

Systems Biology Assignment (Problem Set 2)

Group members: Akansha (215HSBB003), Anirjit (215HSBB006),
Shreya (215HSBB034), Sukanya (215HSBB035)

Problem-1: Solve the following set of coupled ODE's in R.

$$\begin{aligned}\frac{dx_1}{dt} &= x_1 - x_2 \\ \frac{dx_2}{dt} &= -x_2 \\ \frac{dx_3}{dt} &= 10x_1 - x_3^2\end{aligned}$$

with the initial condition $x_1 = x_2 = x_3 = 0$ when $t = 0$

Plot the solutions x_1 , x_2 and x_3 as a function of t on the same graph.

Code

```
# Question1 - Solve the following set of coupled ODE's in R.
# dx1/dt = x1 - x2
# dx2/dt = -x2
# dx3/dt = 10x1 - x3^2
# with the initial condition x1 = x2 = x3 = 0 when t = 0
# Plot the solutions x1, x2 and x3 as a function of t on the same graph

library(deSolve)
library(scatterplot3d)

rigidnode<-function(t,y,params)
{
  #change symbols
  x1=y[1]
  x2=y[2]
  x3=y[3]

  dx1=x1-x2
  dx2=-x2
  dx3=10*x1-x3^2

  list(c(dx1,dx2,dx3))

  #In R, anything that is written lastly is automatically returned in the function
}

initial=c(x1=0,x2=0,x3=0)
parameters=c(0,0,0)
time=seq(0,20,0.1)
```

```
out<-ode(times=time,y=initial,func=rigidnode,parms=parameters)
```

```
plot(out)
```

```
initial=c(x1=1,x2=1,x3=1)
```

```
parameters=c(0,0,0)
```

```
time=seq(0,20,0.1)
```

```
out2<-ode(times=time,y=initial,func=rigidnode,parms=parameters)
```

```
plot(out2)
```

Plots

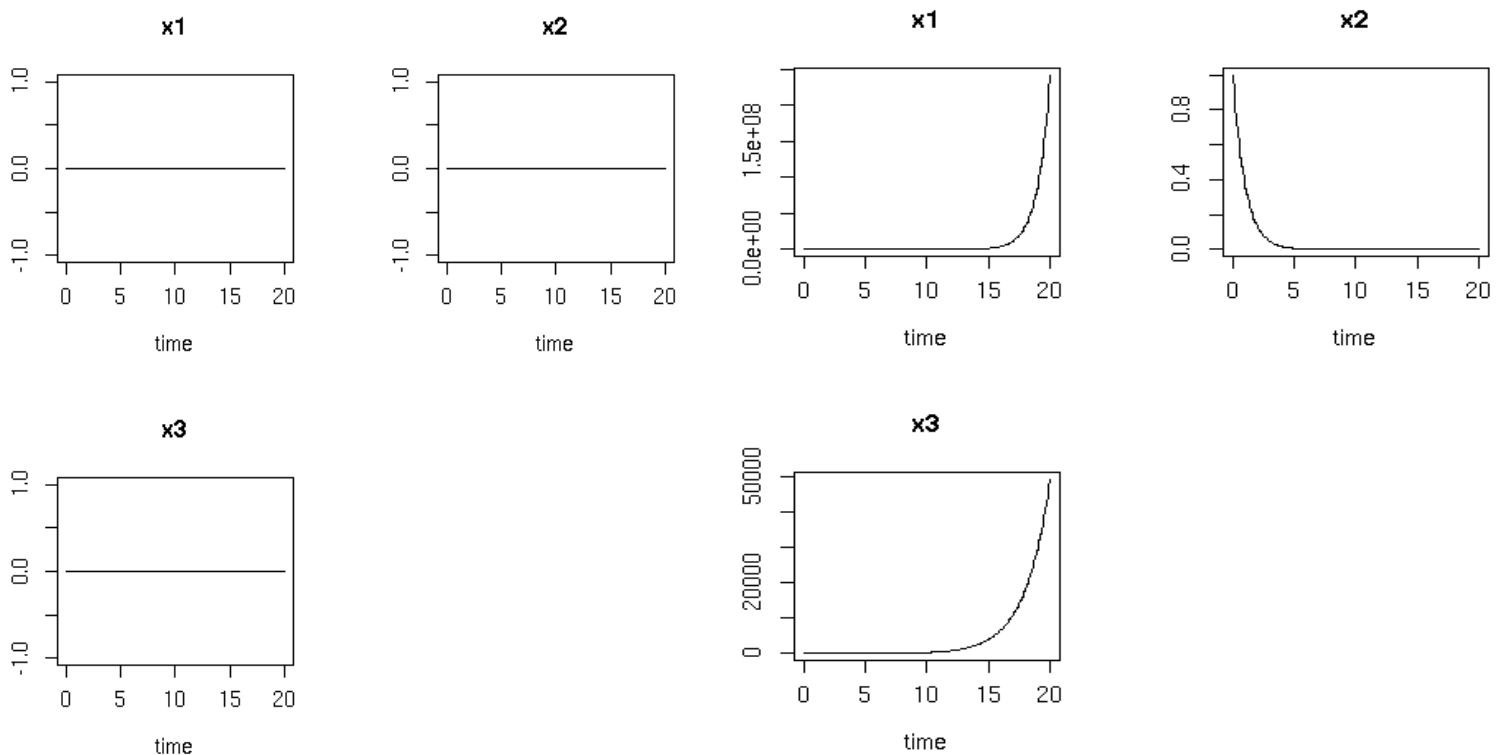


Figure1: when initial conditions $x_1=0, x_2=0$, and $x_3=0$ Figure2: when initial conditions $x_1=1, x_2=1$, and $x_3=1$

Explanation

In figure 1 we observed that for the initial condition all the three functions remained constant (no change w.r.t time). When the initial conditions were changed to 1 (figure2) we observed that the functions x_1 and x_3 increased exponentially but at different rates and the function x_2 decreased exponentially.

