Code for new Supplementary Figure - post SIM rejection

August 2, 2016

1 Get the combined datasets for all the scenarios

Load the necessary libraries:

```
library(ggplot2)
library(clusterGeneration)

## Loading required package:  MASS

source("functions.R")
```

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

```
allFiles <- list.files("simResultsComb")

##save the three Vars for the different combinations
VarsAll <- expand.grid(p = c(5,10,15,20),
                       randVar = 1:5,
                       het = c(0.2, 1, 5),
                       sampleType = c("same", "diff"),
                       I = c(10,20))

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

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

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

    VarsAll[rowNr, c("p","randVar","het","sampleType","I",
                     "univ","univKnown",
                     "unknown","known")] <-
      Vars[rowNr, c("p","randVar","het","sampleType","I",
                    "univ","univKnown",
                    "unknown","known")]

    ##get number of studies, number of parameters, scenario number, etc
    p <- VarsAll[rowNr, "p"]
```

```
    randVar <- VarsAll[rowNr, "randVar"]
    het <- VarsAll[rowNr, "het"]
    sampleType <- VarsAll[rowNr, "sampleType"]
    I <- VarsAll[rowNr, "I"]

    ##get theoretical variances of multivariate and univariate estimators
    varMult <- varUniv <- varUnivT1 <- varUnivT2 <- matrix(0, p, p)

    ##simulate Sigma again
    ##set.seed(381048+1940*randVar)
    ##Sigma <- cov2cor(genPositiveDefMat(p)$Sigma)*het

    D <- diag(diag(Sigma))

    for(site in 1:I)
    {
      ##StudyEmpVars[[site]] <- StudyEmpVars[[site]] + Sigma

      varMult <- varMult+solve(StudyEmpVars[[site]]+Sigma)

      Uisite <- diag(diag(StudyEmpVars[[site]]))

      varUnivT1 <- varUnivT1+solve(Uisite+D)
      varUnivT2 <- varUnivT2+
        solve(Uisite+D) %*% (StudyEmpVars[[site]]+Sigma) %*% solve(Uisite+D)
    }

    varMult <- solve(varMult)
    varUnivT1 <- solve(varUnivT1)
    varUniv <- varUnivT1 %*% varUnivT2 %*% varUnivT1

    VarsAll[rowNr, c("knownEmpWithin","univKnownEmpWithin")] <-
      c(varMult[1], varUniv[1])
}
VarsAll$Ratio <- VarsAll$known/VarsAll$unknown
VarsAll$RelEff <- VarsAll$unknown/VarsAll$univ
VarsAll$RelEffT <- VarsAll$known/VarsAll$univKnown
VarsAll$RelEffTAsympt <- VarsAll$knownEmpWithin/VarsAll$univKnownEmpWithin

range(VarsAll$Ratio)

## [1] 0.8521896 0.9982430
```

```
##add a column that specifically gives the value of Sigma^2 (from het)
VarsAll$Sigma2 <- ""
VarsAll$Sigma2[VarsAll$het == 0.2] <- "Sigma^2 = 1/5"
VarsAll$Sigma2[VarsAll$het == 1] <- "Sigma^2 = 1"
VarsAll$Sigma2[VarsAll$het == 5] <- "Sigma^2 = 5"
VarsAll$Sigma2 <- factor(VarsAll$Sigma2,
                         levels= c("Sigma^2 = 1/5", "Sigma^2 = 1", "Sigma^2 = 5"))

VarsAllSame <- VarsAll[VarsAll$sampleType=="same",]
VarsAllDiff <- VarsAll[VarsAll$sampleType=="diff",]
```

```
##get the minimum value for Sigma^2 = 5
min(VarsAll$RelEff[VarsAll$Sigma2 == "Sigma^2 = 5"])

## [1] 0.9404613

##change format so it's ggplot-friendly
RelEffSame <-
  rbind(cbind(as.matrix(VarsAllSame[,c("p", "I", "randVar", "RelEff"),]),"RelEff"),
        cbind(as.matrix(VarsAllSame[,c("p", "I", "randVar", "RelEffT"),]),"RelEffT"),
        cbind(as.matrix(VarsAllSame[,c("p", "I", "randVar", "RelEffTAsympt"),]),"RelEffTAsympt"))
colnames(RelEffSame)[5] <- "Estimate"
RelEffSame <- as.data.frame(RelEffSame)
RelEffSame$Sigma2 <- factor(c(as.character(VarsAllSame$Sigma2), as.character(VarsAllSame$Sigma2),
                              as.character(VarsAllSame$Sigma2)),
                            levels = c("Sigma^2 = 1/5", "Sigma^2 = 1", "Sigma^2 = 5"))

RelEffSame$p <- as.numeric(as.character(RelEffSame$p))
RelEffSame$RelEff <- as.numeric(as.character(RelEffSame$RelEff))

##

RelEffDiff <-
  rbind(cbind(as.matrix(VarsAllDiff[,c("p", "I", "randVar", "RelEff"),]),"RelEff"),
        cbind(as.matrix(VarsAllDiff[,c("p", "I", "randVar", "RelEffT"),]),"RelEffT"),
        cbind(as.matrix(VarsAllDiff[,c("p", "I", "randVar", "RelEffTAsympt"),]),"RelEffTAsympt"))
colnames(RelEffDiff)[5] <- "Estimate"
RelEffDiff <- as.data.frame(RelEffDiff)
RelEffDiff$Sigma2 <- factor(c(as.character(VarsAllDiff$Sigma2), as.character(VarsAllDiff$Sigma2),
                              as.character(VarsAllDiff$Sigma2)),
                            levels = c("Sigma^2 = 1/5", "Sigma^2 = 1", "Sigma^2 = 5"))

RelEffDiff$p <- as.numeric(as.character(RelEffDiff$p))
RelEffDiff$RelEff <- as.numeric(as.character(RelEffDiff$RelEff))

##take out the asymptotic relative efficiency
RelEffSame <- RelEffSame[RelEffSame$Estimate != "RelEffTAsympt",]
RelEffDiff <- RelEffDiff[RelEffDiff$Estimate != "RelEffTAsympt",]
```

