Examples of using the Q_B criterion

Pi-Wen Tsai*and Steven G. Gilmour†

Abstract

This document contains information about how to run `R` functions provided by Tsai and Gilmour for the calculation of the Q and Q_B criteria. The papers by Tsai, Gilmour and Mead (2000, 2007) and Tsai and Gilmour (2010) have used some of these functions for calculating the values of Q and Q_B criteria for designs with two- or three-level factors with the first-order or second-order maximal model respectively.

The program file **Qb_funs.r** contains a number of R functions that we have written for calculating the Q and Q_B criteria for designs under different situations.

Five main functions are introduced in this document.

1. **cal.Q3L(d)** – The Q criterion for three-level designs.
2. **cal.gwc(d)** – Generalized word counts for designs with three-levels.
3. **cal.Qb3L(d, p)** – The Q_B criterion for three-level designs with a given prior p.
4. **cal.Db3L(d, p, tau)** – The D_B criterion for three-level designs for the cases with (1) linear main effects $p = (1, \pi, \pi)$ and (2) linear and quadratic main effects as the primary terms $p = (1, 1, \pi)$ for different value of τ.
5. **plot.hist(p, noRun, noF)** – a histogram for helping experimenters to check if their priors provide models with sensible numbers of parameters.
6. **cal.Q2L(d)** – The Q criterion for two-level designs.
7. **cal.gwc(d,2)** – Generalized word counts for designs with two-levels.

1 The Q-criterion

cal.Q3L() – this function calculates the Q-criterion for a three-level design with the second-oder polynomial maximal model. We provide two different scalings: one is the orthogonal linear and quadratic contrasts and the other is unit-length orthogonal contrasts.

```
cal.Q3L=function(d, type=T)
d is the noRun*noF matrix for design with noRun  runs with noF factors,
  coded with (-1, 0, 1) for three-level designs,
             (-1,1) for two-level designs.
type=T (default) is for the unit-length orthogonal scale.
  For the normal orthogonal scale, type=F.
```

*http://math.ntnu.edu.tw/∼pwtsai

†http://www.maths.qmul.ac.uk/∼sgg/

(a) $Q – the value of the Q-criterion.

(b) $d – design d.

The Q-criterion for the second-order model is calculated as

$$Q = \frac{1}{\Delta} \sum_{i=1}^{v} \sum_{j=0}^{v} \frac{1}{a_{ii}} \frac{a_{ij}^2}{a_{ii} a_{jj}} w_{ij},$$

where a_{ij}, for $i, j = 0, \ldots, v$, is the (i, j)th entry of the information matrix, $\mathbf{A} = \mathbf{X}'\mathbf{X}$, for the maximal model of interest, and w_{ij} is the the number of models in which terms i and j are both included, and Δ is the total number of models.

Case for three-level designs Read in all the six three-level designs in 18 runs that were generated by Tsai et al (2000).

```
>rm(list=ls())
>source("Qb_funs.r")
>txt1="http://math.ntnu.edu.tw/~pwtsai/QB/tg10/Col_F6R18.out"
>dd=matrix(scan(txt1),  byrow=T, ncol=18)
```

