Appendix

Yunran Chen

2017/12/11

```
library(tibble)
library(dplyr)
library(purrr)
library(tidyr)
library(ggplot2)
library(stringr)
library(BAS)
library("R2jags")
library(knitr)

bioassay.read = read.table("http://stat.duke.edu/sites/stat.duke.edu/files/bioassay.txt",
                      header=T,stringsAsFactors = FALSE) %>% as.tibble()

bioassay=bind_cols(map_df(bioassay.read %>% select(uterus,weight,EE,ZM),~.x %>% as.numeric(.)),
                  map_df(bioassay.read %>% select(protocol,lab,group),~.x %>% as.factor(.))
                  ) %>% as.tibble()
bioassay.fac=bioassay %>% mutate(EE=as.factor(EE),ZM=as.factor(ZM))
X.fac=model.matrix(data=bioassay.fac,object =~ EE+ZM+lab+protocol+weight+EE:protocol+ZM:protocol+EE:lab+
```

Containing Part I to Part IV

I use three different ways to settle down the problems. For part I(MLE), I mainly use t-test and F-test. For part II(BMA with hyper-g-n prior), I mainly use the posterior model probability, inclusive probability and the posterior pdf for each predictor. For part III(jags), I would just use visualization (Credible Interval) to illustrate. Actually, for each part, we can use boxplot to illustrate for convenience and directly. The long table and graph would be attached at the end of the document.

Part IV

The three method can get the same result for

Uterotrophic bioassay successful overall at identifying effects. Some labs fail to detect such effects. The dose response vary across labs. All three methods agree that "Huntingd","Poulenc" stands out as being different from each other. The protocols differ in their sensitivity to detect the effects. Protocol B is recommeded.

The main difference:

1.The frequentists act more strict. For "outliers", they show less tolerence. For the changing dose point, they require more.(EE3 v.s. EE1) Generally, Baysian methods take uncertainty into consideration which behave more "moderate".

Method 3 (jags) showing more tolerence for "outliers" than Method 2 (bma) than Method 1 (MLE). This is because I adopted a prior with heavy tail in Method 3. Some "outliers" considered by Method1 may be "normal" in method 2,3("Bayer"). Some "outlier" considered by method 2 may be "normal" in method 1,3.("TNO")

2.It is easy for frequentist to get the estimation of parameters. But it is complex to construct suitable test. Although it is really time-consuming to get the posterior distribution of parameters. But it would be easier to analysis based on the data. And the solution seems more natural.

Improvements:

1.I use different methods to answer the questions in three parts. So it may be hard to compare.

2. For PartIII, I should adopt a prior for selecting the variable to decrease the computation.

Part I

summary

```
##1)We can consider `weight` and `uterus` as continuous variable. All other variables only have seperate
summary(bioassay)
```

```
##      uterus          weight             EE               ZM
##  Min.   : 10.4   Min.   : 27.7   Min.   : 0.000   Min.   :0.0000
##  1st Qu.: 32.8   1st Qu.: 48.4   1st Qu.: 0.010   1st Qu.:0.0000
##  Median : 80.0   Median : 56.1   Median : 0.300   Median :0.0000
##  Mean   :100.8   Mean   :106.7   Mean   : 1.875   Mean   :0.1009
##  3rd Qu.:124.4   3rd Qu.:200.2   3rd Qu.: 3.000   3rd Qu.:0.0000
##  Max.   :468.3   Max.   :341.0   Max.   :10.000   Max.   :1.0000
##  NA's   :4       NA's   :2
##  protocol       lab           group
##  A:1032   Hatano  : 264   2      : 246
##  B: 792   InEnvTox: 264   5      : 246
##  C: 594   Nihon   : 264   6      : 246
##  D: 263   Mitsubis: 263   7      : 246
##           Citijapa: 198   8      : 246
##           Sumitomo: 198   9      : 246
##           (Other) :1230   (Other):1205
```

```
bioassay %>% select(EE) %>% table()
```

```
## .
##    0 0.01 0.03  0.1  0.3    1    3   10
##  486  234  239  246  246  246  738  246
```

```
bioassay %>% select(ZM) %>% table()
```

```
## .
##    0  0.1    1
## 2189  246  246
```

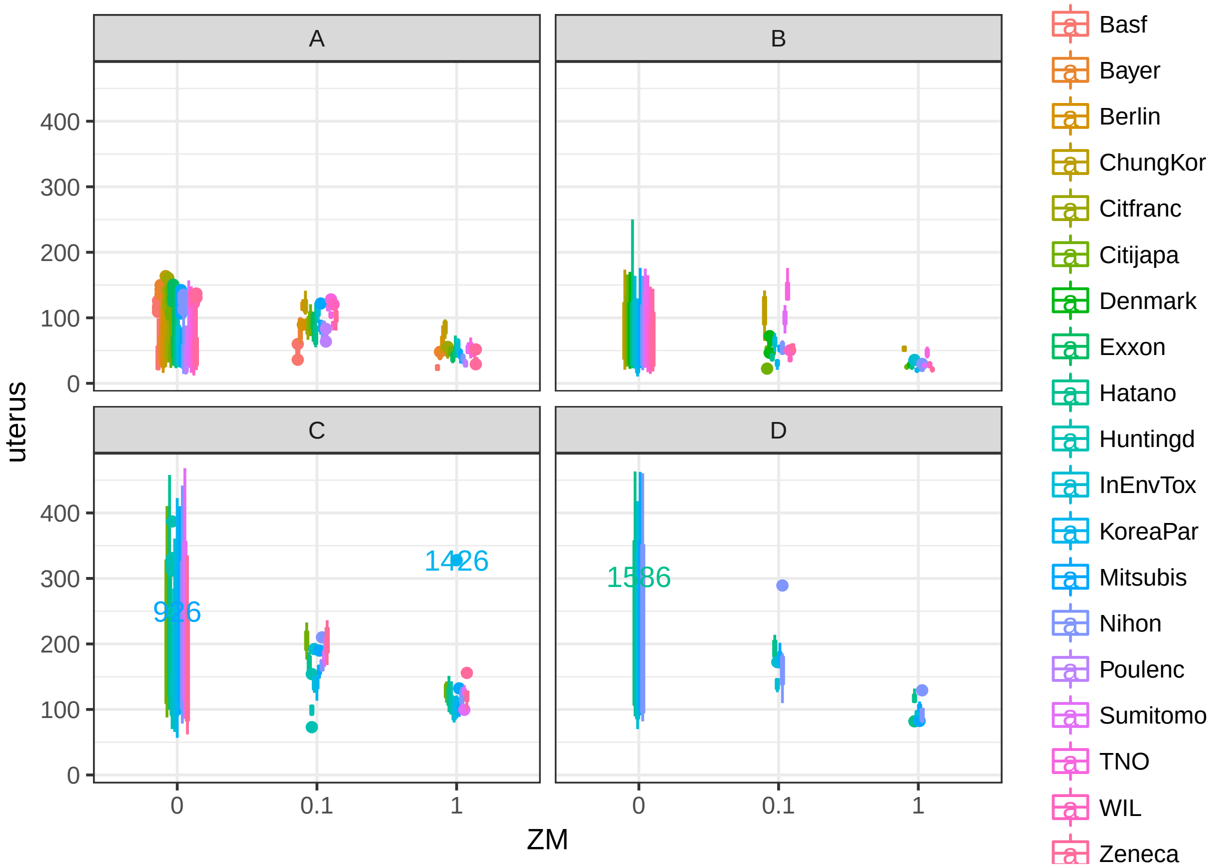
```
##2)There are significant interaction among variables. Because each labs adopted differet treatment.
table(bioassay$EE,bioassay.fac$lab)
```

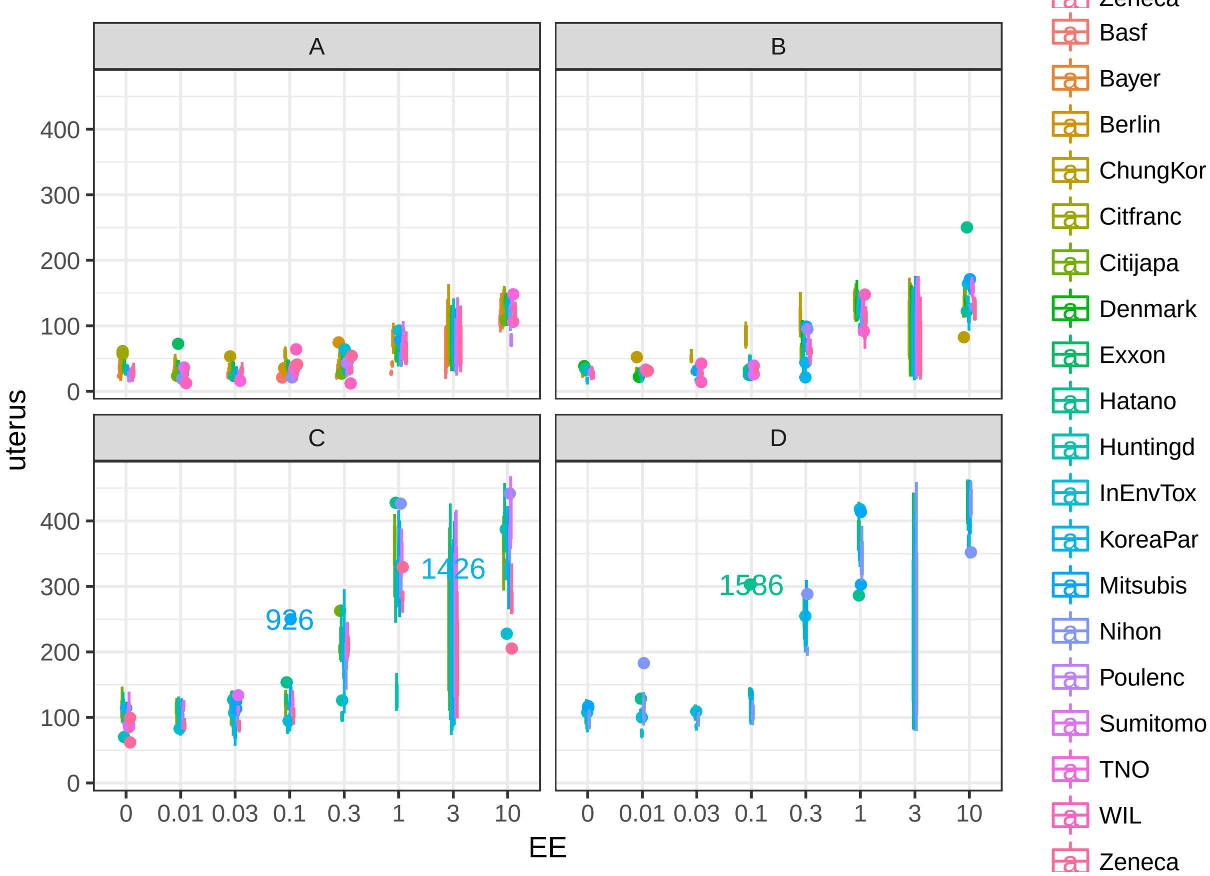
```
##
##        Basf Bayer Berlin ChungKor Citfranc Citijapa Denmark Exxon Hatano
##   0      12    12     12       24       12       36      12     6     48
##   0.01    0     0      6       12        6       18       6     6     24
##   0.03    6     0      6       12        6       18       6     6     24
##   0.1     6     6      6       12        6       18       6     6     24
##   0.3     6     6      6       12        6       18       6     6     24
##   1       6     6      6       12        6       18       6     6     24
##   3      18    18     18       36       18       54      18    18     72
##   10      6     6      6       12        6       18       6     6     24
##
##        Huntingd InEnvTox KoreaPar Mitsubis Nihon Poulenc Sumitomo TNO WIL
```

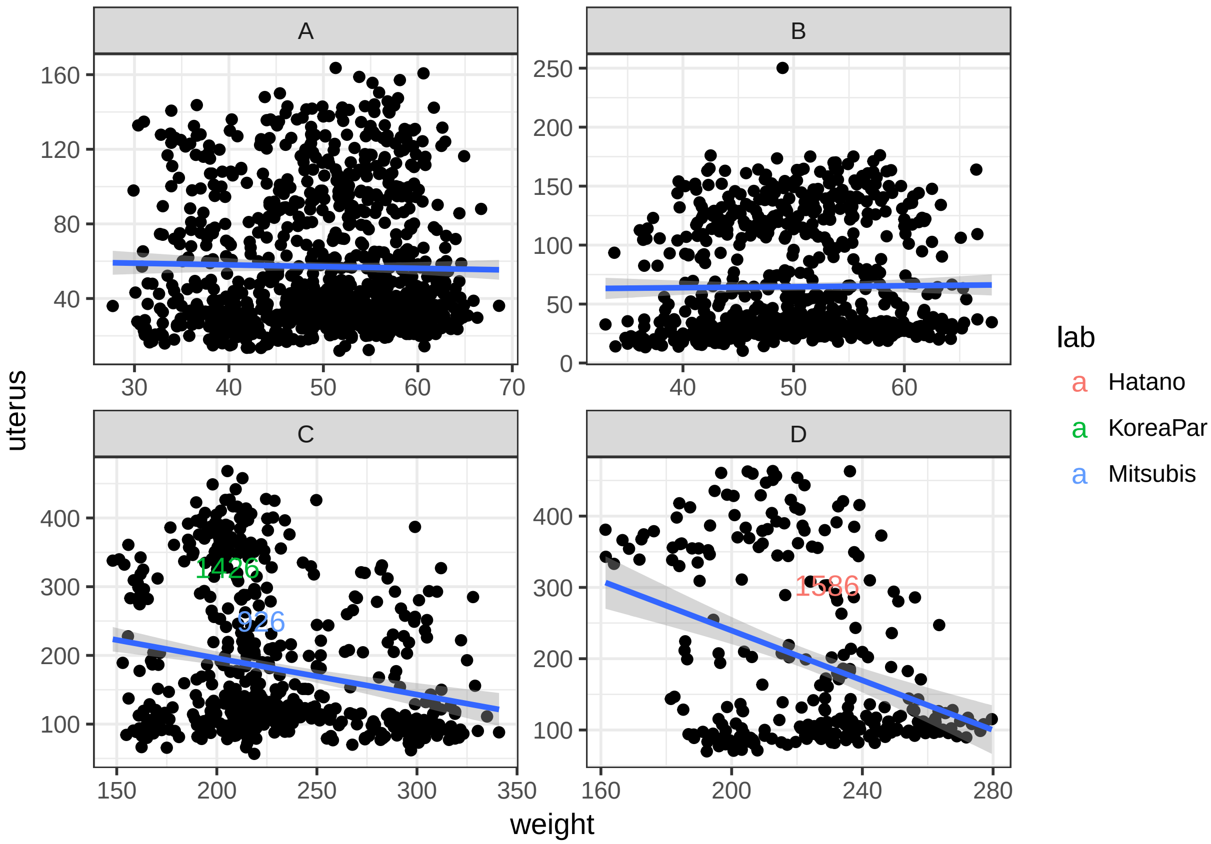
```
##   0          12        48        24        48    48       12        36  24  24
##   0.01        6        24        12        24    24        6        18  12  12
##   0.03        6        24        12        23    24        6        18  12  12
##   0.1         6        24        12        24    24        6        18  12  12
##   0.3         6        24        12        24    24        6        18  12  12
##   1           6        24        12        24    24        6        18  12  12
##   3          18        72        36        72    72       18        54  36  36
##   10          6        24        12        24    24        6        18  12  12
##
##        Zeneca
##   0        36
##   0.01     18
##   0.03     18
##   0.1      18
##   0.3      18
##   1        18
##   3        54
##   10       18
```

```
##visualization for points
#ggplot(data=bioassay,mapping = aes(y = lab,x = protocol,color=as.factor(ZM)))+
#  geom_jitter(alpha=0.5)
#ggplot(data=bioassay.fac,mapping = aes(y = lab,x = ZM,color=protocol))+
#  geom_jitter(alpha=0.5)
#ggplot(data=bioassay,mapping = aes(y = lab,x = protocol,color=as.factor(EE)))+
#  geom_jitter(alpha=0.5)
#ggplot(data=bioassay,mapping = aes(y = lab,x = protocol,color=as.factor(EE)))+
#     geom_count(alpha=0.5,position = "jitter")
```