2 Make Figure S4

```
I_10_Same_Panel_A <- panelFigS4S5(I=10, subsetSigma2 = "Sigma^2 = 1/5", RelEffSubs = RelEffSame) +
  labs(color="", shape="",
       title=expression(atop("(a)", paste("Random effects: ",
                                          I, " = ", 10, ", ",
                                          sigma^2/bar(S^2),  phantom() %~~% phantom() , 1/5, ", ",
                                          "Equal sample sizes"))))
```

```
I_10_Same_Panel_B <- panelFigS4S5(I=10, subsetSigma2 = "Sigma^2 = 1", RelEffSubs = RelEffSame) +
  labs(color="", shape="",
       title=expression(atop("(b)", paste("Random effects: ",
```

```
                                        I, " = ", 10, ", ",
                                        sigma^2/bar(S^2),  phantom() %~~% phantom() , 1, ", ",
                                        "Equal sample sizes"))))
```

```
I_10_Same_Panel_C <- panelFigS4S5(I=10, subsetSigma2 = "Sigma^2 = 5", RelEffSubs = RelEffSame) +
  labs(color="", shape="",
       title=expression(atop("(c)", paste("Random effects: ",
                                        I, " = ", 10, ", ",
                                        sigma^2/bar(S^2),  phantom() %~~% phantom() , 5, ", ",
                                        "Equal sample sizes"))))
```

```
I_10_Diff_Panel_A <- panelFigS4S5(I=10, subsetSigma2 = "Sigma^2 = 1/5", RelEffSubs = RelEffDiff) +
  labs(color="", shape="",
       title=expression(atop("(d)", paste("Random effects: ",
                                        I, " = ", 10, ", ",
                                        sigma^2/bar(S^2),  phantom() %~~% phantom() , 1/5, ", ",
                                        "Unequal sample sizes"))))
```

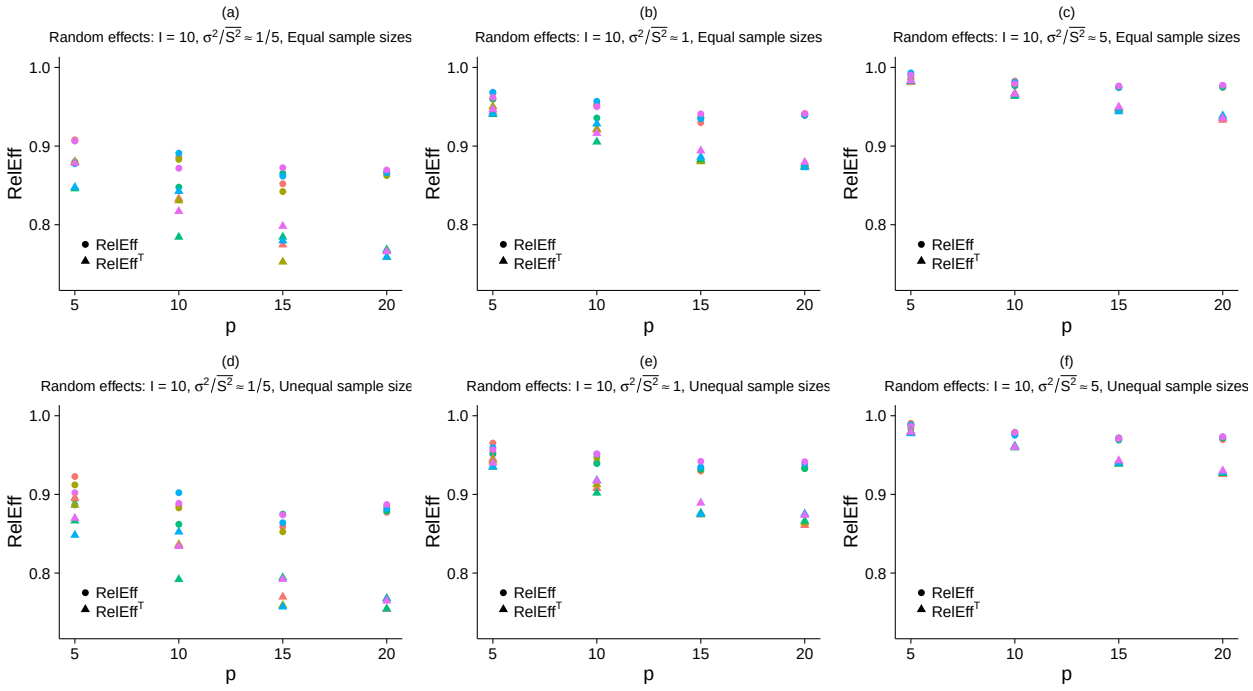
```
I_10_Diff_Panel_B <- panelFigS4S5(I=10, subsetSigma2 = "Sigma^2 = 1", RelEffSubs = RelEffDiff) +
  labs(color="", shape="",
       title=expression(atop("(e)", paste("Random effects: ",
                                        I, " = ", 10, ", ",
                                        sigma^2/bar(S^2),  phantom() %~~% phantom() , 1, ", ",
                                        "Unequal sample sizes"))))
```

```
I_10_Diff_Panel_C <- panelFigS4S5(I=10, subsetSigma2 = "Sigma^2 = 5", RelEffSubs = RelEffDiff) +
  labs(color="", shape="",
       title=expression(atop("(f)", paste("Random effects: ",
                                        I, " = ", 10, ", ",
                                        sigma^2/bar(S^2),  phantom() %~~% phantom() , 5, ", ",
                                        "Unequal sample sizes"))))
```