Study designs: design indices 1, 2, 8, 22, and 220 and store them in a `list` DD.

```
>ind.k=c(1, 2, 8, 25,220)
>dsz=length(ind.k)
>DD=list()
>dd=t(dd)-2
>for (study.ind in ind.k){
       d=dd[, (6*(study.ind-1)+1) : (6*study.ind) ]
       colnames(d)=LETTERS[1:6]
       DD=append(DD, list(d))
   }
```

```
>d=DD[[2]]
>cal.Q3L(d)
```

```
$d
       A  B  C  D  E  F
 [1,] -1 -1 -1 -1  0 -1
 [2,] -1 -1  0  1 -1  0
 [3,] -1  0 -1  0 -1  1
 [4,] -1  0  1 -1  1  0
 [5,] -1  1  0  0  1 -1
 [6,] -1  1  1  1  0  1
 [7,]  0 -1  0  0  1  1
 [8,]  0 -1  1 -1 -1  1
 [9,]  0  0 -1  1  1  0
[10,]  0  0  1  0  0 -1
[11,]  0  1 -1  1 -1 -1
[12,]  0  1  0 -1  0  0
[13,]  1 -1 -1  0  0  0
```

```
[14,]  1 -1  1  1  1 -1
[15,]  1  0  0 -1 -1 -1
[16,]  1  0  0  1  0  1
[17,]  1  1 -1 -1  1  1
[18,]  1  1  1  0 -1  0

$Q
   Q
1.16
```

```
>tmp=c()
>for ( i in 1:length(DD)){
     d=DD[[i]]
     q=cal.Q3L(d)$Q
     tmp=c(tmp, q)
   }
>tmp
```

```
    Q     Q     Q     Q     Q
1.163 1.160 1.175 1.181 1.220
```

Use the scaling used in Tsai et al (2000)

```
>tmp=c()
>for ( i in 1:length(DD)){
     d=DD[[i]]
     q=cal.Q3L(d, type=F)$Q
     tmp=c(tmp,q )
   }
>tmp
```

```
    Q     Q     Q     Q     Q
2.266 2.269 2.296 2.324 2.408
```

Case for two-level designs A six two-level design obtained from Hadamard matrix of order 16.

```
>txt1="http://math.ntnu.edu.tw/~pwtsai/QB/tg10/pb16.out"
>dd2=read.table(txt1)  #code with 0 1
>dd2=replace(dd2, dd2==0, -1) #change 0 to -1
>out=rbind(
     c(1:6),c(1:5,8), c(1:4, 8, 12), c(1:4, 8, 13), c(1,2,3,7,8,11) )
>for (i in 1:nrow(out)){
     d2=dd2[,out[i,]]
     cat("Q :", cal.Q2L(d2), "\n")
     print(cal.gwc(d2, 2))
   }
```

```
Q : 13.19
$worList
```

```
    [,1] [,2] [,3] [,4] [,5] [,6]
       1    2    3    4    5    6
bij    0    0    4    3    0    0

Q : 6.053
$worList
    [,1] [,2] [,3] [,4] [,5] [,6]
       1    2    3    4    5    6
bij    0    0    2    1    0    0

Q : 4.97
$worList
    [,1] [,2] [,3] [,4] [,5] [,6]
       1    2    3    4    5    6
bij    0    0    2    0    0    1

Q : 3.568
$worList
    [,1] [,2] [,3] [,4] [,5] [,6]
       1    2    3    4    5    6
bij    0    0    1    1    1    0

Q : 6.053
$worList
    [,1] [,2] [,3] [,4] [,5] [,6]
       1    2    3    4    5    6
bij    0    0    2    1    0    0
```

2 Generalized word counts

cal.gwc()– this function calculates the generalized word count $b_k(i, j)$ for three-level designs for the second-order polynomial maximal model, with orthogonal scalings for the levels, or $b_k(k)$ for two-level designs.

```
cal.gwc=function(d, noLev)
d is the noRun*noF matrix for design with noRun  runs with noF factors,
  coded with (-1, 0, 1) for three-level designs
  (the default \verb|cal.gwc(d)|) and
             (-1,1) for two-level designs (\verb|cal.gwc(d,2)|).
noLev is the levels of factor in the design.
  default "noLev=3", noLev=2 for a two-level design.
```

output:

(a) $worList – generalized word count $b_k(i, j)$ or $b_k(k)$ with the orders ranked by the number of factors (k) in the words.

(b) $bList – generalized word count $b_k(i, j)$ with the orders ranked by the number of effects (i+2j) in the words.

(c) $word.A – the ANOVA-type of minimum aberration.