EDA


A
B
C
D
uterus
ZM
0
0.1
1
400
300
200
100
926
1426
1536
Basf
Bayer
Berlin
ChungKor
Citfranc
Citijapa
Denmark
Exxon
Hatano
Huntingd
InEnvTox
KoreaPar
Mitsubis
Nihon
Poulenc
Sumitomo
TNO
WIL
Zeneca


A
B
C
D
uterus
EE
0
0.01
0.03
0.1
0.3
1
3
10
400
300
200
100
926
1426
1536
Basf
Bayer
Berlin
ChungKor
Citfranc
Citijapa
Denmark
Exxon
Hatano
Huntingd
InEnvTox
KoreaPar
Mitsubis
Nihon
Poulenc
Sumitomo
TNO
WIL
Zeneca



Model and Results

```
lm.full.fac=lm(data = bioassay.fac[-c(1586,926),],formula = uterus~EE+ZM+lab+protocol+weight+EE:protocol

#plot(lm.full.fac) ##use the plot to check model assumption. (detect outliers, normality, influential p

#bioassay.fac %>% select(EE,ZM) %>% table() -- there is imbalanced distribution

#summary(lm.full.fac)
#Adjusted R-squared:  0.9538

##if consider EE,ZM as numeric
#lm.full.q=lm(data = bioassay,formula = #uterus~poly(EE,2)+poly(ZM,2)+lab+protocol+weight+EE:protocol+Z
#anova(lm.full.q)
#summary(lm.full.q) #0.8286 #p=69
```

```
eff.tbl=function(lm.obj){
  if (is.matrix(lm.obj)){
    ind.mat=matrix(0,nrow=ncol(lm.obj),ncol=7)
    colnames(ind.mat)=c("EE","ZM","lab","protocol","EEdose","ZMdose","interaction")
    ind.mat=cbind(coef=lm.obj%>%colnames(.),ind.mat)
  }else{
    ind.mat=matrix(0,nrow=nrow(summary(lm.obj)$coefficients),ncol=7)
    colnames(ind.mat)=c("EE","ZM","lab","protocol","EEdose","ZMdose","interaction")
    ind.mat=cbind(coef=summary(lm.obj)$coefficients%>%rownames(.),ind.mat)
  }
```

```
  ind.mat[str_detect(ind.mat[,1],"EE"),"EE"]=1
  ind.mat[str_detect(ind.mat[,1],"ZM"),"ZM"]=1
  ind.mat[str_detect(ind.mat[,1],"EE.*protocol"),"interaction"]=1
  ind.mat[str_detect(ind.mat[,1],"ZM.*protocol"),"interaction"]=1
  ind.mat[str_detect(ind.mat[,1],"EE.*lab"),"interaction"]=1
  ind.mat[str_detect(ind.mat[,1],"ZM.*lab"),"interaction"]=1
  for (dose in c(bioassay.fac$EE %>% levels())){
     ind.mat[str_detect(ind.mat[,1],paste0("EE",dose)),"EEdose"]=dose
  }
  for (dose in c(bioassay.fac$ZM %>% levels())){
     ind.mat[str_detect(ind.mat[,1],paste0("ZM",dose)),"ZMdose"]=dose
  }
  for (lab in c(bioassay.fac$lab %>% levels())){
     ind.mat[str_detect(ind.mat[,1],paste0("lab",lab)),"lab"]=lab
  }
  for (protocol in c(bioassay.fac$protocol %>% levels())){
     ind.mat[str_detect(ind.mat[,1],paste0("protocol",protocol)),"protocol"]=protocol
  }
  ind.tbl=ind.mat%>% as.tibble()
  return(ind.tbl)
}

t.test=function(lm.obj,str.ee,str.lab,str.ori){
ind.tbl=eff.tbl(lm.obj)
cov.coef=vcov(lm.obj)
p=nrow(ind.tbl)
i.levels=bioassay.fac %>% pull(str.ori) %>% levels() #original colnames in bioassay.fac
i.n=i.levels %>% length(.)
lambda=matrix(nrow=i.n,ncol=nrow(ind.tbl))
lab.ee=t.value=p.value=denominator=nominator=numeric(i.n)

for (i in 1:i.n){
  lambda[i,]=((ind.tbl[str.lab]==i.levels[i])&(ind.tbl[str.ee]=="1"))|((ind.tbl[str.ee]=="1")&(ind.tbl[
  lab.ee[i]=summary(lm.obj)$coefficients[lambda[i,],"Estimate"] %>% sum()
  denominator[i]=(cov.coef[lambda[i,],lambda[i,]]) %>% sum() %>% sqrt()
  nominator[i]=lab.ee[i] %>% abs()
  t.value[i]=(nominator[i])/(denominator[i])
  p.value[i]=pt(q = t.value[i],df = summary(lm.obj)$df[2],lower.tail = FALSE)
}

res=list(t.test=tibble(i.levels,p.value,estimator=lab.ee),value=tibble(i.levels,variance=denominator,es
return(res)
}
```

a.1

uterotrophic bioassay successful overal at identifying effects of EE and ZM. F-test for EE,ZM are significant. For the significant coefficients, all EE are positive, ZM are negative.

```
#0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



anova(lm.full.fac)
```

```
## Analysis of Variance Table
```

```
##
## Response: uterus
##               Df  Sum Sq Mean Sq   F value    Pr(>F)    
## EE             7 6493181  927597 2437.9005 < 2.2e-16 ***
## ZM             2 2091034 1045517 2747.8158 < 2.2e-16 ***
## lab           18 2399373  133298  350.3335 < 2.2e-16 ***
## protocol       3 7374975 2458325 6460.9417 < 2.2e-16 ***
## weight         1  116191  116191  305.3720 < 2.2e-16 ***
## EE:protocol   21 2228787  106133  278.9368 < 2.2e-16 ***
## ZM:protocol    6  964465  160744  422.4660 < 2.2e-16 ***
## EE:lab       123  307285    2498    6.5659 < 2.2e-16 ***
## ZM:lab        36  105732    2937    7.7190 < 2.2e-16 ***
## Residuals   2457  934864     380                        
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coefs=rownames(summary(lm.full.fac)$coefficients)

res=summary(lm.full.fac)$coefficients %>%
  cbind(coef=coefs,.) %>%
  as.tibble() #get the coefficients matrix

colnames(res)=c("coef","estimate","std_error","t_value","p_value")

res %>% filter(as.numeric(p_value)<0.05) %>% slice(1:10)
```

```
## # A tibble: 10 x 5
##           coef           estimate           std_error           t_value
##          <chr>              <chr>               <chr>             <chr>
##  1 (Intercept)   17.5174725602682    6.0832664338643  2.87961619809253
##  2         EE3   69.3169543543242   9.89986651384673  7.00180696955581
##  3        EE10   85.6474733135621   9.90017366264282  8.65110817568213
##  4       ZM0.1  -44.2697077991087   11.2619188955346  -3.9309204949666
##  5         ZM1  -68.4662160221504    11.261889153971 -6.07946101103372
##  6 labChungKor   17.0236675479605   7.21540009027292  2.35935184951284
##  7 labCitfranc   19.9168751325116    8.1423844433654  2.44607403040768
##  8   protocolC   56.9794121460095   5.40377846249726  10.5443649367664
##  9   protocolD   47.3180475238927   6.15630309938495  7.68611401355793
## 10      weight  0.101351933549335  0.0269475706902319   3.7610786780893
## # ... with 1 more variables: p_value <chr>
```

a.2 do some labs fail to detect such effects

There are some labs fail to detect such effects, as follows:"Huntingd""Bayer""ChungKor""TNO""Zeneca". Or just pick out the labs with significant p value at "EE:lab_i" but opposite t value. ("Huntingd""Bayer""Zeneca")

```r
a.2.ee=t.test(lm.full.fac,"EE","lab","lab")$t.test
a.2.ee
```

```
## # A tibble: 19 x 3
##    i.levels      p.value estimator
##       <chr>        <dbl>     <dbl>
##  1     Basf 2.937598e-05  164.2023
##  2    Bayer 1.297653e-05  148.7924
##  3   Berlin 8.081249e-10  270.6481
##  4 ChungKor 9.973079e-14  244.7607
##  5 Citfranc 3.911202e-05  176.8314
```

```
## 6 Citijapa 2.504636e-12  197.9101
## 7  Denmark 1.725524e-06  227.4437
## 8    Exxon 2.001695e-04  211.2410
## 9   Hatano 2.850135e-17  221.4205
## 10 Huntingd 1.178475e-05 -211.9685
## 11 InEnvTox 7.907738e-15  202.7865
## 12 KoreaPar 2.228704e-06  167.7607
## 13 Mitsubis 4.957299e-15  204.5481
## 14    Nihon 1.228075e-14  201.2653
## 15  Poulenc 1.477856e-04  161.9877
## 16 Sumitomo 3.479523e-21  269.7772
## 17      TNO 1.605547e-11  224.8464
## 18      WIL 1.125449e-07  171.9591
## 19   Zeneca 1.077435e-08  160.2037
```

```r
a.2.ee %>% filter(((p.value<0.05)&(estimator<0))|(p.value>0.05))# %>% pull(i.levels)
```

```
## # A tibble: 1 x 3
##   i.levels      p.value estimator
##      <chr>        <dbl>     <dbl>
## 1 Huntingd 1.178475e-05 -211.9685
```

```r
a.2.zm=t.test(lm.full.fac,"ZM","lab","lab")$t.test
a.2.zm
```

```
## # A tibble: 19 x 3
##    i.levels      p.value  estimator
##       <chr>        <dbl>      <dbl>
##  1     Basf 4.220254e-09 -112.73592
##  2    Bayer 6.804663e-02  -29.08390
##  3   Berlin 4.049193e-06  -87.23322
##  4 ChungKor 6.389885e-02  -22.01560
##  5 Citfranc 1.763902e-04  -69.79584
##  6 Citijapa 2.304572e-09  -73.19550
##  7  Denmark 6.965919e-05  -81.37726
##  8    Exxon 6.126571e-07  -94.87320
##  9   Hatano 1.954502e-19 -103.41602
## 10 Huntingd 3.707009e-04   73.75550
## 11 InEnvTox 2.959297e-12  -79.17206
## 12 KoreaPar 1.941093e-04  -56.54709
## 13 Mitsubis 3.288518e-20 -105.47859
## 14    Nihon 7.278050e-20 -104.46423
## 15  Poulenc 4.017327e-02  -34.12421
## 16 Sumitomo 4.094885e-13  -89.55271
## 17      TNO 1.913899e-01  -12.61549
## 18      WIL 9.920550e-04  -44.74128
## 19   Zeneca 2.106639e-01   10.00848
```

```r
a.2.zm %>% filter(((p.value<0.05)&(estimator>0))|(p.value>0.05))#%>% pull(i.levels)
```

```
## # A tibble: 5 x 3
##   i.levels      p.value estimator
##      <chr>        <dbl>     <dbl>
## 1    Bayer 0.0680466340 -29.08390
## 2 ChungKor 0.0638988547 -22.01560
## 3 Huntingd 0.0003707009  73.75550
```

```
## 4      TNO 0.1913899445 -12.61549
## 5   Zeneca 0.2106639006  10.00848
```

```r
res %>% filter(str_detect(res$coef,"EE.*lab")) %>%
   filter(p_value<0.05,t_value<0)
```

```
## # A tibble: 3 x 5
##                coef          estimate        std_error           t_value
##               <chr>             <chr>            <chr>             <chr>
## 1      EE3:labBayer -28.2023694053915 13.8970614794358 -2.02937645826235
## 2 EE0.1:labHuntingd  -30.049402348541 14.7449822505579  -2.0379408966331
## 3     EE3:labZeneca -23.5618392029676 11.6930187049281 -2.01503476540555
## # ... with 1 more variables: p_value <chr>
```

```r
res %>% filter(str_detect(res$coef,"ZM.*lab")) %>%
   filter(p_value<0.05,t_value<0)
```

```
## # A tibble: 0 x 5
## # ... with 5 variables: coef <chr>, estimate <chr>, std_error <chr>,
## #   t_value <chr>, p_value <chr>
```

a.3 the change dose for EE? vary across labs?

From the output of summary(See the end of part I). The change dose for EE is EE3. Dose larger than this is significant, less than this is not significant. The value varies across labs. Because for different labs, the dose changing points is different. For example, SUmitomo. EE0.3 may be the changing dose point. For Huntingd, EE0.1 may be the changing dose point.

b. does the dose reponse vary across labs? are there certain labs stands out as being different? From the output of summary. There exist several significant interaction cofficients, meaning dose reponse vary across labs. The labs Berlin and Sumitomo stands out as being different.(with pvalue<0.001 for EE:labs). The labs Bayer,Poulenc,Zeneca stands out as being different.(with pvalue<0.005 for ZM:labs)

```r
res %>% filter(str_detect(res$coef,"EE.*lab")) %>%
  filter(p_value<0.001)
```

```
## # A tibble: 2 x 5
##              coef         estimate        std_error          t_value
##             <chr>            <chr>            <chr>            <chr>
## 1   EE1:labBerlin 49.4837223304949 13.8970357690877 3.56073936576932
## 2 EE1:labSumitomo 42.0319864090083 11.6927892205068 3.59469290144157
## # ... with 1 more variables: p_value <chr>
```

```r
res %>% filter(str_detect(res$coef,"ZM.*lab")) %>%
  filter(p_value<0.005)
```

```
## # A tibble: 3 x 5
##                coef         estimate        std_error          t_value
##               <chr>            <chr>            <chr>            <chr>
## 1   ZM0.1:labBayer 50.8727460593738 15.9269030374418 3.19413924601535
## 2 ZM0.1:labPoulenc 49.0113737477311 15.9267123836315 3.07730638735599
## 3    ZM1:labZeneca 44.7841287734452  13.358278781101 3.35253736707494
## # ... with 1 more variables: p_value <chr>
```

c.Do the protocols differ in sensitivity to detect? Which one recommend?

