Supplementary Documentation

Conceptualization of the Morbidity Ratio Proxy Derivation

The premise is that the probability of infection dose response model is not representative of disparate health outcomes within a diverse population. This is due to the inherent limitations of the mechanistic dose response models' derivation. The dose response models are derived by starting with a dose of pathogens that reach an infectious location (*j*) from an average Poisson distributed exposed dose of a pathogens (*d*). From this Poisson dose, there is a binomial probability that *k* number of pathogens will survive to initiate an infection. The dose response derivation is shown again in equation S.1, where the likelihood of an infectious response from the original average exposed dose *d* of pathogens is demonstrated resulting in the exponential dose-response model. As can be seen from equation S.1, there are no host-side influences or dynamics. While this provides for a generalized and adaptable model, capable of representing multiple host animal species, it does not account for host-side influences, and therefore cannot represent disparate health outcomes as desired in this research.

$$p(d) = \sum_{k=k_{min}}^{\infty} \frac{(d \cdot r)^2 e^{-d \cdot r}}{k!} \sum_{j=m}^{\infty} \frac{\left[d(1-r)^{j-k} e^{-d(1-r)}\right]}{(j-k)!} \rightarrow$$

$$p(d) = \sum_{k=k_{min}}^{\infty} \frac{(d \cdot r)^k e^{-d \cdot r}}{k!} \cong p(d) = 1 - \exp(-\mathrm{k} \cdot \mathrm{d}) \quad \text{(S.1)}$$

We have developed a morbidity ratio proxy that can use data from Legionellosis – a set of two illness outcomes from a realized probability of infection. Therefore, if we represent that within a total population *T*, there is a specific demographic group *A* (*e.g.* elderly population) where we can calculate the following.

1. For A there is a population P_A within which exists a total number of cases C_A
2. For T there is a population P_T and there exists a total number of cases C_T
3. For T there exists a non-specific attack rate AR_T which is the rate of illness given those within *T* which were exposed to *Legionella pneumophila*
4. For *A* within *T* there is a ratio of cases C_A:C_T in which *A* cases are relative to total cases in *T*
5. For *A* within *T* there is a ratio of populations P_A:P_T in which P_A is relative to the total population P_T
6. For *A* within *T* the relative rate of illness in *A* given illness in *T* (R_A) can be calculated as relative ratios using equation S.2

$$R_A = \frac{C_A/P_A}{C_T/P_T} \quad \text{(S.2)}$$

7. Therefore, given *3* above, for *A* within *T* we can use equation S.3 to calculate a morbidity ratio proxy for *A* ($\widetilde{MR}_A$) using a national attack rate AR_T. In equation S.3 the product is taken since AR_T will increase as the relative rate of illness in *A* increases.

$$\widetilde{MR}_A = R_A \cdot AR_T \quad \text{(S.3)}$$

8. Therefore, since we know that the probability of infection (P_{inf}) is defined a function of the average exposed dose *d*, and parameter *k* then equation S.4 will calculate the probability of illness (P_{ill}) for *A* within *T* given exposure and probability of infection after exposure to *d* and knowing parameter *k*. From this P_{ill} we can then calculate DALYs using associated WHO disability weights.

$$P_{ill} = P_{inf} \cdot \widetilde{MR}_A \quad \text{(S.4)}$$

Risk Simulation Results for 6.6 and 9.0 L min^{-1} Flow Rates

This supplementary information contains the results from all of the flow rates that were not presented in the main manuscript. As can be seen by comparing the trends of the results in the images below, they are simulating the same delineation in risks between demographics groups.

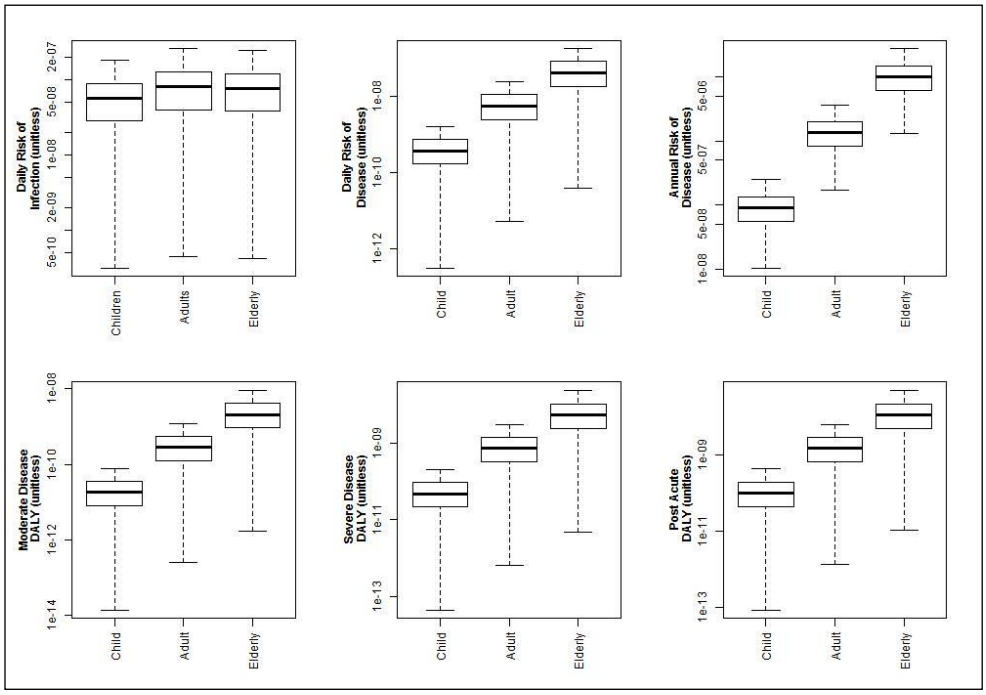

Figure S.1. Probability of infection, illness and DALY values for age groups at 6.1 L•min^{-1} flow rate

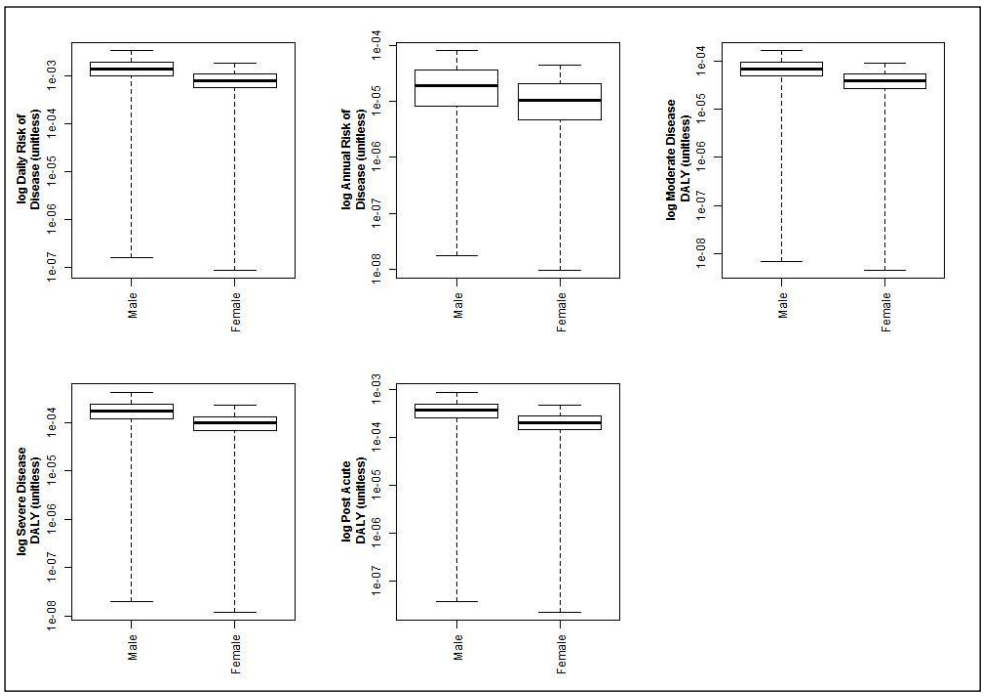

Figure S.2. Probability of infection, illness and DALY values for sex groups at 6.1 L•min^{-1} flow rate

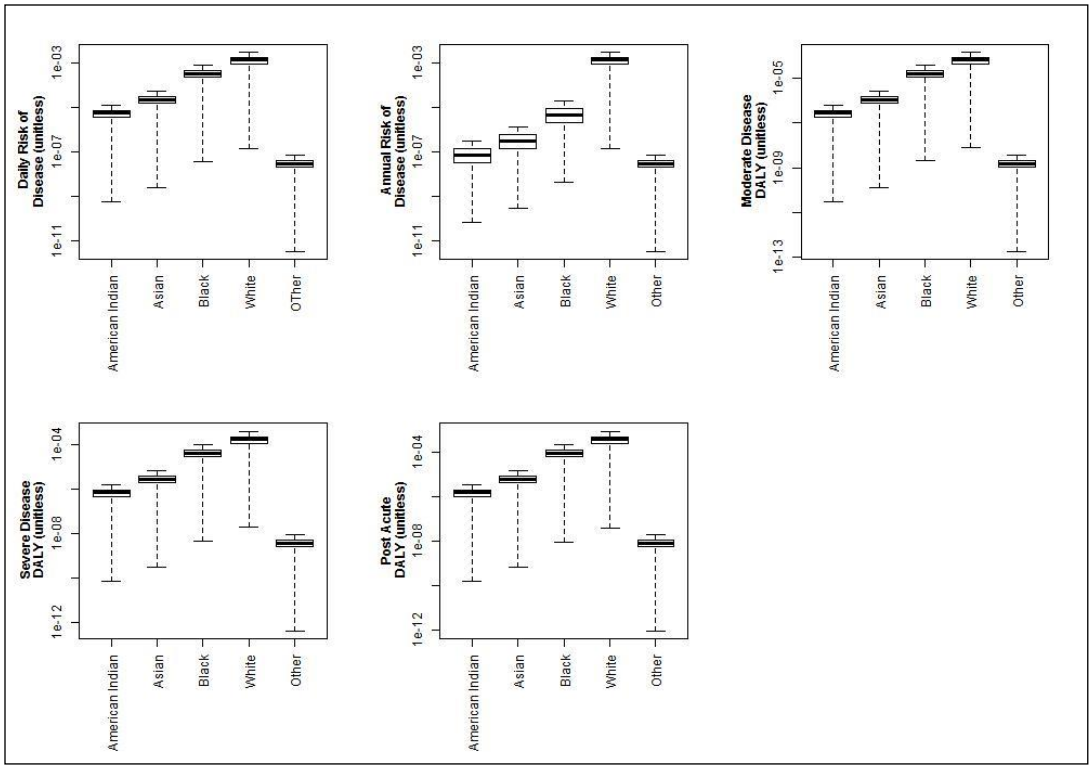

Figure S.3. Probability of infection, illness and DALY values for racial groups at 6.1 L•min^{-1} flow rate

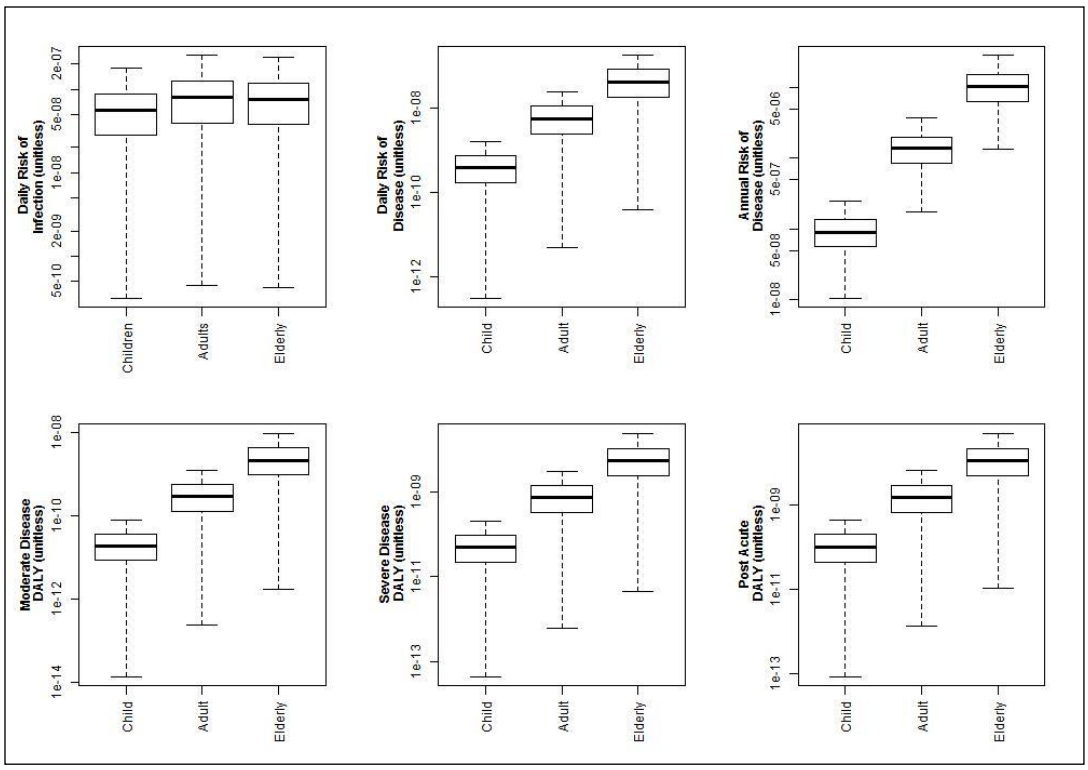

Figure S.4. Probability of infection, illness and DALY values for age groups at 9.0 L•min^{-1} flow rate

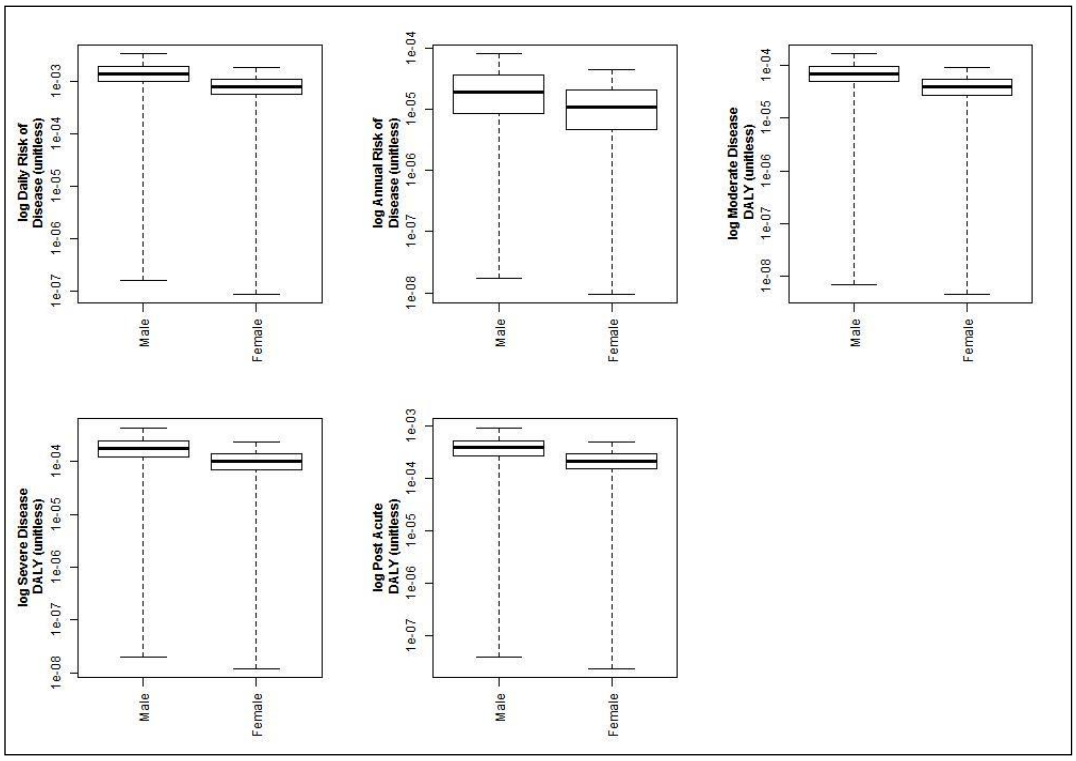

Figure S.5. Probability of infection, illness and DALY values for sex groups at 9.0 L•min^{-1} flow rate

