

Supplementary Material

for

Modeling the Sustainability of a Ceramic Water Filter Intervention

by

Jonathan Mellor, Lydia Abebe, Beeta Ehdaie, Rebecca Dillingham and
James Smith

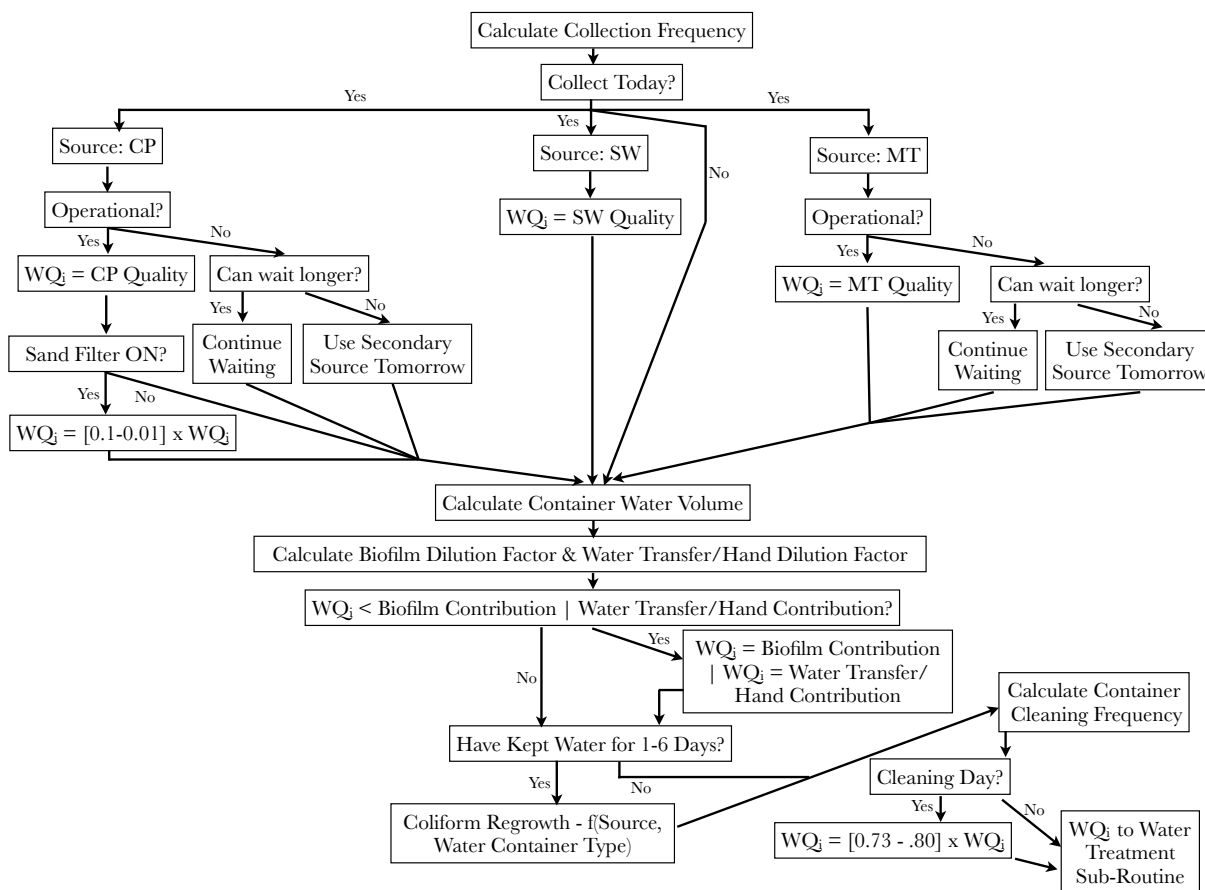


Figure S1: Flow chart of Collect Water Sub-Routine of original model which was used in this study. This flow chart is repeated for each household for each day of the simulation. Square brackets indicate values that are stochastically varied between minimum and maximum values. SW - surface water, CP - community piped, MT - municipal tap, WQ_i daily water quality of i^{th} household. All input data was taken from field measurements (Mellor et al., 2012, 2013) (Reprinted with permission from (Mellor et al., 2012). Copyright 2012 American Chemical Society.)

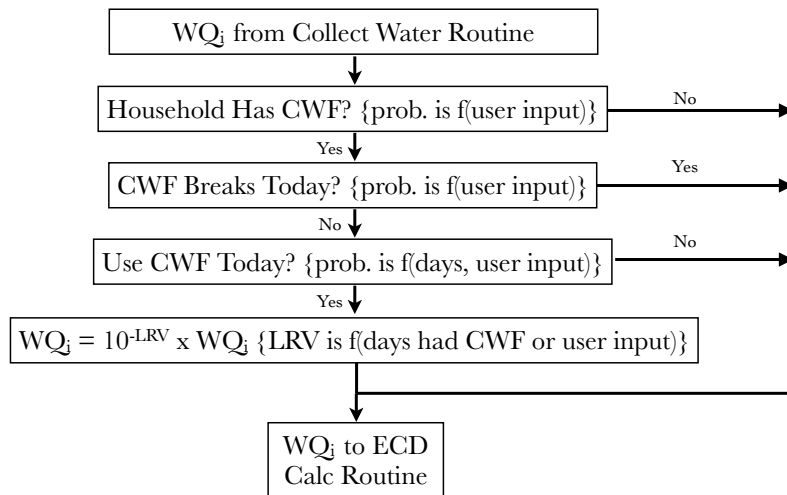


Figure S2: Flow chart of Basic CWF Routine. WQ_i is obtained for each household on each day from the Collect Water Routine (Figure S1). Default values for filter prevalence, breakage percent and compliance are 100%, 20% (over two years) and 90% respectively and are based on field measurements. LRVs are obtained from the piecewise linear fits of Figure ?? or user input. The WQ_i output is then sent to the ECD Calculation Routine as is given in Figure S4.

Table S1: Basic CWF Routine default values and the ranges used in the behavior space analysis. The default values were based on field measurements and represent the baseline scenario.

Parameter	Default Value	Range of Values in Behavior Space Analysis
Filter Prevalence	100%	0 - 100%
Filter Compliance	90%	0 - 100%
Breakage Percent	20%	0 - 100%

Table S2: Other parameter values used in additional behavior space analyses. All parameter ranges are based on values typically found in this study or by previous researchers.

Parameter	Range of Values in Behavior Space Analysis
Filter LRV	10^{-5} , 5×10^{-5} , 10^{-4} , 5×10^{-4} , 10^{-3} , 5×10^{-3} , 10^{-2} , 5×10^{-2} , 10^{-1}
Cleaning Interval	Every 0 - 730 days
Yearly Compliance Decline	0 - 100%
Breakage Date	Day 0 to 730
Threshold Water Quality	0, 0.5, 1, 5, 10, 50, 100, 500, 1000, 2000 CFU/100ml
Willingness to Pay	20, 30, 50, 70, 80, 100, 150, 200, 250, 300, 500 South African Rand

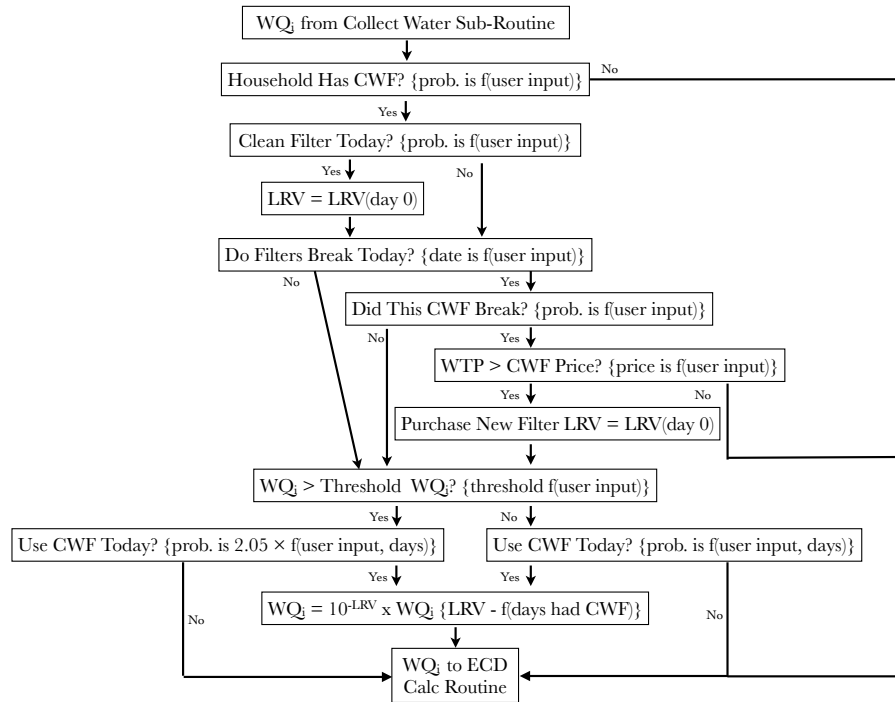


Figure S3: Flow chart of Additional CWF Routines. WQ_i is obtained for each household on each day from the Collect Water Routine (Figure S1). The probability of then having and using a water filter on a given day is based on whether a household has a working filter or decides to purchase one. If a household has a working filter they can clean it. When cleaned, a filter reverts to its day 0 microbial effectiveness. If a household's filter breaks, it can purchase a new filter if their WTP is greater than the purchase price. WTP is based on the field data shown in Figure S6. The probability of a household using a filter is also based on the threshold water quality above which a household is 2.05 times as likely to use their filter. It is finally based on compliance rates which are a function of the days the model has been run and user input based on field data. LRVs are obtained from the piecewise linear fits of Figure ?? or user input. The WQ_i output is then sent to the ECD Calculation Routine as is given in Figure S4. This is the basic flow chart used to generate the multi-parameter analysis of Figure 5. Relevant portions of this routine were also used to study the cleaning interval and compliance decrease experiments shown in Figure 2. In addition, the two parameter analyses shown in Figure 3 were conducted as described in the Additional Routines Tested section. In both of these cases, the unrelated portions of the code were bypassed.

