Biostat gyakorlatok

Sólymos Péter

2019-10-04

Eltávolításos mintavétel

Előkészületek

```
library(bSims)
```

```
## Loading required package: intrval
## Loading required package: mefa4
## Loading required package: Matrix
## mefa4 0.3-6   2019-06-20
## Loading required package: MASS
## Loading required package: deldir
## deldir 0.1-16
## bSims 0.1-3   2019-09-18      chrrr-chrrr-chrrr
```

```
library(detect)
```

```
## Loading required package: Formula
## Loading required package: stats4
## Loading required package: pbapply
## detect 0.4-2      2018-08-29
```

```
library(Distance)
```

```
## Loading required package: mrds
## This is mrds 2.2.0
## Built: R 3.6.0; ; 2019-04-27 00:52:54 UTC; unix
##
## Attaching package: 'Distance'
## The following object is masked from 'package:mrds':
##
##     create.bins
```

```
load("../_data/josm/josm.rda")
source("../functions.R")
```

Tájkép létrehozása

```
(l <- bsims_init(extent=10))
```

```
## bSims landscape
##   1 km x 1 km
##   stratification: H
```

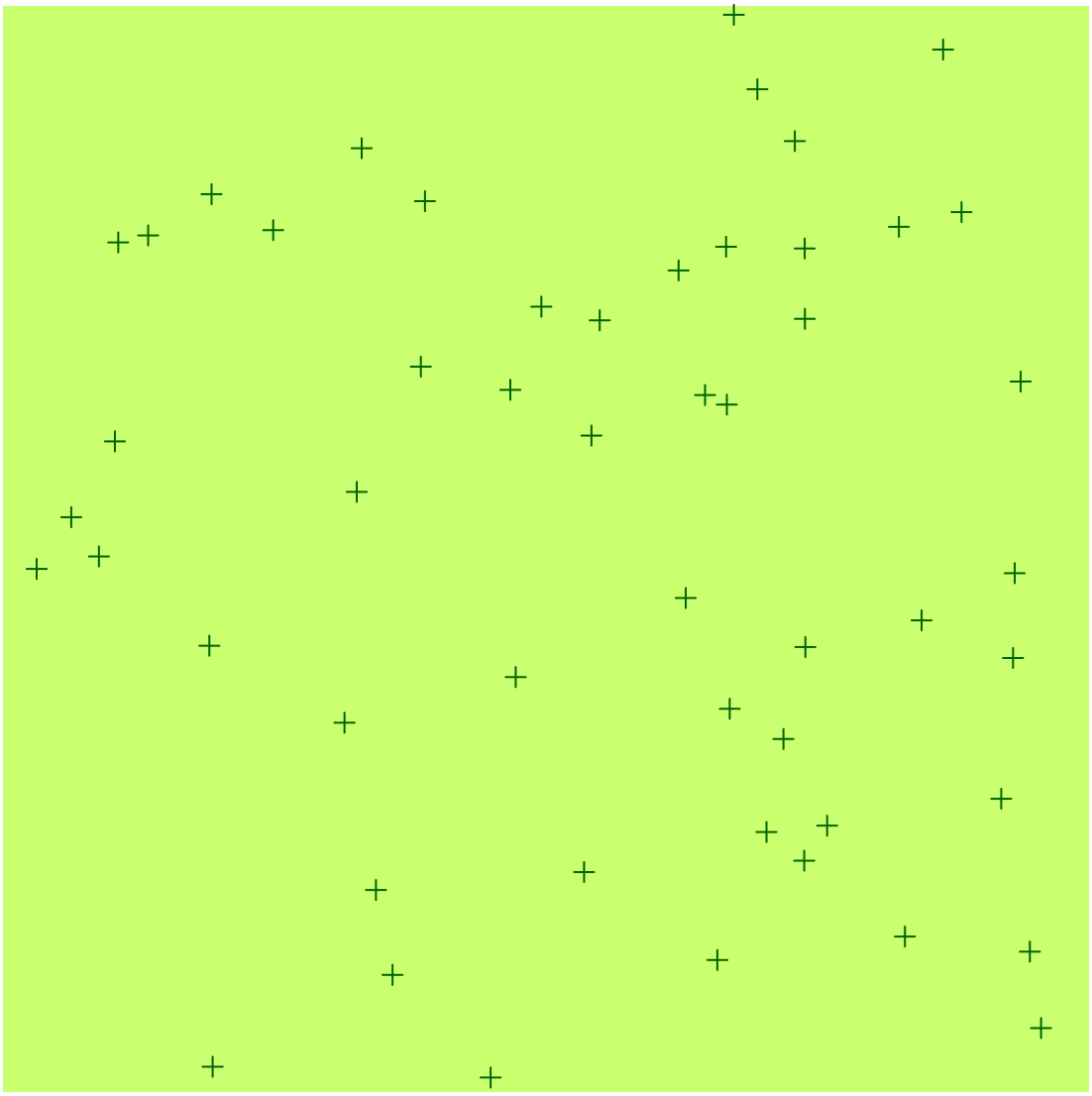
```
plot(l)
```


100 ha területen helyezzünk el átlagban 0.5 madarat hektáronként, Poisson térbeli folyamat, a várható érték tehát 50

```
set.seed(1)
(a <- bsims_populate(l, density=0.5))
```

```
## bSims population
##   1 km x 1 km
##   stratification: H
##   total abundance: 52
```

```
plot(a)
```