From the result from anova. The protocols differ. And the variance of protocol C,D is super large. Protocol A and B would be recommended.

```r
anova(lm.full.fac)["EE:protocol", ]
```

```
## Analysis of Variance Table
##
## Response: uterus
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## EE:protocol 21 2228787  106133  278.94 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm.full.fac)["ZM:protocol", ]
```

```
## Analysis of Variance Table
##
## Response: uterus
##             Df Sum Sq Mean Sq F value    Pr(>F)
## ZM:protocol  6 964465  160744  422.47 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
res %>% filter(str_detect(res$coef,"^protocol"))
```

```
## # A tibble: 3 x 5
##        coef          estimate        std_error            t_value
##       <chr>             <chr>            <chr>              <chr>
## 1 protocolB -2.16077131127482  2.4955877941489 -0.865836624277823
## 2 protocolC  56.9794121460095 5.40377846249726   10.5443649367664
## 3 protocolD  47.3180475238927 6.15630309938495   7.68611401355793
## # ... with 1 more variables: p_value <chr>
```

```r
res %>% filter(str_detect(res$coef,"EE.*protocol"))
```

```
## # A tibble: 21 x 5
##                coef           estimate        std_error             t_value
##               <chr>              <chr>            <chr>               <chr>
##  1 EE0.01:protocolB   1.19108136858824 4.31635717989136   0.275945969934356
##  2 EE0.03:protocolB     3.541307048283 4.31635426940738   0.820439386401249
##  3  EE0.1:protocolB   7.45488173303661 4.31641616363892    1.72709985562463
##  4  EE0.3:protocolB   35.0983219034887 4.31635334591939    8.13147559772188
##  5    EE1:protocolB   58.2145220892542 4.31639052433566    13.4868524432724
##  6    EE3:protocolB   34.4029534792872 4.31637929244136    7.97032678280428
##  7   EE10:protocolB   17.5749522283604 4.32796829285087    4.06078580968154
##  8 EE0.01:protocolC -0.323072655511128  4.9003836743018 -0.065928032779425
##  9 EE0.03:protocolC   2.72255040119417 4.90038109624472   0.555579320816605
## 10  EE0.1:protocolC   14.0429742703248 4.92125955112887    2.85353254068942
## # ... with 11 more rows, and 1 more variables: p_value <chr>
```

```r
res %>% filter(str_detect(res$coef,"ZM.*protocol"))
```

```
## # A tibble: 6 x 5
##               coef          estimate        std_error           t_value
##              <chr>             <chr>            <chr>             <chr>
## 1 ZM0.1:protocolB  -61.037963794127 4.98061458836321  -12.255106816885
## 2   ZM1:protocolB -53.8667551088683 4.98057530455602 -10.8153680679405
## 3 ZM0.1:protocolC -158.414580874021 5.67997807489639 -27.8899986558329
## 4   ZM1:protocolC -171.324039321919 5.66139549748667  -30.261803719238
## 5 ZM0.1:protocolD -183.644764786495 7.20742931513241 -25.4799258871567
```

```
## 6   ZM1:protocolD -217.250601221133 7.20370581533531 -30.1581723060718
## # ... with 1 more variables: p_value <chr>
```

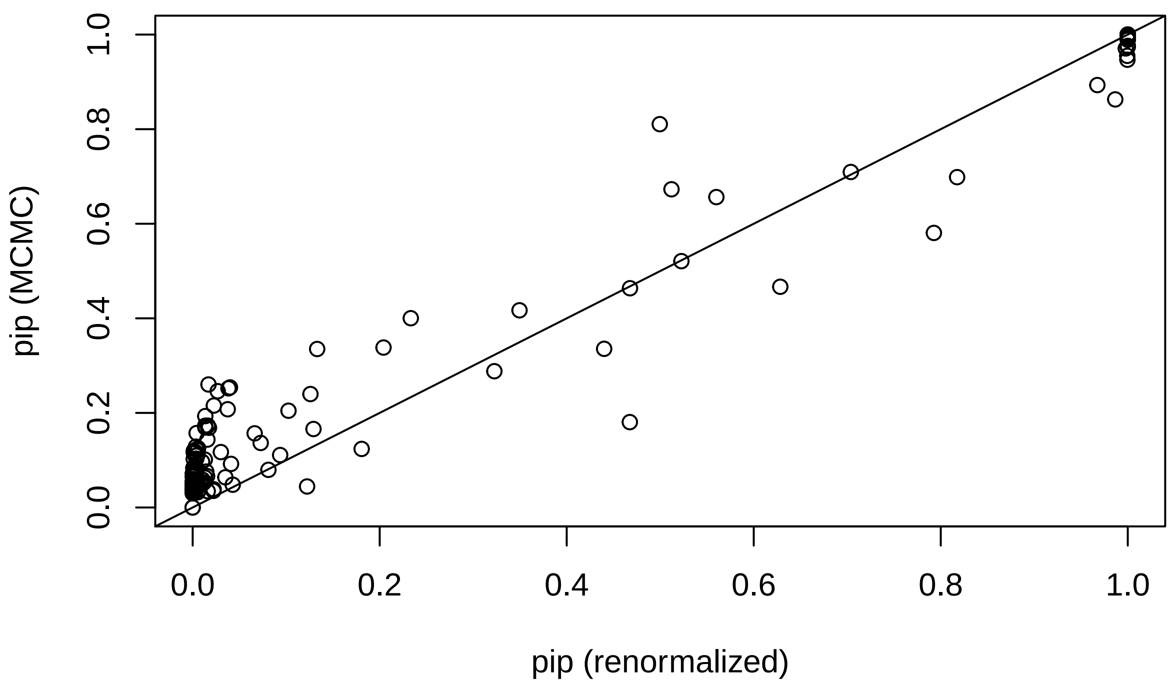
Part II

With the same model, using Bayesian Average Model to estimate the parameter with hyper-g-n prior(mixture prior). To answer the questions, I mainly concentrate on inclusion probability, posterior model probability and the shape of posterior distribution for β_i (or confidence interval).

```
n=nrow(bioassay.fac)

bas.fac.2=bas.lm(formula =uterus~EE+ZM+lab+protocol+weight+EE:protocol+ZM:protocol+EE:lab+ZM:lab,
                 data=bioassay.fac,
                 prior="hyper-g-n",
                 alpha=n,
                 #n.models=20000,
                 method = "MCMC",
                 thin=10,
                 #initprobs = "eplogp",
                 MCMC.iterations = 500000)
```

```
## Warning in bas.lm(formula = uterus ~ EE + ZM + lab + protocol + weight + :
## dropping 4 rows due to missing data
```

```
##diagonose
diagnostics(bas.fac.2,"pip")
```




```
##More iteration would be better
```

a.1 Similar to Part I, we want to test whether the coefficients before `EE` all equal to 0. Here I use sum posterior model probabiliters over all models that include `EE`, is 0.99982. Similar for ZM,is 0.99982. For the coefficients `EE` with high inclusion probability, all of them are larger than 0 with high probability. Simlar for `ZM`(negative). So the method successful overall at identifying effects.

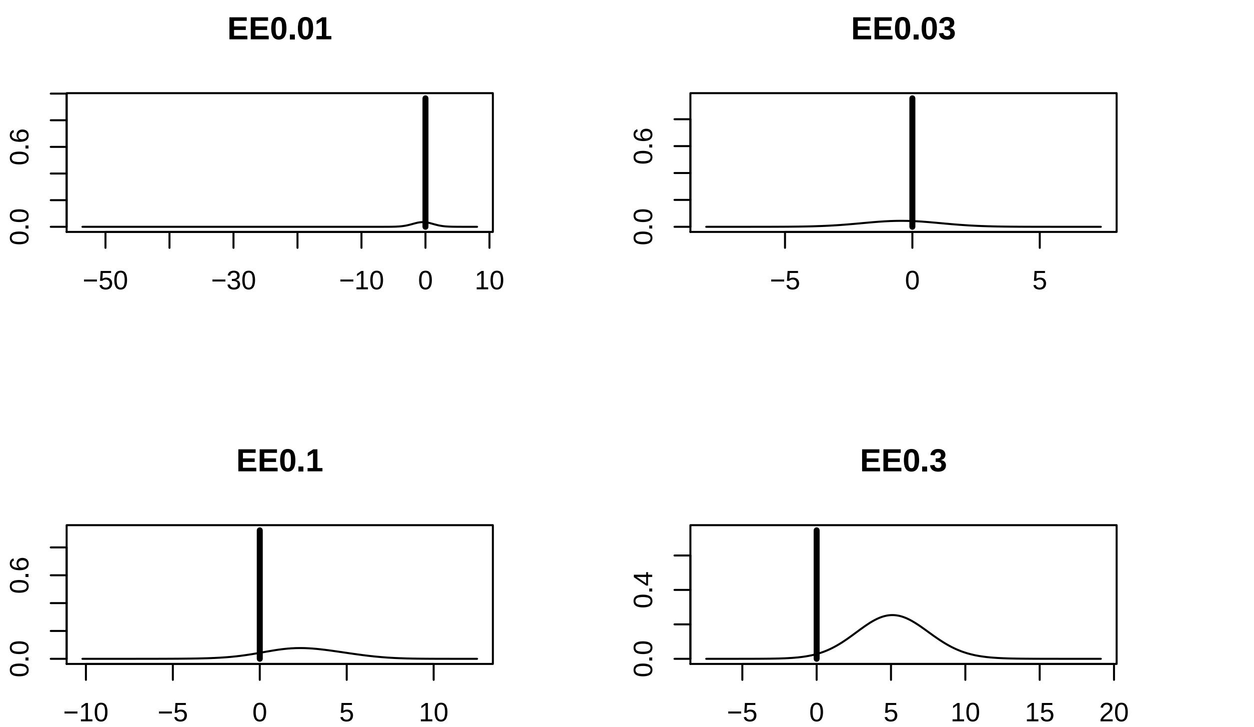
```r
a_1=function(str_ee){
  which.mat=list2matrix.which(bas.fac.2,)
  ind.tbl.bas=eff.tbl(X.fac) #part II model assumption is the same to part I
  head(ind.tbl.bas)
  ind.var=((ind.tbl.bas[str_ee]=="1")&(ind.tbl.bas$interaction=="0"))
  n.ind=sum(ind.var)
  poll.in=(which.mat[,ind.var] %*% rep(1,n.ind))>0
  res=list(prob=sum(poll.in*bas.fac.2$postprobs),ind=ind.var)
  return(res)
}



a_1("EE")$prob
```

```
## [1] 0.99992
```

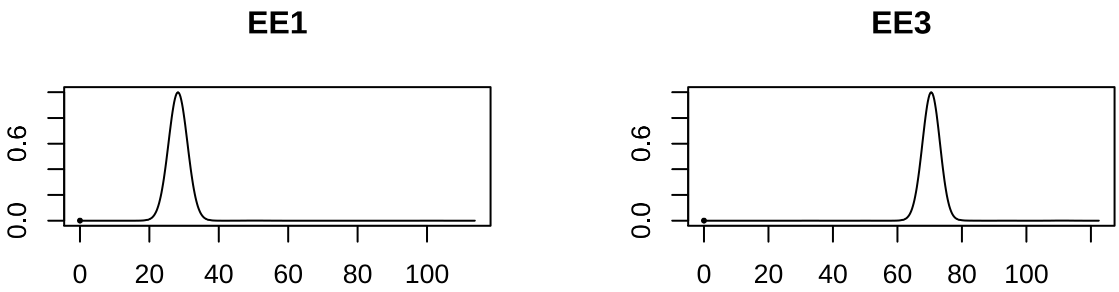
```r
a_1("ZM")$prob
```

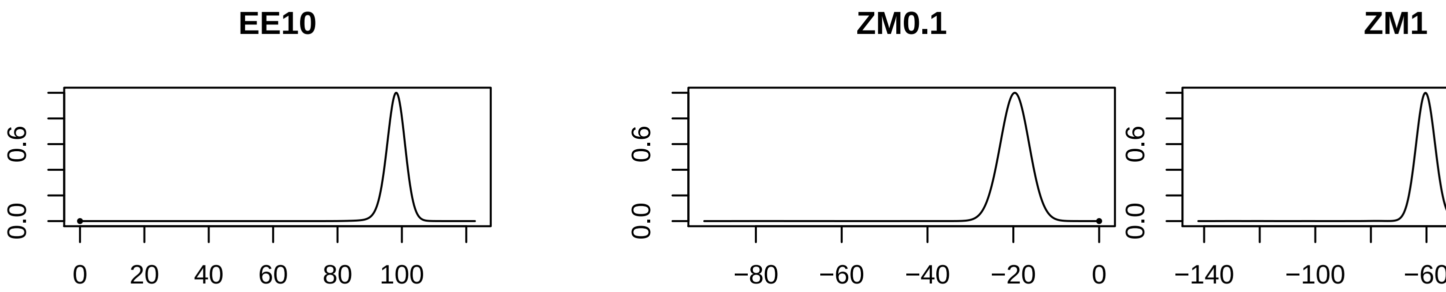
```
## [1] 0.99978
```

```r
#image(bas.fac.2)--to much predictor, cannot visualization
#plot(bas.fac.2)

par(mfrow=c(2,2))
ind.ee=which(a_1("EE")$ind)
coef(bas.fac.2) %>% plot(.,ask=F,subset=ind.ee)
```




```r
ind.zm=which(a_1("ZM")$ind)
coef(bas.fac.2) %>% plot(.,ask=F,subset=ind.zm)
```





a.2 Concerntrate on `EE:lab_i`, `ZM:lab_i`. Pick out the labs with high inclusive probability but opposite location compared to others. ("Huntingd""Zeneca""TNO""KoreaPar""ChungKor""Poulenc""EnvTox") Simply, the rule is for `EE`, if `EE:lab_i` is at the left side of the verticle black line and the verticle black line is short, we may consider the lab fail to detect. Similar for `ZM`.

a.3 Concerntrate on `EE`,`ZM`, see which dose level the inclusive probability significantly change. EE1. And concerntrate on `EE:lab_i`,`ZM:lab_i` to see which dose level the inclusive probability significantly change. They vary.

b. The dose response vary across labs, because there exist `EE:lab_i`,`ZM:lab_i` with high inclusive probability. There are certain labs ("Huntingd""Zeneca""TNO""KoreaPar""ChungKor""Poulenc""EnvTox") stand out as being different.

c.Protocols differ in the sensitivity to detect effects. Protocol B would be recommended because the length of CI for `protocolB:EE` and `protocolB:ZM` are small compared to others.

```
coefs=rownames(confint(coef(bas.fac.2)))
ci.length=apply(confint(coef(bas.fac.2)),1,function(x) as.numeric(x[2])-as.numeric(x[1]))

bas.coef=confint(coef(bas.fac.2)) %>%
  cbind(coef=coefs,.) %>%
  as.tibble() %>% #get the coefficients matrix
  mutate(ci.length=ci.length)

bas.coef %>% filter(str_detect(coef,"EE.*protocol")) %>% arrange(sort(ci.length)) %>% slice(1:10)
```

```
## # A tibble: 10 x 5
##                coef            `2.5%`           `97.5%`                 beta
##               <chr>             <chr>             <chr>                <chr>
##  1 EE0.01:protocolB                 0                 0  -0.0580646028824629
##  2 EE0.03:protocolB                 0                 0 -0.00783646148478639
##  3  EE0.1:protocolB                 0 5.84581272698972    0.584951728986413
##  4  EE0.3:protocolB 26.9241930774477 40.9172070895798     34.2580916086904
```

```
##   5    EE1:protocolB 54.6311461919624 68.1905290200169     61.4867093062725
##   6    EE3:protocolB 30.0642593477745 43.3277363201775     36.8148784557911
##   7   EE10:protocolB 8.52790074748414 22.0889966047053     15.3405036938238
##   8 EE0.01:protocolC                0                0  -0.0046476724141366
##   9 EE0.03:protocolC                0                0 -0.00543334813404808
##  10  EE0.1:protocolC 5.72336409178617 22.0076864017156     13.7904554310719
## # ... with 1 more variables: ci.length <dbl>
```

```
bas.coef %>% filter(str_detect(coef,"ZM.*protocol")) %>% arrange(sort(ci.length)) %>% slice(1:10)
```

```
## # A tibble: 6 x 5
##               coef             `2.5%`            `97.5%`               beta
##              <chr>              <chr>              <chr>              <chr>
## 1 ZM0.1:protocolB -73.5957139540424 -55.6871231046338 -64.0330338372631
## 2   ZM1:protocolB -64.5058694805319 -46.1637598070733 -55.1843371808255
## 3 ZM0.1:protocolC -173.321075342566 -152.377826348389 -162.565235616864
## 4   ZM1:protocolC  -185.13012850059 -163.725458977888 -174.636649804026
## 5 ZM0.1:protocolD -204.317098397391  -177.37385284959 -190.575490650842
## 6   ZM1:protocolD -236.163138923718 -209.358303681769 -222.038289298826
## # ... with 1 more variables: ci.length <dbl>
```

Part III

Because I set iteration=30000, and include many predictors. So the jags would be slow. So I load the data I got. An improvement for this is to adjust the distribution for σ_L^2/λ_l behave like double exponential distribution. The hyperparameter a here is important for adjusting whether we want our model more robust. I chose a=2 for I want to let my model get less sensitive to labs "outliers". I use credible interval to answer these questions.

