

```

1  #=====
2  # corona-bca.R
3  #=====
4
5  # NOTE: This script should be placed in a folder named 'code' and another folder
6  # named 'output' should be created at the same directory level.
7
8  #-----
9  # PRELIMINARIES:
10 #-----
11 if(1==1){
12   # Clear console and environment:
13   cat('\'014')
14   rm(list = setdiff(ls(), lsf.str()))
15
16   # install.packages('tikzDevice')
17   library(tikzDevice)
18
19   # Streamlined functions for formatting:
20   s2 <- function(x){return(sprintf('%.2f',x))}
21   s3 <- function(x){return(sprintf('%.3f',x))}
22   s4 <- function(x){return(sprintf('%.4f',x))}
23
24   # Define paths for file handling:
25   this.dir <- dirname(parent.frame(2)$ofile) # source file dir
26   setwd(this.dir) # set wd to source file dir
27   code.path <- getwd() # define code path
28   setwd('..') # set wd up one level
29   setwd(paste(getwd(), '/output', sep='')) # set wd to output folder
30   output.path <- getwd() # define output path
31   setwd('..') # set wd up one level
32   setwd(paste(getwd(), '/input', sep='')) # set wd to input folder
33   input.path <- getwd() # define input path
34
35   # Create output file:
36   # date.time <- gsub(' ', '-', Sys.time())
37   # date.time <- gsub(':', '-', date.time)
38   # out.file.name <- paste(output.path, '/corona-bca_', date.time, '.out', sep='')
39   out.file.name <- paste(output.path, '/corona-bca.out', sep='')
40   outfile <- file.create(out.file.name)
41
42   # WRITE SCRIPT TO OUTPUT FILE:
43   if(1==1){
44     source.file.name <- paste(code.path, '/corona-bca.R', sep='')
45     Rscript <- readLines(source.file.name)
46     for(i in 1:length(Rscript)){cat('\'n', Rscript[i], file=out.file.name, append=T)}
47     cat(
48       '\n\n|-----|',
49       file=out.file.name, append=T)
50     cat('\'n| R SCRIPT ABOVE
51       |', file=out.file.name, append=T)
52     cat(
53       '\n|-----|',
54       file=out.file.name, append=T)
55     cat('\'n| R OUTPUT BELOW
56       |', file=out.file.name, append=T)
57     cat(
58       '\n|-----|\'n',
59       file=out.file.name, append=T)
60   }
61 }
62
63 #-----
64 # FUNCTIONS:
65 #-----
66 {
67   SIR.fn <- function(parms){
68
69     # Unpack parameters from input list:

```

```

62 {
63   T      <- parms$T
64   dT     <- parms$dT
65   N      <- parms$N
66   R0     <- parms$R0
67   gamma  <- parms$gamma
68   rho.lo <- parms$rho.lo
69   rho.hi <- parms$rho.hi
70   threshold <- parms$threshold
71   I.0    <- parms$I.0
72   VSL    <- parms$VSL
73   sd     <- parms$sd
74   GDP.now <- parms$GDP.now
75   g      <- parms$g
76   shock.1 <- parms$shock.1
77   shock.2 <- parms$shock.2
78   r      <- parms$r
79 }
80
81 beta <- R0*gamma/(N*(1-rho.lo)) # Keeling and Rohini 2008
82
83 XX <- matrix(dT,T/dT,1)
84 S <- XX
85 I <- XX
86 R <- XX
87 D <- XX
88 c <- XX
89 D2<- XX
90
91 S[1] <- N
92 I[1] <- I.0
93 t <- 0
94 for(tt in seq(dT,T,dT)){t <- t + 1
95   S[t+1] <- S[t] - beta*I[t]*S[t] * dT
96
97   # I[t+1] <- I[t] + beta*I[t]*S[t] * dT -
98   #           (I[t]>threshold)*gamma/(1-rho.hi)*I[t] * dT -
99   #           (I[t]<threshold)*gamma/(1-rho.lo)*I[t] * dT
100
101   # I[t+1] <- I[t] + beta*I[t]*S[t] * dT -
102   #   (I[t]>threshold)* ( gamma/(1-rho.hi)*(I[t]-threshold) +
103   #   gamma/(1-rho.lo)*threshold ) * dT -
104   #   (I[t]<threshold)*gamma/(1-rho.lo)*I[t] * dT
105
106   I[t+1] <- I[t] + (beta*I[t]*S[t] - gamma*I[t]) * dT -
107   (I[t]>=threshold)* ( gamma*rho.hi/(1-rho.hi)*(I[t]-threshold) + gamma*rho.lo/(1-
108   rho.lo)*threshold ) * dT -
109   (I[t]<threshold)*gamma*rho.lo/(1-rho.lo)*I[t] * dT
110
111   R[t+1] <- R[t] + gamma*I[t] * dT
112   D[t]   <- (I[t]>=threshold)*(rho.hi/(1-rho.hi)*gamma*(I[t]-threshold) +
113   rho.lo/(1-rho.lo)*gamma*threshold) * dT +
114   (I[t]< threshold)*rho.lo/(1-rho.lo)*gamma*I[t] * dT
115
116   c[t]   <- beta*I[t]*S[t] * dT # new cases
117 }
118 S <- S[1:length(XX)]
119 I <- I[1:length(XX)]
120 R <- R[1:length(XX)]
121 D <- D[1:length(XX)]
122 c <- c[1:length(XX)]
123 D2<- D2[1:length(XX)]
124
125 return(list(S, I, R, D, c, D2))
126 }
127
128 #-----