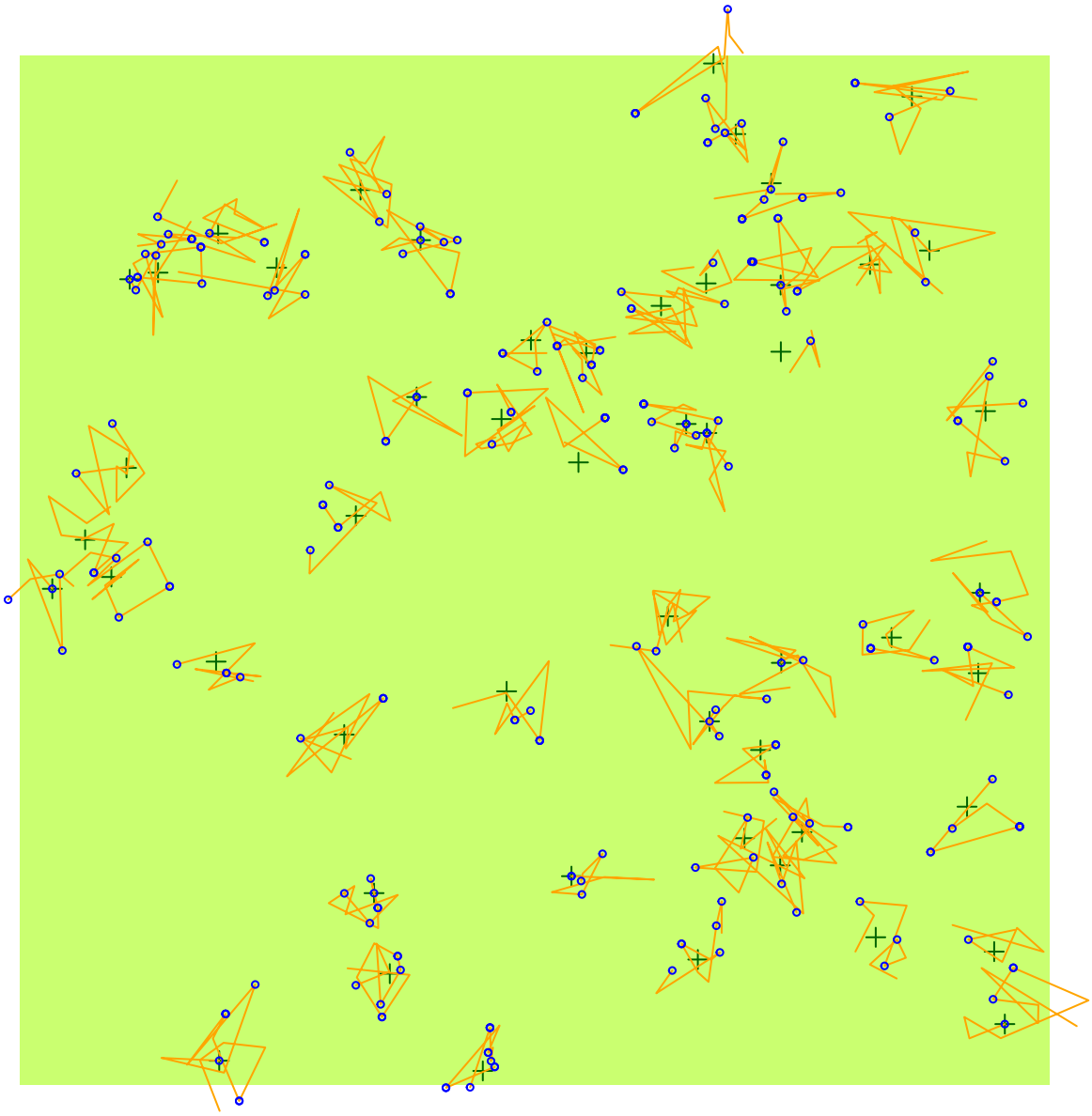


10 perc alatt, percenkénti 0.5-ös hangadási gyakoriséggal, némi mozgás hozzáadásával (1 per perc, 2D normál kernel izotróp 25m-es standard hibával)

```
(b <- bsims_animate(a,
  vocal_rate=0.5, duration=10,
  move_rate=1, movement=0.25))
```

```
## bSims events
##   1 km x 1 km
##   stratification: H
##   total abundance: 52
##   duration: 10 min
```

```
plot(b)
```



Vizsgáljuk meg a vokális eseményeket:

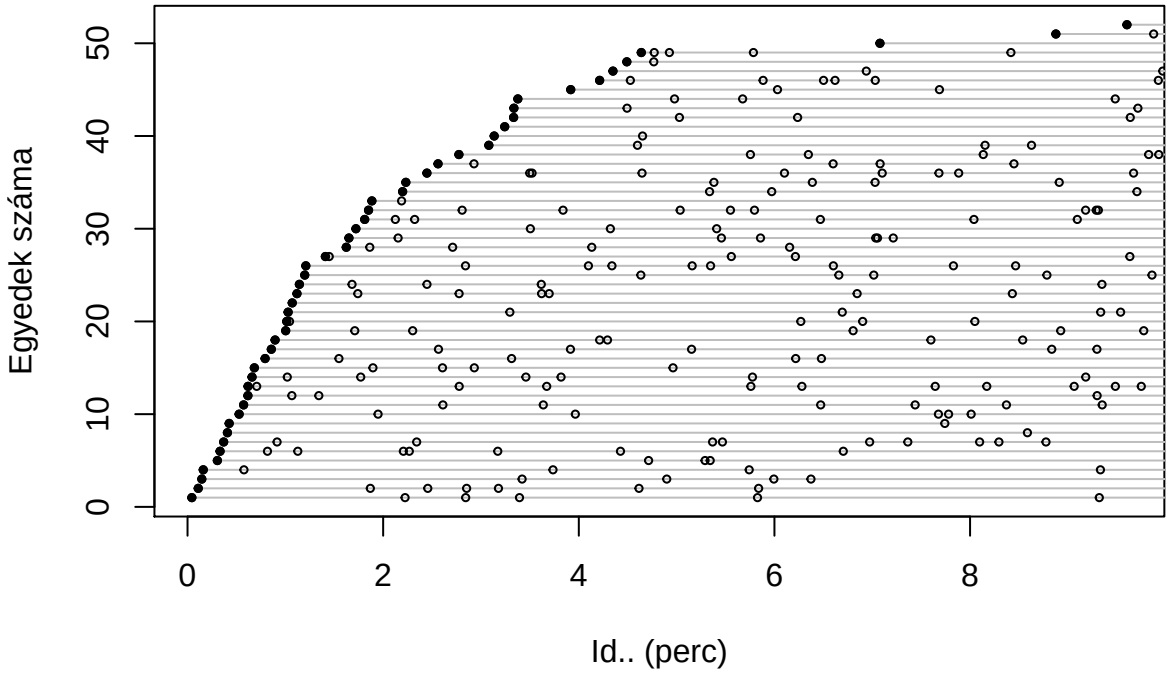
```
v <- get_events(b, event_type="vocal")
head(v)
```

```
##            x         y          t v  i
## 3   2.386150  2.773207 0.04400409 1 22
## 6  -3.069430 -4.766688 0.10948147 1 21
## 8   4.599685 -1.209372 0.14586603 1 26
## 9   1.695365 -1.468027 0.16067010 1 45
## 20  4.564380 -4.410656 0.30507246 1 35
## 23 -3.863095  2.849075 0.33254314 1 20
```

```
plot(v, xlab="Idő (perc)", ylab="Egyedek száma")
```

```
## Warning in title(...): conversion failure on 'Idő (perc)' in 'mbcsToSbcs':
## dot substituted for <c5>
```

```
## Warning in title(...): conversion failure on 'Idő (perc)' in 'mbcsToSbcs':
## dot substituted for <91>
```



Túlélési modell sűrűségfüggvény: Exponenciális eloszlás, $f(t) = \phi e^{-t\phi}$

```
(phi <- b$vocal_rate[1])
```

```
## [1] 0.5
```

```
v1 <- v[!duplicated(v$i),] # 1st detections
(phi_hat <- fitdistr(v1$t, "exponential")$estimate)
```

```
##      rate
## 0.4808192
```

```
hist(v1$t, xlab="Első detektálásig eltelt idő (perc)", freq=FALSE, main="",
  col="lightgrey", ylab="f(t)")
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Első detektálásig eltelt idő (perc)' in
## 'mbcsToSbcs': dot substituted for <c5>
```

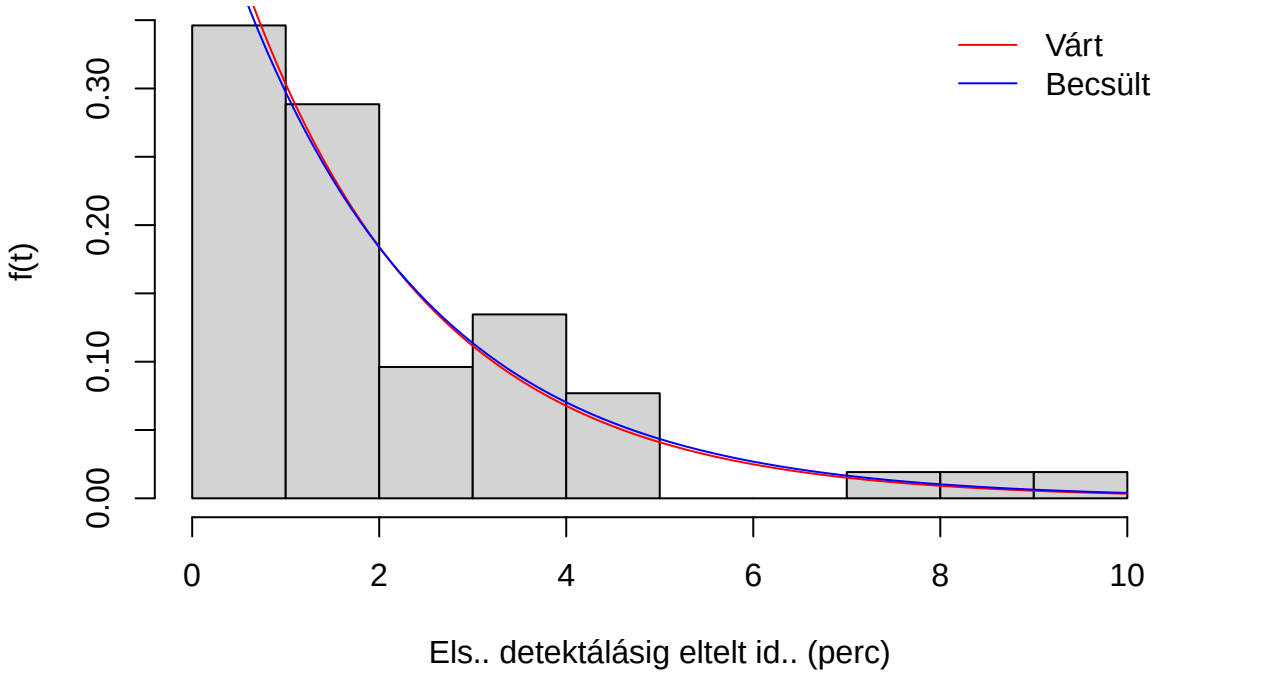
```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Első detektálásig eltelt idő (perc)' in
## 'mbcsToSbcs': dot substituted for <91>
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Első detektálásig eltelt idő (perc)' in
## 'mbcsToSbcs': dot substituted for <c5>
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on 'Első detektálásig eltelt idő (perc)' in
## 'mbcsToSbcs': dot substituted for <91>
```

```
curve(dexp(x, phi), add=TRUE, col=2)
curve(dexp(x, phi_hat), add=TRUE, col=4)
```

```
legend("topright", bty="n", lty=1, col=c(2,4),
  legend=c("Várt", "Becsült"))
```



