differential-equations

Following the tutorial from:

Differential Equations in R Part : Representing Basic Dynamics https://www.youtube.com/watch?v=1iNXQypailI

Note: I fixed a few mistakes with the functions where it was using more global scope than I wanted.

Install the library deSolve from Packages -> Install

Use the library deSolve

```
library(deSolve)
```

Solving the continous equation

$$\frac{dN}{dt} = rN$$

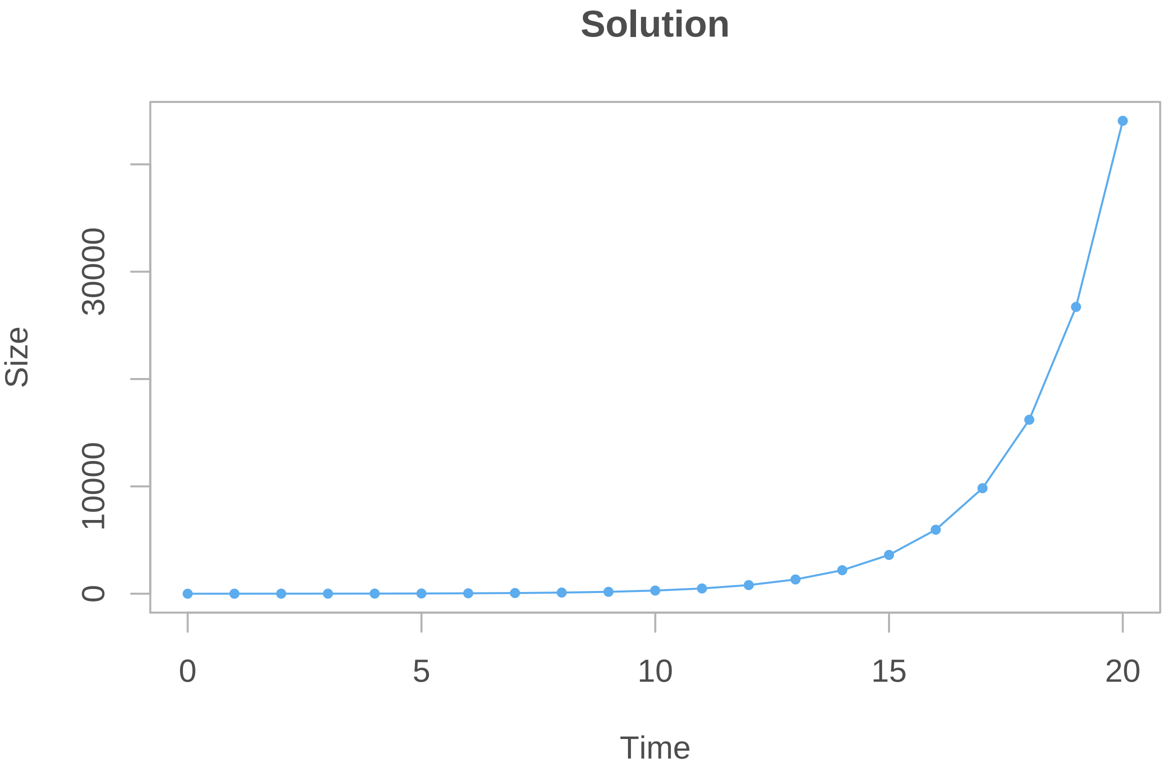
Create a function

```
cgrowth <- function(times, y, parms) {
  r <- parms[1]
  N <- y[1]
  dN.dt <- r * N
  return(list(dN.dt))
}

p  <- 0.5
y0 <- 2
t  <- 0:20

sol <- ode(y = y0, times = t, func = cgrowth, parms = p)
sol
```

```
##    time            1
## 1     0     2.000000
## 2     1     3.297445
## 3     2     5.436572
## 4     3     8.963395
## 5     4    14.778153
## 6     5    24.365058
## 7     6    40.171205
## 8     7    66.231149
## 9     8   109.196755
## 10    9   180.035094
## 11   10   296.827805
## 12   11   489.386525
## 13   12   806.862352
## 14   13  1330.291724
## 15   14  2193.281112
## 16   15  3616.110779
## 17   16  5961.961552
## 18   17  9829.617263
## 19   18 16206.305348
## 20   19 26719.691878
## 21   20 44053.345008
```

Plotting it would be

```
plot(sol, type='o', xlab="Time", ylab="Size", main="Solution",
     pch=16, cex=0.7, fg="grey70", col="steelblue2", col.axis="grey30",
     col.lab="grey30", col.main="grey30")
```



Solving a continous time logistic equation numerically

$$\frac{dN}{dt} = rN(1 - \frac{N}{K})$$

```
clogistic <- function(times, y, params) {
  r <- params[1]
  K <- params[2]
  N <- y[1]
  dN.dt <- r * N * (1 - (N/params[2]))
  return(list(dN.dt))
}
```