

Using moment equations to understand heterogeneity

Definition

We want to write $Y_o(v; D)$ for **Type**, **order**, **variable**, and **Domain**.

We define total as follows: $T_o(v; D) = \int v(a)^o D(a) da$. Since we're mainly interested with susceptibility for now, T_o represents $T_o(.; S)$.

We define $M_i = T_i/T_0$. Then, M_1 is the mean susceptibility and $\kappa = \frac{M_2 M_0}{M_1^2} - 1$ is the squared coefficient of variance (CV).

SI example

We have $\dot{S}(a) = -\Lambda \sigma(a) S(a)$. Integrating gives us $\dot{T}_0 = -\Lambda T_1$. More generally, we have $\dot{T}_i = -\Lambda T_{i+1}$. Given that M_1 is the mean susceptibility, we can also write:

$$\dot{S} = -\Lambda M_1 S$$

Using M defined above, we also have the following equations: $\dot{M}_i = -\Lambda(M_{i+1} - M_i M_1)$.

Given that $M_2 = (1 + \kappa)M^2$ and assuming that κ stays constant, we can integrate the equation above to obtain the following equation:

$M = \hat{M} S^\kappa$, where \hat{M} is the mean susceptibility of the susceptible population at a disease free equilibrium.

Idea 1 - $\kappa_i = \frac{M_i M_{i-2}}{M_{i-1}^2} - 1$

$$\begin{aligned} \kappa_i &= \frac{M_{i-2} M_{i-1} \dot{M}_i + M_{i-1} M_i \dot{M}_{i-2} - 2 M_{i-2} M_i \dot{M}_{i-1}}{M_{i-1}^3} \\ &= -\Lambda \frac{M_{i-2} M_{i-1} (M_{i+1} - M_i M_1) + M_{i-1} M_i (M_{i-1} - M_{i-2} M_1) - 2 M_{i-2} M_i (M_i - M_{i-1} M_1)}{M_{i-1}^3} \\ &= -\Lambda \frac{M_{i-2} M_{i-1} M_{i+1} + M_{i-1}^2 M_i - 2 M_{i-2} M_i^2}{M_{i-1}^3} \\ &= -\Lambda \frac{M_{i-2} (\kappa_{i+1} + 1) M_i^2 + M_{i-1}^2 M_i - 2 M_{i-2} M_i^2}{M_{i-1}^3} \\ &= -\Lambda \frac{(\kappa_{i+1} - 1) M_{i-2} M_i^2 + M_{i-1}^2 M_i}{M_{i-1}^3} \\ &= -\Lambda \frac{(\kappa_{i+1} - 1) (\kappa_i + 1) M_i M_{i-1}^2 + M_{i-1}^2 M_i}{M_{i-1}^3} \\ &= -\Lambda \frac{(\kappa_{i+1} \kappa_i - \kappa_i + \kappa_{i+1}) M_i}{M_{i-1}} \end{aligned}$$

Idea 2 - $\kappa_i = \frac{M_i}{M_{i-1}M_1} - 1$

$$\begin{aligned}
\dot{\kappa}_i &= \frac{M_{i-1}M_1\dot{M}_i - M_iM_1\dot{M}_{i-1} - M_iM_{i-1}\dot{M}_1}{M_{i-1}^2M_1^2} \\
&= -\Lambda \frac{M_{i-1}M_1(M_{i+1} - M_iM_1) - M_iM_1(M_i - M_{i-1}M_1) - M_iM_{i-1}(M_2 - M_1^2)}{M_{i-1}^2M_1^2} \\
&= -\Lambda \frac{M_{i-1}M_1M_{i+1} - M_iM_1M_i - M_iM_{i-1}M_2 + M_iM_{i-1}M_1^2}{M_{i-1}^2M_1^2} \\
&= -\Lambda \frac{(\kappa_{i+1} + 1)M_iM_{i-1}M_1^2 - M_iM_1M_i - (\kappa_2 + 1)M_iM_{i-1}M_1^2 + M_iM_{i-1}M_1^2}{M_{i-1}^2M_1^2} \\
&= -\Lambda \frac{(\kappa_{i+1} + 1)M_iM_{i-1}M_1 - (\kappa_i + 1)M_{i-1}M_iM_1 - \kappa_2M_iM_{i-1}M_1}{M_{i-1}^2M_1} \\
&= -\Lambda \frac{(\kappa_{i+1} + 1)M_i - (\kappa_i + 1)M_i - \kappa_2M_i}{M_{i-1}} \\
&= -\Lambda M_i \frac{\kappa_{i+1} - (\kappa_2 + \kappa_i)}{M_{i-1}} \\
&= -\Lambda M_1(\kappa_i + 1)\{\kappa_{i+1} - (\kappa_2 + \kappa_i)\}
\end{aligned}$$

