

# Development Revisited



# How do we structure?

- Think about the components
- What are logically distinct parts?
  - Model and simulation details
  - Model setup and model code
  - Experiment and reporting



# Mixed

```
growth.rate <- 0.015
initial.human.pop <- 7 * 10^9
start.time <- 0
end.time <- 100
population.vec <- c(initial.human.pop)
timesteps <- seq(from=start.time + 1, to=end.time)
for (new.time in timesteps)
{
  current.count <- tail(population.vec, 1)
  new.additions <- growth.rate * current.count
  next.count <- current.count + new.additions
  population.vec <- c(population.vec, next.count)
}
plot(c(start.time, timesteps), population.vec)
```



# Model extracted

```
step_simple_growth <- function(current.population, growth.rate)
{
  new.additions <- growth.rate * current.population
  next.population <- current.population + new.additions
  next.population
}
human.annual.growth <- 0.015
initial.human.pop <- 7 * 10^9
start.time <- 0
end.time <- 100
population <- c(initial.human.pop)
timesteps <- seq(from=start.time + 1, to=end.time)
for (new.time in timesteps)
{
  updated.population <- step_simple_growth(current.population=tail(population, 1),
                                           growth.rate=human.annual.growth)
  population <- c(population, updated.population)
}
plot(c(start.time, timesteps), population)
```



# Tidied reporting

```
step_deterministic_growth <- function(latest, growth.rate)
{
  new.additions <- growth.rate * latest$count
  next.count <- latest$count + new.additions
  data.frame(count=next.count)
}
human.annual.growth <- 0.015
initial.human.pop <- 7 * 10^9
start.time <- 0
end.time <- 100
populations <- data.frame(count=initial.human.pop)

timesteps <- seq(from=start.time + 1, to=end.time)
for (new.time in timesteps)
{
  updated.population <- step_deterministic_growth(latest=tail(populations, 1),
                                                  growth.rate=human.annual.growth)
  populations <- rbind(populations, updated.population)
}
populations$time <- c(start.time, timesteps)
plot_populations(populations)
```



# But model still entangled

```
step_deterministic_growth <- function(latest, growth.rate)
{
  new.additions <- growth.rate * latest$count
  next.count <- latest$count + new.additions
  data.frame(count=next.count)
}
human.annual.growth <- 0.015
initial.human.pop <- 7 * 10^9
start.time <- 0
end.time <- 100
populations <- data.frame(count=initial.human.pop)
timestep <- 2
timesteps <- seq(from=start.time + timestep, to=end.time, by=timestep)
for (new.time in timesteps)
{
  updated.population <- step_deterministic_growth(latest=tail(populations, 1),
                                                  growth.rate=human.annual.growth * timestep)
  populations <- rbind(populations, updated.population)
}
populations$time <- c(start.time, timesteps)
plot_populations(populations)
```



# Model (fully?) separated

```
step_deterministic_growth <- function(latest, growth.rate, timestep)
{
  new.additions <- growth.rate * timestep * latest$count
  next.count <- latest$count + new.additions
  data.frame(count=next.count)
}
human.annual.growth <- 0.015
initial.human.pop <- 7 * 10^9
start.time <- 0
end.time <- 100
populations <- data.frame(count=initial.human.pop)
timestep <- 2
timesteps <- seq(from=start.time + timestep, to=end.time, by=timestep)
for (new.time in timesteps)
{
  updated.population <- step_deterministic_growth(latest=tail(populations, 1),
                                                  growth.rate=human.annual.growth,
                                                  timestep=timestep)
  populations <- rbind(populations, updated.population)
}
populations$time <- c(start.time, timesteps)
plot_populations(populations)
```



# Model (fully?) separated

```
num.cattle <- 100
initial.infecteds <- 2
initial.susceptibles <- num.cattle - initial.infecteds
ecoli.transmission <- 0.2
ecoli.recovery <- 0.1
start.time <- 0
end.time <- 200
populations <- data.frame(time=start.time,
                          susceptibles=initial.susceptibles,
                          infecteds=initial.infecteds)

timestep <- 2
latest.pop <- populations
while (latest.pop$time < end.time)
{
  latest.pop <- step_deterministic_SIS(latest=latest.pop,
                                     transmission.rate=ecoli.transmission,
                                     gamma=ecoli.recovery,
                                     timestep=timestep)

  populations <- rbind(populations, latest.pop)
}
plot_populations(populations)
```