Kumulatív sűrűségfüggvény: megadja várhatóan mennyi esemény következik be t idő alatt, $F(t) = \int_0^t f(t)dt = 1 - e^{-t\phi} = p_t$

```
br <- c(3, 5, 10)
i <- cut(v1$t, c(0, br), include.lowest = TRUE)
table(i)
```

```
## i
##  [0,3]  (3,5] (5,10]
##     38     11      3
```

```
plot(stepfun(v1$t, (0:nrow(v1))/nrow(v1)), do.points=FALSE, xlim=c(0,10),
  xlab="Első detektálásig eltelt idő (perc)", ylab="F(t)", main="")
```

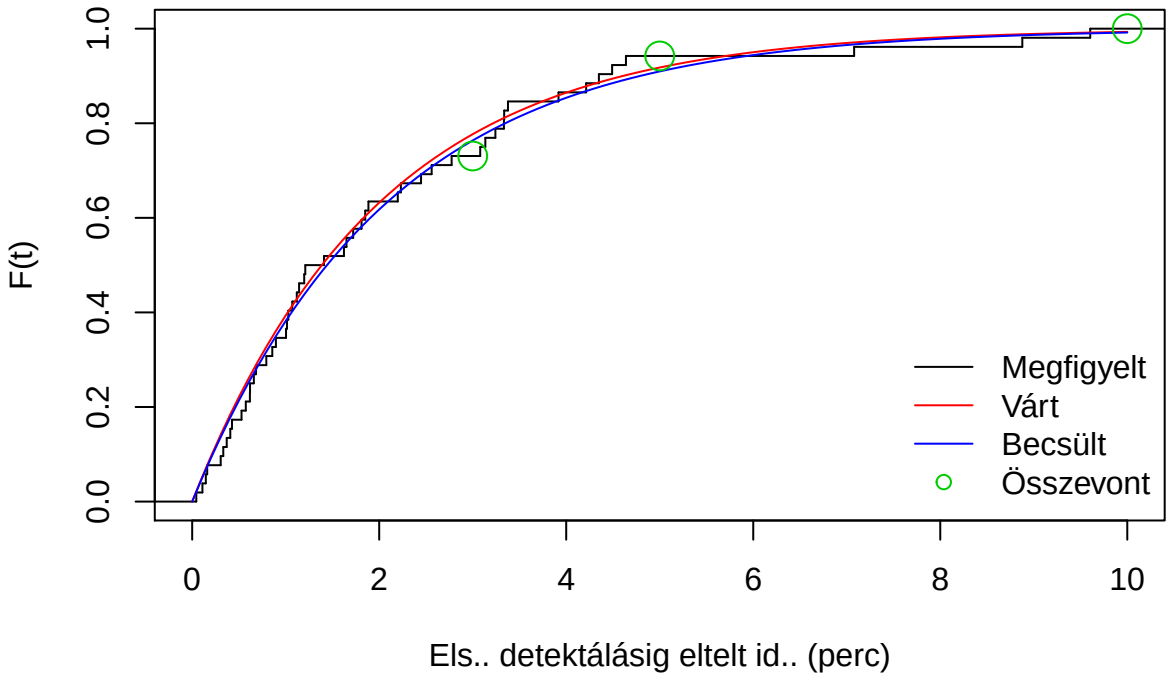
```
## Warning in title(...): conversion failure on 'Első detektálásig eltelt idő
## (perc)' in 'mbcsToSbcs': dot substituted for <c5>
```

```
## Warning in title(...): conversion failure on 'Első detektálásig eltelt idő
## (perc)' in 'mbcsToSbcs': dot substituted for <91>
```

```
## Warning in title(...): conversion failure on 'Első detektálásig eltelt idő
## (perc)' in 'mbcsToSbcs': dot substituted for <c5>
```

```
## Warning in title(...): conversion failure on 'Első detektálásig eltelt idő
## (perc)' in 'mbcsToSbcs': dot substituted for <91>
```

```
curve(1-exp(-phi*x), add=TRUE, col=2)
curve(1-exp(-phi_hat*x), add=TRUE, col=4)
legend("bottomright", bty="n", lty=c(1,1,1,NA),
  col=c(1,2,4,3), pch=c(NA,NA,NA,21),
  legend=c("Megfigyelt", "Várt", "Becsült", "Összevont"))
points(br, cumsum(table(i))/sum(table(i)), cex=2, col=3, pch=21)
```



Eltávolításos mintavétel: muntinomiális független változó az összevont kumulált adatokkal, adott időintervallumokban megfigyelt új egyedek száma

```
(y <- matrix(as.numeric(table(i)), nrow=1))
```

```
##      [,1] [,2] [,3]
## [1,]   38   11    3
```

```
(d <- matrix(br, nrow=1))
```

```
##      [,1] [,2] [,3]
## [1,]    3    5   10
```

```
(phi_hat1 <- exp(cmulti.fit(y, d, type="rem")$coef))
```

```
## [1] 0.4682625
```

```
phi # setting
```

```
## [1] 0.5
```

```
phi_hat # from time-to-event data
```

```
##      rate
## 0.4808192
```

Valódi pontszámlás adatok elemzése

```
yall <- Xtab(~ SiteID + Dur + SpeciesID,
  josm$counts[josm$counts$DetectType1 != "V",])
yall <- yall[sapply(yall, function(z) sum(rowSums(z) > 0)) > 100]

spp <- "TEWA"

Y <- as.matrix(yall[[spp]])
D <- matrix(c(3, 5, 10), nrow(Y), 3, byrow=TRUE,
  dimnames=dimnames(Y))
```

```r
head(Y[rowSums(Y) > 0,])
```

```
##         0-3min 3-5min 5-10min
## CL10106      4      0       0
## CL10112      2      0       0
## CL10120      1      1       0
## CL10170      1      0       0
## CL10172      0      0       2
## CL10181      0      0       1
```

```r
head(D)
```

```
##         0-3min 3-5min 5-10min
## CL10102      3      5      10
## CL10106      3      5      10
## CL10108      3      5      10
## CL10109      3      5      10
## CL10111      3      5      10
## CL10112      3      5      10
```

