

Appendix 1 for **Impact of Ebola disease progression on transmission and control in Liberia**

Dan Yamin, PhD^{1*}, Shai Gertler^{1*}, Martial L. Ndeffo-Mbah, PhD^{1*}, Laura A. Skrip, MPH¹,
Mosoka Fallah, PhD², Tolbert G. Nyenswah, MPH², Frederick L. Altice, MD MA^{3,4}, & Alison P.
Galvani, PhD^{1,4}

¹ Center for Infectious Disease Modeling, Yale School of Public Health, New Haven, CT, USA

² Ministry of Health and Social Welfare, 1000 Monrovia 10, Liberia

³ Section of Infectious Diseases, Yale University School of Medicine, New Haven, CT, USA

⁴ Department of Epidemiology of Microbial Diseases, Yale School of Public Health, New Haven, CT, USA

* Authors contributed equally

Data analysis

Our data on Montserrado County, Liberia, was obtained from the Ministry of Health and Social Welfare (MoHSW). They included aggregated cumulative cases (13) as well as contact tracing data (Appendix 1 Figure 1). The data differentiates between three Ebola diagnostics: ‘confirmed’, ‘probable’, and ‘suspected’ cases. Cases are considered ‘confirmed’ if they were laboratory confirmed with a PCR test, ‘probable’ if Ebola-specific symptoms were diagnosed clinically, ‘suspected’ when there has been known contact with a ‘probable’ or ‘confirmed’ case, and in the presence of sudden high fever or three clinical symptoms of Ebola (32). Generally, these data are provided on a daily basis on weekdays, but not on weekends. The data also indicate the number of newly reported Ebola-related deaths. The contact tracing data are provided on a daily basis at an individual level that includes the date on which a case was reported as well as the dates of fatalities. For our modeling analyses, we considered any of the three categories to be a case of Ebola because under-reporting is likely. In addition, Pearson correlation between the cumulative number of ‘probable’, ‘confirmed’ or ‘suspected’ cases

observed in both data sources was at least 97.5%. To determine the number of cases on a daily basis, we applied a linear interpolation to the cumulative incidence.

Please insert Appendix 1 Figure 1 (App1Fig1.jpg)

Primary contact tracing data in Montserrado County was provided by the Liberian Ministry of Health and Social Welfare between August 7th and August 26th. The data specifies the contacts of 246 infected individuals, among whom 113 were dead, traced since symptom onset. Over this timeframe, for each infected individual i we used the number of contacts, C_i , as provided in the data, excluding one report of a person who declared 72 contacts, which is more than 10 standard deviations higher than the average. This contact data was also used to construct a probability distribution of the contacts per infected individual, $f(C)$, (Appendix 1 Figure 2). Outside To extrapolate beyond the timeframe of the contact-tracing data, we sampled the contact of infected individuals from this distribution, $C_i \sim f(C)$. For non-survivors during the late phase of infection, the number of contacts was drawn from the distribution, but conservatively restricted between one and $\min(C_i, 5)$, as we assumed that during the late phase individuals are only in contact with household members (See Methods, Main text).

Please insert Appendix 1 Figure 2 (App1Fig2.jpg)

Estimation of the rate ratio of transmission risk given contact

To evaluate the daily infectiousness of an infected individual given contact assumptions, we integrated daily measures of VL through disease progression. The VL estimates used were based on daily RNA copy levels detected among survivors and non-survivors per day of infection during the 2000 - 2001 Ebola outbreak in Uganda (8). Consistent with previous studies (16,17),

we assumed that for any given contact, each ten-fold increase in VL leads to an r -fold increase in infectiousness, i.e., $g(V^s(t)) = r^{\text{Log}(V^s(t))}$. We evaluated values of r based on Dowell et al. (9) who assessed the relative risk of becoming infected given three types of common contacts: 1) conversation, 2) sharing a meal, and 3) sharing a bed. This study on the 1995 DRC Ebola outbreak surveyed 173 individuals who had contacts with 27 Ebola patients, and specifies the relative risk of infection and 95% confidence intervals given contacts with Ebola patients in their early phase and late phase of infection (9).

For a specific contact, the ratio between the relative risk of infection in the early phase and the late phase is given by:

$$\frac{RR^{Late\ phase}}{RR^{Early\ phase}} = \frac{r^{\frac{1}{(\tau_2 - \tau_1 - 1)} \sum_{t=\tau_1+1}^{\tau_2} \log[V(t)]}}{r^{\frac{1}{(\tau_1 - \tau_0)} \sum_{t=\tau_0}^{\tau_1} \log[V(t)]}}, \quad (1)$$

where RR represents the relative risk of infection, τ_0 represents the first day of symptom onset, τ_1 the last day of the early phase, and τ_2 the last day of the late phase. Rearranging,

$$r = \left(\frac{RR^{Late\ phase}}{RR^{Early\ phase}} \right)^{\left(\frac{1}{(\tau_2 - \tau_1 - 1)} \sum_{t=\tau_1+1}^{\tau_2} \log[V(t)] - \frac{1}{(\tau_1 - \tau_0)} \sum_{t=\tau_0}^{\tau_1} \log[V(t)] \right)^{-1}}. \quad (2)$$

For each type of contact, we sampled values for $RR^{Late\ Phase} / RR^{Early\ Phase}$ (9), and used Eq. (2) to estimate the distribution of r -fold increase in infectiousness for each ten-fold increase in VL (Appendix 1 Figure 3). In our analysis, we employed only samples that satisfied $RR^{Late\ Phase} > RR^{Early\ Phase} \geq 1$. This approach ensures that only realistic values are considered for the evaluation of r . Namely, an individual who had contact with a patient in the

late Ebola phase would have a higher risk of becoming infected than if he had a similar contact with a patient in the early Ebola phase.

We compared three different scenarios, which correspond to the three types of contact.

Our analysis suggests that a less close type of contact (i.e., sharing a conversation rather than sharing a meal), resulted in higher r estimates. While all individuals are more likely to transmit for closer types of contacts, the differential in risk of transmitting between individuals with relatively high VL and low VL decreases with the closeness of the contact, from conversation to sharing a meal to sharing a bed.

Please insert Appendix 1 Figure 3 (App1Fig3.jpg)

Additionally and conservatively, we did not assume that the risk of infection would increase with clinical symptoms such as bleeding or vomiting, which is more likely to occur during the late phase of infection.

Simulation studies

For every infected individual recorded in our data set, we independently sampled the incubation duration, the infection period, and early phase duration. In each iteration, we stored a vector that tracked the number of infectious individuals (Appendix 1 Figure 4, shown in yellow) and the number of individuals exposed to Ebola per day (Appendix 1 Figure 4, shown in blue). We also sampled the daily VL and the number of contacts in early and late phase infection (Main text, Table 1). For any given day, we evaluated the probability that a newly infected individual could have been infected by any of the currently infectious individuals (Main text, Eq. (1)). Based on these probabilities we assigned the source of infection for each exposed individual.