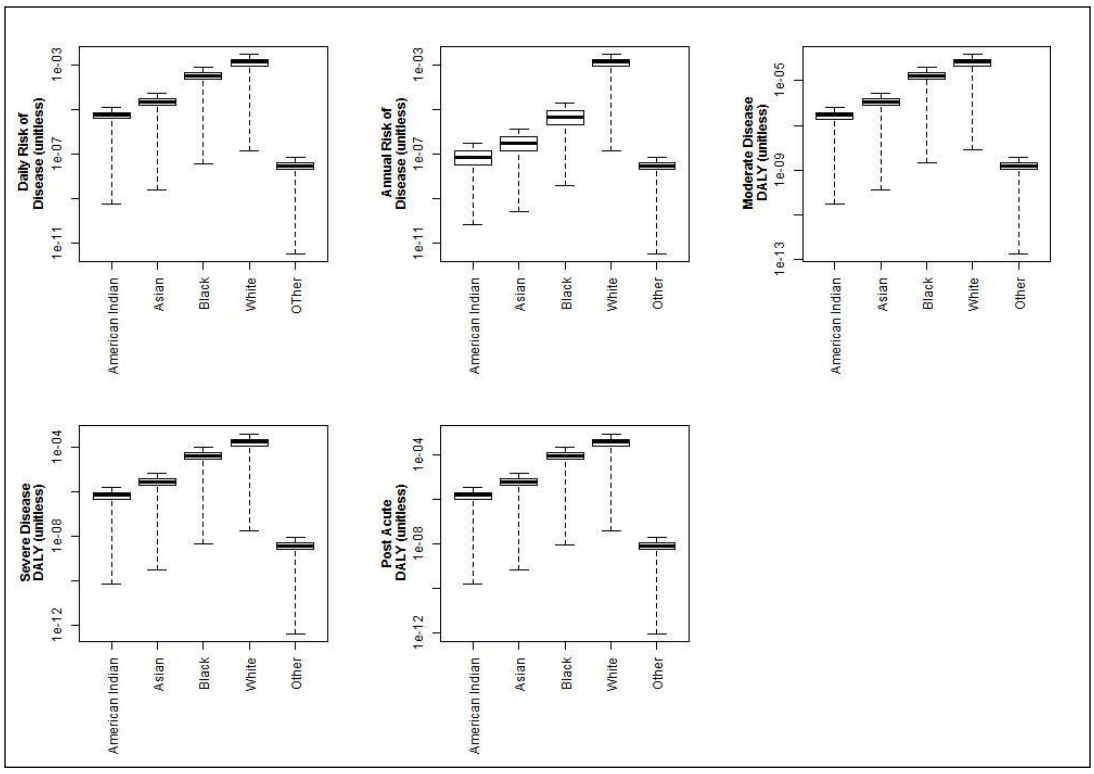

Figure S.6. Probability of infection, illness and DALY values for racial groups at 9.0 L•min^{-1} flow rate

AIC Values

As mentioned in the text the Akaike Information Criterion weights (AICw) values are presented here. Table S.1 shows the AIC and AICw values for the distributions where probability distribution optimization was possible and performed. Only the best performing distributions are shown so as to not clutter the table.

Table S.1. AIC values and AICw for each of the variables where probability distributions were optimized.

Probability Distribution	AIC (unitless)	AIC Weights (unitless)
Flow Rate #1 Cold Water		
normal	49.23	0.21
lognormal	48.38	0.33
Weibull	48.26	0.35
exponential	62.09	3.46 (10^{-3})
logistic	50.64	0.11
Cauchy	58.42	2.17 (10^{-2})
Flow Rate #1 Hot Water		
normal	-1.97	0.01
lognormal	-10.03	0.52
Weibull	-7.90	0.18
exponential	-8.40	0.23
logistic	-3.31	0.02
Cauchy	-4.76	0.04

Probability Distribution	AIC (unitless)	AIC Weights (unitless)
Flow Rate #2 Cold Water		
normal	48.72	0.29
lognormal	50.48	0.12
Weibull	47.82	0.46
exponential	72.88	$1.67(10^{-6})$
logistic	50.48	0.12
Cauchy	58.09	$2.71\ (10^{-2})$
Flow Rate #2 Hot Water		
normal	10.44	0.02
lognormal	3.57	0.53
Weibull	5.51	0.20
exponential	5.56	0.20
logistic	9.67	0.03
Cauchy	9.57	0.02
Flow Rate #3 Cold Water		
normal	1.18	0.29
lognormal	0.71	0.37
Weibull	1.97	0.19
exponential	43.50	$1.86\ (10^{-10})$
logistic	2.53	0.15
Cauchy	9.29	$4.50\ (10^{-2})$
Flow Rate #3 Hot Water		
normal	-6.68	0.01
lognormal	-14.56	0.65
Weibull	-10.80	0.10
exponential	-8.44	0.03
logistic	-9.03	0.04
Cauchy	-11.89	0.17
Pulmonary Deposition Rate		
normal	-13.34	0.29
lognormal	-12.01	0.15
Weibull	-14.13	0.43
exponential	2.93	$8.47\ (10^{-5})$
logistic	-11.75	0.13
Cauchy	-3.71	$2.35\ (10^{-2})$
Tracheobronchial Deposition		
normal	-16.16	0.16
lognormal	-2.31	$1.52\ (10^{-4})$
Weibull	-19.05	0.66
exponential	-7.84	$2.42\ (10^{-2})$
logistic	-16.18	0.16
Cauchy	-12.73	0.028
Upper Respiratory Tract Deposition		
normal	-13.20	0.02
lognormal	-18.62	0.25
Weibull	-18.13	0.19
exponential	-20.13	0.52

Probability Distribution	AIC (unitless)	AIC Weights (unitless)
logistic	-12.55	0.01
Cauchy	-11.60	0.01

An Advanced Risk Modeling Method to Estimate Legionellosis Risks Within a Diverse Population During Showering Exposures