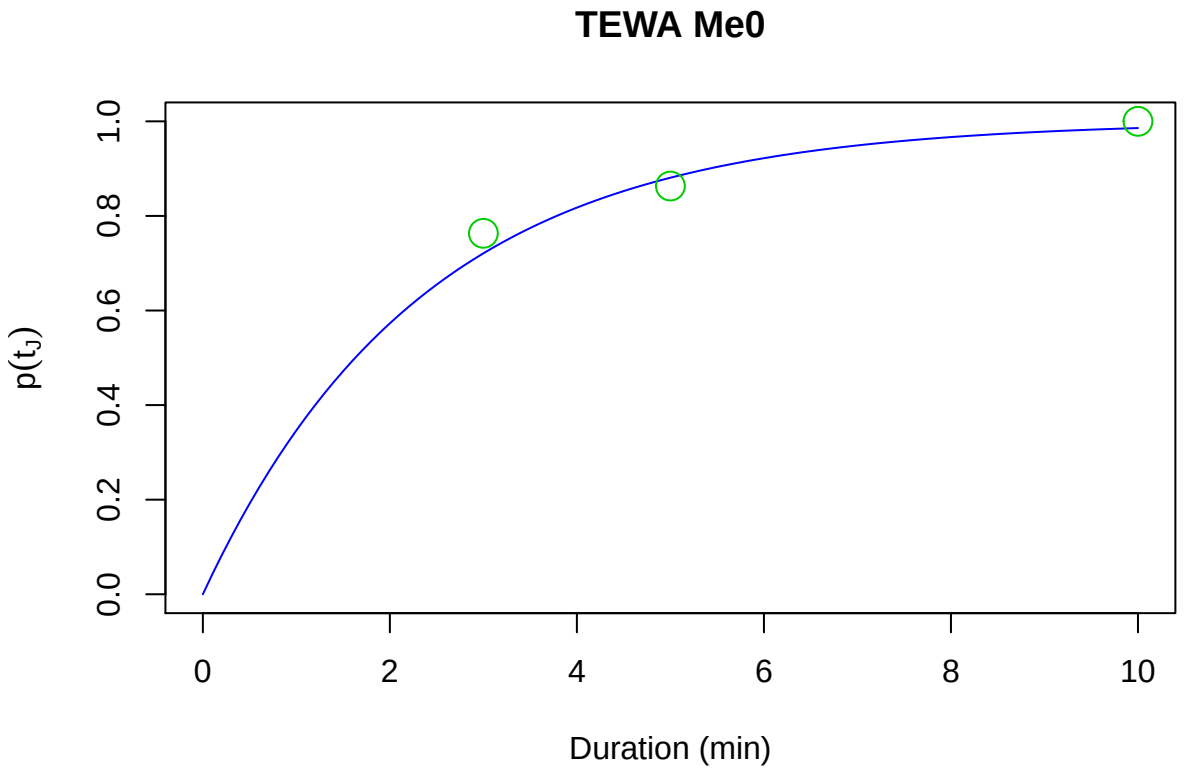
```r
Me0 <- cmulti(Y | D ~ 1, type="rem")
summary(Me0)
```

```
##
## Call:
## cmulti(formula = Y | D ~ 1, type = "rem")
##
## Removal Sampling (homogeneous singing rate)
## Conditional Maximum Likelihood estimates
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## log.phi_(Intercept) -0.85470    0.01739  -49.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -3205
## BIC =  6418
```

```r
(phi_Me0 <- exp(coef(Me0)))
```

```
## log.phi_(Intercept)
##             0.42541
```

```r
curve(1-exp(-x*phi_Me0), xlim=c(0, 10), ylim=c(0, 1), col=4,
  xlab="Duration (min)", ylab=expression(p(t[J])),
  main=paste(spp, "Me0"))
points(D[1,], cumsum(colSums(Y))/sum(Y), cex=2, col=3, pch=21)
```



Távolság becslés

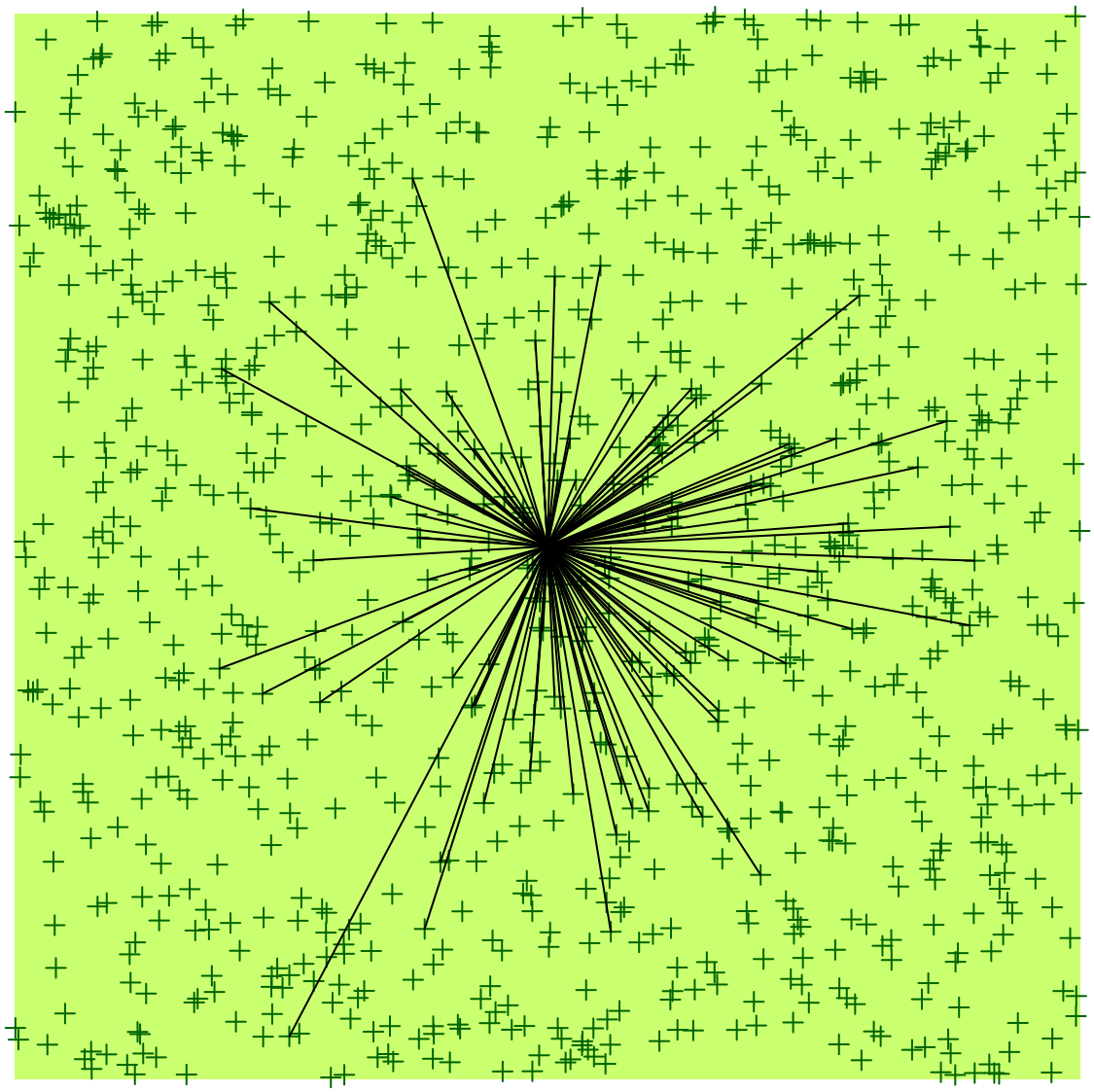
Távolság függvény: a detektálási valószínűség a távolsággal monoton csökken, $g(0) = 1$ azaz a megfigyelő közvetlen közelében a valószínűség 1. "Fél-normál" $g(d) = e^{-(d/\tau)^2}$, $\tau^2/2$ a varianciája.

```
shiny::runApp(system.file("shiny/distfunH.R", package="bSims"))
```

```
tau <- 2

set.seed(123)
l <- bsims_init()
a <- bsims_populate(l, density=10)
b <- bsims_animate(a, initial_location=TRUE)

(o <- bsims_detect(b, tau=tau))
```

```
## bSims detections
##   1 km x 1 km
##   stratification: H
##   total abundance: 1013
##   no events, duration: 10 min
##   detected: 128 seen/heard
```

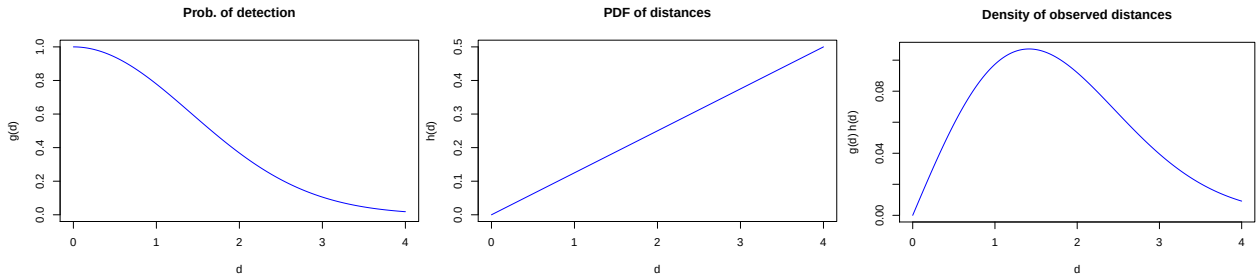
```
plot(o)
```



A megfigyelt távolságok gyakoriság eloszlása ($g(d)h(d)$) függ: a távolság függvénytől, és a különböző távolságú pontok gyakoriságától ami pontszámlálás esetén $h(d) = \pi 2d/A = \pi 2d/\pi r_{max}^2 = 2d/r_{max}^2$