As of October 2, according to the US CDC (30), there were 3696 reported cases and 1998 reported deaths in Liberia. Calculating case fatality by dividing the number of deaths reported to date by the number of cases reported to date (i.e. ~54%), will be biased downwards, because a substantial proportion of the unresolved cases reported will still progress to fatalities. Following discussion with contact tracers, and consistent with a recent study (20), the reported fatality date of non-survivors represents the day of death while for survivors the date of case is typically reported the day after symptom onset, around which we consider uncertainty by varying incubation duration, and infectious duration (Main text, Table 1). Further, the actual number of infected individuals beyond the end date of our contact-tracing data (i.e., September 22nd) is unknown. Thus, in each iteration, we used only cases whose entire infectious period, as obtained from the stochastic process of transmission (Appendix 1, Figure 4), was within the model timeframe. Taking this adjustment into consideration, case fatality as obtained from the model was estimated to be 63% (95% CI 60-64%).

Please insert Appendix 1 Figure 4 (App1Fig4.jpg)

Effectiveness of intervention strategies

Due to logistical challenges in case detection (24) and shortage of isolation units, we evaluated the probability of achieving disease elimination from case isolation of 50% to 100% of the non-survivors compared with of all infected individuals, when the type of contact is dominated through conversation or by sharing a meal or by sharing a bed (Appendix 1, Figure 5). Our results indicate that isolation of 70-100% within four days from onset of their symptoms has a high chance of achieving disease elimination for all types of contacts.

Please insert Appendix 1 Figure 5 (App1Fig5.jpg)

Additionally, we assessed a more pragmatic strategy that included self-quarantine among 50-100% of the infected, considering contact reduction from symptom onset that varied between 0 to 100% (Appendix 1 Figure 6). Consistent with our main finding, our results indicate that a reduction in contacts of 60%-100% is likely to curtail transmission.

Please insert Appendix 1 Figure 6 (App1Fig6.jpg)

Appendix 2 for **Impact of Ebola disease progression on transmission and control in Liberia**

Dan Yamin, PhD^{1*}, Shai Gertler^{1*}, Martial L. Ndeffo-Mbah, PhD^{1*}, Laura A. Skrip, MPH¹,
Mosoka Fallah, PhD², Tolbert G. Nyenswah, MPH², Frederick L. Altice, MD MA^{3,4}, & Alison
P. Galvani, PhD^{1,4}

¹ Center for Infectious Disease Modeling, Yale School of Public Health, New Haven, CT, USA

² Ministry of Health and Social Welfare, 1000 Monrovia 10, Liberia

³ Section of Infectious Diseases, Yale University School of Medicine, New Haven, CT, USA

⁴ Department of Epidemiology of Microbial Diseases, Yale School of Public Health, New Haven, CT, USA

* Authors contributed equally

Model algorithm

Constructing the matrix

For each individual, we defined a row vector comprising of the exposure day, an incubation period and the symptoms period. Every element in the vector represents a single day. The date of exposure was marked with a " - 1", and the symptoms period was assigned the values of the relative infectiousness, $f_i^S(t)$. An example of such a vector:

(-1, 0, 0, 0, 0, 0, 0, 0, 0, 208.14, 559.764, 1169.51, 1964.58, 2800.02, 1804.77, 1736.56, 1201.36, 352.745)

Visually, this may be represented:

Please insert Appendix 2 Figure 1 (App2Fig1.jpg)

This vector was then embedded onto the time axis using the dates of the symptomatic periods, according to the data. The relative infectiousness is a function of the viral load, $g(V_i^S(t))$, combined with the individuals' contacts, $C_i(t)$, and is presented in the Main text, Eq. (2).

If the number of contacts of an individual i was known, it was used to parameterize C_i . When the number of contacts was unknown, we sampled from a probability distribution, $f(C)$, constructed from the contact-tracing data: $C_i \sim f(C)$, (Appendix 1, Figure 2).

In each iteration of the stochastic simulation, we sampled values for $RR^{Late Phase} / RR^{Early Phase}$. Using Eq. (2) in Appendix 1, we calculated r , and from the relation $g(V^S(t)) = r^{Log(V^S(t))}$, we assigned each individual with their viral load contribution to the relative infectiousness. In the stochastic simulation, the variables were sampled from distributions (Main text, Table 1). A vector was assigned to each individual, and embedding the vectors in time yielded a matrix (Appendix 1, Figure 4).

Assigning transmission

Using the above matrix, each exposed case was probabilistically assigned with the contact from whom they were infected. In each day, the numbers of exposed and symptomatic individuals were in the same column in the matrix. Symptomatic individuals were considered potential infectors, and were probabilistically assigned as the source of transmission to their infected contact based on their relative infectiousness, $f_i^S(t)$.

Please insert Appendix 2 Figure 2 (App2Fig2.jpg)

We used the relative infectiousness to determine probabilities of being the infector, normalizing over all possible infectors of that day (Main text, Equation (1)).

Testing intervention strategies

In every iteration the overall R_0 was calculated, and at the end of each of the simulations, the probability of disease elimination was calculated:

$$Pr(R_0 < 1) = \frac{N^{(R_0 < 1)}}{N}, \quad (1)$$

where $N^{(R_0 < 1)}$ denotes the number of iterations in which $R_0 < 1$ was obtained, and N is the overall number of iterations.

Quarantine of infected individuals

Two parameters were varied over this analysis: 1) number of days that have passed since symptoms onset and until recovery/death. 2) the coverage of population on which the policy is successfully implemented.

For every iteration, upon successful quarantine of individuals, all of the individuals' vectors were set to zero, and the R_0 was calculated (Main text, equation (3)). This procedure was repeated for each day of the timeframe considered (Appendix 1, Figure 5A). To vary the coverage of quarantine, the corresponding proportion of the population's individuals' vectors were set to zero. This was repeated with different values of coverage, implemented by sampling a random

number, $r \sim \text{Uniform}(0, 1)$, and setting to zero the individuals' vector only if $r < C$, with C representing the tested coverage, varying from half the population ($C = 0.5$) to full coverage ($C = 1$). This stochastic process was conducted separately for the non-survivors (Main text, Figure 2A) and for all infected individuals (Appendix 1, Figure 5C).

Reducing number of contacts

Reducing the contacts of the infected individuals was implemented by decreasing the probability of being selected as the infector. Eq. (1) in the Main text transforms into:

$$\hat{P}_{i,j}(t) = I_j(t) \frac{\gamma f_i^S(t)}{\gamma \sum_{k=1}^n f_k^S(t) + (1 - \gamma)}, \quad (2)$$

where γ is the proportion of contacts, $0 < \gamma < 1$, with $\gamma = 1$ implying no intervention, and $\gamma = 0$ is complete isolation of the individual from all contacts. This re-normalization lowers the chance of an individual to be chosen as the infector, which is effectively the same as reducing the number of contacts (Eq. (2), Main text).

