

ThemaOpdracht

Muniru

25/04/2022

Thema Opdracht 8

Les 1 Opdrachten

1.0

```
rate = lambda Y : (Y + 10) * .9
value = 0
previous = 0
for n in range(30):
    print(f"{n}\tValue:{value} \tPrevious:{previous}")
    previous = value
    value = round(rate(value))
```

```
## 0    Value:0    Previous:0
## 1    Value:9    Previous:0
## 2    Value:17   Previous:9
## 3    Value:24   Previous:17
## 4    Value:31   Previous:24
## 5    Value:37   Previous:31
## 6    Value:42   Previous:37
## 7    Value:47   Previous:42
## 8    Value:51   Previous:47
## 9    Value:55   Previous:51
## 10   Value:58   Previous:55
## 11   Value:61   Previous:58
## 12   Value:64   Previous:61
## 13   Value:67   Previous:64
## 14   Value:69   Previous:67
## 15   Value:71   Previous:69
## 16   Value:73   Previous:71
## 17   Value:75   Previous:73
## 18   Value:76   Previous:75
## 19   Value:77   Previous:76
## 20   Value:78   Previous:77
## 21   Value:79   Previous:78
## 22   Value:80   Previous:79
## 23   Value:81   Previous:80
## 24   Value:82   Previous:81
```

```
## 25 Value:83 Previous:82
## 26 Value:84 Previous:83
## 27 Value:85 Previous:84
## 28 Value:86 Previous:85
## 29 Value:86 Previous:86
```

[1] Welke parameters (constanten) zitten er in de vergelijking?

De +10.

[2] Welke variabele uit de vergelijking verandert steeds?

Het aantal M&M's.

[3] Wat is de initiële waarde van de variabele uit vraag 2?

0

[4] Wat is het tijdsframe van dit experiment? ($t = 0$ tot $t = \dots$)

29

[5] Wanneer is het evenwichtsmoment?

29

[6] Uit welke variabele uit de vergelijking kunnen we dat aflezen?

Dat is uit de input en output af te lezen want die zijn het zelfde.

2.0

```
library(deSolve)

parameters <- c(addVolume = 10, pV = 0.1)

# define model
volume <- function(t,y,parms){
  with(as.list(c(parms)),{
    dY <- addVolume - pV * (y + addVolume)
    return(list(c(dY)))
  })
}

#initial state
state <- c(Volume = 0)

#define time sequence you want to run the model
```

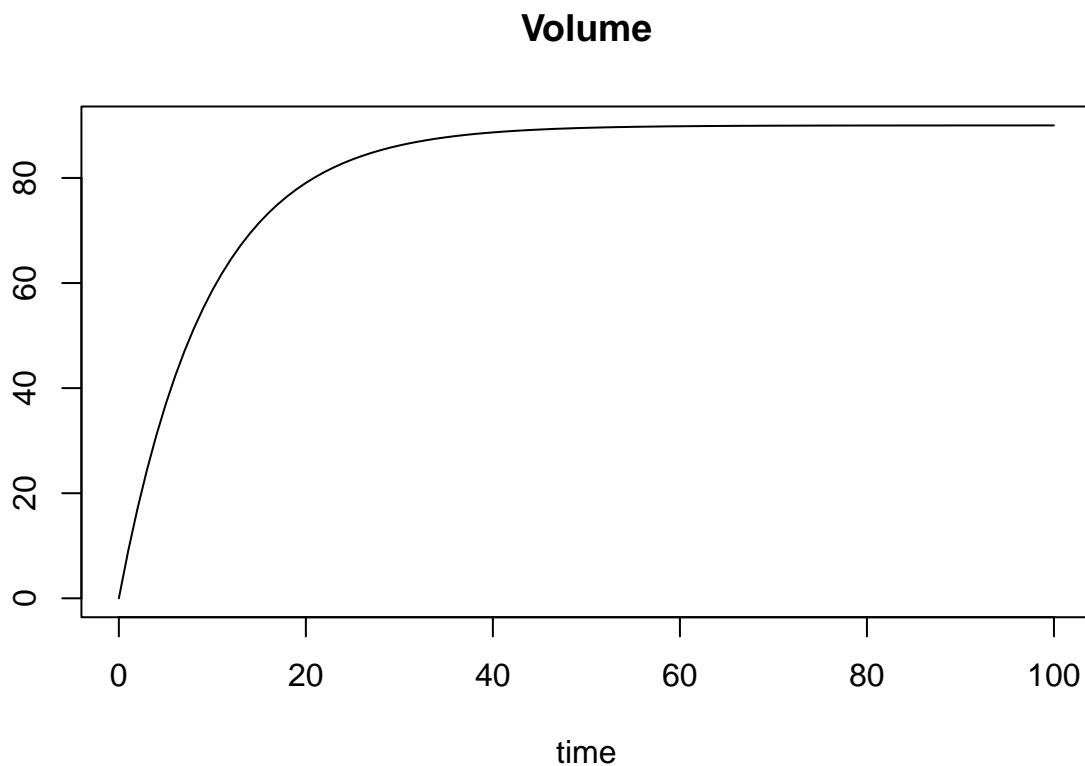
```
times <- seq(0, 100, by = 1)

# run simulation using continuous approach
out <- ode(times = times, y = state, parms = parameters, func = volume, method = "euler")

head(out)
```

```
##      time  Volume
## [1,]    0  0.0000
## [2,]    1  9.0000
## [3,]    2 17.1000
## [4,]    3 24.3900
## [5,]    4 30.9510
## [6,]    5 36.8559
```

```
plot(out)
```



