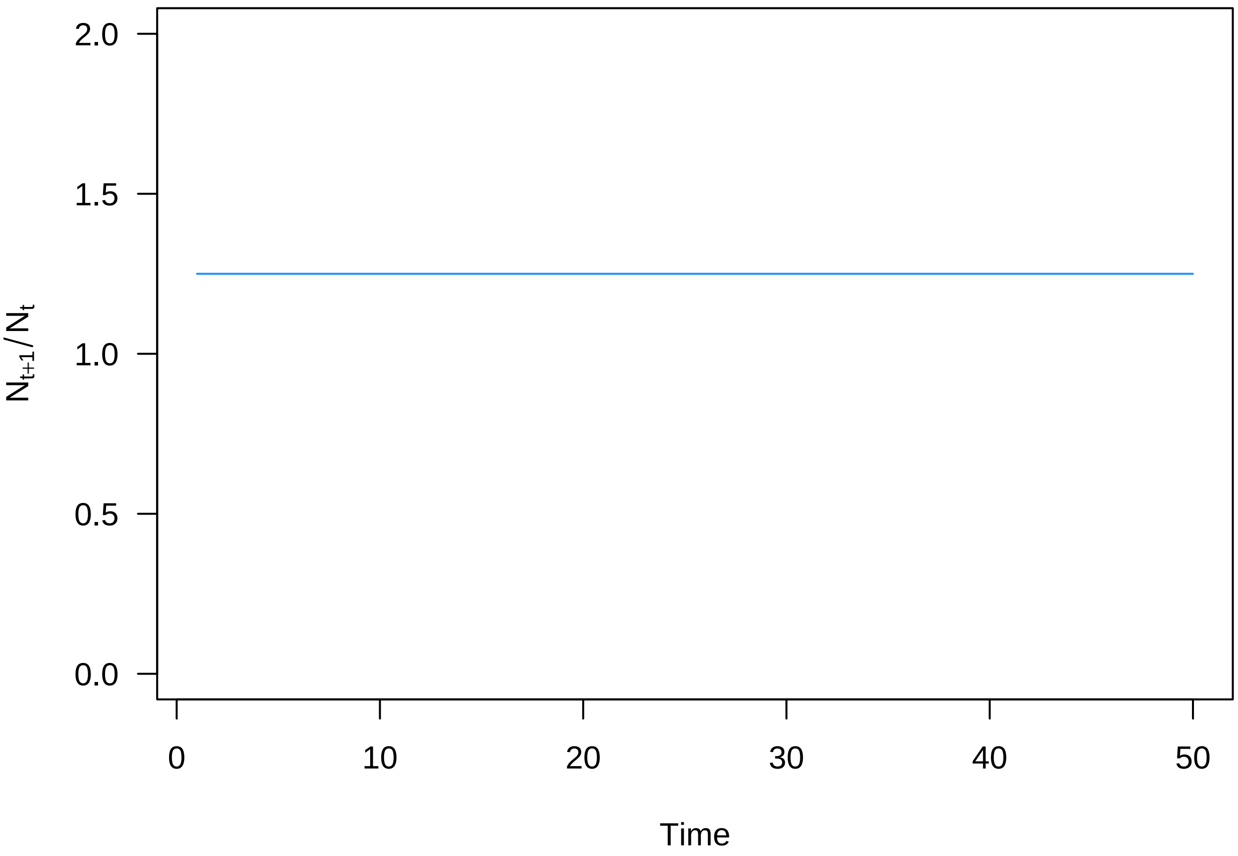
Constrained growth

Density-dependent population dynamics

The exponential model of population growth we introduced in the first week was a good starting point, but it was unrealistic for more species. That is, species do not typically grow towards infinitely large population sizes, as there are often forces that constrain population growth. Case outlines some of these at the start of chapter 5, and we will build on some of these ideas to start developing models of population dynamics including competition for space and resources. For now, let's consider competition as a simple process of density-dependence in growth rates. That is, the population should change depending on the size of the population. Let's explore this a bit. Recall the exponential model.

```
expoGrowth <- function(n, lam){
  n*lam
}

expoDynamics <- function(n,lambda, steps=100){
  if(length(lambda) < steps){
    lambda <- rep(lambda, steps)
  }
  ret <- c()
  ret[1] <- n
  for(i in 1:steps){
    ret[i+1] <- expoGrowth(ret[i], lambda[i])
  }
  return(ret)
}
```

```
n0 <- 20
t <- 50

dynamics <- expoDynamics(n0, 1.25, steps=t)
relativeChange <- dynamics[2:length(dynamics)] / dynamics[1:(length(dynamics)-1)]


par(mar=c(4,4,0.5,0.5))
plot(1:(t), relativeChange,
  type='l', las=1,
  xlab='Time',
  ylab=expression(N[t+1] / N[t]),
  col='dodgerblue',
  ylim=c(0,2))
```



This means that the rate of change in population size remains constant over time. Let's introduce a model that incorporates density-dependence and the re-visit this plot.

Logistic growth

It may be more realistic to assume that populations intrinsically limit themselves. That is, competition for space, resources, and mates, produces an upper limit to the population size (but not the growth rate). One way to think about this is that you can have a garden in which the number of individual plants is limited by available space or light, but the growth rates of each of the individual plants could be independent of these effects.

Discrete model

In the discrete model, we see that the population still grows at rate r (which is equal to $\ln(\lambda)$) , but overall population size is discounted by a scaling term which relates the population size (N_t) to an upper threshold. This threshold is the **carrying capacity (K)**, which is the maximum sustainable population size, given potentially limiting resource such as resources, space, etc.

$$N_{t+1} = N_t + (rN_t(1 - \frac{N_t}{K})) \tag{1}$$

Continuous model

In the continuous model, time step size goes to 0 in the limit (i.e., the time steps are really tiny). When the population size exceeds K (for either discrete or continuous models) population growth becomes negative, leading to a tendency for the system to go to K. However, this is sensitive to population growth rate (λ or r), as large growth rates can lead to complex dynamics, including damped oscillations, limit cycles, and chaos.

$$\frac{dN}{dt} = rN\left[1 - \frac{N}{K}\right], \qquad r, K > 0 \tag{2}$$

Note: in this model, r and K must be greater than 0.

$\lambda < 1$, $r < 0$: population decrease to 0 \ $\lambda = 1$, $r = 0$: population does not change \ $\lambda > 1$, $r > 0$: population increase to carrying capacity (K) \

Assumptions of the logistic model:

- Constant carrying capacity
- Linear density dependence (population size limits population growth, with each additional individual reducing growth rate equally).

Equilibria:

Equilibrial points are points which serve as attractors to the system. In the limit of infinite time, these points are points that satisfy the condition that the total change in the system is 0. For the exponential model, there was a single equilibrium point, which was when N=0 (i.e., the population was extinct). This point is considered unstable, for reasons we've discussed before. That is, if we were to perturb the system from this equilibrial state (perhaps we add a couple individuals into the system), the dynamics will diverge from this equiblrium point, and instead the population will grow towards infinity. But how can we make N=0 a stable equilibrium point. That is, when would perturbing the system (through the addition of a few individuals) lead the system to go back to this N=0 equilibrium?