Put all 6 panels together for Figure S4:

```
multiplot(I_10_Same_Panel_A, I_10_Diff_Panel_A,
          I_10_Same_Panel_B, I_10_Diff_Panel_B,
          I_10_Same_Panel_C, I_10_Diff_Panel_C,
          cols=3)

## Loading required package:  grid
```



3 Make Figure S5

```
I_20_Same_Panel_A <- panelFigS4S5(I=20, subsetSigma2 = "Sigma^2 = 1/5", RelEffSubs = RelEffSame) +
  labs(color="", shape="",
       title=expression(atop("(a)", paste("Random effects: ",
                                          I, " = ", 20, ", ",
                                          sigma^2/bar(S^2),  phantom() %~~% phantom() , 1/5, ", ",
                                          "Equal sample sizes"))))
```

```
I_20_Same_Panel_B <- panelFigS4S5(I=20, subsetSigma2 = "Sigma^2 = 1", RelEffSubs = RelEffSame) +
  labs(color="", shape="",
       title=expression(atop("(b)", paste("Random effects: ",
                                          I, " = ", 20, ", ",
                                          sigma^2/bar(S^2),  phantom() %~~% phantom() , 1, ", ",
                                          "Equal sample sizes"))))
```

```
I_20_Same_Panel_C <- panelFigS4S5(I=20, subsetSigma2 = "Sigma^2 = 5", RelEffSubs = RelEffSame) +
  labs(color="", shape="",
       title=expression(atop("(c)", paste("Random effects: ",
                                          I, " = ", 20, ", ",
                                          sigma^2/bar(S^2),  phantom() %~~% phantom() , 5, ", ",
                                          "Equal sample sizes"))))
```

```
I_20_Diff_Panel_A <- panelFigS4S5(I=20, subsetSigma2 = "Sigma^2 = 1/5", RelEffSubs = RelEffDiff) +
  labs(color="", shape="",
       title=expression(atop("(d)", paste("Random effects: ",
```

```
                                        I, " = ", 20, ", ",
                                        sigma^2/bar(S^2),  phantom() %~~% phantom() , 1/5, ", ",
                                        "Unequal sample sizes"))))
```

```
I_20_Diff_Panel_B <- panelFigS4S5(I=20, subsetSigma2 = "Sigma^2 = 1", RelEffSubs = RelEffDiff) +
  labs(color="", shape="",
       title=expression(atop("(e)", paste("Random effects: ",
                                        I, " = ", 20, ", ",
                                        sigma^2/bar(S^2),  phantom() %~~% phantom() , 1, ", ",
                                        "Unequal sample sizes"))))
```

```
I_20_Diff_Panel_C <- panelFigS4S5(I=20, subsetSigma2 = "Sigma^2 = 5", RelEffSubs = RelEffDiff) +
  labs(color="", shape="",
       title=expression(atop("(f)", paste("Random effects: ",
                                        I, " = ", 20, ", ",
                                        sigma^2/bar(S^2),  phantom() %~~% phantom() , 5, ", ",
                                        "Unequal sample sizes"))))
```

Put all 6 panels together for Figure 5:

```
multiplot(I_20_Same_Panel_A, I_20_Diff_Panel_A,
          I_20_Same_Panel_B, I_20_Diff_Panel_B,
          I_20_Same_Panel_C, I_20_Diff_Panel_C,
          cols=3)
```