HW06_Diffusion

Dani Sclafani, Julia Parish, Kristin Gill

2022-05-05

Load Libraries

```
library(tidyverse)
library(deSolve)
library(here)
```

Source Diffusion Function

```
source(here("R/diffusion.R"))
```

Diffusion Model Parameters

- `initialC` = initial concentration (mg/L)
- `dx` = length of each segment (m)
- `nx` = number of discrete segments (m)
- `nt` = number of discrete time intervals (s)
- `dt` = seconds in each time interval (s)
- `area` = area of cross section of container (m2)
- `D` = diffusivity (how easily the chemical diffuses (s/m2)

Run the in class difdussion model examples

```
# run our diffusion model (iterative difference equation) with initial concentration of 10, for 8 times

# using diffusion parameters 0.5 s/m2, 10 m2
result = diff1(initialC = 10, nx = 10, dx = 1, nt = 8, dt = 1, D = 0.5, area = 10)

# a list is returned with our 3 data frames for concentration (conc), qin and qout
result
```

```
## $conc
##           [,1]     [,2]     [,3]      [,4]      [,5]        [,6]        [,7]
## [1,] 10.000000 0.000000 0.000000 0.0000000 0.0000000 0.000000000 0.000000000
## [2,]  7.500000 2.500000 0.000000 0.0000000 0.0000000 0.000000000 0.000000000
## [3,]  6.250000 3.125000 0.625000 0.0000000 0.0000000 0.000000000 0.000000000
## [4,]  5.468750 3.281250 1.093750 0.1562500 0.0000000 0.000000000 0.000000000
## [5,]  4.921875 3.281250 1.406250 0.3515625 0.0390625 0.000000000 0.000000000
## [6,]  4.511719 3.222656 1.611328 0.5371094 0.1074219 0.009765625 0.000000000
## [7,]  4.189453 3.142090 1.745605 0.6982422 0.1904297 0.031738281 0.002441406
## [8,]  3.927612 3.054810 1.832886 0.8331299 0.2777100 0.064086914 0.009155273
##              [,8] [,9] [,10]
```