Prepare the data

```
# Create a data list with inputs for JAGS

X.fac=model.matrix(data=bioassay,object =~EE+ZM+lab+protocol+weight+EE:protocol+ZM:protocol+EE:lab+ZM:la
n=nrow(bioassay)
## scale X such that X^TX has ones on the diagonal;
## scale divides by the standard deviation so we need
## to divide by the sqrt(n-1)
scaled.X = scale(X.fac)/sqrt(n-1)
# are diagonal elements 1?
# check
#t(scaled.X) %*% scaled.X
data = list(Y = bioassay$uterus,
            X=scaled.X,
            p=ncol(scaled.X),
            n = n)

#extract the scales from the scaled object and fix--add to attr
data$scales = attr(scaled.X, "scaled:scale")*sqrt(n-1) # fix scale
data$Xbar = attr(scaled.X, "scaled:center")
```

Jags code

```
##For jags: need to use <-; use precision instead of sigma_sq
rr.model = function() {
  a <- 2
```

```
  shape<-a/2

  for (i in 1:n) {
    mu[i] <- alpha0 + inprod(X[i,], alpha)
    prec[i] <- phi
    Y[i] ~ dnorm(mu[i], prec[i])
  }
  phi ~ dgamma(1.0E-6, 1.0E-6)  ##jags do not allow improper prior
  alpha0 ~ dnorm(0, 1.0E-6)

  for (j in 1:p) {
    phi.l[j] <- pow(i.phi.l[j], -2)
    prec.beta[j] <- lambda.l[j]*phi*phi.l[j]
    alpha[j] ~ dnorm(0, prec.beta[j])
    # transform back to original coefficients
    beta[j] <- alpha[j]/scales[j]
    lambda.l[j] ~ dgamma(shape, shape)
    i.phi.l[j] ~ dt(0,1,1)%_%T(0,)

  }

  # transform intercept to usual parameterization
  beta0 <- alpha0 - inprod(beta[1:p], Xbar)

  sigma <- pow(phi, -.5)
}

# parameters to monitor
parameters = c("beta0", "beta", "sigma","lambda.l", "phi.l")

# run jags from R  (see Resources to install)
stack.sim.hfac = jags(data,
                 inits=NULL,
                 par=parameters,
                 model=rr.model,
                 n.iter=30000)
saveRDS(stack.sim.hfac, "stack.sim.rds")
stack.sim=readRDS("stack.sim.rds")
```

Load the data I stored.

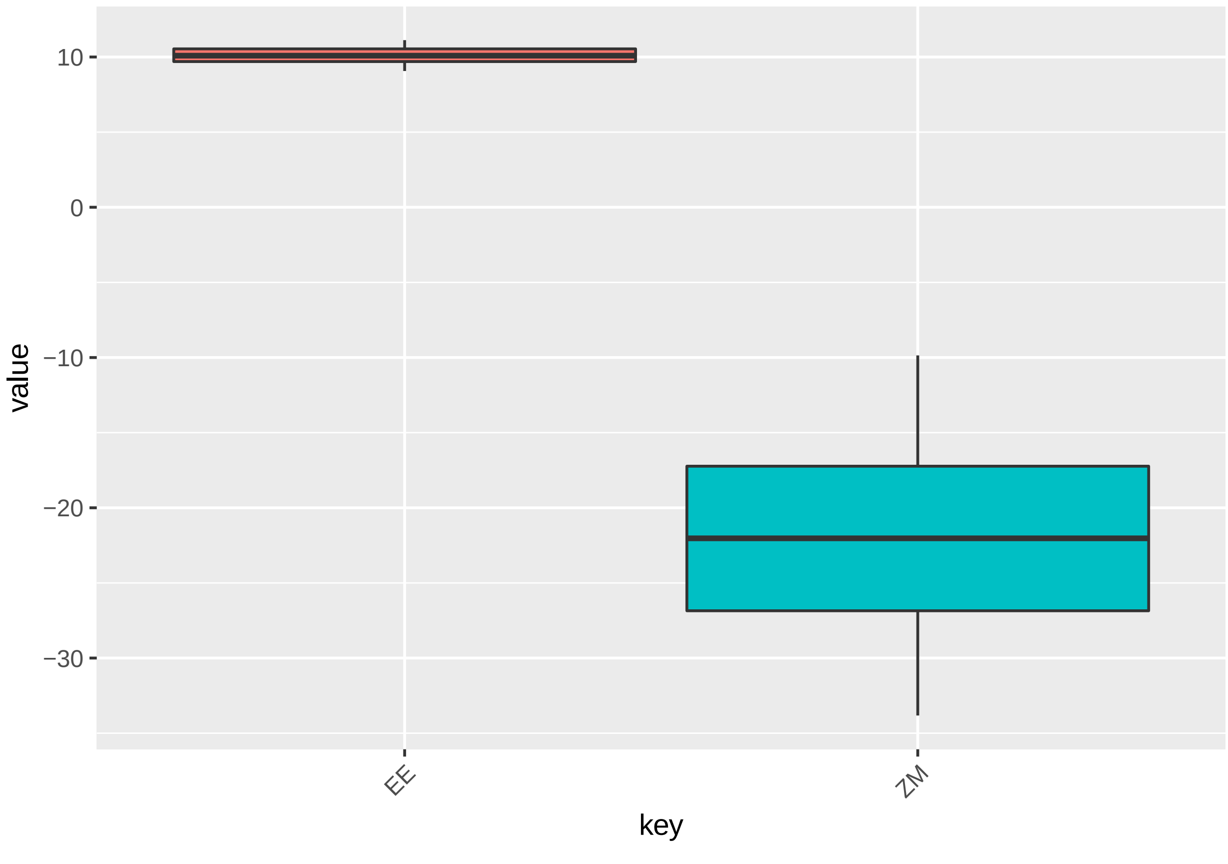
```
stack.sim=readRDS("stack.sim.rds")
```

Analysis on simulation result.

```
# create an MCMC object with the results for the MCMC draws
stack.mcmc = as.mcmc(stack.sim$BUGSoutput$sims.matrix)  #get the simulation points

quan=function(x){
  qu=quantile(x,c(.025, .975))
  avg=mean(x)
  res=c(qu,avg=avg)
  return(res)}
ci.all=apply(stack.mcmc,2,quan) %>% t(.)
mcmc.df=stack.mcmc %>% as.tibble()
```

a.1 The method is successful overall at identifying effects because the credible interval for `EE` is above 0, the credible interval for `ZM` is below 0.

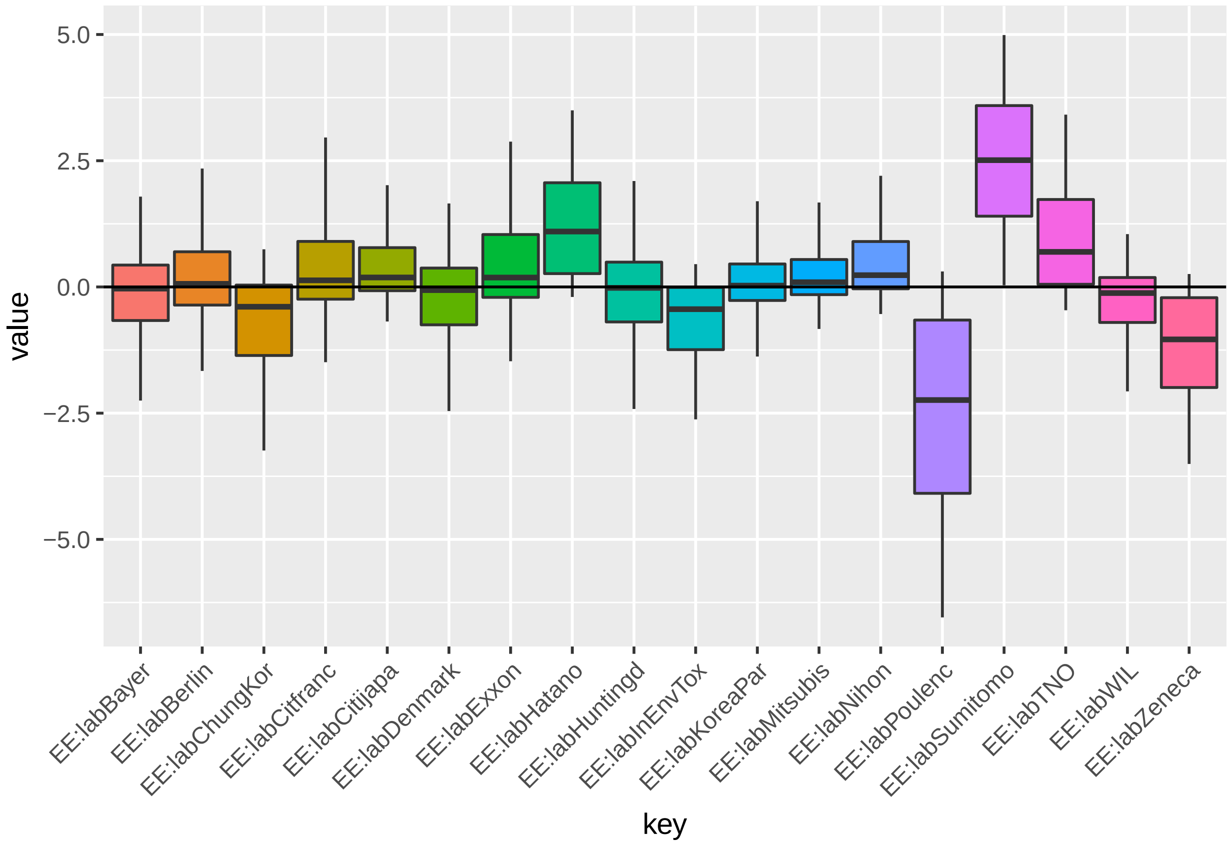
```
##code from https://stackoverflow.com/questions/21310609/ggplot2-box-whisker-plot-show-95-confidence-in

X.fac.jags=model.matrix(data=bioassay,object =~EE+ZM+lab+protocol+weight+EE:protocol+ZM:protocol+EE:lab

quantiles_95 <- function(x) {
  r <- quantile(x, probs=c(0.05, 0.25, 0.5, 0.75, 0.95))
  names(r) <- c("ymin", "lower", "middle", "upper", "ymax")
  r
}

mcmc.df.beta=mcmc.df %>% select(`beta[1]`:`beta[2]`)
names(mcmc.df.beta)=c("EE","ZM")
mcmc.df.beta = mcmc.df.beta %>% gather(.)
ggplot(data = mcmc.df.beta,mapping = aes(x = key,y = value,fill=as.factor(key)))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ guides(fill=FALSE)+stat_summary(fun.data = q
```



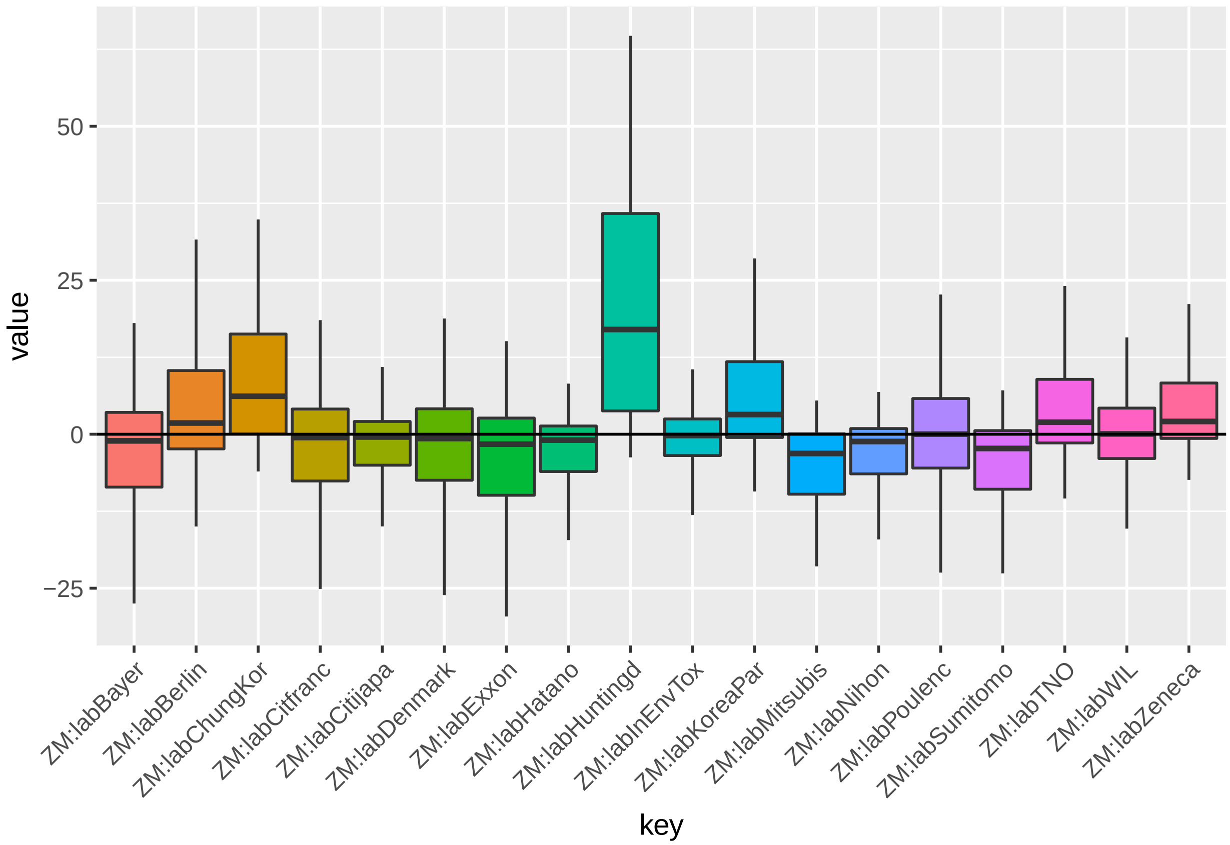
a.2 There are labs Poulenc,ChungKor,EnvTox,Zeneca,KoreaPar,Hungtingd fail to detect such effects.(below the 0)

```
realname=X.fac.jags %>% colnames()
jagsname=mcmc.df %>% names()

temp.ee.lab=jagsname[str_detect(realname,"EE.*lab")]
real.tag=realname[str_detect(realname,"EE.*lab")]
mcmc.df.beta=mcmc.df %>% select(temp.ee.lab[1:18])
```

```
colnames(mcmc.df.beta)=real.tag
mcmc.df.beta = mcmc.df.beta %>% gather(.)
ggplot(data = mcmc.df.beta,mapping = aes(x = key,y = value,fill=as.factor(key)))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ guides(fill=FALSE)+stat_summary(fun.data = c
```

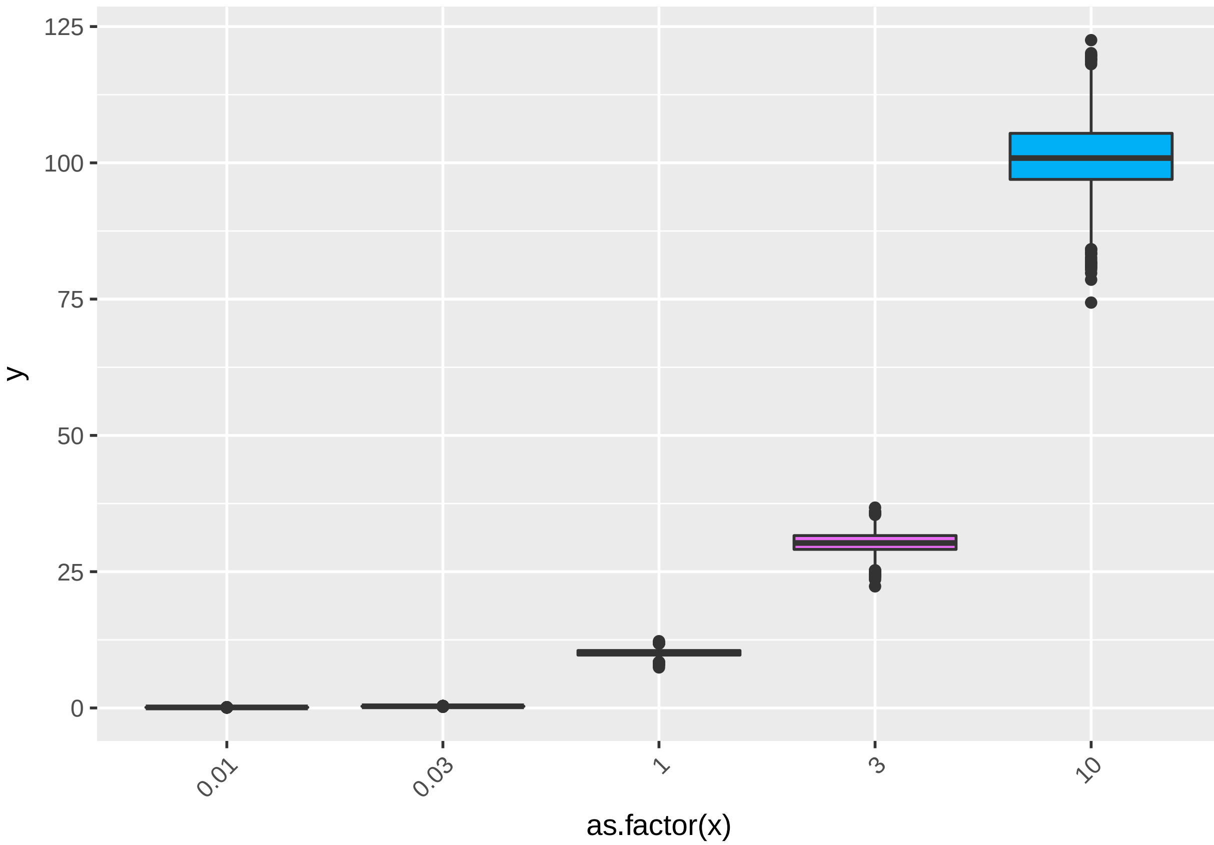



```
temp.zm.lab=jagsname[str_detect(realname,"ZM.*lab")]
real.tag=realname[str_detect(realname,"ZM.*lab")]
mcmc.df.beta=mcmc.df %>% select(temp.zm.lab[1:18])
colnames(mcmc.df.beta)=real.tag
mcmc.df.beta = mcmc.df.beta %>% gather(.)
ggplot(data = mcmc.df.beta,mapping = aes(x = key,y = value,fill=as.factor(key)))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ guides(fill=FALSE)+stat_summary(fun.data = c
```



a.3 what the change point for the dose for `EE`. Under the model that `EE` and `ZM` are continuous, based on the linear regression assumption, we may think 1 is the change point. However, it is not rigorous. This changing point may vary across labs but it is hard to get the conclusion directly.

