# Sensitivity analysis of deterministic models through Latin hypercube sampling:
A model for transmission of HIV among homosexual men *

John M. Drake & Pejman Rohani

1   A model for the transmission of HIV among homosexual men

In 2000, approximately 30% of gay men in San Francisco were infected with HIV, 50% of which were taking combination antiretroviral therapy (ART). ART was effective at reducing the AIDS death rate in San Francisco, but because ART does not completely eliminate infectivity, it was unclear whether the net effect of increased distribution of ART would be to increase or decrease the incidence of HIV. To study the effects of possible increased ART distribution, including the effect of emerging resistance, Blower *et al.* introduced the following model (Blower, S.M., *et al.* 2000. A tale of two futres: HIV and antiretroviral therapy in San Francisco. *Science* 287:650-654.):
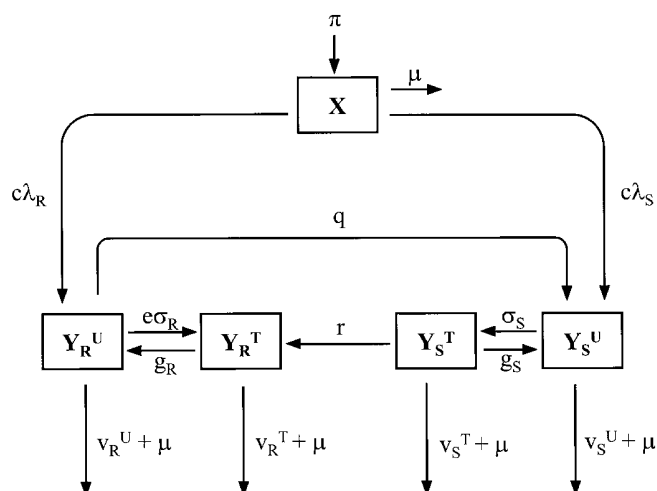
This model has compartments for susceptible individuals (X), untreated individuals infected with either drug-sensitive (Y_S^U) or drug-resistant strains (Y_R^U), and ART-treated individuals infected with either drug-sensitive (Y_S^T) or drug-resistant strains (Y_R^T). The parameter's subscript specifies whether the infection is drug-sensitive (S) or drug-resistant (R). The superscript identifies whether the individuals are treated with ART (T) or untreated (U). Parameters include rate at which gay men join the sexually active community (π), average time during which new partners are acquired ($1/\mu$), average number of

---

new sex partners per year (c), probability of a drug-resistant case (relative to a drug-sensitive case) transmitting drug-sensitive viruses (p), average time for an untreated drug-resistant infection to revert to a drug-sensitive infection ($1/q$), per capita effective treatment rate (σ), relative efficacy of ART in treating drug-resistant infections (e), rate of emergence of resistance due to acquired resistance (r), proportion of cases that give up ART per year (g), and average disease progression rate (ν). Finally, λ specifies the per capita force-of-infection for drug-sensitive (λ_S) and drug-resistant (λ_R) HIV. These force-of-infection terms are given by

$$\lambda_S = \frac{\beta_S^U Y_S^U + \beta_S^T Y_S^T + p_S^U \beta_S^U Y_R^U + p_S^T \beta_S^T Y_R^T}{N}$$

$$\lambda_R = \frac{\beta_R^U Y_R^U + \beta_R^T Y_R^T}{N}$$

where β_S^U, β_S^T, β_R^U, and β_R^T are the transmission rates for the four classes of infected cases and N is the total population size. Notice that we are here assuming transmission to be frequency-dependent.