Problem-2: Linearize the following system of differential equations:

$$\frac{dX}{dt} = 3Y - 4X^3$$

$$\frac{dY}{dt} = 1.2Ye^X - 2Y^2$$

Solution

$$\frac{\partial f}{\partial x} = -12x^2$$

$$\frac{\partial g}{\partial x} = 1.2ye^x$$

$$\frac{\partial f}{\partial y} = 3$$

$$\frac{\partial g}{\partial y} = 1.2e^x - 4y$$

Problem-3. Consider the system : $\frac{dx}{dt} = x + 4y + xe^x$, $\frac{dy}{dt} = -y - ye^x$

- Check that $(0, 0)$ is an equilibrium point of the system
- Find the general expression for the Jacobian of this system
- Find the Jacobian at the point $(0, 0)$

Solution

$$\frac{\partial f}{\partial x} = x + 4y + xe^x$$

$$\frac{\partial g}{\partial x} = -y - ye^x$$

$$\frac{\partial f}{\partial x} = 1 + xe^x + e^x$$

$$\frac{\partial g}{\partial x} = -ye^x$$

$$\frac{\partial f}{\partial y} = 4$$

$$\frac{\partial g}{\partial y} = -1 - e^x$$

$$\lambda \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 + xe^x + e^x & -ye^x \\ 4 & -1 - e^x \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Putting the value of $x=0$ & $y=0$ in the jacobian matrix \rightarrow

$$\lambda \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & 4 \\ 0 & -2 \end{bmatrix}}_{\text{jacobian matrix}} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\bar{A} = \begin{bmatrix} 2 & 4 \\ 0 & -2 \end{bmatrix}$$

$$|\bar{A} - \lambda I| = 0$$

$$\begin{bmatrix} 2-\lambda & 4 \\ 0 & -2-\lambda \end{bmatrix} = 0$$

$$(2-\lambda)(-2-\lambda) - 0 = 0$$

$$\lambda^2 - 4 = 0$$

$$\lambda = \pm 2$$

$$\lambda_1 = +2 \text{ \& } \lambda_2 = -2$$

as we know,

If $\lambda_1 > 0$, $\lambda_2 < 0$ both real \rightarrow Saddle point, system is unstable

So, the system is unstable.

Problem-4: The famous Lotka-Volterra predator-prey model in non-dimensional form is given by

$$\frac{du}{d\tau} = u(1 - v)$$
$$\frac{dv}{d\tau} = \alpha v(u - 1)$$

where $u(\tau)$ is the non-dimensional prey population, $v(\tau)$ is the non-dimensional predator population and α is a positive rate constant.

(i) Using R, solve this differential equation for $u(\tau) = v(\tau) = 1.3$ and $\alpha = 0.1$. Set $\tau_{end} = 100$. Plot u and v as functions of τ . Then plot $u(\tau)$ versus $v(\tau)$ (phase space trajectory) and see if you get a closed curve in the phase space.

(ii) The system has two sets of fixed points, $(0, 0)$ and $(1, 1)$. Linearize the differential equation about these fixed points and show that the fixed point $(0, 0)$ is a saddle point and the fixed point $(1, 1)$ is a center (purely imaginary eigen values). Use R to do it.

Code

```
# The famous Lotka-Volterra predator-prey model in non-dimensional form is given
# by
# du/dτ= u(1 - v)
# dv/dτ= αv(u - 1)
# where u(τ ) is the non-dimensional prey population, v(τ ) is the non-dimensional predator
#population and α is a positive rate constant.
# (i) Using R, solve this differential equation for u(τ ) = v(τ ) = 1.3 and α = 0.1. Set τ end = 100.
# Plot u and v as functions of τ . Then plot u(τ ) versus v(τ ) (phase space trajectory) and see if
#you get a closed curve in the phase space.
# (ii) The system has two sets of fixed points, (0, 0) and (1, 1). Linearize the differential equation
# about these fixed points and show that the fixed point (0, 0) is a saddle point and the fixed
#point (1, 1) is a center (purely imaginary eigen values). Use R to do it.

library(deSolve)
library(scatterplot3d)

#(i)
lotka_volterra<-function(t,y,params)
{
  #change symbols
  u=y[1]
  v=y[2]

  a=params[1]

  du=u*(1-v)
  dv=a*v*(u-1)

  list(c(du,dv))
}
```

```
#In R, anything that is written lastly is automatically returned in the function  
}
```

```
initial=c(u=1.3,v=1.3)  
parameters=c(0.1)  
time=seq(0,100,0.1)  
out<-ode(times=time,y=initial,func=lotka_voltera,parms=parameters)  
  
plot(out)
```

```
df=as.data.frame(out)  
plot(df$time,df$u,type="l",col="blue",xlab="Time",ylab="")  
lines(df$time,df$v,col="red",type="l")  
legend("topleft",legend=c("u","v"),col=c("blue","red"),lty=1)
```

```
matplot(df$u,df$v,xlab="u",ylab="v")
```

```
 #(ii)
```

```
library(rootSolve)
```

```
#Defining the functions
```

```
f= expression(u*(1-v))
```

```
g=expression(a*v*(u-1))
```

```
#Differentiation of both functions with respect to u and v
```

```
a11=D(f,"u")
```

```
a12=D(g,"u")
```

```
a21=D(f,"v")
```

```
a22=D(g,"v")
```

```
#Define the operating points (1st case)
```

```
u=v=0
```

```
a=0.1
```

```
#Evaluate the elements for the given values of operating points
```

```
a11_n=eval(a11)
```

```
a12_n=eval(a12)
```

```
a21_n=eval(a21)
```

```
a22_n=eval(a22)
```

```
#Creation of a Jacobian matrix
```

```
Jacobian<-matrix(c(a11_n,a12_n,a21_n,a22_n),nrow=2,ncol=2)
```

```
J_eigen<-eigen(Jacobian)
```

```
#Define the operating points (2nd case)
```

```
u=v=1
```

```
a=0.1
```

```

#Evaluate the elements for the given values of operating points
a11_n=eval(a11)
a12_n=eval(a12)
a21_n=eval(a21)
a22_n=eval(a22)

#Creation of a Jacobian matrix
Jacobian<-matrix(c(a11_n,a12_n,a21_n,a22_n),nrow=2,ncol=2)
J_eigen<-eigen(Jacobian)

