Figure S3: Fixed effects framework with AR(1), block diagonal structures

August 28, 2016

```
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 = 16, 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,
        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))
```

1 AR(1) correlation matrices

These results are for the $I = 2, p \geq 2$ autoregressive case with AR(1) with equal within-study variances, so the parameters we vary are: r, ρ_1, ρ_2, p. We save the relative efficiencies for only one of coefficients, as they are all equal. We consider $\rho_1 = 0$.

```
##save results in data frame
bigMat <- expand.grid(rho1 = 0,
                      rho2 = (0:3)/4,
                      r = c(1, 3, 5, 9)/10,
                      p = 2:20,
                      RelEff = NA)
for(i in 1:nrow(bigMat))
{
    rho1 <- bigMat[i, 1]
    rho2 <- bigMat[i, 2]
    r <- bigMat[i, 3]
    p <- bigMat[i, 4]
```

```r
    ##get variance-covariance matrices
    S1 <- r*ARMAcor(phi=1, rho=rho1, n=p)
    S2 <- (1-r)*ARMAcor(phi=1, rho=rho2, n=p)
    U1 <- diag(diag(S1))
    U2 <- diag(diag(S2))

    varMVMA <- solve(solve(S1)+solve(S2))
    varUVMA <- solve(solve(U1)+solve(U2)) %*%
      (solve(U1) %*% S1 %*% solve(U1) +
         solve(U2) %*% S2 %*% solve(U2)) %*%
      solve(solve(U1)+solve(U2))

    bigMat$RelEff[i] <- varMVMA[1,1]/varUVMA[1,1]
}

##for Panel b), r = 0.5:
bigMat.b <- bigMat[bigMat[,"r"]==0.5, ]
##transform back to data frame (for ggplot)
bigMat.b <- as.data.frame(bigMat.b)
##make rho2 into factor (for ggplot)
bigMat.b$rho2 <- as.factor(bigMat.b$rho2)
```

```r
panelA <- ggplot(bigMat.b,
                 aes(x=p, y=RelEff))+
  geom_line(aes(color=rho2, shape=rho2)) +
  geom_point(size=3.0, aes(color=rho2, shape=rho2)) +
  themeUsed+
  scale_color_discrete(name = "",
                       labels =
                         c(expression(paste(rho[2], "=",
                                            0)),
                           expression(paste(rho[2], "=",
                                            0.25)),
                           expression(paste(rho[2], "=",
                                            0.5)),
                           expression(paste(rho[2], "=",
                                            0.75)))) +
  scale_shape_discrete(name = "",
                       labels =
                         c(expression(paste(rho[2], "=",
                                            0)),
                           expression(paste(rho[2], "=",
                                            0.25)),
                           expression(paste(rho[2], "=",
                                            0.5)),
                           expression(paste(rho[2], "=",
                                            0.75)))) +
  labs(title=
         expression(atop("(a): AR(1)",
                         paste("Fixed effects: I = 2, ",
                               rho[1], " = ", 0, ", ",
                               r, " = ", 0.5))))
```

The following results are for the $I = 20, p \geq 2$ AR(1) case with equal within-study variances and $S_i^2 \equiv S^2, \rho_i = \frac{\rho(i-1)}{I}$, so the parameters we vary are ρ, p. We save the relative efficiencies for only one of coefficients, as they are all equal.

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

##save results in data frame
bigMat <- expand.grid(rho = c(0, 0.25, 0.5, 0.75, 1),
                      p = 2:20,
                      RelEff = NA)

for(n in 1:nrow(bigMat))
{

    rho <- bigMat[n, 1]
    p <- bigMat[n, 2]

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

    ##get all the within-study variance matrices
    Si <-
      lapply(i,
             function(i, rho, p, I){ARMAcor(phi=1,
                                            rho=rho*(i-1)/I,
                                            n=p) },
             rho, p, I)
    ##get inverses of Si
    ##use: http://mathoverflow.net/questions/65795/inverse-of-an-ar1-or-laplacian-or-kac-murdock-szeg%C3
    ##Si.inv <- lapply(i,
    ##                 function(i, rho, p, I){
    ##                   rho.i <- rho*(i-1)/I;
    ##                   Sinv <- matrix(0, p, p);
    ##                   Sinv
    ##                 }

    ##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[n] <- VarMVMA[1,1]/VarUVMA[1,1]
}

##make rho into factor (for ggplot)
bigMat$rho <- paste("rho=", bigMat$rho, sep="")
```

```
panelB <- ggplot(bigMat,
                 aes(x=p, y=RelEff))+
  geom_line(aes(color=rho, shape=rho)) +
  geom_point(size=3.0, aes(color=rho, shape=rho)) +
  scale_y_continuous(limits = c(min(bigMat$RelEff), 1)) +
  scale_color_discrete(name = "",
                       labels =
                         c(expression(paste(rho, "=",
                                            0)),
                           expression(paste(rho, "=",
                                            0.25)),
                           expression(paste(rho, "=",
                                            0.5)),
                           expression(paste(rho, "=",
                                            0.75)),
                           expression(paste(rho, "=",
                                            1)))) +
  scale_shape_discrete(name = "",
                       labels =
                         c(expression(paste(rho, "=",
                                            0)),
                           expression(paste(rho, "=",
                                            0.25)),
                           expression(paste(rho, "=",
                                            0.5)),
                           expression(paste(rho, "=",
                                            0.75)),
                           expression(paste(rho, "=",
                                            1)))) +
  themeUsed+
  labs(title=
         expression(atop("(b): AR(1)",
                         paste("Fixed effects: I = 20, ",
                               S[list(kk,i)], " = ", 1, ", ",
                               rho[i], " = ", rho(i-1)/I))))
```

2 Block diagonal correlation matrices

These results are for the $I = 2, p \geq 2$ case with block diagonal matrices with block size of 5, compound symmetry within the blocks, so the parameters we vary are: r, ρ_1, ρ_2, p. We save the relative efficiencies for only one of coefficients, as they are all equal. We consider $\rho_1 = 0$.