This flow diagram can be expressed as the following equations:

$$\frac{dX}{dt} = \pi - (c(\lambda_S + \lambda_R) + \mu)X$$

$$\frac{dY_S^U}{dt} = \lambda_S cX + qY_R^U + g_S Y_S^T - Y_S^U(\sigma_S + \nu_S^U + \mu)$$

$$\frac{dY_S^T}{dt} = Y_S^U \sigma_S - Y_S^T(g_S + r + \nu_S^T + \mu)$$

$$\frac{dY_R^U}{dt} = Xc\lambda_R + Y_R^T g_R - Y_R^U(q + e\sigma_R + \nu_R^U + \mu)$$

$$\frac{dY_R^T}{dt} = Y_R^U e\sigma_R + Y_S^T r - Y_R^T(g_R + \nu_R^T + \mu)$$

Our objectives in this exercise are:

1. Numerically solve the differential equation

2. Use latin hypercube sampling to explore the robustness of model results to uncertainty in a 9-dimensional parameter space

Because we will need regularly to evaluate the force-of-infection, we write two R functions that take as arguments all the various β's and p's, the state variables, and the population size (N).

```
> lambdaS <- function(betaSU,YSU,betaST,YST,pSU,YRU,pST,YRT,N){
+   (betaSU*YSU+betaST*YST+pSU*betaSU*YRU+pST*betaST*YRT)/N
+ }
> lambdaR <- function(betaRU,YRU,betaRT,YRT,N) (betaRU*YRU+betaRT*YRT)/N
```

Further, for biological plausibility, we will require that $\beta_S^T < \beta_S^U$ and that $\beta_R^U < \beta_S^U$. To enforce this condition, we will introduce some new parameters $\alpha_1 < 1$ and $\alpha_2 < 1$ so that $\beta_S^T = \alpha_1 \beta_S^U$ and $\beta_R^U = \alpha_2 \beta_S^U$. Additionally, since $\beta_R^T < \beta_R^U$ we will use the reparameterization $\beta_R^T = \alpha_3 \beta_R^U$ in terms of α_3.

Turning to the treatment parameters (σ), Blower *et al.* proposed that these quantities could be obtained by rearranging the equations

$$F_S = \sigma_S/(\sigma_S + \nu_S^U + \mu)$$
$$F_R = \sigma_R/(\sigma_R + \nu_R^U + q + \mu)$$

where F_S is the fraction of drug-sensitive cases that are treated and F_R is the fraction of drug-resistant cases that are treated. This is important because F_S and F_R are in principle estimable quantities.

This model, then, may be represented in R in terms of its time derivatives using the following function. Notice that we use a dummy variables (Y) to track the cumulative number of HIV infections over time.

```
> hiv.model.sf <- function(t, x, params){
+    X <- x[1]
+    Y <- x[2]
+    YSU <- x[3]
+    YST <- x[4]
+    YRU <- x[5]
+    YRT <- x[6]
+    N <- X+YSU+YST+YRU+YRT
+    betaST <- params$alpha*params$betaSU
+    betaRU <- params$alpha2*params$betaSU
+    betaRT <- params$alpha3*betaRU
+    sigmaS <- params$FS*(params$nuSU+params$mu)/(1-params$FS)
+    sigmaR <- params$FR*(params$nuRU+params$mu+params$q)/(1-params$FR)
+    lambdaS <- as.numeric(lambdaS(params$betaSU,YSU,betaST,YST,params$pSU,YRU,params$pST,YRT,N))
+    lambdaR <- as.numeric(lambdaR(betaRU,YRU,betaRT,YRT,N))
+    dX <- params$pi - (params$c*(lambdaS+lambdaR)+params$mu)*X
+    dY <- (params$c*(lambdaS+lambdaR)+params$mu)*X
+    dYSU <- lambdaS*params$c*X + params$q*YRU + params$gS*YST - YSU*(sigmaS+params$nuSU+params$mu)
+    dYST <- YSU*sigmaS - YST*(params$gS + params$r + params$nuST + params$mu)
+    dYRU <- X*params$c*lambdaR + YRT*params$gR - YRU*(params$q + params$e*sigmaR
+                                                    + params$nuRU + params$mu)
+    dYRT <- YRU*params$e*sigmaR + YST*params$r - YRT*(params$gR + params$nuRT + params$mu)
+    list(c(dX, dY, dYSU, dYST, dYRU, dYRT))
+ }
```