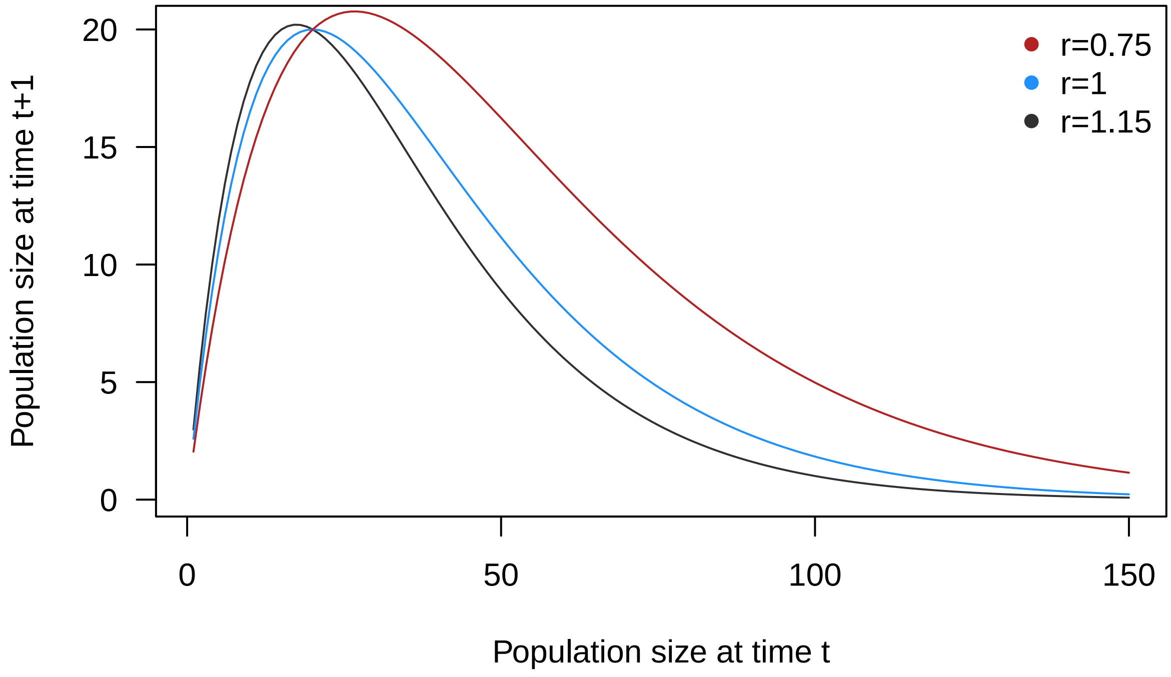
In the logistic model, we have two equiblrium points.

$N = K, 0 < r < 3.5$ (stable) $N = 0, 0 < r < 3.5$ (unstable)

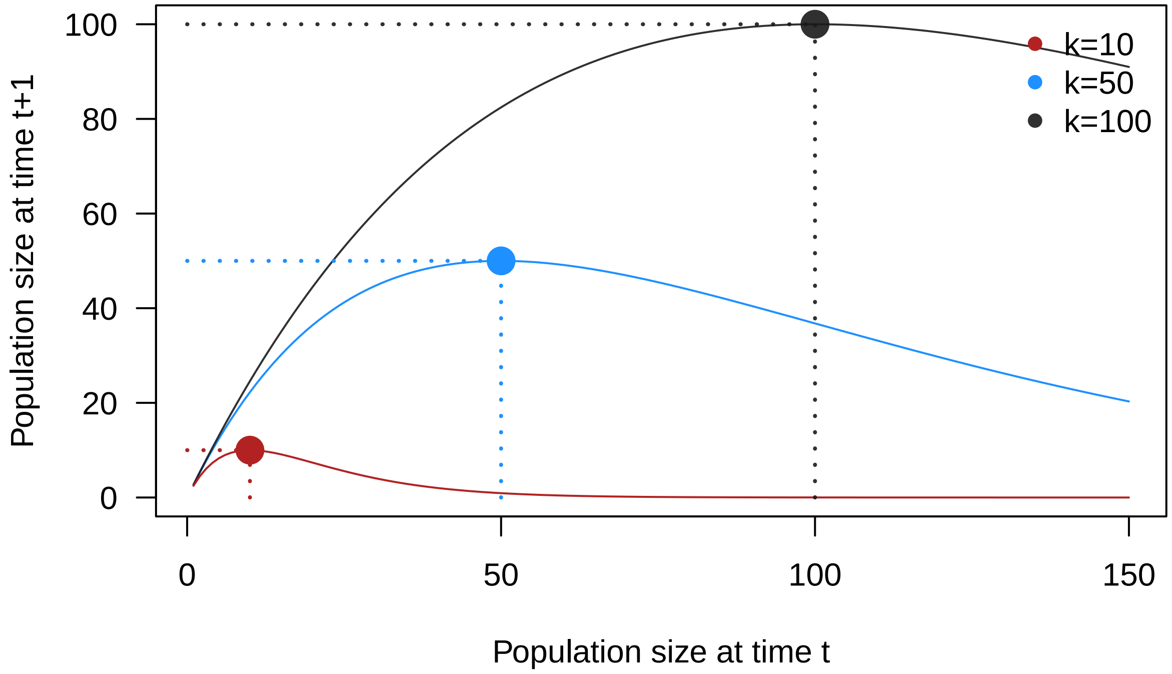
$N = K, r < 0$ (unstable) $N = 0, r < 0$ (stable)

Modeling growth and exploring chaos

```r
n0 <- 1:150
k <- 20
r <- 1


logisticGrowth <- function(n, r, k){
  n*exp(r*(1-(n / k)))
}


colz <- c(grey(0.1,0.9), 'dodgerblue', 'firebrick', 'forestgreen')
#effect of growth rate
plot(n0, logisticGrowth(n0,1.15,20),
  type='l', las=1,
  xlab='Population size at time t',
  ylab='Population size at time t+1',
  col=colz[1])
lines(n0, logisticGrowth(n0,1,20),
  col=colz[2])
lines(n0, logisticGrowth(n0,0.75,20),
  col=colz[3])
legend('topright', bty='n',
  c('r=0.75', 'r=1', 'r=1.15'),
  pch=16, col=colz[c(3,2,1)])
```