```

Plots

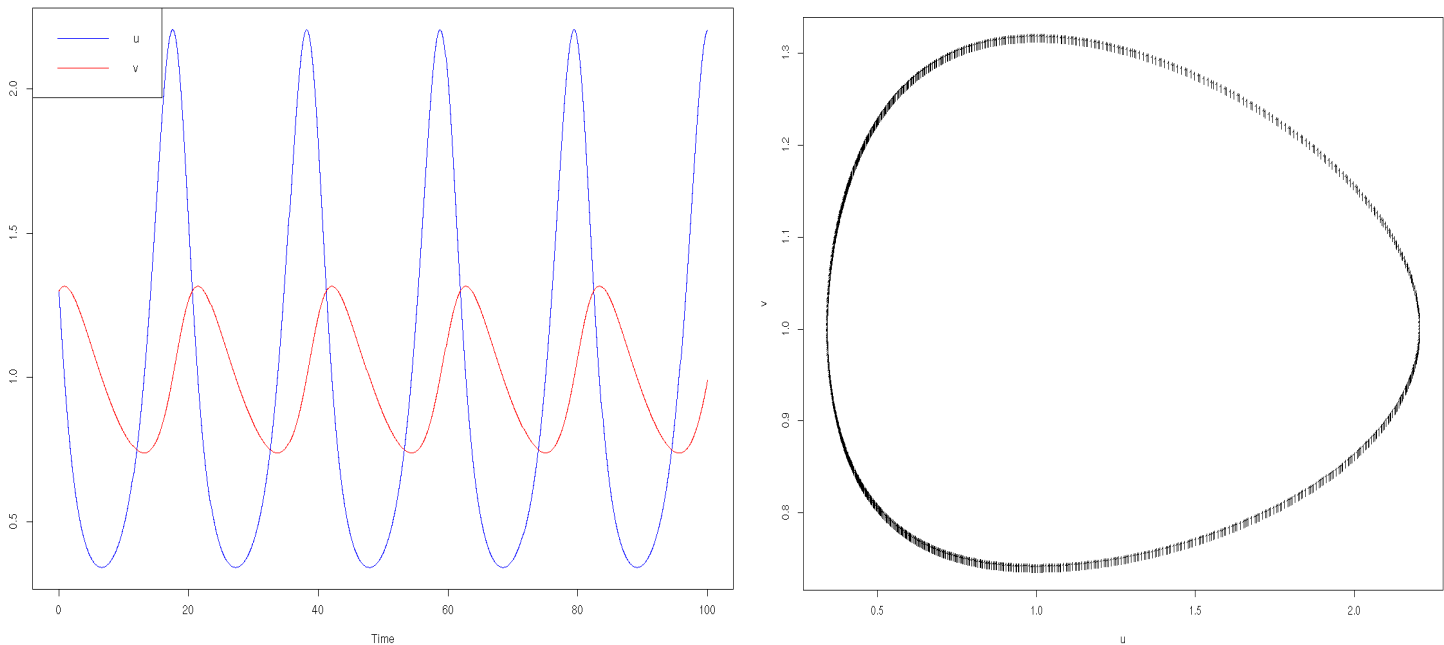


Figure: Lotka-volterra plot (left), phase – space trajectory (right)

Explanation

(i) In the model system, the predators thrive when there are plentiful prey but, ultimately, outstrip their food supply and decline. As the predator population is low, the prey population will increase again. These dynamics continue in a population cycle of growth and decline.

In the model *rate constants of population growth* are always greater than zero, and as such the sign of the eigenvalues above will always differ. Hence the fixed point at the origin is a saddle point.

The instability of this fixed point is of significance. If it were stable, non-zero populations might be attracted towards it, and as such the dynamics of the system might lead towards the extinction of both species for many cases of initial population levels. However, as the fixed point at the origin is a saddle point, and hence unstable, it follows that the extinction of both species is difficult in the mode

(ii) **For the operating points (0,0)**

```
> View(J_eigen)
> J_eigen$vectors
      [,1] [,2]
[1,]   -1    0
[2,]    0   -1
> J_eigen$values
[1]  1.0 -0.1
```

Both eigen values are real , one is < 0 and other is > 0 so the point (0,0) is unstable node ,and is saddle point

For the operating points (1,1)

```
> J_eigen$values
[1] 0+0.3162278i 0-0.3162278i
> J_eigen$vectors
      [,1] [,2]
[1,] -0.9534626+0.0000000i -0.9534626+0.0000000i
[2,]  0.0000000+0.3015113i  0.0000000-0.3015113i
```

Both eigen values are imaginary , so the point (1,1) is a center point

Problem-5: Consider a population in which a predator and its prey co-exist. For prey, the growth rate per capita = 3.5, death rate per capita = 0.5 and the rate at which a prey is eaten by predator is 1. The growth and death rate per capita of the predator is 1 and 3.2 respectively. The equations for this simple predator and prey model are as follows:

1

$$\frac{d[prey]}{dt} = (growth\ rate\ for\ prey)(prey) - (death\ rate\ for\ prey)(prey) - (rate\ at\ which\ prey\ eaten\ by\ predator)(prey)(predator)$$

$$\frac{d[predator]}{dt} = (growth\ rate\ for\ predator)(predator)(prey) - (death\ rate\ of\ predator)(predator)$$

Let the initial number of prey is 15 and predator is 7.

Solve this system of equations in R to sketch the number of predator and prey as function of time.

(Introduce your own symbols for the quantities in the equations)

Code

```
# Consider a population in which a predator and its prey co-exist. For prey, the
# growth rate per capita = 3.5, death rate per capita = 0.5 and the rate at which a prey is eaten
# by
# predator is 1. The growth and death rate per capita of the predator is 1 and 3.2 respectively.
# The
# equations for this simple predator and prey model are as follows:
# d[prey]/dt
# = (growth rate for prey)(prey) - (death rate for prey)(prey) - (rate at which prey eaten by
# predator)(prey)(predator)
# d[predator]/dt
# = (growth rate for predator)(predator)(prey) - (death rate of predator)(predator)
# Let the initial number of prey be 15 and predator is 7.
# Solve this system of equations in R to sketch the number of predator and prey as function of
# time.
# ( Introduce your own symbols for the quantities in the equations)
```

```
library(deSolve)
library(scatterplot3d)
```

```
growth_rate<-function(t,y,params)
{
  #change symbols
  gpred=params[1] #growth rate of predator
  dpred=params[2] #death rate of predator
  gprey=params[3] #growth rate of prey
  dprey=params[4] #death rate of prey
```

```
rate=params[5] #rate at which prey is eaten by predator
```

```
predator=y[1]
```

```
prey=y[2]
```

```
dpredator=(gpred*predator*prey)-(dpred*predator)
```

```
dprey=(gprey*prey)-(dprey*prey)-(rate*prey*predator)
```

```
list(c(dpredator,dprey))
```

```
#In R, anything that is written lastly is automatically returned in the function  
}
```

```
initial=c(predator=15,prey=7)
```

```
parameters=c(1,3.2,3.5,0.5,1)
```

```
time=seq(0,20,0.01)
```

```
out<-ode(times=time,y=initial,func=growth_rate,parms=parameters)
```