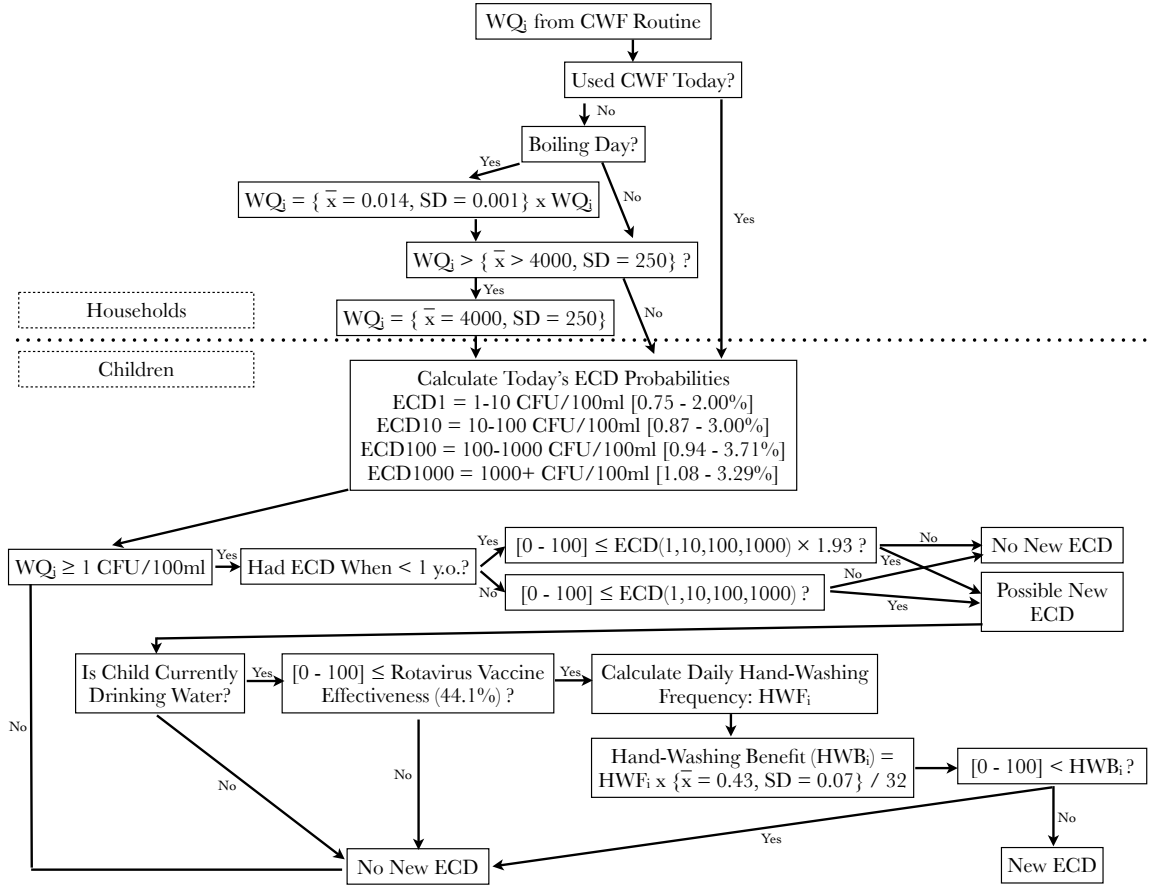


Figure S4: ECD calculation subroutine. The Households portion occurs for each household for each day while the Children portion occurs for each child each day. The children live within households and drink the household's water. Curly brackets indicate a variable that is stochastically varied according to a normal distribution with mean and standard deviations indicated. Square brackets indicate values that are stochastically varied between minimum and maximum values. Parentheses indicate a functional relationship, i.e. the probability of getting ECD is a function of WQ_i . (Adapted with permission from (Mellor et al., 2012). Copyright 2012 American Chemical Society.)

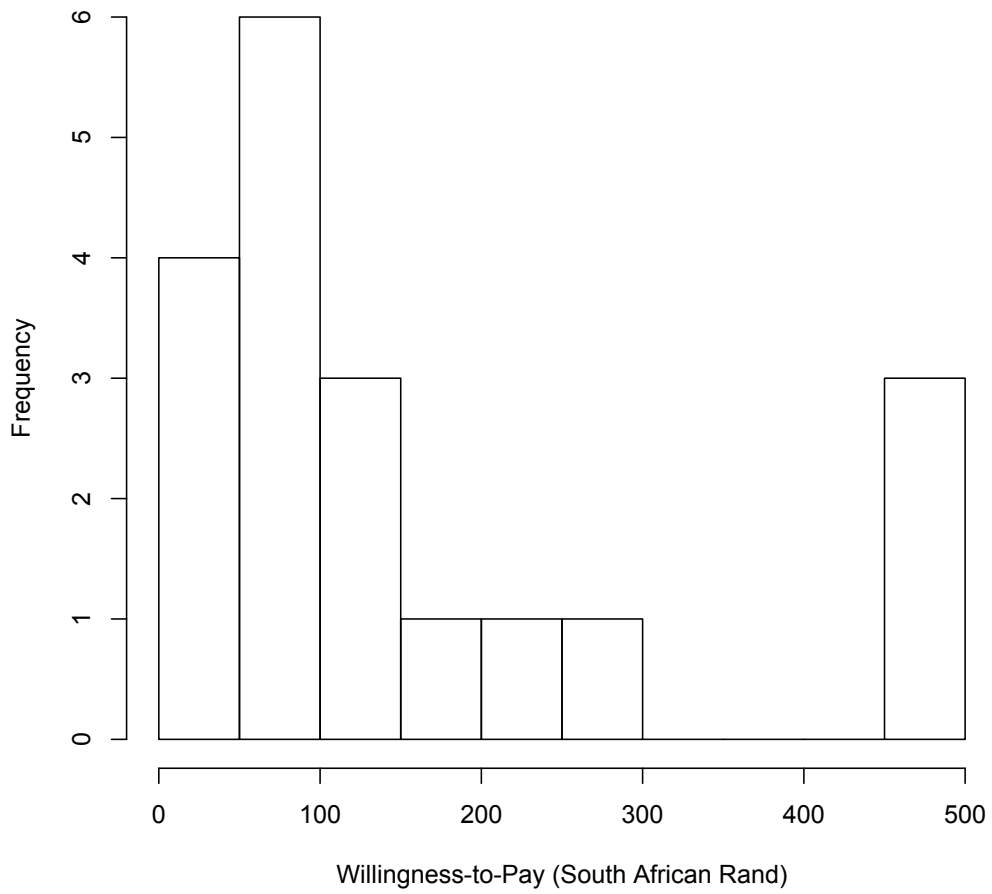


Figure S5: Willingness-to-pay field survey data. $\$1 \approx 8.9$ South African Rand

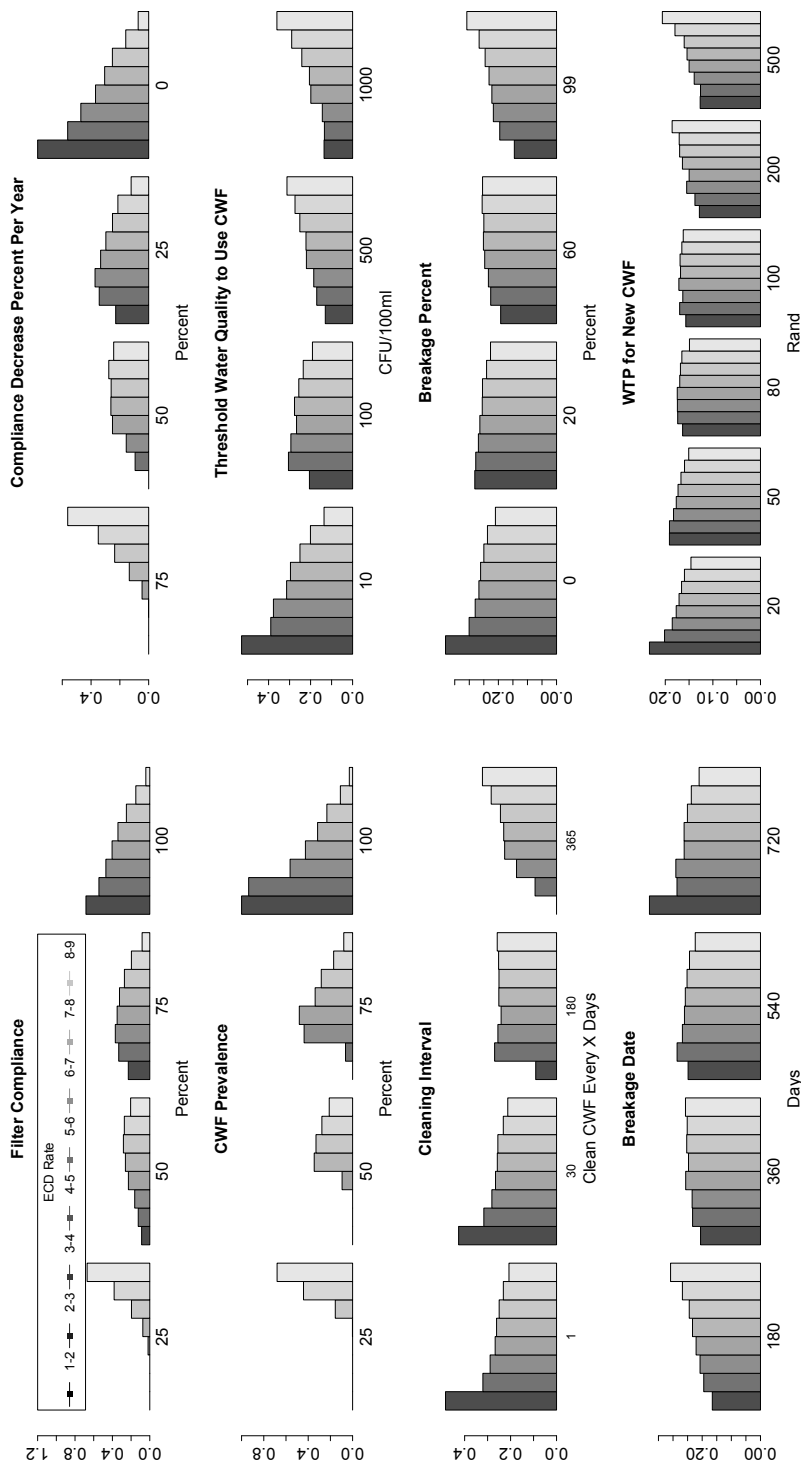


Figure S6: Eight parameter behavior space analyses indicating the relative importance of the more salient versus less salient parameters. Plots are normalized histograms which indicate the percent of runs with a given parameter value leading to ECD rates in a given range. Results indicate that compliance, prevalence and compliance declines were the most important parameters.

Table S3: First Multiple Parameter Behavior Space Analysis

Parameter	Range of Values in Behavior Space Analysis
Filter LRV	10^{-5} , 5×10^{-5} , 10^{-4} , 5×10^{-4} , 10^{-3} , 5×10^{-3} , 10^{-2} , 5×10^{-1} , 10^{-1}
Filter Compliance	0 - 100% (10% increments)
Filter Prevalence	0 - 100% (10% increments)
Yearly Linear Compliance Decline	0 - 100% (10% increments)

Table S4: Second Multiple Parameter Behavior Space Analysis. WTP = willingness to pay. CWF = ceramic water filter.