Using parameter values suggested by Blower *et al.*, we solve the equations in the usual way,

```
> require(deSolve)
> times <- seq(0, 20, by=1)   #solve for 20 years
> params <- list(betaSU=0.1,
+                alpha=0.1+(0.5-0.1)/2,
+                pSU=0.5,
+                pST=0.5,
+                FS=0.7,
+                FR=0.7,
+                alpha2=0.5,
+                alpha3=0.5,
+                pi=2133,
+                c=1.7,
+                mu=1/30,
+                q=52/6,
+                gS=0.05,
```

```
+                    nuSU=1/12,
+                    r=0.10,
+                    nuST=1/27,
+                    gR=0.1,
+                    e=0.5,
+                    nuRU=1/27,
+                    nuRT=1/27)
> pop.size <- 800000*0.5*0.15
> xstart <- c(X=pop.size*0.7, Y=pop.size*0.3, YSU=pop.size*0.3*0.5*0.999,
+             YST=pop.size*0.3*0.5*0.85, YRU=pop.size*0.3*0.5*0.001, YRT=pop.size*0.3*0.5*0.15)
> out <- as.data.frame(lsoda(xstart, times, hiv.model.sf, params))
```
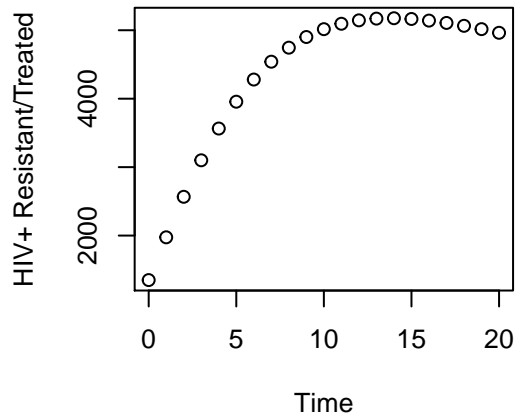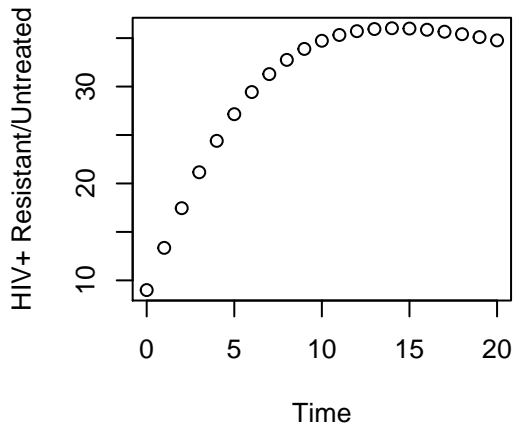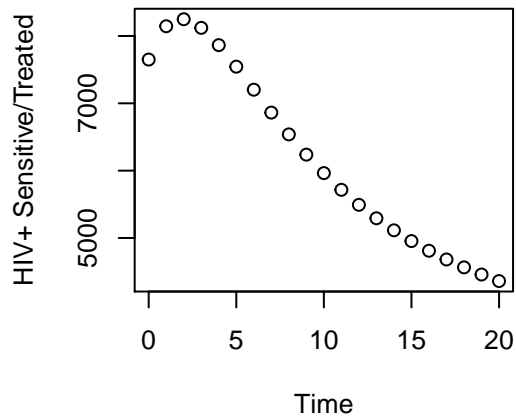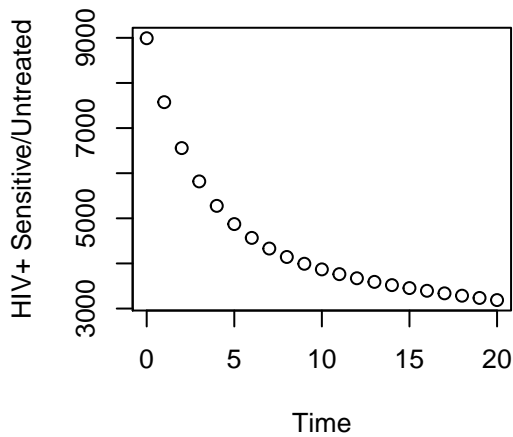
and plot the resulting trajectory of each infected class.

```
> par(mfrow=c(2,2))
> plot(out$time, out$YSU, xlab='Time', ylab='HIV+ Sensitive/Untreated')
> plot(out$time, out$YST, xlab='Time', ylab='HIV+ Sensitive/Treated')
> plot(out$time, out$YRU, xlab='Time', ylab='HIV+ Resistant/Untreated')
> plot(out$time, out$YRT, xlab='Time', ylab='HIV+ Resistant/Treated')
```