(d) $word.B – the β-aberration.

```
>print(cal.gwc(d), 3)
```

```
$worList
      Factors Lin Quad b_k(i,j)
 [1,]       1   0    1   0.0000
 [2,]       1   1    0   0.0000
 [3,]       2   0    2   0.0000
 [4,]       2   1    1   0.0000
 [5,]       2   2    0   0.0000
 [6,]       3   0    3   2.5000
 [7,]       3   1    2   2.2500
 [8,]       3   2    1   7.5000
 [9,]       3   3    0   0.7500
[10,]       4   0    4   0.8438
[11,]       4   1    3   3.0000
[12,]       4   2    2   6.1875
[13,]       4   3    1   2.2500
[14,]       4   4    0   1.2187
[15,]       5   0    5   0.0000
[16,]       5   1    4   1.1250
[17,]       5   2    3   4.5000
[18,]       5   3    2   1.5000
[19,]       5   4    1   1.5000
[20,]       5   5    0   0.3750
[21,]       6   0    6   0.0156
[22,]       6   1    5   0.7969
[23,]       6   2    4   0.6563
[24,]       6   3    3   0.8437
[25,]       6   4    2   1.0781
[26,]       6   5    1   0.6094
[27,]       6   6    0   0.0000

$bList
   Factors Lin Quad b_k(i,j)
1        1   1    0   0.0000
2        1   0    1   0.0000
2        2   2    0   0.0000
3        2   1    1   0.0000
3        3   3    0   0.7500
4        2   0    2   0.0000
4        3   2    1   7.5000
4        4   4    0   1.2187
5        3   1    2   2.2500
5        4   3    1   2.2500
5        5   5    0   0.3750
6        3   0    3   2.5000
6        4   2    2   6.1875
6        5   4    1   1.5000
6        6   6    0   0.0000
7        4   1    3   3.0000
7        5   3    2   1.5000
7        6   5    1   0.6094
8        4   0    4   0.8438
8        5   2    3   4.5000
8        6   4    2   1.0781
```

```
9          5   1    4   1.1250
9          6   3    3   0.8437
10         5   0    5   0.0000
10         6   2    4   0.6563
11         6   1    5   0.7969
12         6   0    6   0.0156

$word.A
     [,1]
[1,]  0.0
[2,]  0.0
[3,] 13.0
[4,] 13.5
[5,]  9.0
[6,]  4.0

$word.B
         [,1]
 [1,]  0.0000
 [2,]  0.0000
 [3,]  0.7500
 [4,]  8.7188
 [5,]  4.8750
 [6,] 10.1875
 [7,]  5.1094
 [8,]  6.4219
 [9,]  1.9688
[10,]  0.6563
[11,]  0.7969
[12,]  0.0156
```

Computer the GWP for two-level designs.

```
>cal.gwc(d2,2)
```

```
$worList
    [,1] [,2] [,3] [,4] [,5] [,6]
       1    2    3    4    5    6
bij    0    0    2    1    0    0
```

3 The Q_B-criterion

1. **cal.Qb3L()** – this function calculates the Q_B-criterion for a three-level design with the second-oder polynomial maximal model with different prior beliefs on each being in the true model.

   We need to specify the initial π_1, π_2 and π_3, which are respectively, the prior probability of each linear effect being in the best model, the probability of each quadratic effect being in the best model, given that the corresponding linear effect is, and the probability of each interaction being in the best model, given that the corresponding linear effects are.

```
cal.Qb3L=function(d, pp)
d is the noRun*noF matrix for design with noRun runs with noF factors,
  coded with (-1, 0, 1) for three-level designs.
pp is the 1*3 vector for the prior probability of each type of effects
  being in the true model.
```

cal.Qb3L(d, pp)

(a) \$d – the setting of design.
(b) \$Q – the value of the Q-criterion.
(c) \$Q.b – the value of the Q_B-criterion.
(d) \$xi.mat – the value of the ξ_{ij} the sum of prior probabilities of models contain i factors and j two-factor interactions being the best model as discussed in Tsai and Gilmour (2009).