When $i = 2$, we have $\dot{\kappa} = -\Lambda M(\kappa + 1)(\kappa_3 - 2\kappa)$, where $\kappa = \kappa_2$. For gamma distribution, $\kappa_3 = 2\kappa$. Let's assume that $r = \kappa_3/\kappa$ stays constant. Then, we have $\dot{\kappa} = -\Lambda M(r - 2)(\kappa + 1)(\kappa)$. We can do this:

$$\begin{aligned}
\frac{d\kappa}{dt} &= \frac{d\kappa}{dS} \frac{dS}{dt} \\
-\Lambda M(r - 2)(\kappa + 1)(\kappa) &= -\Lambda M S \frac{d\kappa}{dS} \\
(r - 2)(\kappa + 1)(\kappa) &= S \frac{d\kappa}{dS} \\
\frac{(r - 2)}{S} &= \frac{1}{\kappa(\kappa + 1)} \frac{d\kappa}{dS} \\
\int \frac{(r - 2)}{S} dS &= \int \frac{1}{\kappa(\kappa + 1)} d\kappa \\
(r - 2) \log(S) &= \log(\kappa) - \log(\kappa + 1) + C
\end{aligned}$$

We let the initial values $(S(0), \kappa(0)) = (1, \hat{\kappa})$, then we have $C = \log(\frac{\hat{\kappa}+1}{\hat{\kappa}})$. We can continue with the derivative:

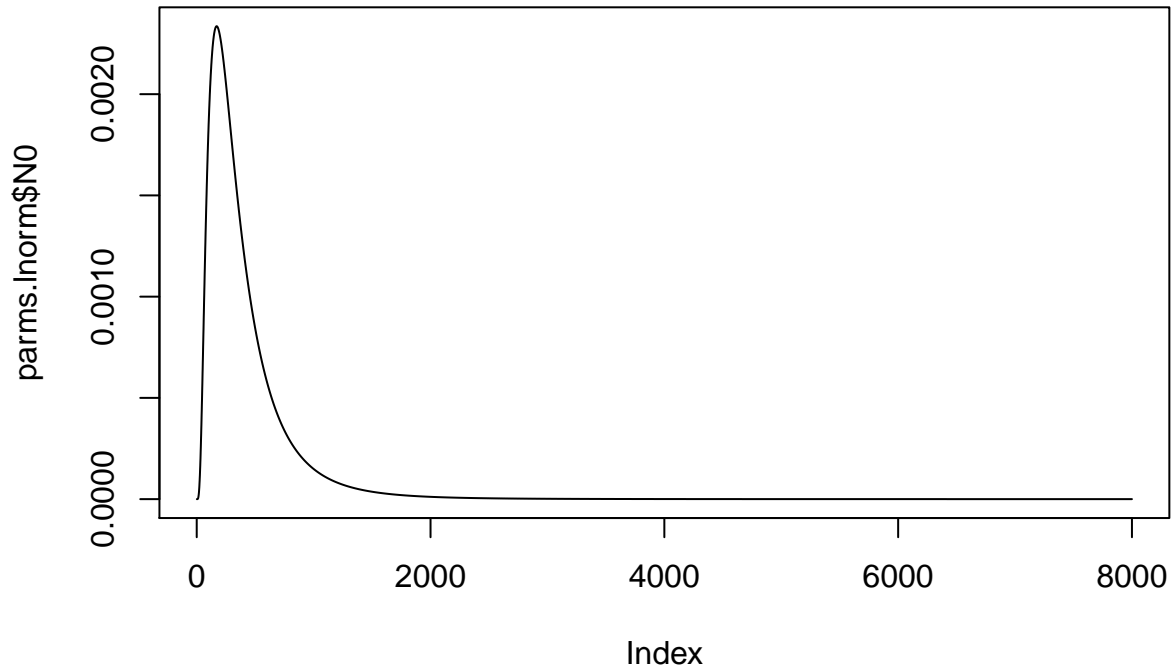
$$\begin{aligned}
\log(S^{r-2}) &= \log(e^C \frac{\kappa}{\kappa + 1}) \\
S^{r-2} &= e^C \frac{\kappa}{\kappa + 1} \\
S^{r-2}(\kappa + 1) &= e^C \kappa \\
S^{r-2} &= (e^C - S^{r-2})\kappa \\
\kappa &= \frac{S^{r-2}}{e^C - S^{r-2}}
\end{aligned}$$