This exercise will investigate the effectiveness of ART at reducing the number of new infections. Is it not known what the coverage rate will be. Therefore, we will study the behavior of the model over a range of values from 0 to 1. For simplicity, we will assume $F_S = F_R = F$.

We first calculate the number of infections prevented over the range of F where all other parameters are held at our best guesses. We refer to this as our *baseline* estimate. Notice how we use the function `diff` to obtain the number of new infections in each year.

```
> infections.prevented.baseline<-c()
> for(F in seq(0,0.98,by=0.02)){
+    params <- list(betaSU=0.1, alpha=0.1+(0.5-0.1)/2 , pSU=0.5, pST=0.5, FS=F, FR=F, alpha2=0.5 , alpha3
+    out <- as.data.frame(lsoda(xstart, times, hiv.model.sf, params))
+    new.infections.art <- sum(diff(out$Y))
+
+    params <- list(betaSU=0.1, alpha=0.1+(0.5-0.1)/2 , pSU=0.5, pST=0.5, FS=0, FR=0, alpha2=0.5 , alpha3
+    out <- as.data.frame(lsoda(xstart, times, hiv.model.sf, params))
```

```
+    new.infections.no.art <- sum(diff(out$Y))
+
+    infections.prevented.baseline <- c(infections.prevented.baseline, new.infections.no.art-new.infectio
+ }
> plot(seq(0,0.98,by=0.02), infections.prevented.baseline, type='l', lwd=3, ylim=c(-200,20000),
+               xlab='Fraction of cases treated',
+               ylab='Infections prevented')
```
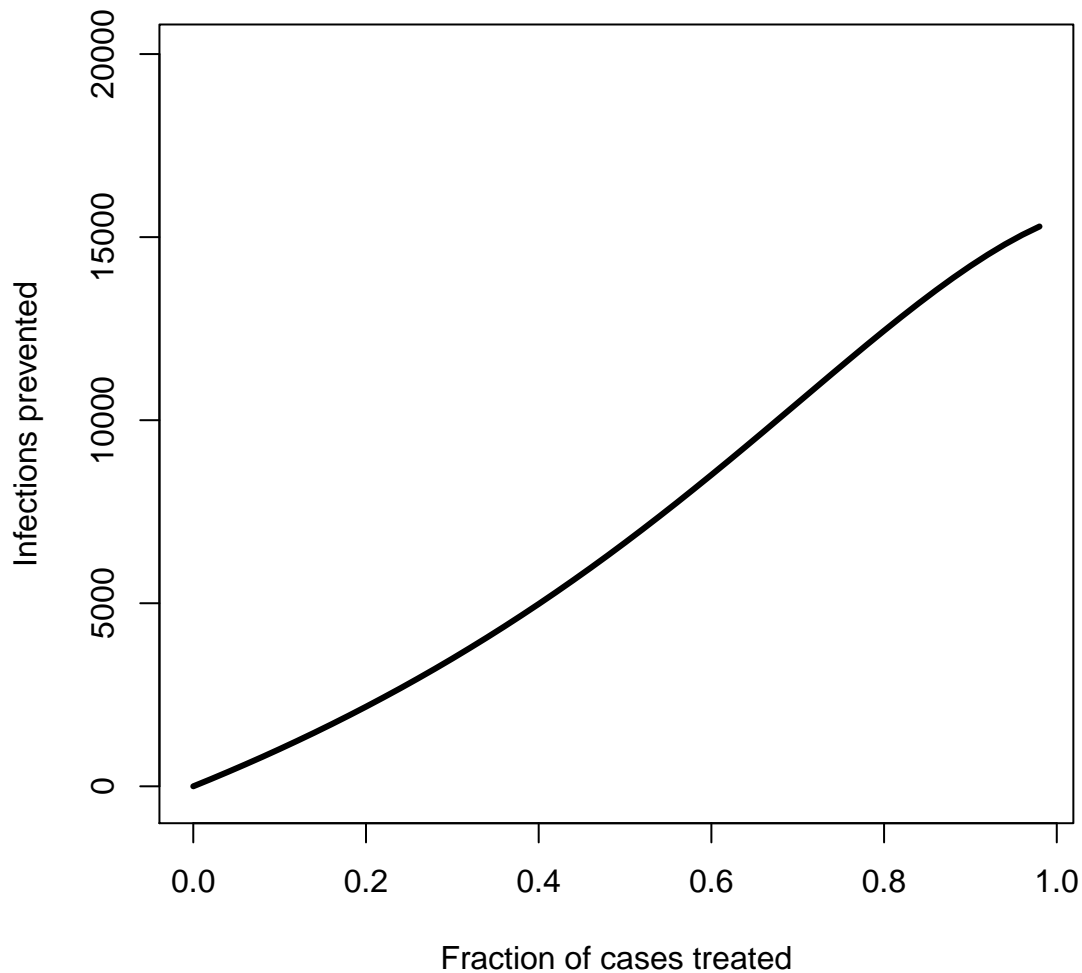