```

```

129 # MAIN PROGRAM:
130 #-----
131
132 # PARAMETERS:
133 {
134   T           <- 365           # time horizon [days]
135   dT          <- .1           # time step for numerical integration [days]
136   N           <- 327*10^6     # U.S. population
137   R0          <- 2.4          # Basic reproductive rate [2.4]
138   gamma       <- 1/6.5        # recovery rate [1/6.5]
139   rho.lo      <- .005         # Pr death if infected below threshold [.005]
140   rho.hi      <- .015         # Pr death if infected above threshold [.015]
141   I.0         <- 4125         # initial condition
142   threshold   <- 36*10^6     # health care system capacity, number of infection cases that
   can receive high standard of care
143   VSL         <- 10*10^6     # Value of statistical life [US$]
144   sd          <- .38         # fractional reduction in contact rate from social distancing
145   GDP.now     <- 20*10^12    # current GDP [US$]
146   g           <- .0175       # GDP growth rate [/yr]
147   shock.1     <- .02         # fraction of GDP lost when outbreak is uncontrolled [.02]
148   shock.2     <- .062        # fraction of GDP lost when outbreak is controlled [.062]
149   eta         <- 1           # uncontrolled recovery rate convergence parameter
150   r           <- .03         # discount rate [\yr]
151   T.GDP       <- 30          #
152
153 # Pack up parameters into a list:
154 {
155   parms          <- list(T)
156   parms$T        <- T
157   parms$dT       <- dT
158   parms$N        <- N
159   parms$R0       <- R0
160   parms$gamma    <- gamma
161   parms$rho.lo   <- rho.lo
162   parms$rho.hi   <- rho.hi
163   parms$I.0      <- I.0
164   parms$threshold <- threshold
165   parms$VSL      <- VSL
166   parms$sd       <- sd
167   parms$GDP.now  <- GDP.now
168   parms$g        <- g
169   parms$shock.1  <- shock.1
170   parms$shock.2  <- shock.2
171   parms$eta      <- eta
172   parms$r        <- r
173   parms$T.GDP    <- T.GDP
174   parms$output.path <- output.path
175   parms0 <- parms
176 }
177
178 }
179
180 # SCENARIO FUNCTION:
181 scenario.fn <- function(parms, figswitch, tabswitch) {
182
183   # Unpack parameters from input list:
184   {
185     T           <- parms$T
186     dT          <- parms$dT
187     N           <- parms$N
188     R0          <- parms$R0
189     gamma       <- parms$gamma
190     rho.lo      <- parms$rho.lo
191     rho.hi      <- parms$rho.hi
192     threshold   <- parms$threshold
193     I.0         <- parms$I.0
194     VSL         <- parms$VSL
195     sd          <- parms$sd
196     GDP.now     <- parms$GDP.now

```

```

197 g <- parms$g
198 shock.1 <- parms$shock.1
199 shock.2 <- parms$shock.2
200 eta <- parms$eta
201 r <- parms$r
202 T.GDP <- parms$T.GDP
203 output.path <- parms$output.path
204
205
206 # Goldman Sachs projection hardwired:
207 g.0 <- matrix(g,T.GDP,1)
208 g.2 <- rbind(matrix(c(-shock.2,
209 shock.2*.055/.062,
210 shock.2*.035/.062,
211 shock.2*.020/.062),4,1),
212 matrix(g,T.GDP-4,1))
213
214 g.2 <- rbind(matrix(c(-.062,
215 .055,
216 .035,
217 .020),4,1),
218 matrix(g,T.GDP-4,1))
219 }
220
221 # SIR model:
222 {
223 outs <- SIR.fn(parms)
224 S1 <- outs[[1]] # susceptible
225 I1 <- outs[[2]] # infected
226 R1 <- outs[[3]] # recovered
227 D1 <- outs[[4]] # deaths
228 c1 <- outs[[5]] # new infection cases
229
230 # Reduced contact rate due to social distancing:
231 parms$R0 <- R0*(1-sd)
232
233 outs <- SIR.fn(parms)
234 S2 <- outs[[1]]
235 I2 <- outs[[2]]
236 R2 <- outs[[3]]
237 D2 <- outs[[4]]
238 c2 <- outs[[5]]
239 }
240
241 # GDP CURVES:
242 {
243 years <- seq(0,T.GDP,1)
244 GDP.0 <- matrix(0,length(years),1) # GDP path if no outbreak
245 GDP.1 <- matrix(0,length(years),1) # GDP path if outbreak uncontrolled
246 GDP.2 <- matrix(0,length(years),1) # GDP path if outbreak controlled
247
248 # All GDP paths start at same point
249 GDP.0[1] <- GDP.now
250 GDP.1[1] <- GDP.now
251 GDP.2[1] <- GDP.now
252
253 # Growth without outbreak:
254 for(t in 2:T.GDP){
255 GDP.0[t] <- GDP.0[t-1]*(1+g.0[t])
256 }
257
258 # Growth with controlled outbreak based on GS forecast:
259 for(t in 2:T.GDP){
260 GDP.2[t] <- GDP.2[t-1]*(1+g.2[t-1])
261 }
262
263 # Growth with controlled outbreak adjusted for different assumed initial shock:
264 for(t in 1:T.GDP){
265 GDP.2[t] <- GDP.0[t] - (GDP.0[t]-GDP.2[t])*shock.2/(-g.2[1])