How accurate is this approximation?

Ex 1 - lognormal distribution

Plot of probability distribution function:

```
plot(parms.lnorm$N0, type = "l")
```

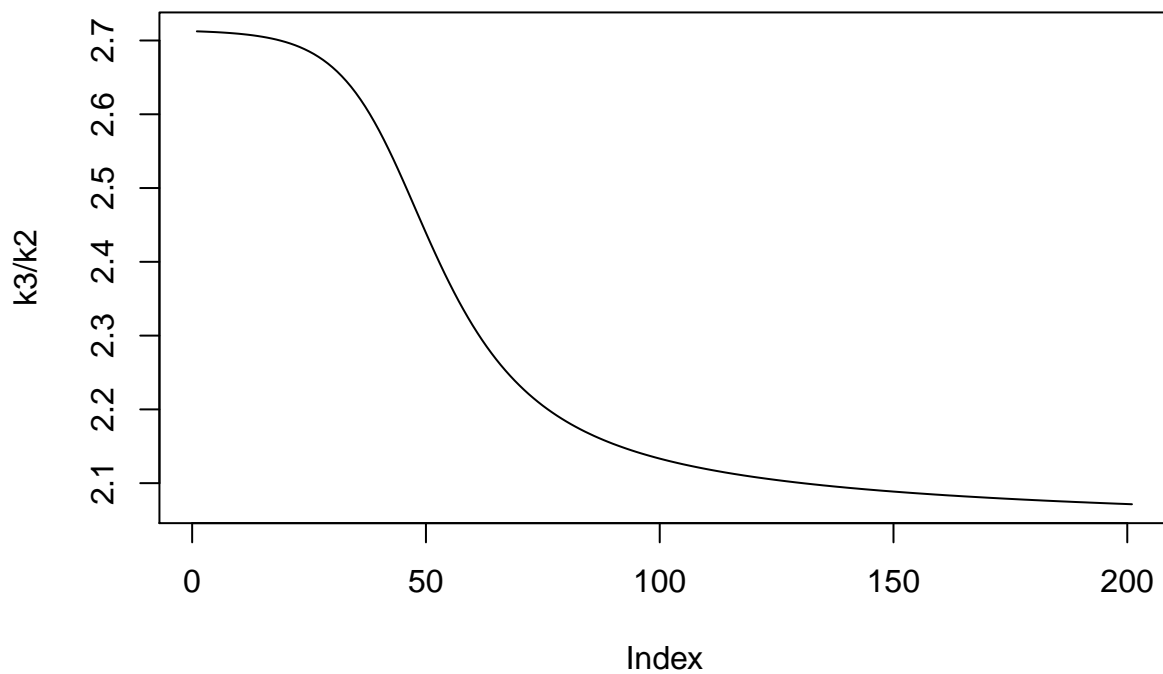


```
r.lnorm.het <- rk(unlist(yini.lnorm.het), func = hetero.model, parms = parms.lnorm, time = tvec)

c <- parms.lnorm$N.vec
N.l <- parms.lnorm$N.length
S.mat <- r.lnorm.het[,2:(N.l+1)]
totS <- rowSums(S.mat)
M1 <- rowSums(sweep(S.mat, 2, c, "*"))/rowSums(S.mat)
M2 <- rowSums(sweep(S.mat, 2, c^2, "*"))/rowSums(S.mat)
M3 <- rowSums(sweep(S.mat, 2, c^3, "*"))/rowSums(S.mat)
k2 <- M2/M1^2 - 1
k3 <- M3/(M2 * M1) - 1
```

