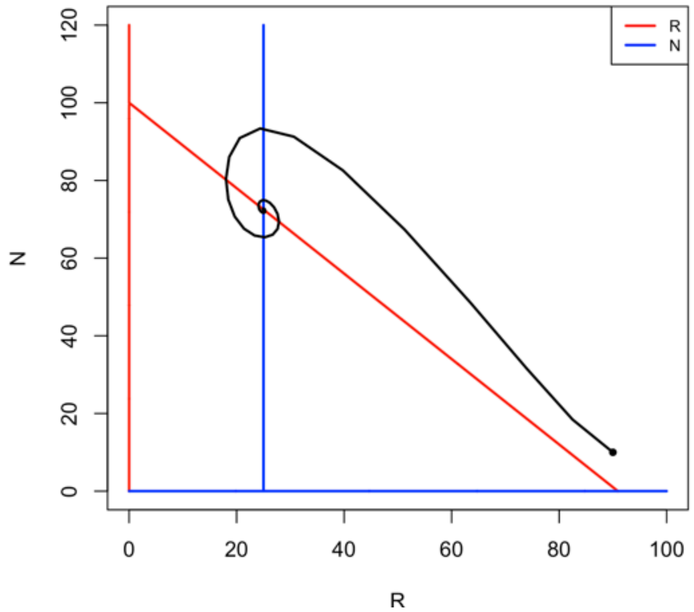
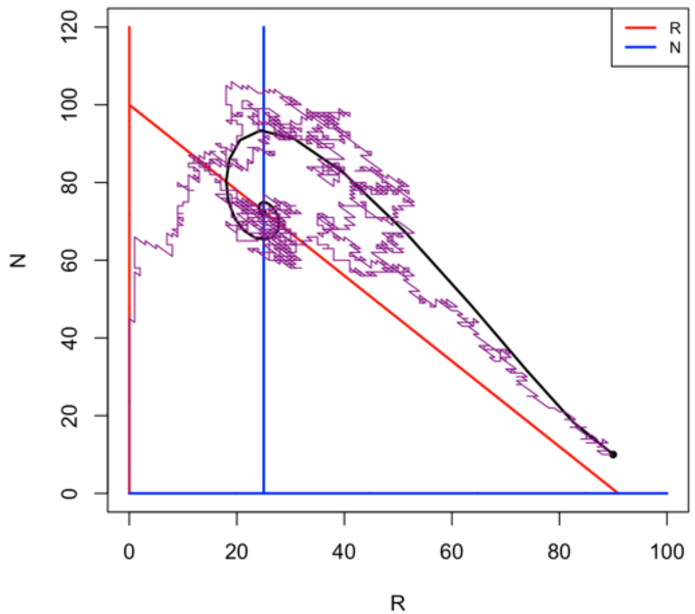
Gillespie simulations

```r
model <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    dR <- b*R*(1 - R/K) - d*R - a*R*N
    dN <- a*R*N - delta*N
    return(list(c(dR, dN)))
  })
}
p <- c(b=1.1, d=0.1, K=100, a=0.01, delta=0.25)
s <- c(R=90,N=10)
plane(0,100,0,120,eps=-0.01);run(traject=TRUE)

t <- 0; tmax <- 100; nrun <- 5
updates <- list(
  birth=c(1,0), deathR=c(-1, 0), kill=c(-1, 1), deathN=c(0,-1))
nsols <- list()
plane(0,100,0,120);run(traject=TRUE)

for (i in seq(nrun)){
  t <-0; dead <- FALSE; state <- s
  nsol <- as.data.frame(t(c(time=0,state)))
  while(t < tmax && !dead) {
    rates <- with(as.list(c(state,p)),
                  c(max(0,b*R*(1 - R/K)), d*R, a*R*N, delta*N))
    summed <- cumsum(rates)
    total <- summed[length(summed)]
    if (total < 1e-12) {dead <- TRUE}
    else {
      u <- runif(1,0,total)
      state <- state + updates[[which.max(u < summed)]]
      t <- t + rexp(1)/total
    }
    nsol[nrow(nsol)+1,] <- c(t,as.numeric(state))
  }
  lines(nsol$R,nsol$N,col=colors[i])
  nsols[[i]] <- nsol
}
```





Exercise 5.11