It appears that ART could prevent as many as 15,000 cases over 20 years. But, how reliable is this estimate? After all, there are twenty parameters in the model and none of these are known exactly. We can answer this question using latin hypercube sampling. The first step is to generate a matrix of sample points in the 18-dimensional unit cube (i.e., all parameters but F_S and F_R. In R, this is performed with the following lines.

```
> require(lhs)            #add the lhs library
```

```
> h <- 500                    #choose number of points
> lhs<-maximinLHS(h,18)    #simulate
```

Now, to map these points in the unit cube to our parameters, we need minimum and maximum values for each.

```
> betaSU.min <- 0.0
> betaSU.max <- 0.2
> alpha.min <- 0.2
> alpha.max <- 0.4
> pSU.min <- 0.4
> pSU.max <- 0.6
> pST.min <- 0.4
> pST.max <- 0.6
> alpha2.min <- 0.4
> alpha2.max <- 0.6
> alpha3.min <- 0.4
> alpha3.max <- 0.6
> pi.min <- 2133-250
> pi.max <- 2133+250
> c.min <- 1.2
> c.max <- 2.2
> mu.min <- 0.0333-0.01
> mu.max <- 0.0333+0.01
> q.min <- 8.667-1
> q.max <- 8.667+1
> gS.min <- 0.04
> gS.max <- 0.06
> nuSU.min <- 0.08333-0.01
> nuSU.max <- 0.08333+0.01
> r.min <- 0.05
> r.max <- 0.15
> nuST.min <- 0.037-0.01
> nuST.max <- 0.037+0.01
> gR.min <- 0.05
> gR.max <- 0.15
> e.min <- 0.4
> e.max <- 0.6
> nuRU.min <- 0.037-0.01
> nuRU.max <- 0.037+0.01
> nuRT.min <- 0.037-0.01
> nuRT.max <- 0.037+0.01
```