R code for QMRA model

Mark H. Weir Ph.D. March 29, 2019

The following annotated R script is the source code used for model development and evaluation of the morbidity ratio proxy developed in this research. This source code will automatically model the risks for each of the three flow rates, create directories for each flow rate to save plots and tables to.

```
# Header ------------------------------------------------------------

#=============================================================================================================|

# 2D_Sim_Legionella_Case_Study_V6.R is an R (www.r-project.org) source code that operates a 2-dimensional      |
# simulation for Quantitative Microbial Risk Assessment (QMRA; qmra.cph.ohio-state.edu) modelling. In           |
# this case we are modelling the risk of Legionella pneumophila infection and resulting Disability             |
# Adjusted Life Years (DALYs) from this illness. There is an attempt in modelling a DALY for Pontiac           |
# Fever versus the more dangerous pneumonia form of Legionellosis.                                             |
# Multiple flow rates can be modeled in this code, therefore the line just below 2D_Simulation_Loops           |
# section title, a line Qnum <- can be seen. This is where the user chooses from 1 of 3 flow rates             |
# The code is broken into sections due to length to allow for easier navigation sections can be easily opened  |
# and closed using RStudio which is a programming GUI for R and recommended for use of the code.               |
#                                                                                                              |

# Coding and model redevelopment performed by Mark H. Weir Ph.D. of CAMRA Consultants LLC, NSF International   |
# and College of Public Health at The Ohio State University                                                    |

#                                                                                                              |

# All use and reproduction rights are reserved by Mark H. Weir Ph.D. and CAMRA Consultants LLC.                |
# CAMRA Consultants LLC. weirmarkh@gmail.com, camraconsultants@gmail.com; weir.95@osu.edu                      |
#==============================================================================================================
=========================|

#=============================================================================================================
|

# Set working directory and install required packages if not installed                                         |

this.dir <- dirname(parent.frame(2)$ofile)        # Will not work with line execution needs to be sourced      #|
```

```
setwd(this.dir)                                                                                                    #|
if(rownames(installed.packages(reshape))==FALSE){install.packages("reshape")}else{require(reshape)}                #|
if(rownames(installed.packages(ggplot2))==FALSE){install.packages("ggplot2")}else{require(ggplot2)}
if(rownames(installed.packages(grid))==FALSE){install.packages("grid")}else{require(grid)}
if(rownames(installed.packages(gridExtra))==FALSE){install.packages("gridExtra")}else{require(gridExtra)}

#=============================================================================================================
|



# Source bounded forms of chosen distributions - syntax for all is trDist(n,param1,param2,LB,UB), where          |#
param1 is parameter 1, param2 is parameter 2, LB is lower bound, and UB is upper bound                            |
# Each of these source codes are developed and written by Mark H. Weir Ph.D. see primary header for contact |source("trlogis.R");
source("trweibull.R");          source("tLnormal.R");          source("TriRand.r");          source("tnormal.r")      #|
#=============================================================================
=========================|

#=============================================================================================================
======|

# Risk annualization function, adapted from code provided by Kerry Hamilton Ph.D. Drexel University (kh495@drexel.edu) |
#=============================================================================================================
=====|

d2A_n = 183 # Showering every other day

dailyRisks2AnnualRisks <- function(dailyRisks, numPerDay=1, numPerYear=d2A_n){
  calcAnnualRisk <- function(dailyRiskCol){
    sampledRisks = rep(sample(dailyRiskCol,numPerYear, replace=TRUE),numPerDay)
    annualRisk = 1-prod(1-sampledRisks)
    return(annualRisk)

  }
  annualRisks <- c()
  for(i in 1:length(dailyRisks))

    annualRisks[i] = calcAnnualRisk(dailyRisks) return(annualRisks)

}
#=============================================================================================================|

# First for loop is to run the code for all flow rates 1 is flow rate 1 (5.1 L/min) 2 is flow rate 2 (6 L/min) |
# 3 flow rate 3 (9 L/min)                                                                                      |

#=============================================================================
============================|      # Set variables and iterations ----------------------------------------------
-------------
```

```
for (q in 1:3) # This allows for the modeling of all three flow rates individually with one code execution.
{
  # ensure that the working directory is set to a central location so that next flow rate does not overwrite previous

  setwd(this.dir)

  #=============================================================================================================|

  # Set up the number of outer and inner loops for the 2-Dimensional Simulation and set the seed. Testing   |      #
  of the model will set the inner to 1000 and outer to 101. However, for an offical run for reporting or  | #
  publication will require the 10000 and 1001 inner and outer iterations to ensure that the Law of Large |      #
  numbers will be envoked within each of the outer iterations.                                             |

  #                                                                                                        |

  outer = 1001                    # Outer is for the outer loop that is for the variable variables         |
  inner = 10000                   # Inner is for the inner loop that is for the uncertain variables        |
  set.seed(36)                  # Set the seed for reproducible research                                 |
  #=============================================================================
  ======================|



  #=============================================================================
  ==========================|     # Set_Vectors_For_Variables ----------------------------------------
  ----------------

  # g is for daily risk (DRisk), h is for annualized risk (ARisk)                                          |
  #=============================================================================
  ======================|

  gC <- matrix(nrow=inner, ncol=outer); hC <- matrix(nrow=inner, ncol=outer);

  gA <- matrix(nrow=inner, ncol=outer); hA <- matrix(nrow=inner, ncol=outer); gE

  <- matrix(nrow=inner, ncol=outer); hE <- matrix(nrow=inner, ncol=outer); Pop <-

  matrix(nrow=inner,ncol=outer); PopA <- matrix(nrow=inner,ncol=outer)

  AnnStiched <- matrix(nrow = inner,ncol=outer); ARisks <- matrix(nrow=inner,ncol=1);
  ARisksStiched <- matrix(nrow=inner,ncol=outer)

  #=============================================================================
  ==========================|     # Set Variable value or matrices -------------------------------------
  -------------

  # Set up your matrices for variables - These can also be numeric vectors as seen below                   |
  # For those with (outer) defining the number of rows those are variable variables in the outer loop      |
  # For those with (inner) defining the number of rows those are uncertain variables in the inner loop     |      #
                                                                                                           |
```

```
# Variable variables                                                                                    |
 Inh_C <- numeric(outer)       # Inhalation rate, children (m^3/min) from Exposure Factors Handbook (REF)  |
 Inh_A <- numeric(outer)       # Inhalation rate, adults (m^3/min)   from Exposure Factors Handbook (REF)  |
 Inh_E  <- 0.012               # Inhalation rate, elderly (m^3/min)   from Exposure Factors Handbook (REF) |
 Frac_aero_Q1 <- numeric(outer) # Frac_aero is the fraction of aerosolized organisms in the range          |
                                 # of  1-5 microns, unitless at flow rate 1 (REF)                         |

Frac_aero_Q2 <- numeric(outer) # Frac_aero is the fraction of aerosolized organisms in the range          |
                                # of 1-5 microns, unitless at flow rate 2 (REF)                          |

Frac_aero_Q3 <- numeric(outer) # Frac_aero is the fraction of aerosolized organisms in the range          |
                               # of 1-5 microns, unitless at flow rate 3 (REF)                           |

ASFrac_Q1_cold <- numeric(outer) # Aerosol size fraction flow rate 1 cold water                          |

ASFrac_Q2_cold <- numeric(outer) # Aerosol size fraction flow rate 1 cold water                          |

ASFrac_Q3_cold <- numeric(outer) # Aerosol size fraction flow rate 1 cold water                          |

ASFrac_Q1_hot <- numeric(outer) # Aerosol size fraction flow rate 1 cold water                           |

ASFrac_Q2_hot <- numeric(outer) # Aerosol size fraction flow rate 1 cold water                           |

ASFrac_Q3_hot <- numeric(outer) # Aerosol size fraction flow rate 1 cold water                           |

#                                                                                                       |

Hotmix <- numeric()           # Percentage of Hot Water to mix with Cold Water for desired final temp    |

Coldmix <- numeric()          # Same as for hot water but percentage for Cold Water                      |

Temp <- numeric()             # Final water temperature after mixing                                     |

SDur <- numeric(outer)        # Shower duration (hrs)                                                    |


LPart <- numeric(outer)     # Legionella partition coefficient resulting units L/m^3 (REF)              |
Tc <- matrix(nrow=outer,ncol=1)    # Cold Water temperatures for the USA                                  |
Th <- matrix(nrow=outer,ncol=1)    # Hot Water temperatures, assumed variable variable                    |
#                                                                                                       |

#-------------------------------------------------------------------------------------------------------|

#                                                                                                       |

# R1 = Region 1 = Upper Resp Tract; R2 = Region 2 = Tracheobroncheal; R3 = Region 3 = Pulmonary         |
#                                                                                                       |

C_airQ1 <- numeric(inner)     # Concentration in the air considering aerosol fraction and partition coef |
                              # at flow rate  #1 CFU/L                                                  |
```

```
C_airQ2 <- numeric(inner)    # Concentration in the air considering aerosol fraction and partition coef
                             # at flow rate  #2 CFU/L

C_airQ3 <- numeric(inner)    # Concentration in the air considering aerosol fraction and partition coef
                             # at flow rate  #3 CFU/L

dose_IQ1C <- numeric(inner) # Inhaled dose at flow rate 1 for children (CFU)
dose_IQ2C <- numeric(inner) # Inhaled dose at flow rate 2 for children (CFU)
dose_IQ3C <- numeric(inner) # Inhaled dose at flow rate 3 for children (CFU)
dose_IQ1A <- numeric(inner) # Inhaled dose at flow rate 1 for adults (CFU)
dose_IQ2A <- numeric(inner) # Inhaled dose at flow rate 2 for adults (CFU)
dose_IQ3A <- numeric(inner) # Inhaled dose at flow rate 3 for adults (CFU)
dose_IQ1E <- numeric(inner) # Inhaled dose at flow rate 1 for elderly (CFU)
dose_IQ2E <- numeric(inner) # Inhaled dose at flow rate 2 for elderly (CFU)
dose_IQ3E <- numeric(inner) # Inhaled dose at flow rate 3 for elderly (CFU)
#
dose_dQ1C <- numeric(inner) # dose delivered to region 3 in children flow rate 1

dose_dQ1A <- numeric(inner) # dose delivered to region 3 in children flow rate 1

dose_dQ1E <- numeric(inner) # dose delivered to region 3 in children flow rate 1

dose_dQ2C <- numeric(inner) # dose delivered to region 3 in adults flow rate 2

dose_dQ2A <- numeric(inner) # dose delivered to region 3 in adults flow rate 2

dose_dQ2E <- numeric(inner) # dose delivered to region 3 in adults flow rate 2

dose_dQ3C <- numeric(inner) # dose delivered to region 3 in elderly flow rate 3

dose_dQ3A <- numeric(inner) # dose delivered to region 3 in elderly flow rate 3

dose_dQ3E <- numeric(inner) # dose delivered to region 3 in elderly flow rate 3

#

FR1 <- numeric(inner)        # Deposition fraction in Region 1 (unitless) (data from REF)
FR2 <- numeric(inner)        # Deposition fraction in Region 2 (unitless) (data from REF)
FR3 <- numeric(inner)        # Deposition fraction in Region 3 (unitless) (data from REF)
CLeg <- numeric(inner)       # Legionella concentration (CFU/L) in tap water (data from REF)
k <- numeric(inner)          # Dose response model parameter (unitless) (REF)
