```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.66 (6956) x86_64-apple-darwin13.4.0]

>
> rm(list=ls(all=TRUE))
>
> set.seed(123)
>
> library(systemfit)
Loading required package: Matrix
Loading required package: car
Loading required package: lmtest
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

> library(plyr)
> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:plyr’:

    arrange, count, desc, failwith, id, mutate, rename, summarise, summarize

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

> library(RColorBrewer)
> library(miscTools)
>
> setwd("~/Dropbox/beliefs_incomplete data/Paper/PSRM/final/replication_archive/SettlerMortality")
>
>
> ### Load the data
> load("data_settlermortality.rda")
>
>
> ### Run original model
> m <- systemfit(logpgp95 ~ avexpr +  laborer, inst = ~logem4 +  laborer, data = data, method = "2SLS")
> summary(m)

systemfit results
method: 2SLS

         N DF     SSR  detRCov   OLS-R2 McElroy-R2
system 64 61 56.1517 0.920519 0.181244   0.181244

     N DF     SSR      MSE     RMSE       R2 Adj R2
eq1 64 61 56.1517 0.920519 0.959437 0.181244 0.1544

The covariance matrix of the residuals
         eq1
eq1 0.920519

The correlations of the residuals
    eq1
eq1   1


2SLS estimates for 'eq1' (equation 1)
Model Formula: logpgp95 ~ avexpr + laborer
Instruments: ~logem4 + laborer

            Estimate Std. Error t value   Pr(>|t|)
(Intercept) 1.881683   1.115009 1.68759   0.096597 .
avexpr      0.947867   0.167837 5.64753 4.5305e-07 ***
laborer     0.073707   0.547370 0.13466   0.893327
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.959437 on 61 degrees of freedom
Number of observations: 64 Degrees of Freedom: 61
SSR: 56.151663 MSE: 0.920519 Root MSE: 0.959437
Multiple R-Squared: 0.181244 Adjusted R-Squared: 0.1544

>
>
> ### Run the sensitivity analysis for campaign/no campaign
> etavec = seq(1, 21, length.out=1000)
>
> results <-  NULL
>
> for(i in 1:length(etavec)){
+     # create data given eta
+     data$logem4_true <- NA
+     data$logem4_true[data$campaign==0] <- data$logem4[data$campaign==0]
+     data$logem4_true[data$campaign==1] <- data$logem4[data$campaign==1] - log(etavec[i])
+         # b/c log(x)-log(eta) is equivalent to x/eta
+
+     # run the model with the simulated data: first stage
+     s1 <- lm(avexpr ~ logem4_true + laborer, data = data)
```

```
+
+     # run the model with the simulated data: IV
+     m <- systemfit(logpgp95 ~ avexpr +  laborer, inst = ~ logem4_true +  laborer, data = data, method = "2SLS")
+
+
+     # save the coefficients and standard errors of both stages
+     est1 <- coef(s1)[2]
+     err1 <- sqrt(vcov(s1)[2,2])
+
+     est2 <- coef(m)[2]
+     err2 <- sqrt(vcov(m)[2,2])
+
+     results <- rbind(results, c(etavec[i], est1, err1, est2, err2))
+ }
>
> colnames(results) <- c("eta", "pointest_first", "stderr_first", "pointest_second", "stderr_second")
>
> results <- as.data.frame(results)
>
> results$lowerci95_first <- results$pointest_first + qnorm(0.025)*results$stderr_first
> results$upperci95_first <- results$pointest_first + qnorm(0.975)*results$stderr_first
>
> results$lowerci95_second <- results$pointest_second + qnorm(0.025)*results$stderr_second
> results$upperci95_second <- results$pointest_second + qnorm(0.975)*results$stderr_second
>
>
>
>
> ### FIGURE 3
>
> ### Panel (a): first stage
> quartz(type="pdf", width=5, height=5, file="output/Soldiers_stage1.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest_first), results$pointest_first, type="n", ylim=c(min(results$lowerci95_first), max(results$upperci95_first)), xlab = expression(eta), ylab="First Stage
Coefficient", xaxt="n")
> polygon(c(1:length(results$pointest_first), rev(1:length(results$pointest_first))), c(results$lowerci95_first, rev(results$upperci95_first)), col="grey", border=NA)
> points(1:length(results$pointest_first), results$pointest_first, type="l", lwd=3)
>