Mathematica 9.0 code

```
Impact of Ebola disease progression on transmission and control in Liberia

Initialization
init[]:=
(
SetDirectory[NotebookDirectory[]];
days=DateRange[{2014,5,1},{2014,12,1}];
)(*init*)

Data
(*In "id" vector, "ext" represents cases from WHO. Numbers are coded cases collected by LMoHSW*)

id={"ext1","ext2","ext76","ext4","ext5","ext82","ext11","ext12","ext13","ext14","ext88","ext148",
"ext15","ext89","ext16","ext17","ext91","ext18","ext92","ext149","ext19","ext93","ext20","ext94",
"ext150","ext21","ext95","ext151","ext22","ext96","ext23","ext152","ext153","ext154","ext24","ext
27","ext102","ext28","ext103","ext155","ext29","ext30","ext31","ext106","ext156","ext32","ext107"
,"ext157","ext33","ext108","ext158","ext34","ext109","ext35","ext110","ext36","ext111","ext37","e
xt112","ext38","ext113","ext39","ext114","ext40","ext115","ext159","ext41","ext116","ext160","ext
```

161", "ext42", "ext117", "ext162", "ext163", "ext164", "ext43", "ext118", "ext165", "ext166", "ext167", "ext44", "ext119", "ext168", "ext169", "ext170", "ext45", "ext120", "ext171", "ext46", "ext121", "ext47", "ext122", "ext172", "ext48", "ext123", "ext173", "ext49", "ext124", "ext174", "ext175", "ext176", "ext50", "ext125", "ext177", "ext178", "ext179", "ext51", "ext126", "ext180", "ext181", "ext182", "ext183", "ext52", "ext127", "ext184", "ext185", "ext186", "ext187", "ext53", "ext128", "ext188", "ext189", "ext190", "ext191", "ext54", "ext129", "ext55", "ext130", "ext192", "ext56", "ext131", "ext193", "ext194", "ext195", "ext196", "ext197", "ext198", "ext199", "ext57", "311.", "ext132", "ext200", "ext201", "ext202", "ext203", "ext204", "ext205", "ext206", "ext58", "1.", "2.", "134.", "217.", "225.", "254.", "257.", "265.", "307.", "136.", "203.", "219.", "297.", "298.", "86.", "230.", "250.", "251.", "256.", "271.", "306.", "259.", "264.", "267.", "272.", "273.", "274.", "277.", "294.", "302.", "310.", "252.", "255.", "261.", "266.", "295.", "170.", "209.", "212.", "253.", "263.", "269.", "282.", "283.", "284.", "285.", "286.", "289.", "290.", "194.", "206.", "226.", "228.", "233.", "234.", "237.", "262.", "268.", "270.", "288.", "2565.", "185.", "208.", "229.", "235.", "236.", "275.", "87.", "89.", "91.", "147.", "178.", "181.", "186.", "187.", "190.", "196.", "211.", "216.", "222.", "26.", "98.", "99.", "111.", "115.", "120.", "139.", "142.", "163.", "164.", "165.", "169.", "172.", "174.", "184.", "192.", "193.", "8.", "29.", "61.", "83.", "95.", "97.", "155.", "166.", "179.", "180.", "200.", "204.", "205.", "207.", "28.", "101.", "103.", "109.", "117.", "173.", "188.", "199.", "201.", "202.", "77.", "104.", "116.", "145.", "148.", "157.", "158.", "168.", "171.", "175.", "177.", "182.", "183.", "195.", "197.", "198.", "41.", "62.", "113.", "140.", "144.", "146.", "149.", "150.", "151.", "156.", "161.", "162.", "167.", "27.", "31.", "44.", "106.", "121.", "125.", "128.", "130.", "131.", "132.", "133.", "138.", "143.", "1092.", "1093.", "22.", "47.", "49.", "58.", "59.", "60.", "63.", "64.", "67.", "82.", "88.", "90.", "92.", "96.", "100.", "102.", "108.", "112.", "114.", "119.", "127.", "141.", "7.", "42.", "43.", "46.", "52.", "54.", "56.", "57.", "70.", "71.", "74.", "75.", "78.", "84.", "85.", "93.", "105.", "107.", "110.", "118.", "159.", "2566.", "12.", "39.", "48.", "50.", "65.", "68.", "69.", "72.", "76.", "79.", "80.", "81.", "11.", "13.", "21.", "33.", "34.", "45.", "55.", "73.", "9.", "10.", "14.", "15.", "16.", "17.", "18.", "19.", "24.", "25.", "32.", "35.", "36.", "37.", "38.", "40.", "23.", "ext133", "ext507", "ext508", "ext509", "ext510", "ext511", "ext512", "ext513", "ext514", "ext59", "ext555", "ext556", "ext557", "ext558", "ext559", "ext560", "ext561", "ext562", "ext563", "ext5134", "ext515", "ext516", "ext517", "ext518", "ext519", "ext520", "ext521", "ext522", "ext60", "ext564", "ext565", "ext566", "ext567", "ext568", "ext569", "ext570", "ext571", "ext572", "ext573", "ext523", "ext524", "ext525", "ext526", "ext527", "ext528", "ext529", "ext530", "ext61", "ext573", "ext574", "ext575", "ext576", "ext577", "ext578", "ext579", "ext580", "ext581", "ext136", "ext531", "ext532", "ext533", "ext534", "ext535", "ext536", "ext537", "ext538", "ext62", "ext582", "ext583", "ext584", "ext585", "ext586", "ext587", "ext588", "ext589", "ext590", "ext591", "ext137", "ext539", "ext540", "ext541", "ext542", "ext543", "ext544", "ext545", "ext546", "ext63", "ext592", "ext593", "ext594", "ext595", "ext596", "ext597", "ext598", "ext599", "ext600", "ext601", "ext138", "ext547", "ext548", "ext549", "ext550", "ext551", "ext552", "ext553", "ext554", "ext64", "ext602", "ext603", "ext604", "ext605", "ext606", "ext607", "ext608", "ext609", "ext610", "ext611", "ext139", "ext207", "ext208", "ext209", "ext210", "ext211", "ext212", "ext213", "ext214", "ext215", "ext216", "ext217", "ext218", "ext219", "ext220", "ext221", "ext222", "ext223", "ext224", "ext225", "ext226", "ext227", "ext228", "ext229", "ext230", "ext231", "ext232", "ext233", "ext234", "ext235", "ext236", "ext237", "ext238", "ext239", "ext65", "ext240", "ext241", "ext242", "ext243", "ext244", "ext245", "ext246", "ext66", "ext247", "ext248", "ext249", "ext250", "ext251", "ext252", "ext253", "ext254", "ext255", "ext256", "ext67", "ext257", "ext258", "ext259", "ext260", "ext261", "ext262", "ext263", "ext264", "ext265", "ext266", "ext267", "ext268", "ext269", "ext270", "ext271", "ext272", "ext68", "ext143", "ext273", "ext274", "ext275", "ext276", "ext277", "ext278", "ext279", "ext280", "ext281", "ext282", "ext283", "ext284", "ext285", "ext286", "ext287", "ext288", "ext289", "ext290", "ext69", "ext291", "ext292", "ext293", "ext294", "ext70", "ext145", "ext295", "ext296", "ext297", "ext298", "ext299", "ext300", "ext301", "ext302", "ext