Parameter	Range of Values in Behavior Space Analysis
Breakage Date	Day 1 to 730 by 60 day increments
Breakage Percent	0, 20, 40, 60, 80 and 99%
Threshold Water Quality	0, 0.5, 1, 5, 10, 50, 100, 500, 1000 and 2000 CFU/100ml
Cleaning Interval	1 to 730 by 60 day increments
WTP for New CWF	20, 30, 50, 70, 80, 100, 150, 200, 250, 300, 500 South African Rand

```

extensions [array table]
breed [ households household]
breed [ children child ]

undirected-link-breed [housetokids housetokid] ;; directed
link from child to HH

households-own [
  pri-water-source ;A households primary water source. 1 =
    River, 2 = Community Piped, 3 = Municipal Tap, 4 = Hose (CP system in Tshibvumo
    where pipes come from the river, but don't go through any kind of treatment)
  sec-water-source ;A households secondary water source. 1 =
    River, 2 = Community Piped, 3 = Municipal Tap, 4 = Hose (CP system in Tshibvumo
    where pipes come from the river, but don't go through any kind of treatment)
  daily-wq ;The main water household quality parameter
    that changes every day according to the water chain model.
  storage-container ;Storage container type. 1 = Wide necked, 2
    = narrow neck
  days-keep-water ;Number of days keep water, used for
    biological regrowth
  hh-days-can-wait ;The maximum number of days a household can
    wait until they must get water
  today-source ;The source a household is using today, is
    usually equal to pri-water-source except when that source isn't working
  river-rand-number ;Random number used to determine which
    biological regrowth regime to use.
  pipe-rand-number ;Random number used to determine which
    biological regrowth regime to use.
  hand-wash-number ;Number used in hand-washing effectiveness
    calculation
  tap-min ;Minimum number of days between water
    collection
  tap-max ;Maximum number of days between water
    collection
  days-waiting ;Days waiting for water to start working
    again after checked for the first time
  has-filter
  jc-total ;Amount of total coliform bacteria associated
    with the sidewalls of the storage containers
  cup-total ;Amount of total coliform bacteria associated
    with the water transfer devices used for wide mounth containers
  clean-min ;Minimum number of days between container
    cleanings
  clean-max ;Maximum number of days between container
    cleanings
  boil-max ;Maximum number of days between water
    boilings
  boil-min ;Minimum number of days between water
    boilings
  hw-max ;Maximum number of times wash hands per day
  hw-min ;Minimum number of times wash hands per day
  bhw-total ;Total coliform bacteria associated with
    hands before washing hands.
  filrannum ;Random number that determines what linear
    filter deterioration track a household will take
  wtp ;WIP for a CWF
  days-have-filter ;Days have had this CWF
  hh-cf-usage ;A daily compliance variable
]

children-own[
  sex ;1 = male, 2 = female
  age ;0 to 730 days
  ecd ;Does have ecd or not
  ecd-cases ;Total number of ecd cases per child
  height ;Height
  stuntdays ;Days with stunted growth
  growthdelta ;Daily incremental increase in height when
    have acute case of ECD
  pastgot ;Did a kid have a prolonged case of ECD
    during their first year

```

```

have-had-diarrhea
doublecases ;Total cases of prolonged ECD
vaccinated ;Vaccinated against rotavirus?
growth-factor

]

globals [xmin xmax ymin ymax ;Parameters for Interface
height-array ;Standard curve array for boys
feight-array ;Standard curve array for girls
bheight730 ;Array of boys heights at age 2
gheight730 ;Array of girls heights at age 2
pipewq ;Array of piped WQ measurements
mtwq ;Array of MT WQ measurements
surfwq ;Array of surface WQ measurements
all-ecd-cases ;Total number of ECD cases
grand-total-days-with-ecd ;Total of all kid's days-with-ecd
totalstuntdays ;Total number of stunted growth days
nostuntdays ;Total number of days without stunting
haz0 haz60 haz120 haz180 haz240 haz300 haz360 haz420 haz480 haz540 haz600 haz660
haz730 ;Array of HAZs at each age in days
haz30 haz90 haz150 haz210 haz270 haz330 haz390 haz450 haz510 haz570 haz630 haz690
last-day-pipe-work ;The 'tick' when the piped system last worked
last-day-MT-work ;The 'tick' when the MT system last worked
last-day-hose-work ;The 'tick' when the hose system last worked
BSD GSD ;Arrays of the standard deviations of both
boys and girls heights
pre-pro-haz
rannum1000 rannum100 rannum10 rannum1 ;Variables for different col-ecd scenarios
for the ranges of coliform levels
mean-daily-wq-list ;Daily mean water qualities.
all-double-cases ;Sums up all double ECD cases
all-single-cases ;Sums up all single ECD cases
malheight-array malfeight-array ;make and efmial height arrays
collection-times ;Counts total number of times houses collect
water
water-usage-array ;Array of probabilities of a child drinking
water on a given day
days-keep-water-list ;List of days-keep-water
boil-events ;Counter for number of times houses boil
their water
daily1000 daily100 daily10 daily1 daily0 ;Counters to show percent of houses with
water quality of a given quality
median-daily-wq-list ;Median daily WQ
total-days-waiting ;Total days waiting
haz-table ;child stunting tables
haz-table-double ;child stunting table for "double" cases
ecdage-list ;ages of ECD incidences in the communities
ecdage-list-double ;ages of double ECD incidences in the
communities
broken ;number of broken filters
grand-total-ecd-cases ;total number of ECD cases overall
]

to setup

ca

set-current-directory "/Users/jem3w/Documents/limpopo/abm/"
set xmin 30.4336
set xmax 30.4630
set ymin -22.790
set ymax -22.7718
set-default-shape households "house"
set bheight730 []
set gheight730 []
set haz0 [] set haz60 [] set haz120 [] set haz180 [] set haz240 [] set haz300 []
set haz360 [] set haz420 [] set haz480 [] set haz540 [] set haz600 [] set
haz660 [] set haz730 []
set haz30 [] set haz90 [] set haz150 [] set haz210 [] set haz270 [] set haz330 [] set
haz390 [] set haz450 [] set haz510 [] set haz570 [] set haz630 [] set haz690 []
set all-ecd-cases []
set grand-total-days-with-ecd []
set all-double-cases []
set all-single-cases []
set mean-daily-wq-list []

```

```

set median-daily-wq-list []
set pre-pro-haz []
set totalstuntdays 0
set nostuntdays 0
set days-keep-water-list [1 ]
set mtwq []
set pipewq []
set surfwq []

;Pipe, MT, and SURF water quality lists. Each element of the list represents a field
measurement.
set pipewq [440 585 420 460 0 0 540 120 1220 1160 40 40 80 680 0 0 640
200 180 120 480 560 0 40]
set mtwq [300 0 380 195 20 35 0 180 0 0 0 0 0 40 0 280 200 300 0 0
80 60 20 500 0 0 60 0 0]
set surfwq [880 680 1150 1330 1210 1120 2120 1260 2120 1220 160 3000 1240
1160 1280 1020 20 300 715 500 930 860 640 990 480 870 120 140 820
1200 1565 1280 200 305 100 160 200 4120 200 900 560 500 100 140
600 540 3020 3020 860 740 720 600 240 120 200 380 1060 720 0 0 60
160 780 1020 640 420 700 580 1300 1460 100 140]
;If performing water quality tests then take water quality values from Interface input
if pipe-wq-test = TRUE [set pipewq [] set pipewq fput pipe-quality-test pipewq]
if surf-wq-test = TRUE [set surfwq [] set surfwq fput surf-quality-test surfwq]
if mt-wq-test = TRUE [set mtwq [] set mtwq fput mt-quality-test mtwq]

;SD from http://www.cdc.gov/growthcharts/who\_charts.htm also Mei 2007
set BSD array:from-list [1.90 1.95 2.01 2.05 2.09 2.12 2.15 2.18 2.21 2.25
2.29 2.34 2.38 2.43 2.49 2.54 2.59 2.65 2.70 2.76 2.82 2.88 2.94 3.00
3.06 3.12 3.18 3.24 3.3 3.36 3.42] ;3.12 and after are approximations
set GSD array:from-list [1.86 1.95 2.03 2.10 2.16 2.22 2.26 2.31 2.37 2.41
2.47 2.52 2.57 2.63 2.68 2.74 2.79 2.85 2.90 2.96 3.01 3.07 3.12 3.17
3.22 3.27 3.32 3.37 3.42 3.47 3.52]

make-height-array
setup-households 0 0
setup-hh-water-sources
setup-containers
setup-collect-freq
setup-boil
setup-hw
setup-initial-children
setup-haz

end

to go

collect-water2
treat-water
kids-drink
calculate-height4
child-old
do-plots

tick
if ticks > totaldays [
stop ]

end

to setup-initial-children
let j 1
;Set up 410 initial child - an average of 1 per household
while [ j <= 410 ]
[
ask household random 410
[
hatch-children 1 [ set color green set size 1

```

```

set ecd 0
set ecd-cases 0
set pastgot 0
set doublecases 0
ifelse (random 100 > 50) ; sex ratio from https://www.cia.gov/library/publications/
the-world-factbook/geos/sf.html
[set sex true]
[set sex false]
create-housetokid-with myself ;creates link between households and children born
there
set age 0

;Sets initial height equal to CDC norms and standard deviations for newborns
ifelse (sex = TRUE)
[set height random-normal 48.14 1.9019] ;SD Using R and http://www.cdc.gov/
growthcharts/who/boys_length_weight.htm
[set height random-normal 47.72 1.8584] ;SD Using R and http://www.cdc.gov/
growthcharts/who/girls_length_weight.htm
]
]
set j j + 1
]
end

to-report wq-report ;Reports mean of the median-daily-wq-list
report mean median-daily-wq-list
end
to-report wq-report2 ;Reports mean of the mean-daily-wq-list
report mean mean-daily-wq-list
end
to-report keep-water-report ;Reports how many days keep water on average
report mean days-keep-water-list
end
to-report child-number ;Total number of children
report count children
end
to-report ave-height ;Average height of 2 year olds
report mean bheight730
end
to-report ave-height2
report mean gheight730
end
to-report girl-length ;Number of girls and boys
report length gheight730
end
to-report boy-length
report length bheight730
end
to-report total-single ;Total single ECD cases
report mean all-ecd-cases - mean all-double-cases
end
to-report total-double ;Total double ECD cases
report mean all-double-cases
end
to-report ecd-all ;Mean ALL ECD cases
report mean all-ecd-cases
end
to-report ecd-all2 ;Median ALL ECD cases
report median all-ecd-cases
end

to-report total-stunt-days ;Calculates the percentage of stunting days
report totalstuntdays / (totalstuntdays + nostuntdays) * 100
end
to-report percent-boil-days ;Percent of days that folks boil their water
report boil-events / (ticks * 410) * 100
end
to-report daily-1000 ;Percent of households with the following water
qualities
report daily1000 / (daily1000 + daily100 + daily10 + daily1 + daily0) * 100
end

```

```

to-report daily-100
  report daily100 / (daily1000 + daily100 + daily10 + daily1 + daily0) * 100
end
to-report daily-10
  report daily10 / (daily1000 + daily100 + daily10 + daily1 + daily0) * 100
end
to-report daily-1
  report daily1 / (daily1000 + daily100 + daily10 + daily1 + daily0) * 100
end
to-report daily-0
  report daily0 / (daily1000 + daily100 + daily10 + daily1 + daily0) * 100
end
to-report totaldayswaiting ;days waiting for water source to work
  report total-days-waiting / collection-times
end

to collect-water2

;First decide which sources are working (pipe, MT and hose) Data from average of
  reported how-freq-sorce-work for each.
;That data is in terms of how many days per week each source works. 1 = everyday, 7 =
  once a week
  if random-float 7 > 3.18 [set last-day-pipe-work ticks]
  if random-float 7 > 4.79 [set last-day-MT-work ticks]
  if random-float 7 > 2.3 [set last-day-hose-work ticks]
  if pipe-rel-test = TRUE [if random-float 7 > pipe-test [set last-day-pipe-work ticks
    ]] ;Can adjust the frequency of each source working by adjusting source-test
    inputs in Interface
  if MT-rel-test = TRUE [if random-float 7 > MT-test [set last-day-MT-work ticks]]
  if hose-rel-test = TRUE [if random-float 7 > hose-test [set last-day-hose-work ticks
    ]]

ask households[
  set days-keep-water days-keep-water + 1 ;Increment days keep water for incubation
  experiment and cleaning

  let collect-freq (random (tap-max - tap-min) + tap-min) ;Calculate collection
  frequency as a random number bewteen tap-min and tap-max, in terms of every 1,
  2, 3 days...

  if collect-test = TRUE ;If performing the
  water collection frequency test
  [set collect-freq collect-level ;Set collect-freq to
  the collect-level (which is input on the Interface)
  set last-day-pipe-work ticks ;Make each source work
  everyday so that you can study the collection frequency without the
  confounding
  set last-day-MT-work ticks ;question of deciding
  whether or not the source is working
  set last-day-hose-work ticks
  ]
;print today-source
if remainder ticks collect-freq = 0 or days-waiting > 0 [;If today is a multiple of
the collection-frequency or they are waiting for source to start working again
then we'll look to collect today

  if today-source = 2 [ ;For piped water system
  ifelse (ticks - last-day-pipe-work = 0) ;If the piped system is
  working today then collect water
  [
  set today-source pri-water-source ;Reset to primary water
  souce if had to go to secondary source during previous day.
  set days-waiting 0 ;Set days-waiting to
  zero
  set daily-wq item random length pipewq pipewq ;Set daily-wq as random
  choice from water quality data
  set total-days-waiting days-keep-water + total-days-waiting ;Add to the
  total days waiting
  set collection-times collection-times + 1 ;Increment
  collection-times
  ]
  ]