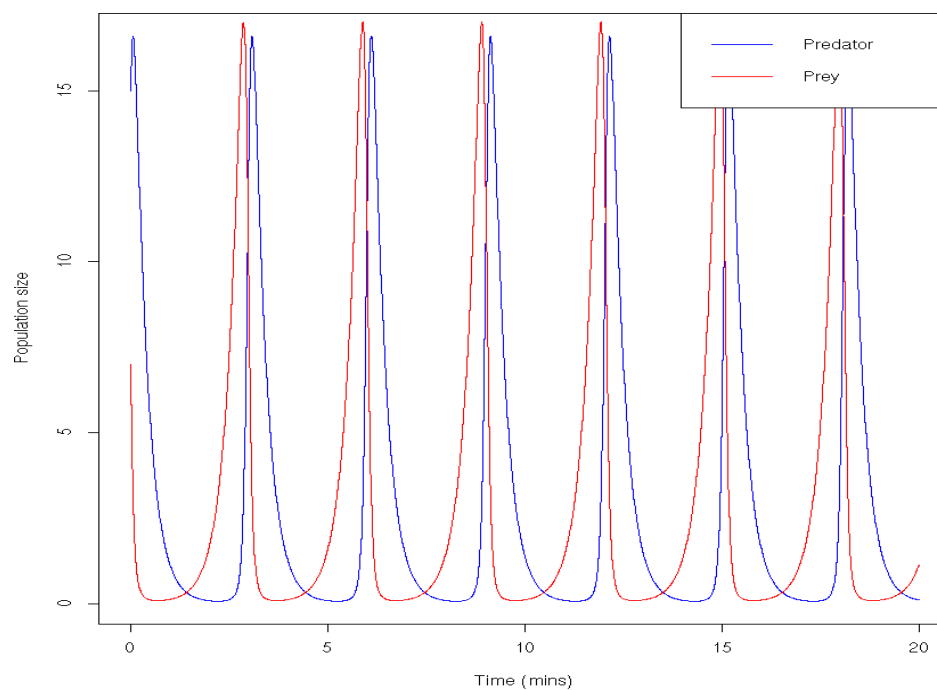
```
df=as.data.frame(out)
```

```
plot(df$time,df$predator,type="l",col="blue",xlab="Time (mins)",ylab="Population size")
```

```
lines(df$time,df$prey,col="red")
```

```
legend("topright",legend=c("Predator","Prey"),col=c("blue","red"),lty=1)
```

Plot



Explanation

Lotka Volterra Model is a fundamental model of Complex Ecology of the world that assumes just one prey for predator and vice-versa . The model signifies the equilibrium between the population of predator and prey. As we can see the plots whenever the population of predators is high the population of the prey decreases and vice - versa.

Problem-6 Consider the logistic model for the density dependent growth:

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

where, $N(t)$ is population as a function of time, r and K are parameters of the model.

The exact solution to the above logistic equation is given by,

$$N(t) = \frac{N(0)e^{rt}}{1 + \frac{N(0)}{K}(e^{rt} - 1)}$$

(b) For the values $r = 0.0312$ and $K = 198.6$, compute $N(t)$ as a function of t using above solution formula. Plot this curve on the same plot along with data points. Upto which year, the fit is good? Is the logistic equation a good model for this population growth data?. Comment.

Code

```
# Consider the logistic model for the density dependent growth:
# dN/dt= r*N*(1 - (N/K))
# where, N(t) is population as a function of time, r and K are parameters of the model.
# The exact solution to the above logistic equation is given by,
# Nt=(No*exp(r*t))/(1+((No/k)*(exp(r*t)-1)))
# (a) Plot time versus population in a graph
# (b) For the values r = 0.0312 and K = 198.6, compute N(t) as a function of t using above
# solution formula. Plot this curve on the same plot along with data points. Upto which year, the
# fit
# is good? Is the logistic equation a good model for this population growth data?. Comment.

setwd("/home/ibab/Akansha_Msc11/Semester_4/System_Biology/Problem_set2")

#Reading the data into dataframe
df=read.table("Population_data_q6.txt",header=T)

plot(df$t,df$population,type="l",xlab="Time",ylab="Population size (millions)")

library(deSolve)
library(scatterplot3d)
```

```

population_model<-function(t,y,params)
{
  #change symbols
  r=params[1]
  K=params[2]

  N=y

  dN= r*N*(1-(N/K))

  list(dN)

  #In R, anything that is written lastly is automatically returned in the function
}

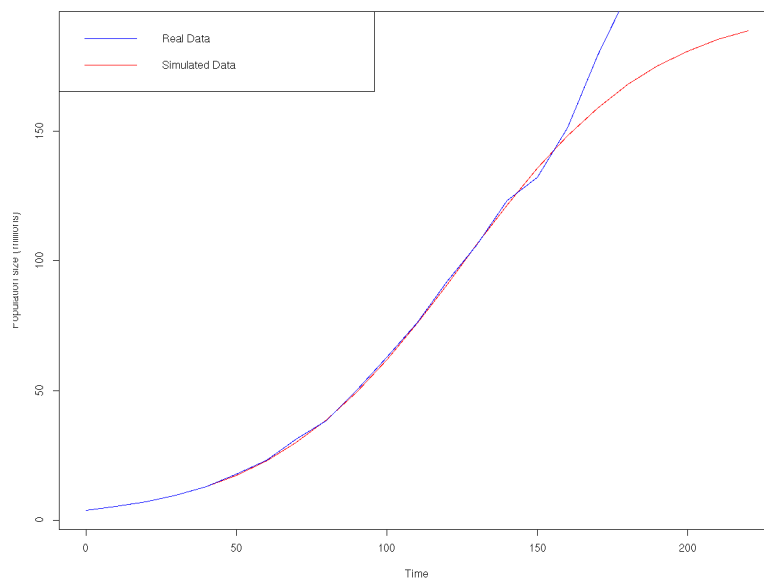
initial=c(N=3.9)
parameters=c(0.0312,198.6)
time=seq(0,220,10)
out<-ode(times=time,y=initial,func=population_model,parms=parameters)

plot(out)

df2=as.data.frame(out)
plot(df2$time,df2$N,type="l",col="red",xlab="Time",ylab="Population size (millions)")
lines(df$time,df$population,col="blue")
legend("topleft",legend=c("Real Data","Simulated Data"),col=c("blue","red"),lty=1)

```

Plot



Explanation

From the above plot it can be inferred that the fit is good upto year 1950.

