Mosquito combinatorics (occupancy spectrum)

Ben Bolker, with help from Leonid Bogachev, Ethan Bolker, and Ira Gessel

November 1, 2017

Version: Wed Nov 1 16:05:39 2017

1 Preliminaries

```
library(rbenchmark)
library(plyr) ## for laply

## Warning:  package 'plyr' was built under R version 3.2.5

library(plotrix)

## Warning:  package 'plotrix' was built under R version 3.2.5

library(reshape2)

## Warning:  package 'reshape2' was built under R version 3.2.5

library(abind)

## Warning:  package 'abind' was built under R version 3.2.5

library(ggplot2); theme_set(theme_bw())

## Warning:  package 'ggplot2' was built under R version 3.2.5

  ## tweak to squash panels togethe
zmargin <- theme(panel.spacing=grid::unit(0,"lines"))
library(grid)
library(numDeriv)

## Warning:  package 'numDeriv' was built under R version 3.2.5
```

2 Introduction

Suppose B=number of birds (bins); M=number of mosquitoes (balls). V is the (unordered) occupancy of a particular configuration (e.g. $\{3,1,1,0\}$ means

one bird sampled three times, two birds sampled once, one bird not sampled); $S = \{s_i\}$=occupancy spectrum (number of birds sampled i times, e.g. $S = \{1, 2, 0, 1\}$ for the previous example). The number of birds sampled at least once is $K = B - s_0 = B - \sum_{i=1} s_i$.

We have $|V| = \sum s_i = B$; $\sum v_i = \sum i s_i = M$.

Define the multinomial coefficient $\mathcal{M}(S) \equiv \binom{(\sum s_i)!}{\prod s_i!}$ **c**heck? frac, not binom?

Then the likelihood of observing an occupancy spectrum S is

$$P(S|B, M) = \frac{1}{B^M} \mathcal{M}(S)\mathcal{M}(V) \tag{1}$$

where $\mathcal{M}(V)$ can also be written as $M!/\prod_i (i!)^{s_i}$.

These are standard *Maxwell-Boltzmann* statistics (as opposed to some of the previous formulae, which were essentially Einstein-Bose type, incorrectly [for this problem] treating some configurations as equivalently).

3 Code

3.1 Analytical formula

The following functions implement this idea (allowing for the possibility of returning the log-probability, and allowing the possibility of specifying the occupancy spectrum with the s_0 element excluded — and filling it in using $s_0 = B - K$).

```
## multinomial coefficient
mchoose <- function(n,log=FALSE) {
    m <- lfactorial(sum(n))-sum(lfactorial(n))
    if (log) m else exp(m)
}
occprob <- function(n,B,M,log=FALSE,add.zero=TRUE) {
    if (add.zero) nx <- c(B-sum(n),n) else {
        nx <- n; n <- n[-1]
    }
    r <- -M*log(B)+mchoose(nx,log=TRUE)+
        lfactorial(M)-sum(n*lfactorial(seq_along(n)))
    if (log) r else exp(r)
}
```

3.2 Simulation

I simulated one realization of the process by sampling birds with replacement and counting the number of occurrences of each bird.

This can be done using `sample()` and then tabulating the results:

```
vfun0 <- function(B,M) table(factor(sample(1:B,size=M,replace=TRUE),levels=1:B))
#Result is Vi -> unordered occupancy configuration
```

(if I omitted the `factor(...,levels=1:B)` statement I would get a table without zeros/unsampled birds included).

Equivalently one can draw a multinomial sample:

```
vfun <- function(B,M) { c(rmultinom(1,size=M,prob=rep(1,B))) }
```

It turns out the latter is much faster, as indicated by the following benchmark:

```
##              test replications elapsed relative
## 1  vfun(1000, 50)         1000    0.09    1.000
## 2 vfun0(1000, 50)         1000    1.02   11.333
```

We may want to collapse these samples to occupancy spectra, e.g.

```
set.seed(101)
B <- 7; M <- 5
v <- vfun(B,M)
table(factor(v,levels=0:M))

##
## 0 1 2 3 4 5
## 3 3 1 0 0 0
```

A function to tabulate S and optionally collapse the result to a dot-separated string:

```
sfun <- function(B,M,collapse=TRUE) {
    tt <- table(factor(vfun(B,M),levels=0:M))
    if (!collapse) tt else paste(tt,collapse=".")
}
```