# Simulation extracted

```
run_simulation <- function(step_function, latest.df, end.time, ...)
{
  population.df <- latest.df
  keep.going <- (latest.df$time < end.time)
  while (keep.going)
  {
    data <- step_function(latest.df, ...)
    latest.df <- data$updated.pop
    population.df <- rbind(population.df, latest.df)
    keep.going <- (latest.df$time < end.time) && (!data$end.experiment)
  }
  population.df
}
human.birth.rate <- 0.06
human.death.rate <- 0.02
initial.human.pop <- 1
start.time <- 0
end.time <- 100
initial.populations <- data.frame(time=start.time, count=initial.human.pop)
timestep <- 1
final.populations <- run_simulation(timestep_stochastic_birth_death,
                                   initial.populations,
                                   end.time,
                                   birth.rate=human.birth.rate,
                                   death.rate=human.death.rate,
                                   timestep=timestep)

plot_populations(final.populations)
```



# Simulation extracted

```
human.birth.rate <- 0.06
human.death.rate <- 0.02
initial.human.pop <- 1
start.time <- 0
end.time <- 100
initial.populations <- data.frame(time=start.time,
                                  count=initial.human.pop)

timestep <- 1
final.populations <- run_simulation(timestep_stochastic_birth_death,
                                   initial.populations,
                                   end.time,
                                   birth.rate=human.birth.rate,
                                   death.rate=human.death.rate,
                                   timestep=timestep)

plot_populations(final.populations)
```



# New subject area!

```
num.cattle <- 100
initial.infecteds <- 2
initial.susceptibles <- num.cattle - initial.infecteds
ecoli.transmission <- 1
ecoli.recovery <- 1/3
start.time <- 0
end.time <- 100
initial.populations <- data.frame(time=start.time,
                                  susceptibles=initial.susceptibles,
                                  infecteds=initial.infecteds)

timestep <- 0.1
final.populations <- run_simulation(timestep_stochastic_SIS,
                                   initial.populations,
                                   end.time,
                                   transmission.rate=ecoli.transmission,
                                   recovery.rate=ecoli.recovery,
                                   timestep=timestep)

plot_populations(final.populations)
```



# New model

```
num.cattle <- 100
initial.infecteds <- 2
initial.susceptibles <- num.cattle - initial.infecteds
ecoli.transmission <- 1
ecoli.recovery <- 1/3
start.time <- 0
end.time <- 100
initial.populations <- data.frame(time=start.time,
                                  susceptibles=initial.susceptibles,
                                  infecteds=initial.infecteds)

final.populations <- run_simulation(gillespie_stochastic_SIS,
                                   initial.populations,
                                   end.time,
                                   transmission.rate=ecoli.transmission,
                                   recovery.rate=ecoli.recovery)

plot_populations(final.populations)
```



# New model and technique

```
num.cattle <- 100
initial.infecteds <- 2
initial.susceptibles <- num.cattle - initial.infecteds
ecoli.transmission <- 1
ecoli.recovery <- 1/3
start.time <- 0
end.time <- 100
initial.populations <- data.frame(time=start.time,
                                  susceptibles=initial.susceptibles,
                                  infecteds=initial.infecteds)

timestep <- 0.1
final.populations <- run_integration(ode_deterministic_SIS,
                                     initial.populations,
                                     end.time,
                                     transmission.rate=ecoli.transmission,
                                     recovery.rate=ecoli.recovery,
                                     timestep=timestep)

plot_populations(final.populations)
```



# Old model and technique

```
num.cattle <- 100
initial.infecteds <- 2
initial.susceptibles <- num.cattle - initial.infecteds
ecoli.transmission <- 1
ecoli.recovery <- 1/3
start.time <- 0
end.time <- 100
initial.populations <- data.frame(time=start.time,
                                  susceptibles=initial.susceptibles,
                                  infecteds=initial.infecteds)

timestep <- 0.1
final.populations <- run_simulation(timestep_stochastic_SIS,
                                   initial.populations,
                                   end.time,
                                   transmission.rate=ecoli.transmission,
                                   recovery.rate=ecoli.recovery,
                                   timestep=timestep)

plot_populations(final.populations)
```