What does κ_3/κ_2 look like?

```
plot(k3/k2, type = "l")
```

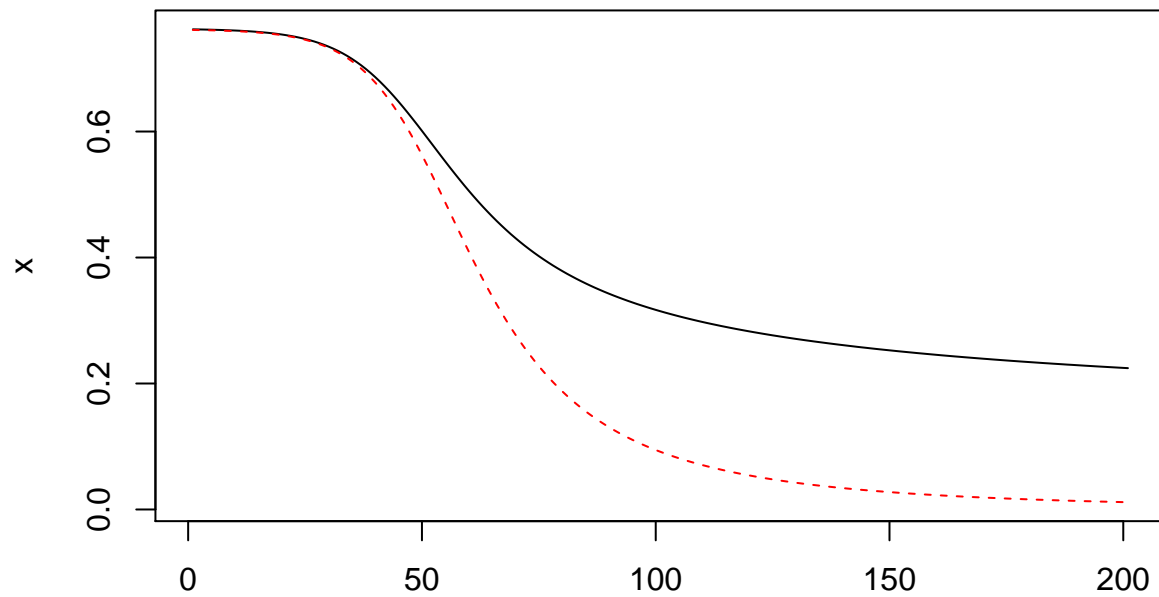


How well does our equation approximate κ ?

```
kappa0 <- k2[1]
r <- k3[1]/k2[1]
constant <- log((kappa0 + 1)/kappa0)

kappa.approx <- totS^(r-2)/(exp(constant) - totS^(r-2))

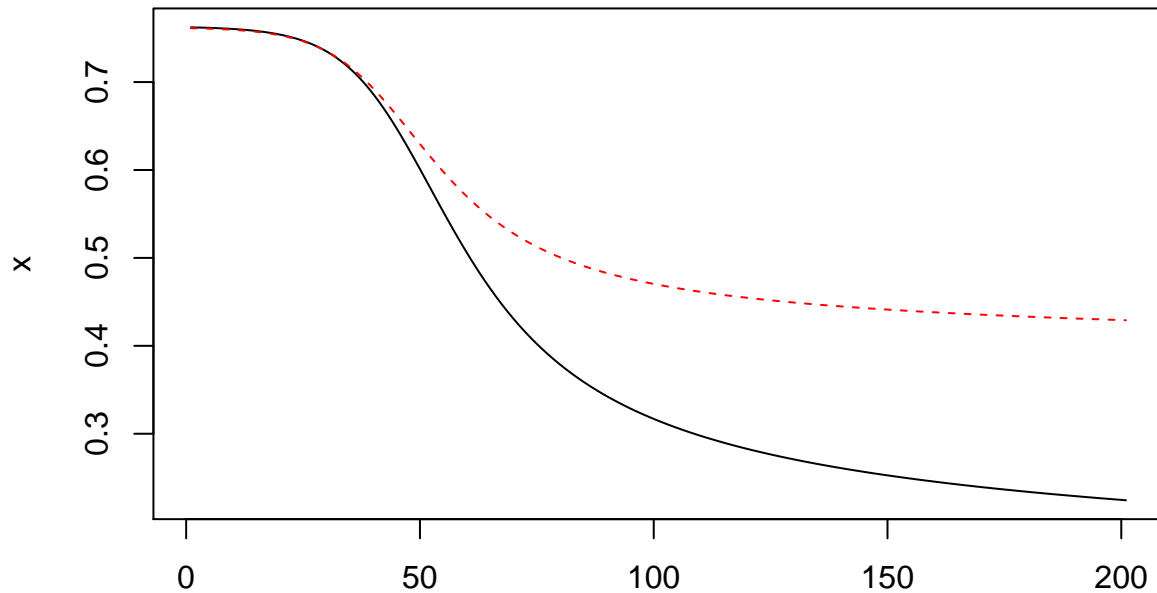
matplot(cbind(k2, kappa.approx), type = "l")
```



It seems like only the initial trajectory is similar... Also, this is a bad approximation because CV^2 goes 0. What happens if we replace r in the equation with κ_3/κ_2 instead of keeping it constant?

```
r2 <- k3/k2
kappa.approx2 <- totS^(r2-2)/(exp(constant) - totS^(r2-2))

matplot(cbind(k2, kappa.approx2), type = "l")
```



At least this doesn't cause κ to go to 0... Can we approximate r so that $\lim_{S \rightarrow 1} r = \hat{r}$ and $\lim_{S \rightarrow 0} r = 2$ using a linear equation?

$$r \approx (\hat{r} - 2)S + 2$$

Comparison of 3 simulations

- Heterogeneous model
- Approximated model: linear equation for κ_3/κ_2 coupled with approximated equation for κ
- Approximated model: constant κ

