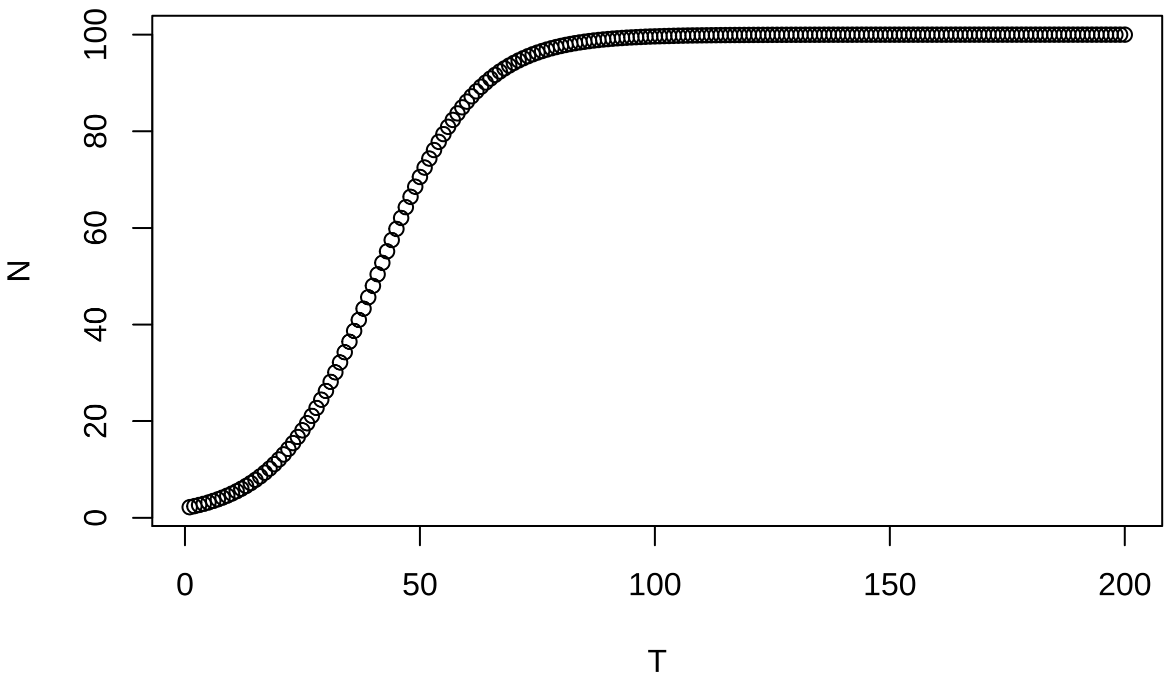
Section12

Exercises for section 12

12.1.0.0.1

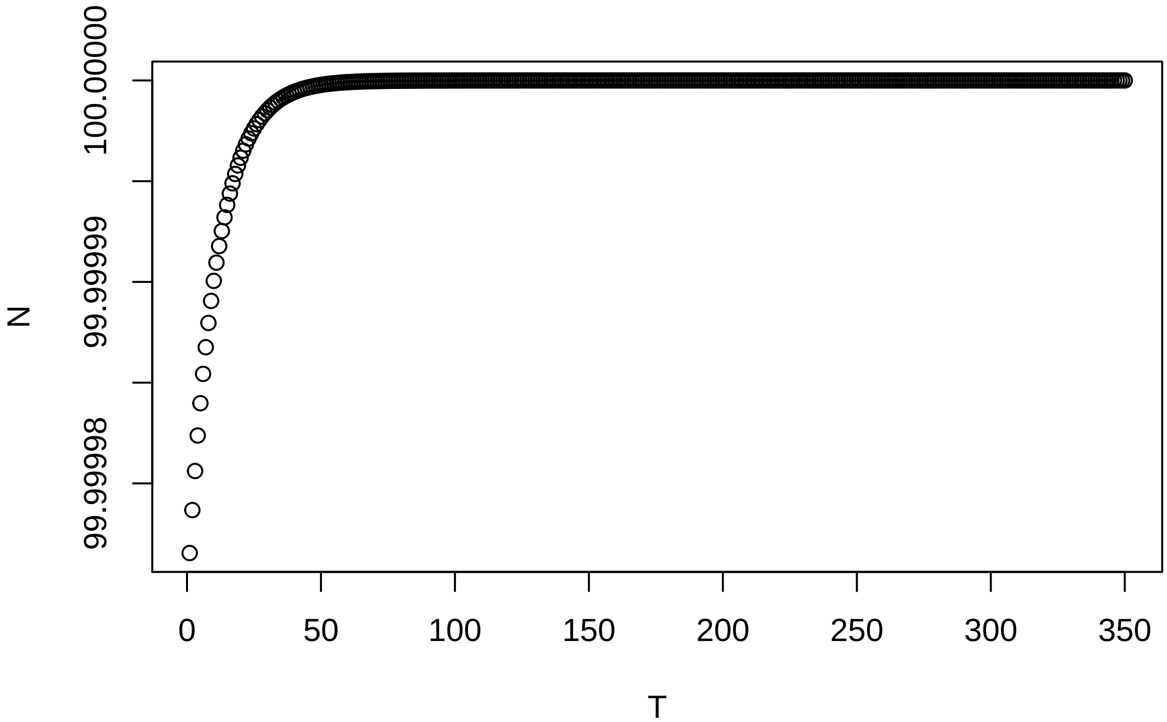
Original method

```
a <- 1.1
b <- 0.001
T <- seq(from=1,to=200,by=1)
N <- numeric(length(T))
n <- 2
for (t in T) {
  n <- a*n/(1+b*n)
  N[t] <- n
}
plot(T,N)
```