The Q_B criterion which is given as

$$Q_B = \sum_{i=1}^{v}\sum_{j=0}^{v} \frac{1}{a_{ii}} \frac{a_{ij}^2}{a_{ii}a_{jj}} p_{ij}. \tag{1}$$

```
>pp=c(0.9, 0.5, 0.5)
>cal.Qb3L(d,pp)
```

```
$d
    A  B  C  D  E  F
1  -1 -1 -1 -1 -1 -1
2  -1 -1  1  0  0 -1
3  -1  0 -1  1  0  0
4  -1  0  0 -1  1  1
5  -1  1  0  0 -1  0
6  -1  1  1  1  1  1
7   0 -1  0  0  1  0
8   0 -1  1  1 -1  1
9   0  0 -1  0 -1  1
10  0  0  1 -1  0  0
11  0  1 -1 -1  1 -1
12  0  1  0  1  0 -1
13  1 -1 -1  1  1  0
14  1 -1  0 -1  0  1
15  1  0  0  1 -1 -1
16  1  0  1  0  1 -1
17  1  1 -1  0  0  1
18  1  1  1 -1 -1  0

$Q
 del
1.22

$Q.b
[1] 0.9838

$xi.mat
   [,1] [,2] [,3]   [,4]
```

```
xi    0    0    0 0.9130
xi    1    0    0 0.8131
xi    1    1    0 0.3872
xi    2    0    0 0.7232
xi    2    1    0 0.3423
xi    2    0    1 0.3423
xi    2    1    1 0.1593
xi    2    2    0 0.1593
xi    3    0    1 0.3019
xi    3    1    1 0.1391
xi    3    0    2 0.1391
xi    4    0    2 0.1210
```

```
>pp=c(1, 1, 0.2)
>cal.Qb3L(d,pp)$Q.b
```

```
[1] 1.031
```

Table 1: Calculate the Q_B values for the four six-factor designs with three levels in Tsai and Gilmour (2011).

```
>txt1="http://math.ntnu.edu.tw/~pwtsai/QB/tg10/Qb18T6.out"
>dd=read.table(txt1)
>dsz=nrow(dd)/6
>DD=list()
>dd=t(dd)-2
>for (i in 1:dsz){
       d=dd[, (6*(i-1)+1) : (6*i) ]
       colnames(d)=LETTERS[1:6]
       DD=append(DD, list(d))
   }
```

```
>d=DD[[1]]
>cal.Q3L(d)
```

```
$d
     A  B  C  D  E  F
V1  -1 -1 -1 -1  0 -1
V2  -1 -1  0  1 -1  0
V3  -1  0 -1  0 -1  1
V4  -1  0  1 -1  1  0
V5  -1  1  0  0  1 -1
V6  -1  1  1  1  0  1
V7   0 -1  0  0  1  1
V8   0 -1  1 -1 -1  1
V9   0  0 -1  1  1  0
V10  0  0  1  0  0 -1
V11  0  1 -1  1 -1 -1
V12  0  1  0 -1  0  0
```

```
V13  1 -1 -1  0  0  0
V14  1 -1  1  1  1 -1
V15  1  0  0 -1 -1 -1
V16  1  0  0  1  0  1
V17  1  1 -1 -1  1  1
V18  1  1  1  0 -1  0

$Q
   Q
1.16
```

```
>mat.Q=matrix(NA, nrow=length(DD), ncol=2)
```

```
>pp=c(0.9,0.5,0.5)
>for (i in 1:length(DD)){
      d=DD[[i]]
      q=cal.Q3L(d)$Q
      q.b=cal.Qb3L(d, pp)$Q.b
      mat.Q[i,]=c(q, q.b)
   }
>mat.Q
```

```
      [,1]   [,2]
[1,] 1.160 0.9341
[2,] 1.181 0.9505
[3,] 1.232 1.0411
[4,] 1.261 1.0637
```

Table 1: The summary of the Q and Q_B criteria for the four six-factor designs with three levels in 18 runs, p=c(0.9, 0.5, 0.5)