```

```

if (sand-filter-on = TRUE) ;If the sand-filter is
    ON
[
    set daily-wq daily-wq * (random 10 + 1) / 100 ;Setting up Slow Sand
    Filter to have 1-2 log removal (90-99%)
]
set days-keep-water 0 ;Set incubation days
    keep water variable to zero

]
[
    ifelse hh-days-can-wait > days-waiting ;If the source isn't
        working today then check to see if a house has waited as long as it can
        this was taken from survey data
    [set days-waiting days-waiting + 1] ;If it can wait longer
        then increment days-waiting
    [set today-source sec-water-source] ;If not, then use
        secondary source on the next day
]
]
if today-source = 3 [ ;For MF water system
    ifelse (ticks - last-day-MF-work = 0) ;If the system is
        working today then collect water
    [
        set today-source pri-water-source ;Reset to primary water
            souce if had to go to secondary source.
        set days-waiting 0 ;Set days-waiting to
            zero
        set daily-wq item random length mtwq mtwq ;Set daily-wq as a
            random choice from water quality data
        set collection-times collection-times + 1 ;Increment collection
            times variable
        set total-days-waiting days-keep-water + total-days-waiting
        set days-keep-water 0 ;Set incubation
            variable to zero
    ]
    [
        ifelse hh-days-can-wait > days-waiting ;If the source hasn't
            worked recently then see if a house has waited as long as it can
        [set days-waiting days-waiting + 1] ;If it can wait longer
            then increment days-waiting
        [set today-source sec-water-source] ;If not, then use
            secondary source on the next day
    ]
]
]
if today-source = 4[ ;For Hose water system
    ifelse (ticks - last-day-hose-work = 0) ;If the system is working today
        then collect water
    [
        set days-waiting 0
        set today-source pri-water-source ;Reset to primary water
            souce if had to go to secondary source.
        set daily-wq item random length pipewq pipewq ;Set daily-wq according
            to the data
        set collection-times collection-times + 1
        set total-days-waiting days-keep-water + total-days-waiting
        set days-keep-water 0 ;Set incubation
            variable to zero
    ]
    [
        ifelse hh-days-can-wait > days-waiting ;If the source hasn't
            worked recently then see if a house has waited as long as it can
        [set days-waiting days-waiting + 1] ;If it can wait longer
            then increment days-waiting
        [set today-source sec-water-source ] ;If not, then use
            secondary source on the next day
    ]
]
]
if today-source = 1 [ ;For River System - is
    always working
    set today-source pri-water-source ;Reset to primary
        water souce if had to go to secondary source.

```

```

set days-waiting 0 ;Set days-waiting to
zero
set daily-wq item random length surfwq surfwq ;Set daily-wq
according to the data
set collection-times collection-times + 1
set total-days-waiting days-keep-water + total-days-waiting
set days-keep-water 0 ;Set incubation variable
to zero
]
]

let volume random (20 - .5) + .5 ;Calculating the volume
of the water in storage containers. Volumes were measured during HHB study and
had a flat distribution from 0.5 to 20L

let rannum (random-float (volume - .5) + .5) / .5 ;The ability of bacteria
measured for the jc-total experiment to disattach itself and contaminate the
;water is unknown. To model this phenomenon we take the typical volume of stored
water and assume that the dilution factor is somewhere between 1 and
;Volume / 0.5 L where 0.5 L was the volume of the water used in the jc-total
experiment.
;This is then used as a dilution factor in the code below.
if rannum > 39 [print rannum]
if (daily-wq < (jc-total / rannum)) [set daily-wq (jc-total / rannum)] ;WQ cannot
go below the biofilm layer amount => 108 = 9.65L / 0.1L

if storage-container = 2 [set daily-wq daily-wq + (cup-total + bhw-total) / (volume
/ .5)] ;If a household has an open style container then add in cup-total as
the amount added by dipping in a cup
;No dilution factor is used in this case because the experiment closely mimicked
the actual way folks will do it.

]

if coliform-growing = TRUE[
ask households[

;Coliform incubation growth for River/Closed is located in position 3 in the
following vectors under today-source = 1
;Coliform incubation growth for Pipe/Closed is located in positions 2,3 in the
following vectors under today-source = 2 or 4
;Coliform incubation growth for River/Open is located in positions 0,1,2 in the
following vectors under today-source = 1
;Coliform incubation growth for Pipe/Open is located in positions 0,1 in the
following vectors under today-source = 2 or 4

if (today-source = 1) ;River
[

if (days-keep-water = 1) [set daily-wq daily-wq * (item river-rand-number
[0.7695 7.6905 11.0753 7.7073])] ;Numbers based on incubation survey
if (days-keep-water = 2) [set daily-wq daily-wq * (item river-rand-number
[1.6361 1.9938 1.8913 3.1962])]
if (days-keep-water = 3) [set daily-wq daily-wq * (item river-rand-number
[0.6398 0.3126 0.0020534 0.3881])]
if (days-keep-water = 4) [set daily-wq daily-wq * (item river-rand-number
[1.3877 2.2384 148.2500 2.2143])]
if (days-keep-water = 5) [set daily-wq daily-wq * (item river-rand-number
[0.9933 0.3299 1.7251 0.3629])]
if (days-keep-water = 6) [set daily-wq daily-wq * (item river-rand-number
[0.6763 3.3632 0.9932 1.4730])]

]
if (today-source = 2 or today-source = 4) ;Piped or Hose
[

if (days-keep-water = 1) [set daily-wq daily-wq * (item pipe-rand-number
[47.2800 43.9091 4.4211 23.7500])] ;Numbers based on incubation survey
if (days-keep-water = 2) [set daily-wq daily-wq * (item pipe-rand-number [0.1447
1.6687 3.4881 1.7474])]
if (days-keep-water = 3) [set daily-wq daily-wq * (item pipe-rand-number [5.2632
0.6203 1.1058 0.3976])]

```



```

        if(days-keep-water = 4) [set daily-wq daily-wq * (item pipe-rand-number [1.4089
            1.1740 0.4167 2.3864])]
        if(days-keep-water = 5) [set daily-wq daily-wq * (item pipe-rand-number [1.0174
            0.2828 0.8000 0.7937])]
        if(days-keep-water = 6) [set daily-wq daily-wq * (item pipe-rand-number [1.1752
            1.2560 1.3426 0.9400])]
    ]
]
ask households[
    let clean-freq (random (clean-max - clean-min) + clean-min) ;Calculate cleaning
        frequency to be somewhere between clean min and clean max values
    if clean-test = TRUE ;If in clean-test mode set clean-freq to value on
        Interface
        [set clean-freq clean-level]
    if(remainder ticks clean-freq = 0)
        {
            let randnum random-float (.80 - .73) + .73 ;0.80 is a 20% decrease
                seen in biofilm follow-up experiment whereby we scraped the sides of the
                containers and got a median 20% resuspension of bacteria. The 0.73 is the
                percentage difference between Good and OK bucket washers in HHB study
            set daily-wq daily-wq * randnum
        }
]
end

to treat-water

ask households
[
    let boil-freq (random (boil-max - boil-min) + boil-min) ;boil-freq is a number
        that represents how frequently to boil, 1 would be everyday, 2 every other day,
        7 once a week etc...
    if boil-test = TRUE [set boil-freq boil-level] ;If in boil testing
        mode then set boil-freq equal to that on the Interface
    let filter-treat-today FALSE
    if has-filter = 1 ;If a house had a water
        filter
    [
        if ticks = break-date and random 100 < break-percent [set has-filter 0 set
            broken broken + 1] ;Is today the filter breaking day? If so, a
            percent will break
        if has-filter = 0 and WIP >= random-normal filter-price 25 [set has-filter 1 set
            days-have-filter 0] ;If a house's filter breaks then they can purchase
            a new filter if the price is below their WIP
        if remainder ticks cf-clean-freq = 0 [set days-have-filter 0]
            ;A house can also clean their
            lower reservoir
        let ticks2 0
        ifelse constant-filter-effectiveness = TRUE ;ifelse to turn ON or
            OFF the feature to have the CWF effectivenesses to remain fixed at day 0
            levels
        [set ticks2 0]
        [
            counter increments the filter's age ;if OFF then this
            set ticks2 days-have-filter
            set days-have-filter days-have-filter + 1
        ]
    ]
    set hh-cf-usage hh-cf-usage * (year-compliance-down / 100) ^ (1 / 365)
]

```

```

;Compliance can decrease over time
let today-cf-usage hh-cf-usage

if threshold = TRUE [if daily-wq >= threshold-wq [set today-cf-usage hh-cf-usage
* 2.0493819]] ;Households are more likely to treat their water if it is
above a threshold
if today-cf-usage > 100 [set today-cf-usage 100]
if hh-cf-usage > 100 [set hh-cf-usage 100]

ifelse effectiveness-pre-set = TRUE ;
Effectiveness is pre-set by user
[
if random-float 100 < today-cf-usage [set daily-wq daily-wq * effectiveness]
]
;or is
based on the field measurements for < 1 year or the 2-3 year timeframe.

if ticks2 >= 0 and ticks2 <= 365 and random-float 100 < today-cf-usage
[
set filter-treat-today TRUE

if filrannum = 0 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.006670276 + 4.025715384))]
if filrannum = 1 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.005830596 + 3.556302501))]
if filrannum = 2 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.004302276 + 3.877946952))]
if filrannum = 3 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.002408141 + 3.579326204))]
if filrannum = 4 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.002920785 + 3.496583734))]
if filrannum = 5 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.001556412 + 3.073240317))]
if filrannum = 6 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.000659039 + 2.416640507))]
if filrannum = 7 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.003028795 + 1.939519253))]
if filrannum = 8 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.002205166 + 2.62324929))]
if filrannum = 9 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.007257286 + 2.465382851))]
if filrannum = 10 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.00190856 + 1.98811284))]
if filrannum = 11 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.005870949 + 1.477349991))]
if filrannum = 12 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.006008339 + 3.338456494))]
if filrannum = 13 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.002775483 + 2.899541923))]
if filrannum = 14 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.001252466 + 2.40654018))]
if filrannum = 15 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.005611106 + 3.155639634))]
if filrannum = 16 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.002885321 + 2.755874856))]
if filrannum = 17 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.006589247 + 3.550228353))]
if filrannum = 18 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.005826309 + 3.122215878))]
if filrannum = 19 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.004270591 + 2.117271296))]
if filrannum = 20 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.00085923 + 2.62838893))]
if filrannum = 21 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.006788884 + 3.042236765))]
if filrannum = 22 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.003020307 + 3.327358934))]
if filrannum = 23 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.005111595 + 2.937016107))]
if filrannum = 24 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.001438865 + 2.667452953))]
if filrannum = 25 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.004212457 + 1.113943352))]
if filrannum = 26 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.000419865 + 2.867938651))]