```

```

266     }
267
268     # Growth with uncontrolled outbreak:
269     GDP.1[2] <- GDP.now * (1-shock.1)
270     for(t in 3:T.GDP){
271         GDP.1[t] <- GDP.0[t] - (GDP.0[t]-GDP.2[t])* ((GDP.0[2]-GDP.1[2])/(GDP.0[2]-GDP.2[2]
272         ]))^eta
273     }
274
275     PV.0 <- 0
276     PV.1 <- 0
277     PV.2 <- 0
278     for(t in 1:T.GDP){
279         PV.0 <- PV.0 + GDP.0[t]/(1+r)^(t-1)
280         PV.1 <- PV.1 + GDP.1[t]/(1+r)^(t-1)
281         PV.2 <- PV.2 + GDP.2[t]/(1+r)^(t-1)
282     }
283 }
284
285 # CREATE FIGURES:
286 if(figswitch==1){
287
288     # Infection curves:
289     {
290         tikz(paste(output.path, '/corona_fig_1A.tex', sep=''),
291             width      = 3.2,
292             height     = 3.2,
293             fontsize   = 12,
294             standalone = TRUE)
295         par(mar=c(4,4.5,1,1)) # figure border whitespace [bottom,left,top,right]
296
297         bottom <- 0
298         top    <- max(c(I1,I2))*1.1
299         left   <- 0
300         right  <- 300
301
302         plot(seq(dT,T,dT),I2,type='l',col='black',
303             main='',
304             xlab='',
305             ylab='',
306             axes=FALSE,
307             cex=1,
308             xlim=c(left,right),
309             ylim=c(bottom,top),
310             cex.axis=.8,
311             cex.lab =.5)
312
313         lines(seq(dT,T,dT),I1,lty=2,col='black')
314         lines(seq(dT,T,dT),threshold*matrix(1,T/dT,1),lty=3,col='black')
315
316         axis(1,at=seq(0,350,50),las=1,labels=TRUE,cex.axis=.65) # Draw x axis
317         axis(2,                las=1,labels=TRUE,cex.axis=.65) # Draw y axis
318
319         mtext('Number infected',side=2, line=3.25, cex=.8, las=0) # y-axis label
320         mtext('$t$ [days]',      side=1, line=2.50, cex=.8, las=1) # x-axis label
321
322         legend(left+(right-left)*.6,
323             bottom+(top-bottom)*.9,
324             legend=c('Controlled','Uncontrolled','Threshold'),
325             col=c('black','black','black'),
326             lty=c(1,2,3),
327             cex=.6)
328
329         text(left+(right-left)*.1,bottom+(top-bottom)*1,'(A)',cex=.8)
330
331         dev.off()
332     }
333 }

```

```

334 # GDP curves:
335 {
336   tikz(paste(output.path, '/corona_fig_1B.tex', sep=''),
337       width      = 3.2,
338       height     = 3.2,
339       pointsize  = 12,
340       standAlone = TRUE)
341
342   par(mar=c(4, 4.5, 1, 1)) # figure border whitespace [bottom, left, top, right]
343
344   top      <- GDP.now*1.1
345   bottom  <- GDP.now*0.9
346   left    <- 0
347   right   <- 5
348
349   plot(seq(0, 5, 1), GDP.2[1:6], type='l', col='black',
350        main='',
351        xlab='',
352        ylab='',
353        axes=FALSE,
354        cex=1,
355        ylim = c(bottom, top),
356        xlim = c(left, right),
357        cex.axis=.8,
358        cex.lab =.5)
359
360   lines(seq(0, 5, 1), GDP.0[1:6], lty=3, col='black')
361   lines(seq(0, 5, 1), GDP.1[1:6], lty=2, col='black')
362
363   axis(1, at=seq(0, 5, 1), las=1, labels=TRUE, cex.axis=.65) # Draw x axis
364   axis(2,          las=1, labels=TRUE, cex.axis=.65) # Draw y axis
365
366   mtext('GDP [\\$]', side=2, line=3.5, cex=.8, las=0) # y-axis label
367   mtext('$t$ [yrs]', side=1, line=2.5, cex=.8, las=1) # x-axis label
368
369   legend(left+(right-left)*.6,
370         bottom+(top-bottom)*.3,
371         legend=c('No outbreak', 'Controlled', 'Uncontrolled'),
372         col=c('black', 'black', 'black'),
373         lty=c(3, 1, 2),
374         cex=.6)
375
376   text(left+(right-left)*.1, bottom+(top-bottom)*1, '(B)', cex=.8)
377
378   dev.off()
379 }
380
381 }
382
383 # PRINT TABLE OF RESULTS TO OUTPUT FILE:
384 if(tabswitch==1){
385   cat('\n=====\\n', file=
386   out.file.name, append=TRUE)
387   cat('          Uncontrolled    Controlled          Delta\\n', file=
388   out.file.name, append=TRUE)
389   cat('-----\\n', file=
390   out.file.name, append=TRUE)
391   cat('Infections          ',
392       sprintf('%12.4g %12.4g %12.4g\\n', sum(c1), sum(c2), sum(c1)-sum(c2)), file=
393   out.file.name, append=TRUE)
394   cat('Deaths          ',
395       sprintf('%12.4g %12.4g %12.4g\\n', sum(D1), sum(D2), sum(D1)-sum(D2)), file=
396   out.file.name, append=TRUE)
397   cat('PV of GDP loss          ',
398       sprintf('%12.4g %12.4g %12.4g\\n', PV.0-PV.1, PV.0-PV.2, PV.2-PV.1), file=
399   out.file.name, append=TRUE)
400   cat('Value of lives lost          ',
401       sprintf('%12.4g %12.4g %12.4g\\n', VSL*sum(D1), VSL*sum(D2), VSL*(sum(D1)-sum(D2))),
402       file=out.file.name, append=TRUE)