Second method

```
T<-seq(from=1,to=350,by=1)
N <- numeric(length(T))
for (t in 1:length(T)) {
  n <- a*n/(1+b*n)
  N[t] <- n
}
plot(T,N)
```

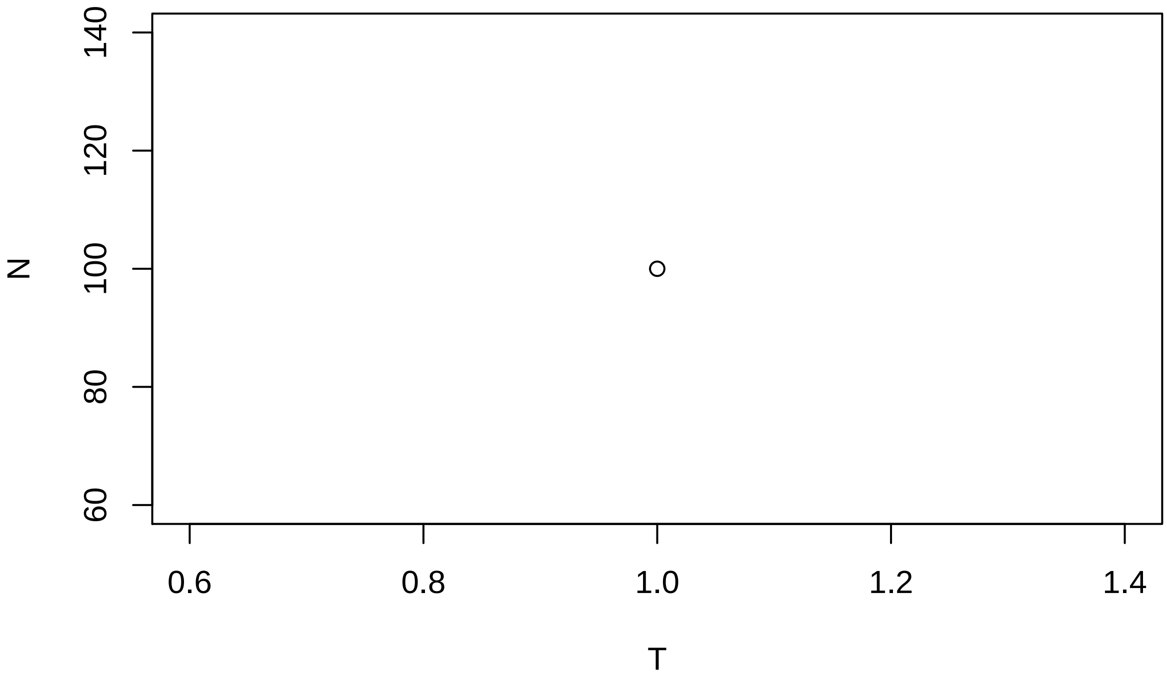


It works

12.1.0.0.2

Second method

```
T<-c(1)
N <- numeric(length(T))
for (t in 1:length(T)) {
  n <- a*n/(1+b*n)
  N[t] <- n
}
plot(T,N)
```



When T has a length of 1, only a single point it plotted

```
# T<-numeric()
# N <- numeric(length(T))
# for (t in 1:length(T)) {
#   n <- a*n/(1+b*n)
#   N[t] <- n
# }
# plot(T,N)
```

When T is empty (or has a length of 0) an error is plotted instead of an empty graph because T has no length and vectors of different lengths cannot be used to plot x and y coordinates

12.2.0.0.1

```
phi <- 20
conv <- FALSE
i<-0
while (!conv) {
  phi.new <- 1+1/phi
  conv <- phi==phi.new
  phi <- phi.new
  i<-i+1
}
```

It takes 40 iterations for phi to become constant

12.2.0.0.2

```
a <- 1.1
b <- 0.001
n<-1
while (TRUE) {
  n.new <- a*n/(1+b*n)
```

```
  if (n==n.new) break
  n <- n.new
}
print(n)
```

```
## [1] 100
```

```
N[200]
```

```
## [1] NA
```

n in this case is equal to the value the computations above converged at for Beverton-Holt