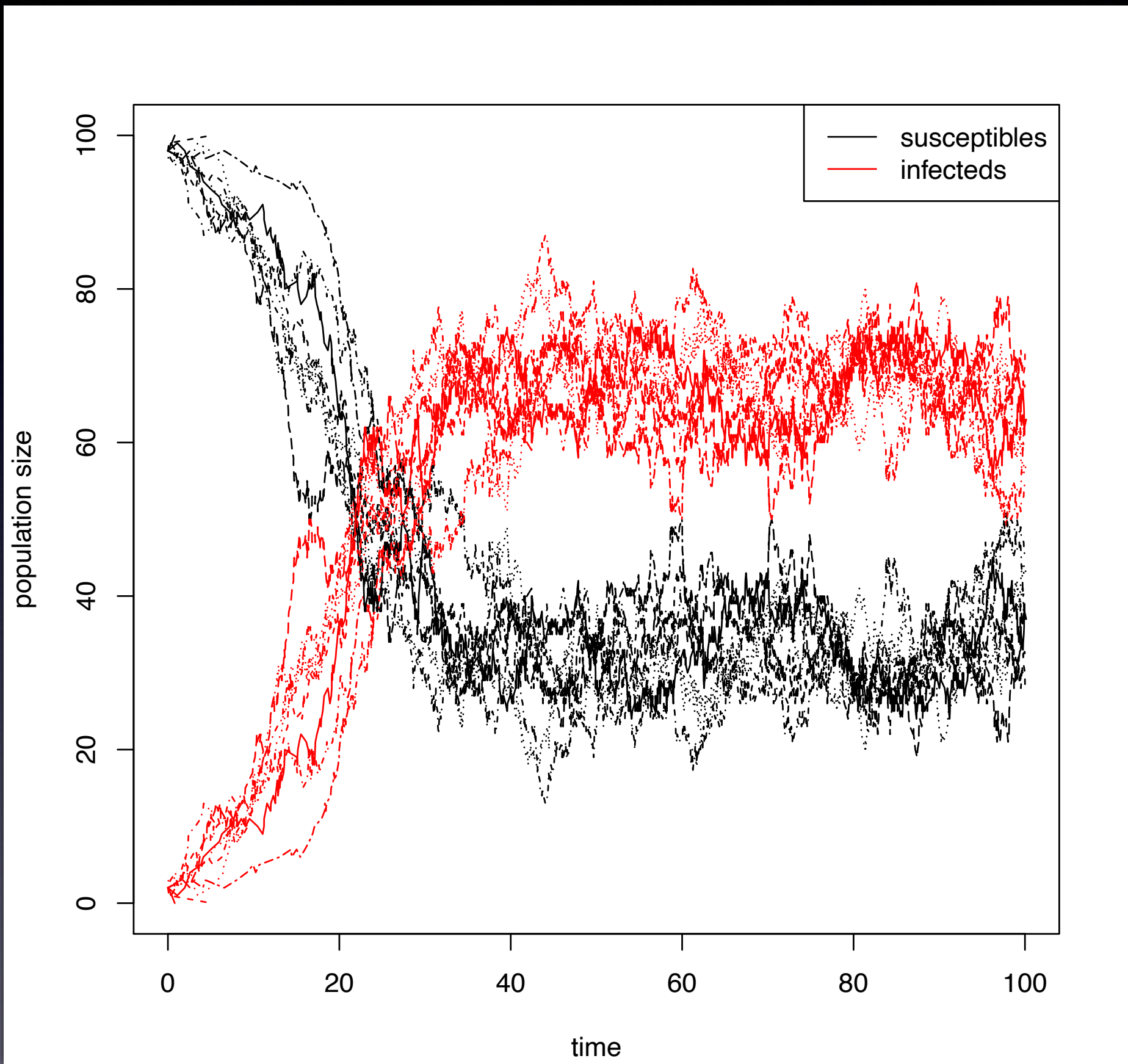
# Enhancing the analysis

```
num.cattle <- 100
initial.infecteds <- 2
initial.susceptibles <- num.cattle - initial.infecteds
ecoli.transmission <- 1
ecoli.recovery <- 1/3
start.time <- 0
end.time <- 100
initial.populations <- data.frame(time=start.time,
                                  susceptibles=initial.susceptibles,
                                  infecteds=initial.infecteds)

first.graph <- TRUE
for (i in 1:10)
{
  final.populations <- run_simulation(gillespie_stochastic_SIS,
                                    initial.populations, end.time,
                                    transmission.rate=ecoli.transmission,
                                    ecoli.recovery=ecoli.recovery)

  plot_populations(final.populations, new.graph=first.graph, xlim=c(0, end.time))
  first.graph <- FALSE
}
```