```

```

if filrannum = 27 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.004259082 + 2.411619706))]
if filrannum = 28 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.004623036 + 3.350248018))]
if filrannum = 29 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.004813594 + 3.903089987))]
if filrannum = 30 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.000362068 + 0.698970004))]
if filrannum = 31 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.006941793 + 2.983626287))]
if filrannum = 32 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.000135645 + 3.025305865))]
if filrannum = 33 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.003439103 + 2))]
]
if ticks2 > 365 and ticks2 <= 1095 and random-float 100 < today-cf-usage
[
set filter -treat-today TRUE

if filrannum = 0 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.00532072 + 3.533127444))]
if filrannum = 1 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.00484736 + 3.197421108))]
if filrannum = 2 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.003238484 + 3.489662744))]
if filrannum = 3 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.001890193 + 5.148218289))]
if filrannum = 4 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.000707785 + 2.688838788))]
if filrannum = 5 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.002590313 + 3.450614356))]
if filrannum = 6 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.00061272 + 1.952448326))]
if filrannum = 7 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.000921538 + 1.170370477))]
if filrannum = 8 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.000129325 + 3.475338485))]
if filrannum = 9 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.002071592 + 0.572604641))]
if filrannum = 10 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.004508888 + 2.937232793))]
if filrannum = 11 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.000695112 + -0.919261984))]
if filrannum = 12 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.001150044 + 0.725646383))]
if filrannum = 13 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.002268672 + 2.714555866))]
if filrannum = 14 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.001431639 + 2.471938276))]
if filrannum = 15 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.002568549 + 0.170065494))]
if filrannum = 16 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.00162379 + 2.295416071))]
if filrannum = 17 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.002531632 + 2.069198777))]
if filrannum = 18 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.00261323 + 1.949441972))]
if filrannum = 19 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.000483276 + 0.734901482))]
if filrannum = 20 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.00123372 + 3.392315737))]
if filrannum = 21 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.001744233 + 1.200939389))]
if filrannum = 22 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.000671232 + 1.979947251))]
if filrannum = 23 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.001022801 + 1.444606627))]
if filrannum = 24 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.001138927 + 2.557975412))]
if filrannum = 25 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.00011201 + -0.382720071))]
if filrannum = 26 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
-0.002305709 + 3.86277302))]
if filrannum = 27 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
0.000195816 + 0.78558195))]
if filrannum = 28 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *

```

```

        0.000181124 + 1.596729609))]]
    if filrannum = 29 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
        -0.001903897 + 2.841050617))]
    if filrannum = 30 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
        -0.000939514 + 0.909737541))]
    if filrannum = 31 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
        -0.003296797 + 1.65320293))]
    if filrannum = 32 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
        -0.001175884 + 3.504013918))]
    if filrannum = 33 [set daily-wq daily-wq * 10 ^ ( - (ticks2 *
        -0.004240837 + 4.803177866))]

    if daily-wq < 0 [set daily-wq 0]

]
]

if filter-treat-today = FALSE [ ;If not use filter then
    houses are assumed to occasionally revert to their old boiling practices
    if remainder ticks boil-freq = 0 ;From Clasen Papers (
        Guatemala 2010, India 2008, Vietnam 2007) Used normal dist calc to get SDs
        from CIs (http://onlinestatbook.com/chapter8/mean.html)
    [ set boil-events boil-events + 1 ;Counts number of boil
        events
        let boil-effectiveness random-normal 0.01431 0.001109
        if (boil-effectiveness < 0) [set boil-effectiveness 0] ;Just in case get a
            negative number (very rare)
        set daily-wq boil-effectiveness * daily-wq ;Mean, SD, N Guat: .88 .075 206
            :: India: 2.1 .025 1088 :: Viet: 1.52 0.046 245 Took weighted average by N
            for means and SDs to get these values
    ]
]
]
end
to kids-drink
    ;probabilities of getting ECD based on WQi. Numbers taken from the best available
    literature on the subject
    set rannum1000 random-float (3.29 - 1.08) + 1.08
    set rannum100 random-float (3.71 - 0.94) + 0.94
    set rannum10 random-float (3 - 0.87) + 0.87
    set rannum1 random-float (2 - 0.75) + 0.75

    set rannum1000 rannum1000
    set rannum100 rannum100
    set rannum10 rannum10
    set rannum1 rannum1

let daily-wq-list2 []
ask households [
    if (daily-wq > random-normal 4000 250) [set daily-wq random-normal 4000 250] ;4000
        was the highest recorded during Dec-July 2010-11 household testing, 250 is
        arbitrary

    set daily-wq-list2 fput daily-wq daily-wq-list2 ;Take
        each household's daily WQ value and put it into daily-wq-list2

let hw-freq (random (hw-max - hw-min) + hw-min) / 32 ;hw-freq is a number between 0
    and 1 that signifies the percent of 32 that people wash their hands see below
    for meaning of 32
if hw-test = TRUE [set hw-freq hw-level / 32] ;If in HW testing mode then set all
    hw-freq to a hw-level input on Interface

set hand-wash-number 100 * ( ((hw-freq) * abs(random-normal .43 .0695))); + 100;From
    Curtis 2000 paper (Domestic hygiene and diarrhoea pinpointing the problem),

```

```

washing at all critical times would mean 32 total times per day
                                ;.31 .43                                ;
                                So, taking linear
                                approximation. Also, using
                                Aiello 2008, hand-washing
                                decreases diarrheal diseases
                                by 31% (95% CI, 19-42%).
                                Calculated 6.95 using
                                NORMINV in

let increase-risk-number 0.518      ;There is an increased risk of getting ECD if a
child has had it in the past. Guerrant and Checkley studies.

if (daily-wq >= 1000)                ;Worst case senario
[ ask housetokid-neighbors [
    let rannum random-float 100 ;(item 0 [hand-wash-number] of housetokid-neighbors)
    ;Calculating a random number between 0 and something less than or equal to
    100 depending on the hand-washing behavior of the households. This is used
    to slightly increase the probability that someone might get diarrhea if they
    drink poor water.
    if pastgot = 1 [set rannum rannum * increase-risk-number] ; Ave of 15.5%/29.9%
    and 1.91/3.12 = 0.565 is from Moore 2010 paper - kids who get Prolonged
    diarrhea before age 1 are twice as likely to get persistent later in life
    if (rannum <= rannum1000) [ifelse ecd = 1 [set ecd 2][set ecd 1 set stundays 0]]
    ;Based on Brown 2008 Escherichia coli in household drinking water and
    diarrheal disease risk: evidence from Cambodia
    set daily1000 daily1000 + 1      ;Incrementing incidences of daily-wq of this
    quality
]
]

if (daily-wq >= 100) and (daily-wq < 1000)    ;;extreme risk population
[ ask housetokid-neighbors [
    let rannum random-float 100;(item 0 [hand-wash-number] of housetokid-neighbors)
    if pastgot = 1 [set rannum rannum * increase-risk-number] ; Ave of 15.5%/29.9%
    and 1.91/3.12 = 0.565 is from Moore 2010 paper - kids who get Prolonged
    diarrhea before age 1 are twice as likely to get persistent later in life
    if (rannum <= rannum100) [ifelse ecd = 1 [set ecd 2][set ecd 1 set stundays 0]]
    ;Based on Brown 2008 Escherichia coli in household drinking water and
    diarrheal disease risk: evidence from Cambodia
    set daily100 daily100 + 1      ;Incrementing incidences of daily-wq of this
    quality
]
]

if (daily-wq >= 10) and (daily-wq < 100)    ;;high risk population
[ask housetokid-neighbors [
    let rannum random-float 100;(item 0 [hand-wash-number] of housetokid-neighbors)
    if pastgot = 1 [set rannum rannum * increase-risk-number] ; Ave of 15.5%/29.9%
    and 1.91/3.12 = 0.565 is from Moore 2010 paper - kids who get Prolonged
    diarrhea before age 1 are twice as likely to get persistent later in life
    if (rannum <= rannum10) [ifelse ecd = 1 [set ecd 2][set ecd 1 set stundays 0]]
    ;Based on Brown 2008 Escherichia coli in household drinking water and
    diarrheal disease risk: evidence from Cambodia
    set daily10 daily10 + 1      ;Incrementing incidences of daily-wq of this
    quality
]
]

if (daily-wq < 10 and daily-wq >= 1)        ;;low risk population
[ask housetokid-neighbors [
    let rannum random-float 100;(item 0 [hand-wash-number] of housetokid-neighbors)
    if pastgot = 1 [set rannum rannum * increase-risk-number] ; Ave of 15.5%/29.9%
    and 1.91/3.12 = 0.565 is from Moore 2010 paper - kids who get Prolonged
    diarrhea before age 1 are twice as likely to get persistent later in life
    if (rannum <= rannum1) [ifelse ecd = 1 [set ecd 2][set ecd 1 set stundays 0]]
    ;Based on Brown 2008 Escherichia coli in household drinking water and
    diarrheal disease risk: evidence from Cambodia
    set daily1 daily1 + 1      ;Incrementing incidences of daily-wq of this
    quality
]
]

```

```

]
]
if (daily-wq < 1) [ set daily0 daily0 + 1 ;Incrementing incidences of daily-wq
of this quality
]
ask children[
if((stundays = 0 and ecd = 1) or ecd = 2)
[
if (random-float 100 >= array:item water-usage-array age) [ ;Decide whether
or not kids are drinking water based on MAL-ED data about child drinking
habits
if (ecd = 1 and stundays = 0)[set ecd 0]
if (ecd = 2)[set ecd 1]
]
if (random-float 100 < 44.1)[ ;Assume that all kids are vaccinated
and there is a 44.1% reduction in ECD from the
if (ecd = 1 and stundays = 0)[set ecd 0]
if (ecd = 2)[set ecd 1] ;Rota Virus vaccine (Madhi 2010, Table
3 for South Africa)
]
if (random-float 100 < item 0 [hand-wash-number] of housetokid-neighbors) [ ;
Reduces ECD for kids whose households washing their hands
if (ecd = 1 and stundays = 0)[set ecd 0]
if (ecd = 2)[set ecd 1]
]
]
]
;]
;Put mean and median daily WQ values into their respective lists.
set mean-daily-wq-list fput mean daily-wq-list2 mean-daily-wq-list
set median-daily-wq-list fput median daily-wq-list2 median-daily-wq-list

end

to calculate-height4
ask children[
if((ecd = 1 and stundays = 0) or ecd = 2)[ ;If got ECD this time
if age <= 365 [set pastgot 1] ;did the kid get ECD in their first
year?
set ecd-cases ecd-cases + 1
set have-had-diarrhea 1
set grand-total-ecd-cases grand-total-ecd-cases + 1
ifelse ecd = 1 ;if have single case then look up
field data to find the child of most similar age who got ECD and find out
what their growth stunting rate was
[
let $match-list sort-by [abs(?1 - age) < abs(?2 - age)] ecdage-list
set growth-factor (- (table:get haz-table first $match-list) / 240)
]
]
;Same as above except for double cases
let $match-list sort-by [abs(?1 - age) < abs(?2 - age)] ecdage-list-double
set growth-factor (- (table:get haz-table-double first $match-list) / 240)

