Exercise 2.3: Ricker model

William Edwin (Bill) Ricker in the 50’s put forward the following model for describing a population of salmons:

$$ x_{t+1} = \exp \left[ r \left( 1 - \frac{x_t}{K} \right) \right] x_t $$

(1)

where \( r \) is an intrinsic growth rate and \( K \) is the carrying capacity of the environment. The exponential term takes into account the role of predators, the overcrowded conditions, etc.

**Exercise:** Find the fixed points and show that the route to chaos is the same as the one of the logistic map.

**Solution:** Contrary to the logistic map, the iterates of the Ricker’s map are always positive for every \( x_t \). The point \( x_t \) is a fixed point if \( x_t = x_{t+1} \), then the fixed points are \( x^* = 0 \) and \( x^* = K \). The derivative of the Ricker equation (1) is:

$$ m = \frac{df(x_t)}{dx_t} = \exp \left[ r \left( 1 - \frac{x_t}{K} \right) \right] \times \left[ 1 - \frac{r x_t}{K} \right] $$

Then:

If \( x^* = 0 \), it results \( m = \exp(r) \), greater than 1, so the fixed point is unstable.

If \( x^* = K \), it results \( m = 1 - r \). Now, however, one has to see whether it is \( 0 < r < 2 \), in this case it is \( |1 - r| < 1 \), so the fixed point is globally and asymptotically stable. Otherwise, if it is \( r > 2 \), the fixed point is unstable.

The structure of the code is the same as that of the logistic map, with the function

```r
f.x<- function(x,r){exp(r*(1-(x/K)))*x}
```

#Ricker model: cobweb plot.
f.x<- function(x,r){exp(r*(1-(x/200)))*x}

##### function to draw the time plot ######
f.temp<-function(xinit,nstep,r){ # starting function f.temp
  xt<- numeric()
x<- xinit
  xt[1]<- x
  for(i in 2:nstep){
...
y <- f.x(x, r)
x <- y
xt[i] <- x
}
plot(xt, type="b", xlab="time", ylab="x(t)",
     cex.lab=1.7, cex.axis=1.3, lwd=2)
# xt # comment to skip iterates
#
##### function to draw the cobweb plot ####
iter <- function(xinit, nstep, r){
    x <- xinit
    y <- f.x(x, r)
    segments(x, 0, x, y, lty=1, lwd=2)
    for(i in 1:nstep){
        points(x, y, pch=19, cex=1.5)
        segments(x, y, y, y, lty=1, lwd=2)
        x <- y
        y <- f.x(x, r)
        segments(x, x, x, y, lty=1, lwd=2)
    }
}
# ending function iter
### parameters and initial conditions
r <- 1.9
K <- 200
xinit <- 2
nstep <- 18
f.temp(xinit, nstep, r) # call up the time plot
### preparation of the cobweb plot
windows()
plot(0, 0, type="n", xlim=c(0, 1000), ylim=c(0, 260), xlab="x(t)",
     ylab="x(t+1)",
     cex.lab=1.5, cex.axis=1.2)
# plot of the function f.x
curve(f.x(x, r), from = 0, to = 1000, lty=5, col="blue", lwd=2, add=T)
segments(0, 0, 1000, 1000, lty=3, lwd=2, col="magenta") # bisector
iter(xinit, nstep, r) # call up the cobweb plot
segments(200, 0, 260, 200, lty=4, lwd=2, col="red")

Figure 1 (left) shows the oscillatory convergence to the fixed point \(x^* = K\).
When the number of fishes reaches the carrying capacity, the system is in the stationary state. Indeed, we see that the dot-dash line \(x_t = K\) and the dotted line \(x_{t+1} = x_t\) intersect the curve of equation (1) on the period-1 attractor \(x^* = K\).

Let us increase \(r\). We know what will happen. At \(r = 2\) the period-1 attractor becomes unstable and generates a period-2 attractor, and so on, up to the chaotic region as it is shown in Fig. 1 (right).
Fig. 1 Ricker map (1) with $K = 200$: cobweb plot, $r = 1.9$ (left), bifurcation diagram (right).

The code is as Code 2.4 Nonlinear logistic map: bifurcation diagram, with $f.x<- function(x,r)\exp(r*(1-(x/200)))*x$

# Ricker map: bifurcation diagram

f.x<- function(x,r){exp(r*(1-(x/200)))*x}
ntrans<- 1000 # transient
rin<- 1.6
rfin<- 3.6
n<- 400 # number of iterations after the transient
nt< ntrans+n # total number of iterations
nr< 300 # number of r step
xinit<- 0.2
r< seq(rin,rfin,length=nr)
plot(0,0,type="n",xlim=c(rin,rfin),ylim=c(0,750),xlab="r",ylab="x(t)",
cex.lab=1.5,cex.axis=1.2)
for(i in r) { # starting loop on r values
  x<- xinit
  for(j in 1:nt) { # starting loop on the iterations
    y<- f.x(x,i)
    if(j > ntrans) points(i,y,pch='.',cex=3)
    x<- y
  } # ending loop on the iterations
} # ending loop on r values