Code for Figure 2

August 2, 2016

1 Create datasets

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

```
library(ggplot2)
library(Matrix)

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.key = element_blank(),
        legend.text.align = 0,
        legend.position = c(0.15,0.2),
        legend.background = element_rect(fill="transparent"),
        axis.line.x = element_line(color="black", size = 0.5), ##this is to show axes - bug in this vers
        axis.line.y = element_line(color="black", size = 0.5))
```

2 AR(1) correlation matrices

Autoregressive case... for both within-study and between-study matrices...

```
##number of studies
I <- 20

##save results in data frame
##het is sigma^2/S^2
bigMat <- expand.grid(r = 1/2,
                      p = 2:20,
                      rhoBS=0.5,
                      het = c(0, 1/5, 1, 5),
                      RelEff = NA)
for(rr in 1:nrow(bigMat))
```

```
{
    r <- bigMat[rr, 1]
    p <- bigMat[rr, 2]
    rhoBS <- bigMat[rr, 3]
    het <- bigMat[rr, 4]

    ##index over the studies
    i <- 1:I
    ##within-study correlations
    rho.iW <- (i-1)/I
    ##get the *total* variance matrices (within-study PLUS between-study)
    Si <-
      lapply(i,
             function(i, p, I, het, rhoBS){ARMAcor(phi=1,
                                                   rho=(i-1)/I,
                                                   n=p) +
                 het * ARMAcor(phi=1,
                               rho=rhoBS,
                               n=p) },
             p, I, het, rhoBS)

    ##calculate the variances
    VarMVMA <- VarUVMA2 <-
      matrix(0, p, p)

    for(i in 1:I)
    {
      VarMVMA <- VarMVMA + solve(Si[[i]])
      VarUVMA2 <- VarUVMA2 + Si[[i]]
    }
    VarMVMA <- solve(VarMVMA)
    VarUVMA <- VarUVMA2/I^2

    bigMat$RelEff[rr] <- VarMVMA[1,1]/VarUVMA[1,1]
}

##make het a factor (for ggplot)
bigMat$het <- as.factor(bigMat$het)

levels(bigMat$het)

## [1] "0"   "0.2" "1"   "5"
```

3 Block diagonal correlation matrices

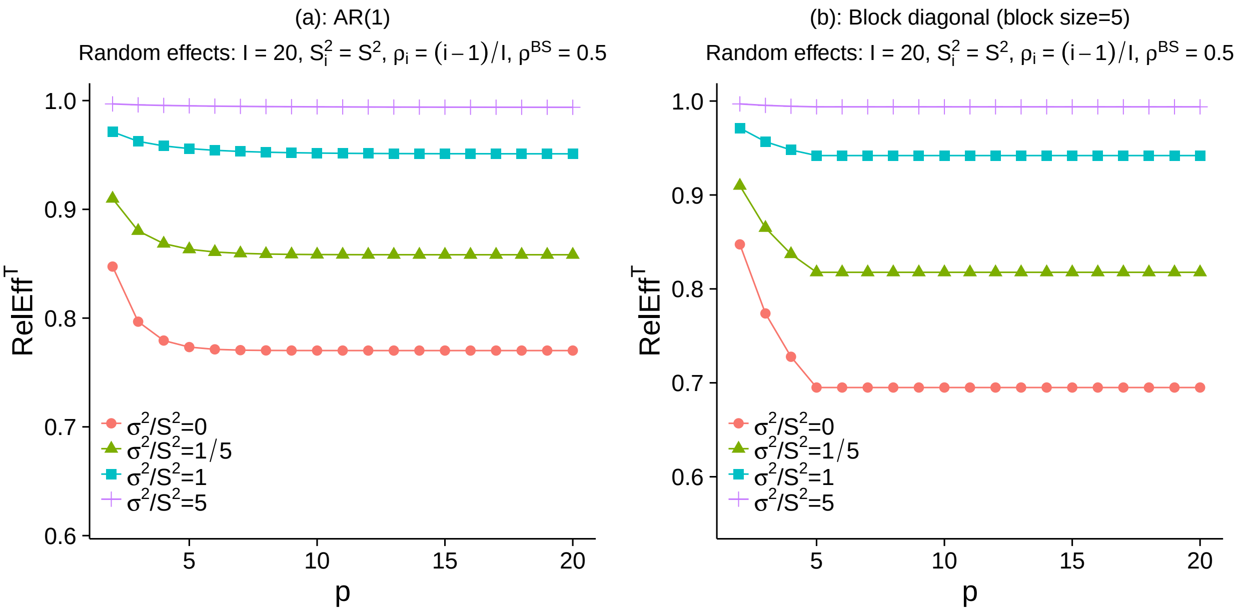
Block diagonal case... for both within-study and between-study matrices...

```
##number of studies
I <- 20

##save results in data frame
##het is sigma^2/S^2
```

```
bigMat <- expand.grid(r = 1/2,
                      p = 2:20,
                      rhoBS=0.5,
                      het = c(0, 1/5, 1, 5),
                      RelEff = NA)
for(rr in 1:nrow(bigMat))
{
    r <- bigMat[rr, 1]
    p <- bigMat[rr, 2]
    rhoBS <- bigMat[rr, 3]
    het <- bigMat[rr, 4]

    nrBlocks <- round(p/5)

    ##have one "incomplete block" if p is not divisible by 5
    if(p %% 5 != 0)
    {
      nrBlocks <- nrBlocks + 1
    }

    ##index over the studies
    i <- 1:I

    ##get all the within-study variance matrices
    Si <-
      lapply(i,
             function(i, p, I, nrBlocks, het, rhoBS){S <- blockDiag((i-1)/I, sizeBlock=5, nrBlocks=nrBlo
               het * blockDiag(rhoBS, sizeBlock=5, nrBlocks=nrBlocks);
             S[1:p, 1:p]},
             p, I, nrBlocks, het, rhoBS)

    ##calculate the variances
    VarMVMA <- VarUVMA2 <-
      matrix(0, p, p)

    for(i in 1:I)
    {
      VarMVMA <- VarMVMA + solve(Si[[i]])
      VarUVMA2 <- VarUVMA2 + Si[[i]]
    }
    VarMVMA <- solve(VarMVMA)
    VarUVMA <- VarUVMA2/I^2

    bigMat$RelEff[rr] <- VarMVMA[1,1]/VarUVMA[1,1]
}

##make het a factor (for ggplot)
bigMat$het <- as.factor(bigMat$het)

levels(bigMat$het)

## [1] "0"   "0.2" "1"   "5"
```

