

2. Gün

2. Uygulama: Rakip tehlikeler

1.1. Rakip tehlikeler

Modelimizi oluşturduk artık daha fazla bölme ve rakip tehlikeler durumunu inceleyerek daha karmaşık hale getirebiliriz. Sonraki alıştırmada enfekte kişiler arasındaki ölüm olasılığını tanımlamak için I'dan çıkan bir mortalite bölmesi eklemek istiyoruz.

Sonraki adımlar için modelleme yaptığımız hastalık için CFR'nin %30 olduğunu ortaya koyan bilgileri topladığımızı düşünelim. Bu bilgileri ve incelediğimiz kavramları kullanarak yeni M bölmesini dahil etmeye çalışın, bir mortalite oranı (μ) belirleyin ve modeli çalıştırın.

Aşağıdaki komut dizisindeki boşlukları doldurmaya çalışın:

```
## Kohort modelimize mortalite ekleyin

# Popülasyon büyüklüğü:
N <- 1000

# MODEL GİRDİLERİ:

# Başlangıç koşulları
initial_state_values <- c(
  I = N,
  ? # ←----- Fill
  R = 0)

# Günlük parametre değerleri

# %30'luk bir CFR için mu nedir?

mu= ? # ←----- Fill

parameters <- c(gamma = 1/2, mu)

# ZAMAN ADIMLARI:

# 0'dan 365 güne kadar günlük aralıklarla
times <- seq(from = 0, to = 50, by = 1) # zamanında modeli çözmek için zama
n adımları dizisini saklayan vektör

# MODEL FONKSİYONU:
```

Bulaşıcı hastalık dinamiklerinin R'de modellenmesi üzerine kısa kurs

```
# Model fonksiyonu şu girdi argümanları alır (yazılan sıraya göre): zaman, durum ve parametreler
cohort_model2 <- function(time, state, parameters) {

  with(as.list(c(state, parameters)), {

    # N toplam popülasyon boyutunun (her bir bölmedeki insan sayısı toplamı) hesaplanması
    N <- I+R

    # Diferansiyel denklemler
    dI <- -(gamma+ ?) * I # ←----- Fill
    dM <- ?                # ←----- Fill
    dR <- gamma * I

    return(list(c(dI, ?, dR))) # ←----- Fill
  })
}

# MODEL ÇIKTISI (diferansiyel denklemlerin çözülmesi):

# Ode entegrasyon algoritması kullanarak diferansiyel denklemlerin çözümü
output2 <- as.data.frame(ode(y = initial_state_values,
                             times = times,
                             func = cohort_model2,
                             parms = parameters))

# Çıktı veri setini uzun formata dönüştürün
output2_long <- melt(as.data.frame(output2), id = "time")

# Yeni çıktının grafiğini çizin
ggplot(data = output2_long,
        aes(x = time, y = value, colour = variable, group = variable)) +
  geom_line(size=2) +
  xlab("Time (days)") +
  ylab("Number") +
  labs(colour = "Compartment",
       title = "Cohort model")
```

1.2. Simülasyonumuzdan CFR'yi bulun

Artık modelimize mortaliteyi ekledik, deneyin ve modelden tahmin edilen CFR'nin %30'a karşılık gelip gelmediğini kontrol edin

```
# Bunun için her bir bölmedeki insan oranını görmek isteyebiliriz
```

```
output3_long<- output2_long  
output3_long$value<-output2_long$value/N
```

```
# S, I ve R bölmelerindeki insanların oranının zamana bağlı grafiğini çizin
```

```
ggplot(data = output3_long,  
       aes(x = time, y = value, colour = variable, group = variable)) +  
  geom_line(size=2) +  
  xlab("Time (days)") +  
  ylab("Number") +  
  labs(colour = "Compartment",  
       title = "Cohort model")
```

```
# Grafiğe bakarak hastalık nedeniyle ölen insanların oranını söyleyebilir misiniz?
```