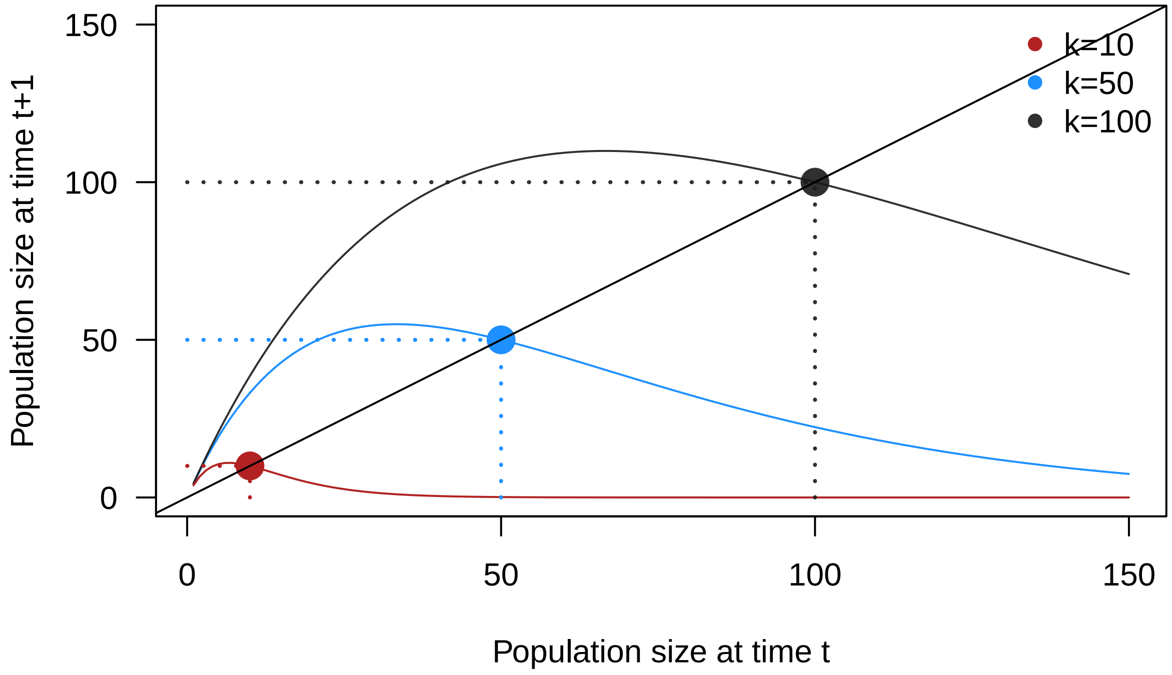


```
#effect of carrying capacity
plot(n0, logisticGrowth(n0,1, 50),
  type='l', las=1, ylim=c(0,100),
  xlab='Population size at time t',
  ylab='Population size at time t+1',
  col=colz[2])
lines(n0, logisticGrowth(n0,1,100),
  col=colz[1])
lines(n0, logisticGrowth(n0,1,10),
  col=colz[3])
legend('topright', bty='n',
  c('k=10', 'k=50', 'k=100'),
  pch=16, col=colz[c(3,2,1)])




#Look at the peaks of the growth (where is the maximum population size here?)

plotSegs <- function(kx,ky, color){
  segments(x0=0,x1=kx, y0=ky,y1=ky, col=color, lwd=2, lty=3)
  segments(x0=kx,x1=kx, y0=0,y1=ky, col=color, lwd=2, lty=3)
  points(kx,ky, pch=16, cex=2, col=color)
}

plotSegs(10,10, color=colz[3])
plotSegs(50,50, color=colz[2])
plotSegs(100,100, color=colz[1])
```



```
# But why is this? What situations would cause this to not be the case?

plot(n0, logisticGrowth(n0,1.5, 50),
  type='l', las=1, ylim=c(0,150),
  xlab='Population size at time t',
  ylab='Population size at time t+1',
  col=colz[2])
lines(n0, logisticGrowth(n0,1.5,100),
  col=colz[1])
lines(n0, logisticGrowth(n0,1.5,10),
  col=colz[3])
legend('topright', bty='n',
  c('k=10', 'k=50', 'k=100'),
  pch=16, col=colz[c(3,2,1)])



#Look at the peaks of the growth (where is the maximum population size here?)
plotSegs(10,10, color=colz[3])
plotSegs(50,50, color=colz[2])
```

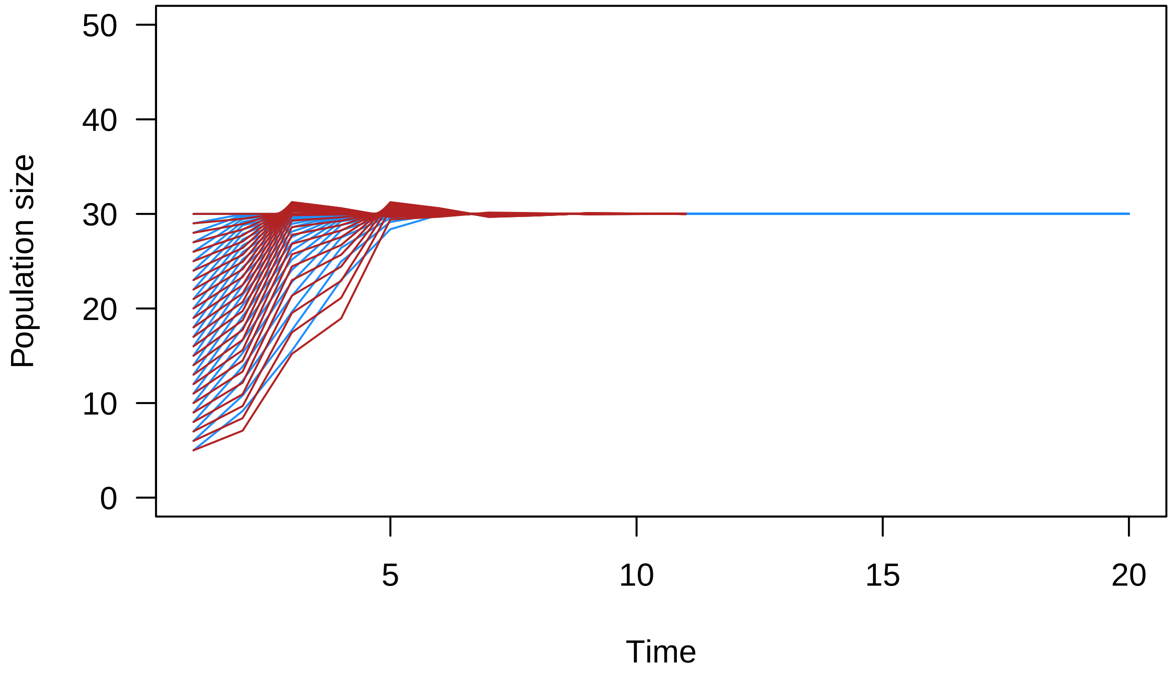
```r
plotSegs(100,100, color=colz[1])


# this line intersects points where population change from t to t+1 is 0. These are equilibria.
abline(a=0,b=1)
```



Alright. So now we can look at the actual dynamics across many generations.

```r
logisticGrowth <- function(n, r, k){
  return(n * (1+r*(1-(n/k))))
}


logisticDynamics <- function(n,r,k, steps=100){
  ret <- c()
  ret[1] <- n
  if(length(r) == 1){
    r <- rep(r, steps)
  }
  for(i in 1:(steps-1)){
    ret[i+1] <- logisticGrowth(ret[i], r[i], k)
  }
  return(ret)
}
```

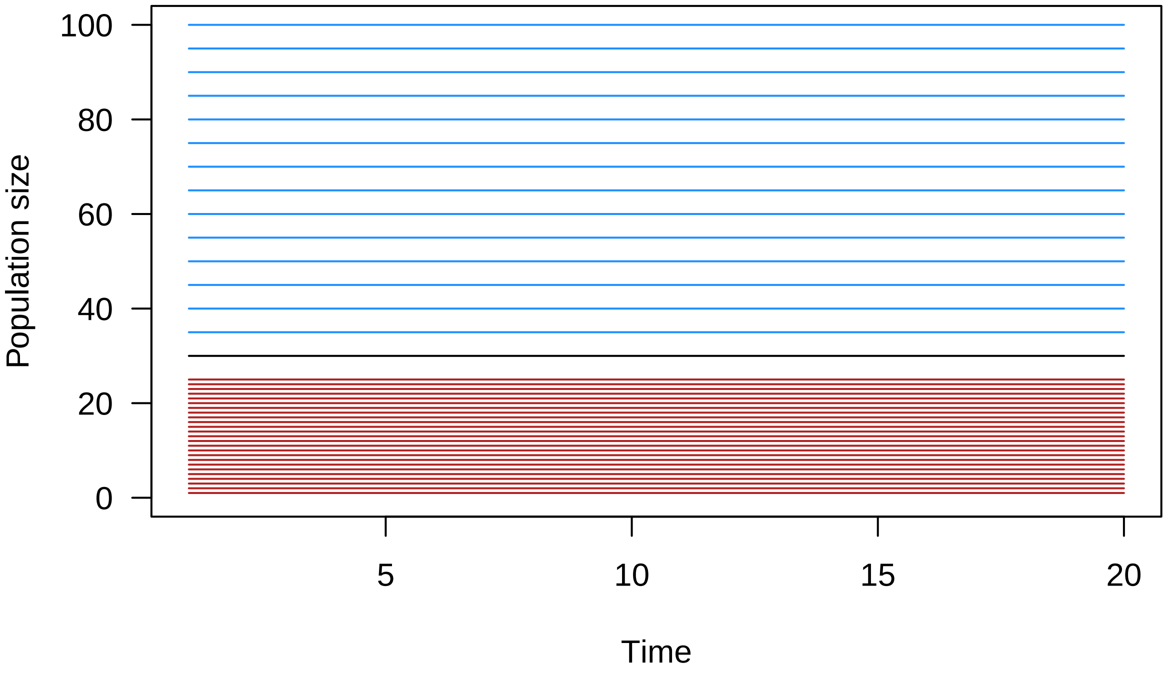
```
stps <- 20

plot(1:stps,
 logisticDynamics(n=30, r=1, k=30, steps=stps),
  type='l', las=1, ylim=c(0,50),
  xlab='Time',
  ylab='Population size',
  col=1)

#sapply(seq(1,25,by=1), function(x){
# lines(logisticDynamics(n=x, r=1, k=30, steps=stps), col='firebrick')
#})


sapply(seq(5,30,by=1), function(x){
 lines(logisticDynamics(n=x, r=1, k=30, steps=stps), col='dodgerblue')
})

sapply(seq(5,30,by=1), function(x){
 lines(logisticDynamics(n=x, r=rep(c(0.5,1.5),5), k=30, steps=stps), col='firebrick')
})
```