```
##save results in data frame
bigMat <- expand.grid(rho1 = 0,
                      rho2 = (0:3)/4,
                      r = c(1, 3, 5, 9)/10,
                      p = 2:20,
                      RelEff = NA)
for(i in 1:nrow(bigMat))
{
    rho1 <- bigMat[i, 1]
    rho2 <- bigMat[i, 2]
```

```r
    r <- bigMat[i, 3]
    p <- bigMat[i, 4]
    nrBlocks <- round(p/5)

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

    ##get variance-covariance matrices
    S1 <- r*blockDiag(rho1, sizeBlock=5, nrBlocks=nrBlocks)
    S2 <- (1-r)*blockDiag(rho2, sizeBlock=5, nrBlocks=nrBlocks)
    ##reduce the size to p x p
    S1 <- S1[1:p, 1:p]
    S2 <- S2[1:p, 1:p]

    U1 <- diag(diag(S1))
    U2 <- diag(diag(S2))

    varMVMA <- solve(solve(S1)+solve(S2))
    varUVMA <- solve(solve(U1)+solve(U2)) %*%
      (solve(U1) %*% S1 %*% solve(U1) +
         solve(U2) %*% S2 %*% solve(U2)) %*%
      solve(solve(U1)+solve(U2))

    bigMat$RelEff[i] <- varMVMA[1,1]/varUVMA[1,1]
}

##for Panel b), r = 0.5:
bigMat.b <- bigMat[bigMat[,"r"]==0.5, ]
##transform back to data frame (for ggplot)
bigMat.b <- as.data.frame(bigMat.b)
##make rho2 into factor (for ggplot)
bigMat.b$rho2 <- as.factor(bigMat.b$rho2)
```

```r
panelC <- ggplot(bigMat.b,
                 aes(x=p, y=RelEff))+
  geom_line(aes(color=rho2, shape=rho2)) +
  geom_point(size=3.0, aes(color=rho2, shape=rho2)) +
  themeUsed+
  scale_color_discrete(name = "",
                       labels =
                         c(expression(paste(rho[2], "=",
                                            0)),
                           expression(paste(rho[2], "=",
                                            0.25)),
                           expression(paste(rho[2], "=",
                                            0.5)),
                           expression(paste(rho[2], "=",
                                            0.75)))) +
  scale_shape_discrete(name = "",
                       labels =
```

```
                             c(expression(paste(rho[2], "=",
                                                0)),
                               expression(paste(rho[2], "=",
                                                0.25)),
                               expression(paste(rho[2], "=",
                                                0.5)),
                               expression(paste(rho[2], "=",
                                                0.75)))) +
  labs(title=
         expression(atop("(c): Block diagonal (block size=5)",
                         paste("Fixed effects: I = 2, ",
                               rho[1], " = ", 0, ", ",
                               r, " = ", 0.5))))
```

The following results are for the $I = 20, p \geq 2$ case with block diagonal matrices with block size of 5, compound symmetry within the blocks, with equal within-study variances and $S_i^2 \equiv S^2, \rho_i = \frac{\rho(i-1)}{I}$, so the parameters we vary are ρ, p. We save the relative efficiencies for only one of coefficients, as they are all equal.

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

##save results in data frame
bigMat <- expand.grid(rho = c(0, 0.25, 0.5, 0.75, 1),
                      p = 2:20,
                      RelEff = NA)

for(n in 1:nrow(bigMat))
{
    rho <- bigMat[n, 1]
    p <- bigMat[n, 2]
    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, rho, p, I, nrBlocks){S <- blockDiag(rho*(i-1)/I, sizeBlock=5, nrBlocks=nrBlocks
             S[1:p, 1:p]},
             rho, p, I, nrBlocks)

    ##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[n] <- VarMVMA[1,1]/VarUVMA[1,1]
}

##make rho into factor (for ggplot)
bigMat$rho <- paste("rho=", bigMat$rho, sep="")
```

```
panelD <- ggplot(bigMat,
                 aes(x=p, y=RelEff))+
  geom_line(aes(color=rho, shape=rho)) +
  geom_point(size=3.0, aes(color=rho, shape=rho)) +
  themeUsed+
  scale_y_continuous(limits = c(min(bigMat$RelEff), 1)) +
  scale_color_discrete(name = "",
                       labels =
                         c(expression(paste(rho, "=",
                                            0)),
                           expression(paste(rho, "=",
                                            0.25)),
                           expression(paste(rho, "=",
                                            0.5)),
                           expression(paste(rho, "=",
                                            0.75)),
                           expression(paste(rho, "=",
                                            1)))) +
  scale_shape_discrete(name = "",
                       labels =
                         c(expression(paste(rho, "=",
                                            0)),
                           expression(paste(rho, "=",
                                            0.25)),
                           expression(paste(rho, "=",
                                            0.5)),
                           expression(paste(rho, "=",
                                            0.75)),
                           expression(paste(rho, "=",
                                            1)))) +
  labs(title=
         expression(atop("(d): Block diagonal (block size=5)",
                         paste("Fixed effects: I = 20, ",
                               S[list(kk,i)], " = ", 1, ", ",
                               rho[i], " = ", rho(i-1)/I))))
```

3 Put all four panels together

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

## Loading required package:  grid
```