Exercises1

Yongqi WANG, Hangjia ZHAO

Problem 1

a)

The model reaches equilibrium points as the $x_{t+1} = f(x_t) = x_t$, let x^* be the equilibrium points of the system. Solving $f(x) = rx(1-x) = x$, we have: $x^* = 0$ or $x^* = 1 - \frac{1}{r}$ when $r \neq 0$.

b)

The stability of the points can be checked by the gradient of $|f'(x^*)| = |r - 2rx^*|$.

When $r = 0.5$, $x_1^* = 0, |f'(x_1^*)| = 0.5 < 1$ and hence **attractive and stable**.

$x_2^* = -1$, discarded since x can only take value larger or equal to zero.

When $r = 1.5$, $x_1^* = 0, |f'(x_1^*)| = 1.5 > 1$ and hence **repelling and unstable**.

$x_2^* = \frac{1}{3}, |f'(x_2^*)| = 0.5 < 1$ and hence **attractive and stable**.

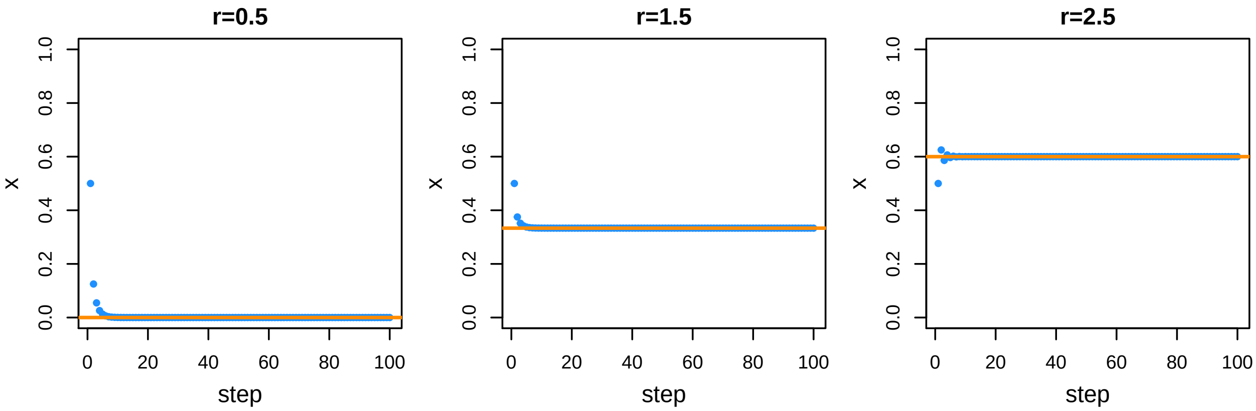
When $r = 2.5$, $x_1^* = 0, |f'(x_1^*)| = 2.5 > 1$ and hence **repelling and unstable**.

$x_2^* = \frac{3}{5}, |f'(x_2^*)| = 0.5 < 1$ and hence **attractive and stable**.

c)

```r
getRep <- function(init, steps, r) {
  x <- rep(0, steps)
  x[1] <- init
  for (i in 2:steps) {
    x[i] <- r * x[i - 1] * (1 - x[i - 1])
  }
  return(x)
}
steps = 100
x1 = getRep(0.5, steps, 0.5)
x2 = getRep(0.5, steps, 1.5)
x3 = getRep(0.5, steps, 2.5)
par(mar = c(4, 4, 2, 0.5)) # margin size
par(mgp = c(2.5, 1, 0)) # axis localtion
par(cex.lab = 1.25) # size of y axis label
par(mfrow=c(1,3)) # 1x3 fig
plot(x1, xlab = "step", ylab = "x", main = "r=0.5", col = "dodgerblue",
     pch = 20, ylim = c(0, 1))
abline(h = 0, col = "#ff8c00", lwd = 2)
plot(x2, xlab = "step", ylab = "x", main = "r=1.5", col = "dodgerblue",
     pch = 20, ylim = c(0, 1))
abline(h = 1/3, col = "#ff8c00", lwd = 2)
plot(x3, xlab = "step", ylab = "x", main = "r=2.5", col = "dodgerblue",
```

```
     pch = 20, ylim = c(0, 1))
abline(h = 3/5, col = "#ff8c00", lwd = 2)
```