What if growth rate is not 1?

r=0

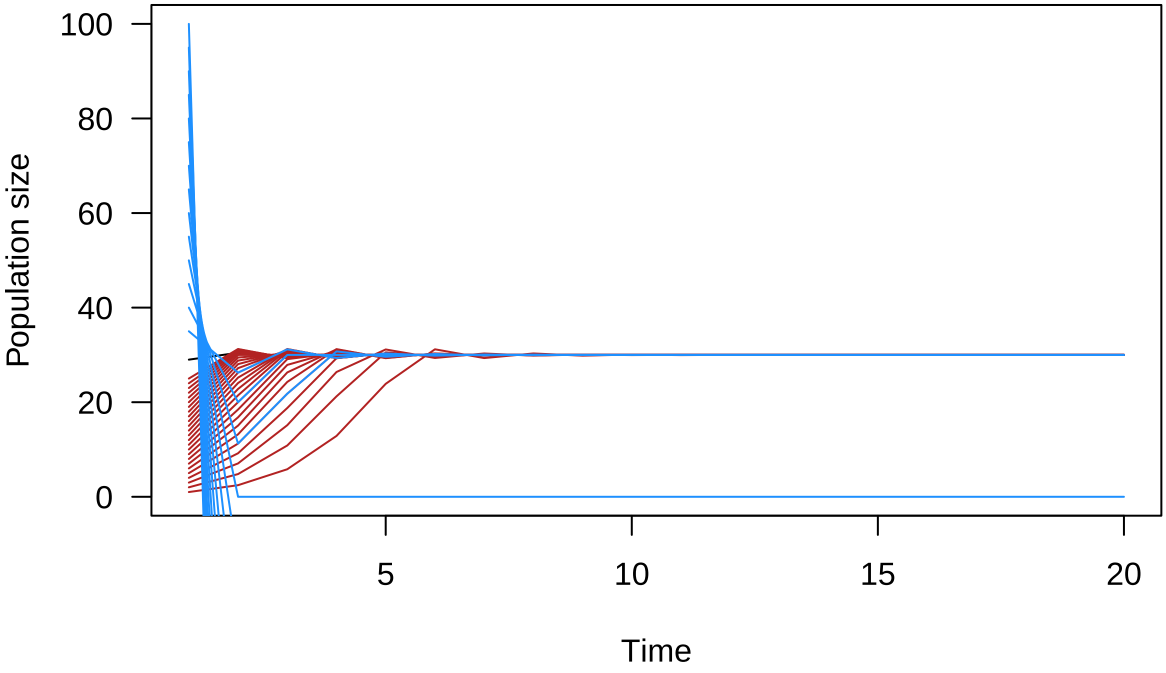
```
stps <- 20

plot(1:stps,
 logisticDynamics(n=30, r=0, k=30, steps=stps),
  type='l', las=1,ylim=c(0,100),
  xlab='Time',
  ylab='Population size',
  col=1)

sapply(seq(1,25,by=1), function(x){
 lines(logisticDynamics(n=x, r=0, k=30, steps=stps), col='firebrick')
})

sapply(seq(35,100,by=5), function(x){
 lines(logisticDynamics(n=x, r=0, k=30, steps=stps), col='dodgerblue')
})
```



r=1.5

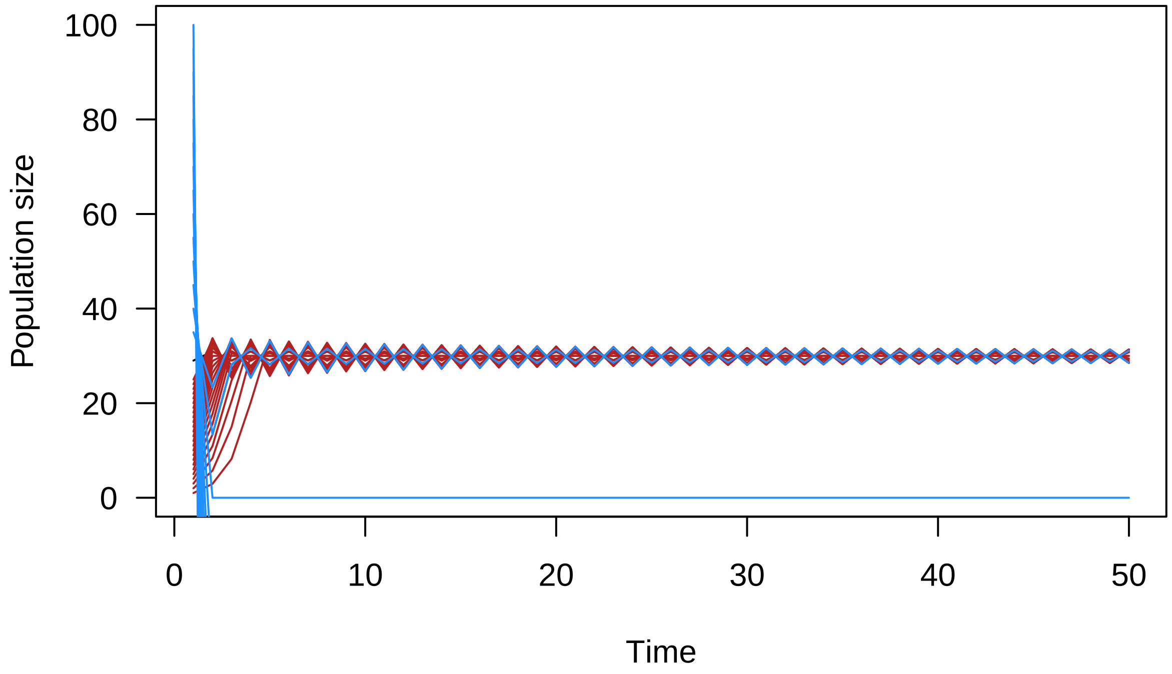
```
stps <- 20

plot(1:stps,
 logisticDynamics(n=29, r=1.5, k=30, steps=stps),
  type='l', las=1,ylim=c(0,100),
  xlab='Time',
  ylab='Population size',
  col=1)
```

```
sapply(seq(1,25,by=1), function(x){
 lines(logisticDynamics(n=x, r=1.5, k=30, steps=stps), col='firebrick')
})

sapply(seq(35,100,by=5), function(x){
 lines(logisticDynamics(n=x, r=1.5, k=30, steps=stps), col='dodgerblue')
})
```