4 Put both panels together

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

## Loading required package:  grid
```



```
panelA <- ggplot(bigMat,
                 aes(x=p, y=RelEff))+
  geom_line(aes(color=het, shape=het)) +
  geom_point(size=3.0, aes(color=het, shape=het)) +
  themeUsed+
  scale_y_continuous(name = expression(paste(RelEff^T)),
                     limits=c(min(bigMat$RelEff)*0.8, max(bigMat$RelEff))) +
  scale_color_discrete(name = "",
                       labels =
                         c(expression(paste(sigma^2, "/", S^2, "=",
                                            0)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            1/5)),
                           ##expression(paste(sigma^2, "/", S^2, "=",
                           ##                 1/2)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            1)),
                           ##expression(paste(sigma^2, "/", S^2, "=",
                           ##                 2)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            5)))) +
  scale_shape_discrete(name = "",
                       labels =
                         c(expression(paste(sigma^2, "/", S^2, "=",
                                            0)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            1/5)),
                           ##expression(paste(sigma^2, "/", S^2, "=",
                           ##                1/2)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            1)),
                           ##expression(paste(sigma^2, "/", S^2, "=",
                           ##                 2)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            5)))) +
  labs(title=expression(atop("(a): AR(1)", paste("Random effects: I = 20, ",
                                          S[i]^2, " = ", S^2, ", ",
                                          rho[i], " = ", (i-1)/I, ", ",
                                          ##"\n",
                                          rho^BS, " = ", 0.5))))
```

```
panelB <- ggplot(bigMat,
                 aes(x=p, y=RelEff))+
  geom_line(aes(color=het, shape=het)) +
  geom_point(size=3.0, aes(color=het, shape=het)) +
  themeUsed+
  scale_y_continuous(name = expression(paste(RelEff^T)),
                     limits=c(min(bigMat$RelEff)*0.8, max(bigMat$RelEff))) +
  scale_color_discrete(name = "",
                       labels =
                         c(expression(paste(sigma^2, "/", S^2, "=",
                                            0)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            1/5)),
                           ##expression(paste(sigma^2, "/", S^2, "=",
                           ##                 1/2)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            1)),
                           ##expression(paste(sigma^2, "/", S^2, "=",
                           ##                 2)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            5)))) +
  scale_shape_discrete(name = "",
                       labels =
                         c(expression(paste(sigma^2, "/", S^2, "=",
                                            0)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            1/5)),
                           ##expression(paste(sigma^2, "/", S^2, "=",
                           ##                1/2)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            1)),
                           ##expression(paste(sigma^2, "/", S^2, "=",
                           ##                 2)),
                           expression(paste(sigma^2, "/", S^2, "=",
                                            5)))) +
  labs(title=expression(atop("(b): Block diagonal (block size=5)",
                             paste("Random effects: I = 20, ",
                                          S[i]^2, " = ", S^2, ", ",
                                          rho[i], " = ", (i-1)/I, ", ",
                                          ##"\n",
                                          rho^BS, " = ", 0.5))))
```