```

```

396     cat('Total value lost      ',
397         sprintf('%12.4g %12.4g %12.4g\n', PV.0-PV.1+VSL*sum(D1), PV.0-PV.2+VSL*sum(D2),
398             PV.2-PV.1+VSL*(sum(D1)-sum(D2))), file=out.file.name, append=TRUE)
399     cat('=====\n', file=
400         out.file.name, append=TRUE)
401     }
402     return(list(I1, I2, c1, c2, D1, D2, PV.0, PV.1, PV.2, GDP.0, GDP.1, GDP.2))
403 }
404 # BENCHMARK RUN:
405 outs <- scenario.fn(parms, 1, 1)
406
407 # CREATE FIGURE 2 WITH 3 GDP CURVES:
408 if(1==1){
409     parms <- parms0
410     parms$eta <- 1
411     outs <- scenario.fn(parms, 0, 0)
412     GDP.0 <- outs[[10]]
413     GDP.2 <- outs[[12]]
414     GDP.1.same <- outs[[11]]
415     parms <- parms0
416     parms$eta <- 2
417     outs <- scenario.fn(parms, 0, 0)
418     GDP.1.fast <- outs[[11]]
419     parms <- parms0
420     parms$eta <- .5
421     outs <- scenario.fn(parms, 0, 0)
422     GDP.1.slow <- outs[[11]]
423
424     # Three GDP curves:
425     {
426         tikz(paste(output.path, '/corona_fig_2.tex', sep=''),
427             width = 3.2,
428             height = 3.2,
429             pointsize = 12,
430             standAlone = TRUE)
431
432         par(mar=c(4, 4.5, 1, 1)) # figure border whitespace [bottom, left, top, right]
433
434         top <- GDP.now*1.1
435         bottom <- GDP.now*0.9
436         left <- 0
437         right <- 5
438
439         plot(seq(0, 5, 1), GDP.2[1:6], type='l', col='black',
440             main='',
441             xlab='',
442             ylab='',
443             axes=FALSE,
444             cex=1,
445             ylim = c(bottom, top),
446             xlim = c(left, right),
447             cex.axis=.8,
448             cex.lab =.5)
449
450         lines(seq(0, 5, 1), GDP.0[1:6], lty=3, col='black')
451         lines(seq(0, 5, 1), GDP.1.fast[1:6], lty=2, col='gray')
452         lines(seq(0, 5, 1), GDP.1.slow[1:6], lty=2, col='gray')
453         lines(seq(0, 5, 1), GDP.1.same[1:6], lty=2, col='black')
454
455         axis(1, at=seq(0, 5, 1), las=1, labels=TRUE, cex.axis=.65) # Draw x axis
456         axis(2, las=1, labels=TRUE, cex.axis=.65) # Draw y axis
457
458         mtext('GDP [\\$]', side=2, line=3.5, cex=.8, las=0) # y-axis label
459         mtext('$t$ [yrs]', side=1, line=2.5, cex=.8, las=1) # x-axis label
460
461         legend(left+(right-left)*.4,
462             bottom+(top-bottom)*.3,

```

```

463     legend=c('No outbreak','Controlled','Uncontrolled benchmark','Uncontrolled
         fast','Uncontrolled slow'),
464     col=c('black','black','black','gray','gray'),
465     lty=c(3,1,2,2,2),
466     cex=.6)
467
468     dev.off()
469 }
470
471     eta <- parms0$eta
472 }
473
474 # FIND BREAK-EVEN VALUES:
475 if(1==1){
476     # R0
477     if(1==1){
478         cat('Finding break-even value for R0\n')
479         parms <- parms0
480         R0.range <- seq(1.0,7,.01)
481         NB <- matrix(0,length(R0.range),1)
482         TD1 <- matrix(0,length(R0.range),1)
483         TD2 <- matrix(0,length(R0.range),1)
484         for(R0 in R0.range){
485             parms$R0 <- R0
486             i <- which(R0.range==R0)
487             outs <- scenario.fn(parms,0,0)
488             D1 <- outs[[5]]
489             D2 <- outs[[6]]
490             PV.1 <- outs[[8]]
491             PV.2 <- outs[[9]]
492             NB[i] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
493             TD1[i] <- sum(D1)
494             TD2[i] <- sum(D2)
495         }
496         pdf(paste(output.path,'/fig_be_R0.pdf',sep=''))
497         plot(R0.range,NB,type='l')
498         lines(R0.range,matrix(0,length(R0.range),1),lty=2)
499         dev.off()
500         temp <- which(NB>0)
501         R0.be1 <- R0.range[min(temp)]
502         R0.be2 <- R0.range[max(temp)]
503         lo <- min(temp);if(lo==1) {R0.be1 <- -999}
504         hi <- max(temp);if(hi==length(NB)) {R0.be2 <- 999}
505
506         pdf(paste(output.path,'/fig_TD_R0.pdf',sep=''))
507         plot(R0.range,TD1,type='l')
508         lines(R0.range,TD2,lty=2)
509         lines(R0.range,TD1-TD2,lty=3)
510         dev.off()
511     }
512 }
513
514 # Save Figure 4:
515 if(1==1){
516     tikz(paste(output.path,'/corona_fig_4.tex',sep=''),
517         width = 3,
518         height = 3,
519         pointsize = 12,
520         standAlone = TRUE)
521
522     par(mar=c(3.5,3.5,1,1)) # figure border whitespace [bottom,left,top,right]
523
524     left <- min(R0.range)
525     right <- max(R0.range)
526     top <- max(TD1/10^6)
527     bottom <- 0
528
529     plot(R0.range,TD1/10^6,
530         type='l',