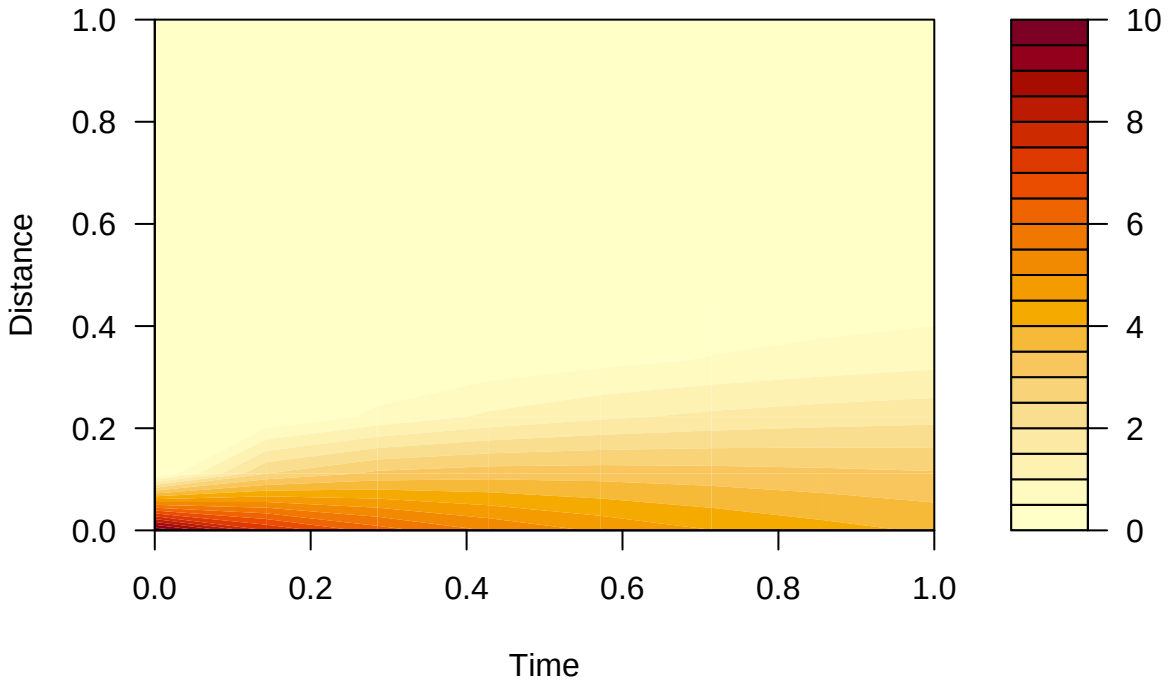
```
## [1,] 0.0000000000    0     0
## [2,] 0.0000000000    0     0
## [3,] 0.0000000000    0     0
## [4,] 0.0000000000    0     0
## [5,] 0.0000000000    0     0
## [6,] 0.0000000000    0     0
## [7,] 0.0000000000    0     0
## [8,] 0.0006103516    0     0
##
## $qout
##           [,1]     [,2]     [,3]     [,4]       [,5]       [,6]        [,7]
## [1,] 25.000000 0.000000 0.000000 0.000000 0.00000000 0.00000000 0.000000000
## [2,] 12.500000 6.250000 0.000000 0.000000 0.00000000 0.00000000 0.000000000
## [3,]  7.812500 6.250000 1.562500 0.000000 0.00000000 0.00000000 0.000000000
## [4,]  5.468750 5.468750 2.343750 0.390625 0.00000000 0.00000000 0.000000000
## [5,]  4.101562 4.687500 2.636719 0.781250 0.09765625 0.00000000 0.000000000
## [6,]  3.222656 4.028320 2.685547 1.074219 0.24414062 0.02441406 0.000000000
## [7,]  2.618408 3.491211 2.618408 1.269531 0.39672852 0.07324219 0.006103516
## [8,]  0.000000 0.000000 0.000000 0.000000 0.00000000 0.00000000 0.000000000
##      [,8] [,9] [,10]
## [1,]    0    0     0
## [2,]    0    0     0
## [3,]    0    0     0
## [4,]    0    0     0
## [5,]    0    0     0
## [6,]    0    0     0
## [7,]    0    0     0
## [8,]    0    0     0
##
## $qin
##      [,1]      [,2]     [,3]     [,4]     [,5]       [,6]       [,7]
## [1,]    0 25.000000 0.000000 0.000000 0.000000 0.00000000 0.00000000
## [2,]    0 12.500000 6.250000 0.000000 0.000000 0.00000000 0.00000000
## [3,]    0  7.812500 6.250000 1.562500 0.000000 0.00000000 0.00000000
## [4,]    0  5.468750 5.468750 2.343750 0.390625 0.00000000 0.00000000
## [5,]    0  4.101562 4.687500 2.636719 0.781250 0.09765625 0.00000000
## [6,]    0  3.222656 4.028320 2.685547 1.074219 0.24414062 0.02441406
## [7,]    0  2.618408 3.491211 2.618408 1.269531 0.39672852 0.07324219
## [8,]    0  0.000000 0.000000 0.000000 0.000000 0.00000000 0.00000000
##             [,8] [,9] [,10]
## [1,] 0.000000000    0     0
## [2,] 0.000000000    0     0
## [3,] 0.000000000    0     0
## [4,] 0.000000000    0     0
## [5,] 0.000000000    0     0
## [6,] 0.000000000    0     0
## [7,] 0.006103516    0     0
## [8,] 0.000000000    0     0
```

```
# used filled contour to plot results
head(result$conc)
```