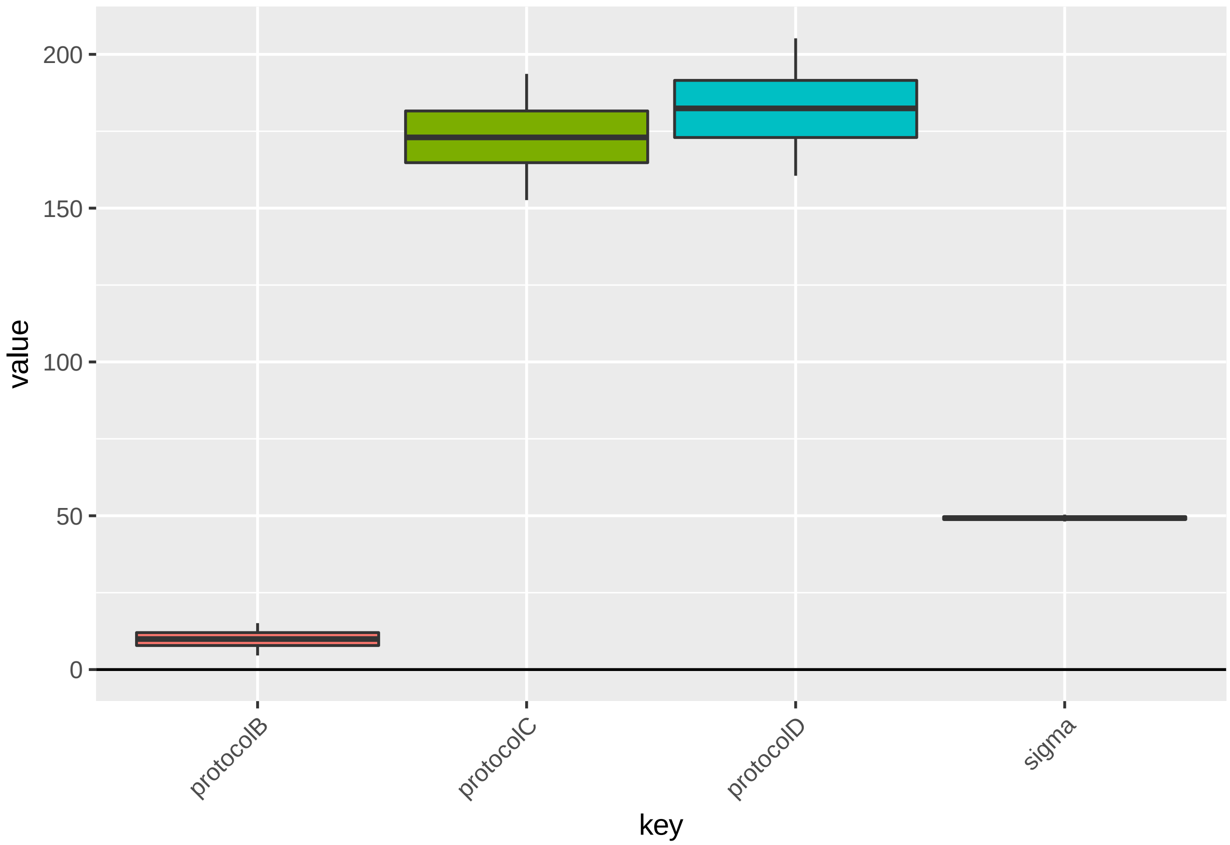
```
mcmc.df.beta=mcmc.df %>% pull(`beta[1]`)
a=0.01*mcmc.df.beta
b=0.03*mcmc.df.beta
c=mcmc.df.beta
d=3*mcmc.df.beta
e=10*mcmc.df.beta
mcmc.df.beta=cbind(y=c(a,b,c,d,e),x=rep(c("0.01","0.03","1","3","10"),each=nrow(mcmc.df))) %>% as.tibbl
  mutate(y=as.numeric(y))

ggplot(data = mcmc.df.beta,mapping = aes(x = as.factor(x),y=y,fill=factor(x)))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ guides(fill=FALSE)+geom_boxplot()+
  scale_x_discrete(limits=c("0.01","0.03","1","3","10"))
```



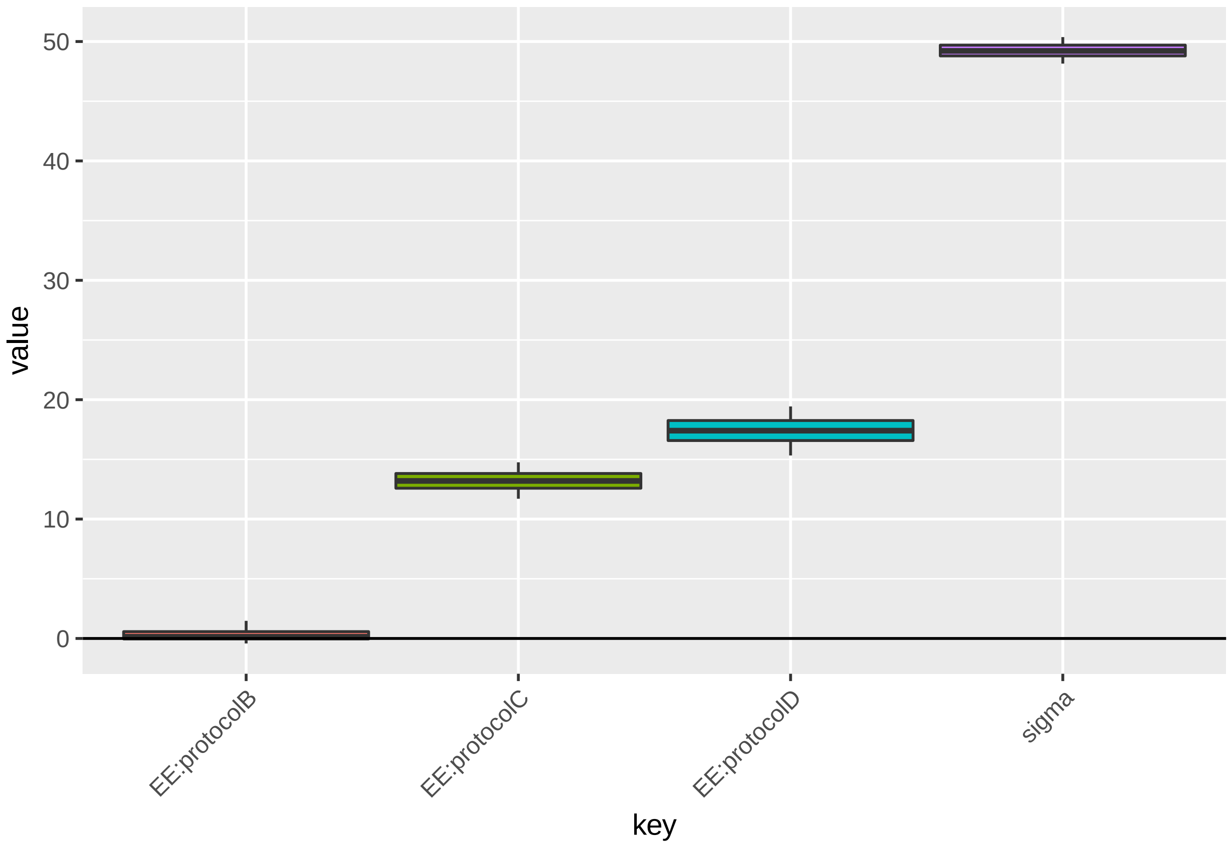
b.The dose response vary across labs. See from the credible interval "boxplot" for a.2. The position and length(with color– 95% credible interval) for each lab is different. Lab Poulenc,Huntingd stands out as being different. However, there are still some overlap between the "outliers" and other labs. So we may say they are not too different from others.

c.The protocol differ in the sensitivity to detecting the effects especially for EE. Protocol B is recommended for its lower variance compared to others.

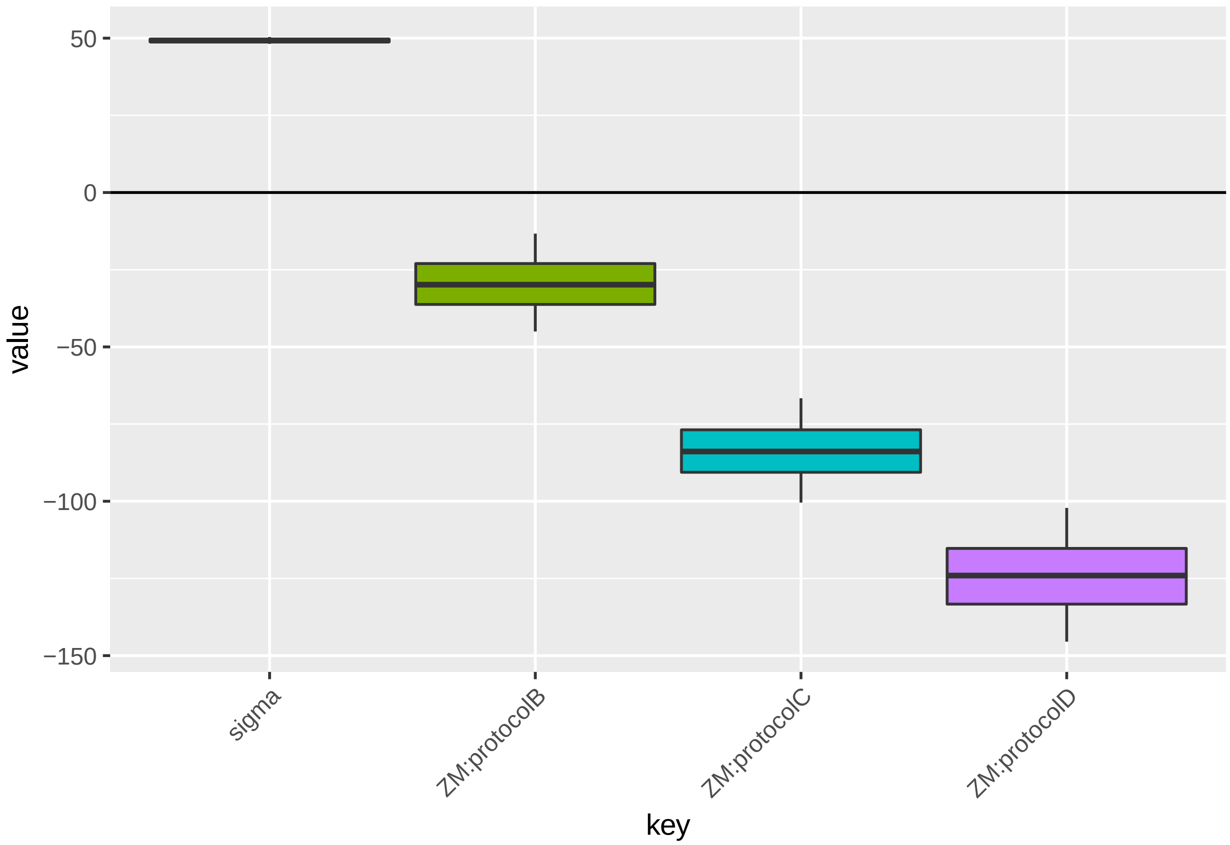
```
#consider sigma as measuring error

temp.ee.lab=jagsname[str_detect(realname,"^protocol")]
real.tag=realname[str_detect(realname,"^protocol")]
mcmc.df.beta=mcmc.df %>% select(temp.ee.lab[1:3],sigma)
colnames(mcmc.df.beta)=c(real.tag,"sigma")
mcmc.df.beta = mcmc.df.beta %>% gather(.)
ggplot(data = mcmc.df.beta,mapping = aes(x = key,y = value,fill=as.factor(key)))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ guides(fill=FALSE)+stat_summary(fun.data = 
```




```
temp.ee.lab=jagsname[str_detect(realname,"EE.*protocol")]
real.tag=realname[str_detect(realname,"EE.*protocol")]
mcmc.df.beta=mcmc.df %>% select(temp.ee.lab[1:3],sigma)
colnames(mcmc.df.beta)=c(real.tag,"sigma")
mcmc.df.beta = mcmc.df.beta %>% gather(.)
ggplot(data = mcmc.df.beta,mapping = aes(x = key,y = value,fill=as.factor(key)))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ guides(fill=FALSE)+stat_summary(fun.data = 
```




```
temp.zm.lab=jagsname[str_detect(realname,"ZM.*protocol")]
real.tag=realname[str_detect(realname,"ZM.*protocol")]
mcmc.df.beta=mcmc.df %>% select(temp.zm.lab[1:3],sigma)
colnames(mcmc.df.beta)=c(real.tag,"sigma")
mcmc.df.beta = mcmc.df.beta %>% gather(.)
ggplot(data = mcmc.df.beta,mapping = aes(x = key,y = value,fill=as.factor(key)))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ guides(fill=FALSE)+stat_summary(fun.data = 
```




```
mcmc.df.lambda=mcmc.df %>% select(`lambda.l[1]`:`lambda.l[66]`) %>% gather(.)
mcmc.df.phi.l=mcmc.df %>% select(`phi.l[1]`:`phi.l[66]`) %>% gather(.)
beta.levels=paste0("beta[",1:66,"]","beta0")
lambda.levels=paste0("lambda.l[",1:66,"]")
phi.l.levels=paste0("phi.l[",1:66,"]")
```