> axis(1, at=seq(1, 1001, length.out=11), labels = seq(1, 21, length.out=11), las=2)
> abline(h=0, col = "black")
> lines(c(1,1), c(results$lowerci95_first[1], results$upperci95_first[1]), lwd=3)
> points(1, results$pointest_first[1], pch=16)
> dev.off()
null device
          1
>
>
>
> ### Panel (b): second stage
> quartz(type="pdf", width=5, height=5, file="output/Soldiers_stage2.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest_second), results$pointest_second, type="n", ylim=c(min(results$lowerci95_second[1:501]), max(results$upperci95_second[1:501])), xlab = expression(eta),
ylab="Second Stage Coefficient", xaxt="n")
> polygon(c(1:length(results$pointest_second), rev(1:length(results$pointest_second))), c(results$lowerci95_second, rev(results$upperci95_second)), col="grey", border=NA)
> points(1:length(results$pointest_second), results$pointest_second, type="l", lwd=3)
>
> axis(1, at=seq(1, 1001, length.out=11), labels = seq(1, 21, length.out=11), las=2)
> abline(h=0, col = "black")
> lines(c(1,1), c(results$lowerci95_second[1], results$upperci95_second[1]), lwd=3)
> points(1, results$pointest_second[1], pch=16)
> dev.off()
null device
          1
>
>
>
>
>
> ### FIGURE 2
>
> data0 <- data$logem4 - data$campaign * log(3)
> data1 <- data$logem4 - data$campaign * log(6)
> data2 <- data$logem4 - data$campaign * log(11)
> data3 <- data$logem4 - data$campaign * log(16)
> data4 <- data$logem4 - data$campaign * log(21)
>
> ## Panel (a): Observed vs Simulated Data
> quartz(type="pdf", width=5, height=5, file="output/Soldiers_Data1.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(data$logem4[data$campaign==1], data$logem4[data$campaign==1], pch=16, xlim=c(min(data$logem4), max(data$logem4)), ylim=c(min(data4), max(data$logem4)), col="grey", xlab="Log
Mortality", ylab="Simulated Log Mortality")
> points(data$logem4[data$campaign==0], data$logem4[data$campaign==0], pch=16)
> points(data$logem4[data$campaign==1], data0[data$campaign==1], pch=16, col="orange")
> points(data$logem4[data$campaign==1], data1[data$campaign==1], pch=16, col="red")
> points(data$logem4[data$campaign==1], data2[data$campaign==1], pch=16, col="green")
> points(data$logem4[data$campaign==1], data3[data$campaign==1], pch=16, col="darkgreen")
> points(data$logem4[data$campaign==1], data4[data$campaign==1], pch=16, col="blue")
> text(3, 2.1, "3", col="orange", cex=1.2)
> text(3, 1.4, "6", col="red", cex=1.2)
> text(3, 0.85, "11", col="green", cex=1.2)
> text(3, 0.5, "16", col="darkgreen", cex=1.2)
> text(3, 0.18, "21", col="blue", cex=1.2)
> dev.off()
null device
          1
>
>
> ## Panel (b): Densities
> quartz(type="pdf", width=5, height=5, file="output/Soldiers_Data2.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(density(data$logem4), xlim=c(-2,9), lwd=3, xlab="Log Mortality", main="")
> points(density(data0), type="l", lwd=3, col="orange")
> points(density(data1), type="l", lwd=3, col="red")
> points(density(data2), type="l", lwd=3, col="green")
> points(density(data3), type="l", lwd=3, col="darkgreen")
> points(density(data4), type="l", lwd=3, col="blue")
> text(3.15, 0.415, "3", col="orange", cex=1.2)
> text(2.6, 0.43, "6", col="red", cex=1.2)
> text(1.95, 0.35, "11", col="green", cex=1.2)
> text(1.52, 0.31, "16", col="darkgreen", cex=1.2)
> text(1.1, 0.285, "21", col="blue", cex=1.2)
> text(4.8, 0.435, "1", col="black", cex=1.2)
> dev.off()
null device
          1
>
>
>
>
```

```
>
> ### EXTENSIONS
>
>
> ## EXT 1: only p percent of C=1 are affected
>
> etavec <- seq(1, 21, length.out=100)
> pvec <- seq(0, 1, length.out=101)
>
> results <-  NULL
>
> for(i in 1:length(etavec)){
+     for(j in 1:length(pvec)){
+         res <- NULL
+         for(k in 1:100){
+             # create data given eta and p
+             data$select <- 0
+             data$select[data$campaign==1] <- rbinom(length(data$select[data$campaign==1]), 1, prob=pvec[j])
+
+             data$logem4_true <- NA