303", "ext304", "ext305", "ext306", "ext307", "ext308", "ext309", "ext310", "ext311", "ext312", "ext313", "ext314", "ext315", "ext316", "ext317", "ext318", "ext319", "ext320", "ext321", "ext322", "ext323", "ext324", "ext325", "ext326", "ext327", "ext328", "ext329", "ext330", "ext331", "ext332", "ext333", "ext334", "ext335", "ext336", "ext337", "ext338", "ext339", "ext340", "ext341", "ext342", "ext343", "ext344", "ext345", "ext346", "ext347", "ext348", "ext349", "ext350", "ext351", "ext352", "ext353", "ext354", "ext355", "ext356", "ext357", "ext358", "ext359", "ext360", "ext361", "ext362", "ext71", "ext146", "ext363", "ext364", "ext365", "ext366", "ext367", "ext368", "ext369", "ext370", "ext371", "ext372", "ext373", "ext374", "ext375", "ext376", "ext377", "ext378", "ext379", "ext380", "ext381", "ext382", "ext383", "ext384", "ext385", "ext386", "ext387", "ext388", "ext389", "ext390", "ext391", "ext392", "ext393", "ext394", "ext395", "ext396", "ext397", "ext398", "ext399", "ext400", "ext401", "ext402", "ext403", "ext404", "ext405", "ext406", "ext407", "ext408", "ext409", "ext410", "ext411", "ext412", "ext413", "ext414", "ext415", "ext416", "ext417", "ext418", "ext419", "ext420", "ext421", "ext422", "ext423", "ext424", "ext425", "ext426", "ext427", "ext428", "ext429", "ext430", "ext431", "ext432", "ext72", "ext147", "ext433", "ext434", "ext435", "ext436", "ext437", "ext438", "ext439", "ext440", "ext441", "ext442", "ext443", "ext444", "ext445", "ext446", "ext447", "ext448", "ext449", "ext450", "ext451", "ext452", "ext453", "ext454", "ext455", "ext456", "ext457", "ext458", "ext459", "ext460", "ext461", "ext462", "ext463", "ext464", "ext465", "ext466", "ext467", "ext468", "ext469", "ext470", "ext471", "ext472", "ext473", "ext474", "ext475", "ext476", "ext477", "ext478", "ext479", "ext480", "ext481", "ext482", "ext483", "ext484", "ext485", "ext486", "ext487", "ext488", "ext489", "ext490", "ext491", "ext492", "ext493", "ext494", "ext495", "ext496", "ext497", "ext498", "ext499", "ext500", "ext501", "ext502", "ext503", "ext504", "ext505", "ext506", "ext73", "ext612", "ext613", "ext614", "ext615", "ext616", "ext617", "ext618", "ext619", "ext620", "ext621", "ext622", "ext623", "ext624", "ext625", "ext626", "ext627", "ext628", "ext629", "ext630", "ext631", "ext632", "ext633", "ext634", "ext635", "ext636", "ext637", "ext638", "ext639", "ext640", "ext641", "ext642", "ext643", "ext644", "ext645", "ext646", "ext647", "ext648", "ext649", "ext650", "ext651", "ext652", "ext653", "ext654", "ext655", "ext656", "ext657", "ext658", "ext659", "ext660", "ext661", "ext662", "ext663", "ext664", "ext665", "ext666", "ext667", "ext668", "ext669", "ext670", "ext671", "ext672", "ext673", "ext674", "ext675", "ext676", "ext677", "ext678", "ext679", "ext680", "ext681", "ext682", "ext683", "ext684", "ext685", "ext686", "ext687", "ext688", "ext689", "ext690", "ext691", "ext692", "ext693", "ext694", "ext695", "ext696", "ext697", "ext698", "ext699", "ext700", "ext701", "ext702", "ext703", "ext704", "ext705", "ext706", "ext707", "ext708", "ext709", "ext710", "ext711", "ext712", "ext713", "ext714", "ext715", "ext7


```

nonFatalSD={0.4,1,0.4,0.6,0.4,0.5,0.8,0.7,0.7,0.2,0.1,0.1,0.1,0.1};

Constructing "disease vector"
constructSickVector[]:=
(
scaledViralLoadDead=viralLoadDead[[1;;sickLengthDead]];
scaledViralLoadAlive=viralLoadAlive[[1;;sickLengthAlive]];

contactCalibrationDead=Table[1,{i,1,sickLengthDead}];
contactCalibrationDead[[1;;sickLengthDead-stageCutoff]]=contactFirstStageDead;
contactCalibrationDead[[sickLengthDead-stageCutoff+1;;sickLengthDead]]=contactSecondStageDead;

contactCalibrationAlive=Table[1,{i,1,sickLengthAlive}];
contactCalibrationAlive[[1;;sickLengthDead-stageCutoff-1]]=contactFirstStageAlive;
contactCalibrationAlive[[sickLengthDead-stageCutoff;;sickLengthDead]]=contactSecondStageAlive;

sickVectorDead=Join[{-
1},Table[0,{i,1,incubationDead}],scaledViralLoadDead*contactCalibrationDead];
sickVectorAlive=Join[{-
1},Table[0,{i,1,incubationAlive}],scaledViralLoadAlive*contactCalibrationAlive];
)(*constructSickVector*)

Constructing matrix
constructMatrix[]:=
(
ebolaMatrix =
Table[0,{i,1,Length[infectionDate]},{j,1,Length[days]+maxIncubationPeriod+maxSickPeriod+1}];
(*Initialize zero matrix*)
symptomDateIndex=Table[0,{i,1,Length[infectionDate]}]; (*For later use*)
For[i=1,i<=Length[infectionDate],i++,
index=Position[days,infectionDate[[i]]][[1,1]];
If[status[[i]]=="Dead",
ebolaMatrix[[i]][[index+2-Length[sickVectorsForMatrix[[i]]];;index+1]]=sickVectorsForMatrix[[i]];
symptomDateIndex[[i]]=index+2-sickLengthForMatrix[[i]],
ebolaMatrix[[i]][[index-2-incubationAliveForMatrix[[i]]];;index-
2+sickLengthForMatrix[[i]]]=sickVectorsForMatrix[[i]];
symptomDateIndex[[i]]=index-1;
];
];

infectedByDay=Table[0,{i,1,Length[days]}]; (*Overall infected every day*)
For[j=1,j<=Length[days],j++,
infectedByDay[[j]]=-1*Total[Cases[Transpose[ebolaMatrix][[j]],x_/;x<0]];
];

rMatrixTEMP=ebolaMatrix; (*Zeroizing negative elements*)
For[i=1,i<=Length[ebolaMatrix],i++,
For[j=1,j<=Length[ebolaMatrix[[1]]],j++,
If[ebolaMatrix[[i,j]]<0,
rMatrixTEMP[[i,j]]=0
];
];
];

rMatrix= Table[0,{i,1,Length[infectionDate]},{j,1,Length[days]}]; (*Initialize zero matrix*)
For[j=1,j<=Length[days],j++,
If[Count[Transpose[rMatrixTEMP][[j]],x_/;x>0]==0,
Return,
If[coverage!= 0&&RandomReal[]<coverage,
probSelected=RandomChoice[Join[{{1-
propContacts}},propContacts/Total[Transpose[rMatrixTEMP][[j]]]*Transpose[rMatrixTEMP][[j]]]-
>Join[{-1},Range[Length[Transpose[rMatrixTEMP][[j]]]],infectedByDay[[j]]],
probSelected=RandomChoice[Transpose[rMatrixTEMP][[j]]-
>Range[Length[Transpose[rMatrixTEMP][[j]]]],infectedByDay[[j]]];
];
For[k=1,k<=Length[probSelected],k++,
If[probSelected[[k]]==-1,
Return,
rMatrix[[probSelected[[k]],j]]+=1
];
];
];