r=2

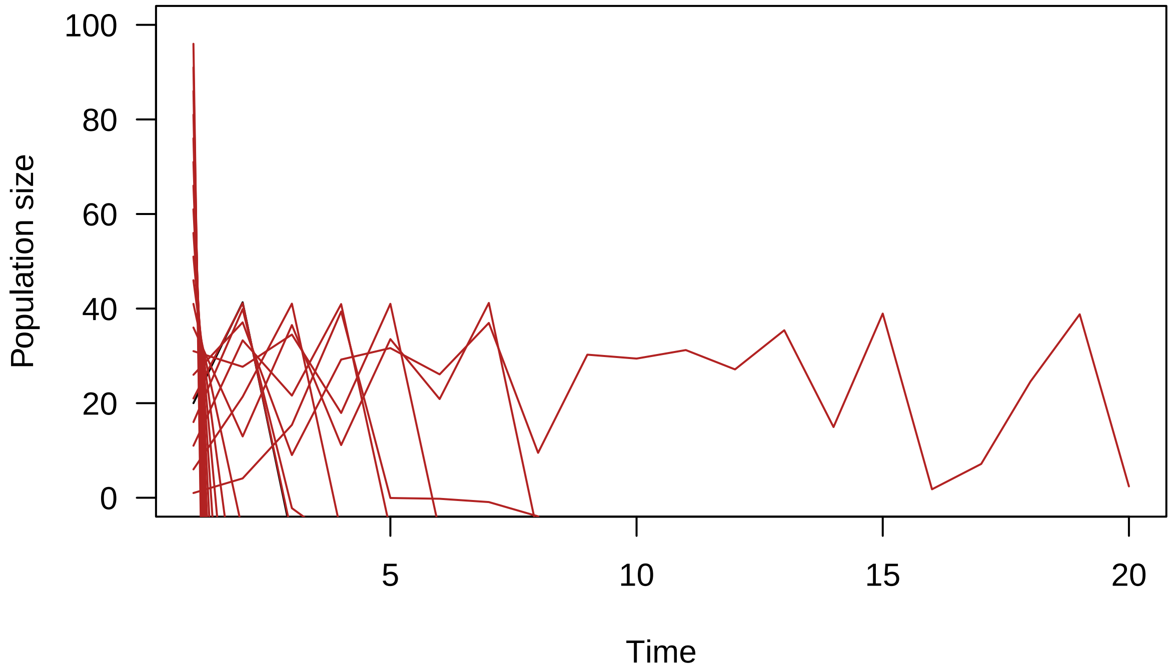
```
stps <- 50

plot(1:stps,
 logisticDynamics(n=29, r=2, k=30, steps=stps),
  type='l', las=1,ylim=c(0,100),
  xlab='Time',
  ylab='Population size',
  col=1)

sapply(seq(1,25,by=1), function(x){
 lines(logisticDynamics(n=x, r=2, k=30, steps=stps), col='firebrick')
})

sapply(seq(35,100,by=5), function(x){
 lines(logisticDynamics(n=x, r=2, k=30, steps=stps), col='dodgerblue')
})
```



(between 3 and 3.449 – oscillates between 2 values)

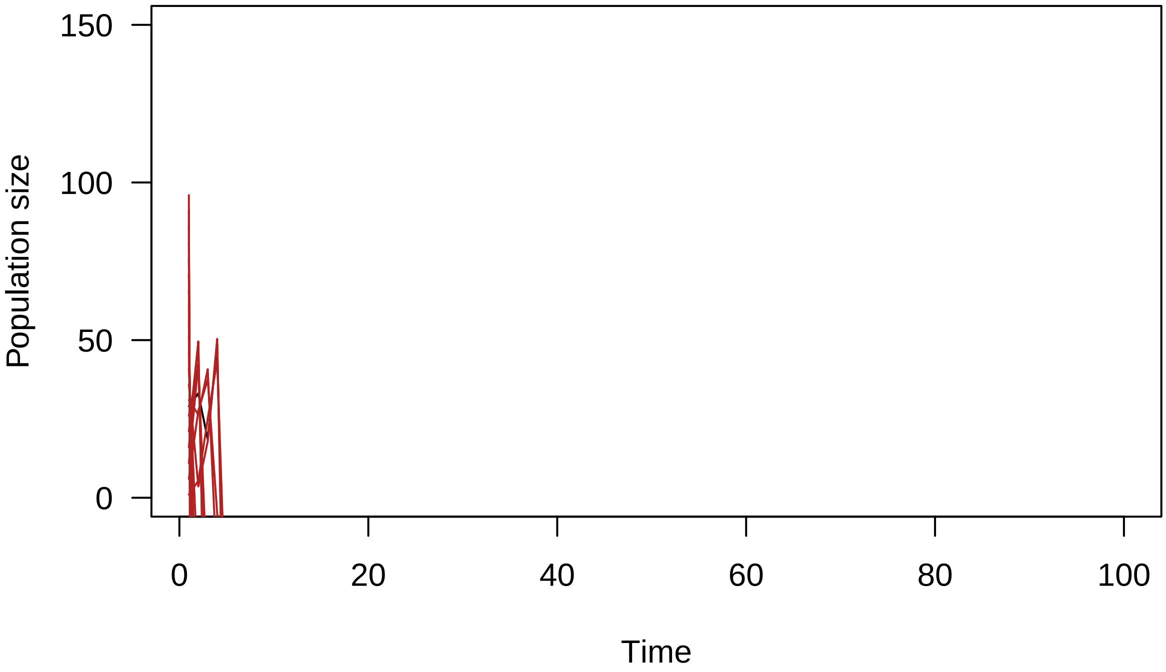
r=3.2

```
stps <- 20

plot(1:stps,
 logisticDynamics(n=20, r=3.2, k=30, steps=stps),
  type='l', las=1,ylim=c(0,100),
  xlab='Time',
  ylab='Population size',
  col=1)


sapply(seq(1,100,by=5), function(x){
 lines(logisticDynamics(n=x, r=3.2, k=30, steps=stps), col='firebrick')
})
```



(onset of chaos) r > 3.56

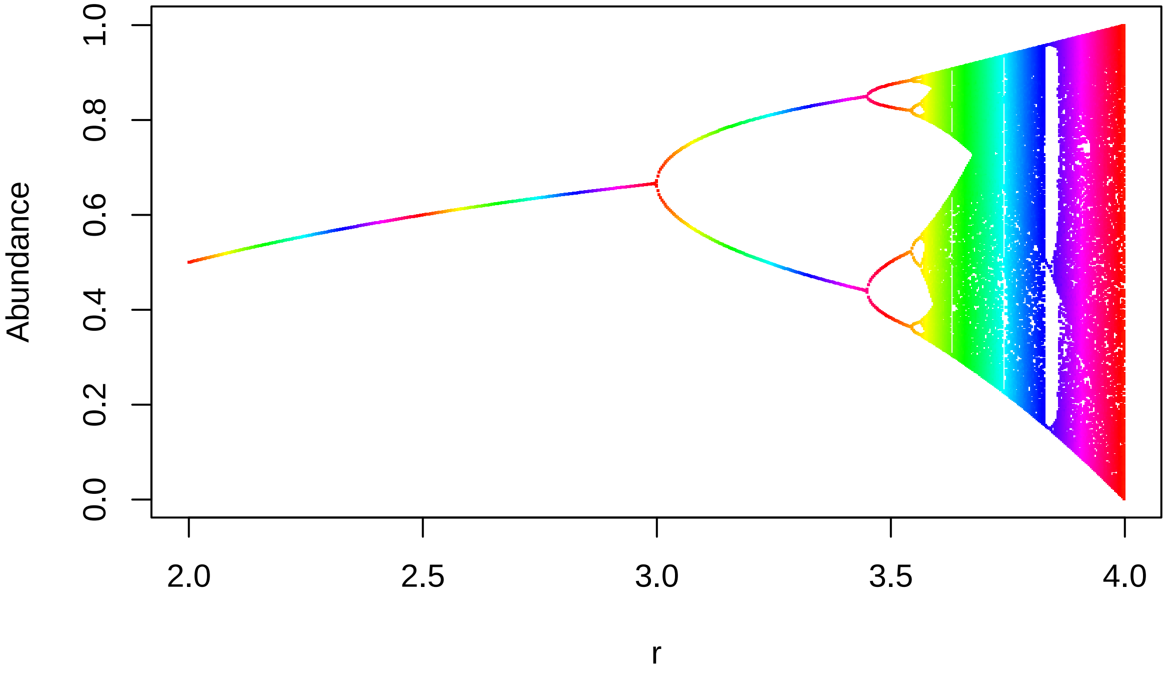
```
stps <- 100

plot(1:stps,
 logisticDynamics(n=29, r=4.25, k=30, steps=stps),
  type='l', las=1, ylim=c(0,150),
  xlab='Time',
  ylab='Population size',
  col=1)

sapply(seq(1,100,by=5), function(x){
 lines(logisticDynamics(n=x, r=4.5, k=30, steps=stps), col='firebrick')
})
```