```



```

531     xlab='',
532     ylab='',
533     axes=FALSE,
534     cex.axis=.8,
535     cex.lab=.5)
536
537 lines(R0.range, TD2/10^6, lty=2)
538 lines(R0.range, (TD1-TD2)/10^6, lty=3)
539 # lines(c(parms0$R0, parms0$R0), c(0, top), lty=2, col='gray')
540
541 axis(1, las=1, labels=TRUE, cex.axis=.65) # Draw x axis
542 axis(2, las=1, labels=TRUE, cex.axis=.65) # Draw y axis
543
544 mtext('Total deaths [million]', side=2, line=2.2, cex=.65, las=0) # y-axis label
545 mtext('R$_0$', side=1, line=2.0, cex=.65, las=1) # x-axis label
546
547 legend(left+(right-left) *0.0,
548        bottom+(top-bottom)*1.0,
549        legend=c('uncontrolled', 'controlled', 'difference'),
550        col=c('black', 'black', 'black'),
551        lty=c(1, 2, 3),
552        cex=.5)
553
554 #text(left+(right-left)*.1, bottom+(top-bottom)*.9, '(A)', cex=.8)
555
556 dev.off()
557 }
558
559 # rho.lo
560 if(1==1){
561   cat('Finding break-even value for rho.lo\n')
562   parms <- parms0
563   rho.lo.range <- seq(0, rho.hi, .0001)
564   NB <- matrix(0, length(rho.lo.range), 1)
565   for(rho.lo in rho.lo.range){
566
567     parms$rho.lo <- rho.lo
568     outs <- scenario.fn(parms, 0, 0)
569     D1 <- outs[[5]]
570     D2 <- outs[[6]]
571     PV.1 <- outs[[8]]
572     PV.2 <- outs[[9]]
573     NB[which(rho.lo.range==rho.lo)] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
574
575   }
576   pdf(paste(output.path, '/fig_be_rho.lo.pdf', sep=''))
577   plot(rho.lo.range, NB, type='l')
578   lines(rho.lo.range, matrix(0, length(rho.lo.range), 1), lty=2)
579   dev.off()
580   temp <- which(NB>0)
581   rho.lo.be1 <- rho.lo.range[min(temp)]
582   rho.lo.be2 <- rho.lo.range[max(temp)]
583   lo <- min(temp); if(lo==1) {rho.lo.be1 <- -999}
584   hi <- max(temp); if(hi==length(NB)) {rho.lo.be2 <- 999}
585 }
586
587 # rho.hi
588 if(1==1){
589   cat('Finding break-even value for rho.hi\n')
590   parms <- parms0
591   rho.hi.range <- seq(0, .05, .0001)
592   NB <- matrix(0, length(rho.hi.range), 1)
593   for(rho.hi in rho.hi.range){
594
595     parms$rho.hi <- rho.hi
596     outs <- scenario.fn(parms, 0, 0)
597     D1 <- outs[[5]]
598     D2 <- outs[[6]]
599     PV.1 <- outs[[8]]

```

```

600     PV.2 <- outs[[9]]
601     NB[which(rho.hi.range==rho.hi)] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
602
603 }
604 pdf(paste(output.path, '/fig_be_rho.hi.pdf', sep=''))
605 plot(rho.hi.range, NB, type='l')
606 lines(rho.hi.range, matrix(0, length(rho.hi.range), 1), lty=2)
607 dev.off()
608 temp <- which(NB>0)
609 rho.hi.be1 <- rho.hi.range[min(temp)]
610 rho.hi.be2 <- rho.hi.range[max(temp)]
611 lo <- min(temp); if(lo==1) {rho.hi.be1 <- -999}
612 hi <- max(temp); if(hi==length(NB)) {rho.hi.be2 <- 999}
613 }
614
615 # sd
616 if(1==1){
617   cat('Finding break-even value for sd\n')
618   parms <- parms0
619   sd.range <- seq(0, .9, .0005)
620   NB <- matrix(0, length(sd.range), 1)
621   for(sd in sd.range){
622
623     parms$sd <- sd
624     outs <- scenario.fn(parms, 0, 0)
625     D1 <- outs[[5]]
626     D2 <- outs[[6]]
627     PV.1 <- outs[[8]]
628     PV.2 <- outs[[9]]
629     NB[which(sd.range==sd)] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
630
631   }
632   pdf(paste(output.path, '/fig_be_sd.pdf', sep=''))
633   plot(sd.range, NB, type='l')
634   lines(sd.range, matrix(0, length(sd.range), 1), lty=2)
635   dev.off()
636   temp <- which(NB>0)
637   sd.be1 <- sd.range[min(temp)]
638   sd.be2 <- sd.range[max(temp)]
639   lo <- min(temp); if(lo==1) {sd.be1 <- -999}
640   hi <- max(temp); if(hi==length(NB)) {sd.be2 <- 999}
641 }
642
643 # VSL
644 if(1==1){
645   cat('Finding break-even value for VSL\n')
646   parms <- parms0
647   VSL.range <- seq(.5, 20, .05)*10^6
648   NB <- matrix(0, length(VSL.range), 1)
649   for(VSL in VSL.range){
650
651     parms$VSL <- VSL
652     outs <- scenario.fn(parms, 0, 0)
653     D1 <- outs[[5]]
654     D2 <- outs[[6]]
655     PV.1 <- outs[[8]]
656     PV.2 <- outs[[9]]
657     NB[which(VSL.range==VSL)] <- PV.2-PV.1+VSL*sum(D1)-VSL*sum(D2)
658
659   }
660   pdf(paste(output.path, '/fig_be_VSL.pdf', sep=''))
661   plot(VSL.range, NB, type='l')
662   lines(VSL.range, matrix(0, length(VSL.range), 1), lty=2)
663   dev.off()
664   temp <- which(NB>0)
665   VSL.be1 <- VSL.range[min(temp)]
666   VSL.be2 <- VSL.range[max(temp)]
667   lo <- min(temp); if(lo==1) {VSL.be1 <- -999}
668   hi <- max(temp); if(hi==length(NB)) {VSL.be2 <- 999}

