

Code for Figure 2

August 6, 2016

Load the necessary libraries, source the file with the R functions:

```
library(ggplot2)

source("functions.R")
```

Will use the same theme throughout, so just declare this variable:

```
themeUsed <- theme_bw(base_size = 20) +
  theme(axis.line = element_line(colour = "black"),
        plot.title = element_text(size = 15, hjust=0.5),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        legend.background = element_rect(fill="transparent"),
        legend.key = element_blank(),
        legend.text.align = 0,
        legend.position = c(0.15,0.28),
        axis.line.x = element_line(color="black", size = 0.5), ##this is to show axes - bug in this version
        axis.line.y = element_line(color="black", size = 0.5))
```

1 Panel a)

Get what is needed for panel a): Load the files representing the summary for each scenario with 0 heterogeneity and save the results in a single dataframe, Vars0:

```
allFiles <- list.files("simResultsComb")
##remove files that have "bd_het" in the title
allFiles <- allFiles[!grep("bd_het", allFiles)]

##save the three Vars for the different combinations
VarsAll <- expand.grid(I = c(5, 10, 15, 20),
                      p = 2:10,
                      corrBtw = c(0,0.5),
                      varBtw = c(0, 1))

##only keep combinations of corrBtw=0 & varBtw=0, corrBtw=0.5 & varBtw=1
VarsAll <- VarsAll[(VarsAll$corrBtw == 0 & VarsAll$varBtw == 0) |
                  (VarsAll$corrBtw == 0.5 & VarsAll$varBtw == 1) ,]
```

```

VarsAll$known <- VarsAll$unknown <- VarsAll$univKnown <- VarsAll$univ <- NA

for(file in 1:length(allFiles))
{
  load(paste("simResultsComb",
             allFiles[file], sep="/"))

  rowNr <- gsub("combine_cost_of_estimation_", "", allFiles[file])
  rowNr <- gsub(".RData", "", rowNr)
  rowNr <- as.numeric(rowNr)

  VarsAll[rowNr, colnames(VarsAll)] <- Vars[rowNr, colnames(VarsAll)]
}
VarsAll$Ratio <- VarsAll$known/VarsAll$unknown
VarsAll$RelEff <- VarsAll$unknown/VarsAll$univ
VarsAll$RelEffT <- VarsAll$known/VarsAll$univKnown
range(VarsAll$Ratio)

## [1] 0.8381268 0.9947634

##only keep the ones with varBtw = 0 and I=20
Vars0 <- VarsAll[VarsAll$varBtw == 0 & VarsAll$I == 20 ,]

```

Load the files representing the summary for each scenario with non-0 heterogeneity and save the results in a single dataframe, VarsAll:

```

allFiles <- list.files("simResultsComb")
##keep only files that have "bd_het" in the title
allFiles <- allFiles[grep("bd_het", allFiles)]

##save the three Vars for the different combinations
VarsAll <- expand.grid(I = 20,
                     p = 2:10,
                     corrBtw = c(0.5),
                     varBtw = c(1/5, 1, 5))

VarsAll$known <- VarsAll$unknown <- VarsAll$univKnown <- VarsAll$univ <- NA

for(file in 1:length(allFiles))
{
  load(paste("simResultsComb", allFiles[file], sep="/"))

  rowNr <- gsub("combine_cost_of_estimation_bd_het_", "", allFiles[file])
  rowNr <- gsub(".RData", "", rowNr)
  rowNr <- as.numeric(rowNr)

  VarsAll[rowNr, colnames(VarsAll)] <- Vars[rowNr, colnames(VarsAll)]
}
VarsAll$Ratio <- VarsAll$known/VarsAll$unknown
VarsAll$RelEff <- VarsAll$unknown/VarsAll$univ
VarsAll$RelEffT <- VarsAll$known/VarsAll$univKnown
range(VarsAll$Ratio)

## [1] 0.9538238 0.9989515