3.3 Examples

Try this out for a trivial example ($B = 4$, $M = 2$).

```
B <- 4; M <- 2
S <- list(c(3,0,1),c(2,2,0))
sapply(S,occprob,B=B,M=M,add.zero=FALSE)

## [1] 0.25 0.75
```

Run 1000 simulations and tabulate:

```
r <- replicate(1000,sfun(B,M))
table(r)/1000

## r
## 2.2.0 3.0.1
## 0.749 0.251
```

These match.
Now a slightly larger example:

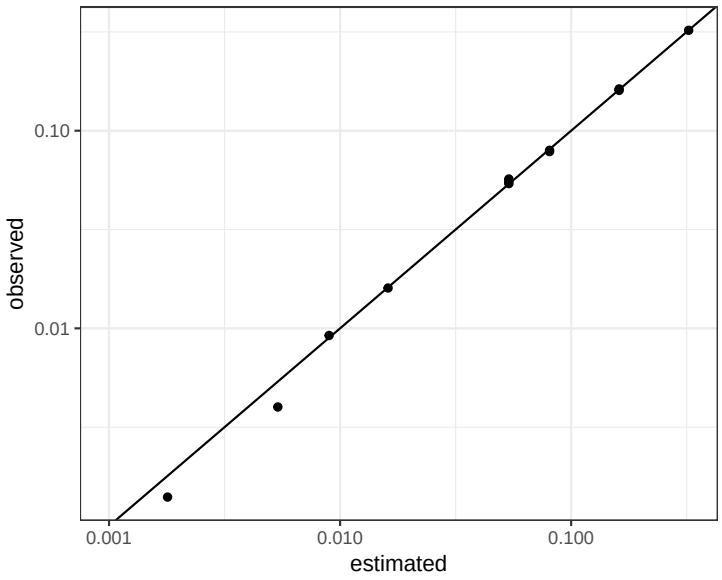
```
B <- 6; M <- 3
## enumerate possible occupancy spectra:
S <- list(c(3,3,0),c(4,1,1),c(5,0,0,1))
sapply(S,occprob,B=B,M=M,add.zero=FALSE)

## [1] 0.55555556 0.41666667 0.02777778
```

```
nsim <- 2000
table(replicate(nsim,sfun(B,M)))/nsim

##
## 3.3.0.0 4.1.1.0 5.0.0.1
##  0.5490  0.4215  0.0295
```

Pretty good match. One more try:

```
M <- 7; B <- 5
nsim <- 5000
(tt <- table(replicate(nsim,sfun(B,M)))/nsim)

##
## 0.3.2.0.0.0.0.0 0.4.0.1.0.0.0.0 1.1.3.0.0.0.0.0 1.2.1.1.0.0.0.0
##          0.1600          0.0570          0.1628          0.3222
## 1.3.0.0.1.0.0.0 2.0.2.1.0.0.0.0 2.1.0.2.0.0.0.0 2.1.1.0.1.0.0.0
##          0.0554          0.0796          0.0540          0.0784
## 2.2.0.0.0.1.0.0 3.0.0.1.1.0.0.0 3.0.1.0.0.1.0.0 3.1.0.0.0.0.1.0
##          0.0160          0.0092          0.0040          0.0014
```

```
## utility function: x.y.z format -> numeric vector
spec2num <- function(x) lapply(strsplit(x,"\\."),as.numeric)
## get occupancy spectra from names of sim table ...
S <- spec2num(names(tt))
est.p <- sapply(S,occprob,B=B,M=M,add.zero=FALSE)
```



4 Fitting

4.1 Preliminaries

```
source("mosqfuns2.R")
```

The only part of the log-likelihood that depends on B (or s_0, which implicitly depends on B via $s_0 = B - K$) is

$$-M \log B + \log B! - \log(B-K)!$$

In other words, K is a sufficient statistic for estimating B by maximum likelihood. (Alternatively, we can estimate B via the method of moments, since we can calculate an expected value for K: the answers turn out to be quite similar.)

We will try four different estimates: two depend (only) on K, one on the probability of doublets (W), one on the time to first collision τ_0. In each case we will compute the probability distribution of the summary statistic for $B = 40$, $M = 20$, calculate the estimated value of B for each value of the summary statistic, and calculate bias, variance, and mean-squared error for the estimate.