	Q	Q_B
D2	1.1620	0.9425
D25	1.1836	0.9615
Df2.1	1.2341	1.0472
Df2.2	1.2612	1.0647

Table 2 Specify the possible priors in a matrix "mat.pp" and calculate the Q_B-criteria for the four six-factor designs under different priors.

```
>mat.pp=rbind(
    c(0.9, 0.1, 0.5), c(0.9, 0.1, 0.8), c(0.9, 0.2, 0.5),
    c(0.9, 0.2, 0.8), c(0.9, 0.3, 0.2), c(0.9, 0.3, 0.6),
    c(0.9, 0.5, 0.2), c(0.9, 0.5, 0.5), c(0.8, 0.3, 0.2),
    c(0.8, 0.3, 0.3), c(0.8, 0.5, 0.5), c(0.8, 0.8, 0.5),
    c(0.7, 0.2, 0.5), c(0.7, 0.2, 0.8), c(0.7, 0.5, 0.2),
    c(0.7, 0.5, 0.6), c(1, 0.4, 0.4), c(1, 1, 0.2))
>mat.Q=matrix(NA, nrow=nrow(mat.pp), ncol=length(DD))
>for (ij in 1:nrow(mat.pp)) {
     pp=mat.pp[ij,]
```

```
     for (i in c(1:length(DD))){
        d=DD[[i]]
        q.b=cal.Qb3L(d, pp)$Q.b
        mat.Q[ij,i]=q.b
      }
   }
>mst.Q=cbind(mat.pp, mat.Q)
>mst.Q
```

```
      [,1] [,2] [,3]   [,4]   [,5]   [,6]   [,7]
 [1,]  0.9  0.1  0.5 0.8929 0.8980 0.5439 0.5662
 [2,]  0.9  0.1  0.8 0.9245 0.9418 0.6788 0.6887
 [3,]  0.9  0.2  0.5 0.9147 0.9233 0.6417 0.6643
 [4,]  0.9  0.2  0.8 0.9116 0.9307 0.7235 0.7345
 [5,]  0.9  0.3  0.2 0.6103 0.6099 0.6524 0.6740
 [6,]  0.9  0.3  0.6 0.9575 0.9733 0.7694 0.7876
 [7,]  0.9  0.5  0.2 0.6823 0.6855 1.0830 1.1046
 [8,]  0.9  0.5  0.5 0.9341 0.9505 1.0411 1.0637
 [9,]  0.8  0.3  0.2 0.5124 0.5119 0.5520 0.5678
[10,]  0.8  0.3  0.3 0.6027 0.6040 0.5867 0.6064
[11,]  0.8  0.5  0.5 0.7989 0.8109 0.9213 0.9411
[12,]  0.8  0.8  0.5 0.8261 0.8427 1.4229 1.4417
[13,]  0.7  0.2  0.5 0.5911 0.5940 0.4361 0.4521
[14,]  0.7  0.2  0.8 0.7225 0.7332 0.5100 0.5225
[15,]  0.7  0.5  0.2 0.4766 0.4778 0.7399 0.7510
[16,]  0.7  0.5  0.6 0.6995 0.7101 0.7783 0.7937
[17,]  1.0  0.4  0.4 1.0355 1.0495 1.0465 1.0767
[18,]  1.0  1.0  0.2 0.9713 0.9868 3.2925 3.3195
```