The logistic map

The logistic map is a diagram which demonstrates the chaotic dynamics of population growth when population growth rate becomes large. We essentially simulate the dynamics of the model across a very fine gradient of population growth rates and the same initial population size. For the plot below, we simulate the dynamics and then return the last M values of the simulated dynamics for plotting. We plot the resulting last M timesteps of the simulation and the corresponding population growth rate. The divergence in dynamics is therefore a sensitivity to initial conditions.

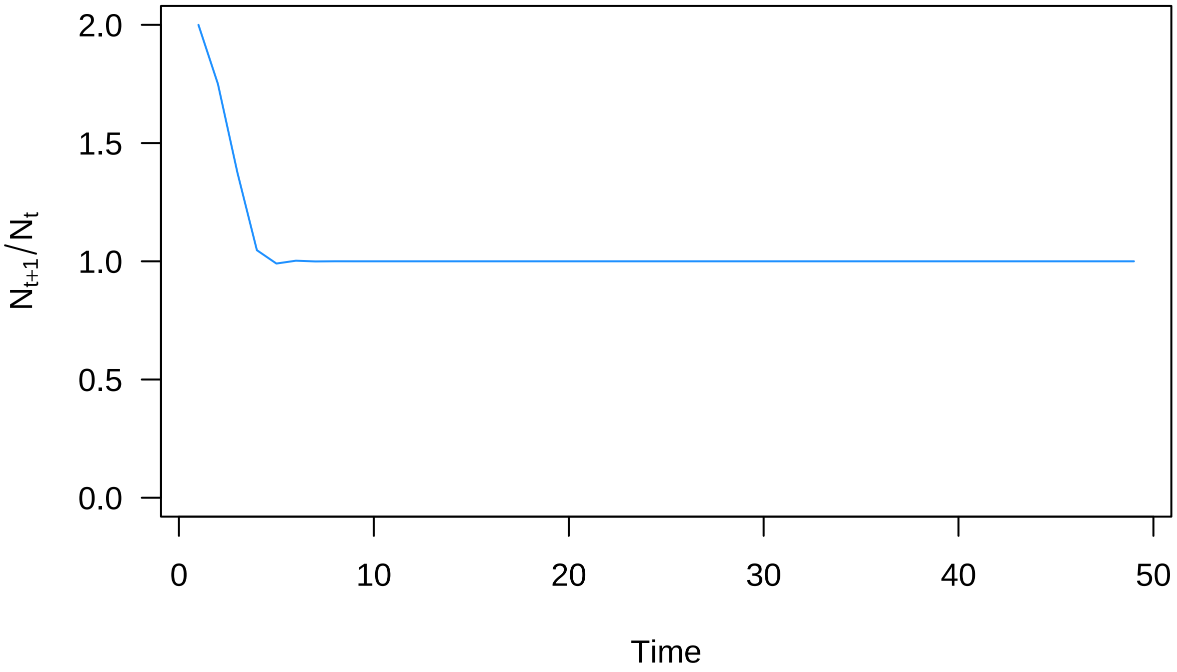
```
logistic.map <- function(r, x, N, M){
  z <- 1:N
  z[1] <- x
  for(i in c(1:(N-1))){
    z[i+1] <- r *z[i]  * (1 - z[i])
  }
  z[c((N-M):N)]
}

## Set scanning range for bifurcation parameter r
my.r <- seq(2, 4, by=0.003)
logMap <- as.vector(sapply(my.r, logistic.map,  x=0.1, N=1000, M=300))
r <- sort(rep(my.r, 301))
plot(logMap ~ r, pch=".",
    ylab='Abundance',
    col=rainbow(50000))
```



Revisiting density dependence relative to the exponential model

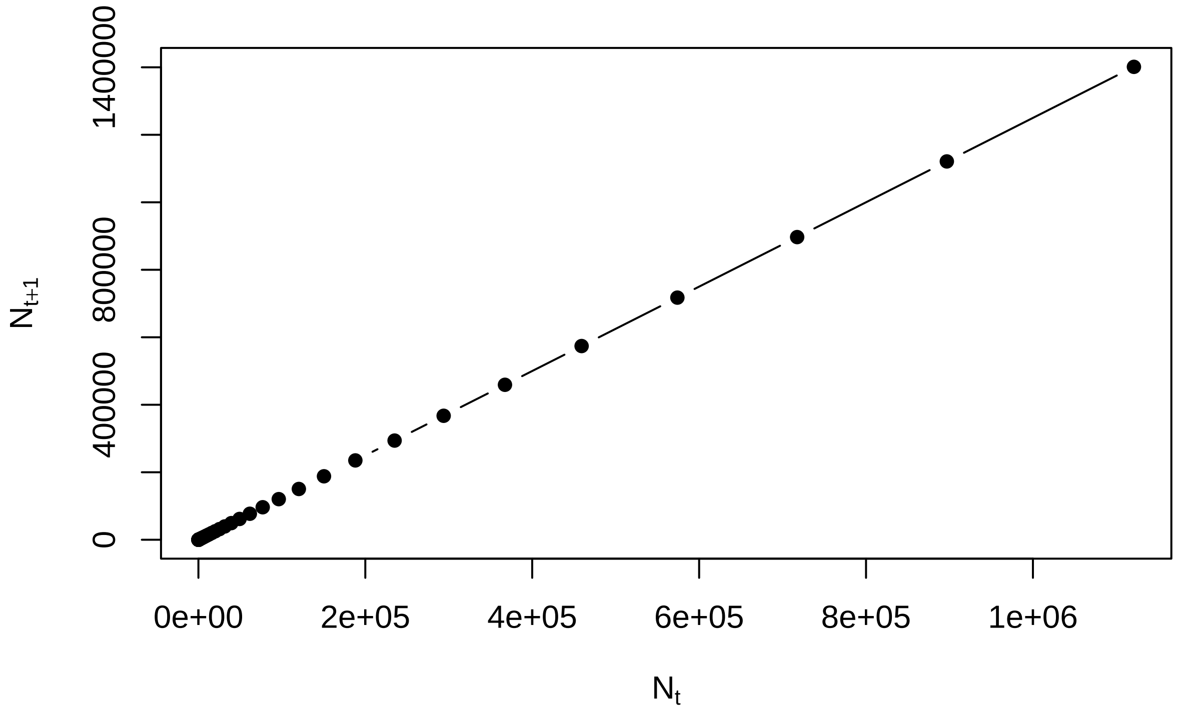
Case describes density-dependence using plots of population size over time compared to the rate of change in population size (Figure 5.1 and 5.2). We simply divided population size at N_{t+1} by N_t. For the exponential model, this will always output the same value, which is λ, the population growth rate. When we plot this out for the logistic model, we see that the rate of change starts off high and quickly saturates to 1, meaning that population size at time t is the same as time $t + 1$.

```
n0 <- 20
t <- 50

dynamicsL <- logisticDynamics(n0, 1.25, steps=t, k=100)
relativeChangeL <- dynamicsL[2:length(dynamicsL)] / dynamicsL[1:(length(dynamicsL)-1)]

plot(1:(t-1), relativeChangeL,
  type='l', las=1,
  xlab='Time',
  ylab=expression(N[t+1] / N[t]),
  col='dodgerblue',
  ylim=c(0,2))
```