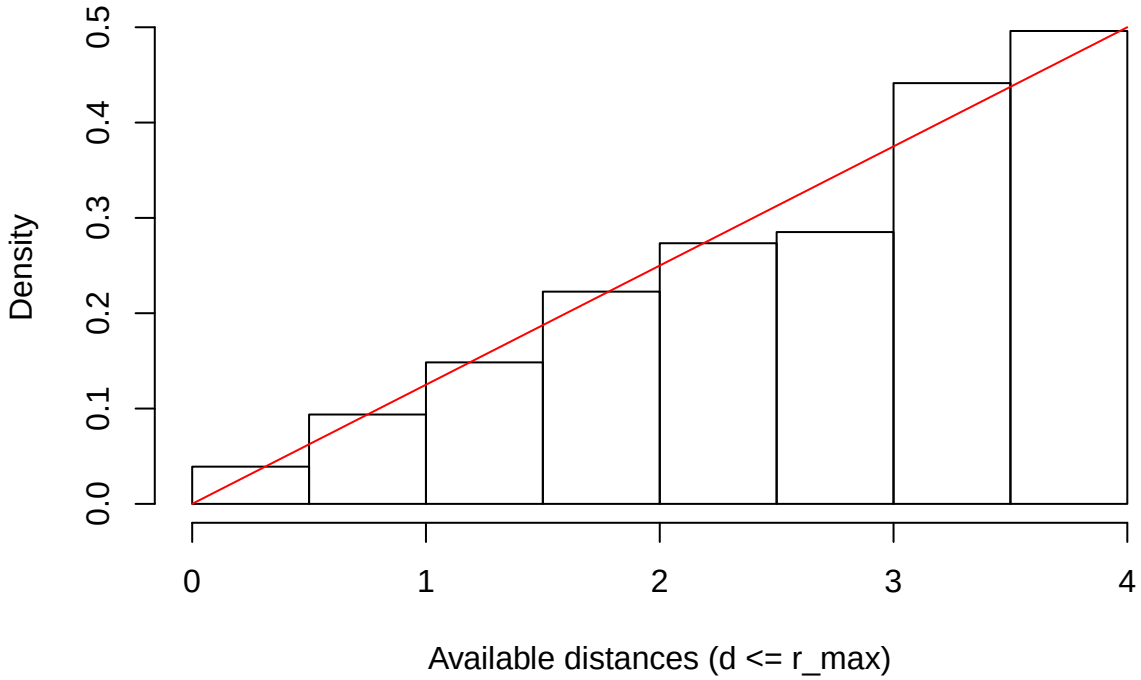
```
g <- function(d, tau, b=2, hazard=FALSE)
  if (hazard)
    1-exp(-(d/tau)^-b) else exp(-(d/tau)^b)
h <- function(d, rmax)
  2*d/rmax^2
```

```
rmax <- 4

d <- seq(0, rmax, 0.01)
plot(d, g(d, tau), type="l", col=4, ylim=c(0,1),
  xlab="d", ylab="g(d)", main="Prob. of detection")
plot(d, h(d, rmax), type="l", col=4,
  xlab="d", ylab="h(d)", main="PDF of distances")
plot(d, g(d, tau) * h(d, rmax), type="l", col=4,
  xlab="d", ylab="g(d) h(d)", main="Density of observed distances")
```



da a fészkek megfigyelőtől vett távolságát adja

```
da <- sqrt(rowSums(a$nests[,c("x", "y")]^2))

hist(da[da <= rmax], freq=FALSE, xlim=c(0, rmax),
  xlab="Available distances (d <= r_max)", main="")
curve(2*x/rmax^2, add=TRUE, col=2)
```



A megfigyelt távolságokat így kapjuk meg

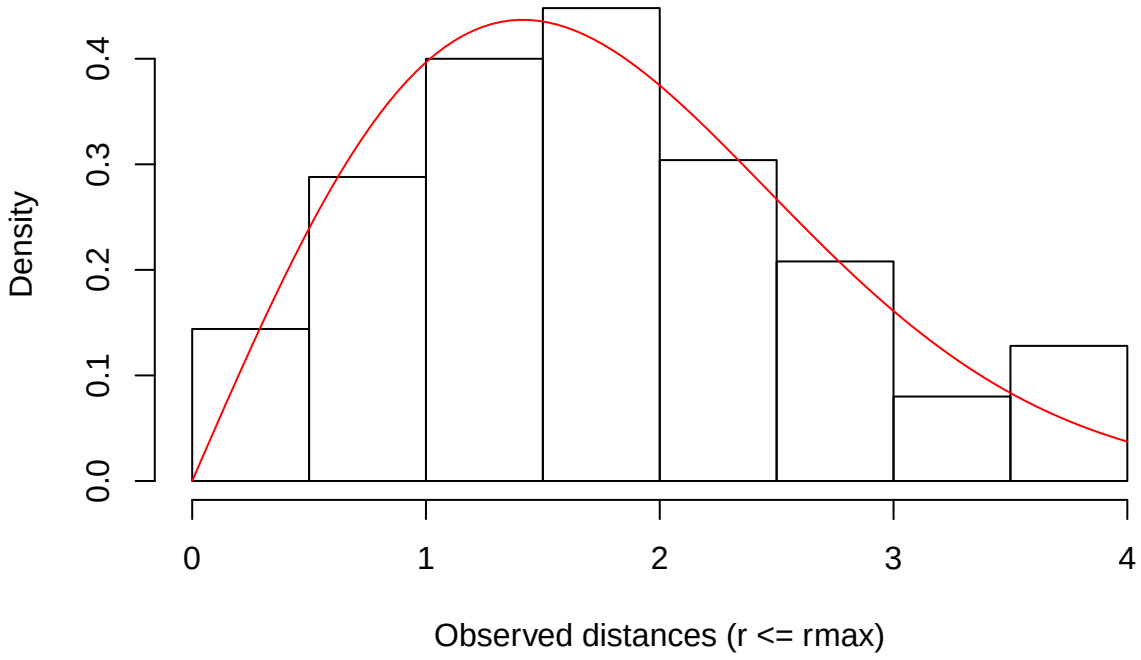
```
head(dt <- get_detections(o))
```

```
##              x          y t v         d  i  j
## 9  -2.13683299 -1.4639392 0 0 2.5902072  9  9
## 15 -0.22748175  0.3605372 0 0 0.4263039 15 15
## 19 -0.02865112 -0.3528623 0 0 0.3540235 19 19
## 45 -0.71086142 -1.5128051 0 0 1.6714973 45 45
## 47 -2.78899963  0.3554139 0 0 2.8115544 47 47
## 58 -3.08015181 -1.1482640 0 0 3.2872246 58 58
```

A megfigyelt távolságok valószínűségi eloszlása: a sűrűségfüggvény integrállal standardizált változata

```
f <- function(d, tau, b=2, hazard=FALSE, rmax=1)
  g(d, tau, b, hazard) * h(d, rmax)
tot <- integrate(f, lower=0, upper=rmax, tau=tau, rmax=rmax)$value
```

```
hist(dt$d[dt$d <= rmax], freq=FALSE, xlim=c(0, rmax),
  xlab="Observed distances (r <= rmax)", main="")
curve(f(x, tau=tau, rmax=rmax) / tot, add=TRUE, col=2)
```



Ha mind a magfigyelt és nem megfigyelt egyedek távolságát tudnánk akkor könnyű dolgunk lenne, mert a fél-normál távolság függvény könnyen linearizálhatjuk, $log(g(d)) = log(e^{-(d/\tau)^2}) = -(d/\tau)^2 = x\frac{1}{\tau^2} = 0 + x\beta$, azaz GLM-mel becsülhetjük a τ értékét: $x = -d^2$, $\hat{\tau} = \sqrt{1/\hat{\beta}}$.

```
dat <- data.frame(
  distance=da,
  x=-da^2,
  detected=ifelse(rownames(o$nests) %in% dt$i, 1, 0))
summary(dat)
```