Logistic equation is a good model to give population estimate upto a certain year as after that the model saturates whereas the real data grow exponentially

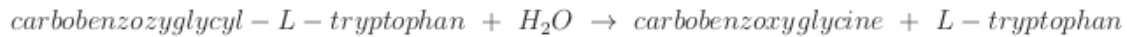
Problem-7 In a Michaelis -Menten process, the reaction rates 'V' were measured as a function of substrate concentration 'S'. The data is presented below:

S : 0.5, 1.0, 1.5, 2.0, 4, 6, 8, 10, 15, 20, 25, 30, 40, 50, 75, 100

V : 0.16, 0.30, 0.42, 0.52, 0.85, 1.06, 1.22, 1.34, 1.54, 1.67, 1.75, 1.81, 1.90, 1.95, 2.03, 2.07

in R, plot 'S' versus 'V' on a linear scale and get read the approximate values of V_{max} and K_m from the graph. Print them.

The hydrolysis of *carbobenzozyglycyl-L-tryptophan* catalyzed by pancreatic *carboxypeptidase* occurs according to the reaction:



The following data on the rate of formation of *L-tryptophan* at 25°C and pH 7.5 was obtained:

Substrate Concentration S (mM)	Rate V (mM/second)
2.5	0.024
5	0.036
10	0.053
15	0.060
20	0.064

(a) Plot the curve of concentration S versus rate V. Assuming that this curve follows the Michaelis-Menton kinetics of the form $V = \frac{V_{max}S}{K_m + S}$, read the approximate values of V_{max} and K_m from the curve.

(b) Michaelis-Menton equation can be linearized with the relation given by,

$$\frac{1}{V} = \frac{K_m}{V_{max}} \frac{1}{S} + \frac{1}{V_{max}}$$

Make an appropriate plot of the above relationship to get the values of V_{max} and K_m .

Code

```
#Michaelis-Menten Equations
```

```
#a)
```

```
S=c(0.5, 1.0, 1.5, 2.0, 4, 6, 8, 10, 15, 20, 25, 30, 40, 50, 75, 100)
```

```
V=c(0.16, 0.30, 0.42, 0.52, 0.85, 1.06, 1.22, 1.34, 1.54, 1.67, 1.75, 1.81, 1.90, 1.95, 2.03, 2.07)
```

```
plot(S,V,type="l",xlab="Substrate Concentration",ylab="Velocity")
```

```
print("From the plot Vmax is approximately 2 units and Km is 6")
```

```
#b)
```

```
S=c(2.5,5,10,15,20)
```

```
V=c(0.024,0.036,0.053,0.060,0.064)
```

```
plot(S,V,type="l",xlab="Substrate Concentration",ylab="Velocity")
```

```
print("From the plot Vmax is approximately 0.06mM/sec and Km is 5mM")
```

```
#c)
```

```
S=1/S
```

```
V=1/V
```

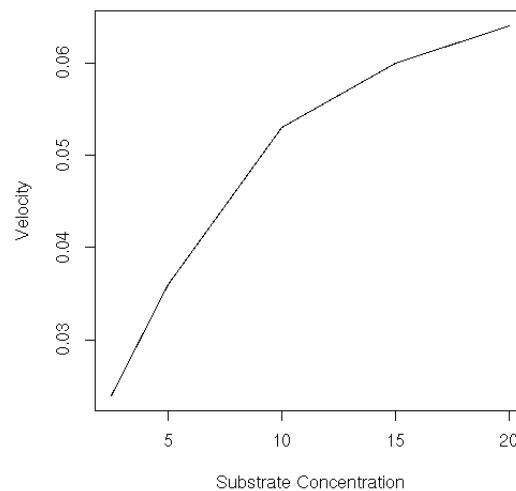
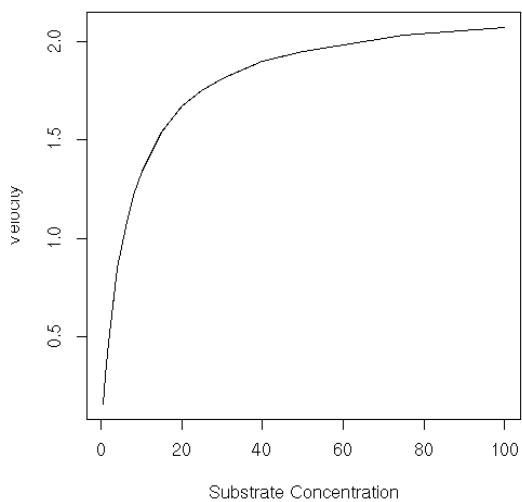
```
plot(S,V,type="l",xlab="Substrate Concentration",ylab="Velocity")
```

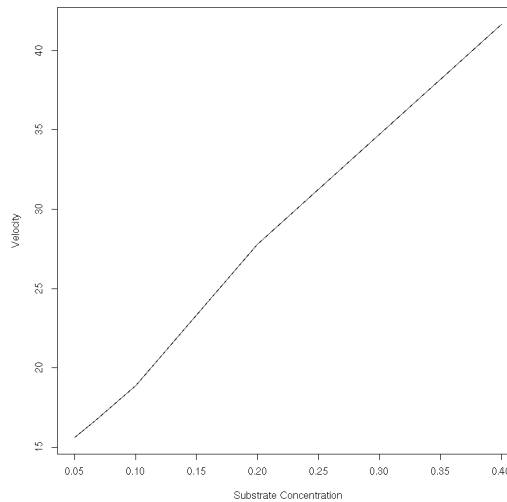
```
df=data.frame(S,V)
```

```
model<-lm(formula = V ~ S,data = df)
```

```
summary(model)
```

Plots





Explanation

For plots 1 and 2 we could calculate K_m and V_{max} directly from the plots and values are printed in the code itself.

For plot 3 we did linear model fitting of V (dependent variable) with S (independent variable). The intercept ($1/V_{max}$) = 11.80 and slope (K_m/V_{max}) = 75.43, therefore V_{max} = 0.084 mM/sec while K_m = 6.41mM.

Problem-8 The equations for the rate of change of ATP and glucose in the cell are as follows:

$$\frac{d[ATP]}{dt} = 2K_1[G][ATP] - \frac{k_p[ATP]}{[ATP] + k_m}$$

$$\frac{dG}{dt} = V_{in} - K_1[G][ATP]$$

The parameter values for the model are the following:

$$V_{in} = 0.36, \quad k_1 = 0.02, \quad K_p = 6 \text{ and } K_m = 13.$$

The concentrations of glucose and ATP were 6.6 and 7.6 units, respectively.