Now we can generate a "parameter set" by rescaling our simulated latin hypercube sample.

```
> params.set <- cbind(
+    betaSU = lhs[,1]*(betaSU.max-betaSU.min)+betaSU.min,
+    alpha = lhs[,2]*(alpha.max-alpha.min)+alpha.min,
+    pSU = lhs[,3]*(pSU.max-pSU.min)+pSU.min,
+    pST = lhs[,4]*(pST.max-pST.min)+pST.min,
+    alpha2 = lhs[,5]*(alpha2.max-alpha2.min)+alpha2.min,
```

```
+    alpha3 = lhs[,6]*(alpha3.max-alpha3.min)+alpha3.min,
+    pi = lhs[,7]*(pi.max-pi.min)+pi.min,
+    c = lhs[,8]*(c.max-c.min)+c.min,
+    mu = lhs[,9]*(mu.max-mu.min)+mu.min,
+    q = lhs[,10]*(q.max-q.min)+q.min,
+    gS = lhs[,11]*(gS.max-gS.min)+gS.min,
+    nuSU = lhs[,12]*(nuSU.max-nuSU.min)+nuSU.min,
+    r = lhs[,13]*(r.max-r.min)+r.min,
+    nuST = lhs[,14]*(nuST.max-nuST.min)+nuST.min,
+    gR = lhs[,15]*(gR.max-gR.min)+gR.min,
+    e = lhs[,16]*(e.max-e.min)+e.min,
+    nuRU = lhs[,17]*(nuRU.max-nuRU.min)+nuRU.min,
+    nuRT = lhs[,18]*(nuRT.max-nuRT.min)+nuRT.min)
```

Now we have all the simulated parameters we need. We can begin to exlore how these parameters affect the outcome of ART (cumulative number of infections prevented).

To facilitate later plotting, we decide how many different levels of F to consider.

```
> l <- 19
```

Next, even though we have simulated 500 points, to speed up computing we will use only a fraction of these for this demonstration.

```
> h2 <-250
```

Finally, we set up a nested loop, first to cycle through different values of F, then to cycle through different simulated parameter sets. Note the pre-allocated data frame and use of the counter j, also to speed up evaluation.

```
> j <- 1
> data <- data.frame(matrix(rep(NA,l*h2*21),nrow=l*h2))
> for(i in 1:h2){
+   for (F in seq(0,0.9,length=l)){
+     xstart <- c(X=pop.size*0.7, Y=pop.size*0.3, YSU=pop.size*0.3*0.5*0.999,
+              YST=pop.size*0.3*0.5*0.85, YRU=pop.size*0.3*0.5*0.001, YRT=pop.size*0.3*0.5*0.15)
+
+     params <- as.list(c(params.set[i,], FS=0, FR=0))
+     out <- as.data.frame(lsoda(xstart, times, hiv.model.sf, params))
+     new.infections.no.art <- sum(diff(out$Y))
+
+     params <- as.list(c(params.set[i,], FS=F, FR=F))
+     out <- as.data.frame(lsoda(xstart, times, hiv.model.sf, params))
+     new.infections.art <- sum(diff(out$Y))
+
+     infections.prevented <- new.infections.no.art-new.infections.art
+     data[j,1:20] <- params
+     data[j,21] <- infections.prevented
+     j <- j+1
+   }
+ }
> names(data) <- c(names(params),'Infections.Prevented')
```