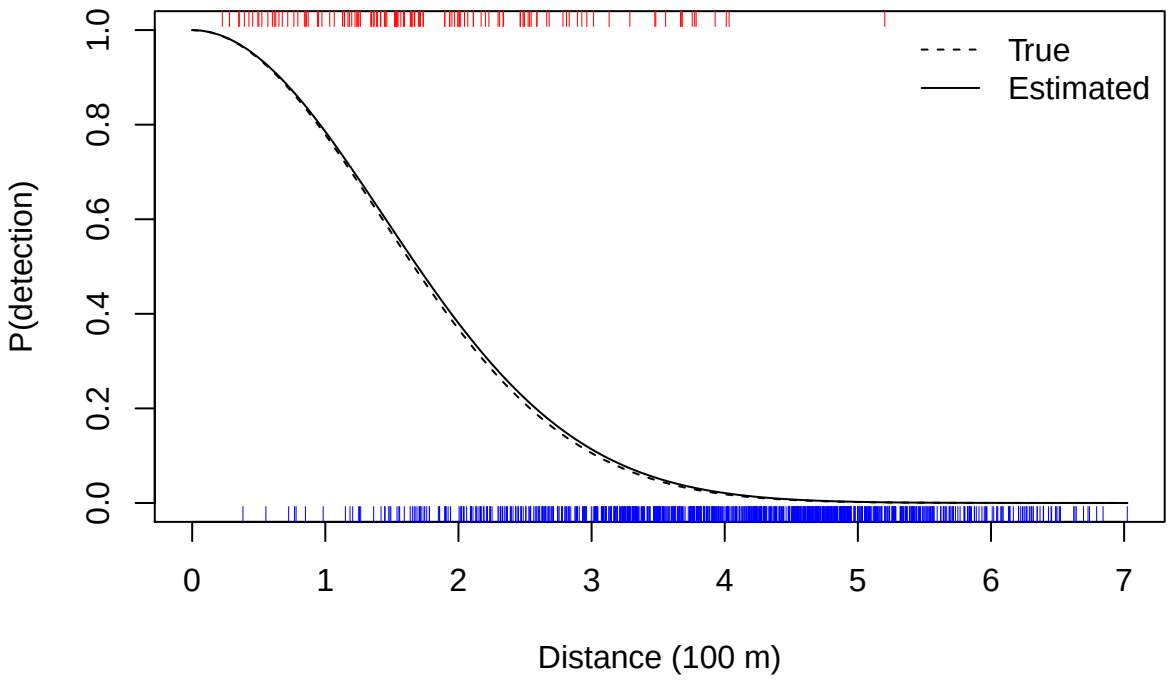
```
##     distance            x                detected
##  Min.   :0.2264   Min.   :-49.33230   Min.   :0.0000
##  1st Qu.:2.8758   1st Qu.:-23.73953   1st Qu.:0.0000
##  Median :3.9653   Median :-15.72398   Median :0.0000
##  Mean   :3.8274   Mean   :-16.68895   Mean   :0.1264
##  3rd Qu.:4.8723   3rd Qu.: -8.27033   3rd Qu.:0.0000
##  Max.   :7.0237   Max.   : -0.05125   Max.   :1.0000
```

```
mod <- glm(detected ~ x - 1, data=dat, family=binomial(link="log"))
c(true=tau, estimate=sqrt(1/coef(mod)))
```

```
##       true estimate.x
##   2.000000   2.034018
```

```
curve(exp(-(x/sqrt(1/coef(mod)))^2),
  xlim=c(0,max(dat$distance)), ylim=c(0,1),
  xlab="Distance (100 m)", ylab="P(detection)")
curve(exp(-(x/tau)^2), lty=2, add=TRUE)
rug(dat$distance[dat$detected == 0], side=1, col=4)
rug(dat$distance[dat$detected == 1], side=3, col=2)
```

```
legend("topright", bty="n", lty=c(2,1),
  legend=c("True", "Estimated"))
```



A valóségban azonban csak a megfigyelt egyedek távolságát ismerjük Az következő link részletezi a távolság függvény illesztését. Itt most a fél-normálra koncentrálunk (`key = "hn"`) mindenféle egyéb igazítás nélkül (`adjustment=NULL`). A program τ négyzetgyökének logaritmusát becsli:

```
dd <- ds(dt$d, truncation = rmax, transect="point",
  key = "hn", adjustment=NULL)
```

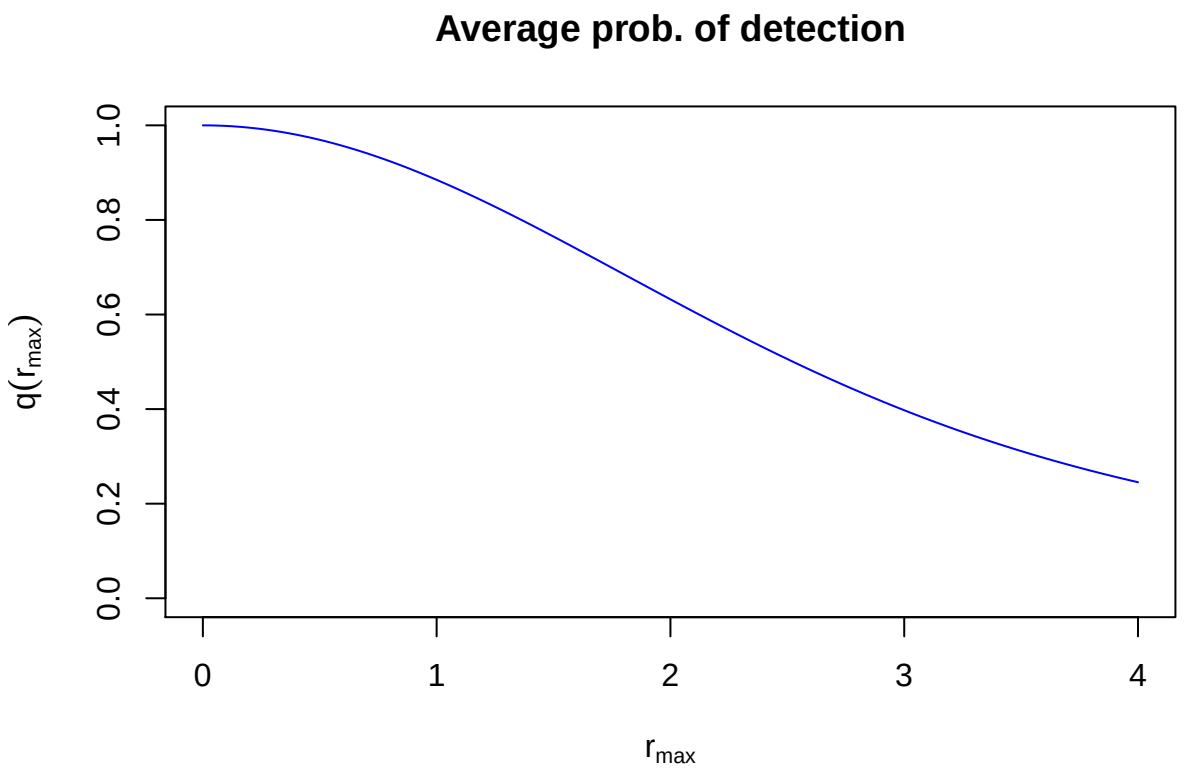
```
## Fitting half-normal key function

## Key only model: not constraining for monotonicity.

## AIC= 315.502

## No survey area information supplied, only estimating detection function.
```

```
c(true=tau, estimate=exp(dd$ddf$par)^2)
```

```
##     true estimate
## 2.000000 2.175567
```