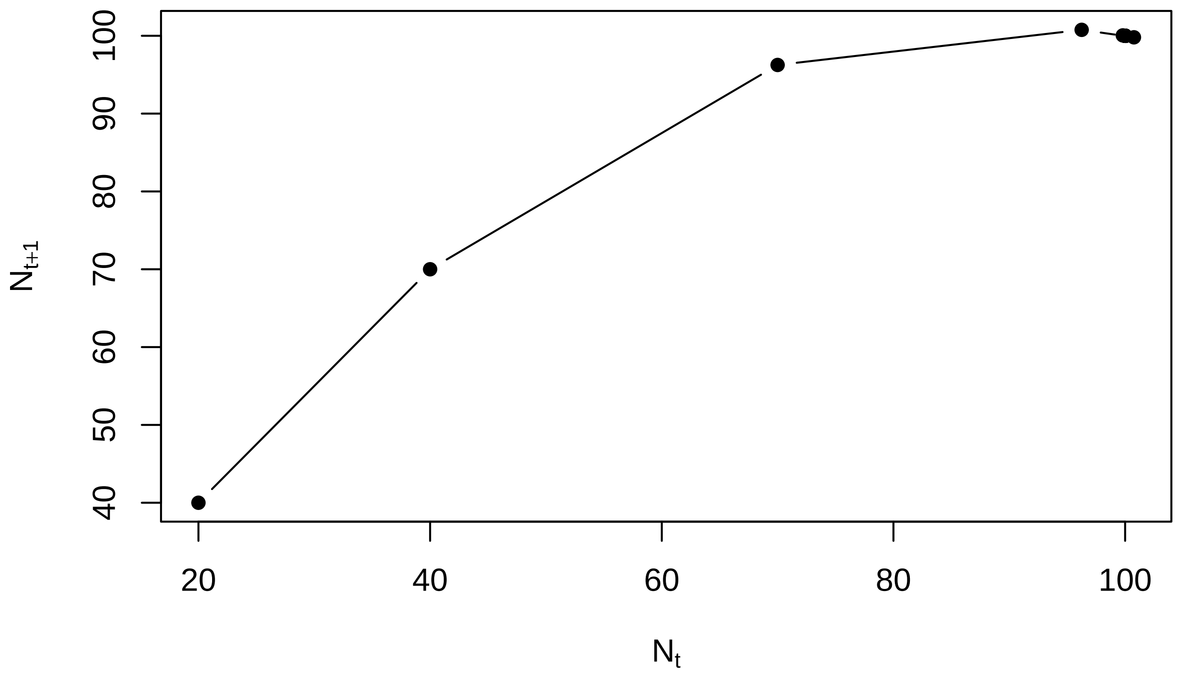
We can explore this another way as well, to really cement the point about density dependence. Let's consider the plot of N_t and $N_t + 1$. In the exponential model, this would look like a straight line.

```
plot(dynamics[1:(length(dynamics)-1)],
    dynamics[2:length(dynamics)],
    pch=16, type='b',
    ylab=expression(N[t+1]),
    xlab=expression(N[t])
)
```



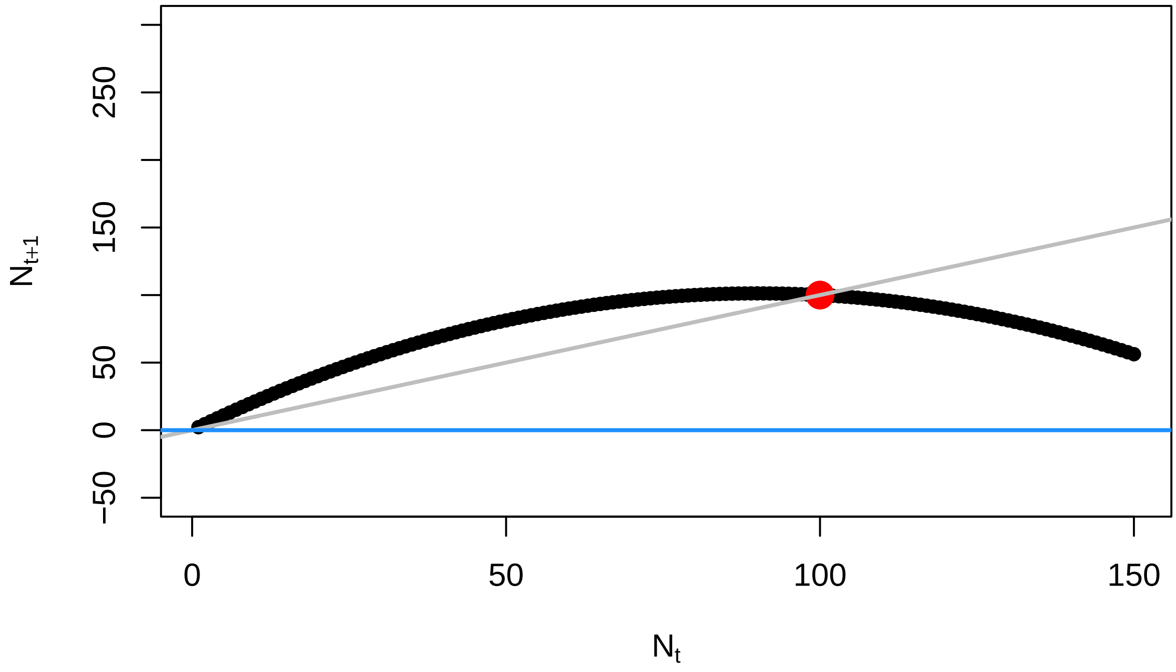
And it does. This indicates a lack of density-dependence, as the relative increase in population size is constant, such that the distance between points will increase, but the slope of the relationship will always stay the same (i.e., the population is growin at the same rate independent of population size). Now let's do the same for the logistic model.

```
plot(dynamicsL[1:(length(dynamicsL)-1)],
    dynamicsL[2:length(dynamicsL)],
    pch=16, type='b',
    ylab=expression(N[t+1]),
    xlab=expression(N[t])
)
```