[1] Verander de parameters. Wat is het effect?

De add volume heeft effect op het volume waar die stagneerd. en de pV op de boog van de lijn.

[2] Verander de initiële waarde. Wat is het effect?

Dit heeft invloed om hoe groot het verschil is tussen de waarden aan de y as.

[3] Verander de timeframe. Wat is het effect?

Het effect hiervan is te zien op de lijn. Er komen minder meetwaarden en daardoor komen er minder hoeken

[4] Verander het originele model zodat de correctie niet is op basis van $(y + \text{addVolume})$ maar y . Wanneer is er nu een evenwichts toestand?

Bij 161.

3.0

Afwezig

Week 1 Opdracht

1.0

[1] Welke parameters moet je programmeren?

R: Het nummer van transcripties

r: De afname van de snelheid van de transcripties

m: het nummer van nieuwe transcripties die worden geproduceerd per sec

[2] Zoek bronnen op internet die uitleggen waar de formule $dR/dt = -rR + m$ vandaan komt.

<https://arep.med.harvard.edu/pdf/Chen99.pdf>

[3] Teken het biologisch model en leg de vertaling naar de formule uit.

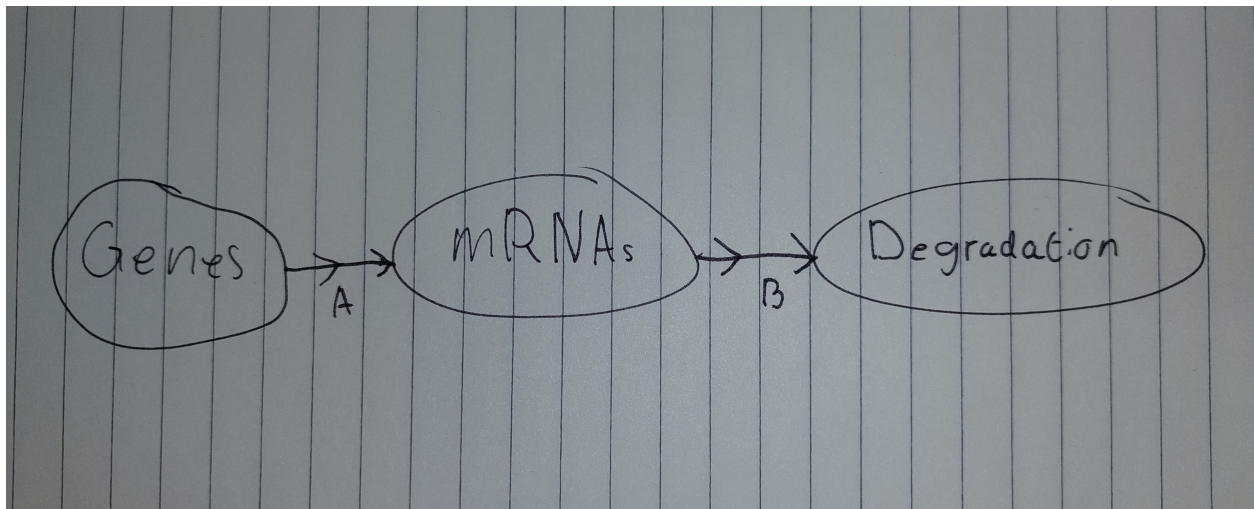


Figure 1: Genes—A—>mRNAs—B—>Degredation

[4] Wat is de return waarde van de model functie in R? Waarom return je die en niet R zelf?