Table 2: Q_B for some six-level designs with 18 runs under various priors

	π_1	π_2	π_3	D2	D25	Df2.1	Df2.2
1	0.90	0.10	0.50	0.8998	0.9064	0.5500	0.5672
2	0.90	0.10	0.80	0.9383	0.9604	0.6944	0.6913
3	0.90	0.20	0.50	0.9220	0.9324	0.6478	0.6653
4	0.90	0.20	0.80	0.9260	0.9501	0.7389	0.7370
5	0.90	0.30	0.20	0.6127	0.6126	0.6534	0.6742
6	0.90	0.30	0.60	0.9674	0.9861	0.7782	0.7891
7	0.90	0.50	0.20	0.6850	0.6887	1.0839	1.1047
8	0.90	0.50	0.50	0.9425	0.9615	1.0472	1.0647
9	0.80	0.30	0.20	0.5160	0.5159	0.5533	0.5681
10	0.80	0.30	0.30	0.6086	0.6109	0.5897	0.6069
11	0.80	0.50	0.50	0.8117	0.8273	0.9298	0.9425
12	0.80	0.80	0.50	0.8405	0.8616	1.4311	1.4431
13	0.70	0.20	0.50	0.6030	0.6083	0.4446	0.4535
14	0.70	0.20	0.80	0.7456	0.7630	0.5318	0.5261
15	0.70	0.50	0.20	0.4812	0.4832	0.7413	0.7512
16	0.70	0.50	0.60	0.7173	0.7330	0.7905	0.7958
17	1.00	0.40	0.40	1.0355	1.0496	1.0465	1.0767
18	1.00	1.00	0.20	0.9713	0.9868	3.2925	3.3195

Table 3 some priors with linear main effect important or L+Q important.

```
>mat.pp=rbind(
     c(1, 0.3, 0.3),c(1, 0.6, 0.6),c(1, 0.1, 0.1),
     c(1, 1, 0.3),c(1, 1, 0.6),c(1, 1, 0.1))
>mat.Q=matrix(NA, nrow=nrow(mat.pp), ncol=length(DD))
>for (ij in 1:nrow(mat.pp)) {
     pp=mat.pp[ij,]
     for (i in c(1:length(DD))){
        d=DD[[i]]
        q.b=cal.Qb3L(d, pp)$Q.b
        mat.Q[ij, i]= q.b
      }
   }
>mst.Q=cbind(mat.pp, mat.Q)
>mst.Q
```

```
     [,1] [,2] [,3]   [,4]   [,5]   [,6]   [,7]
[1,]    1  0.3  0.3 0.8756 0.8798 0.8191 0.8516
[2,]    1  0.6  0.6 0.8975 0.9193 0.9855 1.0014
[3,]    1  0.1  0.1 0.4935 0.4894 0.3828 0.4003
[4,]    1  1.0  0.3 0.9908 1.0117 2.7415 2.7684
[5,]    1  1.0  0.6 0.7387 0.7582 0.8971 0.9091
[6,]    1  1.0  0.1 0.8332 0.8402 3.4015 3.4189
```

Table 3: Q_B for some six-level designs with 18 runs with various priors

	π_1	π_2	π_3	D2	D25	Df2.1	Df2.2
1	1.0000	0.3000	0.3000	0.8756	0.8798	0.8191	0.8516
2	1.0000	0.6000	0.6000	0.8976	0.9195	0.9856	1.0014
3	1.0000	0.1000	0.1000	0.4935	0.4894	0.3828	0.4003
4	1.0000	1.0000	0.3000	0.9908	1.0118	2.7415	2.7684
5	1.0000	1.0000	0.6000	0.7389	0.7584	0.8972	0.9091
6	1.0000	1.0000	0.1000	0.8332	0.8402	3.4015	3.4189

4 The D_B criterion

cal.Db3L() – this function calculates the D_B criterion. Here $D_B = \log(\mathbf{X}'\mathbf{X}+\mathbf{K}/\tau^2)$ as suggested by DuMouchel and Jones (1994).

```
cal.Db3L=function(d, pp, tau)
d is the noRun*noF matrix for design with noRun runs with noF factors.
pp is the 1*3 vector for the prior probability of each type of effects
   being in the true model. There
tau is the turning constant for the relative important between the
   primary and potential terms.
```

cal.Db3L(d, pp, tau)

$D.B – the value of the Q_B-criterion.

Here we consider the two cases when

1. linear effects are the primary terms and quadratic effects and two-factor linear by linear interactions are the potential terms, and
2. linear and quadratic effects are the primary terms and two-factor linear by linear interactions are the potential terms.

The values of the prior probabilities are either $(1, \pi, \pi)$ or $(1, 1, \pi)$ for the six three-level factor design in 18 runs.