```



```

];
];

cropTop=1;
cropBottom=Length[ebolaMatrix];(*Counted from top*)
cropLeft=Min[Table[Position[ebolaMatrix[[row]],-1],{row,1,Length[id]}]];
cropRight=Max[Table[Position[ebolaMatrix[[row]],-1],{row,1,Length[id]}]];(*Counted from left*)

rMatrixCrop=rMatrix[[cropTop;;cropBottom,cropLeft;;cropRight]];

rPerDayAll={};
rPerDayDead={};
rPerDayAlive={};
rDead={};
rAlive={};

For[i=cropTop,i<=cropBottom,i++,
If[cropLeft<= symptomDateIndex[[i]]&&symptomDateIndex[[i]]+sickLengthForMatrix[[i]]<=cropRight,
rPerDayAll=Insert[rPerDayAll,rMatrix[[i,symptomDateIndex[[i]];;symptomDateIndex[[i]]+sickLengthFo
rMatrix[[i]-1]],-1];
If[status[[i]]=="Dead",
rPerDayDead=Insert[rPerDayDead,rMatrix[[i,symptomDateIndex[[i]];;symptomDateIndex[[i]]+sickLength
ForMatrix[[i]-1]],-1];
rDead=Join[rDead,{Total[rMatrix[[i,symptomDateIndex[[i]];;symptomDateIndex[[i]]+sickLengthForMatr
ix[[i]-1]]}],
rPerDayAlive=Insert[rPerDayAlive,rMatrix[[i,symptomDateIndex[[i]];;symptomDateIndex[[i]]+sickLeng
thForMatrix[[i]-1]],-1];
rAlive=Join[rAlive,{Total[rMatrix[[i,symptomDateIndex[[i]];;symptomDateIndex[[i]]+sickLengthForMa
trix[[i]-1]]}],
];
];
];

rAll=Join[rAlive,rDead];
rPerDayAll=Flatten[rPerDayAll,{2}];
rPerDayDead=Flatten[rPerDayDead,{2}];
rPerDayAlive=Flatten[rPerDayAlive,{2}];
)(*constructMatrix*)
Constructing "visual" matrix
constructVisualMatrix[]:=
(
visualEbolaMatrixTEMP=ebolaMatrix[[cropTop;;cropBottom,cropLeft;;cropRight+maxIncubationPeriod+ma
xSickPeriod]];

(*indexOfPeopleInfected=Table[0,{day,1,Length[visualEbolaMatrixTEMP[[1]]]}];*)
visualEbolaMatrix=Table[0,{day,1,Length[visualEbolaMatrixTEMP[[1]]]}];

For[day=1,day<=Length[visualEbolaMatrixTEMP[[1]],day++,
indexOfPeopleInfected=Flatten[Position[Transpose[visualEbolaMatrixTEMP][[day]],-1]];
visualEbolaMatrix[[day]]=Total[visualEbolaMatrixTEMP[[indexOfPeopleInfected]]];
If[visualEbolaMatrix[[day]]==0,
visualEbolaMatrix[[day]]=Table[0,{i,1,Length[visualEbolaMatrixTEMP[[1]]]}];
];
];

cutBottom=Length[visualEbolaMatrix];
While[visualEbolaMatrix[[cutBottom]]== Table[0,{r,1,Length[visualEbolaMatrix[[cutBottom]]]}],
cutBottom-=1
];

cutRight=Length[Transpose[visualEbolaMatrix]];
While[Transpose[visualEbolaMatrix][[cutRight]]==
Table[0,{r,1,Length[Transpose[visualEbolaMatrix][[cutRight]]]}],
cutRight-=1
];

visualEbolaMatrixCropped=visualEbolaMatrix[[1;;cutBottom,1;;cutRight]];
plotVisualMatrix=MatrixPlot[visualEbolaMatrixCropped,ImageSize->Large];

totalVisualInfections=Table[0,{j,1,Length[Transpose[visualEbolaMatrixCropped]]}];
totalVisualViralLoad=Table[0,{j,1,Length[Transpose[visualEbolaMatrixCropped]]}];