```

```

    set doublecases doublecases + 1
    set ecd 1
    set stuntedays 0
  ]
]
ifelse(ecd = 1) ;If have ECD now then grow at a slowed rate
[
  ifelse(sex = TRUE) ;Boys and girls grow differently \
    [set growthdelta (((array:item height-array (age + 1)) - (array:item BSD floor
      ((age + 1) / 30)) * (growth-factor - (height - array:item height-array age)
      / (array:item BSD floor (age / 30)) ) - height))]
    [set growthdelta (((array:item feight-array (age + 1)) - (array:item GSD floor
      ((age + 1) / 30)) * (growth-factor - (height - array:item feight-array age)
      / (array:item GSD floor (age / 30)) ) - height))]

  set height height + growthdelta ;Set height
  set stuntedays stuntedays + 1 ;Count the number of stunted
    growth days from now for 180 days
  set totalstuntedays totalstuntedays + 1 ;stuntedays

  if (stuntedays > 240) [set ecd 0 set stuntedays 0] ;growth is always stunting for
    240 days
]
;If healthy
[
  ifelse(sex = TRUE) ;Boys and girls growing at different rates.
    [set height (height + (array:item height-array (age + 1) - array:item height-
      array age))]
    [set height (height + (array:item feight-array (age + 1) - array:item feight-
      array age))]

  set nostuntedays nostuntedays + 1
]
]
end

to child-old
ask children
[
  set age age + 1

  if (age > 730)
  [
    ;Save boy and girls heights
    ifelse(sex = TRUE)
    [set bheight730 fput height bheight730]
    [set gheight730 fput height gheight730]
    set all-ecd-cases fput ecd-cases all-ecd-cases ;Put number of cases of ECD
      into all-ecd-cases vector
    set all-double-cases fput doublecases all-double-cases

    die
  ]
;Save HAZ scores every 60 days for all kids. Don't save scores every day because of
memory problems.
ifelse(sex = TRUE)[

  if(age = 1)[set haz0 fput ((height - array:item height-array age) / array:item BSD
    0) haz0 ]

  if (age = 30)[set haz30 fput ((height - array:item height-array age) / array:item
    BSD 1) haz30]
]
]

```

```

if (age = 90)[set haz90 fput ((height - array:item height-array age) / array:item
  BSD 3) haz90]
if (age = 150)[set haz150 fput ((height - array:item height-array age) / array:item
  BSD 5) haz150]
if (age = 210)[set haz210 fput ((height - array:item height-array age) / array:item
  BSD 7) haz210]
if (age = 270)[set haz270 fput ((height - array:item height-array age) / array:item
  BSD 9) haz270]
if (age = 330)[set haz330 fput ((height - array:item height-array age) / array:item
  BSD 11) haz330]
if (age = 390)[set haz390 fput ((height - array:item height-array age) / array:item
  BSD 13) haz390]
if (age = 450)[set haz450 fput ((height - array:item height-array age) / array:item
  BSD 15) haz450]
if (age = 510)[set haz510 fput ((height - array:item height-array age) / array:item
  BSD 17) haz510]
if (age = 570)[set haz570 fput ((height - array:item height-array age) / array:item
  BSD 19) haz570]
if (age = 630)[set haz630 fput ((height - array:item height-array age) / array:item
  BSD 21) haz630]
if (age = 690)[set haz690 fput ((height - array:item height-array age) / array:item
  BSD 23) haz690]
if (age = 60)[set haz60 fput ((height - array:item height-array age) / array:item
  BSD 2) haz60]
if (age = 120)[set haz120 fput ((height - array:item height-array age) / array:item
  BSD 4) haz120]
if (age = 180)[set haz180 fput ((height - array:item height-array age) / array:item
  BSD 6) haz180]
if (age = 240)[set haz240 fput ((height - array:item height-array age) / array:item
  BSD 8) haz240]
if (age = 300)[set haz300 fput ((height - array:item height-array age) / array:item
  BSD 10) haz300]
if (age = 360)[set haz360 fput ((height - array:item height-array age) / array:item
  BSD 12) haz360]
if (age = 420)[set haz420 fput ((height - array:item height-array age) / array:item
  BSD 14) haz420]
if (age = 480)[set haz480 fput ((height - array:item height-array age) / array:item
  BSD 16) haz480]
if (age = 540)[set haz540 fput ((height - array:item height-array age) / array:item
  BSD 18) haz540]
if (age = 600)[set haz600 fput ((height - array:item height-array age) / array:item
  BSD 20) haz600]
if (age = 660)[set haz660 fput ((height - array:item height-array age) / array:item
  BSD 22) haz660]
if (age = 730)[set haz730 fput ((height - array:item height-array age) / array:item
  BSD 24) haz730]
]
[
if (age = 30)[set haz30 fput ((height - array:item feight-array age) / array:item
  GSD 1) haz30]
if (age = 90)[set haz90 fput ((height - array:item feight-array age) / array:item
  GSD 3) haz90]
if (age = 150)[set haz150 fput ((height - array:item feight-array age) / array:item
  GSD 5) haz150]
if (age = 210)[set haz210 fput ((height - array:item feight-array age) / array:item
  GSD 7) haz210]
if (age = 270)[set haz270 fput ((height - array:item feight-array age) / array:item
  GSD 9) haz270]
if (age = 330)[set haz330 fput ((height - array:item feight-array age) / array:item
  GSD 11) haz330]
if (age = 390)[set haz390 fput ((height - array:item feight-array age) / array:item
  GSD 13) haz390]
if (age = 450)[set haz450 fput ((height - array:item feight-array age) / array:item
  GSD 15) haz450]
if (age = 510)[set haz510 fput ((height - array:item feight-array age) / array:item
  GSD 17) haz510]
if (age = 570)[set haz570 fput ((height - array:item feight-array age) / array:item
  GSD 19) haz570]
if (age = 630)[set haz630 fput ((height - array:item feight-array age) / array:item
  GSD 21) haz630]
if (age = 690)[set haz690 fput ((height - array:item feight-array age) / array:item
  GSD 23) haz690]

if (age = 1) [set haz0 fput ((height - array:item feight-array age) / array:item
  GSD 0) haz0 ]

```



```

    if (age = 60)[set haz60 fput ((height - array:item feight-array age) / array:item
      GSD 2) haz60]
    if (age = 120)[set haz120 fput ((height - array:item feight-array age) / array:item
      GSD 4) haz120]
    if (age = 180)[set haz180 fput ((height - array:item feight-array age) / array:item
      GSD 6) haz180]
    if (age = 240)[set haz240 fput ((height - array:item feight-array age) / array:item
      GSD 8) haz240]
    if (age = 300)[set haz300 fput ((height - array:item feight-array age) / array:item
      GSD 10) haz300]
    if (age = 360)[set haz360 fput ((height - array:item feight-array age) / array:item
      GSD 12) haz360]
    if (age = 420)[set haz420 fput ((height - array:item feight-array age) / array:item
      GSD 14) haz420]
    if (age = 480)[set haz480 fput ((height - array:item feight-array age) / array:item
      GSD 16) haz480]
    if (age = 540)[set haz540 fput ((height - array:item feight-array age) / array:item
      GSD 18) haz540]
    if (age = 600)[set haz600 fput ((height - array:item feight-array age) / array:item
      GSD 20) haz600]
    if (age = 660)[set haz660 fput ((height - array:item feight-array age) / array:item
      GSD 22) haz660]
    if (age = 730)[set haz730 fput ((height - array:item feight-array age) / array:item
      GSD 24) haz730]
  ]
]
end

to setup-collect-freq
;Sets up collection frequency list using data from HHB study. Tap-Min is the minimum
  number of days between collections, tap-max is the maximum days and hh-days can
;wait is the max number of days a HH can wait. tap-min and tap-max were the min and
  max number of days reported by respondents during multiple types of questions
;during HHB.
;;Data is in terms of collects every X days

let collect-freq-list []

file-open "water_collect_freq.txt" ;; open txt file containing lat/long
  decimal format coordinates for HHB households (50 entries)
while [not file-at-end?]
[set collect-freq-list lput file-read collect-freq-list]
file-close
let a array:from-list n-values 196 [item ? collect-freq-list]

ask households [
  set tap-min 99
  set tap-max 99
  set hh-days-can-wait 99

  if who < 49
  [
    set tap-min array:item a (who * 4)
    set tap-max array:item a ((who * 4) + 1)
    set hh-days-can-wait array:item a ((who * 4) + 3)
  ]
]

ask households [
  if tap-min = 99 [ ;Basically, if no tap-min has yet to be assigned
    let iii 0
    let found-nearby-house 0 ;Tags to specify whether or not a nearby house has
      been found.

    while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
      searching outward radially
    [
      let near-quantity 999 ;re-setting the "near" sources to 999

```

```

ask households in-radius iii
[
  if (tap-min != 99 and who < 49);If nearby HHB house has legit storage
  container tag use that data to set near-storage
  [
    set near-quantity tap-min
    set found-nearby-house 1
  ]
]
set iii iii + 1
if tap-min = 99 and found-nearby-house = 1 ;Break out of the previous
search and set the formally unspecified house to be a nearby house
[set tap-min near-quantity]
]
]

ask households [
  if tap-max = 99 [ ;Basically, if no storage-countainer has yet to be assigned
  let iii 0
  let found-nearby-house 0 ;Tags to specificity whether or not a nearby house has
  been found.

  while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
  searching outward radially
  [
    let near-quantity 999 ;re-setting the "near" sources to 999
    ask households in-radius iii
    [
      if (tap-max != 99 and who < 49);If nearby HHB house has legit storage
      container tag use that data to set near-storage
      [
        set near-quantity tap-max
        set found-nearby-house 1
      ]
    ]
    set iii iii + 1
    if tap-max = 99 and found-nearby-house = 1 ;Break out of the previous
    search and set the formally unspecified house to be a nearby house
    [set tap-max near-quantity]
  ]
]
]

ask households [
  if hh-days-can-wait = 99 [ ;Basically, if no storage-countainer has yet to be
  assigned
  let iii 0
  let found-nearby-house 0 ;Tags to specificity whether or not a nearby house has
  been found.

  while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
  searching outward radially
  [
    let near-quantity 999 ;re-setting the "near" sources to 999
    ask households in-radius iii
    [
      if (hh-days-can-wait != 99 and who < 49);If nearby HHB house has legit
      storage container tag use that data to set near-storage
      [
        set near-quantity hh-days-can-wait
        set found-nearby-house 1
      ]
    ]
    set iii iii + 1
    if hh-days-can-wait = 99 and found-nearby-house = 1 ;Break out of the
    previous search and set the formally unspecified house to be a nearby

```

```

        house
        [set hh-days-can-wait near-quantity]

    ]
]
end

to setup-households [x y]

    let i 0
    let j 1
    let coord-list [ ]

    file-open "hbb_abm_gps_locations2.txt"          ;; open txt file containing lat/
        long decimal format coordinates for HBB households (50 entries)
        ;; open txt file containing lat/
        long decimal format coordinates
        for all households

    while [not file-at-end?]
    [set coord-list lput file-read coord-list]
    file-close

    file-open "ceramic_abm_gps_locations2.txt"      ;; txt file with all the Ceramic
        Filter locations 176 entries
    while [not file-at-end?]
    [set coord-list lput file-read coord-list]
    file-close

    file-open "remaining_abm_gps_locations2.txt"    ;185 remaining households from Jeff
        and Census
    while [not file-at-end?]
    [set coord-list lput file-read coord-list]
    file-close
    let a array:from-list n-values 820 [item ? coord-list] ;822 is the total number of
        household locations (50+176+185)*2 (2 is because it includes both lat and long)

    while [ j < 820] ;822 is the total number times two of the households.
    [
        create-ordered-households 1
        ;; create households and place on display
        [
            set xcor (array:item a j - xmin)/(xmax - xmin) * world-width + min-pxcor
            set ycor (array:item a i - ymin)/(ymax - ymin) * world-height + min-pycor
        ]
    ]

    ask households [

        setxy xcor ycor
        set color orange
        set size 1.5
    ]

    set i i + 2
    set j j + 2
    ]

    ask households [

        ifelse random 100 <= cf-prevalence [set has-filter 1][set has-filter 0]
        let wtp-list [50 100 300 150 30 500 50 20 500 150 70 200 100 100 80 250 500 150
            100]
        set wtp one-of wtp-list
        set hh-cf-usage cf-compliance
    ]