Table 4. Three values of τ are discussed here: $\tau = 1$ is the default value, $\tau = 4$ is when the potential terms are more important, and $\tau = 0.25$ when the potential terms are not really of interest.

```
>pp=rbind(c(1,0.3,0.3), c(1,1,0.3))
>for (j in 1:2){
     p=pp[j,]
     mat.Q=c()

     for (i in 1:length(DD)){
       d=DD[[i]]
       t1=cal.Db3L(d,p,1)
       t2=cal.Db3L(d,p,4)
       t3=cal.Db3L(d,p,1/4)
       tmp=c(t1,t2,t3)
       mat.Q=cbind(mat.Q, tmp)
     }
     print(mat.Q)
   }
```

```
        tmp    tmp     tmp    tmp
[1,] 35.766 35.296 37.7309 38.231
[2,]  5.661  2.932  0.8687  6.073
[3,] 78.651 78.694 81.2535 80.816
        tmp     tmp  tmp  tmp
[1,] 33.240 32.2224 -Inf -Inf
[2,]  4.219  0.7789 -Inf -Inf
[3,] 68.615 68.5404 -Inf -Inf
```

Table 4: D_B for some six-level designs with 18 runs with (1) linear primary terms and (2) linear and quadratic primary terms under various τ

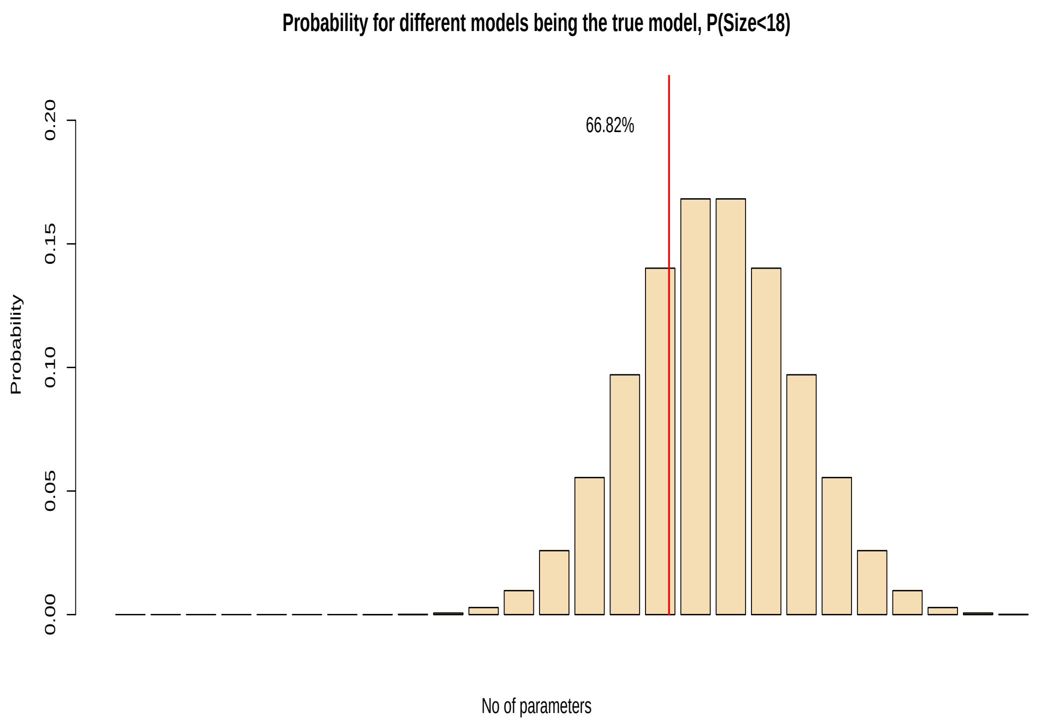
	τ	D2	D25	Df2.1	Df2.2
1	1.00	35.77	35.30	37.73	38.23
2	4.00	5.66	2.93	0.87	6.07
3	0.25	78.65	78.69	81.25	80.82
4	1.00	33.24	32.22	-Inf	-Inf
5	4.00	4.22	0.78	-Inf	-Inf
6	0.25	68.62	68.54	-Inf	-Inf