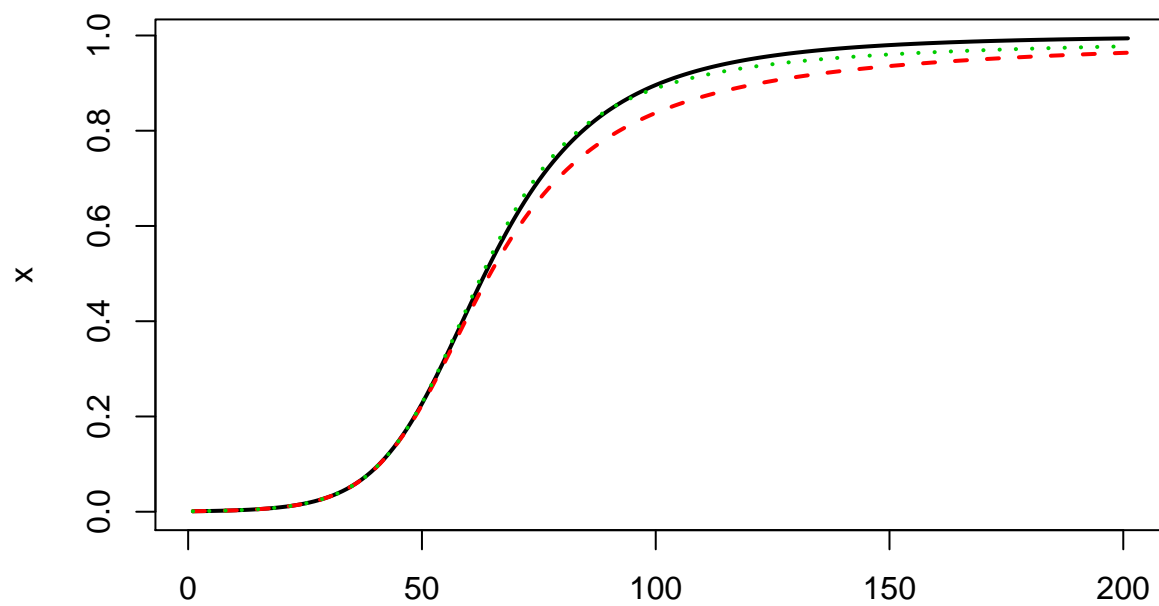
```
yini.lnorm.hom <- findY(parms.lnorm, type = "homo")

approx.model.T <- approx.model(parms.lnorm, constantK = TRUE)
approx.model.F <- approx.model(parms.lnorm, constantK = FALSE)

r.lnorm.app1 <- rk(unlist(yini.lnorm.hom), func = approx.model.T, parms = parms.lnorm, time = tvec)
r.lnorm.app2 <- rk(unlist(yini.lnorm.hom), func = approx.model.F, parms = parms.lnorm, time = tvec)
```

Prevalence trajectory

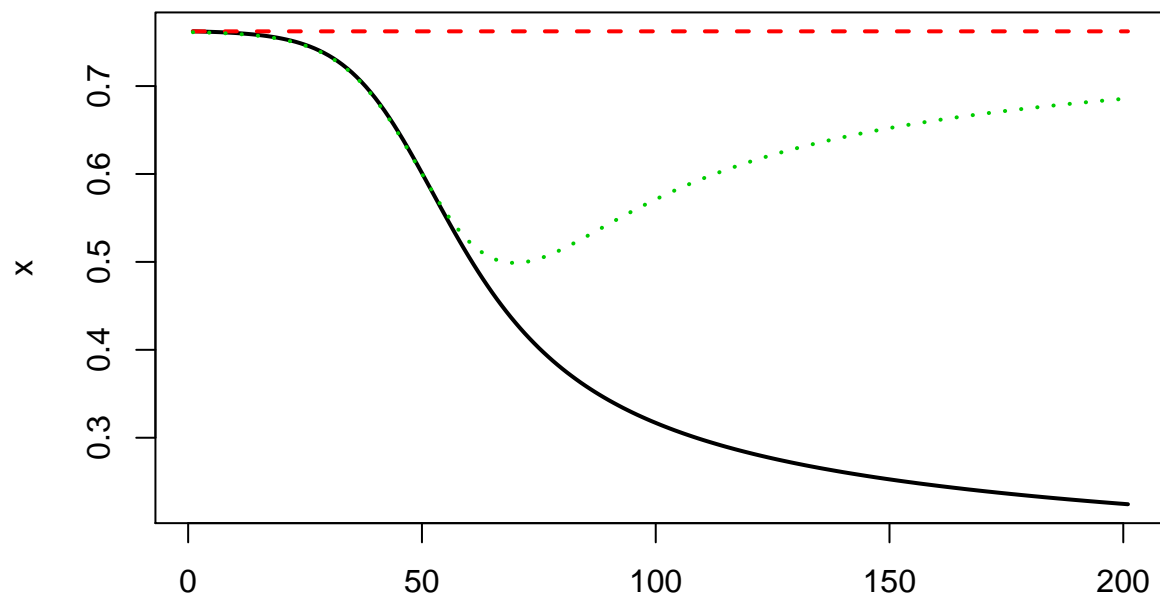
```
matplot(cbind(r.lnorm.het[,N.1+2], r.lnorm.app1[,3], r.lnorm.app2[,3]), type = "l", lwd = 2)
```