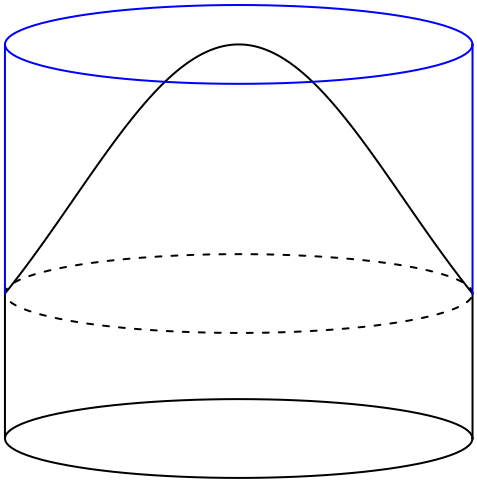
Átlagos detektálhatóság: amikor az r_{max} távolságig integrálunk: $q(r_{max}) = \int_0^{r_{max}} g(d)h(d)dd$, ami a következőképpen szemléltethető (a "levágott tészta" aránya a henger térfogatához képest, πr_{max}^2)

```
q <- sapply(d[d > 0], function(z)
  integrate(f, lower=0, upper=z, tau=tau, rmax=z)$value)

plot(d, c(1, q), type="l", col=4, ylim=c(0,1),
  xlab=expression(r[max]), ylab=expression(q(r[max])),
  main="Average prob. of detection")
```



Amit analitikus formában is megkaphatunk: $\pi\tau^2[1 - exp(-d^2/\tau^2)]/(\pi r_{max}^2)$

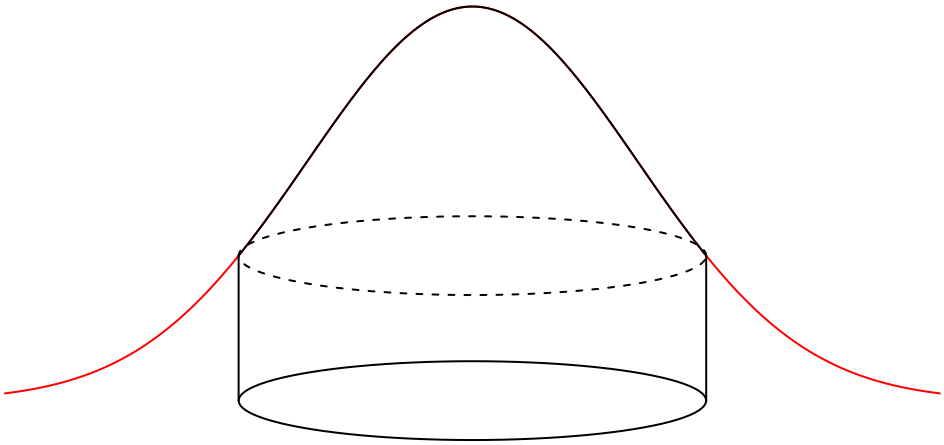
```
tau <- 2
rmax <- 2
w <- 0.1
m <- 2
plot(0, type="n", xlim=m*c(-rmax, rmax), ylim=c(-w, 1+w),
  axes=FALSE, ann=FALSE)
yh <- g(rmax, tau=tau)
lines(seq(-rmax, rmax, rmax/100),
  g(abs(seq(-rmax, rmax, rmax/100)), tau=tau))
draw_ellipse(0, yh, rmax, w, lty=2)
lines(-c(rmax, rmax), c(0, yh))
lines(c(rmax, rmax), c(0, yh))
draw_ellipse(0, 0, rmax, w)
draw_ellipse(0, 1, rmax, w, border=4)
lines(-c(rmax, rmax), c(yh, 1), col=4)
lines(c(rmax, rmax), c(yh, 1), col=4)
```



A pontszámláláskor elég nehézkes a távolság becslése ezért gyakran távolság intervallumokat használnak

A kumulatív valószínűségi függvény használható ebben az esetben a multinomiális celle gyakoriságok számítására $\pi(r) = 1 - e^{-(r/\tau)^2}$ (ezt az integrál térfogatával kell normalizálni, ami $\pi\tau^2$). Ez a "levágott tészta" térfogatát adja meg az összes tészta térfogatához képest.

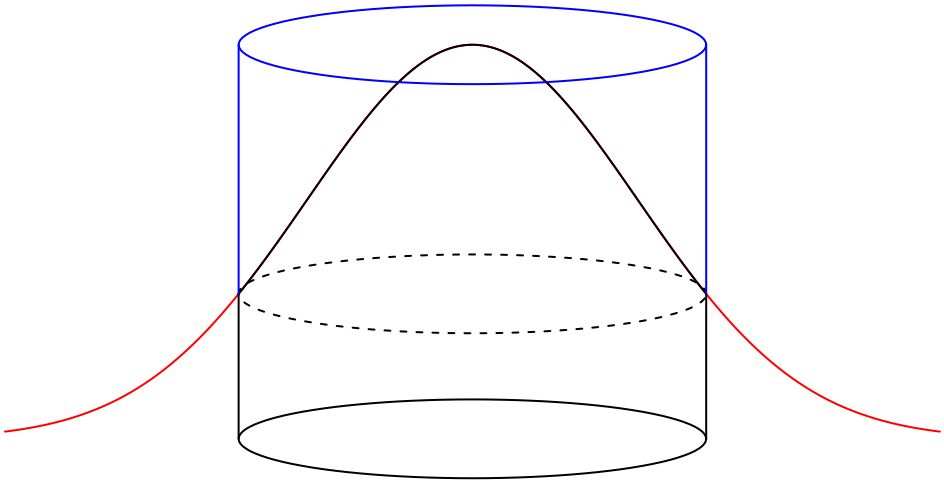
```
plot(0, type="n", xlim=m*c(-rmax, rmax), ylim=c(-w, 1+w),
  axes=FALSE, ann=FALSE)
yh <- g(rmax, tau=tau)
lines(seq(-m*rmax, m*rmax, rmax/(m*100)),
  g(seq(-m*rmax, m*rmax, rmax/(m*100)), tau=tau),
  col=2)
lines(seq(-rmax, rmax, rmax/100),
  g(abs(seq(-rmax, rmax, rmax/100)), tau=tau))
draw_ellipse(0, yh, rmax, w, lty=2)
lines(-c(rmax, rmax), c(0, yh))
lines(c(rmax, rmax), c(0, yh))
draw_ellipse(0, 0, rmax, w)
```



Az ún. effektív detektálási távolság éppen τ a fél-normál távolság függvény esetén, azaz a távolságon kívül megfigyelt egyedek aránya megegyezik a tévolségon belül nem detektált egyedek arányával

```
plot(0, type="n", xlim=m*c(-rmax, rmax), ylim=c(-w, 1+w),
  axes=FALSE, ann=FALSE)
yh <- g(rmax, tau=tau)
lines(seq(-m*rmax, m*rmax, rmax/(m*100)),
```

```
  g(seq(-m*rmax, m*rmax, rmax/(m*100)), tau=tau),
  col=2)
lines(seq(-rmax, rmax, rmax/100),
  g(abs(seq(-rmax, rmax, rmax/100)), tau=tau))
draw_ellipse(0, yh, rmax, w, lty=2)
lines(-c(rmax, rmax), c(0, yh))
lines(c(rmax, rmax), c(0, yh))
draw_ellipse(0, 0, rmax, w)
draw_ellipse(0, 1, rmax, w, border=4)
lines(-c(rmax, rmax), c(yh, 1), col=4)
lines(c(rmax, rmax), c(yh, 1), col=4)
```