```

```

For[j=1,j<=Length[Transpose[visualEbolaMatrixCropped]],j++,

totalVisualInfections[[j]]=Total[Cases[Transpose[visualEbolaMatrixCropped][[j]],x_/;x<0]];
totalVisualViralLoad[[j]]=Total[Cases[Transpose[visualEbolaMatrixCropped][[j]],x_/;0<x]];
];

visualLines={totalVisualViralLoad,Table[0,{j,1,Length[Transpose[visualEbolaMatrixCropped]]},totalVisualInfections];

plotVisualLines=MatrixPlot[visualLines,ImageSize->Large];
)(*constructVisualMatrix*)

Relative risk calibration
generateR[viralLoadIN_]:=
(
RcoefOUT=0;
While[RcoefOUT<=1,
Switch[viralLoadIN,
1,
meal=Max[1,Exp[RandomVariate[NormalDistribution[Log[1.2],0.415]]]];
RRmeal=Max[meal,Exp[RandomVariate[NormalDistribution[Log[2.2],0.31]]]]/meal;
RcoefOUT=RRmeal^(1/((52*Sum[fatalMean[[i]],{i,5,10}]/6+3*Sum[nonFatalMean[[i]],{i,5,10}]/6)/55-
((52*Sum[fatalMean[[i]],{i,1,4}]/4+3*Sum[nonFatalMean[[i]],{i,1,4}]/4)/55)));
2,
talk=Max[1,Exp[RandomVariate[NormalDistribution[Log[0.7],0.525]]]];
RRtalk=Max[talk,Exp[RandomVariate[NormalDistribution[Log[3.9],0.58]]]]/talk;
RcoefOUT=RRtalk^(1/((52*Sum[fatalMean[[i]],{i,5,10}]/6+3*Sum[nonFatalMean[[i]],{i,5,10}]/6)/55-
((52*Sum[fatalMean[[i]],{i,1,4}]/4+3*Sum[nonFatalMean[[i]],{i,1,4}]/4)/55)));
3,
bed=Max[1,Exp[RandomVariate[NormalDistribution[Log[1.3],0.34]]]];
RRbed=Max[bed,Exp[RandomVariate[NormalDistribution[Log[2.2],0.33]]]]/bed;
RcoefOUT=RRbed^(1/((52*Sum[fatalMean[[i]],{i,5,10}]/6+3*Sum[nonFatalMean[[i]],{i,5,10}]/6)/55-
((52*Sum[fatalMean[[i]],{i,1,4}]/4+3*Sum[nonFatalMean[[i]],{i,1,4}]/4)/55)));
];
];
RcoefOUT
)(*generateR*)

Analysis
Defining distributions
define[iterationsIN_]:=
(
numOfIterations=iterationsIN;
maxSickPeriod=14;
minSickPeriod=5;

maxIncubationPeriod=15;
minIncubationPeriod=5;

incubationDist=Round[RandomVariate[TriangularDistribution[{minIncubationPeriod-
0.5,maxIncubationPeriod+0.5},8],{numOfIterations,Length[id]}]];
sickPeriodDist=Round[RandomVariate[TriangularDistribution[{minSickPeriod-
0.5,maxSickPeriod+0.5},8],{numOfIterations,Length[id]}]];

contactFirstStageDist=Round[RandomVariate[HistogramDistribution[numContacts],{numOfIterations,Length[id]}]];
contactSecondStageDist=Round[RandomVariate[TruncatedDistribution[{0.5,5.5},HistogramDistribution[numContacts]],{numOfIterations,Length[id]}]];
stageCutoffDist=Round[RandomVariate[UniformDistribution[{0.5,5.5}],{numOfIterations,Length[id]}]];
];

Rcoef=Table[generateR[viralLoad],{i,1,numOfIterations}];

viralLoadFatalDist=
Table[Table[Table[RandomVariate[TruncatedDistribution[{0,Infinity},NormalDistribution[fatalMean[[i]],fatalSD[[i]]]],{i,1,maxSickPeriod}],{k,1,Length[id]}],{j,1,numOfIterations}];viralLoadNonFatalDist=
Table[Table[Table[RandomVariate[TruncatedDistribution[{0,Infinity},NormalDistribution[nonFatalMean[[i]],nonFatalSD[[i]]]],{i,1,maxSickPeriod}],{k,1,Length[id]}],{j,1,numOfIterations}];
)(*define*)

```

```

Single draw from distributions
singleDraw:=
(
Rcoef=generateR[viralLoad];
define[1];
coverage=0;
propContacts=1;

sickVectorsForMatrix=Table[0,{i,1,Length[id]};
sickLengthForMatrix=Table[0,{i,1,Length[id]};
incubationAliveForMatrix=Table[0,{i,1,Length[id]};
iter=1;

For[person=1,person<=Length[id],person++,

incubationDead=incubationDist[[iter,person]];
incubationAlive=incubationDead;

sickLengthDead =sickPeriodDist[[iter,person]];
sickLengthAlive =sickLengthDead;

contactFirstStageDead=contactFirstStageDist[[iter,person]];
contactSecondStageDead=contactSecondStageDist[[iter,person]];

contactFirstStageAlive=contactFirstStageDist[[iter,person]];
contactSecondStageAlive=contactFirstStageAlive;

stageCutoff=stageCutoffDist[[iter,person]]; (*Days counted from end*)

viralLoadDead=Rcoef[[iter]]^viralLoadFatalDist[[iter,person]];
viralLoadAlive=Rcoef[[iter]]^viralLoadNonFatalDist[[iter,person]];

constructSickVector[];

If[status[[person]]=="Dead",
sickVectorsForMatrix[[person]]=sickVectorDead;
sickLengthForMatrix[[person]]=sickLengthDead,
sickVectorsForMatrix[[person]]=sickVectorAlive;
sickLengthForMatrix[[person]]=sickLengthAlive;
incubationAliveForMatrix[[person]]=incubationAlive
];
];

constructMatrix[];

rAllSingle=rAll;
rDeadSingle=rDead;
rAliveSingle=rAlive;
)(*singleDraw*)

Sensitivity analysis (Iterations, Coverage, Proportion of contacts)
sensAnal[iterationsIN_,coverageIN_,propContactsIN_] :=
(
coverage=coverageIN;
propContacts=propContactsIN;

define[iterationsIN];

rAllSensitivity=Table[0,{i,1,numOfIterations}];
rDeadSensitivity=Table[0,{i,1,numOfIterations}];
rAliveSensitivity=Table[0,{i,1,numOfIterations}];
rPerDayAllSensitivity=Table[0,{i,1,numOfIterations}];
rPerDayDeadSensitivity=Table[0,{i,1,numOfIterations}];
rPerDayAliveSensitivity=Table[0,{i,1,numOfIterations}];
personPickedIndex=Table[{},{i,1,numOfIterations}];
personPickedRVector=Table[{},{i,1,numOfIterations}];

For[iter=1,iter<=numOfIterations,iter++,

sickVectorsForMatrix=Table[0,{i,1,Length[id]};