```

Separate out what is needed for Panel a):

```
Vars <- VarsAll[VarsAll$corrBtw == 0.5,]

##add in the 0 heterogeneity case
Vars <- rbind(Vars, Vars0)

RelEff.b <-
  rbind(cbind(as.matrix(Vars[,c("I", "p", "corrBtw", "varBtw", "RelEff"),]),"RelEff"),
        cbind(as.matrix(Vars[,c("I", "p", "corrBtw", "varBtw", "RelEffT"),]),"RelEffT"))
colnames(RelEff.b)[6] <- "Estimate"
RelEff.b <- as.data.frame(RelEff.b)
sapply(RelEff.b, class)

##          I          p corrBtw  varBtw  RelEff Estimate
## "factor" "factor" "factor" "factor" "factor" "factor"

RelEff.b$p <- as.numeric(as.character(RelEff.b$p))
RelEff.b$corrBtw <- as.numeric(as.character(RelEff.b$corrBtw))
RelEff.b$varBtw <- as.numeric(as.character(RelEff.b$varBtw))
RelEff.b$RelEff <- as.numeric(as.character(RelEff.b$RelEff))

panelA <- ggplot(RelEff.b, aes(x=p, y=RelEff))+
  geom_point(size=3.0, aes(color=as.factor(varBtw), shape=as.factor(varBtw)))+
  geom_line(aes(linetype=Estimate,color=as.factor(varBtw), shape=as.factor(varBtw)))+
  themeUsed +
  ylab(expression(RelEff)) +
  ylim(limits=c(0.45, 1)) +
  scale_color_discrete(name = "",
    labels =
      c(expression(paste(sigma^2, "/", S^2, "=",
        0)),
        expression(paste(sigma^2, "/", S^2, phantom() %~~% phantom(),
        1/5)),
        expression(paste(sigma^2, "/", S^2, phantom() %~~% phantom(),
        1)),
        expression(paste(sigma^2, "/", S^2, phantom() %~~% phantom(),
        5)))) +
  scale_shape_discrete(name = "",
    labels =
      c(expression(paste(sigma^2, "/", S^2, "=",
        0)),
        expression(paste(sigma^2, "/", S^2, phantom() %~~% phantom(),
        1/5)),
        expression(paste(sigma^2, "/", S^2, phantom() %~~% phantom(),
        1)),
        expression(paste(sigma^2, "/", S^2, phantom() %~~% phantom(),
        5)))) +
  labs(color="", shape="",
    title=expression(atop("(a)",paste("Random effects: ",
      S[i]^2, phantom() %~~% phantom() , 1, ", ",
      rho[i], phantom() %~~% phantom(), (i-1)/I, ", ",
      rho^BS, " = ", 0.5, ", ",
      ## "\n",
```

```

I, " = ", 20)))) +
scale_linetype_manual(name = "",
  labels =
    c(expression(paste(RelEff)),
      expression(paste(RelEff^T))),
  values=2:1)

```

2 Panels b) and c)

Load the files representing the summary for each scenario and save the results in a single dataframe, VarsAll:

```

allFiles <- list.files("simResultsComb")
##remove files that have "bd_het" in the title
allFiles <- allFiles[-grep("bd_het", allFiles)]

##save the three Vars for the different combinations
VarsAll <- expand.grid(I = c(5, 10, 15, 20),
  p = 2:10,
  corrBtw = c(0, 0.5),
  varBtw = c(0, 1))

##only keep combinations of corrBtw=0 & varBtw=0, corrBtw=0.5 & varBtw=0.5
VarsAll <- VarsAll[(VarsAll$corrBtw == 0 & VarsAll$varBtw == 0) |
  (VarsAll$corrBtw == 0.5 & VarsAll$varBtw == 1) ,]

VarsAll$known <- VarsAll$unknown <- VarsAll$univKnown <- VarsAll$univ <-
  VarsAll$Bayes <- VarsAll$univBayes <- NA

for(file in 1:length(allFiles))
{
  load(paste("simResultsComb",
    allFiles[file], sep="/"))

  rowNr <- gsub("combine_cost_of_estimation_", "", allFiles[file])
  rowNr <- gsub(".RData", "", rowNr)
  rowNr <- as.numeric(rowNr)

  ##only consider files which have the Bayesian results included
  if(length(grep("Bayes", colnames(Vars)))>0)
  {
    VarsAll[rowNr, colnames(VarsAll)] <- Vars[rowNr, colnames(VarsAll)]
  }
}

VarsAll$Ratio <- VarsAll$known/VarsAll$unknown
VarsAll$RelEff <- VarsAll$unknown/VarsAll$univ
VarsAll$RelEffT <- VarsAll$known/VarsAll$univKnown
VarsAll$RelEffB <- VarsAll$Bayes/VarsAll$univBayes
VarsAll$MVMAcomp <- VarsAll$unknown/VarsAll$Bayes
range(VarsAll$Ratio)

## [1] 0.8381268 0.9947634