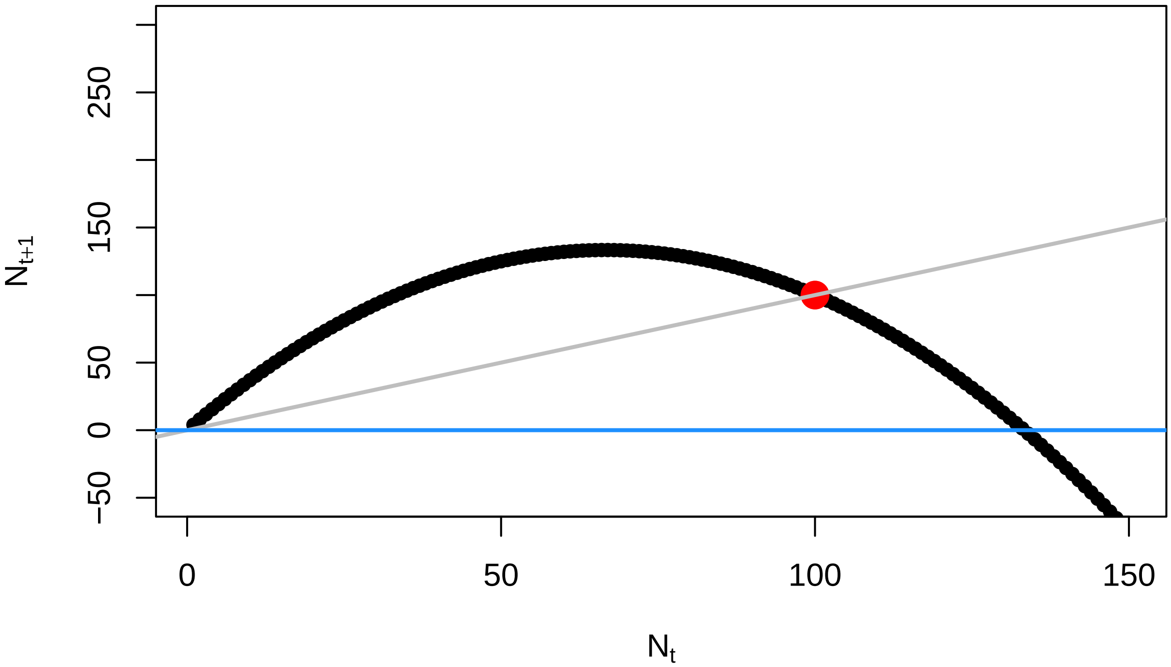
This shows a clear curved relationship, where it starts off with one slope when population size is small, but this slope decreases quickly as population size becomes large, and most of the points are the at the carrying capacity. Let's explore this when we consider the dynamics along the entire abundance gradient (i.e., let's explore this when the population starts over it's carrying capacity).

```
plot(1:150,
    logisticGrowth(1:150, r=1.25, k=100),
    pch=16, type='b', ylim=c(-50,300),
    ylab=expression(N[t+1]),
    xlab=expression(N[t])
)
points(100,100, pch=16, col='red', cex=2)
abline(b=1,a=0, lwd=2, col='grey')
abline(h=0, lwd=2, col='dodgerblue')
```



And we can explore how this curve is influenced by changing the population growth rate.

```
plot(1:150,
    logisticGrowth(1:150, r=3, k=100),
    pch=16, type='b', ylim=c(-50,300),
    ylab=expression(N[t+1]),
    xlab=expression(N[t])
)
points(100,100, pch=16, col='red', cex=2)
abline(b=1,a=0, lwd=2, col='grey')
abline(h=0, lwd=2, col='dodgerblue')
```



It is important to note that the continuous time logistic model would not produce the same rich set of dynamics and deterministic chaos as the discrete time model. This is because there is an implicit lag in response of one time step in the discrete model, allowing for populations to overshoot carrying capacity and start to show the interesting dynamics we visualized above. Another important issue with the logistic model is the ability to reach negative population sizes, as we see in the above plot. That is, if we started a population with 200 individuals and a carrying capacity of 100, the resulting predicted population size would be negative (depending on population growth rate). We can see this by simulating the above model.

```
logisticGrowth(200, r=3, k=100)
```

```
## [1] -400
```

Now we will introduce a model which incorporates density-dependence in population growth rate, but gets around some issues of the logistic model.

Ricker Logistic Equation

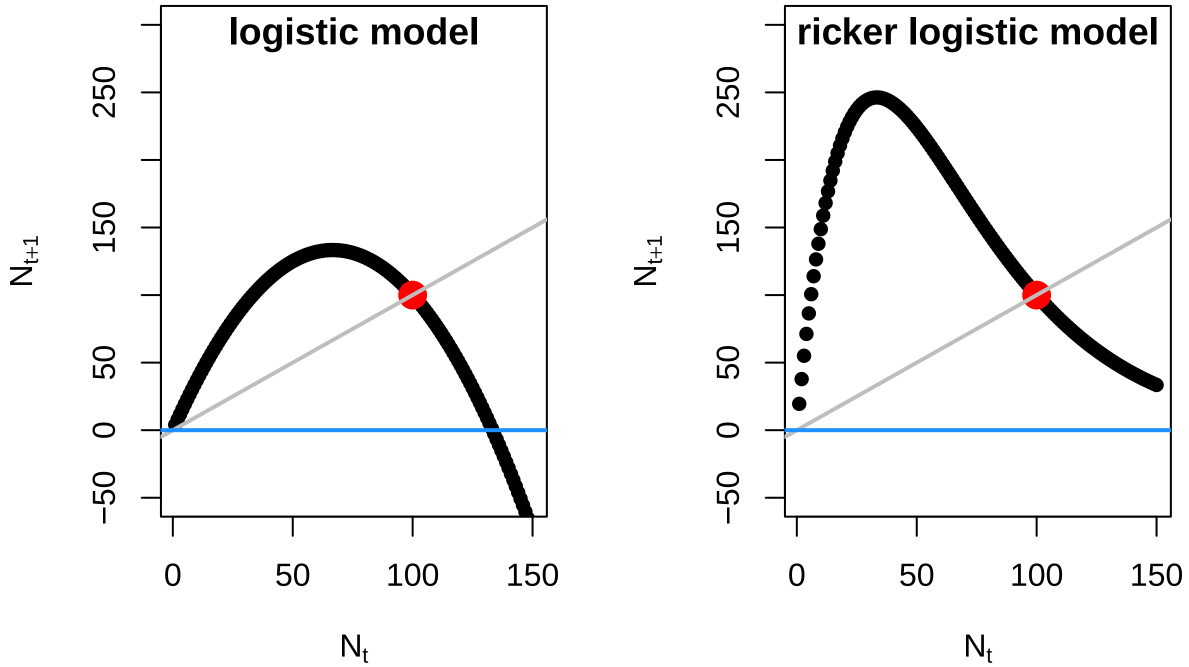
The Ricker logistic equation uses an exponential term to ensure that the population size in the next generation (N_{t+1}) will always be positive, with N_{t+1} approaching 0 as N_t approaches infinity.

$$N_{t+1} = N_t exp\left(R(1 - \frac{N_t}{K})\right) \tag{3}$$

```
rickerGrowth <- function(n, r, k){
  return(n * exp(r*(1-(n/k))))
}
```

```
rickerDynamics <- function(n,r,k, steps=100){
  ret <- c()
  ret[1] <- n
  if(length(r) == 1){
    r <- rep(r, steps)
  }
  for(i in 1:(steps-1)){
    ret[i+1] <- logisticGrowth(ret[i], r[i], k)
  }
  return(ret)
}
```

```
layout(matrix(c(1,2), ncol=2))
plot(1:150,
    logisticGrowth(1:150, r=3, k=100),
    pch=16, type='b', ylim=c(-50,300),
    ylab=expression(N[t+1]),
    xlab=expression(N[t])
)
title(line=-1, 'logistic model')
points(100,100, pch=16, col='red', cex=2)
abline(b=1,a=0, lwd=2, col='grey')
abline(h=0, lwd=2, col='dodgerblue')

plot(1:150,
    rickerGrowth(1:150, r=3, k=100),
    pch=16, type='b', ylim=c(-50,300),
    ylab=expression(N[t+1]),
    xlab=expression(N[t])
)
title(line=-1, 'ricker logistic model')
points(100,100, pch=16, col='red', cex=2)
abline(b=1,a=0, lwd=2, col='grey')
abline(h=0, lwd=2, col='dodgerblue')
```


logistic model

ricker logistic model