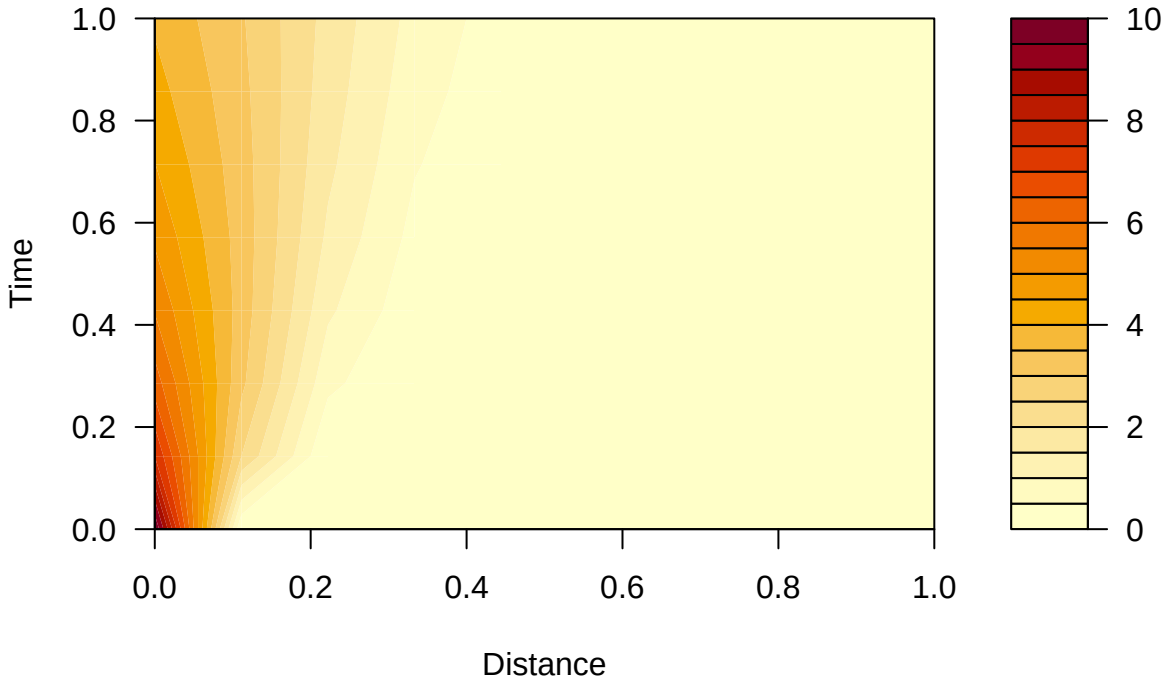
```
##           [,1]     [,2]     [,3]      [,4]      [,5]        [,6] [,7] [,8] [,9]
## [1,] 10.000000 0.000000 0.000000 0.0000000 0.0000000 0.000000000    0    0    0
## [2,]  7.500000 2.500000 0.000000 0.0000000 0.0000000 0.000000000    0    0    0
```

```
## [3,]  6.250000 3.125000 0.625000 0.0000000 0.0000000 0.000000000    0    0    0
## [4,]  5.468750 3.281250 1.093750 0.1562500 0.0000000 0.000000000    0    0    0
## [5,]  4.921875 3.281250 1.406250 0.3515625 0.0390625 0.000000000    0    0    0
## [6,]  4.511719 3.222656 1.611328 0.5371094 0.1074219 0.009765625    0    0    0
##      [,10]
## [1,]     0
## [2,]     0
## [3,]     0
## [4,]     0
## [5,]     0
## [6,]     0
```

```
filled.contour(result$conc, xlab="Time", ylab="Distance")
```




```
# or if you prefer this orientation (Distance on x axis)
filled.contour(t(result$conc), ylab="Time", xlab="Distance")
```