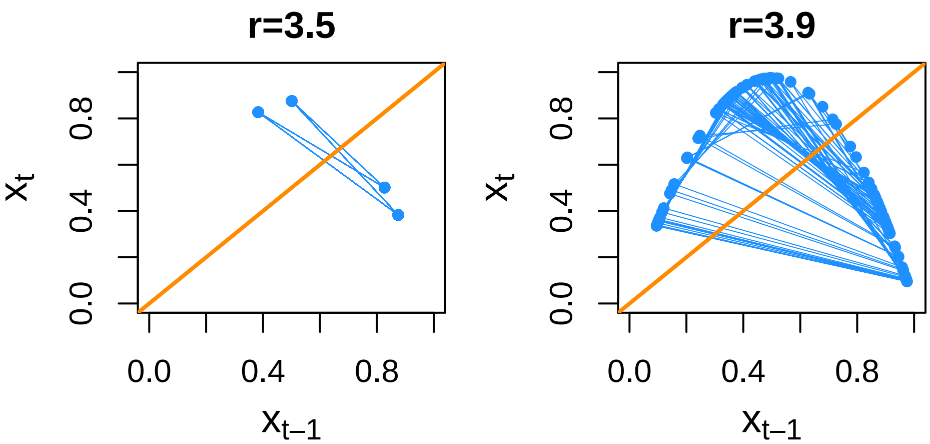
In the figures, result of numerically iterating the difference equation when r = 0.5, 1.5, 2.5. The yellow line represents the predicted stable equilibrium point. Initial values were set to 0.5.

d)

e)

As seen in the figures below, both paths can not be traced back to a stable fixed point and huge oscillation is observed in the right sub-plot where $r = 3.9$.

```
x = getRep(0.5, 100, 3.5)
xstm1 <- x[-length(x)]
xst <- x[-1]
y = getRep(0.5, 100, 3.9)
ystm1 <- y[-length(y)]
yst <- y[-1]
par(mar = c(4, 4, 2, 0.5)) # margin size
par(mgp = c(2.5, 1, 0)) # axis localtion
par(cex.lab = 1.25) # size of y axis label
par(mfrow=c(1,2)) # 1x2
plot(xstm1, xst, xlab = expression(x[t - 1]), ylab = expression(x[t]),
     main = "r=3.5", col = "dodgerblue", pch = 20, xlim = c(0, 1),
     ylim = c(0,1))
lines(xstm1, xst, col = "dodgerblue", lwd = 0.5)
abline(b = 1, a = 0, col = "#ff8c00", lwd = 2)
plot(ystm1, yst, xlab = expression(x[t - 1]), ylab = expression(x[t]),
     main = "r=3.9", col = "dodgerblue", pch = 20, xlim = c(0, 1),
     ylim = c(0,1))
lines(ystm1, yst, col = "dodgerblue", lwd = 0.5)
abline(b = 1, a = 0, col = "#ff8c00", lwd = 2)
```



Problem 2

a)

$$
\begin{aligned}
\int \frac{dx(t)}{x(t)(1-\frac{x(t)}{K})} &= \int \lambda dt \quad \text{separation of variables} \\
\int \frac{dx(t)}{x(t)} + \int \frac{dx(t)}{K-x(t)} &= \int \lambda dt \\
\ln|x(t)| - \ln|K-x(t)| &= \lambda t + C \\
\ln|\frac{K-x(t)}{x(t)}| &= -(\lambda t + C) \\
|\frac{K-x(t)}{x(t)}| &= e^{-(\lambda t + C)} \\
\frac{K-x(t)}{x(t)} &= e^{-\lambda t} C_0
\end{aligned}
$$

From that we can get:

$$x(t) = \frac{K}{1+C_0 e^{-\lambda t}}, \quad C_0 = \frac{K-x_0}{x_0}$$

And hence the $x(t) = \frac{Kx_0 e^{\lambda t}}{K+x_0(e^{\lambda t}-1)}$.

b)

