```

# simple machine repair model; m machines, r repairpersons; expon up and
# repair times, everything independent

mrp <- function(meanup,meanrep,timelim,m,r,dbg=FALSE) {
   # set up structures
   simlist <- newsim(timelim,m,
      appcols=c(’startqtime’,’startuptime’),dbg=dbg)
   simlist$reactevent <- mrpreact
   simlist$uprate <- 1.0 / meanup
   simlist$reprate <- 1.0 / meanrep
   simlist$nmach <- m
   simlist$nrepairpersons <- r
   simlist$queue <- newqueue(simlist)  # queue for the repairpersons
   simlist$nup <- m  # all machines up initially
   simlist$nrepbusy <- 0  # number of busy repairpersons
   simlist$breakevnt <- 1  # breakdown
   simlist$repairevnt <- 2  # good as new!
   # bookkeeping
   # say we are interested in mean wait until repair, and overall up time
   # proportion per machine
   simlist$nrepairs <- 0
   simlist$totqtime <- 0.0
   simlist$totuptime <- 0.0

   # get the ball rolling: set breakdown events for the machines, which
   # are all currently up at time 0.0
   for (i in 1:m) {
      whenbreak <- rexp(1,simlist$uprate)
      # schedule a breakdown event for machine i at time whenbreak, with
      # startqtime NA, not in a queue as of now, and start of up time 0
      schedevnt(simlist,whenbreak,simlist$breakevnt,appdata=c(NA,0))
   }

   # start sim
   mainloop(simlist)

   # sim done
   cat("mean queuing time:  ")
   print(simlist$totqtime / simlist$nrepairs)
   cat("proportion up per machine:  ")
   print(simlist$totuptime / (simlist$nmach * simlist$timelim))
}

# what new events are triggered by the occurrence of an old one?
mrpreact <- function(evnt,simlist) {
   etype <- evnt[’evnttype’]
   if (etype == simlist$breakevnt) {  # machine has gone down
      # record this up time
      simlist$totuptime <-
         simlist$totuptime + simlist$currtime - evnt[4]
      # is there is a free repairperson?
      nrepb <- simlist$nrepbusy
      if (nrepb < simlist$nrepairpersons) {
         # start repair, no queuing
         simlist$nrepbusy <- nrepb + 1
         repduration <- rexp(1,simlist$reprate)
         schedevnt(simlist,simlist$currtime+repduration,simlist$repairevnt,
            appdata=c(NA,NA))
      } else {  # no repairpersons free, add job to queue
         # record start queue wait
         evnt[3] <- simlist$currtime
         appendfcfs(simlist$queue,evnt)
      }
   } else {  # etype = simlist$repairevnt
      # bookkeeping
      simlist$nrepairs <- simlist$nrepairs + 1
      # start next up time for this machine
      uptime <- rexp(1,simlist$uprate)
      schedevnt(simlist,simlist$currtime+uptime,simlist$breakevnt,
         appdata=c(NA,simlist$currtime))
      # repairperson now free
      simlist$nrepbusy <- simlist$nrepbusy - 1
      # check queue for waiting jobs
      if (nrow(simlist$queue$m) > 0) {  # nonempty queue
         qhead <- delfcfs(simlist$queue)
         # bookkeeping
         simlist$totqtime <- simlist$totqtime + simlist$currtime - qhead[3]
         # start job service
         # this repairperson now busy
         simlist$nrepbusy <- simlist$nrepbusy + 1
         srvduration <- rexp(1,simlist$reprate)
         schedevnt(simlist,simlist$currtime+srvduration,simlist$repairevnt,
            qhead[3:4])
      }
   }
}
```