+             data$logem4_true[data$campaign==0] <- data$logem4[data$campaign==0]
+             data$logem4_true[data$campaign==1 & data$select==0] <- data$logem4[data$campaign==1 & data$select==0]
+             data$logem4_true[data$campaign==1 & data$select==1] <- data$logem4[data$campaign==1 & data$select==1] - log(etavec[i])
+
+             # run the model with the simulated data: first stage
+             s1 <- lm(avexpr ~ logem4_true + laborer, data = data)
+
+             # run the model with the simulated data: IV
+             m <- systemfit(logpgp95 ~ avexpr +  laborer, inst = ~ logem4_true +  laborer, data = data, method = "2SLS")
+
+             # save the coefficients and standard errors of both stages
+             est1 <- coef(s1)[2]
+             err1 <- sqrt(vcov(s1)[2,2])
+             pval1 <- coef(summary(s1))[2,4]
+
+             est2 <- coef(m)[2]
+             err2 <- sqrt(vcov(m)[2,2])
+             pval2 <- coef(summary(m))[2,4]
+
+             res <- rbind(res, c(est1, err1, pval1, est2, err2, pval2))
+         }
+         results <- rbind(results, c(etavec[i], pvec[j], colMedians(res)))
+     }
+ }
>
> colnames(results) <- c("eta", "prob", "pointest_first", "stderr_first", "pval_first", "pointest_second", "stderr_second", "pval_second")
>
> results <- as.data.frame(results)
>
> results$lowerci95_first <- results$pointest_first + qnorm(0.025)*results$stderr_first
> results$upperci95_first <- results$pointest_first + qnorm(0.975)*results$stderr_first
>
> results$lowerci95_second <- results$pointest_second + qnorm(0.025)*results$stderr_second
> results$upperci95_second <- results$pointest_second + qnorm(0.975)*results$stderr_second
>
>
> # p-values
> results$colors <- NA
> results$colors <- ifelse(results$pval_second <= .01, "black", results$colors)
> results$colors <- ifelse(results$pval_second > .01 & results$pval_second <= .05, "grey40", results$colors)
> results$colors <- ifelse(results$pval_second > .05 & results$pval_second <= .1, "grey80", results$colors)
> results$colors <- ifelse(results$pval_second > .1, "white", results$colors)
>
>
> # Figure 4(a)
> quartz(type="pdf", width=5, height=5, file="output/ext1.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(results$eta, results$prob, col = results$colors, xlab = expression(eta), ylab = expression(pi), pch=15, cex=0.8)
> dev.off()
null device
          1
>
>
>
>
>
> ## EXT 2: draw eta from a random distribution
>
> etavec <- seq(0, 20, length.out=100)
>
> results <-  NULL
>
> for(i in 1:length(etavec)){
+     res <- NULL
+     for(k in 1:500){
+         # create data given eta
+         data$logem4_true <- NA
+         data$logem4_true[data$campaign==0] <- data$logem4[data$campaign==0]
+         draweta <- 1+(rbeta(length(data$logem4[data$campaign==1]), etavec[i], max(etavec)-etavec[i])*20)
+         data$logem4_true[data$campaign==1] <- data$logem4[data$campaign==1] - log(draweta)
+
+         # run the model with the simulated data: first stage
+         s1 <- lm(avexpr ~ logem4_true + laborer, data = data)
+
+         # run the model with the simulated data: IV
+         m <- systemfit(logpgp95 ~ avexpr +  laborer, inst = ~ logem4_true +  laborer, data = data, method = "2SLS")
+
+         # save the coefficients and standard errors of both stages
+         est1 <- coef(s1)[2]
+         err1 <- sqrt(vcov(s1)[2,2])
+         pval1 <- coef(summary(s1))[2,4]
+
+         est2 <- coef(m)[2]
+         err2 <- sqrt(vcov(m)[2,2])
+         pval2 <- coef(summary(m))[2,4]
+
+         res <- rbind(res, c(mean(draweta), est1, err1, pval1, est2, err2, pval2))
+     }
+     results <- rbind(results, c(etavec[i], colMedians(res)))
+ }
>
> colnames(results) <- c("a", "eta_mean", "pointest_first", "stderr_first", "pval_first", "pointest_second", "stderr_second", "pval_second")
>
> results <- as.data.frame(results)
>
> results$lowerci95_first <- results$pointest_first + qnorm(0.025)*results$stderr_first