Table for the whole model

```
kable(summary(lm.full.fac)$coefficients, format = "markdown")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	17.5174726	6.0832664	2.8796162	0.0040160
EE0.01	1.7634590	6.2225208	0.2833995	0.7768945
EE0.03	1.0348581	9.8997869	0.1045334	0.9167546
EE0.1	2.7516396	9.9000339	0.2779424	0.7810799
EE0.3	-1.2258396	9.8997530	-0.1238253	0.9014637
EE1	4.9137998	9.8997612	0.4963554	0.6196881
EE3	69.3169544	9.8998665	7.0018070	0.0000000
EE10	85.6474733	9.9001737	8.6511082	0.0000000
ZM0.1	-44.2697078	11.2619189	-3.9309205	0.0000870
ZM1	-68.4662160	11.2618892	-6.0794610	0.0000000
labBayer	13.6089109	8.1539860	1.6689888	0.0952470
labBerlin	5.2019905	8.1706700	0.6366663	0.5244015
labChungKor	17.0236675	7.2154001	2.3593518	0.0183847
labCitfranc	19.9168751	8.1423844	2.4460740	0.0145118

	Estimate	Std. Error	t value	Pr(>|t|)
labCitijapa	13.4596775	6.8936320	1.9524798	0.0509947
labDenmark	9.6122452	8.5189868	1.1283320	0.2592899
labExxon	15.6678556	9.8999699	1.5826165	0.1136376
labHatano	11.9391885	6.7473960	1.7694513	0.0769426
labHuntingd	-8.3391169	8.8013746	-0.9474789	0.3434880
labInEnvTox	4.8104074	6.7951943	0.7079131	0.4790663
labKoreaPar	-2.3948281	7.4778542	-0.3202561	0.7488015
labMitsubis	11.9407498	6.7484149	1.7694155	0.0769486
labNihon	5.9191632	6.7487889	0.8770704	0.3805341
labPoulenc	-3.8337704	8.1513040	-0.4703260	0.6381639
labSumitomo	10.7213694	6.8927463	1.5554568	0.1199661
labTNO	5.1741985	7.2817950	0.7105663	0.4774204
labWIL	6.7631678	7.2179124	0.9369978	0.3488518
labZeneca	-4.0680776	6.9129760	-0.5884698	0.5562711
protocolB	-2.1607713	2.4955878	-0.8658366	0.3866643
protocolC	56.9794121	5.4037785	10.5443649	0.0000000
protocolD	47.3180475	6.1563031	7.6861140	0.0000000
weight	0.1013519	0.0269476	3.7610787	0.0001731
EE0.01:protocolB	1.1910814	4.3163572	0.2759460	0.7826128
EE0.03:protocolB	3.5413070	4.3163543	0.8204394	0.4120453
EE0.1:protocolB	7.4548817	4.3164162	1.7270999	0.0842754
EE0.3:protocolB	35.0983219	4.3163533	8.1314756	0.0000000
EE1:protocolB	58.2145221	4.3163905	13.4868524	0.0000000
EE3:protocolB	34.4029535	4.3163793	7.9703268	0.0000000
EE10:protocolB	17.5749522	4.3279683	4.0607858	0.0000504
EE0.01:protocolC	-0.3230727	4.9003837	-0.0659280	0.9474405
EE0.03:protocolC	2.7225504	4.9003811	0.5555793	0.5785491
EE0.1:protocolC	14.0429743	4.9212596	2.8535325	0.0043598
EE0.3:protocolC	100.2001531	4.9010674	20.4445572	0.0000000
EE1:protocolC	186.2957590	4.9056058	37.9760965	0.0000000
EE3:protocolC	170.0100300	4.9148171	34.5913240	0.0000000
EE10:protocolC	157.5564078	4.9217713	32.0121353	0.0000000
EE0.01:protocolD	6.4547359	6.2222407	1.0373652	0.2996678
EE0.03:protocolD	4.8409616	6.2794259	0.7709242	0.4408260
EE0.1:protocolD	24.4881903	6.2874449	3.8947761	0.0001009
EE0.3:protocolD	140.9911182	6.2315658	22.6253118	0.0000000
EE1:protocolD	230.5357409	6.2588387	36.8336284	0.0000000
EE3:protocolD	208.1653132	6.2771622	33.1623280	0.0000000
EE10:protocolD	216.9748633	6.2947772	34.4690300	0.0000000
ZM0.1:protocolB	-61.0379638	4.9806146	-12.2551068	0.0000000
ZM1:protocolB	-53.8667551	4.9805753	-10.8153681	0.0000000
ZM0.1:protocolC	-158.4145809	5.6799781	-27.8899987	0.0000000
ZM1:protocolC	-171.3240393	5.6613955	-30.2618037	0.0000000
ZM0.1:protocolD	-183.6447648	7.2074293	-25.4799259	0.0000000
ZM1:protocolD	-217.2506012	7.2037058	-30.1581723	0.0000000
EE0.1:labBayer	-1.9456140	13.8970874	-0.1400016	0.8886702
EE0.3:labBayer	-1.0689227	13.8971751	-0.0769165	0.9386962
EE1:labBayer	2.4451549	13.8974734	0.1759424	0.8603537
EE3:labBayer	-28.2023694	13.8970615	-2.0293765	0.0425276
EE10:labBayer	13.3618173	13.8973214	0.9614671	0.3364120
EE0.01:labBerlin	-0.8711178	11.5688730	-0.0752984	0.9399834
EE0.03:labBerlin	2.9075964	13.8970389	0.2092242	0.8342906

	Estimate	Std. Error	t value	Pr(>|t|)
EE0.1:labBerlin	-2.8133528	13.8977646	-0.2024320	0.8395957
EE0.3:labBerlin	15.6240363	13.8971717	1.1242601	0.2610125
EE1:labBerlin	49.4837223	13.8970358	3.5607394	0.0003768
EE3:labBerlin	36.1854101	13.8971988	2.6037916	0.0092754
EE10:labBerlin	5.9294383	13.8987704	0.4266160	0.6696964
EE0.01:labChungKor	-4.4112228	9.0183089	-0.4891408	0.6247857
EE0.03:labChungKor	3.6572980	12.2566224	0.2983936	0.7654280
EE0.1:labChungKor	25.1157672	12.2569386	2.0491061	0.0405577
EE0.3:labChungKor	29.9697260	12.2567299	2.4451649	0.0145484
EE1:labChungKor	33.4910175	12.2566535	2.7324765	0.0063310
EE3:labChungKor	5.8710111	12.2568226	0.4789994	0.6319817
EE10:labChungKor	-13.1352580	12.2576968	-1.0715927	0.2840083
EE0.01:labCitfranc	-5.4581770	11.5691596	-0.4717868	0.6371209
EE0.03:labCitfranc	2.1032608	13.8970370	0.1513460	0.8797152
EE0.1:labCitfranc	-4.7742861	13.8972539	-0.3435417	0.7312204
EE0.3:labCitfranc	4.4922701	13.8970495	0.3232535	0.7465308
EE1:labCitfranc	15.3991170	13.8970550	1.1080849	0.2679336
EE3:labCitfranc	-11.4493741	13.8970501	-0.8238708	0.4100929
EE10:labCitfranc	12.3162322	13.8971999	0.8862384	0.3755758
EE0.01:labCitijapa	-3.0151260	7.9637089	-0.3786083	0.7050115
EE0.03:labCitijapa	-4.5806369	11.6928831	-0.3917457	0.6952801
EE0.1:labCitijapa	-6.1052594	11.6939503	-0.5220870	0.6016568
EE0.3:labCitijapa	-3.2741206	11.6927905	-0.2800119	0.7794919
EE1:labCitijapa	35.0675543	11.6929228	2.9990410	0.0027355
EE3:labCitijapa	5.0502495	11.6927877	0.4319115	0.6658436
EE10:labCitijapa	10.5650698	11.6931765	0.9035244	0.3663362
EE0.01:labDenmark	-2.1078621	11.5430753	-0.1826084	0.8551204
EE0.03:labDenmark	-3.8631575	14.5521183	-0.2654705	0.7906693
EE0.1:labDenmark	-5.2945269	14.5519685	-0.3638358	0.7160119
EE0.3:labDenmark	27.6921013	14.5519290	1.9029849	0.0571593
EE1:labDenmark	40.2593492	14.5522648	2.7665350	0.0057080
EE3:labDenmark	8.8664099	14.5520236	0.6092905	0.5423882
EE10:labDenmark	-2.3109937	14.5553893	-0.1587724	0.8738613
EE0.01:labExxon	5.2457749	12.8665555	0.4077062	0.6835249
EE0.03:labExxon	0.6607488	14.9945279	0.0440660	0.9648554
EE0.1:labExxon	-0.3802441	14.9949332	-0.0253582	0.9797713
EE0.3:labExxon	9.4611994	14.9945142	0.6309774	0.5281140
EE1:labExxon	11.9059065	14.9945225	0.7940171	0.4272622
EE3:labExxon	5.4743729	14.9949929	0.3650801	0.7150831
EE10:labExxon	14.6708826	14.9947799	0.9783993	0.3279732
EE0.01:labHatano	-3.1285617	7.5814767	-0.4126586	0.6798928
EE0.03:labHatano	-3.6387506	11.4378435	-0.3181326	0.7504114
EE0.1:labHatano	-3.0319622	11.4536376	-0.2647161	0.7912504
EE0.3:labHatano	2.9262057	11.4357534	0.2558822	0.7980632
EE1:labHatano	25.9831604	11.4358435	2.2720808	0.0231677
EE3:labHatano	14.2633513	11.4358590	1.2472479	0.2124255
EE10:labHatano	23.8446654	11.4651242	2.0797564	0.0376513
EE0.01:labHuntingd	7.8757716	11.6203523	0.6777567	0.4979898
EE0.03:labHuntingd	-5.1484581	14.7358705	-0.3493827	0.7268320
EE0.1:labHuntingd	-30.0494023	14.7449823	-2.0379409	0.0416627
EE0.3:labHuntingd	-91.0483605	14.7382196	-6.1777042	0.0000000
EE1:labHuntingd	-151.3072461	14.7357224	-10.2680576	0.0000000

	Estimate	Std. Error	t value	Pr(>|t|)
EE3:labHuntingd	-101.1729866	14.7419254	-6.8629425	0.0000000
EE10:labHuntingd	-5.3201467	14.7374840	-0.3609942	0.7181348
EE0.01:labInEnvTox	-5.5008881	7.5810410	-0.7256112	0.4681463
EE0.03:labInEnvTox	-2.4986614	11.4377038	-0.2184583	0.8270902
EE0.1:labInEnvTox	2.0522573	11.4391119	0.1794070	0.8576329
EE0.3:labInEnvTox	8.1025280	11.4357597	0.7085256	0.4786862
EE1:labInEnvTox	35.7901398	11.4359495	3.1296168	0.0017709
EE3:labInEnvTox	5.7978376	11.4362920	0.5069683	0.6122226
EE10:labInEnvTox	-5.1590307	11.4364117	-0.4511057	0.6519532
EE0.01:labKoreaPar	-4.5600764	9.0012641	-0.5066040	0.6124782
EE0.03:labKoreaPar	-9.0474745	12.7018088	-0.7122981	0.4763478
EE0.1:labKoreaPar	-12.0801279	12.7043258	-0.9508673	0.3417653
EE0.3:labKoreaPar	-8.2114601	12.7018086	-0.6464796	0.5180292
EE1:labKoreaPar	32.4911955	12.7020103	2.5579569	0.0105884
EE3:labKoreaPar	-5.7024976	12.7024277	-0.4489297	0.6535219
EE10:labKoreaPar	10.6688331	12.7031098	0.8398599	0.4010686
EE0.01:labMitsubis	-6.9589534	7.5812514	-0.9179162	0.3587528
EE0.03:labMitsubis	-3.6469335	11.4534126	-0.3184146	0.7501975
EE0.1:labMitsubis	-5.4309427	11.4626114	-0.4737963	0.6356873
EE0.3:labMitsubis	10.1528255	11.4359123	0.8878020	0.3747342
EE1:labMitsubis	24.0054736	11.4365053	2.0990218	0.0359168
EE3:labMitsubis	12.0242638	11.4361293	1.0514278	0.2931656
EE10:labMitsubis	10.1999962	11.4358854	0.8919289	0.3725184
EE0.01:labNihon	1.1443408	7.5814228	0.1509401	0.8800353
EE0.03:labNihon	-6.8516033	11.4377109	-0.5990362	0.5492039
EE0.1:labNihon	-8.6410539	11.4380866	-0.7554632	0.4500435
EE0.3:labNihon	-4.3106596	11.4359951	-0.3769379	0.7062523
EE1:labNihon	22.3213216	11.4357866	1.9518834	0.0510655
EE3:labNihon	16.6933209	11.4357639	1.4597469	0.1444875
EE10:labNihon	16.7073132	11.4360027	1.4609399	0.1441598
EE0.01:labPoulenc	-2.6746648	11.5688865	-0.2311947	0.8171828
EE0.03:labPoulenc	-1.0820997	13.8970357	-0.0778655	0.9379414
EE0.1:labPoulenc	-2.1316505	13.8971591	-0.1533875	0.8781053
EE0.3:labPoulenc	15.6353563	13.8971182	1.1250790	0.2606654
EE1:labPoulenc	31.5495444	13.8970358	2.2702355	0.0232795
EE3:labPoulenc	-17.8699366	13.8970993	-1.2858753	0.1986078
EE10:labPoulenc	-25.6412231	13.8972852	-1.8450527	0.0651501
EE0.01:labSumitomo	-0.7738165	7.9635207	-0.0971701	0.9225992
EE0.03:labSumitomo	-1.5638764	11.6928688	-0.1337462	0.8936142
EE0.1:labSumitomo	-5.9424405	11.6939738	-0.5081626	0.6113849
EE0.3:labSumitomo	11.7129107	11.6928318	1.0017172	0.3165788
EE1:labSumitomo	42.0319864	11.6927892	3.5946929	0.0003312
EE3:labSumitomo	24.4487282	11.6928765	2.0909079	0.0366388
EE10:labSumitomo	35.6613733	11.6931561	3.0497646	0.0023147
EE0.01:labTNO	0.1459632	9.0558909	0.0161180	0.9871415
EE0.03:labTNO	-0.9345580	12.2903092	-0.0760402	0.9393933
EE0.1:labTNO	-5.5301417	12.2904691	-0.4499537	0.6527835
EE0.3:labTNO	2.7554038	12.2903618	0.2241922	0.8226264
EE1:labTNO	32.7429120	12.2903774	2.6641096	0.0077697
EE3:labTNO	10.9566903	12.2903462	0.8914875	0.3727550
EE10:labTNO	20.5077951	12.2917541	1.6684189	0.0953600
EE0.01:labWIL	2.3868885	9.0177255	0.2646885	0.7912716

	Estimate	Std. Error	t value	Pr(>|t|)
EE0.03:labWIL	-0.9974044	12.2565949	-0.0813770	0.9351488
EE0.1:labWIL	1.6262677	12.2572149	0.1326784	0.8944586
EE0.3:labWIL	-0.5899327	12.2566676	-0.0481316	0.9616153
EE1:labWIL	13.0420573	12.2566878	1.0640768	0.2873985
EE3:labWIL	-12.0139275	12.2569909	-0.9801694	0.3270990
EE10:labWIL	4.3028315	12.2582459	0.3510153	0.7256070
EE0.03:labZeneca	0.2504818	11.6928837	0.0214217	0.9829110
EE0.1:labZeneca	1.1680882	11.6941543	0.0998865	0.9204426
EE0.3:labZeneca	13.2661190	11.6928050	1.1345540	0.2566729
EE1:labZeneca	21.7877479	11.6928006	1.8633473	0.0625326
EE3:labZeneca	-23.5618392	11.6930187	-2.0150348	0.0440097
EE10:labZeneca	-16.9092188	11.6936855	-1.4460128	0.1483012
ZM0.1:labBayer	50.8727461	15.9269030	3.1941392	0.0014202
ZM1:labBayer	32.7792790	15.9267178	2.0581315	0.0396828
ZM0.1:labBerlin	27.7253380	15.9267127	1.7408073	0.0818425
ZM1:labBerlin	-2.2226356	15.9267230	-0.1395539	0.8890239
ZM0.1:labChungKor	57.6940393	14.0160636	4.1162798	0.0000398
ZM1:labChungKor	33.0262842	14.0160945	2.3563115	0.0185356
ZM0.1:labCitfranc	30.2045937	15.9267649	1.8964676	0.0580154
ZM1:labCitfranc	12.7354950	15.9267284	0.7996303	0.4240023
ZM0.1:labCitijapa	31.8017854	13.3591547	2.3805238	0.0173639
ZM1:labCitijapa	7.7386338	13.3582359	0.5793155	0.5624294
ZM0.1:labDenmark	23.1410049	16.6873655	1.3867381	0.1656474
ZM1:labDenmark	8.2176564	16.6873351	0.4924487	0.6224462
ZM0.1:labExxon	23.0663308	15.9269827	1.4482549	0.1476734
ZM1:labExxon	-5.2036023	15.9268284	-0.3267193	0.7439080
ZM0.1:labHatano	9.0102103	13.0802294	0.6888419	0.4909878
ZM1:labHatano	0.3096981	13.0582014	0.0237167	0.9810805
ZM0.1:labHuntingd	72.4342844	16.9076766	4.2841063	0.0000191
ZM1:labHuntingd	114.0571430	16.9017085	6.7482612	0.0000000
ZM0.1:labInEnvTox	23.3265752	13.0584338	1.7863226	0.0741703
ZM1:labInEnvTox	10.2372871	13.0581623	0.7839761	0.4331297
ZM0.1:labKoreaPar	16.1498041	14.5364690	1.1109854	0.2666834
ZM1:labKoreaPar	40.0390337	14.5346073	2.7547379	0.0059172
ZM0.1:labMitsubis	13.2492104	13.0584341	1.0146094	0.3103919
ZM1:labMitsubis	-5.9918796	13.0581623	-0.4588609	0.6463747
ZM0.1:labNihon	12.5949318	13.0584778	0.9645023	0.3348891
ZM1:labNihon	-4.3232424	13.0581774	-0.3310755	0.7406157
ZM0.1:labPoulenc	49.0113737	15.9267124	3.0773064	0.0021118
ZM1:labPoulenc	29.6003373	15.9267334	1.8585316	0.0632131
ZM0.1:labSumitomo	30.8781749	13.3592364	2.3113727	0.0208946
ZM1:labSumitomo	-7.6949612	13.3582819	-0.5760442	0.5646380
ZM0.1:labTNO	77.4798497	14.0160034	5.5279560	0.0000000
ZM1:labTNO	22.6405850	14.0159523	1.6153440	0.1063646
ZM0.1:labWIL	38.1873945	14.0159664	2.7245638	0.0064843
ZM1:labWIL	29.8072517	14.0159523	2.1266662	0.0335472
ZM0.1:labZeneca	77.9602706	13.3595517	5.8355454	0.0000000
ZM1:labZeneca	44.7841288	13.3582788	3.3525374	0.0008129