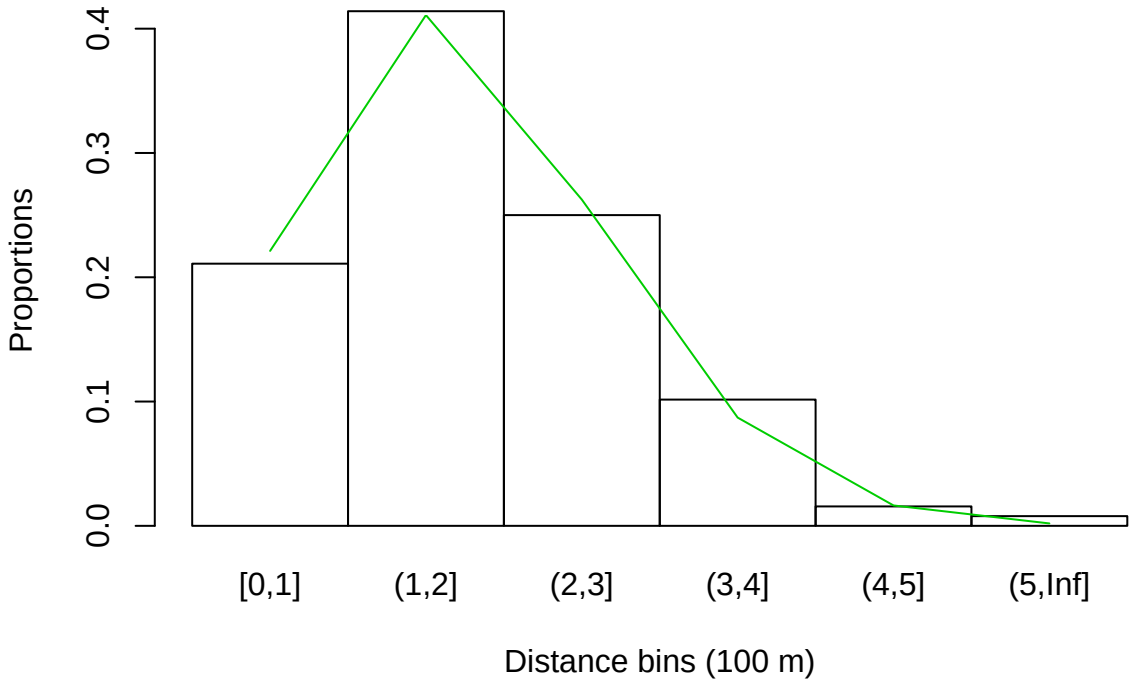
Miért jó ez nekünk? Mert így becsülni tudjuk az effektív mintavételi területet abban az esetben ha nem véges távolságon belül számlálunk (és ez gyakran előfordul).

Az összevont adatokkal a következő képpen dolgozunk

```
br <- c(1, 2, 3, 4, 5, Inf)
dat$bin <- cut(da, c(0, br), include.lowest = TRUE)
(counts <- with(dat, table(bin, detected)))
```

```
##          detected
## bin         0   1
##   [0,1]     7  27
##   (1,2]    42  53
##   (2,3]   111  32
##   (3,4]   227  13
##   (4,5]   287   2
##   (5,Inf] 211   1
```

```
pi_br <- 1-exp(-(br/tau)^2)

barplot(counts[,"1"]/sum(counts[,"1"]), space=0, col=NA,
  xlab="Distance bins (100 m)", ylab="Proportions",
  ylim=c(0, max(diff(c(0, pi_br)))))
lines(seq_len(length(br))-0.5, diff(c(0, pi_br)), col=3)
```




```
(tr <- bsims_transcribe(o, rint=br))
```

```
## bSims transcript
##   1 km x 1 km
##   stratification: H
##   total abundance: 1013
##   no events, duration: 10 min
##   detected: 128 seen/heard
##   1st event detected by bins:
##     [0-10 min]
##     [0-100, 100-200, 200-300, 300-400, 400-500, 500+ m]
```

```
tr$removal
```

```
##          0-10min
## 0-100m        27
## 100-200m      53
## 200-300m      32
## 300-400m      13
## 400-500m       2
## 500+m          1
```

```
Y <- matrix(drop(tr$removal), nrow=1)
D <- matrix(br, nrow=1)

tauhat <- exp(cmulti.fit(Y, D, type="dis")$coef)

c(true=tau, estimate=tauhat)
```

```
##     true estimate
## 2.000000 2.067061
```

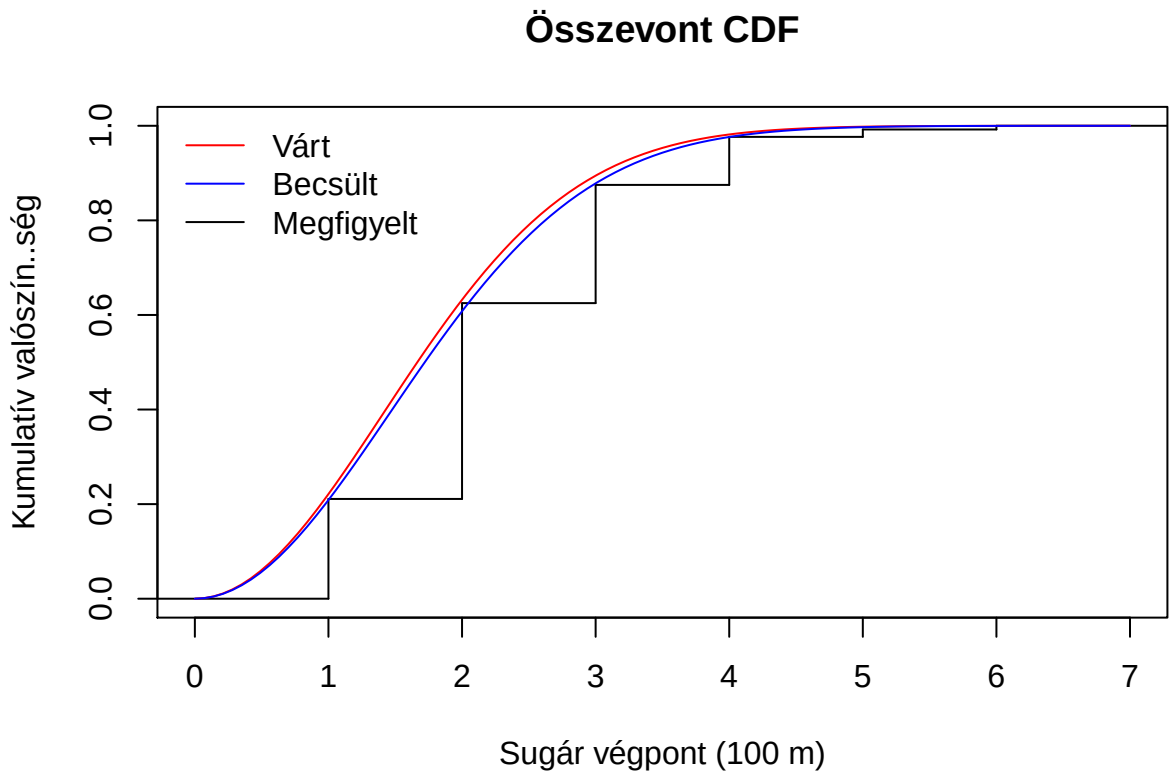
Kumulatív adatok

```
plot(stepfun(1:6, c(0, cumsum(counts[,"1"])/sum(counts[,"1"]))),
  do.points=FALSE, main="Összevont CDF",
  ylab="Kumulatív valószínűség",
  xlab="Sugár végpont (100 m)")
```

```
## Warning in title(...): conversion failure on 'Kumulatív valószínűség' in
## 'mbcsToSbcs': dot substituted for <c5>
```

```
## Warning in title(...): conversion failure on 'Kumulatív valószínűség' in
## 'mbcsToSbcs': dot substituted for <b1>
```

```
curve(1-exp(-(x/tau)^2), col=2, add=TRUE)
curve(1-exp(-(x/tauhat)^2), col=4, add=TRUE)
legend("topleft", bty="n", lty=1, col=c(2, 4, 1),
  legend=c("Várt", "Becsült", "Megfigyelt"))
```



A két folyamat együtt

Ha megkaptuk a következő két feltételes valószínűséget, akkor meg tudjuk becsülni a populáció sűrűségét:

- az egyed észrevehetővé válik, feltéve h. jelen van (p),
- az egyedet detektáljuk, eltéve h. észrevehető (q).

$\hat{C} = \hat{p}\hat{q}$, tehát $\hat{D} = E[Y]/\hat{p}\hat{q}$,

vagy trunkálatlan távolságok esetén $\hat{C} = \hat{p}\hat{A}$, tehát $\hat{D} = E[Y]/\hat{p}\hat{A}$.

Használjuk a Shiny appot

```
shiny::runApp(system.file("shiny/bsimsH.R", package="bSims"))
```

Aki további bonyodalmak felfedezésére vágyik, az a QPAD könyvben találhat érdekességeket (angol nyelven).

Ízelítő a tartalomból:

- adat manipuláció
- regressziós technikák
- amiről szó volt, de részletesebben
- autómata adatrögzítő technikák
- útmenti adatok problémaköre