- Write a R code to simulate the system behaviour till 1000 mins of the reaction.
- Show the time profiles of Glucose and ATP concentrations in the same plot. Interpret the results.
- Change the value of K_m to 20. Run the simulation. What difference do you observe between both the systems?

Code

```
# The equations for the rate of change of ATP and glucose in the cell are as follows:
# d[ATP]/dt=2*K1*[G]*[ATP]-((Kp*[ATP])/([ATP]+Km))
# d[G]/dt = Vin - K1*[G]*[ATP]
# The parameter values for the model are the following:
```

```
# V in = 0.36, k 1 = 0.02, K p = 6 and K m = 13.
# The concentrations of glucose and ATP were 6.6 and 7.6 units, respectively.
# (a) Write a R code to simulate the system behaviour till 1000 mins of the reaction.
# (b) Show the time profiles of Glucose and ATP concentrations in the same plot. Interpret
# the results.
# (c) Change the value of Km to 20. Run the simulation. What difference do you observe
# between both the systems?
```

```
library(deSolve)
#library(scatterplot3d)
```

```
Glu_ATP<-function(t,y,params)
```

```
{
  #change symbols
  Vin=params[1]
  k1=params[2]
  kp=params[3]
  km=params[4]
```

```
  G= y[1]
  ATP = y[2]
```

```
  dG = Vin -(k1*G*ATP)
  dATP=(2*k1*G*ATP)-((kp*ATP)/(ATP+km))
```

```
  list(c(dG,dATP))
```

```
  #In R, anything that is written lastly is automatically returned in the function
}
```

```
# CASE 1 --- Km = 13
initial=c(G=6.6,ATP=7.6)
parameters=c(0.36,0.02,6,13)
time=seq(0,1000)
out<-ode(times=time,y=initial,func=Glu_ATP,parms=parameters)
```

```
plot(out[,1],out[,2],col="blue",type="l",xlab="time",ylab="Concentration")
lines(out[,1],out[,3],col="red")
legend(x="topright",legend=c("Glucose","ATP"),col=c("blue","red"),lty=1,xpd=TRUE)
```

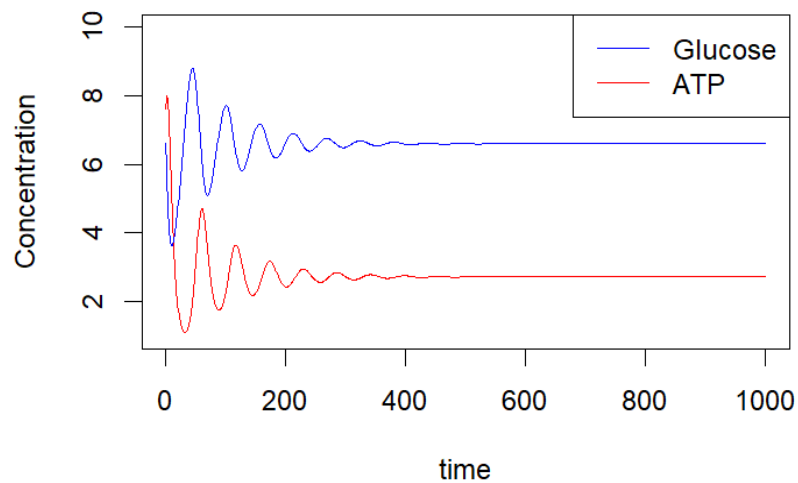
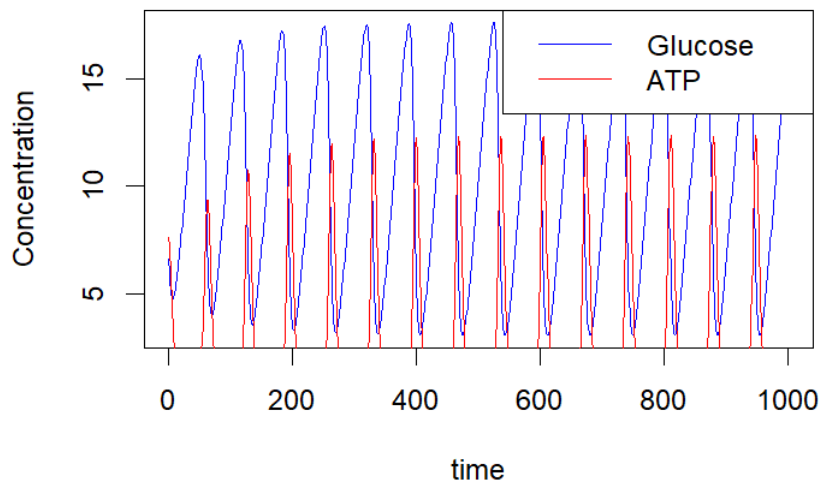
```
# CASE 2 --- Km = 20
initial=c(G=6.6,ATP=7.6)
parameters=c(0.36,0.02,6,20)
time=seq(0,1000)
out2<-ode(times=time,y=initial,func=Glu_ATP,parms=parameters)
```

```
plot(out2[,1],out2[,3],col="red",type="l",xlab="time",ylab="Concentration",ylim=c(1,10))
```



```
lines(out2[,1],out2[,2],col="blue")
legend(x="topright",legend=c("Glucose","ATP"),col=c("blue","red"),lty=1,xpd=TRUE)
```

Plots



Explanation

From both the graphs, we can infer that as the glucose concentration is increasing, the ATP concentration is decreasing (as ATP is required for the production of glucose inside a cell). When the K_m was changed to 20, it can be seen that the steady state was achieved much slowly as compared to the first case.

Problem-9 . A population dynamics model with predation can be written in a dimensionless form as

$$\frac{dx}{d\tau} = \alpha x \left(1 - \frac{x}{K}\right) - \frac{x}{1+x} = f(x)$$

where α is the growth rate and K is the carrying capacity. Let us choose $K = 20$.

(a) For $\alpha = 1.3$, $\alpha = 0.3$ and $\alpha = 0.1$, plot $f(x)$ vs x using R. Geometrically identify the fixed points and comment on their stability.

(b) For $\alpha = 0.3$, numerically solve the above differential equation in R for the initial conditions $x(\tau = 0) = 1.2$.