DRiskC <- numeric(inner)   # DRisk the calculated risk using the exponential dose response model (REF)  | ARiskC
<- numeric(inner) # ARisk the annualized risk using the form 1-(1-DRisk)^n (REF)
DRiskA <- numeric(inner)     # DRisk the calculated risk using the exponential dose response model (REF) |



ARiskA <- numeric(inner)   # ARisk the annualized risk using the form 1-(1-DRisk)^n (REF)
DRiskE <- numeric(inner) # DRisk the calculated risk using the exponential dose response model (REF) |ARiskE
<- numeric(inner) # ARisk the annualized risk using the form 1-(1-DRisk)^n (REF)
PopDRisk <- matrix(nrow=inner, ncol=1) # Population risk, inclusion explusion of all ages daily risk | PopARisk
<- matrix(nrow=inner, ncol=1)  # Population risk, inclusion explusion of all ages annual risk  | #
```

```
LegDL <- 28                    # Legionella pneumophila detection limit in cold water (REF) point estimate |#-----
--   The following are Disability Weights for acute infectious disease (Haagsma et al, 2015)---------- |

DW_am_mu <- 0.051              # Mean for moderate infectious disease; used to simulate Pontiac Fever |DW_as_mu
<- 0.125                       #  Mean for severe infectious disease; used to simulate pneumonia       |
DW_pc_mu <- 0.217              #  Mean for post-acute consequences                                     |
DW_am_LB <- 0.039; DW_am_UB <- 0.06 # Lower and upper bounds for moderate infectious disease         |
DW_as_LB <- 0.104; DW_as_UB <- 0.152 # Lower and upper bounds for severe infectious disease          |
DW_pc_LB <- 0.179; DW_pc_UB <- 0.251 # Lower and upper bounds for post-acute consequences            |
#=============================================================================
=======================|


#=============================================================================
==========================|     # 2D Simulation ----------------------------------


#=============================================================================================================|

# Build folders for each of the flow rates being simulated q is the loop index and results to flow rate  |
# being modeled                                                                                           |

Qnum <- q
if(Qnum==1){Q <- "Flow_Rate_1";if(dir.exists(Q)==FALSE){dir.create(Q); setwd(Q)}else{setwd(Q)}

}else if(Qnum==2){Q <- "Flow_Rate_2";if(dir.exists(Q)==FALSE){dir.create(Q); setwd(Q)}else{setwd(Q)}

}else if(Qnum==3){Q <- "Flow_Rate_3";if(dir.exists(Q)==FALSE){dir.create(Q); setwd(Q)}else{setwd(Q)}

}else{stop("Please a Flow Rate in the Qnum scalar just after 2D_Simulation_Loops section header")}

for(i in 1:outer) #Variable variables (outer loop)

{

  #Inhalation rates

  Inh_C[i] <- TriRand(0.0076,0.0111,0.013); Inh_A[i] <- TriRand(0.012,0.0124,0.013); Inh_E <- 0.012
  # Children are 1 to <16 years old; Adults are 16 to < 61; Elderly are 61 to >= 81

  #Simulaed shower duration          Legionella partitioning coefficient

  SDur[i] <- runif(1,0.0333,0.25); LPart[i] <- runif(1,0.25,0.65)

  #Aerosol fractions

  ASFrac_Q1_cold[i] <- trweibull(1,2.75394690137046,0.030806465787483,0.00001,0.04772)
  ASFrac_Q2_cold[i] <- trweibull(1,4.12490211926184,0.0373445657837071,0.00001,0.04726)
```

```
ASFrac_Q3_cold[i] <- TriRand(0.009694,0.012755,0.017347)
ASFrac_Q1_hot[i] <- TriRand(0.000625,0.001250,0.006875)
ASFrac_Q2_hot[i] <- TriRand(0.001386,0.002718,0.012754)
ASFrac_Q3_hot[i] <- TriRand(0.001015,0.001523,0.006091)

#Set hot and cold mixtures for temperature range

Hotmix[i] <- rtnormal(1,0.577767833,0.039900197,0.5094,0.6472);
Coldmix[i] <- TriRand(0.3184,0.3652,0.5476)
CTemp <- runif(1,105,120)
Tc <- rlogis(1,58.3756026,5.3411813) Th
<- TriRand(105,110,120)
Temp[i] <- Hotmix[i]*120+Coldmix[i]*75

# Determine aerosol fraction based on hot and cold mixing

Frac_aero_Q1[i] <- (Hotmix[i]*ASFrac_Q1_hot[i])+(Coldmix[i]*ASFrac_Q1_cold[i])
Frac_aero_Q2[i] <- (Hotmix[i]*ASFrac_Q2_hot[i])+(Coldmix[i]*ASFrac_Q2_cold[i])
Frac_aero_Q3[i] <- (Hotmix[i]*ASFrac_Q3_hot[i])+(Coldmix[i]*ASFrac_Q3_cold[i])

for(j in 1:inner)     # Uncertain variables (inner loop)

{
  # Calculate concentrations in the air for each flow rate

  CLeg[j] <- runif(1,9.6E2,1E6); # Table 1 in manuscript for citation

  C_airQ1[j] <- CLeg[j]*LPart[i]*Frac_aero_Q1[i]
  C_airQ2[j] <- CLeg[j]*LPart[i]*Frac_aero_Q2[i]
  C_airQ3[j] <- CLeg[j]*LPart[i]*Frac_aero_Q3[i]

  # Calculate doses in the air for each flow rate
  dose_IQ1C[j] <- C_airQ1[j]*Inh_C[i]*(1/1000)*SDur[i]
  dose_IQ1A[j] <- C_airQ1[j]*Inh_A[i]*(1/1000)*SDur[i]
  dose_IQ1E[j] <- C_airQ1[j]*Inh_E*(1/1000)*SDur[i]
  dose_IQ2C[j] <- C_airQ2[j]*Inh_C[i]*(1/1000)*SDur[i]
  dose_IQ2A[j] <- C_airQ2[j]*Inh_A[i]*(1/1000)*SDur[i]
  dose_IQ2E[j] <- C_airQ2[j]*Inh_E*(1/1000)*SDur[i]
  dose_IQ3C[j] <- C_airQ3[j]*Inh_C[i]*(1/1000)*SDur[i]
  dose_IQ3A[j] <- C_airQ3[j]*Inh_A[i]*(1/1000)*SDur[i]
  dose_IQ3E[j] <- C_airQ3[j]*Inh_E*(1/1000)*SDur[i]

  # Sample deposition rates for each of the lung regions: R1 = region 1, R2 = region 2, R3 = region 3

  FR1[j] <- trlogis(1,0.06777134,0.04304692,0.01,0.195)
  FR2[j] <- trweibull(1,0.4134387,0.2102361,0.000001,0.41)
  FR3[j] <- rtLnormal(1,-1.05243558,0.43503011,0.159,0.62)
```

```
    # Calculate doses ro each of the regions for flow rates

    D1_C <- dose_IQ1C[j]*FR1[j]; D1_A <- dose_IQ1A[j]*FR1[j]; D1_E <- dose_IQ1E[j]*FR1[j]
    D2_C <- abs((dose_IQ1C[j]-D1_C)*FR2[j]); D2_A <- abs((dose_IQ1A[j]-D1_A)*FR2[j]); D2_E <- abs((dose_IQ1E[j]-D1_E)*FR2[j])
    dose_dQ1C[j] <- abs((dose_IQ1C[j]-(D1_C+D2_C))*FR3[j])      # delivered dose to region 3 of the lungs flow rate 1
    dose_dQ1A[j] <- abs((dose_IQ1A[j]-(D1_A+D2_A))*FR3[j])

    dose_dQ1E[j] <- abs((dose_IQ1E[j]-(D1_E+D2_E))*FR3[j])
    dose_dQ2C[j] <- abs((dose_IQ2C[j]-(D1_C+D2_C))*FR3[j])      # delivered dose to region 3 of the lungs flow rate 2

    dose_dQ2A[j] <- abs((dose_IQ2A[j]-(D1_A+D2_A))*FR3[j])
    dose_dQ2E[j] <- abs((dose_IQ2E[j]-(D1_E+D2_E))*FR3[j])
    dose_dQ3C[j] <- abs((dose_IQ3C[j]-(D1_C+D2_C))*FR3[j])      # delivered dose to region 3 of the lungs flow rate 3

    dose_dQ3A[j] <- abs((dose_IQ3A[j]-(D1_A+D2_A))*FR3[j])
    dose_dQ3E[j] <- abs((dose_IQ3E[j]-(D1_E+D2_E))*FR3[j])
    k[j] <- TriRand(0.00326, 0.00599, 0.131)

    # Caculate infection risks for each flow rate PopDRisk is the combined population infection risk
    if(Qnum == 1){DRiskC[j] <- 1-exp(-0.00599*dose_dQ1C[j]); DRiskA[j] <- 1-exp(-0.00599*dose_dQ1A[j])
    DRiskE[j] <- 1-exp(-0.00599*dose_dQ1E[j]);

    PopDRisk[j] <- sum(DRiskC[j],DRiskA[j],DRiskE[j])-prod(DRiskC[j],DRiskA[j])-prod(DRiskC[j],
                       DRiskE[j])-prod(DRiskA[j],DRiskE[j])+prod(DRiskC[j],DRiskA[j],DRiskE[j])}
    else if(Qnum == 2) {DRiskC[j] <- 1-exp(-0.00599*dose_dQ2C[j]); DRiskA[j] <- 1-exp(-0.00599*dose_dQ2A[j])
    DRiskE[j] <- 1-exp(-0.00599*dose_dQ2E[j]);
    PopDRisk[j] <- sum(DRiskC[j],DRiskA[j],DRiskE[j])-prod(DRiskC[j],DRiskA[j])-prod(DRiskC[j],
                       DRiskE[j])-prod(DRiskA[j],DRiskE[j])+prod(DRiskC[j],DRiskA[j],DRiskE[j])}
    else if(Qnum == 3) {DRiskC[j] <- 1-exp(-0.00599*dose_dQ3C[j]); DRiskA[j] <- 1-exp(-0.00599*dose_dQ3A[j])
    DRiskE[j] <- 1-exp(-0.00599*dose_dQ3E[j]);
    PopDRisk[j] <- sum(DRiskC[j],DRiskA[j],DRiskE[j])-prod(DRiskC[j],DRiskA[j])-prod(DRiskC[j],
                       DRiskE[j])-prod(DRiskA[j],DRiskE[j])+prod(DRiskC[j],DRiskA[j],DRiskE[j])}
    else{stop("Choose Flow Rate in Qnum Just After 2D_Simulation_Loops Section Header")}

    # FOR TESTING PURPOSES Model annual risk using older method

    ARiskC[j] <- 1-(1-DRiskC[j])^183; ARiskA[j] <- 1-(1-DRiskA[j])^183; ARiskE[j] <- 1-(1-DRiskE[j])^183
                                  # n = 183 --> showering every-other-day

    ARisks[j] <- sum(ARiskC[j],ARiskA[j],ARiskE[j])-prod(ARiskC[j],ARiskA[j])-prod(ARiskC[j],ARiskE[j])-
      prod(ARiskA[j],ARiskE[j])+prod(ARiskC[j],ARiskA[j],ARiskE[j])

}

# g is the matrix that is the sets of columns of all results from the 2D Sim
```

```
    AnnualRisksC <- dailyRisks2AnnualRisks(DRiskC); AnnualRisksA <- dailyRisks2AnnualRisks(DRiskA) AnnualRisksE
    <- dailyRisks2AnnualRisks(DRiskE)


    AnnualsStiched <- sum(AnnualRisksC,AnnualRisksA,AnnualRisksE)-prod(AnnualRisksC,AnnualRisksA)-prod(AnnualRisksC,
                          AnnualRisksE)-prod(AnnualRisksA,AnnualRisksE)+prod(AnnualRisksC,AnnualRisksA,AnnualRisksE)
    AnnualPopRisk <- dailyRisks2AnnualRisks(PopDRisk)

    gC[1:inner,i] <- DRiskC; gA[1:inner,i] <- DRiskA; gE[1:inner,i] <- DRiskE
    hC[1:inner,i] <- AnnualRisksC; hA[1:inner,i] <- AnnualRisksA; hE[1:inner,i] <- AnnualRisksE
    Pop[1:inner,i] <- PopDRisk
    PopA[1:inner,i] <- AnnualPopRisk
    AnnStiched[1:inner,i] <- AnnualsStiched
    ARisksStiched[1:inner,i] <- ARisks
  }

# Output_Results ------------------------------------------------------------

  DailyRisks <- cbind(DRiskC,DRiskA,DRiskE); colnames(DailyRisks) <- c("Children","Adults","Elderly")
  AnnualRisks <- cbind(ARiskC,ARiskA,ARiskE); colnames(AnnualRisks) <- c("Children","Adults","Elderly")

  write.csv(DailyRisks, file = sprintf("Q%s_Daily_Risks.csv",Qnum)) write.csv(AnnualRisks,
  file = sprintf("Q%s_Annual_Risks.csv",Qnum))

  Children_Risks_Raw <- cbind(DRiskC,ARiskC)
  colnames(Children_Risks_Raw) <- c("Daily Risk to Children","Annual Risk to Children")
  Adult_Risks_Raw <- cbind(DRiskA,ARiskA)
  colnames(Adult_Risks_Raw) <- c("Daily Risk to Children","Annual Risk to Children")
  Elderly_Risks_Raw <- cbind(DRiskE,ARiskE)
  colnames(Elderly_Risks_Raw) <- c("Daily Risk to Children","Annual Risk to Children")

  Children_DRisks_summary <- cbind(mean(DRiskC),median(DRiskC),sd(DRiskC),min(DRiskC),max(DRiskC),

                                   quantile(DRiskC,probs = 0.05), quantile(DRiskC,probs = 0.95))
  Children_ARisks_summary <- cbind(mean(AnnualRisksC),median(AnnualRisksC),

                                   sd(AnnualRisksC),min(AnnualRisksC),max(AnnualRisksC),
                                   quantile(AnnualRisksC,probs = 0.05), quantile(AnnualRisksC,probs = 0.95))
  Children_summary <- rbind(Children_DRisks_summary, Children_ARisks_summary)
  colnames(Children_summary) <- c("mean","median","Standard Deviation","Min","Max","Lower 95th","Upper 95th")
  row.names(Children_summary) <- c("Daily Risk Children", "Annual Risk Children")
  write.csv(Children_summary, file=sprintf("Q%s_Children_Summary_Stats.csv",Qnum))

  Adult_DRisks_summary <- cbind(mean(DRiskA),median(DRiskA),sd(DRiskA),min(DRiskA),max(DRiskA),

                                quantile(DRiskA,probs = 0.05), quantile(DRiskA,probs = 0.95)) Adult_ARisks_summary
  <- cbind(mean(AnnualRisksA),median(AnnualRisksA),
```

```
sd(AnnualRisksA),min(AnnualRisksA),max(AnnualRisksA),
quantile(AnnualRisksA,probs = 0.05), quantile(AnnualRisksA,probs = 0.95))
```

```
Adult_summary <- rbind(Adult_DRisks_summary, Adult_ARisks_summary)
colnames(Adult_summary) <- c("mean","median","Standard Deviation","Min","Max","Lower 95th","Upper 95th")
row.names(Adult_summary) <- c("Daily Risk Adult", "Annual Risk Adult")
write.csv(Adult_summary, file=sprintf("Q%s_Adult_Summary_Stats.csv",Qnum))

Elderly_DRisks_summary <- cbind(mean(DRiskE),median(DRiskE),sd(DRiskE),min(DRiskE),max(DRiskE),
                                quantile(DRiskE,probs = 0.05), quantile(DRiskE,probs = 0.95))
Elderly_ARisks_summary <- cbind(mean(AnnualRisksE),median(AnnualRisksE),
                                sd(AnnualRisksE),min(AnnualRisksE),max(AnnualRisksE),
                                quantile(AnnualRisksE,probs = 0.05), quantile(AnnualRisksE,probs = 0.95))
Elderly_summary <- rbind(Elderly_DRisks_summary, Elderly_ARisks_summary)
colnames(Elderly_summary) <- c("mean","median","Standard Deviation","Min","Max","Lower 95th","Upper 95th")
row.names(Elderly_summary) <- c("Daily Risk Elderly", "Annual Risk Elderly")
write.csv(Elderly_summary, file=sprintf("Q%s_Elderly_Summary_Stats.csv",Qnum))



Pop_summary <- cbind(mean(Pop),median(Pop),sd(Pop),min(Pop),max(Pop),
                     quantile(Pop,probs = 0.05), quantile(Pop,probs = 0.95))
PopA_summary <- cbind(mean(PopA),median(PopA),sd(PopA),min(PopA),max(PopA),
                      quantile(PopA,probs = 0.05), quantile(PopA,probs = 0.95))
Pop_Summarys <- rbind(Pop_summary,PopA_summary)
colnames(Pop_Summarys) <- c("mean","median","Standard Deviation","Min","Max","Lower 95th","Upper 95th")
row.names(Pop_Summarys) <- c("Daily Risks","Annual Risks")
write.csv(Pop_Summarys,file=sprintf("Q%s_Population_Summary_Stats.csv",Qnum))


# Sort_Results_for_S_Curve ------------------------------------------------

  # Gs is the g matrix sorted by column - Grs is the Gs matrix sorted by row

  GC <- matrix(nrow=inner, ncol=outer); GA <- matrix(nrow=inner, ncol=outer)
  GE <- matrix(nrow=inner, ncol=outer)
  GCr <- matrix(nrow=inner, ncol=outer); GAr <- matrix(nrow=inner, ncol=outer)
  GEr <- matrix(nrow=inner, ncol=outer)

  HC <- matrix(nrow=inner, ncol=outer); HA <- matrix(nrow=inner, ncol=outer)
  HE <- matrix(nrow=inner, ncol=outer)
  HCr <- matrix(nrow=inner, ncol=outer); HAr <- matrix(nrow=inner, ncol=outer)
  HEr <- matrix(nrow=inner, ncol=outer)

  # Sort by columns
```

```
  for(ii in 1:outer)

  {
    GC[1:inner,ii]  <-as.matrix(sort(gC[,ii],decreasing  =  FALSE))
    GA[1:inner,ii]  <-as.matrix(sort(gA[,ii],decreasing  =  FALSE))
    GE[1:inner,ii]  <-as.matrix(sort(gE[,ii],decreasing  =  FALSE))


    HC[1:inner,ii]  <-as.matrix(sort(hC[,ii],decreasing  =  FALSE))
    HA[1:inner,ii]  <-as.matrix(sort(hA[,ii],decreasing  =  FALSE))
    HE[1:inner,ii]  <-as.matrix(sort(hE[,ii],decreasing  =  FALSE))

  }

  # Sort by rows

  for(ii in 1:inner)

  {
    GCr[ii,1:outer] <- as.matrix(sort(GC[ii,], decreasing = FALSE))
    GAr[ii,1:outer] <- as.matrix(sort(GA[ii,], decreasing = FALSE))