Linear approximation coupled with κ approximation seems like it does a fairly good job...

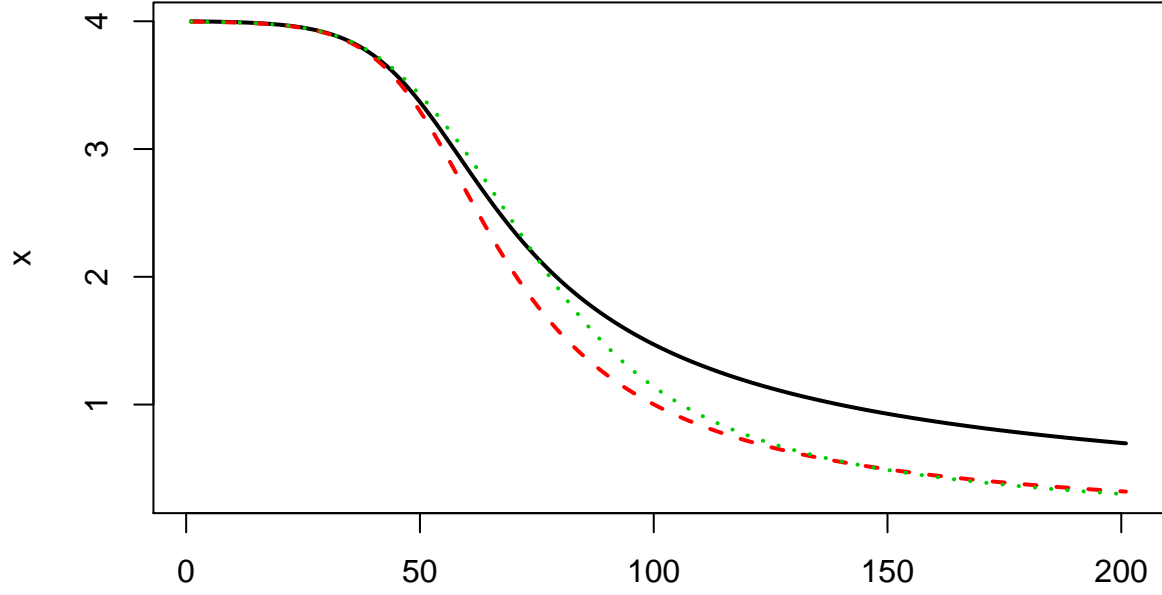
Squared CV

```
matplot(cbind(r.lnorm.het[,N.l+3], r.lnorm.app1[,4], r.lnorm.app2[,4]),type = "l", lwd =2)
```



Mean susceptibility

```
matplot(cbind(r.lnorm.het[,N.l+4], r.lnorm.app1[,5], r.lnorm.app2[,5]),type = "l", lwd=2)
```

SIS example

Here is a simple SIS model: $\dot{S}(a) = \mu(N(a) - S(a)) - \Lambda\sigma(a)S(a)$, where $N(a)$ is the initial distribution of the susceptible individuals in a disease free equilibrium. We are going to define $N_o = T_o(v; N)$. For this model, we have $\dot{T}_i = \mu(N_i - T_i) - \Lambda T_{i+1}$. Using chain rule, we can also get an equation for $M = M_1$:

$$\dot{M} = \mu(N_1 - MN_0)/T_0 - \Lambda\kappa M^2$$