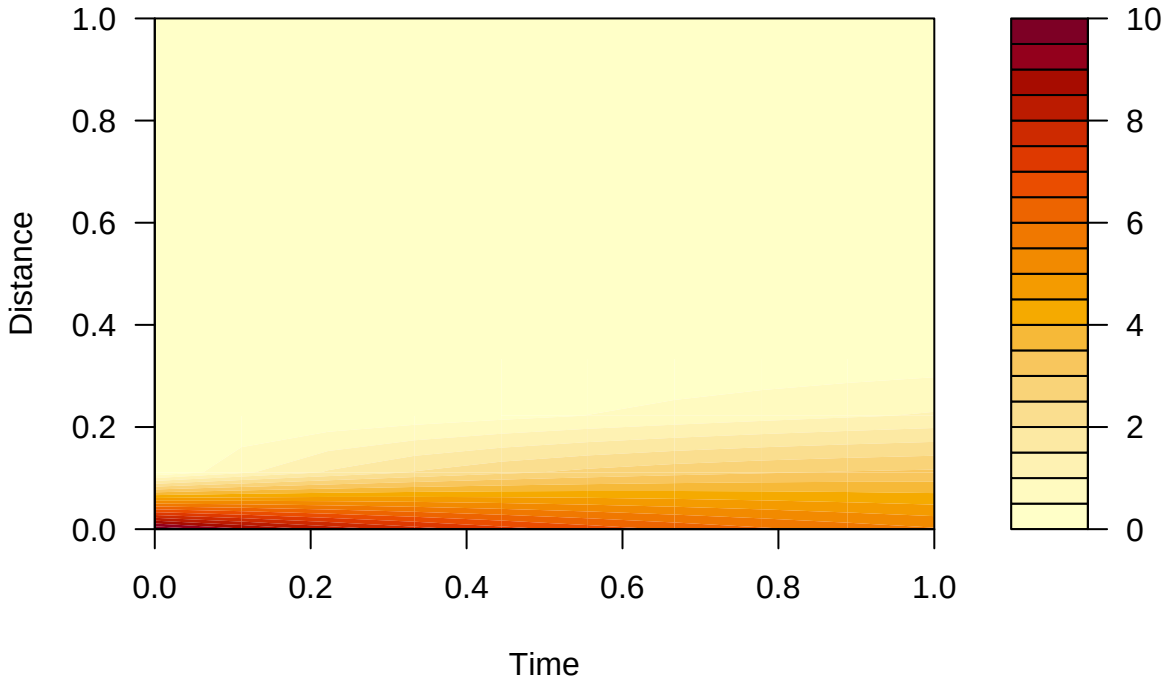


In class example of changing parameters (diffusivity D, and space and time steps (dx, dt))

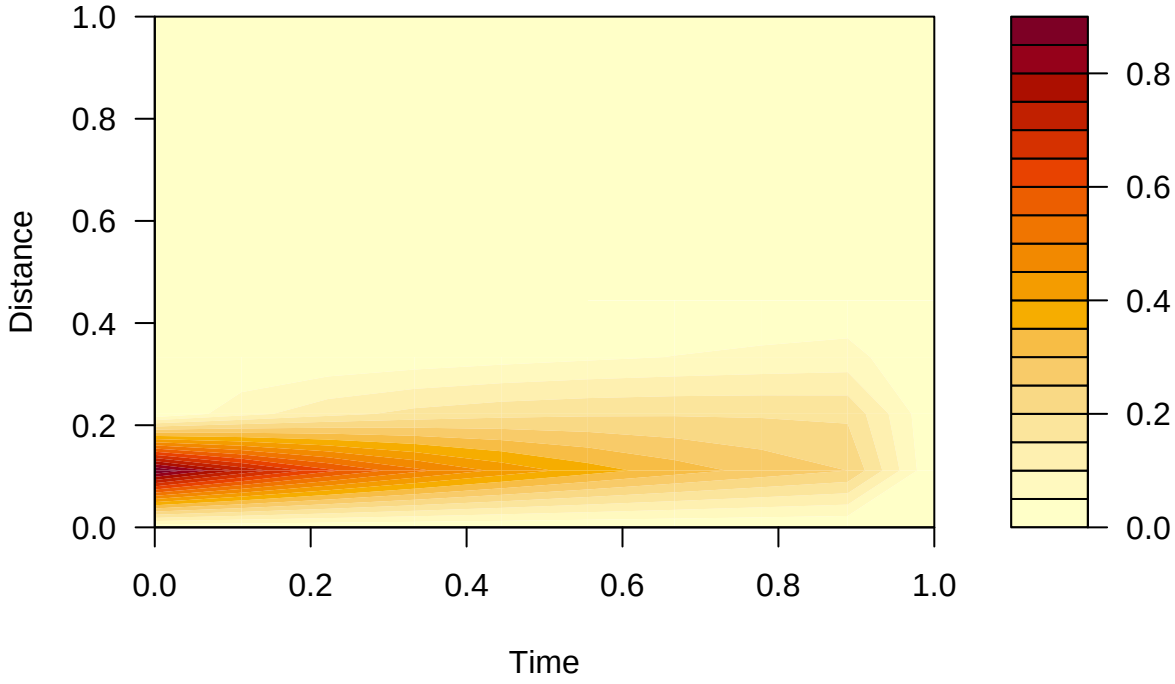
```
# changes diffusivity and other parameters particularly
# diffusivity, dx and dt

res = diff1(initialC = 10, nx = 10, dx = 1, nt = 10, dt = 30, D = 0.006, area = 1)

filled.contour(res$conc, xlab="Time", ylab="Distance")
```




```
# we can also see how much material moved from place to place each time step
filled.contour(res$qin, xlab="Time", ylab="Distance")
```



Homework Responses

Change diffusivity and length of segment

Change space step (dx), time step (dt)