    GEr[ii,1:outer] <- as.matrix(sort(GE[ii,], decreasing =FALSE))


    HCr[ii,1:outer] <- as.matrix(sort(HC[ii,], decreasing = FALSE))
    HAr[ii,1:outer]  <-  as.matrix(sort(HA[ii,],  decreasing  =  FALSE))
    HEr[ii,1:outer] <- as.matrix(sort(HE[ii,], decreasing = FALSE))

  }

# Vectors for Plotting ---------------------------------------------------------

  # Break out the Groups of sorted values based on the group 's placement in each 95th and median

  snip <- (1:ncol(gC))

  Confs  <-  round(quantile(snip,c(0.05,0.5,0.95)),0)

  GC_05 <- as.matrix(GCr[,Confs[1]]); GC_50 <- as.matrix(GCr[,Confs[2]])
  GC_95 <-  as.matrix(GCr[,Confs[3]]);

  GA_05 <- as.matrix(GAr[,Confs[1]]); GA_50 <- as.matrix(GAr[,Confs[2]])
  GA_95 <-  as.matrix(GAr[,Confs[3]]);

  GE_05 <- as.matrix(GEr[,Confs[1]]); GE_50 <- as.matrix(GEr[,Confs[2]]) GE_95
  <-  as.matrix(GEr[,Confs[3]]);


  HC_05 <- as.matrix(HCr[,Confs[1]]); HC_50 <- as.matrix(HCr[,Confs[2]])
  HC_95 <-  as.matrix(HCr[,Confs[3]]);
```

```
HA_05 <- as.matrix(HAr[,Confs[1]]); HA_50 <- as.matrix(HAr[,Confs[2]])
HA_95 <-  as.matrix(HAr[,Confs[3]]);

HE_05 <- as.matrix(HEr[,Confs[1]]); HE_50 <- as.matrix(HEr[,Confs[2]])



HE_95 <- as.matrix(HEr[,Confs[3]]);

# Build probability vector to be used to plot S-curves

PGC <- matrix(nrow=length(GCr[,1]), ncol=1); PGA <- matrix(nrow=length(GAr[,1]), ncol=1);
PGE <- matrix(nrow=length(GEr[,1]),  ncol=1)


PHC <- matrix(nrow=length(HCr[,1]),  ncol=1);  PHA  <-  matrix(nrow=length(HAr[,1]),  ncol=1);
PHE <- matrix(nrow=length(HEr[,1]), ncol=1)

for(jj in 1:length(GCr[,1]))

{
   PGC[jj]  <-  round((jj/length(GCr[,1])),  digits=3)
   PGA[jj]  <-  round((jj/length(GAr[,1])),  digits=3)
   PGE[jj]  <-  round((jj/length(GEr[,1])),  digits=3)


   PHC[jj]  <-  round((jj/length(HCr[,1])),  digits=3)
   PHA[jj]  <-  round((jj/length(HAr[,1])),  digits=3)
   PHE[jj]  <-  round((jj/length(HEr[,1])),  digits=3)

}

Inf_Sens_Vars_names_all <- c("Inhalation Rate","AF Q1 Cold ", "AF Q2 Hot", "Shower Time", "Partitioning Coeff", "DF1", "DF2", "DF3", "Legionella Conc", "k")
 Inf_Sens_Vars_names_Elderly <- c("AF Q1 Cold ", "AF Q2 Hot", "Shower Time", "Partitioning Coeff", "DF1", "DF2", "DF3", "Legionella Conc", "k")

 # Ill_Sens_Vars_names_Child <- c("AF Q1 Cold ", "AF Q2 Hot", "Shower Time", "Partitioning Coeff", "DF1", "DF2", "DF3", "Legionella Conc", "k", "Morbidity Ratio")
 # Ill_Sens_Vars_names_Child <- c("AF Q1 Cold ", "AF Q2 Hot", "Shower Time", "Partitioning Coeff", "DF1", "DF2", "DF3", "Legionella Conc", "k", "Morbidity Ratio")
 # Ill_Sens_Vars_names_Child <- c("AF Q1 Cold ", "AF Q2 Hot", "Shower Time", "Partitioning Coeff", "DF1", "DF2", "DF3", "Legionella Conc", "k", "Morbidity Ratio")

 max.length <- length(CLeg); fill.length <- max.length-length(Inh_C)

 Inf_Sens_Vars_Child <- data.frame(c(Inh_C,rep(NA,fill.length)),c(ASFrac_Q1_cold,rep(NA,fill.length)), c(ASFrac_Q1_hot,rep(NA,fill.length))
                          , c(SDur,rep(NA,fill.length)), c(LPart,rep(NA,fill.length)), FR1, FR2, FR3, CLeg, k)
 colnames(Inf_Sens_Vars_Child) <- Inf_Sens_Vars_names_all

 Inf_Sens_Vars_Adult <- data.frame(c(Inh_A,rep(NA,fill.length)),c(ASFrac_Q1_cold,rep(NA,fill.length)), c(ASFrac_Q1_hot,rep(NA,fill.length))
                          , c(SDur,rep(NA,fill.length)), c(LPart,rep(NA,fill.length)), FR1, FR2, FR3, CLeg, k)
 colnames(Inf_Sens_Vars_Adult) <- Inf_Sens_Vars_names_all
```

```
Inf_Sens_Vars_Elderly <- data.frame(c(ASFrac_Q1_cold,rep(NA,fill.length)), c(ASFrac_Q1_hot,rep(NA,fill.length))
                          , c(SDur,rep(NA,fill.length)), c(LPart,rep(NA,fill.length)), FR1, FR2, FR3, CLeg, k)
colnames(Inf_Sens_Vars_Elderly) <- Inf_Sens_Vars_names_Elderly

Inf_Sens_Child <- matrix(nrow=1, ncol=ncol(Inf_Sens_Vars_Adult))
Inf_Sens_Adult <- matrix(nrow=1, ncol=ncol(Inf_Sens_Vars_Adult))
Inf_Sens_Elderly <- matrix(nrow=1, ncol=ncol(Inf_Sens_Vars_Elderly))
for(i in 1:ncol(Inf_Sens_Vars_Adult))
{
  Child <- cor.test(Inf_Sens_Vars_Child[,i],DRiskC, method = "spearman")
  Inf_Sens_Child[,i] <- Child$estimate

  Adult <- cor.test(Inf_Sens_Vars_Adult[,i],DRiskA, method = "spearman")
  Inf_Sens_Adult[,i] <- Adult$estimate

}

for(i in 1:ncol(Inf_Sens_Vars_Elderly))
{
  Elderly <- cor.test(Inf_Sens_Vars_Elderly[,i],DRiskE, method = "spearman")
  Inf_Sens_Elderly[,i] <- Elderly$estimate
}

colnames(Inf_Sens_Child) <- Inf_Sens_Vars_names_all
colnames(Inf_Sens_Adult) <- Inf_Sens_Vars_names_all
colnames(Inf_Sens_Elderly) <- Inf_Sens_Vars_names_Elderly

Child_Sens <- melt(Inf_Sens_Child)
colnames(Child_Sens) <- c("dummy","Variable","Spearman")
Child_Sens_Plot <- ggplot(Child_Sens,aes(x=Variable,y=Spearman, fill=Variable)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme(legend.position = "none") +
  labs(y = expression(bold(~Spearman~rho)), x = expression(bold(~Model~Variable)),
       title = expression(~Child~Infection~Risk))
ggsave("Child_Sens_Plot.png",Child_Sens_Plot)

Adult_Sens <- melt(Inf_Sens_Adult)
colnames(Adult_Sens) <- c("dummy","Variable","Spearman")
Adult_Sens_Plot <- ggplot(Adult_Sens,aes(x=Variable,y=Spearman, fill=Variable)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme(legend.position = "none") +
  labs(y = expression(bold(~Spearman~rho)), x = expression(bold(~Model~Variable)),
```

```
      title = expression(~Adult~Infection~Risk))
  ggsave("AdultSens_Plot.png",Adult_Sens_Plot)

  Elderly_Sens <- melt(Inf_Sens_Elderly)
  colnames(Elderly_Sens) <- c("dummy","Variable","Spearman")
  Elderly_Sens_Plot <- ggplot(Elderly_Sens,aes(x=Variable,y=Spearman, fill=Variable)) +
   geom_bar(stat = "identity") +
   coord_flip() +
   theme(legend.position = "none") +
   labs(y = expression(bold(~Spearman~rho)), x = expression(bold(~Model~Variable)),
      title = expression(~Elderly~Infection~Risk))
  ggsave("Elderly_Sens_Plot.png",Elderly_Sens_Plot)

 # Plot the S-curve and save as a jpeg to current directory

# Daily Risk Plotting  ------------------------------------------------------------

 jpeg(sprintf("Q%s_DailyRisk_S_Curves.jpeg",Qnum), height=700, width=1000)
 par(cex=1,mfrow=c(1,3), cex.main=1.75, cex.lab=1.7, cex.axis=1.5, mai=c(0.65,0.65,0.5,0.5))
 plot(GC_05,PGC,type="l", col="red", xlim=c(min(GC[GC!=min(GC)]), max(GC)), ylim=c(min(PGC),max(PGC)),
      xlab = "Risk (unitless)", ylab="Percentile of Risk", main="Children Daily Risk", log="x", lwd=2);
 lines(GC_50,PGC,col="blue", lwd=2)
 lines(GC_95,PGC,col="green", lwd=2)
 legend(min(GC[GC!=min(GC)]), max(PGC)-0.01, legend=c(expression("Lower"~"95"^"th"), "Median",expression("Upper"~"95"^"th")),
        lty=c(1,1,1), lwd=c(2,2,2), col=c("red","blue","green"))


 plot(GA_05,PGA,type="l", col="red", xlim=c(min(GA[GA!=min(GA)]), max(GA)), ylim=c(min(PGA),max(PGA)),
      xlab = "Risk (unitless)", ylab="Percentile of Risk", main="Adult Daily Risk", log="x", lwd=2);
 lines(GA_50,PGA,col="blue", lwd=2)
 lines(GA_95,PGA,col="green", lwd=2)
 legend(min(GA[GA!=min(GA)]), max(PGA)-0.01, legend=c(expression("Lower"~"95"^"th"), "Median",expression("Upper"~"95"^"th")),
        lty=c(1,1,1), lwd=c(2,2,2), col=c("red","blue","green"))

 plot(GE_05,PGC,type="l", col="red", xlim=c(min(GE[GE!=min(GE)]), max(GE)), ylim=c(min(PGE),max(PGE)),
```

```
        xlab = "Risk (unitless)", ylab="Percentile of Risk", main="Elderly Daily Risk", log="x", lwd=2);
  lines(GE_50,PGE,col="blue", lwd=2)
  lines(GE_95,PGE,col="green", lwd=2)
  legend(min(GE[GE!=min(GE)]), max(PGE)-0.01, legend=c(expression("Lower"~"95"^"th"), "Median",expression("Upper"~"95"^"th")),
         lty=c(1,1,1), lwd=c(2,2,2), col=c("red","blue","green"))

  dev.off()

  jpeg(sprintf("Q%s_DailyRisk_Histograms_Boxplots.jpeg",Qnum), height=700, width=1000)
  par(cex=1,mfrow=c(1,4), cex.main=1.75, cex.lab=1.7, cex.axis=1.5, mai=c(0.65,0.65,0.5,0.5))
  hist(DRiskC, main="Children Daily Risks", xlab=expression(bold(Risk~unitless)))
  hist(DRiskA, main="Adult Daily Risks", xlab=expression(bold(Risk~unitless)))
  hist(DRiskE, main="Elderly Daily Risks", xlab=expression(bold(Risk~unitless)))
  boxplot(DailyRisks, ylab=expression(bold(Risk~(unitless))), main="Boxplot of Daily Risks")
  dev.off()



#   Annual_Risk_Plotting   ----------------------------------------------------

  jpeg(sprintf("Q%s_AnnualRisk_S_Curves.jpeg",Qnum),  height=700,  width=1000)
  par(cex=1,mfrow=c(1,3), cex.main=1.75, cex.lab=1.7, cex.axis=1.5, mai=c(0.65,0.65,0.5,0.5))
  plot(HC_05,PHC,type="l", col="red", xlim=c(min(HC[HC!=min(HC)]), max(HC)), ylim=c(min(PHC),max(PHC)),
       xlab = "Risk (unitless)", ylab="Percentile of Risk",
       main="Children Annual \nInfection Risk", log="x", lwd=2);
  lines(HC_50,PHC,col="blue", lwd=2)
  lines(HC_95,PHC,col="green", lwd=2)
  legend(min(HC[HC!=min(HC)]), max(PHC)-0.01, legend=c(expression("Lower"~"95"^"th"), "Median",expression("Upper"~"95"^"th")),
         lty=c(1,1,1), lwd=c(2,2,2), col=c("red","blue","green"))


  plot(HA_05,PHA,type="l", col="red", xlim=c(min(HA[HA!=min(HA)]), max(HA)), ylim=c(min(PHA),max(PHA)),
       xlab = "Risk (unitless)", ylab="Percentile of Risk",
       main="Adult Annual \nInfection Risk", log="x", lwd=2);
  lines(HA_50,PHA,col="blue", lwd=2)
  lines(HA_95,PHA,col="green", lwd=2)
  legend(min(HA[HA!=min(HA)]), max(PHA)-0.01, legend=c(expression("Lower"~"95"^"th"), "Median",expression("Upper"~"95"^"th")),
         lty=c(1,1,1), lwd=c(2,2,2), col=c("red","blue","green"))


  plot(HE_05,PHE,type="l", col="red", xlim=c(min(HE[HE!=min(HE)]), max(HE)), ylim=c(min(PHE),max(PHE)),
       xlab = "Risk (unitless)", ylab="Percentile of Risk",
       main="Elderly Annual \nInfection Risk", log="x", lwd=2);
  lines(HE_50,PHE,col="blue", lwd=2)
  lines(HE_95,PHE,col="green", lwd=2)
  legend(min(HE[HE!=min(HE)]), max(PHE)-0.01, legend=c(expression("Lower"~"95"^"th"), "Median",expression("Upper"~"95"^"th")),
```

```
        lty=c(1,1,1), lwd=c(2,2,2), col=c("red","blue","green"))
dev.off()

jpeg(sprintf("Q%s_AnnualRisk_Histograms_boxPlot.jpeg",Qnum), height=700, width=1000)
par(cex=1,mfrow=c(1,4), cex.main=1.75, cex.lab=1.7, cex.axis=1.5, mai=c(0.65,0.65,0.5,0.5))
hist(ARiskC, main="Children Annual \nInfection Risks", xlab=expression(bold(Risk~unitless)))
hist(ARiskA, main="Adult Annual \nInfection Risks", xlab=expression(bold(Risk~unitless)))
hist(ARiskE, main="Elderly Annual \nInfection Risks", xlab=expression(bold(Risk~unitless)))
boxplot(AnnualRisks, ylab=expression(bold(Risk~(unitless))), main="Boxplot of Annual \nInfection Risks")

dev.off()

# Open up space on local directory for improved memory usage

rm("AnnualRisks","ModerateDALY","SeverDALY","PostAcuteDALY",
   "Children_Risks_Raw","Adult_Risks_Raw","Elderly_Risks_Raw")

# Demographics Modelling ----------------------------------------------------------------------------------------------

dir.create("Demographics"); setwd("Demographics")

# Set up the demographic probability of illness from Beers et al(2011)

# Data represent incidence per individual groups crude incidence in the age groups are adjusted for
age,          # sex and race are not adjusted for age

# Raw Scores

pop <- (mean(308.7,281.4)*1000000) # USA Population in 2011

ChildRawPill <- mean(.02,.03); AdultRawPill <- mean(.13,.36,.81,1.44); ElderlyRawPill <- mean(1.94,2.29,2.66)
MaleRawPill <- (.97); FemaleRawPill <- (.53); UnknownRawPill <- (145/(pop*0.0065));
AmIndianRawPill <- (0.21); AsianRawPill <- (.14); BlackRawPill <- (.87); WhiteRawPill <- (0.59);
OtherRawPill <- ((411/pop*0.028)*100000); IR_Pop <- 0.75

# Raw data multiplied by translational term needs to be multiplied by percentage of population in that group (age, sex, race)

MR_Child <- 0.5*(ChildRawPill/IR_Pop); MR_Adult <- 0.5*(AdultRawPill/IR_Pop); MR_Elderly <- 0.5*(ElderlyRawPill/IR_Pop)
MR_Asian <- 0.5*(AsianRawPill/IR_Pop); MR_AmIndian <- 0.5*(AmIndianRawPill/IR_Pop); MR_Black <- 0.5*(BlackRawPill/IR_Pop)
MR_White <- 0.5*(WhiteRawPill/IR_Pop); MR_Female <- 0.5*(FemaleRawPill/IR_Pop)
MR_Male <- 0.5*(MaleRawPill/IR_Pop); MR_Other <- 0.5*(OtherRawPill/IR_Pop)

# Set up matrices for the demographic modelling - Daily Risks and DALYs

CRiskPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); ARiskPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); ERiskPill
<- matrix(nrow = nrow(gC),ncol=ncol(gC))
```

```
AsianRiskPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); BlackRiskPill <- matrix(nrow = nrow(gC),ncol=ncol(gC))


WhiteRiskPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); OtherRiskPill <- matrix(nrow = nrow(gC),ncol=ncol(gC))
FemaleRiskPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); MaleRiskPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)) AIPill
<- matrix(nrow = nrow(gC), ncol = ncol(gC))


ChildMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); AdultMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
ElderlyMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))

AsianMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); BlackMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
WhiteMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); OtherMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
FemaleMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); MaleMDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
AIMDALY <- matrix(nrow = nrow(gC), ncol = ncol(gC))


ChildSDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); AdultSDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
ElderlySDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))

AsianSDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); BlackSDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
WhiteSDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); OtherSDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
FemaleSDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); MaleSDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
AISDALY <- matrix(nrow = nrow(gC), ncol = ncol(gC))


ChildPDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); AdultPDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
ElderlyPDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))

AsianPDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); BlackPDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
WhitePDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); OtherPDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
FemalePDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC)); MalePDALY <- matrix(nrow = nrow(gC),ncol=ncol(gC))
AIPDALY <- matrix(nrow = nrow(gC), ncol = ncol(gC))

# Set up matrices for the demographic modelling - Annual Risks

CRiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); ARiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC));
ERiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC))

AsianRiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); BlackRiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC))
WhiteRiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); OtherRiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC))
FemaleRiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC)); MaleRiskAPill <- matrix(nrow = nrow(gC),ncol=ncol(gC))
AIAPill <- matrix(nrow = nrow(gC), ncol = ncol(gC))

# Demographic Modeling Loop ----------------------------------------------------------------------------------

for(ij in 1:ncol(gC))

{

  CRiskPill[,ij] <- gC[,ij]*MR_Child; ARiskPill[,ij] <- gA[,ij]*MR_Adult
```

```
ERiskPill[,ij] <- gE[,ij]*MR_Elderly
CRiskAPill[,ij] <- hC[,ij]*MR_Child; ARiskAPill[,ij] <- hA[,ij]*MR_Adult
```

```
ERiskAPill[,ij] <- hE[,ij]*MR_Elderly
ChildMDALY[,ij] <- CRiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
ChildSDALY[,ij] <- CRiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB);
ChildPDALY[,ij] <- CRiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB)
                                      ,TriRand(DW_am_LB,DW_am_mu,DW_am_UB))

AdultMDALY[,ij] <- ARiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
AdultSDALY[,ij] <- ARiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB);
AdultPDALY[,ij] <- ARiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),
                                      TriRand(DW_am_LB,DW_am_mu,DW_am_UB))

ElderlyMDALY[,ij] <- ERiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
ElderlySDALY[,ij] <- ERiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB);
ElderlyPDALY[,ij] <- ERiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),
                                        TriRand(DW_am_LB,DW_am_mu,DW_am_UB))

AsianRiskPill[,ij] <- Pop[,ij]*MR_Asian; BlackRiskPill[,ij] <- Pop[,ij]*MR_Black
WhiteRiskPill[,ij] <- Pop[,ij]*MR_White;
OtherRiskPill[,ij] <- Pop[,ij]*MR_Other; AIPill[,ij] <- Pop[,ij]*MR_AmIndian
AsianRiskAPill[,ij] <- ARisksStiched[,ij]*MR_Asian; BlackRiskAPill[,ij] <- ARisksStiched[,ij]*MR_Black;
WhiteRiskAPill[,ij] <- ARisksStiched[,ij]*MR_White; OtherRiskAPill[,ij] <- ARisksStiched[,ij]*MR_Other
AIAPill[,ij] <- ARisksStiched[,ij]*MR_AmIndian

AsianMDALY[,ij] <- AsianRiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
BlackMDALY[,ij] <- BlackRiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
WhiteMDALY[,ij] <- WhiteRiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
OtherMDALY[,ij] <- OtherRiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
AIMDALY[,ij] <- AIPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)

AsianSDALY[,ij] <- AsianRiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB)
BlackSDALY[,ij] <- BlackRiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB)
WhiteSDALY[,ij] <- WhiteRiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB)
OtherSDALY[,ij] <- OtherRiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB)
AISDALY[,ij] <- AIPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB)

AsianPDALY[,ij] <- AsianRiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),
                                          TriRand(DW_am_LB,DW_am_mu,DW_am_UB))
BlackPDALY[,ij] <- BlackRiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),
                                          TriRand(DW_am_LB,DW_am_mu,DW_am_UB))
WhitePDALY[,ij] <- WhiteRiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),
                                          TriRand(DW_am_LB,DW_am_mu,DW_am_UB))
OtherPDALY[,ij] <- OtherRiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),
```

```
                                                TriRand(DW_am_LB,DW_am_mu,DW_am_UB))
    AIPDALY[,ij] <- AIPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),TriRand(DW_am_LB,DW_am_mu,DW_am_UB))

    MaleRiskPill[,ij] <- Pop[,ij]*MR_Male
    FemaleRiskPill[,ij] <- Pop[,ij]*MR_Female
    MaleRiskAPill[,ij] <- ARisksStiched[,ij]*MR_Male
    FemaleRiskAPill[,ij] <- ARisksStiched[,ij]*MR_Female
    MaleMDALY[,ij] <- MaleRiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
    FemaleMDALY[,ij] <- FemaleRiskPill[,ij]*TriRand(DW_am_LB,DW_am_mu,DW_am_UB)
    MaleSDALY[,ij] <- MaleRiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB)
    FemaleSDALY[,ij] <- FemaleRiskPill[,ij]*TriRand(DW_as_LB,DW_as_mu,DW_as_UB)
    MalePDALY[,ij] <- MaleRiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),
                                            TriRand(DW_am_LB,DW_am_mu,DW_am_UB))
    FemalePDALY[,ij] <- FemaleRiskPill[,ij]*sum(TriRand(DW_pc_LB,DW_pc_mu,DW_pc_UB),
                                                TriRand(DW_am_LB,DW_am_mu,DW_am_UB))

}
write.csv(CRiskPill, file="Children_Pill_Raw.csv"); write.csv(ARiskPill, file="Adult_Pill_Raw.csv")
write.csv(ERiskPill, file="Elderly_Pill_Raw.csv"); write.csv(AIPill, file="American_Indian_Pill_Raw.csv")
write.csv(AsianRiskPill, file="Asian_Pill_Raw.csv"); write.csv(BlackRiskPill, file="Black_Pill_Raw.csv");
write.csv(WhiteRiskPill, file="White_Pill_Raw.csv"); write.csv(OtherRiskPill, file="Other_Pill_Raw.csv");
write.csv(MaleRiskPill, file="Male_Pill_Raw.csv"); write.csv(FemaleRiskPill, file="Female_Pill_Raw.csv")

write.csv(ChildMDALY, file="Children_MDALY_Raw.csv"); write.csv(AdultMDALY, file="Adult_MDALY_Raw.csv")
write.csv(ElderlyMDALY, file="Elderly_MDALY_Raw.csv"); write.csv(AIMDALY, file="American_Indian_MDALY_Raw.csv")
write.csv(AsianMDALY, file="Asian_MDALY_Raw.csv"); write.csv(BlackMDALY, file="Black_MDALY_Raw.csv");
write.csv(WhiteMDALY, file="White_MDALY_Raw.csv"); write.csv(OtherMDALY, file="Other_MDALY_Raw.csv");
write.csv(MaleMDALY, file="Male_MDALY_Raw.csv"); write.csv(FemaleMDALY, file="Female_MDALY_Raw.csv")

write.csv(ChildSDALY, file="Children_SDALY_Raw.csv"); write.csv(AdultSDALY, file="Adult_SDALY_Raw.csv")
write.csv(ElderlySDALY, file="Elderly_SDALY_Raw.csv"); write.csv(AISDALY, file="American_Indian_SDALY_Raw.csv")
write.csv(AsianSDALY, file="Asian_SDALY_Raw.csv"); write.csv(BlackSDALY, file="Black_SDALY_Raw.csv");
write.csv(WhiteSDALY, file="White_SDALY_Raw.csv"); write.csv(OtherSDALY, file="Other_SDALY_Raw.csv");
write.csv(MaleSDALY, file="Male_SDALY_Raw.csv"); write.csv(FemaleSDALY, file="Female_SDALY_Raw.csv")

write.csv(ChildPDALY, file="Children_PDALY_Raw.csv"); write.csv(AdultPDALY, file="Adult_PDALY_Raw.csv")
write.csv(ElderlyPDALY, file="Elderly_PDALY_Raw.csv"); write.csv(AIPDALY, file="American_Indian_PDALY_Raw.csv")
write.csv(AsianPDALY, file="Asian_PDALY_Raw.csv"); write.csv(BlackPDALY, file="Black_PDALY_Raw.csv");
write.csv(WhitePDALY, file="White_PDALY_Raw.csv"); write.csv(OtherPDALY, file="Other_PDALY_Raw.csv");
write.csv(MalePDALY, file="Male_PDALY_Raw.csv"); write.csv(FemalePDALY, file="Female_PDALY_Raw.csv")

source("U:/Publications/Legionella_2016_Case_Study/R_Codes/Non_Confounding_RScript/10000_Iter/Pop_testing/Multiple_t_test.r")
```

```
# Matrices for Column and Row Sorting ------------------------------------------------------------------------------

  CRiskPillC <- matrix(nrow=inner,ncol=outer); ARiskPillC <- matrix(nrow=inner,ncol=outer); ERiskPillC <- matrix(nrow=inner,ncol=outer);
  CRiskAPillC <- matrix(nrow=inner,ncol=outer); ARiskAPillC <- matrix(nrow=inner,ncol=outer); ERiskAPillC <- matrix(nrow=inner,ncol=outer);

  CMDALYC <- matrix(nrow=inner,ncol=outer); AMDALYC <- matrix(nrow=inner,ncol=outer); EMDALYC <- matrix(nrow=inner,ncol=outer);
  CSDALYC <- matrix(nrow=inner,ncol=outer); ASDALYC <- matrix(nrow=inner,ncol=outer); ESDALYC <- matrix(nrow=inner,ncol=outer);
  CPDALYC <- matrix(nrow=inner,ncol=outer); APDALYC <- matrix(nrow=inner,ncol=outer); EPDALYC <- matrix(nrow=inner,ncol=outer);

  AsianRiskPillC <- matrix(nrow=inner,ncol=outer)
  BlackRiskPillC <- matrix(nrow=inner,ncol=outer)
  WhiteRiskPillC <- matrix(nrow=inner,ncol=outer)
  OtherRiskPillC <- matrix(nrow=inner,ncol=outer)
  AsianRiskAPillC <- matrix(nrow=inner,ncol=outer)
  BlackRiskAPillC <- matrix(nrow=inner,ncol=outer)
  WhiteRiskAPillC <- matrix(nrow=inner,ncol=outer)
  OtherRiskAPillC <- matrix(nrow=inner,ncol=outer)
  AsianMDALYC <- matrix(nrow=inner,ncol=outer)
  BlackMDALYC <- matrix(nrow=inner,ncol=outer)
  WhiteMDALYC <- matrix(nrow=inner,ncol=outer)
  OtherMDALYC <- matrix(nrow=inner,ncol=outer)
  AsianSDALYC <- matrix(nrow=inner,ncol=outer)
  BlackSDALYC <- matrix(nrow=inner,ncol=outer)
  WhiteSDALYC <- matrix(nrow=inner,ncol=outer)
  OtherSDALYC <- matrix(nrow=inner,ncol=outer)
  AsianPDALYC <- matrix(nrow=inner,ncol=outer)
  BlackPDALYC <- matrix(nrow=inner,ncol=outer)
  WhitePDALYC <- matrix(nrow=inner,ncol=outer)
  OtherPDALYC <- matrix(nrow=inner,ncol=outer)
  AIPillC <- matrix(nrow=inner,ncol=outer)
  AIAPillC <- matrix(nrow=inner,ncol=outer)
  AIMDALYC <- matrix(nrow=inner,ncol=outer)
  AISDALYC <- matrix(nrow=inner,ncol=outer)
  AIPDALYC <- matrix(nrow=inner,ncol=outer)

  MPillC <- matrix(nrow=inner,ncol=outer); FPillC <- matrix(nrow=inner,ncol=outer);
  MAPillC <- matrix(nrow=inner,ncol=outer); FAPillC <- matrix(nrow=inner,ncol=outer);
  MMDALYC <- matrix(nrow=inner,ncol=outer); FMDALYC <- matrix(nrow=inner,ncol=outer);
  MSDALYC <- matrix(nrow=inner,ncol=outer); FSDALYC <- matrix(nrow=inner,ncol=outer);
  MPDALYC <- matrix(nrow=inner,ncol=outer); FPDALYC <- matrix(nrow=inner,ncol=outer);
```

```
  CRiskPillR <- matrix(nrow=inner,ncol=outer); ARiskPillR <- matrix(nrow=inner,ncol=outer); ERiskPillR <- matrix(nrow=inner,ncol=outer);
  CRiskAPillR <- matrix(nrow=inner,ncol=outer); ARiskAPillR <- matrix(nrow=inner,ncol=outer); ERiskAPillR <- matrix(nrow=inner,ncol=outer);

  CMDALYR <- matrix(nrow=inner,ncol=outer); AMDALYR <- matrix(nrow=inner,ncol=outer); EMDALYR <- matrix(nrow=inner,ncol=outer);
  CSDALYR <- matrix(nrow=inner,ncol=outer); ASDALYR <- matrix(nrow=inner,ncol=outer); ESDALYR <- matrix(nrow=inner,ncol=outer);
  CPDALYR <- matrix(nrow=inner,ncol=outer); APDALYR <- matrix(nrow=inner,ncol=outer); EPDALYR <- matrix(nrow=inner,ncol=outer);

  AsianRiskPillR <- matrix(nrow=inner,ncol=outer)
  BlackRiskPillR <- matrix(nrow=inner,ncol=outer)
  WhiteRiskPillR <- matrix(nrow=inner,ncol=outer)
  OtherRiskPillR <- matrix(nrow=inner,ncol=outer)
  AsianRiskAPillR <- matrix(nrow=inner,ncol=outer)
  BlackRiskAPillR <- matrix(nrow=inner,ncol=outer)
  WhiteRiskAPillR <- matrix(nrow=inner,ncol=outer)
  OtherRiskAPillR <- matrix(nrow=inner,ncol=outer)
  AsianMDALYR <- matrix(nrow=inner,ncol=outer)
  BlackMDALYR <- matrix(nrow=inner,ncol=outer)
  WhiteMDALYR <- matrix(nrow=inner,ncol=outer)
  OtherMDALYR <- matrix(nrow=inner,ncol=outer)
  AsianSDALYR <- matrix(nrow=inner,ncol=outer)
  BlackSDALYR <- matrix(nrow=inner,ncol=outer)
  WhiteSDALYR <- matrix(nrow=inner,ncol=outer)
  OtherSDALYR <- matrix(nrow=inner,ncol=outer)
  AsianPDALYR <- matrix(nrow=inner,ncol=outer)
  BlackPDALYR <- matrix(nrow=inner,ncol=outer)
  WhitePDALYR <- matrix(nrow=inner,ncol=outer)
  OtherPDALYR <- matrix(nrow=inner,ncol=outer)
  AIPillR <- matrix(nrow=inner,ncol=outer)
  AIAPillR <- matrix(nrow=inner,ncol=outer)
  AIMDALYR <- matrix(nrow=inner,ncol=outer)
  AISDALYR <- matrix(nrow=inner,ncol=outer)
  AIPDALYR <- matrix(nrow=inner,ncol=outer)

  MPillR <- matrix(nrow=inner,ncol=outer); FPillR <- matrix(nrow=inner,ncol=outer);
  MAPillR <- matrix(nrow=inner,ncol=outer); FAPillR <- matrix(nrow=inner,ncol=outer);
  MMDALYR <- matrix(nrow=inner,ncol=outer); FMDALYR <- matrix(nrow=inner,ncol=outer);
  MSDALYR <- matrix(nrow=inner,ncol=outer); FSDALYR <- matrix(nrow=inner,ncol=outer);
  MPDALYR <- matrix(nrow=inner,ncol=outer); FPDALYR <- matrix(nrow=inner,ncol=outer);
# Sort by Columns ------------------------------------------------------------------------------------------

  for(il in 1:outer)

  {
```

```
CRiskPillC[1:inner,il] <- as.matrix(sort(CRiskPill[,il],decreasing = FALSE))
ARiskPillC[1:inner,il] <- as.matrix(sort(ARiskPill[,il],decreasing = FALSE))
ERiskPillC[1:inner,il] <- as.matrix(sort(ERiskPill[,il],decreasing = FALSE))
MPillC[1:inner,il] <- as.matrix(sort(MaleRiskPill[,il],decreasing = FALSE))
FPillC[1:inner,il] <- as.matrix(sort(FemaleRiskPill[,il],decreasing = FALSE))

CRiskAPillC[1:inner,il] <- as.matrix(sort(CRiskAPill[,il],decreasing = FALSE))
ARiskAPillC[1:inner,il] <- as.matrix(sort(ARiskAPill[,il],decreasing = FALSE))
ERiskAPillC[1:inner,il] <- as.matrix(sort(ERiskAPill[,il],decreasing = FALSE))
MAPillC[1:inner,il] <- as.matrix(sort(MaleRiskAPill[,il],decreasing = FALSE))
FAPillC[1:inner,il] <- as.matrix(sort(FemaleRiskAPill[,il],decreasing = FALSE))

CMDALYC[1:inner,il] <- as.matrix(sort(ChildMDALY[,il],decreasing = FALSE))
AMDALYC[1:inner,il] <- as.matrix(sort(AdultMDALY[,il],decreasing = FALSE))
EMDALYC[1:inner,il] <- as.matrix(sort(ElderlyMDALY[,il],decreasing = FALSE))
MMDALYC[1:inner,il] <- as.matrix(sort(MaleMDALY[,il],decreasing = FALSE))
