reactor9.BsProb,code=FALSE)
mtext(side=1,"b) 9 runs",line=3,cex=0.7)
plot(reactor10.BsProb,code=FALSE)
mtext(side=1,"c) 10 runs",line=3,cex=0.7)
plot(reactor11.BsProb,code=FALSE)
mtext(side=1,"d) 11 runs",line=3,cex=0.7)
plot(reactor12.BsProb,code=FALSE)
mtext(side=1,"e) 12 runs",line=3,cex=0.7)
title("One-at-a-time Experiments",outer=TRUE)

```