What do the results look like? We can plot each simulated value as a point and compare this with our baseline estimate.

```
> plot(seq(0,0.98,by=0.02), infections.prevented.baseline, type='l', lwd=3, ylim=c(-200,20000),
+             xlab='Fraction of cases treated',
+             ylab='Infections prevented')
> points(data$FS, data$Infections.Prevented, pch=19, cex=0.3, col='blue')
```
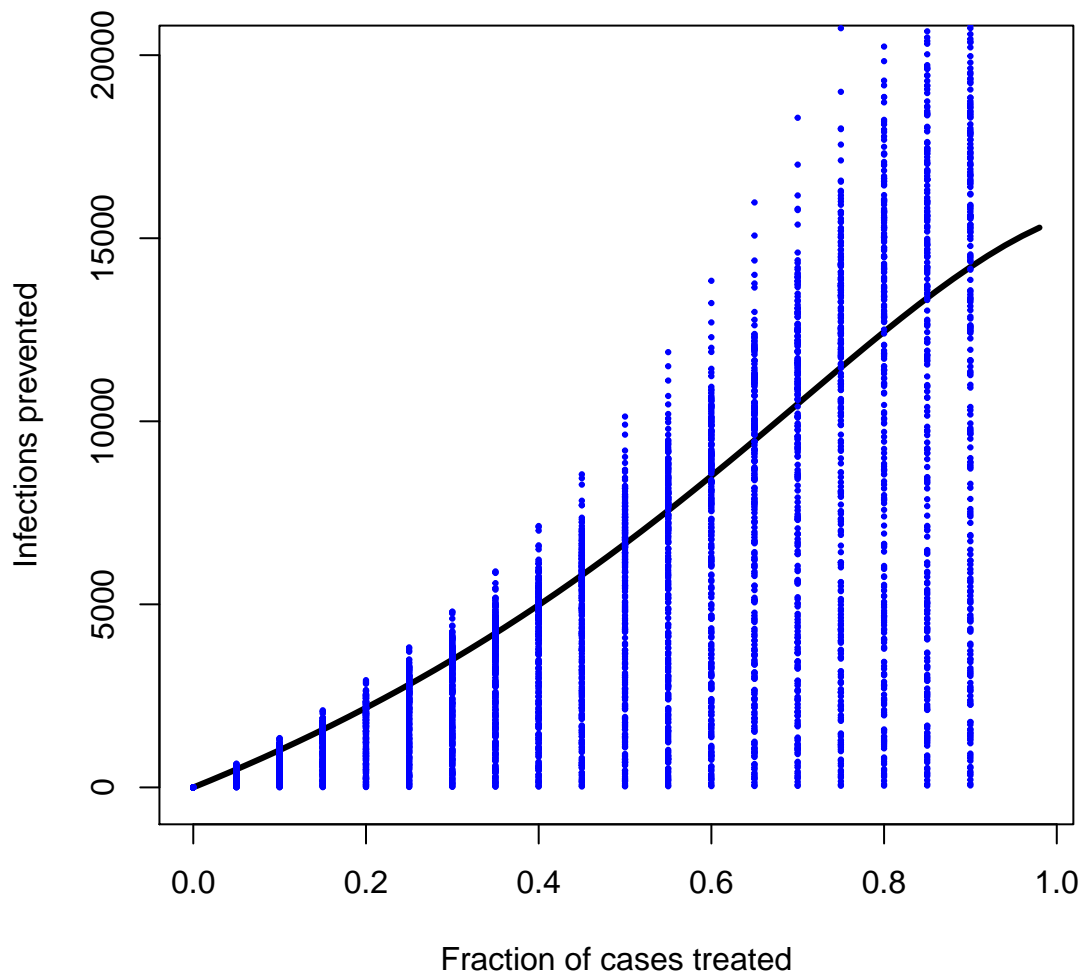