FMDALYC[1:inner,il] <- as.matrix(sort(FemaleMDALY[,il],decreasing = FALSE))

CSDALYC[1:inner,il] <- as.matrix(sort(ChildSDALY[,il],decreasing = FALSE))
ASDALYC[1:inner,il] <- as.matrix(sort(AdultSDALY[,il],decreasing = FALSE))
ESDALYC[1:inner,il] <- as.matrix(sort(ElderlySDALY[,il],decreasing = FALSE))
MSDALYC[1:inner,il] <- as.matrix(sort(MaleSDALY[,il],decreasing = FALSE))
FSDALYC[1:inner,il] <- as.matrix(sort(FemaleSDALY[,il],decreasing = FALSE))

CPDALYC[1:inner,il] <- as.matrix(sort(ChildPDALY[,il],decreasing = FALSE))
APDALYC[1:inner,il] <- as.matrix(sort(AdultPDALY[,il],decreasing = FALSE))
EPDALYC[1:inner,il] <- as.matrix(sort(ElderlyPDALY[,il],decreasing = FALSE))
MPDALYC[1:inner,il] <- as.matrix(sort(MalePDALY[,il],decreasing = FALSE))
FPDALYC[1:inner,il] <- as.matrix(sort(FemalePDALY[,il],decreasing = FALSE))

AsianRiskPillC[1:inner,il] <- as.matrix(sort(AsianRiskPill[,il],decreasing = FALSE))
BlackRiskPillC[1:inner,il] <- as.matrix(sort(BlackRiskPill[,il],decreasing = FALSE))
WhiteRiskPillC[1:inner,il] <- as.matrix(sort(WhiteRiskPill[,il],decreasing = FALSE))
OtherRiskPillC[1:inner,il] <- as.matrix(sort(OtherRiskPill[,il],decreasing = FALSE))
AIPillC[1:inner,il] <- as.matrix((sort(AIPill[,il],decreasing = FALSE)))

AsianRiskAPillC[1:inner,il] <- as.matrix(sort(AsianRiskAPill[,il],decreasing = FALSE))
BlackRiskAPillC[1:inner,il] <- as.matrix(sort(BlackRiskAPill[,il],decreasing = FALSE))
WhiteRiskAPillC[1:inner,il] <- as.matrix(sort(WhiteRiskAPill[,il],decreasing = FALSE))
OtherRiskAPillC[1:inner,il] <- as.matrix(sort(OtherRiskAPill[,il],decreasing = FALSE))
AIAPillC[1:inner,il] <- as.matrix((sort(AIAPill[,il],decreasing = FALSE)))
```

```
    AsianMDALYC[1:inner,il]  <-  as.matrix(sort(AsianMDALY[,il],decreasing  =  FALSE))
    BlackMDALYC[1:inner,il] <- as.matrix(sort(BlackMDALY[,il],decreasing = FALSE))
    WhiteMDALYC[1:inner,il]  <-  as.matrix(sort(WhiteMDALY[,il],decreasing  =  FALSE))
    OtherMDALYC[1:inner,il]  <-  as.matrix(sort(OtherMDALY[,il],decreasing  =  FALSE))
    AIMDALYC[1:inner,il] <- as.matrix((sort(AIMDALY[,il],decreasing = FALSE)))

    AsianSDALYC[1:inner,il] <- as.matrix(sort(AsianSDALY[,il],decreasing = FALSE))
    BlackSDALYC[1:inner,il]  <-  as.matrix(sort(BlackSDALY[,il],decreasing  =  FALSE))
    WhiteSDALYC[1:inner,il] <- as.matrix(sort(WhiteSDALY[,il],decreasing = FALSE))
    OtherSDALYC[1:inner,il] <- as.matrix(sort(OtherSDALY[,il],decreasing = FALSE))
    AISDALYC[1:inner,il] <- as.matrix((sort(AISDALY[,il],decreasing = FALSE)))

    AsianPDALYC[1:inner,il] <- as.matrix(sort(AsianPDALY[,il],decreasing = FALSE))
    BlackPDALYC[1:inner,il]  <-  as.matrix(sort(BlackPDALY[,il],decreasing  =  FALSE))
    WhitePDALYC[1:inner,il] <- as.matrix(sort(WhitePDALY[,il],decreasing = FALSE))
    OtherPDALYC[1:inner,il] <- as.matrix(sort(OtherPDALY[,il],decreasing = FALSE))
    AIPDALYC[1:inner,il] <- as.matrix((sort(AIPDALY[,il],decreasing = FALSE)))
  }

# Sort by Rows ------------------------------------------------------------------------------------------------------

  for(im in 1:inner)

  {

    CRiskPillR[im,1:outer] <- as.matrix(sort(CRiskPillC[im,],decreasing = FALSE))
    ARiskPillR[im,1:outer] <- as.matrix(sort(ARiskPillC[im,],decreasing = FALSE))
    ERiskPillR[im,1:outer] <- as.matrix(sort(ERiskPillC[im,],decreasing = FALSE))
    MPillR[im,1:outer] <- as.matrix(sort(MPillC[im,],decreasing = FALSE))
    FPillR[im,1:outer] <- as.matrix(sort(FPillC[im,],decreasing = FALSE))

    CRiskAPillR[im,1:outer] <- as.matrix(sort(CRiskAPillC[im,],decreasing = FALSE))
    ARiskAPillR[im,1:outer] <- as.matrix(sort(ARiskAPillC[im,],decreasing = FALSE))
    ERiskAPillR[im,1:outer] <- as.matrix(sort(ERiskAPillC[im,],decreasing = FALSE))
    MAPillR[im,1:outer] <- as.matrix(sort(MAPillC[im,],decreasing = FALSE))
    FAPillR[im,1:outer] <- as.matrix(sort(FAPillC[im,],decreasing = FALSE))

    CMDALYR[im,1:outer] <- as.matrix(sort(CMDALYC[im,],decreasing = FALSE))
    AMDALYR[im,1:outer]  <-  as.matrix(sort(AMDALYC[im,],decreasing  =  FALSE))
    EMDALYR[im,1:outer] <- as.matrix(sort(EMDALYC[im,],decreasing = FALSE))
    MMDALYR[im,1:outer]  <-  as.matrix(sort(MMDALYC[im,],decreasing  =  FALSE))
    MMDALYR[im,1:outer]  <-  as.matrix(sort(MMDALYC[im,],decreasing  =  FALSE))
```

```
CSDALYR[im,1:outer] <- as.matrix(sort(CSDALYC[im,],decreasing = FALSE))
ASDALYR[im,1:outer] <- as.matrix(sort(ASDALYC[im,],decreasing = FALSE))
ESDALYR[im,1:outer] <- as.matrix(sort(ESDALYC[im,],decreasing = FALSE))
MSDALYR[im,1:outer] <- as.matrix(sort(MSDALYC[im,],decreasing = FALSE))
MSDALYR[im,1:outer] <- as.matrix(sort(MSDALYC[im,],decreasing = FALSE))

CPDALYR[im,1:outer] <- as.matrix(sort(CPDALYC[im,],decreasing = FALSE))
APDALYR[im,1:outer] <- as.matrix(sort(APDALYC[im,],decreasing = FALSE))
EPDALYR[im,1:outer] <- as.matrix(sort(EPDALYC[im,],decreasing = FALSE))
MPDALYR[im,1:outer] <- as.matrix(sort(MPDALYC[im,],decreasing = FALSE))
MPDALYR[im,1:outer] <- as.matrix(sort(MPDALYC[im,],decreasing = FALSE));

AsianRiskPillR[im,1:outer] <- as.matrix(sort(AsianRiskPillC[im,],decreasing = FALSE))
BlackRiskPillR[im,1:outer] <- as.matrix(sort(BlackRiskPillC[im,],decreasing = FALSE))
WhiteRiskPillR[im,1:outer] <- as.matrix(sort(WhiteRiskPillC[im,],decreasing = FALSE))
OtherRiskPillR[im,1:outer] <- as.matrix(sort(OtherRiskPillC[im,],decreasing = FALSE))
AIPillR[im,1:outer] <- as.matrix((sort(AIPillC[im,],decreasing = FALSE)))

AsianRiskAPillR[im,1:outer] <- as.matrix(sort(AsianRiskAPillC[im,],decreasing = FALSE))
BlackRiskAPillR[im,1:outer] <- as.matrix(sort(BlackRiskAPillC[im,],decreasing = FALSE))
WhiteRiskAPillR[im,1:outer] <- as.matrix(sort(WhiteRiskAPillC[im,],decreasing = FALSE))
OtherRiskAPillR[im,1:outer] <- as.matrix(sort(OtherRiskAPillC[im,],decreasing = FALSE))
AIAPillR[im,1:outer] <- as.matrix((sort(AIAPillC[im,],decreasing = FALSE)))

AsianMDALYR[im,1:outer] <- as.matrix(sort(AsianMDALYC[im,],decreasing = FALSE))
BlackMDALYR[im,1:outer] <- as.matrix(sort(BlackMDALYC[im,],decreasing = FALSE))
WhiteMDALYR[im,1:outer] <- as.matrix(sort(WhiteMDALYC[im,],decreasing = FALSE))
OtherMDALYR[im,1:outer] <- as.matrix(sort(OtherMDALYC[im,],decreasing = FALSE))
AIMDALYR[im,1:outer] <- as.matrix((sort(AIMDALYC[im,],decreasing = FALSE)))

AsianSDALYR[im,1:outer] <- as.matrix(sort(AsianSDALYC[im,],decreasing = FALSE))
BlackSDALYR[im,1:outer] <- as.matrix(sort(BlackSDALYC[im,],decreasing = FALSE))
WhiteSDALYR[im,1:outer] <- as.matrix(sort(WhiteSDALYC[im,],decreasing = FALSE))
OtherSDALYR[im,1:outer] <- as.matrix(sort(OtherSDALYC[im,],decreasing = FALSE))
AISDALYR[im,1:outer] <- as.matrix((sort(AISDALYC[im,],decreasing = FALSE)))

AsianPDALYR[im,1:outer] <- as.matrix(sort(AsianPDALYC[im,],decreasing = FALSE))
BlackPDALYR[im,1:outer] <- as.matrix(sort(BlackPDALYC[im,],decreasing = FALSE))
WhitePDALYR[im,1:outer] <- as.matrix(sort(WhitePDALYC[im,],decreasing = FALSE))
OtherPDALYR[im,1:outer] <- as.matrix(sort(OtherPDALYC[im,],decreasing = FALSE))
AIPDALYR[im,1:outer] <- as.matrix((sort(AIPDALYC[im,],decreasing = FALSE)))
```

```
}

# Develop Percentiles ------------------------------------------------------------------------------------------------

snip <- (1:ncol(gC))

Confs  <-  round(quantile(snip,c(0.05,0.5,0.95)),0)

# Age Percentiles

ChildPill_05 <- as.matrix(CRiskPillR[,Confs[1]]); ChildPill_50 <- as.matrix(CRiskPillR[,Confs[2]])
ChildPill_95 <- as.matrix(CRiskPillR[,Confs[3]]);
AdultPill_05 <- as.matrix(ARiskPillR[,Confs[1]]); AdultPill_50 <- as.matrix(ARiskPillR[,Confs[2]])
AdultPill_95 <- as.matrix(ARiskPillR[,Confs[3]]);
ElderlyPill_05 <- as.matrix(ERiskPillR[,Confs[1]]); ElderlyPill_50 <- as.matrix(ERiskPillR[,Confs[2]])
ElderlyPill_95 <- as.matrix(ERiskPillR[,Confs[3]]);

ChildAPill_05 <- as.matrix(CRiskAPillR[,Confs[1]]); ChildAPill_50 <- as.matrix(CRiskAPillR[,Confs[2]])
ChildAPill_95 <- as.matrix(CRiskAPillR[,Confs[3]]);
AdultAPill_05 <- as.matrix(ARiskAPillR[,Confs[1]]); AdultAPill_50 <- as.matrix(ARiskAPillR[,Confs[2]])
AdultAPill_95 <- as.matrix(ARiskAPillR[,Confs[3]]);
ElderlyAPill_05 <- as.matrix(ERiskAPillR[,Confs[1]]); ElderlyAPill_50 <- as.matrix(ERiskAPillR[,Confs[2]])
ElderlyAPill_95 <- as.matrix(ERiskAPillR[,Confs[3]]);

ChildMDALY_05 <- as.matrix(CMDALYR[,Confs[1]]); ChildMDALY_50 <- as.matrix(CMDALYR[,Confs[2]])
ChildMDALY_95 <- as.matrix(CMDALYR[,Confs[3]]);
AdultMDALY_05 <- as.matrix(AMDALYR[,Confs[1]]); AdultMDALY_50 <- as.matrix(AMDALYR[,Confs[2]])
AdultMDALY_95 <- as.matrix(AMDALYR[,Confs[3]]);
ElderlyMDALY_05 <- as.matrix(EMDALYR[,Confs[1]]); ElderlyMDALY_50 <- as.matrix(EMDALYR[,Confs[2]])
ElderlyMDALY_95 <- as.matrix(EMDALYR[,Confs[3]]);

ChildSDALY_05 <- as.matrix(CMDALYR[,Confs[1]]); ChildSDALY_50 <- as.matrix(CMDALYR[,Confs[2]])
ChildSDALY_95 <- as.matrix(CMDALYR[,Confs[3]]);
AdultSDALY_05 <- as.matrix(AMDALYR[,Confs[1]]); AdultSDALY_50 <- as.matrix(AMDALYR[,Confs[2]])
AdultSDALY_95 <- as.matrix(AMDALYR[,Confs[3]]);
ElderlySDALY_05 <- as.matrix(EMDALYR[,Confs[1]]); ElderlySDALY_50 <- as.matrix(EMDALYR[,Confs[2]])
ElderlySDALY_95 <- as.matrix(EMDALYR[,Confs[3]]);

ChildPDALY_05 <- as.matrix(CPDALYR[,Confs[1]]); ChildPDALY_50 <- as.matrix(CPDALYR[,Confs[2]])
ChildPDALY_95 <- as.matrix(CPDALYR[,Confs[3]]);
AdultPDALY_05 <- as.matrix(APDALYR[,Confs[1]]); AdultPDALY_50 <- as.matrix(APDALYR[,Confs[2]])
AdultPDALY_95 <- as.matrix(APDALYR[,Confs[3]]);
ElderlyPDALY_05 <- as.matrix(EPDALYR[,Confs[1]]); ElderlyPDALY_50 <- as.matrix(EPDALYR[,Confs[2]])
ElderlyPDALY_95 <- as.matrix(EPDALYR[,Confs[3]]);
```

```
# Race Percentiles

  AIPill_05 <- as.matrix(AIPillR[,Confs[1]]); AIPill_50 <- as.matrix(AIPillR[,Confs[2]])
  AIPill_95 <- as.matrix(AIPillR[,Confs[3]]);

  AsianPill_05 <- as.matrix(AsianRiskPillR[,Confs[1]]) AsianPill_50 <- as.matrix(AsianRiskPillR[,Confs[2]])
  AsianPill_95 <- as.matrix(AsianRiskPillR[,Confs[3]]);
  BlackPill_05 <- as.matrix(BlackRiskPillR[,Confs[1]]); BlackPill_50 <- as.matrix(BlackRiskPillR[,Confs[2]])
  BlackPill_95 <- as.matrix(BlackRiskPillR[,Confs[3]]);
  WhitePill_05 <- as.matrix(WhiteRiskPillR[,Confs[1]]); WhitePill_50 <- as.matrix(WhiteRiskPillR[,Confs[2]])
  WhitePill_95 <- as.matrix(WhiteRiskPillR[,Confs[3]]);
  OtherPill_05 <- as.matrix(OtherRiskPillR[,Confs[1]]); OtherPill_50 <- as.matrix(OtherRiskPillR[,Confs[2]])
  OtherPill_95 <- as.matrix(OtherRiskPillR[,Confs[3]]);

  AIAPill_05 <- as.matrix(AIAPillR[,Confs[1]]); AIAPill_50 <- as.matrix(AIPillR[,Confs[2]])
  AIAPill_95 <- as.matrix(AIAPillR[,Confs[3]]);
  AsianAPill_05 <- as.matrix(AsianRiskAPillR[,Confs[1]]); AsianAPill_50 <-as.matrix(AsianRiskAPillR[,Confs[2]])
  AsianAPill_95 <- as.matrix(AsianRiskAPillR[,Confs[3]]);
  BlackAPill_05 <- as.matrix(BlackRiskAPillR[,Confs[1]]); BlackAPill_50 <- as.matrix(BlackRiskAPillR[,Confs[2]])
  BlackAPill_95 <- as.matrix(BlackRiskAPillR[,Confs[3]]);
  WhiteAPill_05 <- as.matrix(WhiteRiskAPillR[,Confs[1]]); WhiteAPill_50 <- as.matrix(WhiteRiskAPillR[,Confs[2]])
  WhiteAPill_95 <- as.matrix(WhiteRiskAPillR[,Confs[3]]);
  OtherAPill_05 <- as.matrix(OtherRiskAPillR[,Confs[1]]); OtherAPill_50 <-as.matrix(OtherRiskAPillR[,Confs[2]])
  OtherAPill_95 <- as.matrix(OtherRiskAPillR[,Confs[3]]);

  AIMDALY_05 <- as.matrix(AIMDALYR[,Confs[1]]); AIMDALY_50 <- as.matrix(AIMDALYR[,Confs[2]])
  AIMDALY_95 <- as.matrix(AIMDALYR[,Confs[3]]);
  AsianMDALY_05 <- as.matrix(AsianMDALYR[,Confs[1]]); AsianMDALY_50 <- as.matrix(AsianMDALYR[,Confs[2]])
  AsianMDALY_95 <- as.matrix(AsianMDALYR[,Confs[3]]);
  BlackMDALY_05 <- as.matrix(BlackMDALYR[,Confs[1]]); BlackMDALY_50 <- as.matrix(BlackMDALYR[,Confs[2]])
  BlackMDALY_95 <- as.matrix(BlackMDALYR[,Confs[3]]);
  WhiteMDALY_05 <- as.matrix(WhiteMDALYR[,Confs[1]]); WhiteMDALY_50 <- as.matrix(WhiteMDALYR[,Confs[2]])
  WhiteMDALY_95 <- as.matrix(WhiteMDALYR[,Confs[3]]);
  OtherMDALY_05 <- as.matrix(OtherMDALYR[,Confs[1]]); OtherMDALY_50 <- as.matrix(OtherMDALYR[,Confs[2]])
  OtherMDALY_95 <- as.matrix(OtherMDALYR[,Confs[3]]);

  AISDALY_05 <- as.matrix(AISDALYR[,Confs[1]]); AISDALY_50 <- as.matrix(AISDALYR[,Confs[2]])
  AISDALY_95 <- as.matrix(AISDALYR[,Confs[3]]);
  AsianSDALY_05 <- as.matrix(AsianSDALYR[,Confs[1]]); AsianSDALY_50 <- as.matrix(AsianSDALYR[,Confs[2]])
  AsianSDALY_95 <- as.matrix(AsianSDALYR[,Confs[3]]);
  BlackSDALY_05 <- as.matrix(BlackSDALYR[,Confs[1]]); BlackSDALY_50 <- as.matrix(BlackSDALYR[,Confs[2]])
```

```
BlackSDALY_95  <- as.matrix(BlackSDALYR[,Confs[3]]);
WhiteSDALY_05 <- as.matrix(WhiteSDALYR[,Confs[1]]); WhiteSDALY_50 <- as.matrix(WhiteSDALYR[,Confs[2]])
WhiteSDALY_95 <- as.matrix(WhiteSDALYR[,Confs[3]]);
OtherSDALY_05 <- as.matrix(OtherSDALYR[,Confs[1]]); OtherSDALY_50 <- as.matrix(OtherSDALYR[,Confs[2]])
OtherSDALY_95 <- as.matrix(OtherSDALYR[,Confs[3]]);

AIPDALY_05 <- as.matrix(AIPDALYR[,Confs[1]]); AIPDALY_50 <- as.matrix(AIPDALYR[,Confs[2]])
AIPDALY_95 <- as.matrix(AIPDALYR[,Confs[3]]);
AsianPDALY_05 <- as.matrix(AsianPDALYR[,Confs[1]]); AsianPDALY_50 <- as.matrix(AsianPDALYR[,Confs[2]])
AsianPDALY_95 <- as.matrix(AsianPDALYR[,Confs[3]]);
BlackPDALY_05 <- as.matrix(BlackPDALYR[,Confs[1]]); BlackPDALY_50 <- as.matrix(BlackPDALYR[,Confs[2]])
BlackPDALY_95 <- as.matrix(BlackPDALYR[,Confs[3]]);
WhitePDALY_05 <- as.matrix(WhitePDALYR[,Confs[1]]); WhitePDALY_50 <- as.matrix(WhitePDALYR[,Confs[2]])
WhitePDALY_95 <- as.matrix(WhitePDALYR[,Confs[3]]);
OtherPDALY_05 <- as.matrix(OtherPDALYR[,Confs[1]]); OtherPDALY_50 <- as.matrix(OtherPDALYR[,Confs[2]])
OtherPDALY_95 <- as.matrix(OtherPDALYR[,Confs[3]]);

# Sex Percentiles

MPill_05 <- as.matrix(MPillR[,Confs[1]]); MPill_50 <- as.matrix(MPillR[,Confs[2]])
MPill_95 <- as.matrix(MPillR[,Confs[3]]);

MAPill_05 <- as.matrix(MAPillR[,Confs[1]]); MAPill_50 <- as.matrix(MAPillR[,Confs[2]])
MAPill_95 <- as.matrix(MAPillR[,Confs[3]]);

MMDALY_05 <- as.matrix(MMDALYR[,Confs[1]]); MMDALY_50 <- as.matrix(MMDALYR[,Confs[2]])