```r
bas.coef.need=bas.coef[,-5]
kable((bas.coef.need), format = "markdown")
```

coef	2.5%	97.5%	beta
Intercept	100.055568813859	101.603060803598	100.828516996638
EE0.01	0	0	-0.0169758635844599
EE0.03	0	0	-0.0192535509875797
EE0.1	-0.0201655874525166	2.38321390824135	0.200356733960047
EE0.3	0	7.33548937968884	1.29791927074527
EE1	22.7501239683756	33.684255463196	28.2664308668672
EE3	65.0864346675953	75.8953375030995	70.4771048996914
EE10	92.1335647695202	103.686716370142	98.0191720139301
ZM0.1	-26.168932843301	-13.0415050978623	-19.7329611000799
ZM1	-67.1789624372026	-53.6152392509271	-60.3267641725413
labBayer	-0.0100306229492202	13.2320560329467	1.6445320586086
labBerlin	0	14.7357212781404	1.72937557635339
labChungKor	9.05968241677786	25.611006321204	16.1087922188008
labCitfranc	7.53227884380485	23.5528039597407	14.8577276601501
labCitijapa	5.6967095620807	19.9149898297334	11.1662872996414
labDenmark	0	17.4602155291683	5.39745865577653
labExxon	7.22878169434169	23.1746916266721	14.5131836777907
labHatano	7.04681057756988	19.9902888160697	12.0015985076056
labHuntingd	-14.4799070954698	0.141921156377982	-1.91427075365849
labInEnvTox	0	12.702323614936	1.758854129823
labKoreaPar	-15.5337425307652	0	-9.82033416819739
labMitsubis	4.78279166856804	17.7870731590992	9.97372874396377
labNihon	0	11.3700766421822	1.51753543906021
labPoulenc	-11.0245052575361	0	-2.49192021569124
labSumitomo	4.33643558263054	21.8668684634691	11.62373250001
labTNO	0	13.5560379798574	2.10891477555543
labWIL	0	10.6148232939581	1.26530854623111
labZeneca	-7.37252559240579	0	-0.524627049764824
protocolB	-0.54046287842722	0.0895778065584104	-0.0721794821573571
protocolC	54.2791101029415	77.8983038902862	69.9453128512877
protocolD	45.9447833358295	73.1551422639078	63.6302991062278
weight	0	0.109755834462574	0.0200066552342362
EE0.01:protocolB	0	0	-0.0580646028824629
EE0.03:protocolB	0	0	-0.00783646148478639
EE0.1:protocolB	0	5.84581272698972	0.584951728986413
EE0.3:protocolB	26.9241930774477	40.9172070895798	34.2580916086904
EE1:protocolB	54.6311461919624	68.1905290200169	61.4867093062725
EE3:protocolB	30.0642593477745	43.3277363201775	36.8148784557911
EE10:protocolB	8.52790074748414	22.0889966047053	15.3405036938238
EE0.01:protocolC	0	0	-0.0046476724141366
EE0.03:protocolC	0	0	-0.00543334813404808
EE0.1:protocolC	5.72336409178617	22.0076864017156	13.7904554310719
EE0.3:protocolC	92.4191960013764	108.997491835907	100.87475623571
EE1:protocolC	182.160778915656	198.713486228357	190.39393079343
EE3:protocolC	166.744653072946	183.562267885563	174.992945517899
EE10:protocolC	147.156727189328	164.491865637864	156.064782400309
EE0.01:protocolD	0	0	0.206757083407791
EE0.03:protocolD	0	0	0.00414693052379268
EE0.1:protocolD	18.4919352657321	37.6715090199483	27.9860566371162
EE0.3:protocolD	128.372508118059	149.597055076892	139.193809517336
EE1:protocolD	218.060448293349	239.567667397686	228.867704775014
EE3:protocolD	201.381795775679	223.028188196978	212.235082822263

coef	2.5%	97.5%	beta
EE10:protocolD	202.351776166603	225.021591017432	213.705137201872
ZM0.1:protocolB	-73.5957139540424	-55.6871231046338	-64.0330338372631
ZM1:protocolB	-64.5058694805319	-46.1637598070733	-55.1843371808255
ZM0.1:protocolC	-173.321075342566	-152.377826348389	-162.565235616864
ZM1:protocolC	-185.13012850059	-163.725458977888	-174.636649804026
ZM0.1:protocolD	-204.317098397391	-177.37385284959	-190.575490650842
ZM1:protocolD	-236.163138923718	-209.358303681769	-222.038289298826
EE0.01:labBayer	0	0	0
EE0.03:labBayer	0	0	0
EE0.1:labBayer	-0.0192330063186859	6.09117439261171	0.649253051740099
EE0.3:labBayer	0	0	0.284889753275117
EE1:labBayer	-15.3424142931042	0.0693614861939942	-1.42586457087195
EE3:labBayer	-2.28637297497919	0.166924317079962	-0.471791526342864
EE10:labBayer	-0.25635508984103	10.1812227369337	0.86594831213598
EE0.01:labBerlin	0	0	0.0134094845633751
EE0.03:labBerlin	0	0	0.14879602681579
EE0.1:labBerlin	0	0	-0.00575866444054121
EE0.3:labBerlin	0	18.2178739461399	1.97299618449755
EE1:labBerlin	0	39.3880462235636	18.9905664883612
EE3:labBerlin	20.4486914208013	43.4701129376506	32.1811025558036
EE10:labBerlin	-3.22634360249806	0.238289056069942	-0.452854164813359
EE0.01:labChungKor	-0.32083819448043	0.682801752859145	-0.308737235913504
EE0.03:labChungKor	0	0	0.225035347068756
EE0.1:labChungKor	15.1891208883723	42.1763845195512	28.1421734827984
EE0.3:labChungKor	0	35.8722719134734	24.0081719880751
EE1:labChungKor	-0.273531878823031	6.56776810252113	0.512957874183628
EE3:labChungKor	-3.10654935841784	6.29499082255001	0.539527082513995
EE10:labChungKor	-44.0305057638951	-12.4866639946116	-27.1170341020573
EE0.01:labCitfranc	0	0	-0.0207745970158964
EE0.03:labCitfranc	-0.844164003439628	0.109560866174454	0.389033044220412
EE0.1:labCitfranc	0	0	0.0451621132528955
EE0.3:labCitfranc	0	0	0.264017560445175
EE1:labCitfranc	0	0	-0.276087027231151
EE3:labCitfranc	-11.1434886902081	0	-1.0564901297952
EE10:labCitfranc	0	0	0.129870725128225
EE0.01:labCitijapa	0	0	-0.0628240427250146
EE0.03:labCitijapa	0	0	-0.0886381708987131
EE0.1:labCitijapa	0	0	-0.0909485642002586
EE0.3:labCitijapa	-10.3534214031258	0	-0.954981697265184
EE1:labCitijapa	-0.00736458599237544	12.0497840993354	1.35945524140029
EE3:labCitijapa	0	5.74546611538903	0.517404613163833
EE10:labCitijapa	0	0	-0.111455766846516
EE0.01:labDenmark	0	0	0.0385491403096029
EE0.03:labDenmark	0	0	0.0291353273720798
EE0.1:labDenmark	0	0	0.154375704891662
EE0.3:labDenmark	0	41.4327900269031	18.7540233365045
EE1:labDenmark	0	18.6359882462655	1.97450071012038
EE3:labDenmark	0	11.9582225274624	1.13737525826132
EE10:labDenmark	-24.8588418893432	0.00650881494025413	-3.16136230573142
EE0.01:labExxon	-0.137604061801549	0.269203463317763	0.372586851489187
EE0.03:labExxon	0	0	0.0554263141342627
EE0.1:labExxon	0	0	0.0742159963778267

coef	2.5%	97.5%	beta
EE0.3:labExxon	0	0	0.285468264871381
EE1:labExxon	-18.2381519200407	0.0116267099549656	-1.69183959362337
EE3:labExxon	0	0	-0.0695735290484854
EE10:labExxon	0	0	0.0452968593956899
EE0.01:labHatano	0	0	-0.0950939048950646
EE0.03:labHatano	0	0	-0.117406472162141
EE0.1:labHatano	0	6.68831489966824	0.541967220917631
EE0.3:labHatano	0	0	-0.105150901203125
EE1:labHatano	-0.779601602491132	0.918934916263321	-0.189619793123771
EE3:labHatano	-0.0136386782915936	1.58936201621064	0.192265675908894
EE10:labHatano	0	17.1270308125972	3.35744835668047
EE0.01:labHuntingd	-0.223816355703703	9.40122937703189	0.813033223742339
EE0.03:labHuntingd	0	0	-0.105259360431332
EE0.1:labHuntingd	-45.2830875706243	0	-25.0412429170108
EE0.3:labHuntingd	-115.841497305509	-77.2513496531074	-95.8410338981138
EE1:labHuntingd	-201.469972146317	-162.718273480607	-181.669359374644
EE3:labHuntingd	-130.017685659601	-90.0742466231031	-109.912163920543
EE10:labHuntingd	-36.4857747223026	0	-11.3264971658355
EE0.01:labInEnvTox	-0.664000501787222	0.448704964256441	-0.222310727764797
EE0.03:labInEnvTox	0	0	-0.0143856945668852
EE0.1:labInEnvTox	0	1.50049032367342	0.23487579211051
EE0.3:labInEnvTox	-0.0629666308739298	0.810143754222619	0.246741911411417
EE1:labInEnvTox	0	13.0836822747519	2.00580078508329
EE3:labInEnvTox	-0.0709735317849667	0.66120974268433	0.147514018788103
EE10:labInEnvTox	-31.3932648034841	-8.702624297668	-19.1861730815806
EE0.01:labKoreaPar	0	0	0.0563963986085987
EE0.03:labKoreaPar	0	0	-0.0798846399411792
EE0.1:labKoreaPar	0	0	-0.12732221244247
EE0.3:labKoreaPar	-11.446793582909	0.0626598101911906	-1.0192915136491
EE1:labKoreaPar	0	13.0808379728296	1.27850367125342
EE3:labKoreaPar	-18.2600787201122	0	-4.24509282013961
EE10:labKoreaPar	0	0	0.185416367824757
EE0.01:labMitsubis	0	0.207404699141916	-0.196948479477514
EE0.03:labMitsubis	0	0	-0.00443799337019274
EE0.1:labMitsubis	-0.958627289686008	0.489111458467349	0.199131273312905
EE0.3:labMitsubis	0	11.5408657535038	1.44100400204005
EE1:labMitsubis	0	0	-0.149250683866199
EE3:labMitsubis	0	0	0.0381656555020163
EE10:labMitsubis	-1.66213031099275	0.00286632650484986	-0.214497081986298
EE0.01:labNihon	-0.0574614649637715	7.12516808449458	0.628047108003837
EE0.03:labNihon	0	0	-0.0590983201971608
EE0.1:labNihon	-3.26570889509142	0	-0.318509539223463
EE0.3:labNihon	-10.058666800277	0	-0.942910717538512
EE1:labNihon	0	0	-0.173738498769824
EE3:labNihon	-0.00730932746730772	12.6178049139012	4.89504185819902
EE10:labNihon	0	11.0548265462707	1.05876247394118
EE0.01:labPoulenc	0	0	-0.248025857208697
EE0.03:labPoulenc	0	0	-0.200874332337052
EE0.1:labPoulenc	-0.123151847540238	0.932064481803846	-0.226425002952008
EE0.3:labPoulenc	0	12.4376710392676	0.979887237381186
EE1:labPoulenc	0	0	0.240240603814661
EE3:labPoulenc	-17.5038435139962	0	-2.9333779133787

coef	2.5%	97.5%	beta
EE10:labPoulenc	-61.2191364544682	-22.6166348709297	-42.4015177572131
EE0.01:labSumitomo	0	0	-0.123632258331759
EE0.03:labSumitomo	0	0	-0.0839931272024959
EE0.1:labSumitomo	-3.17411400635819	0	-0.364914231244223
EE0.3:labSumitomo	-0.108812342469504	9.89976371759892	0.856382867329626
EE1:labSumitomo	0	21.5697632790841	5.86859529418669
EE3:labSumitomo	0	28.0545153830275	12.9382805479146
EE10:labSumitomo	0	29.9589139208255	17.9591724215056
EE0.01:labTNO	0	0	0.0606905930738389
EE0.03:labTNO	0	0	0.0331218335634575
EE0.1:labTNO	0	0	-0.0425468758181033
EE0.3:labTNO	0	0	-0.0528154012330708
EE1:labTNO	-0.261290253046672	5.25207755058647	0.515739825028342
EE3:labTNO	0	20.638409990953	4.91144496256794
EE10:labTNO	-0.148342931963821	10.8365978492384	1.02534131630064
EE0.01:labWIL	0	4.91498097805954	0.48601757665364
EE0.03:labWIL	0	0	0.193493678424892
EE0.1:labWIL	0	12.5784527594793	1.28321215433
EE0.3:labWIL	0	0	-0.0375868940901707
EE1:labWIL	-15.1005973835266	0	-1.66852250765836
EE3:labWIL	0	0	-0.167152909103676
EE10:labWIL	-5.02491348439465	0.0579496452488986	-0.433805538960288
EE0.01:labZeneca	0	0	-0.100368414243997
EE0.03:labZeneca	0	0	-0.0211016111204978
EE0.1:labZeneca	-0.117729646681655	0.415027098567916	0.202171779303242
EE0.3:labZeneca	-0.0508185854094894	11.5675671949487	1.16220687249318
EE1:labZeneca	-15.3757482548871	0	-2.27056692361165
EE3:labZeneca	-43.4755230276264	-21.0154265488135	-32.1111582154707
EE10:labZeneca	-45.2978675316412	-21.5831008997629	-33.1607623369162
ZM0.1:labBayer	0	0.512992752827383	0.447403732231507
ZM1:labBayer	0	0	0.299796765249484
ZM0.1:labBerlin	0	0	0.428121908157927
ZM1:labBerlin	-8.81442383458196	0.0144526830625811	-0.787272849279506
ZM0.1:labChungKor	19.7062726259964	49.8950899882686	34.4507349954458
ZM1:labChungKor	0	37.2145735474906	24.8944065232626
ZM0.1:labCitfranc	0	0	-0.12467183578043
ZM1:labCitfranc	0	0	-0.197442791187807
ZM0.1:labCitijapa	0	17.0199481068083	3.09754241434915
ZM1:labCitijapa	0	0	0.0567535986781877
ZM0.1:labDenmark	0	0	0.33522468756519
ZM1:labDenmark	0	5.22866636685851	0.37094649923403
ZM0.1:labExxon	0	0	0.0775569174946192
ZM1:labExxon	-9.90760345091554	0.0394599009167749	-0.886242849520658
ZM0.1:labHatano	-6.57049142583528	0	-0.5808326120138
ZM1:labHatano	0	0	0.0987711491340328
ZM0.1:labHuntingd	26.9460106929581	78.2262186916054	53.0079711136223
ZM1:labHuntingd	84.9028683705161	135.598097856561	110.109412259374
ZM0.1:labInEnvTox	0	0	0.104438912750612
ZM1:labInEnvTox	-0.0321903714723832	1.61567792977855	0.301153112678961
ZM0.1:labKoreaPar	-15.5999441682147	0	-1.47027216254143
ZM1:labKoreaPar	14.7038534957775	48.5593976116653	31.0224831058589
ZM0.1:labMitsubis	0	0	-0.0399365680407591

coef	2.5%	97.5%	beta
ZM1:labMitsubis	-6.70273604620403	0	-0.529146538346669
ZM0.1:labNihon	-0.0142988067162872	0.0645177511731565	-0.0184772897908792
ZM1:labNihon	-2.17787737000263	1.21467157219097	-0.257860507458655
ZM0.1:labPoulenc	-0.536233953061038	2.41173230515536	0.477268102123288
ZM1:labPoulenc	0	0	0.170299678852661
ZM0.1:labSumitomo	0	29.1489199205844	9.53495946796132
ZM1:labSumitomo	-26.8724344719627	0	-7.66562095472133
ZM0.1:labTNO	36.1382707192728	73.2073026703936	56.4919259170217
ZM1:labTNO	0	31.9857245554529	14.5223460946343
ZM0.1:labWIL	-2.42280513508413	0.059557926500855	0.252024567515321
ZM1:labWIL	0	15.6050314179401	2.01155528718698
ZM0.1:labZeneca	41.4958789859153	70.7020021204021	55.9270327112521
ZM1:labZeneca	24.4235935817464	53.4915883038163	38.9499848095781