Code

```
# A population dynamics model with predation can be written in a dimensionless
# form as
# dx = (a*x*(1 - (x/k))) - (x/(1+x))
#
# where a is the growth rate and K is the carrying capacity. Let us choose K = 20.
# (a) For  $\alpha = 1.3$ ,  $\alpha = 0.3$  and  $\alpha = 0.1$ , plot  $f(x)$  vs  $x$  using R. Geometrically identify the fixed
# points and comment on their stability.
# (b) For  $\alpha = 0.3$ , numerically solve the above differential equation in R for the initial conditions
#  $x(\tau = 0) = 1.2$ .

library(deSolve)
#library(scatterplot3d)

# PART-A
popu_growth<-function(t,y,params)
{
  a=params[1]
  k=params[2]
  x = y[1]

  dx = (a*x*(1 - (x/k))) - (x/(1+x))

  list(c(dx))

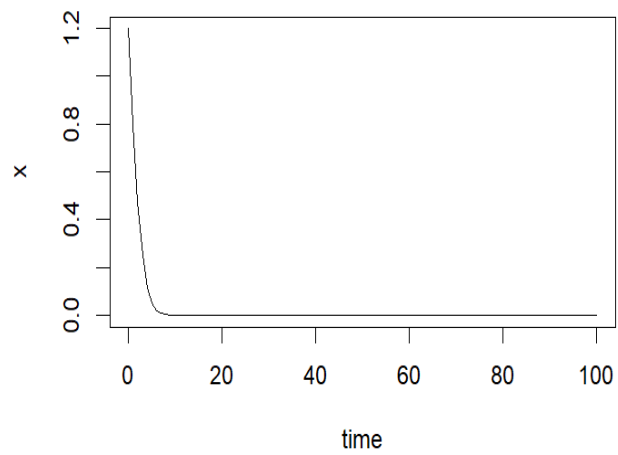
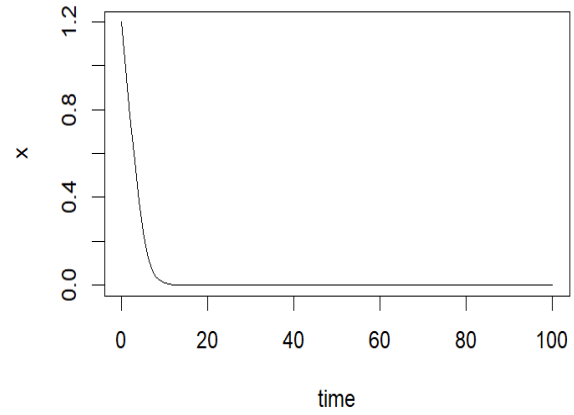
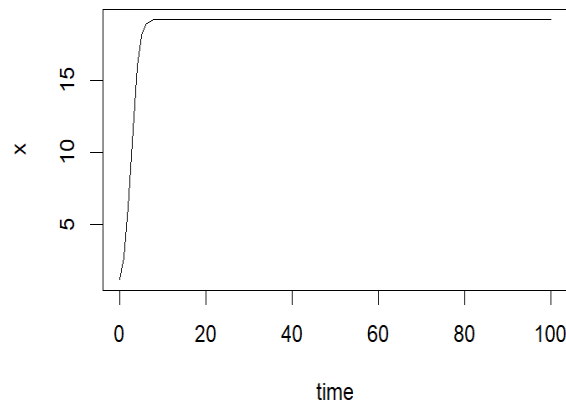
  #In R, anything that is written lastly is automatically returned in the function
}
alpha_val = c(1.3,0.3,0.1)
k=20

initial=c(x=1.2)

time=seq(0,100)
for(alpha in alpha_val)
{
  parameters = c(alpha,k)
```

```
out<-ode(times=time,y=initial,func=popu_growth,parms=parameters)
plot(out[,1],out[,2],type="l",xlab="time",ylab="x")
}
```

Plots



Problem-10 An infectious disease model consists of the following variables:

$S(t)$ = number of susceptible individuals at time t

$I(t)$ = number of infectious individuals at time t

$R(t)$ = number of individuals died at time t

with $N(t) = S(t) + I(t) + R(t)$ = total number of individuals = constant

This model (called 'SIR model') is represented by a set of ordinary differential equations,

$$\frac{dS}{dt} = \frac{-bSI}{N}$$

$$\frac{dI}{dt} = \left(\frac{bSI}{N} - cI \right)$$

$$\frac{dR}{dt} = cI$$

with the condition that $\frac{dN}{dt} = \frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt}$

Solve the above equation in R for $b=2$, $c=0.2$ with the initial conditions :

$S(0) = 99$, $I(0) = 1$ and $R(0) = 0$

Plot the three variables S , I and R as a function of time on the same graph. Interpret the results.

Code

```
# An infectious disease model consists of the following variables:
# S(t) = number of susceptible individuals at time t
# I(t) = number of infectious individuals at time t
# R(t) = number of individuals died at time t
# with N(t) = S(t) + I(t) + R(t) = total number of individuals = constant
# This model (called 'SIR model') is represented by a set of ordinary differential equations,
# dS/dt = (-b*S*I)/N
# dI/dt = (b*S*I)/N - (c*I)
# dR/dt = c*I
# Solve the above equation in R for b= 2, c=0.2 with the initial conditions :
# S(0) = 99, I(0) = 1 and R(0) = 0
# with the condition that
# Plot the three variables S, I and R as a function of time on the same graph. Interpret the
results.
```

```
library(deSolve)
library(scatterplot3d)
```

```
SIR_model<-function(t,y,params)
{
  #change symbols
  b=params[1]
  c=params[2]
```

```

S = y[1]
I = y[2]
R = y[3]
N = (S+I+R)

```

```

dS = (-b*S*I)/N
dI = (b*S*I)/N - (c*I)
dR = c*I

```

```
list(c(dS,dI,dR))
```

```

#In R, anything that is written lastly is automatically returned in the function
}

```

```

initial=c(S=99,I=1,R=0)
parameters=c(2,0.2)
time=seq(0,20,0.1)
out<-ode(times=time,y=initial,func=SIR_model,parms=parameters)