5 Histogram for the prior probabilities of models being the best model

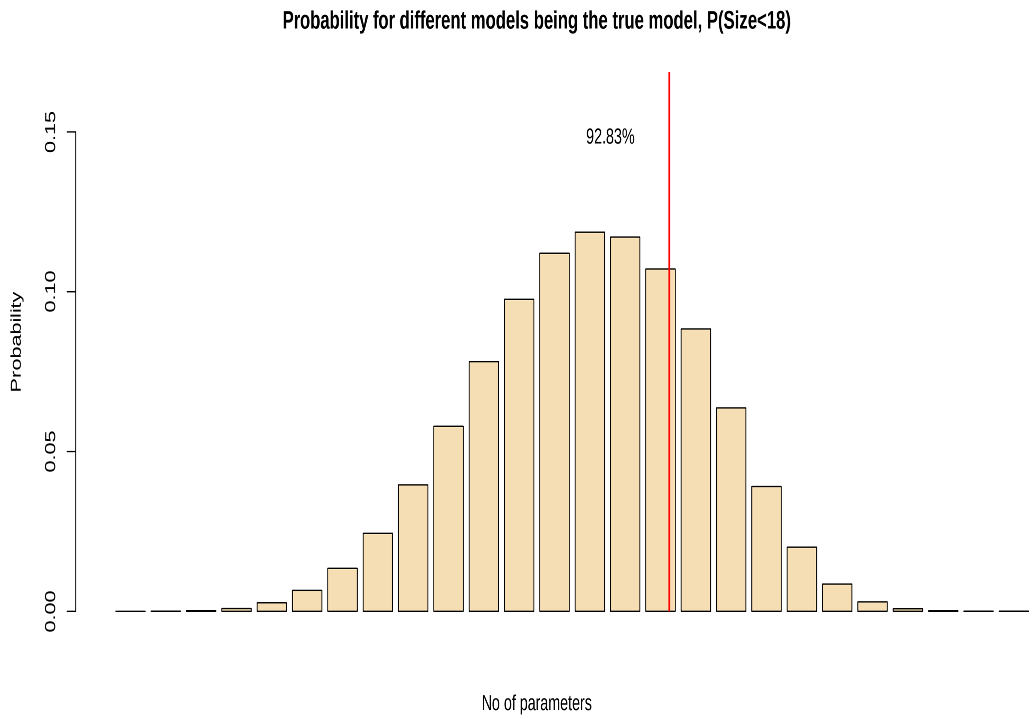
plot.hist() – this function provides a histogram for models being the best model. Tsai *et al.* (2007) suggested this as a tool for checking if a set of prior is sensible for the given experiment's run size.

```
plot.hist=function(pp, noF, noRun)
pp is the 1*3 vector for the prior probability of each type of effects
   being in the true model.
noF is the number of factors.
noRun is the number of run sizes.
```

```
>pp=c(1,0.5,0.5)
>plot.hist(pp,6,18)
```




```
>pp=c(0.9, 0.5, 0.4)
>plot.hist(pp, 6, 18)
```



References

DuMouchel, W. and Jones, B. (1994) A simple Bayesian modification of D-optimal designs to reduce dependence on an assumed model. *Technometrics*, **36**, 37–47.

Tsai, P.-W., Gilmour, S. G. and Mead, R. (2000). Projective three-level main effects designs robust to model uncertainty. *Biometrika*, **87**, 467–475.

Tsai, P.-W., Gilmour, S. G. and Mead, R. (2004). Some new three-level orthogonal main effects designs robust to model uncertainty. *Statistica Sinica*, **14**, 1075–1084.

Tsai, P. W., Gilmour, S. G. and Mead, R. (2007). Three-level main-effects designs exploiting prior information about model uncertainty. *Journal of Statistical Planning and Inference*, **137**, 619–627.