```

```

669     VSL <- parms0$VSL
670 }
671
672 # shock.1
673 if(1==1){
674     cat('Finding break-even value for shock.1\n')
675     parms <- parms0
676     shock.1.range <- seq(0,parms0$shock.2,.0005)
677     NB <- matrix(0,length(shock.1.range),1)
678     for(shock.1 in shock.1.range){
679
680         parms$shock.1 <- shock.1
681         outs <- scenario.fn(parms,0,0)
682         D1 <- outs[[5]]
683         D2 <- outs[[6]]
684         PV.1 <- outs[[8]]
685         PV.2 <- outs[[9]]
686         NB[which(shock.1.range==shock.1)] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(
        D2)
687
688     }
689     pdf(paste(output.path,'/fig_be_shock1.pdf',sep=''))
690     plot(shock.1.range,NB,type='l')
691     lines(shock.1.range,matrix(0,length(shock.1.range),1),lty=2)
692     dev.off()
693     temp <- which(NB>0)
694     shock.1.be1 <- shock.1.range[min(temp)]
695     shock.1.be2 <- shock.1.range[max(temp)]
696     lo <- min(temp);if(lo==1) {shock.1.be1 <- -999}
697     hi <- max(temp);if(hi==length(NB)) {shock.1.be2 <- 999}
698 }
699
700 # shock.2
701 if(1==1){
702     cat('Finding break-even value for shock.2\n')
703     parms <- parms0
704     shock.2.range <- seq(parms0$shock.1,.3,.0005)
705     NB <- matrix(0,length(shock.2.range),1)
706     for(shock.2 in shock.2.range){
707
708         parms$shock.2 <- shock.2
709         outs <- scenario.fn(parms,0,0)
710         D1 <- outs[[5]]
711         D2 <- outs[[6]]
712         PV.1 <- outs[[8]]
713         PV.2 <- outs[[9]]
714         NB[which(shock.2.range==shock.2)] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(
        D2)
715
716     }
717     pdf(paste(output.path,'/fig_be_shock2.pdf',sep=''))
718     plot(shock.2.range,NB,type='l')
719     lines(shock.2.range,matrix(0,length(shock.2.range),1),lty=2)
720     dev.off()
721     temp <- which(NB>0)
722     shock.2.be1 <- shock.2.range[min(temp)]
723     shock.2.be2 <- shock.2.range[max(temp)]
724     lo <- min(temp);if(lo==1) {shock.2.be1 <- -999}
725     hi <- max(temp);if(hi==length(NB)) {shock.2.be2 <- 999}
726 }
727
728 # threshold
729 if(1==1){
730     cat('Finding break-even value for threshold\n')
731     parms <- parms0
732     threshold.range <- seq(0,2*parms0$threshold,parms0$threshold/200)
733     NB <- matrix(0,length(threshold.range),1)
734     for(threshold in threshold.range){
735

```

```

736     parms$threshold <- threshold
737     outs <- scenario.fn(parms, 0, 0)
738     D1 <- outs[[5]]
739     D2 <- outs[[6]]
740     PV.1 <- outs[[8]]
741     PV.2 <- outs[[9]]
742     NB[which(threshold.range==threshold)] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*
       sum(D2)
743
744   }
745   pdf(paste(output.path, '/fig_be_threshold.pdf', sep=''))
746   plot(threshold.range, NB, type='l')
747   lines(threshold.range, matrix(0, length(threshold.range), 1), lty=2)
748   dev.off()
749   temp <- which(NB>0)
750   threshold.be1 <- threshold.range[min(temp)]
751   threshold.be2 <- threshold.range[max(temp)]
752   lo <- min(temp); if(lo==1) {threshold.be1 <- -999}
753   hi <- max(temp); if(hi==length(NB)) {threshold.be2 <- 999}
754 }
755
756 # r
757 if(1==1){
758   cat('Finding break-even value for r\n')
759   parms <- parms0
760   r.range <- seq(0, .1, .001)
761   NB <- matrix(0, length(r.range), 1)
762   for(r in r.range){
763
764     parms$r <- r
765     outs <- scenario.fn(parms, 0, 0)
766     D1 <- outs[[5]]
767     D2 <- outs[[6]]
768     PV.1 <- outs[[8]]
769     PV.2 <- outs[[9]]
770     NB[which(r.range==r)] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
771
772   }
773   pdf(paste(output.path, '/fig_be_r.pdf', sep=''))
774   plot(r.range, NB, type='l')
775   lines(r.range, matrix(0, length(r.range), 1), lty=2)
776   dev.off()
777   temp <- which(NB>0)
778   r.be1 <- r.range[min(temp)]
779   r.be2 <- r.range[max(temp)]
780   lo <- min(temp); if(lo==1) {r.be1 <- -999}
781   hi <- max(temp); if(hi==length(NB)) {r.be2 <- 999}
782 }
783
784 # T.GDP
785 if(1==1){
786   cat('Finding break-even value for T.GDP\n')
787   parms <- parms0
788   T.GDP.range <- seq(10, 100, 1)
789   NB <- matrix(0, length(T.GDP.range), 1)
790   for(T.GDP in T.GDP.range){
791
792     parms$T.GDP <- T.GDP
793     outs <- scenario.fn(parms, 0, 0)
794     D1 <- outs[[5]]
795     D2 <- outs[[6]]
796     PV.1 <- outs[[8]]
797     PV.2 <- outs[[9]]
798     NB[which(T.GDP.range==T.GDP)] <- PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
799
800   }
801   pdf(paste(output.path, '/fig_be_T.pdf', sep=''))
802   plot(T.GDP.range, NB, type='l')
803   lines(T.GDP.range, matrix(0, length(T.GDP.range), 1), lty=2)