```r
kable((ci.all), format = "markdown")
```

	2.5%	97.5%	avg
beta[1]	8.8365014	11.3157390	1.010393e+01
beta[2]	-35.5481331	-7.2303389	-2.194679e+01
beta[3]	-15.3633936	6.2030236	-2.444652e+00
beta[4]	-10.0607058	9.0074942	-2.403781e-01
beta[5]	1.8639767	27.2705865	1.497987e+01
beta[6]	-4.1845423	18.9069612	4.652796e+00
beta[7]	-2.6668646	11.7070554	2.520891e+00
beta[8]	-6.1158319	13.3485869	1.811556e+00
beta[9]	-3.7520377	19.7476013	4.780068e+00
beta[10]	-1.2758324	16.0327748	5.710292e+00
beta[11]	-42.7132883	-2.3263277	-2.378664e+01
beta[12]	-20.4362790	-0.1418263	-1.054050e+01
beta[13]	-28.7650802	-3.7929756	-1.666659e+01
beta[14]	-0.9748967	15.1518507	5.382818e+00
beta[15]	-10.8134752	2.6238819	-2.184642e+00
beta[16]	-27.6174202	1.8925654	-1.096457e+01
beta[17]	-1.5918537	16.9987647	5.536941e+00
beta[18]	-8.0376887	6.4812404	-3.166028e-01
beta[19]	-12.7628959	3.4434788	-2.617430e+00
beta[20]	-6.1073140	8.0502651	2.797168e-01
beta[21]	3.4192654	15.8485231	9.865203e+00
beta[22]	148.5505553	197.4493047	1.732028e+02
beta[23]	156.7079306	209.3245686	1.824887e+02
beta[24]	-0.4820041	-0.2084486	-3.450777e-01
beta[25]	-0.5957285	1.7850036	3.008687e-01
beta[26]	11.4397456	15.0258903	1.320351e+01
beta[27]	14.9067883	19.8549021	1.740912e+01
beta[28]	-48.0333386	-9.1079543	-2.953165e+01
beta[29]	-103.9855754	-63.0491960	-8.379720e+01
beta[30]	-150.1584332	-97.6835043	-1.241106e+02
beta[31]	-2.9856505	2.3114884	-1.305104e-01
beta[32]	-2.2713833	2.9903560	1.859151e-01
beta[33]	-3.7434507	1.1062846	-7.452749e-01
beta[34]	-2.0321831	3.4970301	3.746349e-01

	2.5%	97.5%	avg
beta[35]	-1.0194131	2.4600349	3.871943e-01
beta[36]	-2.9463864	2.2423009	-2.121780e-01
beta[37]	-1.9510140	3.5155818	4.463132e-01
beta[38]	-0.3819274	3.8938185	1.270870e+00
beta[39]	-3.1878659	2.9103142	-8.444010e-02
beta[40]	-3.1058070	0.7042954	-6.952802e-01
beta[41]	-1.7727653	2.1796425	1.003206e-01
beta[42]	-1.1492605	2.0971465	2.289696e-01
beta[43]	-0.7790169	2.6057144	4.854920e-01
beta[44]	-7.3468828	0.6817587	-2.543417e+00
beta[45]	-0.0937489	5.4192344	2.509969e+00
beta[46]	-0.7213705	3.9108728	9.959124e-01
beta[47]	-2.6193193	1.4260846	-2.929527e-01
beta[48]	-3.9167638	0.4977894	-1.225197e+00
beta[49]	-34.7920548	23.1764572	-2.796734e+00
beta[50]	-19.8943344	37.4853244	4.577191e+00
beta[51]	-9.3500232	40.9550427	9.313243e+00
beta[52]	-32.9137200	25.5981573	-1.890928e+00
beta[53]	-18.8270608	13.9090749	-1.440328e+00
beta[54]	-32.5282429	24.9504446	-2.018041e+00
beta[55]	-36.8170466	20.9541657	-4.055351e+00
beta[56]	-21.2480567	11.0096543	-2.588106e+00
beta[57]	-8.4137759	73.5901816	2.194932e+01
beta[58]	-16.2322733	14.0881307	-5.374156e-01
beta[59]	-13.0663692	33.8230748	6.002765e+00
beta[60]	-25.5476929	8.5578773	-5.282263e+00
beta[61]	-21.0228062	9.3443733	-2.981419e+00
beta[62]	-29.0936462	30.4469497	1.402748e-01
beta[63]	-27.4113816	10.7286188	-4.606928e+00
beta[64]	-13.7362072	29.7183498	4.195145e+00
beta[65]	-20.4704319	20.0447721	8.311150e-02
beta[66]	-10.0683533	26.2389378	4.175395e+00
beta0	47.6508185	65.7375120	5.691318e+01
deviance	28440.2695375	28482.0774121	2.845916e+04
lambda.l[1]	0.0031882	2.3446782	5.105991e-01
lambda.l[2]	0.0101551	2.7390222	6.457556e-01
lambda.l[3]	0.0431421	3.9608306	1.146287e+00
lambda.l[4]	0.0587005	3.8361176	1.193139e+00
lambda.l[5]	0.0215315	3.0631228	7.728083e-01
lambda.l[6]	0.0405556	3.8961244	1.096224e+00
lambda.l[7]	0.0477371	3.9363378	1.108687e+00
lambda.l[8]	0.0513177	3.9837403	1.166479e+00
lambda.l[9]	0.0487740	3.9798069	1.079830e+00
lambda.l[10]	0.0291561	3.6682041	9.979569e-01
lambda.l[11]	0.0154245	3.0290944	7.303916e-01
lambda.l[12]	0.0192503	3.2488954	7.862111e-01
lambda.l[13]	0.0171977	3.0697937	7.524282e-01
lambda.l[14]	0.0308350	3.3008325	9.317961e-01
lambda.l[15]	0.0434057	3.7380499	1.068904e+00
lambda.l[16]	0.0308423	3.4736655	9.360505e-01
lambda.l[17]	0.0316535	3.5901378	9.854693e-01
lambda.l[18]	0.0532298	4.0897691	1.170507e+00

	2.5%	97.5%	avg
lambda.l[19]	0.0480695	3.7414024	1.109800e+00
lambda.l[20]	0.0541571	4.0251378	1.157940e+00
lambda.l[21]	0.0157709	2.7868826	6.890049e-01
lambda.l[22]	0.0016490	2.5279695	5.050090e-01
lambda.l[23]	0.0016353	2.3848564	4.976127e-01
lambda.l[24]	0.0030498	2.6038984	5.328624e-01
lambda.l[25]	0.0488746	3.8373979	1.108734e+00
lambda.l[26]	0.0037399	2.5452039	5.480747e-01
lambda.l[27]	0.0044487	2.6285197	5.595168e-01
lambda.l[28]	0.0169323	2.9798364	7.058398e-01
lambda.l[29]	0.0063239	2.6948191	5.880865e-01
lambda.l[30]	0.0063801	2.4701565	5.597687e-01
lambda.l[31]	0.0577059	4.0198951	1.131808e+00
lambda.l[32]	0.0484858	4.0138294	1.152321e+00
lambda.l[33]	0.0448972	3.9895523	1.103072e+00
lambda.l[34]	0.0551721	4.0831373	1.170724e+00
lambda.l[35]	0.0564754	4.0943049	1.156631e+00
lambda.l[36]	0.0473066	4.0688921	1.172528e+00
lambda.l[37]	0.0483418	4.1778554	1.170805e+00
lambda.l[38]	0.0337724	3.5870260	9.853317e-01
lambda.l[39]	0.0622667	4.0493989	1.152929e+00
lambda.l[40]	0.0422016	4.0479546	1.104503e+00
lambda.l[41]	0.0517579	3.5826133	1.120556e+00
lambda.l[42]	0.0553602	3.8964854	1.145804e+00
lambda.l[43]	0.0434383	3.6793352	1.082284e+00
lambda.l[44]	0.0322688	3.3768723	9.522361e-01
lambda.l[45]	0.0245242	3.1545299	8.396896e-01
lambda.l[46]	0.0404574	3.6502095	1.074084e+00
lambda.l[47]	0.0467243	4.1091030	1.160815e+00
lambda.l[48]	0.0449567	3.6406617	9.965603e-01
lambda.l[49]	0.0537043	3.8452006	1.163394e+00
lambda.l[50]	0.0526893	4.0465729	1.141596e+00
lambda.l[51]	0.0428696	3.9683189	1.111151e+00
lambda.l[52]	0.0527862	3.8839950	1.159493e+00
lambda.l[53]	0.0545172	3.9389200	1.194681e+00
lambda.l[54]	0.0535716	4.1943350	1.177619e+00
lambda.l[55]	0.0632276	3.9625311	1.179642e+00
lambda.l[56]	0.0595880	3.7703189	1.127434e+00
lambda.l[57]	0.0401704	3.7402610	1.043949e+00
lambda.l[58]	0.0509127	3.8799609	1.157144e+00
lambda.l[59]	0.0503856	3.8722314	1.145341e+00
lambda.l[60]	0.0499328	3.6672722	1.121327e+00
lambda.l[61]	0.0581146	3.8858736	1.126073e+00
lambda.l[62]	0.0549518	3.6704406	1.161368e+00
lambda.l[63]	0.0473105	4.0724429	1.152643e+00
lambda.l[64]	0.0541267	4.1052285	1.162788e+00
lambda.l[65]	0.0570178	3.8350682	1.149015e+00
lambda.l[66]	0.0500898	3.9531481	1.144425e+00
phi.l[1]	0.0001381	0.4103465	1.931195e-01
phi.l[2]	0.0019335	2.6300644	5.846634e-01
phi.l[3]	0.0470746	1273.4405237	1.435267e+03
phi.l[4]	0.0367869	1495.3680483	1.062121e+04

	2.5%	97.5%	avg
phi.l[5]	0.0038941	7.6883265	4.984731e+01
phi.l[6]	0.0282721	566.4235951	1.012865e+04
phi.l[7]	0.0212342	765.5806686	2.541725e+05
phi.l[8]	0.0363941	1701.1246255	1.862788e+06
phi.l[9]	0.0289427	651.9251126	3.055818e+03
phi.l[10]	0.0000218	194.5860072	2.809590e+02
phi.l[11]	0.0046140	8.0265599	4.076660e+01
phi.l[12]	0.0022147	18.4789977	2.444695e+03
phi.l[13]	0.0035392	5.8721953	3.113231e+00
phi.l[14]	0.0117195	186.5831164	3.635489e+03
phi.l[15]	0.0364760	644.5272140	3.803349e+02
phi.l[16]	0.0093039	140.5030970	8.509154e+02
phi.l[17]	0.0140974	212.7067943	5.047895e+02
phi.l[18]	0.0535304	1474.9979819	6.267587e+04
phi.l[19]	0.0318647	873.1017896	8.021387e+03
phi.l[20]	0.0424449	1934.3901089	4.115974e+03
phi.l[21]	0.0021357	4.2027845	1.038074e+00
phi.l[22]	0.0000213	0.1464476	2.490330e-02
phi.l[23]	0.0000285	0.2553721	5.495620e-02
phi.l[24]	0.0000862	0.5281239	6.399010e-02
phi.l[25]	0.0317122	793.1107355	1.272974e+04
phi.l[26]	0.0002526	0.7681272	1.026946e-01
phi.l[27]	0.0002864	0.6170216	1.090302e-01
phi.l[28]	0.0020085	3.4736911	6.634432e-01
phi.l[29]	0.0005309	1.1221278	2.945256e-01
phi.l[30]	0.0005714	1.0962943	1.523273e-01
phi.l[31]	0.0509132	1567.4702930	5.748538e+03
phi.l[32]	0.0437086	1906.8161635	2.594940e+07
phi.l[33]	0.0282398	1009.2114265	8.998192e+03
phi.l[34]	0.0530758	1545.5022542	9.767835e+03
phi.l[35]	0.0063966	1274.0185798	1.754748e+03
phi.l[36]	0.0079204	2992.3252910	2.585482e+03
phi.l[37]	0.0459128	1522.2095664	1.172974e+05
phi.l[38]	0.0160658	312.5298300	1.438986e+04
phi.l[39]	0.0523330	1672.5737607	1.099922e+03
phi.l[40]	0.0175153	718.7931221	1.236629e+04
phi.l[41]	0.0258521	1478.6482319	1.036675e+05
phi.l[42]	0.0166906	1397.6056301	1.719696e+03
phi.l[43]	0.0341116	1673.2034364	5.090324e+03
phi.l[44]	0.0160282	285.3990834	3.581003e+02
phi.l[45]	0.0105659	66.8013562	1.085811e+04
phi.l[46]	0.0125340	912.6103621	3.686919e+03
phi.l[47]	0.0359206	1424.9071140	4.263917e+04
phi.l[48]	0.0176843	495.2081341	4.727162e+04
phi.l[49]	0.0616489	1971.7105622	1.201982e+04
phi.l[50]	0.0543708	1523.9123946	1.476967e+04
phi.l[51]	0.0341761	1706.3530762	2.095626e+03
phi.l[52]	0.0672256	1583.9766130	2.674527e+05
phi.l[53]	0.0632146	2337.5012071	3.044488e+04
phi.l[54]	0.0521276	1821.2970220	4.967040e+04
phi.l[55]	0.0627032	1459.0331835	2.597141e+04
phi.l[56]	0.0584466	1807.7582769	2.116757e+04

	2.5%	97.5%	avg
phi.l[57]	0.0103806	414.6948496	4.747299e+04
phi.l[58]	0.0445868	1827.5956709	2.976761e+04
phi.l[59]	0.0301180	1507.9346464	2.699966e+03
phi.l[60]	0.0326768	1350.6065404	1.540829e+04
phi.l[61]	0.0417732	2217.6793119	9.760032e+03
phi.l[62]	0.0407344	2246.2814382	1.269737e+04
phi.l[63]	0.0328065	1199.4214958	1.967799e+04
phi.l[64]	0.0389523	1119.1234934	5.794659e+03
phi.l[65]	0.0505939	1611.8788315	4.709987e+03
phi.l[66]	0.0413778	1582.7616668	3.417997e+03
sigma	47.9615921	50.6070721	4.923983e+01