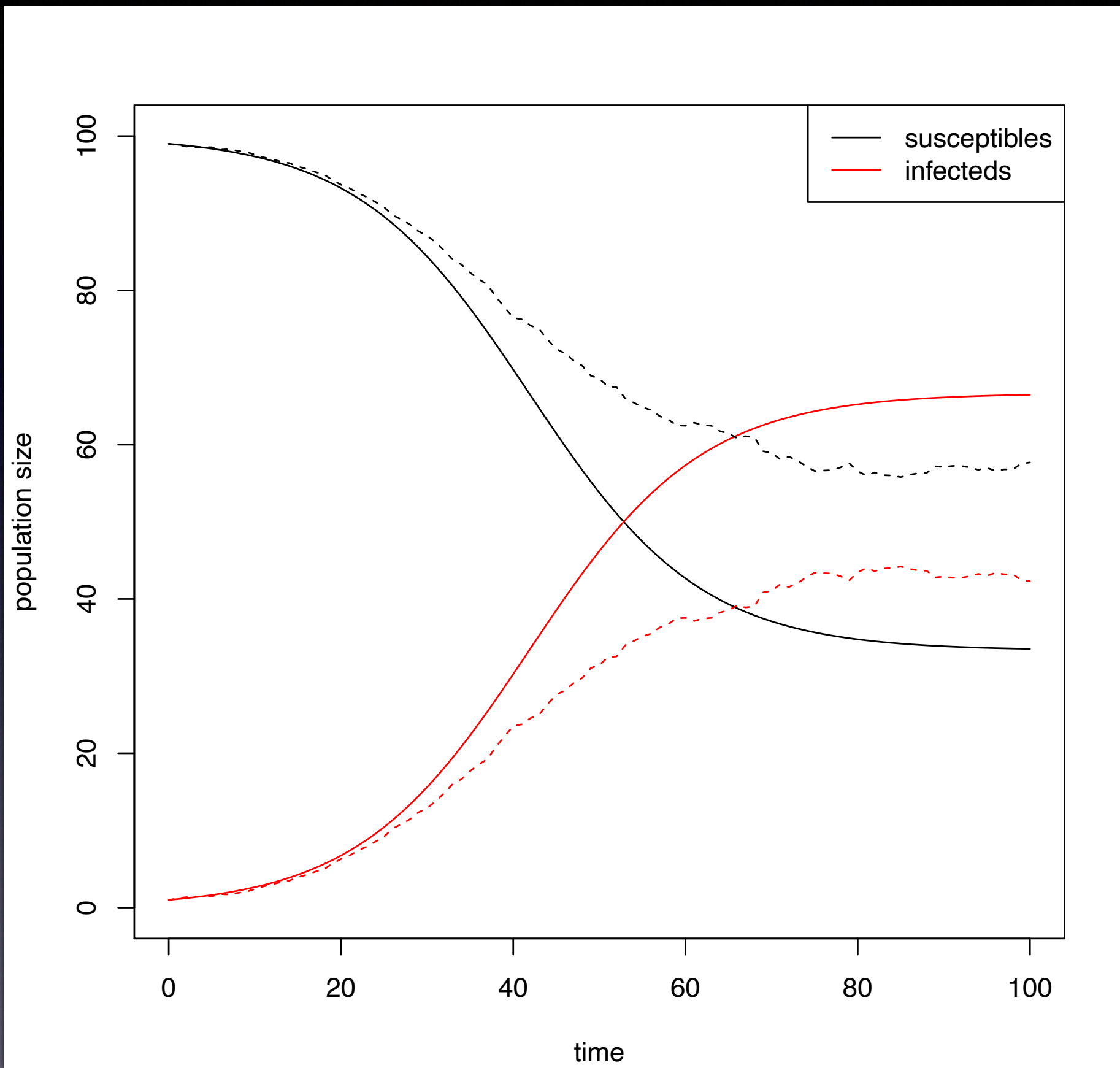
# Integration of models

```
timestep <- 1
final.ode <- run_integration(ode_deterministic_SIS,
                             initial.populations, end.time,
                             transmission.rate=ecoli.transmission,
                             recovery.rate=ecoli.recovery,
                             timestep=timestep)

number.to.average.over <- 20
for (i in 1:number.to.average.over)
{
  final.gillespie <- run_simulation(gillespie_stochastic_SIS,