```

```

    set filrannum random 34
  ]

end

to setup-boil
;Sets up boiling frequencies.  boil-min and boil-max values taken from HHB surveys as
min and max values taken from the different ways those questions were asked
;Data is in terms of boils every X days

let boiling-list [ ]
file-open "boiling.txt" ;File with codes for
while [not file-at-end?]
[set boiling-list lput file-read boiling-list]
file-close
let a array:from-list n-values 98 [item ? boiling-list]
let boil-freq-list []
ask households [
  set boil-min 99
  set boil-max 99
  if who < 49;For all of the HHB households
  [
    set boil-min array:item a (who * 2)
    set boil-max array:item a ((who * 2) + 1)
  ]
]

ask households [
if boil-min = 99 [ ;Basically, if no boil-min has been defined yet
let iii 0
let found-nearby-house 0 ;Tags to specify whether or not a nearby house has
been found.

while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
searching outward radially
[
let near-storage 999 ;re-setting the "near" sources to 999
let near-storage2 999
ask households in-radius iii
[
if (boil-min != 99 and who < 49);If nearby HHB house has legit boil-min
use that data to set near-storage
[
set near-storage boil-min
set near-storage2 boil-max
set found-nearby-house 1
]
]
]
set iii iii + 1
if boil-min = 99 and found-nearby-house = 1 ;Break out of the previous
search and set the formally unspecified house to be a nearby house
[
set boil-min near-storage
set boil-max near-storage2
]
;print who

] ; if storage-container = 1 [set color 86] ;B, 1 or Closed Blue
; if storage-container = 2 [set color 16] ;C, 2 or Open Red
]

]

end

to setup-hw
;Sets up hand-washing frequency as the min and max number of times a day a person
washes their hands.

let hw-list [ ]

```

```

file-open "hw.txt" ;File with codes for hw "min" and "max" times per day
while [not file-at-end?]
[set hw-list lput file-read hw-list]
file-close
let a array:from-list n-values 98 [item ? hw-list]
let hw-freq-list []
ask households [ ;Set all equal to 99 for now as a placeholder
set hw-min 99
set hw-max 99
if who < 49;For all of the HHB households
[
set hw-min array:item a (who * 2) ;Extract min and max values
set hw-max array:item a ((who * 2) + 1)
]
]

ask households [
if hw-min = 99 [ ;Basically, if no hw-min/max has been defined yet
let iii 0
let found-nearby-house 0 ;Tags to specify whether or not a nearby house has
been found.

while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
searching outward radially
[
let near-storage 999 ;re-setting the "near" sources to 999
let near-storage2 999
ask households in-radius iii
[
if (hw-min != 99 and who < 49);If nearby HHB house has legit storage
container tag use that data to set near-storage
[
set near-storage hw-min
set near-storage2 hw-max
set found-nearby-house 1
]
]
]
set iii iii + 1
if hw-min = 99 and found-nearby-house = 1 ;Break out of the previous
search and set the formally unspecified house to be a nearby house
[
set hw-min near-storage
set hw-max near-storage2
]
]
]

;print mean hw-freq-list
end

to setup-containers
;jc-total represents the biofilm layer on the inside of water storage containers,
cup-total is the bacteria associated with the cups, container-cleaning is the
;number of times

let container-list [ ]
file-open "hbb_storage_cont.txt" ;File with codes for the drinking water storage
containers water from HHB 1=B (closed top), 2=C (open top)
while [not file-at-end?]
[set container-list lput file-read container-list]
file-close

let jc-total-list [ ]
file-open "hbb_jc_total.txt" ;File with codes for jc-total
while [not file-at-end?]
[set jc-total-list lput file-read jc-total-list]
file-close

let cup-total-list [ ]

```

```

file-open "hbb_cup-total.txt" ;File with codes for cup_total
while [not file-at-end?]
[set cup-total-list lput file-read cup-total-list]
file-close

let cleaning-list [ ]
file-open "container_cleaning.txt" ;File with codes for container_cleaning
while [not file-at-end?]
[set cleaning-list lput file-read cleaning-list]
file-close
let a array:from-list n-values 98 [item ? cleaning-list]

let bhw-total-list [ ]
file-open "bhw.txt" ;File with codes for jc_total
while [not file-at-end?]
[set bhw-total-list lput file-read bhw-total-list]
file-close

ask households [
set jc_total 99
set cup_total 99
set clean-min 99
set clean-max 99
set bhw_total 99
if who < 49;For all of the HHB households
[
set storage-container item who container-list ;Set storage containers to be
those specified in HHB study, I am omitting Ceramic filter data because it
isn't that accurate,

set jc_total item who jc-total-list
set cup_total item who cup-total-list
set clean-min array:item a (who * 2)
set clean-max array:item a ((who * 2) + 1)
set bhw_total item who bhw-total-list

]
]
ask households [
if storage-container < 1 [ ;Basically, if no storage-countainer has yet to be
assigned
let iii 0
let found-nearby-house 0 ;Tags to specificity whether or not a nearby house has
been found.

while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
searching outward radially
[
let near-storage 999 ;re-setting the "near" sources to 999
ask households in-radius iii
[
if (storage-container = 1 or storage-container = 2 and who < 49);If nearby
HHB house has legit storage container tag use that data to set near-
storage
[
set near-storage storage-container
set found-nearby-house 1
]
]
set iii iii + 1
if storage-container < 1 and found-nearby-house = 1 ;Break out of the
previous search and set the formally unspecified house to be a nearby
house
[set storage-container near-storage]
;print who

]
]
]
ask households[

```

```

ifelse (storage-container = 1) ;If B or closed type storage container then this
sets coliform incubation growth rate
[
;B or Closed Type Storage Containers
set river-rand-number 3 ;Coliform incubation growth for River is
located in position 3 in that vector
set pipe-rand-number random 2 + 2 ;Coliform incubation growth for Pipe is
located in positions 2,3 in that vector
]
[
;If C or Open Type Storage Containers then use
different rate
set river-rand-number random 3 ;Coliform incubation growth for River is
located in positions 0,1,2 in that vector
set pipe-rand-number random 2 ;Coliform incubation growth for Pipe is
located in positions 0,1 in that vector
]
]

ask households [
if jc_total = 99 [ ;Basically, if no jc_total has been defined yet
let iii 0
let found-nearby-house 0 ;Tags to specify whether or not a nearby house has
been found.

while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
searching outward radially
[
let near-storage 999 ;re-setting the "near" sources to 999
ask households in-radius iii
[
if (jc_total != 99 and who < 49);If nearby HHB house has legit storage
container tag use that data to set near-storage
[
set near-storage jc_total
set found-nearby-house 1
]
]
]
set iii iii + 1
if jc_total = 99 and found-nearby-house = 1 ;Break out of the previous
search and set the formally unspecified house to be a nearby house
[set jc_total near-storage]
]
]

ask households [
if cup_total = 99 [ ;Basically, if no cup_total has been defined yet
let iii 0
let found-nearby-house 0 ;Tags to specify whether or not a nearby house has
been found.

while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
searching outward radially
[
let near-storage 999 ;re-setting the "near" sources to 999
ask households in-radius iii
[
if (cup_total != 99 and who < 49);If nearby HHB house has legit storage
container tag use that data to set near-storage
[
set near-storage cup_total
set found-nearby-house 1
]
]
]
set iii iii + 1
if cup_total = 99 and found-nearby-house = 1 ;Break out of the previous
search and set the formally unspecified house to be a nearby house
[set cup_total near-storage ]
;print who
]
]

```

```

]
]

ask households [
if bhw_total = 99 [ ;Basically , if no cup_total has been defined yet
let iii 0
let found-nearby-house 0 ;Tags to specify whether or not a nearby house has
been found.

while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
searching outward radially
[
let near-storage 999 ;re-setting the "near" sources to 999
ask households in-radius iii
[
if (bhw_total != 99 and who < 49);If nearby HHB house has legit storage
container tag use that data to set near-storage
[
set near-storage bhw_total
set found-nearby-house 1
]
]
]
set iii iii + 1
if bhw_total = 99 and found-nearby-house = 1 ;Break out of the previous
search and set the formally unspecified house to be a nearby house
[set bhw_total near-storage ]

]
]

ask households [
if clean-min = 99 [ ;Basically , if no cup_total has been defined yet
let iii 0
let found-nearby-house 0 ;Tags to specify whether or not a nearby house has
been found.

while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
searching outward radially
[
let near-storage 999 ;re-setting the "near" sources to 999
let near-storage2 999
ask households in-radius iii
[
if (clean-min != 99 and who < 49);If nearby HHB house has legit storage
container tag use that data to set near-storage
[
set near-storage clean-min
set near-storage2 clean-max
set found-nearby-house 1
]
]
]
set iii iii + 1
if clean-min = 99 and found-nearby-house = 1 ;Break out of the previous
search and set the formally unspecified house to be a nearby house
[
set clean-min near-storage
set clean-max near-storage2
]
]
]

```



```

    ]
  ]
]

if container-test = TRUE[ ;If in container-testing mode then
; let who-count 0

  ask households[
    ifelse random 100 < closed-percent [set storage-container 1][set storage-
      container 2]
  ]
]

if jc-test = TRUE[
  ask households [set jc_total jc_total_test] ;Sets jc_total for all HH to be that
    which is set in the Interface if in jc-test mode
]
if cup-test = TRUE[
  ask households [set cup_total cup_total_test] ;Sets cup_total for all HH to be that
    which is set in the Interface if in cup-test mode
]

end

to setup-haz

  set ecdage-list [ ]
  file-open "ecdage_jul_data_4mths.csv" ;File with codes for
  while [not file-at-end?]
  [set ecdage-list lput file-read ecdage-list]
  file-close

  let final-list [ ]
  file-open "final_jul_data_4mths.csv" ;File with codes for
  while [not file-at-end?]
  [set final-list lput file-read final-list]
  file-close

  let haz-list list ecdage-list final-list

  set haz-table table:make

  (foreach ecdage-list final-list
  [
    table:put haz-table ?1 ?2
  ])

  set ecdage-list-double [ ]
  file-open "ecdage_jul_data_4mths_double_cases.csv" ;File with codes for
  while [not file-at-end?]
  [set ecdage-list-double lput file-read ecdage-list-double]
  file-close

  let final-list-double [ ]
  file-open "final_jul_data_4mths_double_cases.csv" ;File with codes for
  while [not file-at-end?]
  [set final-list-double lput file-read final-list-double]
  file-close
  let haz-list-double list ecdage-list-double final-list-double
  set haz-table-double table:make

  (foreach ecdage-list-double final-list-double
  [
    table:put haz-table-double ?1 ?2
  ])