```

```

sickLengthForMatrix=Table[0,{i,1,Length[id]};
incubationAliveForMatrix=Table[0,{i,1,Length[id]};

tracedCount=1;

For[person=1,person<=Length[id],person++,

incubationDead=incubationDist[[iter,person]];
incubationAlive=incubationDead;

sickLengthDead =sickPeriodDist[[iter,person]];
sickLengthAlive =sickLengthDead;

If[NumberQ[id[[person]]],
contactFirstStageDead=numContacts[[tracedCount]];
contactSecondStageDead=Round[RandomVariate[TruncatedDistribution[{0.5,Min[(numContacts[[tracedCount]]+0.5),5.5]},HistogramDistribution[numContacts]]]];
contactFirstStageAlive=numContacts[[tracedCount]];
contactSecondStageAlive=numContacts[[tracedCount]];

tracedCount+=1
,
contactFirstStageDead=contactFirstStageDist[[iter,person]];
contactSecondStageDead=contactSecondStageDist[[iter,person]];
contactFirstStageAlive=contactFirstStageDist[[iter,person]];
contactSecondStageAlive=contactFirstStageAlive;
];

stageCutoff=stageCutoffDist[[iter,person]]; (*Days counted from end*)

viralLoadDead=Rcoef[[iter]]^viralLoadFatalDist[[iter,person]];
viralLoadAlive=Rcoef[[iter]]^viralLoadNonFatalDist[[iter,person]];

constructSickVector[];

If[status[[person]]=="Dead",
sickVectorsForMatrix[[person]]=sickVectorDead;
sickLengthForMatrix[[person]]=sickLengthDead,
sickVectorsForMatrix[[person]]=sickVectorAlive;
sickLengthForMatrix[[person]]=sickLengthAlive;
incubationAliveForMatrix[[person]]=incubationAlive
];
];

constructMatrix[];

rand=RandomInteger[{cropTop,cropBottom}];
While[Not[cropLeft<= symptomDateIndex[[rand]]&&(symptomDateIndex[[rand]]+sickLengthAlive-1)<=
cropRight],
rand=RandomInteger[{cropTop,cropBottom}];
];

personPickedIndex[[iter]]=rand;
If[status[[rand]]=="Dead",
personPickedRVector[[iter]]=rMatrix[[rand,symptomDateIndex[[rand]]];;symptomDateIndex[[rand]]+sick
LengthDead-1]],
personPickedRVector[[iter]]=rMatrix[[rand,symptomDateIndex[[rand]]];;symptomDateIndex[[rand]]+sick
LengthAlive-1]];
];

rAllSensitivity[[iter]]=rAll;
rDeadSensitivity[[iter]]=rDead;
rAliveSensitivity[[iter]]=rAlive;
rPerDayAllSensitivity[[iter]]=rPerDayAll;
rPerDayDeadSensitivity[[iter]]=rPerDayDead;
rPerDayAliveSensitivity[[iter]]=rPerDayAlive;
];
)(*sensAnal*)

Secondary cases distribution (one person per iteration)
secondaryCases[]:=

```

```

(
personPickedDeadRVector={};
personPickedAliveRVector={};

For[i=1,i<=Length[personPickedIndex],i++,
If[status[[personPickedIndex[[i]]]]=="Dead",
personPickedDeadRVector=Insert[personPickedDeadRVector,personPickedRVector[[i]],-1],
personPickedAliveRVector=Insert[personPickedAliveRVector,personPickedRVector[[i]],-1]
];
];

Print["Overall ",Length[personPickedIndex]," picked. "];
Print["Survived: ",Length[personPickedAliveRVector]," , Deceased:
",Length[personPickedDeadRVector]];

plotPersonPickedBothRVectorHist=Histogram[Table[Total[personPickedDeadRVector[[i]]],{i,1,Length[personPickedDeadRVector]}],Table[Total[personPickedAliveRVector[[i]]],{i,1,Length[personPickedAliveRVector]}]],
{1,"Probability",ChartStyle->{Blue,Red},ChartLegends->Placed[{"Deceased","Survived"},Top],AxesOrigin->{0,0},PlotRange->{{0,15},{0,1}}];
plotPersonPickedDeadRVectorHist=Histogram[Table[Total[personPickedDeadRVector[[i]]],{i,1,Length[personPickedDeadRVector]}],{1,"Probability",ChartStyle->Blue,PlotLabel->"Secondary cases - Deceased",AxesOrigin->{0,0},PlotRange->{{0,15},{0,1}}];
plotPersonPickedAliveRVectorHist=Histogram[Table[Total[personPickedAliveRVector[[i]]],{i,1,Length[personPickedAliveRVector]}],{1,"Probability",ChartStyle->Red,PlotLabel->"Secondary cases - Survived",AxesOrigin->{0,0},PlotRange->{{0,15},{0,1}}];
plotSinglePersonAllHist=Histogram[Table[Total[personPickedRVector[[i]]],{i,1,Length[personPickedRVector]}],{1,"Probability",ChartStyle->Green,PlotLabel->"All",AxesOrigin->{0,0},PlotRange->{{0,15},{0,1}}];

plotHistogramsSecondary=GraphicsGrid[{{plotPersonPickedBothRVectorHist,
plotPersonPickedAliveRVectorHist,plotPersonPickedDeadRVectorHist,plotSinglePersonAllHist}},ImageSize->1200];
)(*secondaryCases*)

Secondary cases per day (one person per iteration)
secondaryCasesPerDay[]:=
(
personPickedDeadPerDay=Table[0,{i,1,Length[Flatten[personPickedDeadRVector,{2}]]];
confIntDeadPerDay=Table[{0,0},{i,1,Length[Flatten[personPickedDeadRVector,{2}]]];

For[i=1,i<=Length[personPickedDeadPerDay],i++,
personPickedDeadPerDay[[i]]=Mean[Flatten[personPickedDeadRVector,{2}][[i]]];
confIntDeadPerDay[[i,1]]=Sort[Flatten[personPickedDeadRVector,{2}][[i]][[Round[0.975*Length[Flatten[personPickedDeadRVector,{2}][[i]]]]]];
confIntDeadPerDay[[i,2]]=Sort[Flatten[personPickedDeadRVector,{2}][[i]][[Max[Round[0.025*Length[Flatten[personPickedDeadRVector,{2}][[i]]],1]]]];
];

personPickedAlivePerDay=Table[0,{i,1,Length[Flatten[personPickedAliveRVector,{2}]]];
confIntAlivePerDay=Table[{0,0},{i,1,Length[Flatten[personPickedAliveRVector,{2}]]];

For[i=1,i<=Length[personPickedAlivePerDay],i++,
personPickedAlivePerDay[[i]]=Mean[Flatten[personPickedAliveRVector,{2}][[i]]];
confIntAlivePerDay[[i,1]]=Sort[Flatten[personPickedAliveRVector,{2}][[i]][[Round[0.975*Length[Flatten[personPickedAliveRVector,{2}][[i]]]]]];
confIntAlivePerDay[[i,2]]=Sort[Flatten[personPickedAliveRVector,{2}][[i]][[Max[Round[0.025*Length[Flatten[personPickedAliveRVector,{2}][[i]]],1]]]];
];

Print["Secondary cases total - Deceased: ", N[Total[personPickedDeadPerDay]]];
Print["Secondary cases total - Survived: ", N[Total[personPickedAlivePerDay]]];

errorPerDayDeadHi=Table[confIntDeadPerDay[[i,1]],[i,1,Length[confIntDeadPerDay]]]-
personPickedDeadPerDay;