                                   initial.populations, end.time,
                                   transmission.rate=ecoli.transmission,
                                   recovery.rate=ecoli.recovery)

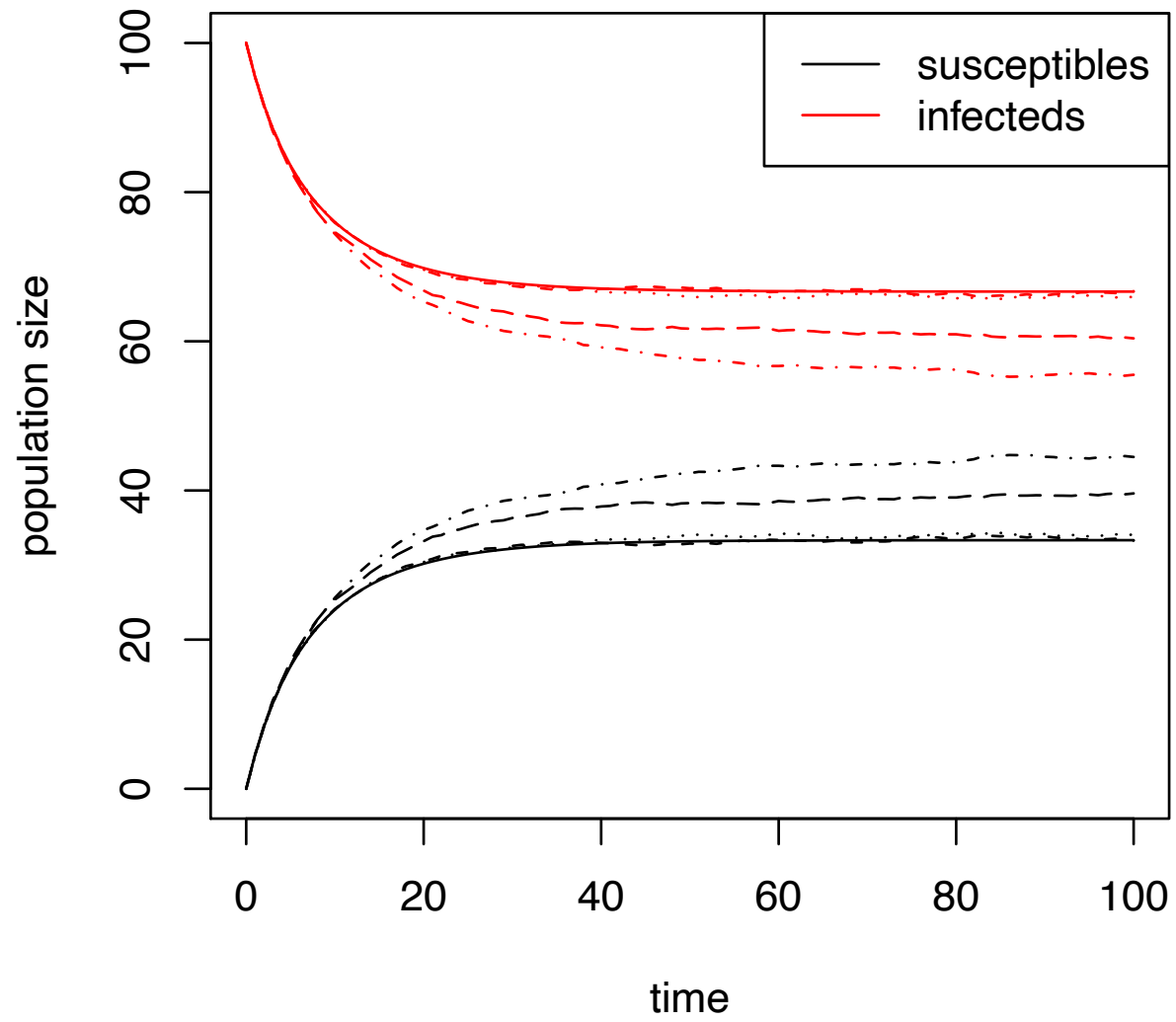
  total.gillespie <- total.gillespie + final.gillespie
}
mean.gillespie <- total.gillespie / number.to.average.over
plot_populations(final.ode)
plot_populations(mean.gillespie, new.graph=FALSE, lty=2)
```