The condition for the equilibrium is that $\frac{\mathrm{d}x}{\mathrm{d}t} = 0$

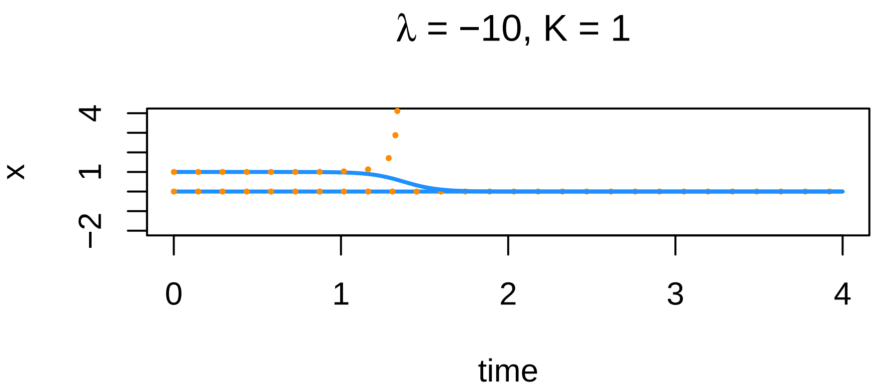
Solving $f'(x) = 0$, we have: $x^* = 0$ or $x^* = K$.

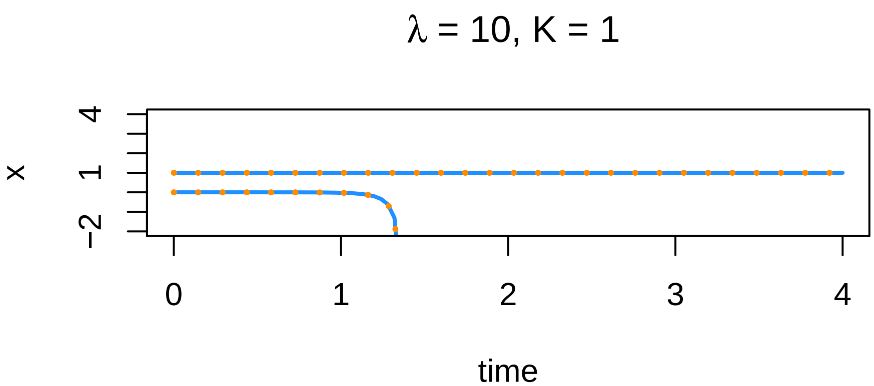
When $x^* = 0, f'(x^*) = \lambda$ and hence the point is **stable** if $\lambda < 0$ and **unstable** otherwise. When $x^* = K, f'(x^*) = -\lambda$ and hence the point is **unstable** if $\lambda < 0$ and **stable** otherwise.

c)

The results of the section b) above suggests the stability of equilibrium points depends on the value of λ chosen. The figures below do support the arugument made in b).

```r
library(deSolve)
k = 1
lambda = -10
parms <- c()
my.atol <- c(1e-06)
times <- c(0:100)/25
sdiffeqns <- function(t, s, parms) {
  sd1 <- lambda * s[1] * (1 - s[1]/k)
  list(c(sd1))
}
# just below 0
out1m <- lsoda(c(0 - 1e-06), times, sdiffeqns, rtol = 1e-10, atol = my.atol)
# just above 0
out1p <- lsoda(c(0 + 1e-06), times, sdiffeqns, rtol = 1e-10, atol = my.atol)
# just below k
out2m <- lsoda(c(k - 1e-06), times, sdiffeqns, rtol = 1e-10, atol = my.atol)
# just above k
out2p <- lsoda(c(k + 1e-06), times, sdiffeqns, rtol = 1e-10, atol = my.atol)
plot(out1m, xlab = "time", ylab = "x", col = "dodgerblue", lty = 1, lwd = 2,
     ylim = c(-2, 4), xlim = c(0, 4),
     main = expression(" "*lambda*" = -10, "*K*" = 1"))
lines(out1m, col = "#ff8c00", lty = 3, lwd = 3)
lines(out2m, col = "dodgerblue", lty = 1, lwd = 2)
lines(out2p, col = "#ff8c00", lty = 3, lwd = 3)
```





Code for the solution above is inspired by the tutorials materials and available on github repo: (https://github.com/wyq977/evolutionary-dynamics-2019)