> results$upperci95_first <- results$pointest_first + qnorm(0.975)*results$stderr_first
>
> results$lowerci95_second <- results$pointest_second + qnorm(0.025)*results$stderr_second
```

```
> results$upperci95_second <- results$pointest_second + qnorm(0.975)*results$stderr_second
>
>
>
> # Figure 4(b)
> quartz(type="pdf", width=5, height=5, file="output/ext2.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest_second), results$pointest_second, type="n", ylim=c(min(results$lowerci95_second[1:51]), max(results$upperci95_second[1:51])), xlab = expression(E(eta[i])),
ylab="Second Stage Coefficient", xaxt="n")
> polygon(c(1:length(results$pointest_second), rev(1:length(results$pointest_second))), c(results$lowerci95_second, rev(results$upperci95_second)), col="grey", border=NA)
> points(1:length(results$pointest_second), results$pointest_second, type="l", lwd=3)
>
> axis(1, at=seq(1, 100, length.out=11), labels = seq(min(results$eta_mean), max(results$eta_mean), length.out=11), las=2)
> abline(h=0, col = "black")
> lines(c(1,1), c(results$lowerci95_second[1], results$upperci95_second[1]), lwd=3)
> points(1, results$pointest_second[1], pch=16)
> dev.off()
null device
          1
>
>
>
>
>
> ### Ext 3: Sensitivity analysis for campaign/no campaign and bishops together
>
> etavec1 = seq(1, 21, length.out=100)
> etavec2 = seq(0.98, 10.8, length.out=100)
>
> results <-  NULL
>
> for(i in 1:length(etavec1)){
+     for(j in 1:length(etavec2)){
+
+         # create data given eta
+         data$logem4_true <- NA
+         data$logem4_true[data$benchmark==0 & data$campaign==0] <- data$logem4[data$benchmark==0 & data$campaign==0]
+         data$logem4_true[data$benchmark==0 & data$campaign==1] <- data$logem4[data$benchmark==0 & data$campaign==1] - log(etavec1[i])
+         data$logem4_true[data$benchmark==1 & data$campaign==1] <- data$logem4[data$benchmark==1 & data$campaign==1] - log(etavec1[i]) + log(etavec2[j]/4.25)
+
+         # run the model with the simulated data: first stage
+         s1 <- lm(avexpr ~ logem4_true + laborer, data = data)
+
+         # run the model with the simulated data: IV
+         m <- systemfit(logpgp95 ~ avexpr +  laborer, inst = ~ logem4_true + laborer, data = data, method = "2SLS")
+
+         # save the coefficients, standard errors, and p-values of both stages
+         est1 <- coef(s1)[2]
+         err1 <- sqrt(vcov(s1)[2,2])
+         pval1 <- coef(summary(s1))[2,4]
+
+         est2 <- coef(m)[2]
+         err2 <- sqrt(vcov(m)[2,2])
+         pval2 <- coef(summary(m))[2,4]
+
+         results <- rbind(results, c(etavec1[i], etavec2[j], est1, err1, pval1, est2, err2, pval2))
+     }
+ }
>
> colnames(results) <- c("eta1", "eta2", "pointest_first", "stderr_first", "pval_first", "pointest_second", "stderr_second", "pval_second")
>
> results <- as.data.frame(results)
>
> results$lowerci95_first <- results$pointest_first + qnorm(0.025)*results$stderr_first
> results$upperci95_first <- results$pointest_first + qnorm(0.975)*results$stderr_first
>
> results$lowerci95_second <- results$pointest_second + qnorm(0.025)*results$stderr_second
> results$upperci95_second <- results$pointest_second + qnorm(0.975)*results$stderr_second
>
>
> # p-values
> results$colors <- NA
> results$colors <- ifelse(results$pval_second <= .01, "black", results$colors)
> results$colors <- ifelse(results$pval_second > .01 & results$pval_second <= .05, "grey40", results$colors)
> results$colors <- ifelse(results$pval_second > .05 & results$pval_second <= .1, "grey80", results$colors)
> results$colors <- ifelse(results$pval_second > .1, "white", results$colors)
>
>
> # Figure 4(c)
> quartz(type="pdf", width=5, height=5, file="output/ext3.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(results$eta1, results$eta2, col = results$colors, xlab = expression(eta), ylab = expression(eta[2]), pch=15, cex=0.8)
> dev.off()
null device
          1
>
```