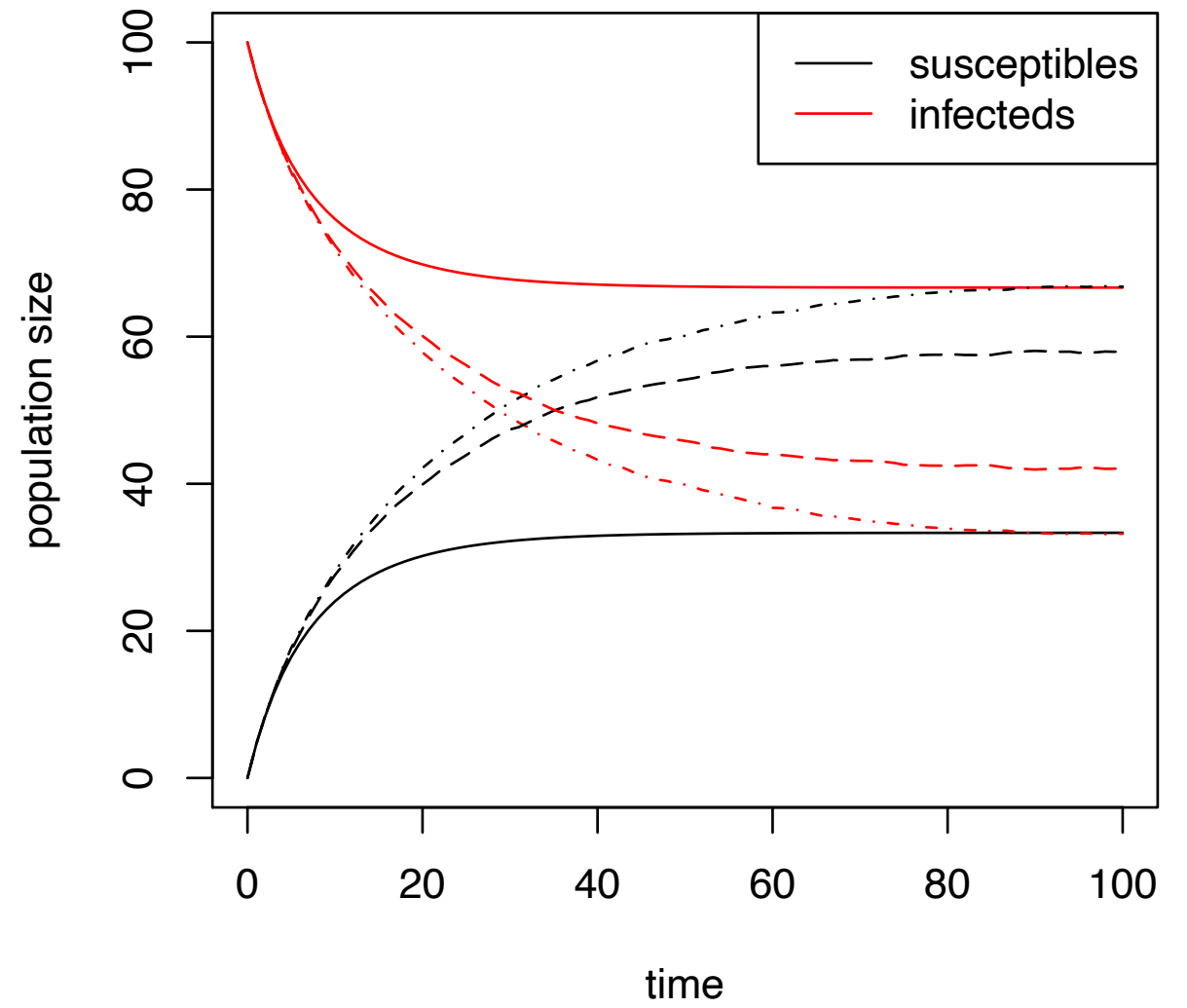




**Deterministic vs. stochastic vs. U(0,1)**



**Deterministic vs. discrete {0,1}**





# How do we structure?

- Think about the components
- What are logically distinct parts?
  - Model and simulation details
  - Model setup and model code
  - Experiment and reporting