MMDALY_95 <- as.matrix(MMDALYR[,Confs[3]]);

MSDALY_05 <- as.matrix(MSDALYR[,Confs[1]]); MSDALY_50 <- as.matrix(MSDALYR[,Confs[2]])
MSDALY_95 <- as.matrix(MSDALYR[,Confs[3]]);

MPDALY_05 <- as.matrix(MPDALYR[,Confs[1]]); MPDALY_50 <- as.matrix(MPDALYR[,Confs[2]])
MPDALY_95 <- as.matrix(MPDALYR[,Confs[3]]);


FPill_05 <- as.matrix(FPillR[,Confs[1]]); FPill_50 <- as.matrix(FPillR[,Confs[2]])
FPill_95 <- as.matrix(FPillR[,Confs[3]]);

FAPill_05 <- as.matrix(FAPillR[,Confs[1]]); FAPill_50 <- as.matrix(FAPillR[,Confs[2]])
FAPill_95 <- as.matrix(FAPillR[,Confs[3]]);

FMDALY_05 <- as.matrix(FMDALYR[,Confs[1]]); FMDALY_50 <- as.matrix(FMDALYR[,Confs[2]])
FMDALY_95 <- as.matrix(FMDALYR[,Confs[3]]);

FSDALY_05 <- as.matrix(FSDALYR[,Confs[1]]); FSDALY_50 <- as.matrix(FSDALYR[,Confs[2]])
FSDALY_95 <- as.matrix(FSDALYR[,Confs[3]]);

FPDALY_05 <- as.matrix(FPDALYR[,Confs[1]]); FPDALY_50 <- as.matrix(FPDALYR[,Confs[2]])
FPDALY_95 <- as.matrix(FPDALYR[,Confs[3]]);
```

```
# Probabilities for Demographic Poltting
```

```
CP <- matrix(nrow=length(CRiskPillR[,1]), ncol=1); AP <- matrix(nrow=length(CRiskPillR[,1]), ncol=1) EP
<- matrix(nrow=length(CRiskPillR[,1]), ncol=1);

for(ib in 1:length(CRiskPillR[,1]))

{
  CP[ib] <- round((ib/length(CRiskPillR[,1])), digits=3) AP[ib]
  <- round((ib/length(ARiskAPillR[,1])),digits = 3) EP[ib] <-
  round((ib/length(ERiskAPillR[,1])),digits = 3)

}

# Dataframe Stacking for Boxplots ----------------------------------------------------------------------------

RiskStack <- cbind(stack(as.data.frame(CRiskPill)),stack(as.data.frame(ARiskPill)),stack(as.data.frame(ERiskPill)),
                   stack(as.data.frame(AsianRiskPill)),stack(as.data.frame(BlackRiskPill)),stack(as.data.frame(WhiteRiskPill)),
                   stack(as.data.frame(OtherRiskPill)),stack(as.data.frame(MaleRiskPill)),stack(as.data.frame(FemaleRiskPill)),
                   stack(as.data.frame(AIPill)))

Pill_All <- cbind(RiskStack[,1],RiskStack[,3],RiskStack[,5],RiskStack[,19],RiskStack[,7],RiskStack[,9],RiskStack[,11],
                  RiskStack[,13],RiskStack[,15],RiskStack[,17])
rm("RiskStack")
# logPill_All <- log(Pill_All)

colnames(Pill_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","OTher","Male","Female")
# colnames(logPill_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","Other","Male","Female")

Pill_summary <- matrix(nrow=ncol(Pill_All),ncol=7) for(iq
in 1:ncol(Pill_All))

{
  Pill_summary[iq,] <- cbind(mean(Pill_All[,iq]),median(Pill_All[,iq]),sd(Pill_All[,iq]),min(Pill_All[,iq]),max(Pill_All[,iq]),
                             quantile(Pill_All[,iq],probs = 0.05), quantile(Pill_All[,iq],probs = 0.95))
}
colnames(Pill_summary) <- c("Mean","Median","Standard Deviation","Minimum","Maximum","Lower 95th","Upper 95th")
rownames(Pill_summary) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","Other","Male","Female")
write.csv(Pill_summary, file=sprintf("Q%s_Pill_Summary_Stats.csv",Qnum))


ARiskStack <- cbind(stack(as.data.frame(CRiskAPill)),stack(as.data.frame(ARiskAPill)),stack(as.data.frame(ERiskAPill)),
                    stack(as.data.frame(AsianRiskAPill)),stack(as.data.frame(BlackRiskAPill)),stack(as.data.frame(WhiteRiskPill)),
                    stack(as.data.frame(OtherRiskPill)),stack(as.data.frame(MaleRiskAPill)),stack(as.data.frame(FemaleRiskAPill)),
                    stack(as.data.frame(AIAPill)))

ARisk_All <- cbind(ARiskStack[,1],ARiskStack[,3],ARiskStack[,5],ARiskStack[,19],ARiskStack[,7],ARiskStack[,9],ARiskStack[,11],
                   ARiskStack[,13],ARiskStack[,15],ARiskStack[,17])
rm("ARiskStack")
# logARisk_All <- log(ARisk_All)
```

```r
colnames(ARisk_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","Other","Male","Female")
# colnames(logARisk_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","Other","Male","Female")
```

```
APill_summary <-
matrix(nrow=ncol(Pill_All),ncol=7) for(iq
in 1:ncol(Pill_All))

{
  APill_summary[iq,] <-
                                  cbind(mean(ARisk_All[,iq]),median(ARisk_All[,iq]),sd(ARisk_All[,iq]),min(ARisk_All[,iq
                                  ]),max(ARisk_All[,iq]), quantile(ARisk_All[,iq],probs = 0.05),
                                  quantile(ARisk_All[,iq],probs = 0.95))
}
colnames(APill_summary) <- c("Mean","Median","Standard Deviation","Minimum","Maximum","Lower 95th","Upper
95th") rownames(APill_summary) <- c("Child","Adult","Elderly","American
Indian","Asian","Black","White","Other","Male","Female") write.csv(APill_summary,
file=sprintf("Q%s_Annual_Pill_Summary_Stats.csv",Qnum))

MDALY_Stack <- cbind(stack(as.data.frame(ChildMDALY)),stack(as.data.frame(AdultMDALY)),stack(as.data.frame(ElderlyMDALY)),
                     stack(as.data.frame(AsianMDALY)),stack(as.data.frame(BlackMDALY)),stack(as.data.frame(WhiteMDALY)),
                     stack(as.data.frame(OtherMDALY)),stack(as.data.frame(MaleMDALY)),stack(as.data.frame(FemaleMDALY)),
                     stack(as.data.frame(AIMDALY)))

MDALY_All <-
                    cbind(MDALY_Stack[,1],MDALY_Stack[,3],MDALY_Stack[,5],MDALY_Stack[,19],MDALY_Stack[,7],MDALY_Stack[,9],MDALY
                    _Stack[,11], MDALY_Stack[,13],MDALY_Stack[,15],MDALY_Stack[,17])
rm("MDALY_Stack")
# logMDALY_All <- log(MDALY_All)

colnames(MDALY_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","Other","Male","Female")
# colnames(logMDALY_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","Other","Male","Female")

MDALY_summary <-
matrix(nrow=ncol(Pill_All),ncol=7) for(iq
in 1:ncol(Pill_All))

{
  MDALY_summary[iq,] <-
                                  cbind(mean(MDALY_All[,iq]),median(MDALY_All[,iq]),sd(MDALY_All[,iq]),min(MDALY_All[,i
                                  q]),max(MDALY_All[,iq]), quantile(MDALY_All[,iq],probs = 0.05),
                                  quantile(MDALY_All[,iq],probs = 0.95))
}
colnames(MDALY_summary) <- c("Mean","Median","Standard Deviation","Minimum","Maximum","Lower 95th","Upper
95th") rownames(MDALY_summary) <- c("Child","Adult","Elderly","American
Indian","Asian","Black","White","Other","Male","Female") write.csv(MDALY_summary,
file=sprintf("Q%s_MDALY_Summary_Stats.csv",Qnum))
```

```
SDALY_Stack <-
                        cbind(stack(as.data.frame(ChildSDALY)),stack(as.data.frame(AdultSDALY)),stack(as.data
                        .frame(ElderlySDALY)),
                        stack(as.data.frame(AsianSDALY)),stack(as.data.frame(BlackSDALY)),stack(as.data.frame
                        (WhiteSDALY)),
                        stack(as.data.frame(OtherSDALY)),stack(as.data.frame(MaleSDALY)),stack(as.data.frame(
                        FemaleSDALY)), stack(as.data.frame(AISDALY)))

SDALY_All <-
                    cbind(SDALY_Stack[,1],SDALY_Stack[,3],SDALY_Stack[,5],SDALY_Stack[,19],SDALY_Stack[,7],SDALY_Stack[,9],SDALY_
                    Stack[,11], SDALY_Stack[,13],SDALY_Stack[,15],SDALY_Stack[,17])
rm("SDALY_Stack")
# logSDALY_All <- log(SDALY_All)

colnames(SDALY_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","Other","Male","Female")
# colnames(logSDALY_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","OTher","Male","Female")

SDALY_summary <- matrix(nrow=ncol(Pill_All),ncol=7)


for(iq in 1:ncol(Pill_All))

{
  SDALY_summary[iq,] <-
                        cbind(mean(SDALY_All[,iq]),median(SDALY_All[,iq]),sd(SDALY_All[,iq]),min(SDALY_All[,i
                        q]),max(SDALY_All[,iq]), quantile(SDALY_All[,iq],probs = 0.05),
                        quantile(SDALY_All[,iq],probs = 0.95))
}
colnames(SDALY_summary) <- c("Mean","Median","Standard Deviation","Minimum","Maximum","Lower 95th","Upper
95th") rownames(SDALY_summary) <- c("Child","Adult","Elderly","American
Indian","Asian","Black","White","Other","Male","Female") write.csv(SDALY_summary,
file=sprintf("Q%s_SDALY_Summary_Stats.csv",Qnum))



PDALY_Stack <- cbind(stack(as.data.frame(ChildPDALY)),stack(as.data.frame(AdultPDALY)),stack(as.data.frame(ElderlyPDALY)),
                    stack(as.data.frame(AsianPDALY)),stack(as.data.frame(BlackPDALY)),stack(as.data.frame(WhitePDALY)),
                    stack(as.data.frame(OtherPDALY)),stack(as.data.frame(MalePDALY)),stack(as.data.frame(FemalePDALY)),
                    stack(as.data.frame(AIPDALY)))

PDALY_All <- cbind(PDALY_Stack[,1],PDALY_Stack[,3],PDALY_Stack[,5],
                    ALY_Stack[,19],PDALY_Stack[,7],PDALY_Stack[,9],PDALY_Stack[,11],PDALY_Sta
                    ck[,13], PDALY_Stack[,15],PDALY_Stack[,17])
```

```
rm("PDALY_Stack")
# logPDALY_All <- log(PDALY_All)

colnames(PDALY_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","Other","Male","Female")
# colnames(logPDALY_All) <- c("Child","Adult","Elderly","American Indian","Asian","Black","White","OTher","Male","Female")

PDALY_summary <-
matrix(nrow=ncol(Pill_All),ncol=7) for(iq
in 1:ncol(Pill_All))

{
   PDALY_summary[iq,] <- cbind(mean(PDALY_All[,iq]),median(PDALY_All[,iq]),

                               sd(PDALY_All[,iq]),min(PDALY_All[,iq]),max(PDALY_All[,
                               iq]), quantile(PDALY_All[,iq],probs = 0.05),
                               quantile(PDALY_All[,iq],probs = 0.95))
}
colnames(PDALY_summary) <- c("Mean","Median","Standard Deviation","Minimum","Maximum","Lower 95th","Upper
95th") rownames(PDALY_summary) <- c("Child","Adult","Elderly","American
Indian","Asian","Black","White","Other","Male","Female") write.csv(PDALY_summary,
file=sprintf("Q%s_PDALY_Summary_Stats.csv",Qnum))

# Boxplots of Risks and DALYS ----------

jpeg(sprintf("Q%s_All_Risk_DALY_Demographic_Boxplots.jpeg",Qnum), height=700,
width=1000) par(mfrow=c(2,4), cex=1, cex.main=1.9, cex.lab=1.3, cex.axis=1.2,
mai=c(1.6,0.95,0.5,0.5)) boxplot(Pill_All, log="y", ylab=" ",xlab="
",pars=list(las=3),outline=FALSE)
Lines <- list(bquote(bold("Daily Risk of")),bquote(bold("Disease (unitless)")))

mtext(do.call(expression, Lines),side=2,line=3:2)

boxplot(MDALY_All,log="y", ylab=" ",xlab="
",pars=list(las=3),outline=FALSE) Lines <-
list(bquote(bold("Moderate Disease")),bquote(bold("DALY
(unitless)")))



mtext(do.call(expression,    Lines),side=2,line=3:2)

boxplot(SDALY_All,log="y", ylab=" ",xlab="
",pars=list(las=3),outline=FALSE) Lines <-
list(bquote(bold("Severe Disease")),bquote(bold("DALY
(unitless)"))) mtext(do.call(expression, Lines),side=2,line=3:2)
```

```
boxplot(PDALY_All,log="y", ylab="  ",xlab="
",pars=list(las=3),outline=FALSE) Lines <-
list(bquote(bold("Post  Acute")),bquote(bold("DALY
(unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

dev.off()

jpeg(sprintf("Q%s_All_Risk_DALY_Racial_Demographic_Boxplots.jpeg",Qnum), height=700,
width=1000) par(mfrow=c(2,4), cex=1, cex.main=1.9, cex.lab=1.3, cex.axis=1.2,
mai=c(1.6,0.95,0.5,0.5)) boxplot(Pill_All[,4:8], log="y",ylab="  ",xlab="
",pars=list(las=3),outline=FALSE)
Lines <- list(bquote(bold("Daily Risk of")),bquote(bold("Illness (unitless)")))

mtext(do.call(expression,    Lines),side=2,line=3:2)

boxplot(MDALY_All[,4:8],log="y", ylab="  ",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("Moderate
Disease")),bquote(bold("DALY  (unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

boxplot(SDALY_All[,4:8], log="y",ylab="  ",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("Severe
Disease")),bquote(bold("DALY  (unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

boxplot(PDALY_All[,4:8], log="y",ylab="  ",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("Post
Acute")),bquote(bold("DALY  (unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

dev.off()

jpeg(sprintf("Q%s_All_Risk_DALY_Age_Demographic_Boxplots.jpeg",Qnum),
height=700, width=1000) par(mfrow=c(2,3), cex=1, cex.main=1.9, cex.lab=1.3,
cex.axis=1.2, mai=c(1.0,0.95,0.5,0.5)) boxplot(DailyRisks[,1:3],log="y",
ylab="  ",xlab="  ",pars=list(las=3),outline=FALSE)
Lines <- list(bquote(bold("Daily Risk of")),bquote(bold("Infection (unitless)")))

mtext(do.call(expression, Lines),side=2,line=3:2)

boxplot(Pill_All[,1:3], log="y",ylab="  ",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("Daily
Risk of")),bquote(bold("Illness (unitless)")))
mtext(do.call(expression,  Lines),side=2,line=3:2)

boxplot(MDALY_All[,1:3], log="y",ylab="  ",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("Moderate
```

```
Disease")),bquote(bold("DALY (unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

boxplot(SDALY_All[,1:3],log="y", ylab="",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("Severe
Disease")),bquote(bold("DALY (unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

boxplot(PDALY_All[,1:3],log="y", ylab="",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("Post
Acute")),bquote(bold("DALY (unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

dev.off()

jpeg(sprintf("Q%s_All_Risk_DALY_Gender_Demographic_Boxplots.jpeg",Qnum),
height=700, width=1000) par(mfrow=c(2,4), cex=1, cex.main=1.9, cex.lab=1.3,
cex.axis=1.2, mai=c(1.0,0.95,0.5,0.5)) boxplot(Pill_All[,9:10],log="y", ylab="
",xlab="",pars=list(las=3),outline=FALSE)
Lines <- list(bquote(bold("log Daily Risk of")),bquote(bold("Illness (unitless)")))

mtext(do.call(expression, Lines),side=2,line=3:2)

boxplot(MDALY_All[,9:10],log="y", ylab="",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("log
Moderate Disease")),bquote(bold("DALY (unitless)")))
mtext(do.call(expression, Lines),side=2,line=3:2)

boxplot(SDALY_All[,9:10],log="y", ylab="",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("log Severe
Disease")),bquote(bold("DALY (unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

boxplot(PDALY_All[,9:10],log="y", ylab="",xlab="
",pars=list(las=3),outline=FALSE) Lines <- list(bquote(bold("log Post
Acute")),bquote(bold("DALY (unitless)"))) mtext(do.call(expression,
Lines),side=2,line=3:2)

dev.off()

setwd(this.dr)

}
```