This is somewhat difficult to interpret, however, so we will summarize these points with a box-and-whisker plot.

```
> boxplot(data$Infections.Prevented~data$FS, ylim=c(-200,20000), border='blue')
> par(new=TRUE)
> plot(seq(0,0.98,by=0.02), infections.prevented.baseline, type='l', lwd=3, ylim=c(-200,20000), xlab='',
```
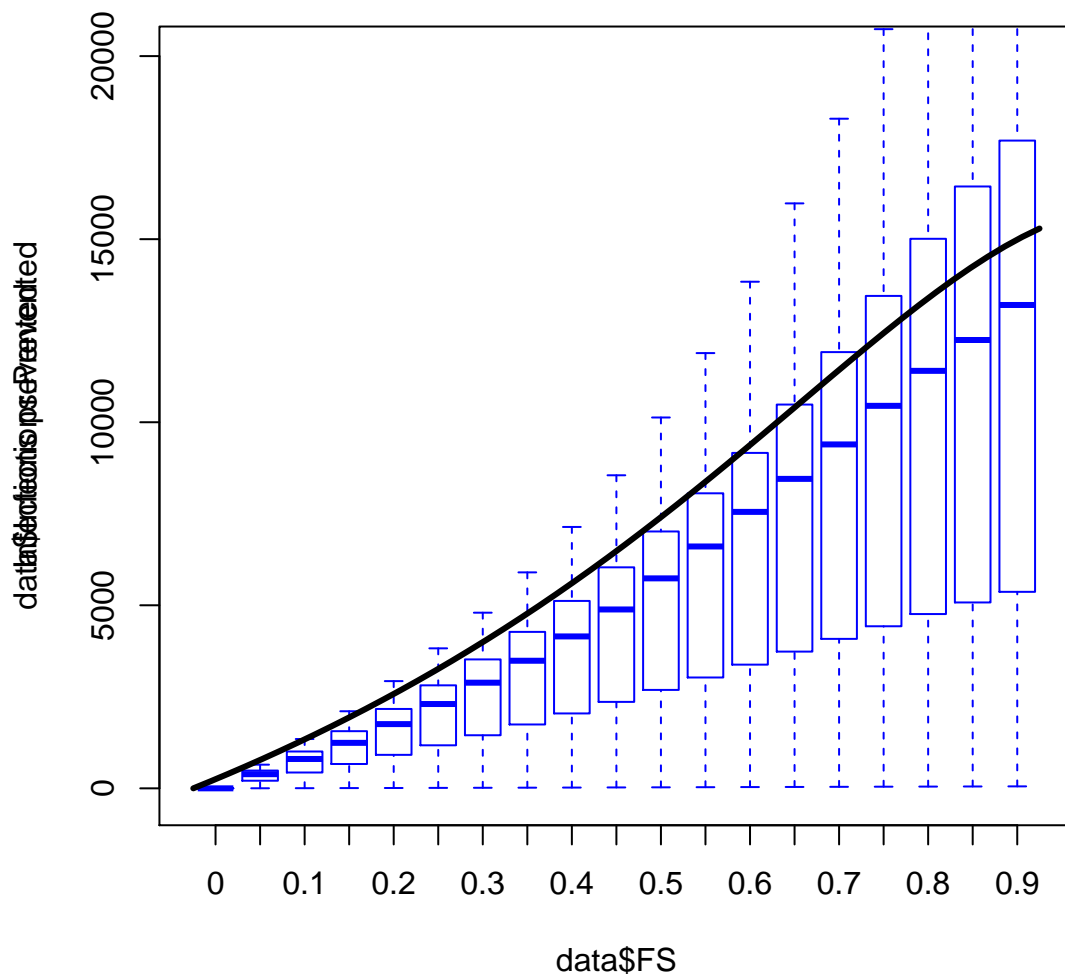
Evidently, our best guesses at parameter values are somewhat optimistic. At least ART is not found to be counter-productive in this respect – an open question at the time this study was conducted.

Exercise 1. In R, the function `cor` may be used to calculate the correlation between two sets of numbers. Write a script to determine which variable is most highly correlated with infections prevented.

Exercise 2. Having determined which variable total infections prevented is most sensitive to, re-visit the range of values that was considered for this parameter. Choose an alternative lower or upper bound and re-run the latin hypercube analysis. Does this change the result very much? What parameter would you recommend to be most important for further research?

One problem with the approach adopted here is that we're always changing more than one variable at a time. Is there any way to determine the effect of each parameter on the outcome while averaging over the other variables? Under such a thought experiment, can we identify which parameters are most important? If we may assume that the effect of a parameter on number of preventions is monotonic, then these questions can be answered by inspecting the partial rank correlations between each of the

parameters and the number of infections prevented. Partial rank correlations can be computed using the `pcc` function in the R package `sensitivity`.

```
> require(sensitivity)
> prcc <- pcc(data[,1:20], data[,21], nboot = 100, rank=TRUE)
```

The resulting partial correlation coefficients are plotted below.

```
> plot(prcc)
```