```

```

804     dev.off()
805     temp <- which(NB>0)
806     T.GDP.be1 <- T.GDP.range[min(temp)]
807     T.GDP.be2 <- T.GDP.range[max(temp)]
808     lo <- min(temp); if(lo==1) {T.GDP.be1 <- -999}
809     hi <- max(temp); if(hi==length(NB)) {T.GDP.be2 <- 999}
810 }
811 }
812
813 # PRINT TABLE OF BREAK-EVEN RESULTS TO OUTPUT FILE:
814 if(1==1){
815   cat('\n===== \n', file=
816     out.file.name, append=TRUE)
817   cat('          Default          Break-even\n', file=out.file.name,
818     append=TRUE)
819   cat('----- \n', file=out.file.name,
820     append=TRUE)
821   cat('R0          ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$R0, R0.be1,
822     R0.be2), file=out.file.name, append=TRUE)
823   cat('Low mortality rate ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$rho.lo,
824     rho.lo.be1, rho.lo.be2), file=out.file.name, append=TRUE)
825   cat('High mortality rate ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$rho.hi,
826     rho.hi.be1, rho.hi.be2), file=out.file.name, append=TRUE)
827   cat('Reduction in contact rate ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$sd, sd.be1,
828     sd.be2), file=out.file.name, append=TRUE)
829   cat('VSL          ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$VSL, VSL.be1,
830     VSL.be2), file=out.file.name, append=TRUE)
831   cat('Uncontrolled shock ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$shock.1,
832     shock.1.be1, shock.1.be2), file=out.file.name, append=TRUE)
833   cat('Controlled shock ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$shock.2,
834     shock.2.be1, shock.2.be2), file=out.file.name, append=TRUE)
835   cat('Threshold          ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$threshold,
836     threshold.be1, threshold.be2), file=out.file.name, append=TRUE)
837   cat('Discount rate      ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$r, r.be1,
838     r.be2), file=out.file.name, append=TRUE)
839   cat('Time horizon        ', sprintf('%12.4g (%6.3g,%6.3g)\n', parms0$T.GDP,
840     T.GDP.be1, T.GDP.be2), file=out.file.name, append=TRUE)
841   cat('===== \n', file=out.file.name,
842     append=TRUE)
843 }
844
845 # NB=0 ISOQUANT shock.1 vs shock.2:
846 if(1==1){
847   cat('Working on shock.1-shock.2 isoquant\n')
848   nodes <- 30
849   shock.1.0 <- shock.1; shock.1.range <- seq(0, .15, .15/(nodes-1))
850   shock.2.0 <- shock.2; shock.2.range <- seq(0, .15, .15/(nodes-1))
851   NBiso <- matrix(0, length(shock.2.range), length(shock.1.range))
852   for(shock.1 in shock.1.range){
853     for(shock.2 in shock.2.range){
854       if(shock.2>=shock.1){
855         parms <- parms0
856         parms$shock.1 <- shock.1
857         parms$shock.2 <- shock.2
858         outs <- scenario.fn(parms, 0, 0)
859         D1 <- outs[[5]]
860         D2 <- outs[[6]]
861         PV.1 <- outs[[8]]
862         PV.2 <- outs[[9]]
863         NBiso[which(shock.2.range==shock.2), which(shock.1.range==shock.1)] <-
864           PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
865       }
866     }
867   }
868 }
869
870 # Save figure:
871 tikz(paste(output.path, '/corona_fig_iso_1.tex', sep=''),
872     width = 3,
873     height = 3,

```

```

859     pointsize = 12,
860     standAlone = TRUE)
861
862     par(mar=c(3.5,3.5,1,1)) # figure border whitespace [bottom,left,top,right]
863
864     left <- min(shock.2.range)
865     right <- max(shock.2.range)
866     top <- max(shock.1.range)
867     bottom <- min(shock.1.range)
868
869     contour(shock.2.range, shock.1.range, NBiso, levels=0,
870            drawlabels=FALSE,
871            xlab='',
872            ylab='',
873            axes=FALSE,
874            cex.axis=.8,
875            cex.lab=.5)
876
877     points(parms0$shock.2, parms0$shock.1, pch=1, cex=.5)
878     # lines(c(0,max(shock.1.range)), c(0,max(shock.1.range)), lty=2, col='gray')
879
880     axis(1, las=1, labels=TRUE, cex.axis=.65) # Draw x axis
881     axis(2, las=1, labels=TRUE, cex.axis=.65) # Draw y axis
882
883     mtext('Uncontrolled shock', side=2, line=2.2, cex=.65, las=0) # y-axis label
884     mtext('Controlled shock', side=1, line=2.0, cex=.65, las=1) # x-axis label
885
886     text(left+(right-left)*.1, bottom+(top-bottom)*.9, '(A)', cex=.8)
887
888     dev.off()
889 }
890
891 # NB=0 ISOQUANT rho.hi vs threshold:
892 if(1==1){
893     cat('Working on rho.hi-threshold isoquant\n')
894     nodes <- 30
895     rho.hi.0 <- rho.hi; rho.hi.range <- seq(0, .05, .05/(nodes-1))
896     threshold.0 <- threshold; threshold.range <- seq(0, threshold.0*2, threshold.0*2/(nodes-1))
897     NBiso <- matrix(0, length(threshold.range), length(rho.hi.range))
898     for(rho.hi in rho.hi.range){
899         for(threshold in threshold.range){
900             parms <- parms0
901             parms$rho.hi <- rho.hi
902             parms$threshold <- threshold
903             outs <- scenario.fn(parms, 0, 0)
904             D1 <- outs[[5]]
905             D2 <- outs[[6]]
906             PV.1 <- outs[[8]]
907             PV.2 <- outs[[9]]
908             NBiso[which(threshold.range==threshold), which(rho.hi.range==rho.hi)] <-
909                 PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
910         }
911     }
912
913     # Save figure:
914     tikz(paste(output.path, '/corona_fig_iso_2.tex', sep=''),
915          width = 3,
916          height = 3,
917          pointsize = 12,
918          standAlone = TRUE)
919
920     par(mar=c(3.5,3.5,1,1)) # figure border whitespace [bottom,left,top,right]
921
922     left <- min(threshold.range)
923     right <- max(threshold.range)
924     top <- max(rho.hi.range)
925     bottom <- min(rho.hi.range)
926