```

```

    ])

end

to setup-hh-water-sources2

  let pri-water-list [ ]
  file-open "hhb_water_sources.txt" ;File with codes for the primary water sources
    of the HHB water sources 1=River, 2=Piped, 3=MT, 4=Hose
  while [not file-at-end?]
  [set pri-water-list lput file-read pri-water-list]
  file-close

  let sec-water-list [ ]
  file-open "hhb_second_water_sources.txt"
  while [not file-at-end?]
  [set sec-water-list lput file-read sec-water-list] ;File with codes for the secondary
    water sources of the HHB water sources 1=River, 2=Piped, 3=MT, 4=Hose
  file-close

  ask households [

    if who < 49 ;For all of the HHB households
    [
      set pri-water-source item who pri-water-list ;Set primary and secondary
        sources to be what the HHB study measured, I am omitting Ceramic filter data
        because it isn't that accurate, specifically, they seem to have mis-
        interpreted household taps and municipal taps
      set sec-water-source item who sec-water-list
    ]
  ]

  ask households [

    if who > 49[
      let i 0

      loop
      [
        let nearest-household min-n-of i other households
        [distance myself]

        set pri-water-source [pri-water-source] of nearest-household
        ;print pri-water-source
        if [pri-water-source] of nearest-household != 0 [stop]
        set i i + 1
      ]
    ]
  ]

end

to setup-hh-water-sources

  let pri-water-list [ ]
  file-open "hhb_water_sources.txt" ;File with codes for the primary water sources
    of the HHB water sources 1=River, 2=Piped, 3=MT, 4=Hose
  while [not file-at-end?]
  [set pri-water-list lput file-read pri-water-list]
  file-close

  file-open "ceramic_primary_water_sources.txt" ;File with codes for the primary water
    sources of the Ceramic water sources 1=River, 2=Piped, 3=MT, 4=Hose
  while [not file-at-end?]
  [set pri-water-list lput file-read pri-water-list]
  file-close
  ;print length pri-water-list

```

```

let sec-water-list [ ]
file-open "hhb_second_water_sources.txt"
while [not file-at-end?]
[set sec-water-list lput file-read sec-water-list] ;File with codes for the secondary
water sources of the HHB water sources 1=River, 2=Piped, 3=MT, 4=Hose
file-close

file-open "ceramic_second_water_sources.txt"
while [not file-at-end?]
[set sec-water-list lput file-read sec-water-list]
file-close

ask households [
set pri-water-source 999 ;999 is a number I assigned in the above text files and
here to indicate that no source has been assigned to a given household
set sec-water-source 999

if who < 49 ;For all of the HHB households
[
set pri-water-source item who pri-water-list ;Set primary and secondary
sources to be what the HHB study measured, I am omitting Ceramic filter data
because it isn't that accurate, specifically, they seem to have mis-
interpreted household taps and municipal taps
set sec-water-source item who sec-water-list
]
]
;ask households [print who]
ask households [

if pri-water-source = 999 or sec-water-source = 999 [ ;Effectively, look at all
households with pri or sec sources as yet unspecified
let iii 0
let found-nearby-house 0 ;Tags to specify whether or not a nearby house has
been found.
let found-nearby-house2 0

while [found-nearby-house = 0] ;while no nearby HHB house has been found keep
searching outward radially
[
let near-pri-water-source 999 ;re-setting the "near" sources to 999
ask households in-radius iii
[
;I am
not letting 26 and 20 determine nearby houses because 26 uses the river
and I am confident that the houses nearby probably don't use this source
, also excluding 20 because they reported using the Pipe system which is
not in the part of Tshibvumo where they are.
if pri-water-source > 0 and pri-water-source < 5 and found-nearby-house =
0 and who < 49 and who != 26 and who != 20 ;If nearby HHB house has
legit source use that data to set near-pri
[
set near-pri-water-source pri-water-source
set found-nearby-house 1
]
]
set iii iii + 1
if pri-water-source = 999 and found-nearby-house = 1 ;Break out of the
previous search and set the formally unspecified house to be a nearby
house
[set pri-water-source near-pri-water-source]
]

set iii 0 ;Same loop as above, just for the
secondary sources

while [found-nearby-house2 = 0]
[
let near-sec-water-source 999
ask households in-radius iii
[
if sec-water-source > 0 and sec-water-source < 5 and found-nearby-house2 =
0 and who < 49
[

```

```

        set near-sec-water-source sec-water-source
        ;set color 46
        set found-nearby-house2 1
    ]
]
set iii iii + 1
if sec-water-source = 999 and found-nearby-house2 = 1
    [set sec-water-source near-sec-water-source]
]

]
set today-source pri-water-source
]

if source-scenario-test = TRUE [ ;If in source-testing mode then setup houses to have
    one of four different sources according to Interface page
    let who-count 0

    if source-scenario = 1 [set river-percent 0 set pipe-percent 100 / 3 set mt-percent
        100 / 3 set hose-percent 100 / 3]
    if source-scenario = 2 [set river-percent 25 set pipe-percent 25 set mt-percent 25
        set hose-percent 25]
    if source-scenario = 3 [set river-percent 50 set pipe-percent 50 / 3 set mt-percent
        50 / 3 set hose-percent 50 / 3]
    if source-scenario = 4 [set river-percent 75 set pipe-percent 25 / 3 set mt-percent
        25 / 3 set hose-percent 25 / 3]
    if source-scenario = 5 [set river-percent 100 set pipe-percent 0 set mt-percent 0
        set hose-percent 0]
    if source-scenario = 6 [set pipe-percent 0 set river-percent 100 / 3 set mt-percent
        100 / 3 set hose-percent 100 / 3]
    if source-scenario = 7 [set pipe-percent 50 set river-percent 50 / 3 set mt-percent
        50 / 3 set hose-percent 50 / 3]
    if source-scenario = 8 [set pipe-percent 75 set river-percent 25 / 3 set mt-percent
        25 / 3 set hose-percent 25 / 3]
    if source-scenario = 9 [set pipe-percent 100 set river-percent 0 set mt-percent 0
        set hose-percent 0]
    if source-scenario = 10 [set mt-percent 0 set pipe-percent 100 / 3 set river-percent
        100 / 3 set hose-percent 100 / 3]
    if source-scenario = 11 [set mt-percent 50 set pipe-percent 50 / 3 set river-percent
        50 / 3 set hose-percent 50 / 3]
    if source-scenario = 12 [set mt-percent 75 set pipe-percent 25 / 3 set river-percent
        25 / 3 set hose-percent 25 / 3]
    if source-scenario = 13 [set mt-percent 100 set pipe-percent 0 set river-percent 0
        set hose-percent 0]
    if source-scenario = 14 [set hose-percent 0 set river-percent 100 / 3 set mt-percent
        100 / 3 set pipe-percent 100 / 3]
    if source-scenario = 15 [set hose-percent 50 set river-percent 50 / 3 set mt-percent
        50 / 3 set pipe-percent 50 / 3]
    if source-scenario = 16 [set hose-percent 75 set river-percent 25 / 3 set mt-percent
        25 / 3 set pipe-percent 25 / 3]
    if source-scenario = 17 [set hose-percent 100 set river-percent 0 set mt-percent 0
        set pipe-percent 0]

ask households [

    ;Setting up households to have one of four source according to the percentages on
    the main page. using who-count to go through the list so the houses are
    randomized
    if who-count < river-percent / 100 * 410 [set pri-water-source 1] ;River
    if who-count >= river-percent / 100 * 410 and who-count < pipe-percent / 100 * 410
        + river-percent / 100 * 410 [set pri-water-source 2] ;Pipe
    if who-count >= pipe-percent / 100 * 410 + river-percent / 100 * 410 and who-count
        < pipe-percent / 100 * 410 + river-percent / 100 * 410 + mt-percent / 100 *
        410 [set pri-water-source 3] ;MT
    if who-count >= pipe-percent / 100 * 410 + river-percent / 100 * 410 + mt-percent
        / 100 * 410 and who-count < pipe-percent / 100 * 410 + river-percent / 100 *
        410 + mt-percent / 100 * 410 + hose-percent / 100 * 410 [set pri-water-source
        4] ;Hose
    set who-count who-count + 1
    if pri-water-source = 4 [set color 14] ;Hose Red
    if pri-water-source = 1 [set color 84] ;River Blue
    if pri-water-source = 2 [set color 4] ;Piped Grey
    if pri-water-source = 3 [set color 54] ;MT Green
    set today-source pri-water-source
]
]

```

```

ask households [
  if who-count < river-percent / 100 * 410 [set sec-water-source 1]
  if who-count >= river-percent / 100 * 410 and who-count < pipe-percent / 100 * 410
    + river-percent / 100 * 410 [set sec-water-source 2]
  if who-count >= pipe-percent / 100 * 410 + river-percent / 100 * 410 and who-count
    < pipe-percent / 100 * 410 + river-percent / 100 * 410 + mt-percent / 100 *
    410 [set sec-water-source 3]
  if who-count >= pipe-percent / 100 * 410 + river-percent / 100 * 410 + mt-percent
    / 100 * 410 and who-count < pipe-percent / 100 * 410 + river-percent / 100 *
    410 + mt-percent / 100 * 410 + hose-percent / 100 * 410 [set sec-water-source
  4]
]

]

ask households [

  if (pri-water-source = 1);River
    [set daily-wq item random length surfwq surfwq]

  if (pri-water-source = 2 or pri-water-source = 4) ;Piped or Hose
    [set daily-wq item random length pipewq pipewq]

  if (pri-water-source = 3);MT
    [set daily-wq item random length mtwq mtwq]

]

end

to make-height-array
  let height-list [ ]
  file-open "boyheight.txt" ;;open txt file
  containing WHO SD0 male height scores by day
  while [not file-at-end?]
  [set height-list lput file-read height-list] ;;looks at txt file and
  places new number at the end of the list using lput
  file-close
  set height-array array:from-list n-values 1833 [item ? height-list] ;;covert list
  to height-array for access during height calculation

  let feight-list [ ]
  file-open "femaleheight.txt" ;;open txt file
  containing WHO SD0 female height scores by day
  while [not file-at-end?]
  [set feight-list lput file-read feight-list] ;;looks at txt file and
  places new number at the end of the list using lput
  file-close
  set feight-array array:from-list n-values 1833 [item ? feight-list] ;;covert list
  to feight-array for access during height calculation

  let malfeight-list [ ]
  file-open "maled_girl_growth.txt" ;;open txt
  file containing WHO SD0 female height scores by day
  while [not file-at-end?]
  [set malfeight-list lput file-read malfeight-list] ;;looks at txt file
  and places new number at the end of the list using lput
  file-close
  set malfeight-array array:from-list n-values 730 [item ? malfeight-list] ;;covert
  list to feight-array for access during height calculation

  let malheight-list [ ]
  file-open "maled_boy_growth.txt" ;;open txt file
  containing WHO SD0 female height scores by day
  while [not file-at-end?]
  [set malheight-list lput file-read malheight-list] ;;looks at txt file
  and places new number at the end of the list using lput
  file-close

```

```

set malheight-array array:from-list n-values 730 [item ? malheight-list] ;; covert
list to feight-array for access during height calculation

let water-usage-list []
file-open "maled_water_usage-percents.csv"
while [not file-at-end?]
[set water-usage-list lput file-read water-usage-list]
file-close
set water-usage-array array:from-list n-values 731 [item ? water-usage-list]

end

to do-plots

set-current-plot "Med WQ"
set-current-plot-pen "Med WQ"
plot median [daily-wq] of households
set-current-plot "Mean WQ"
set-current-plot-pen "Mean WQ"
plot mean [daily-wq] of households

end

```

Bibliography

Mellor, J., Smith, J.A., Learmonth, G.P., Netshandama, V.O., Dillingham, R.A., 2012. Modeling the complexities of water, hygiene, and health in Limpopo Province, South Africa. *Environmental Science & Technology* 46, 13512–13520.

Mellor, J., Smith, J.A., Samie, A., Dillingham, R.A., 2013. Coliform Sources and Mechanisms for Regrowth in Household Drinking Water in Limpopo, South Africa. *Journal of Environmental Engineering* 139, 1152–1161.