```

```
plot(out)
```

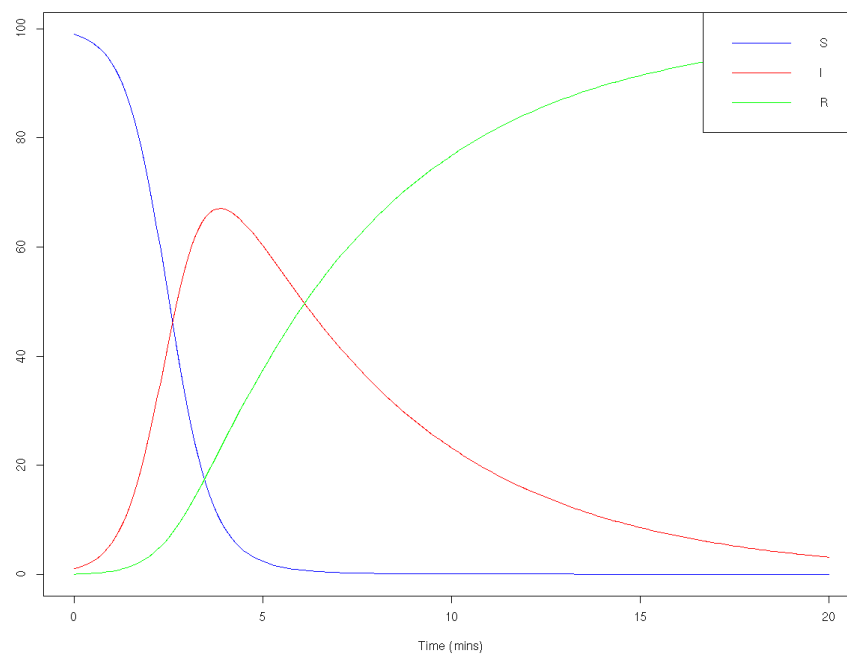
```
scatterplot3d(out[,-1])
```

```

df=as.data.frame(out)
plot(df$time,df$S,type="l",col="blue",xlab="Time (mins)",ylab="")
lines(df$time,df$I,col="red")
lines(df$time,df$R,col="green")
legend(x="topright",legend=c("S","I","R"),col=c("blue","red","green"),lty=1,xpd=TRUE)

```

Plot



Explanation

SIR model is a mathematical model for infectious diseases which comprises 3 components : S - Susceptible individuals, I - Infected individuals and R - Recovered individuals. From the plots, we can infer that initially during the spread of an epidemic, the number of susceptible individuals is high because the SIR model assumes every individual to be susceptible. This leads to an increase of infected individuals after a brief incubation period. But gradually as the disease progresses, the recovery rate increases resulting in a smaller number of infected individuals.

Problem-4: The bacterial growth data representing size of the colony as a function of time is given below:

$t = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0\}$
 $S = \{2.2, 2.9, 3.2, 6, 6, 6.2, 8, 11, 14, 13, 18, 22, 21, 30, 35, 38, 47, 58, 70, 81, 90\}$

Write an R script to fit a non-linear curve of the form $S = At^\alpha$ to this data. For this, perform log transform on both sides of the equation and do a linear fit between $\log(t)$ and $\log(S)$ to get the parameters t and α .

Code

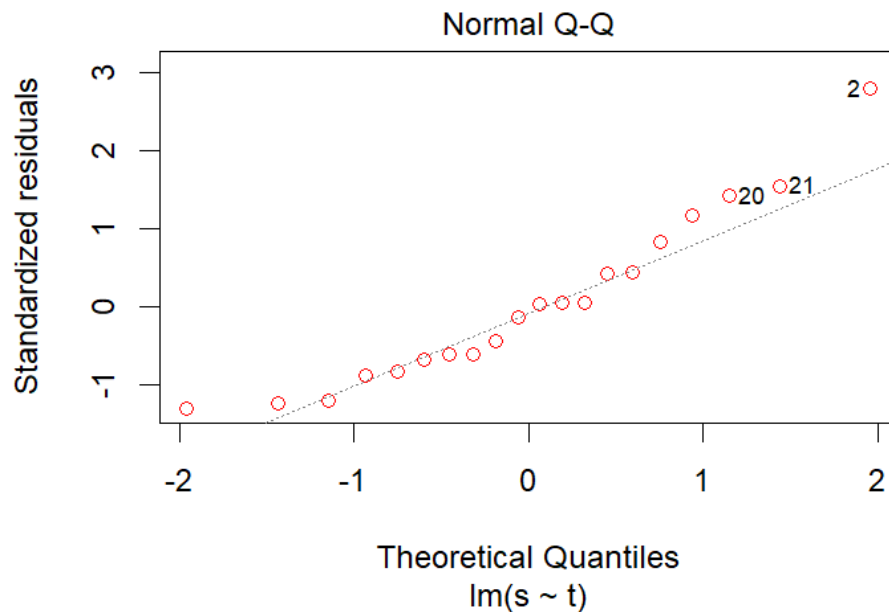
```
# The bacterial growth data representing size of the colony as a function of time is
# given below:
#
# t = {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0}
# S = {2.2, 2.9, 3.2, 6, 6, 6.2, 8, 11, 14, 13, 18, 22, 21, 30, 35, 38, 47, 58, 70, 81, 90}
# Write an R script to fit a non-linear curve of the form S = At $\alpha$  to this data. For this, perform
# log transform on both sides of the equation and do a linear fit between log(t) and log(S) to get
# the parameters t and  $\alpha$ .

# defining the vectors
t = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0)
s = c(2.2, 2.9, 3.2, 6, 6, 6.2, 8, 11, 14, 13, 18, 22, 21, 30, 35, 38, 47, 58, 70, 81, 90)

# creation of data frames with log transformed values
df = data.frame(log(t), log(s))
df2 = subset(df[-1,])
names(df2)=c("t","s")

# performing linear model fitting
model = lm(formula=s ~ t, data=df2)
plot(model,col="red")
# to visualise the linear fitting, observe PLOT no-2
```

Plots



Explanation

For equation $S=A*(t^{\alpha})$, when we take natural log on both sides, the equation becomes $\ln(S) = \ln(A) + \alpha*\ln(t)$. So the summary statistics of the linear fitting is →

```
Call:
lm(formula = s ~ t, data = df2)

Residuals:
    Min       1Q   Median       3Q      Max
-0.45139 -0.25106 -0.01895  0.17847  0.76701

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.12792    0.08186   38.21  < 2e-16 ***
t            1.22915    0.10061   12.22 3.78e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3564 on 18 degrees of freedom
Multiple R-squared:  0.8924,    Adjusted R-squared:  0.8864 
F-statistic: 149.2 on 1 and 18 DF,  p-value: 3.781e-10
```

Value of $A = \exp(3.12792) = 22.82645$

Value of $\alpha = 1.22915$