```

```

927     contour(threshold.range, rho.hi.range, NBiso, levels=0,
928             drawlabels=FALSE,
929             xlab='',
930             ylab='',
931             axes=FALSE,
932             cex.axis=.8,
933             cex.lab=.5)
934
935     points (parms0$threshold, parms0$rho.hi, pch=1, cex=.5)
936
937     axis(1, las=1, labels=TRUE, cex.axis=.65) # Draw x axis
938     axis(2, las=1, labels=TRUE, cex.axis=.65) # Draw y axis
939
940     mtext('Mortality rate above threshold', side=2, line=2.2, cex=.65, las=0) # y-axis label
941     mtext('Heath care system threshold', side=1, line=2.0, cex=.65, las=1) # x-axis label
942
943     text(left+(right-left)*.1, bottom+(top-bottom)*.9, '(B)', cex=.8)
944
945     dev.off()
946 }
947
948 # NB=0 ISOQUANT R0 vs sd:
949 if(1==1){
950     cat('Working on R0-sd isoquant\n')
951     nodes <- 50
952     R0.0 <- R0; R0.range <- seq(1, 7, (7-1)/(nodes-1))
953     sd.0 <- sd; sd.range <- seq(0, 1, (1-0)/(nodes-1))
954     NBiso <- matrix(0, length(sd.range), length(R0.range))
955     for(R0 in R0.range){
956         for(sd in sd.range){
957             parms <- parms0
958             parms$R0 <- R0
959             parms$sd <- sd
960             outs <- scenario.fn(parms, 0, 0)
961             D1 <- outs[[5]]
962             D2 <- outs[[6]]
963             PV.1 <- outs[[8]]
964             PV.2 <- outs[[9]]
965             NBiso[which(sd.range==sd), which(R0.range==R0)] <-
966                 PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
967         }
968     }
969
970     # Save figure:
971     tikz(paste(output.path, '/corona_fig_iso_3.tex', sep=''),
972          width = 3,
973          height = 3,
974          pointsize = 12,
975          standAlone = TRUE)
976
977     par(mar=c(3.5, 3, 1, 1)) # figure border whitespace [bottom, left, top, right]
978
979     bottom <- min(R0.range)
980     top <- max(R0.range)
981     right <- max(sd.range)
982     left <- min(sd.range)
983
984     contour(sd.range, R0.range, NBiso, levels=0,
985             drawlabels=FALSE,
986             xlab='',
987             ylab='',
988             axes=FALSE,
989             cex.axis=.8,
990             cex.lab=.5)
991
992     points (parms0$sd, parms0$R0, pch=1, cex=.5)
993
994     axis(1, las=1, labels=TRUE, cex.axis=.65) # Draw x axis
995     axis(2, las=1, labels=TRUE, cex.axis=.65) # Draw y axis

```

```

996
997     mtext('R$_0$',
          label
          side=2, line=2.0, cex=.65, las=0) # y-axis
998     mtext('Social distancing effectiveness',side=1, line=2.0, cex=.65, las=1) # x-axis
          label
999
1000     text(left+(right-left)*.1,bottom+(top-bottom)*.9, '(C)', cex=.8)
1001
1002     dev.off()
1003 }
1004
1005 # NB=0 ISOQUANT R0 vs rho.hi:
1006 if(1==1){
1007     cat('Working on R0-rho.hi isoquant\n')
1008     nodes <- 50
1009     NBiso <- matrix(0,length(rho.hi.range),length(R0.range))
1010     for(R0 in R0.range){
1011         for(rho.hi in rho.hi.range){
1012             parms <- parms0
1013             parms$R0 <- R0
1014             parms$rho.hi <- rho.hi
1015             outs <- scenario.fn(parms,0,0)
1016             D1 <- outs[[5]]
1017             D2 <- outs[[6]]
1018             PV.1 <- outs[[8]]
1019             PV.2 <- outs[[9]]
1020             NBiso[which(rho.hi.range==rho.hi),which(R0.range==R0)] <-
1021                 PV.2-PV.1+parms0$VSL*sum(D1)-parms0$VSL*sum(D2)
1022         }
1023     }
1024
1025     # Save figure:
1026     tikz(paste(output.path, '/corona_fig_iso_4.tex', sep=''),
1027          width = 3,
1028          height = 3,
1029          pointsize = 12,
1030          standAlone = TRUE)
1031
1032     par(mar=c(3.5,3,1,1)) # figure border whitespace [bottom,left,top,right]
1033
1034     bottom <- min(R0.range)
1035     top <- max(R0.range)
1036     right <- max(rho.hi.range)
1037     left <- min(rho.hi.range)
1038
1039     contour(rho.hi.range,R0.range,NBiso,levels=0,
1040            drawlabels=FALSE,
1041            xlab='',
1042            ylab='',
1043            axes=FALSE,
1044            cex.axis=.8,
1045            cex.lab=.5)
1046
1047     points(parms0$rho.hi,parms0$R0,pch=1,cex=.5)
1048
1049     axis(1,las=1,labels=TRUE,cex.axis=.65) # Draw x axis
1050     axis(2,las=1,labels=TRUE,cex.axis=.65) # Draw y axis
1051
1052     mtext('R$_0$',
          label
          side=2, line=2.0, cex=.65, las=0) # y-axis
1053     mtext('Mortality rate above threshold',side=1, line=2.0, cex=.65, las=1) # x-axis label
1054
1055     text(left+(right-left)*.1,bottom+(top-bottom)*.9, '(D)', cex=.8)
1056
1057     dev.off()
1058 }
1059
1060
1061

```