```

```
VarsAll$I <- as.factor(paste("I=", VarsAll$I, sep=""))
VarsAll$I <- relevel(VarsAll$I, ref="I=5")
```

Get what is needed for panels b) and c):

```
Vars.b <- VarsAll[VarsAll$corrBtw == 0 &
  VarsAll$varBtw == 0,]

##for panel c, have RelEff, RelEffT, RelEffB
##probably easier to just create another object
RelEff.c <-
  rbind(cbind(as.matrix(Vars.b[,c("I", "p", "corrBtw", "varBtw", "RelEff", "MVMAcomp"),]), "RelEff"),
        cbind(as.matrix(Vars.b[,c("I", "p", "corrBtw", "varBtw", "RelEffT", "MVMAcomp"),]), "RelEffT"),
        cbind(as.matrix(Vars.b[,c("I", "p", "corrBtw", "varBtw", "RelEffB", "MVMAcomp"),]), "RelEffB"))
colnames(RelEff.c)[ncol(RelEff.c)] <- "Estimate"
RelEff.c <- as.data.frame(RelEff.c)
sapply(RelEff.c, class)

##      I      p corrBtw varBtw RelEff MVMAcomp
## "factor" "factor" "factor" "factor" "factor" "factor"
## Estimate
## "factor"

RelEff.c$p <- as.numeric(as.character(RelEff.c$p))
RelEff.c$corrBtw <- as.numeric(as.character(RelEff.c$corrBtw))
RelEff.c$varBtw <- as.numeric(as.character(RelEff.c$varBtw))
RelEff.c$RelEff <- as.numeric(as.character(RelEff.c$RelEff))
RelEff.c$MVMAcomp <- as.numeric(as.character(RelEff.c$MVMAcomp))
RelEff.c$I <- relevel(RelEff.c$I, ref="I=5")
```

3 Create plots

Panel b):

```
panelB <- ggplot(RelEff.c, aes(x=p, y=RelEff))+
  geom_point(size=3.0, aes(color=I, shape=I))+
  geom_line(aes(color=I, shape=I, linetype=Estimate))+
  themeUsed+
  ylab(expression(RelEff)) +
  ylim(min(RelEff.c$RelEff)*0.6, 1) +
  labs(color="", shape="",
        title=expression(atop("(b)", paste("Random effects: ",
                                             S[i]^2, phantom() %~~% phantom() , 1, ", ",
                                             rho[i], phantom() %~~% phantom(), (i-1)/I, ", ",
                                             ##"\n",
                                             Sigma, " = ", 0)))) +
  scale_linetype_manual(name = "",
                        labels =
                          c(expression(paste(RelEff)),
                            expression(paste(RelEff^B)),
                            expression(paste(RelEff^T))),
                        values=c(2,3,1))+
```

```
scale_color_manual(values=rev(gg_color_hue(4)), name="")+
scale_shape_manual(values=c(3,15,17,16),name="")
```

Panel c):

```
panelC <- ggplot(RelEff.c, aes(x=p, y=MVMComp))+
  geom_point(size=3.0, aes(color=I, shape=I))+
  geom_line(aes(color=I, shape=I))+
  themeUsed+
  ylab(expression(paste(Var, "(", mu[1]^M, ")", "/",
                        Var, "(", mu[1]^MB, ")))) +
  ylim(min(RelEff.c$MVMComp)*0.75, 1.2) +
  labs(color="", shape="",
        title=expression(atop("(c)", paste("Random effects: ",
                                             S[i]^2, phantom() %~~% phantom() , 1, " ",
                                             rho[i], phantom() %~~% phantom(), (i-1)/I, " ",
                                             ##"\n",
                                             Sigma, " = ", 0)))) +
  scale_color_manual(values=rev(gg_color_hue(4)), name="")+
  scale_shape_manual(values=c(3,15,17,16),name="")
```

```
multiplot(panelA, panelC, panelB, cols=2)
```

```
## Loading required package: grid
```