```

```

errorPerDayDeadLo=Table[confIntDeadPerDay[[i,2]],{i,1,Length[confIntDeadPerDay]}]-
personPickedDeadPerDay;

errorPerDayAliveHi=Table[confIntAlivePerDay[[i,1]],{i,1,Length[confIntAlivePerDay]}]-
personPickedAlivePerDay;
errorPerDayAliveLo=Table[confIntAlivePerDay[[i,2]],{i,1,Length[confIntAlivePerDay]}]-
personPickedAlivePerDay;

Needs["ErrorBarPlots`"];

confIntDead=Table[ErrorBar[{errorPerDayDeadLo[[i]],errorPerDayDeadHi[[i]]},{i,1,Length[personPic
kedDeadPerDay]}];
confIntAlive=Table[ErrorBar[{errorPerDayAliveLo[[i]],errorPerDayAliveHi[[i]]},{i,1,Length[person
PickedAlivePerDay]}];

plotSecondaryPerDay=GraphicsGrid[{{ErrorListPlot[Table[{{i,personPickedAlivePerDay[[i]]},confIntA
live[[i]]},{i,1,Length[personPickedAlivePerDay]}],PlotLabel->"Secondary cases per day -
Survived",PlotRange->

{0,4},PlotStyle->Red,AxesOrigin-
>{0,0}},ErrorListPlot[Table[{{i,personPickedDeadPerDay[[i]]},confIntDead[[i]]},{i,1,Length[person
PickedDeadPerDay]}],PlotLabel->"Secondary cases per day - Deceased",PlotRange->

{0,4},PlotStyle->Blue,AxesOrigin->{0,0}}}],ImageSize->1000;
)(*secondaryCasesPerDay*)

Removing from all population
removeAll[]:=
(
rPerPersonAllSensitivity=Table[Flatten[rPerDayAllSensitivity[[i]],{2}},{i,1,Length[rPerDayAllSens
itivity]}];

rUntilDayAll=Table[Table[0,{i,1,Length[rPerDayAllSensitivity]},{j,1,maxSickPeriod}],
{k,1,Length[coverageRemoveScenarios]}];

For[k=1,k<=Length[coverageRemoveScenarios],k++,
For[i=1,i<=Length[rPerPersonAllSensitivity],i++,
For[p=1,p<=Length[rPerPersonAllSensitivity[[i]]],p++,
rPerPersonAllSensitivity[[i]][p]=PadRight[rPerPersonAllSensitivity[[i]][p],maxSickPeriod,
{0}];
];
For[j=1,j<=maxSickPeriod,j++,
tempCopy=rPerPersonAllSensitivity[[i]];
For[p=1,p<=Length[rPerPersonAllSensitivity[[i]]],p++,
If[RandomReal[]<coverageRemoveScenarios[[k]],
tempCopy[[p]][j;Length[rPerPersonAllSensitivity[[i]][p]]]=0;
If[j==maxSickPeriod,
tempCopy[[p]][j]=0;
];
];
];
rUntilDayAll[[k,i,j]]=Total[Mean[tempCopy]];
];
];
];

plotRemoveAll=GraphicsGrid[{{Table[ListLinePlot[Table[Count[Flatten[rUntilDayAll[[k]],{2}][[j]],x_
/;x<=1]/Length[Flatten[rUntilDayAll[[k]],{2}][[j]]],{j,1,maxSickPeriod}],
PlotRange->{0,1.1},AxesLabel->{"days"},PlotLabel->{"Removing from all, coverage:
",coverageRemoveScenarios[[k]]},{k,1,Length[coverageRemoveScenarios]}],ImageSize->1200;
)(*removeAll*)

Removing deceased from population
removeDead[]:=
(
rPerPersonDeadSensitivity=Table[Flatten[rPerDayDeadSensitivity[[i]],{2}},{i,1,Length[rPerDayDeadS
ensitivity]}];
rPerPersonAliveSensitivity=Table[Flatten[rPerDayAliveSensitivity[[i]],{2}},{i,1,Length[rPerDayAli
veSensitivity]}];

```

```

rUntilDayDead=Table[Table[0,{i,1,Length[rPerDayDeadSensitivity]},{j,1,maxSickPeriod}},{k,1,Length
[coverageRemoveScenarios]};

For[k=1,k<=Length[coverageRemoveScenarios],k++,
For[i=1,i<=Length[rPerPersonDeadSensitivity],i++,
For[p=1,p<=Length[rPerPersonDeadSensitivity[[i]]],p++,
rPerPersonDeadSensitivity[[i]][[p]]=PadRight[rPerPersonDeadSensitivity[[i]][[p]],maxSickPeriod,{0
}];
];
For[p=1,p<=Length[rPerPersonAliveSensitivity[[i]]],p++,
rPerPersonAliveSensitivity[[i]][[p]]=PadRight[rPerPersonAliveSensitivity[[i]][[p]],maxSickPeriod,
{0}];
];
For[j=1,j<=maxSickPeriod,j++,
tempCopy=rPerPersonDeadSensitivity[[i]];
For[p=1,p<=Length[rPerPersonDeadSensitivity[[i]]],p++,
If[RandomReal[]<coverageRemoveScenarios[[k]],
tempCopy[[p]][[j;;Length[rPerPersonDeadSensitivity[[i]][[p]]]]]=0;
If[j==maxSickPeriod,
tempCopy[[p]][[j]]=0;
];
];
];
tempSum=Length[tempCopy]+Length[rPerPersonAliveSensitivity[[i]]];
rUntilDayDead[[k,i,j]]=Total[Mean[tempCopy]]*Length[tempCopy]/tempSum+Total[Mean[rPerPersonAlives
ensitivity[[i]]]*Length[rPerPersonAliveSensitivity[[i]]]/tempSum;
];
];
];

plotRemoveDead=GraphicsGrid[{Table[ListLinePlot[Table[Count[Flatten[rUntilDayDead[[k]],{2}][[j]],
x_/;x<=1]/Length[Flatten[rUntilDayDead[[k]],{2}][[j]]],{j,1,maxSickPeriod}],
PlotRange->{0,1.1},AxesLabel->{"days"},PlotLabel->{"Removing from deceased, coverage:
",coverageRemoveScenarios[[k]]}],{k,1,Length[coverageRemoveScenarios]}],ImageSize->1200];
)(*removeDead*)

Reducing contacts (Iterations per scenario)
reduceContacts[iterUsed_]:=
(
probEradicate[]:=Count[Table[Mean[rAllSensitivity[[iii]]],{iii,1,Length[rAllSensitivity]}]
,x_/;x<=1]/Length[rAllSensitivity];

reduceContactsResults=Table[Table[0,{j,1,Length[contactScenarios]}],{i,1,Length[coverageContactss
cenarios]};

For[ii=1,ii<=Length[coverageContactsScenarios],ii++,
For[jj=1,jj<=Length[contactScenarios],jj++,
sensAnal[iterUsed,coverageContactsScenarios[[ii]],contactScenarios[[jj]]];
reduceContactsResults[[ii,jj]]=probEradicate[];
];
];

plotReduceContacts=Table[ListLinePlot[Transpose[{contactScenarios,reduceContactsResults[[ii]]}],P
lotRange->{0,1.1},AxesOrigin->{0,0},ImageSize->Medium,AxesLabel->{"% contacts"},
PlotLabel->{"Reducing contacts, coverage:
",coverageContactsScenarios[[ii]]}],{ii,1,Length[coverageContactsScenarios]};
)(*reduceContacts*)

```