First simulate probability distributions of W, K, τ_0.

```
Kfun <- function(B,M) {
    unname(B-sfun(B,M,collapse=FALSE)[1])
}
```

Compute doublets:

```
Wfun <- function(B,M) {
    r <- vfun(B,M)
    mean(r*(r-1))
}
```

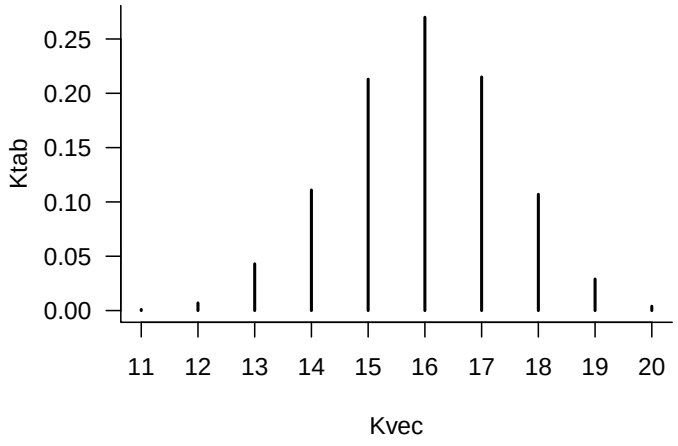
Compute time of first collision:

```
Cfun <- function(B,ssize=2*B) {
    ss <- sample(B,size=ssize,replace=TRUE)
    min(which(duplicated(ss)))
}
```

use set.seed()? fix downstream stuff so this part can be run out of order?

```
nsim <- 1000; M <- 20; B <- 40
Kvec <- replicate(nsim,Kfun(B,M))
Wvec <- replicate(nsim,Wfun(B,M))
Cvec <- replicate(nsim,Cfun(B))
```

```
Ktab <- table(Kvec)/nsim
plot(Ktab)
```




```
Kvec2 <- as.numeric(names(Ktab))
```

store SD and RMSE rather than var and MSE?

```
estres <- matrix(NA,nrow=4,ncol=3,
                 dimnames=list(c("MM","MLE","doublets","collision"),
                 c("bias","var","MSE")))
```

4.2 Method of moments

Based on binomial or Poisson approximations, we should have the expected value of K ($\hat{K}$) equal to approximately $B(1-(1-(1/B))^M)$, or (in another approximation, based on M, B both large) $B(1-\exp(-M/B))$ (or even more approximately) M: the last one recovers the case where $B \gg M$, so we expected each mosquito to bite a different bird ...

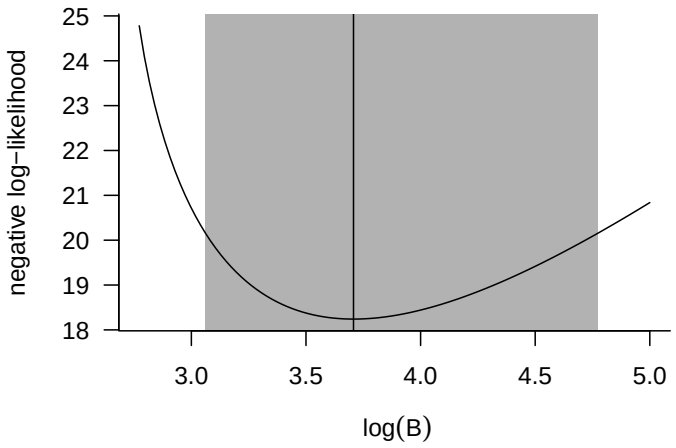
The `Bhat_approx` function implements a root-finding solution to find a method-of-moments estimate for B based on known K and M.

```
B_true <- 40
Bhat_mm <- sapply(Kvec2[Kvec2<20],Bhat_approx,M=20)
B_mean_mm <- sum(Bhat_mm*Ktab[Kvec2<20])
estres["MM","bias"] <- bias_mm <- B_mean_mm -B_true
estres["MM","var"] <- sum((Bhat_mm-B_mean_mm)^2*Ktab[Kvec2<20])
estres["MM","MSE"] <- sum((Bhat_mm-B_true)^2*Ktab[Kvec2<20])
```

4.3 MLE/likelihood ratio test

We can compute $P(B+1)/P(B)$ (and get a relatively simple expression that we try to equate to 1), or by brute force:

```
## Warning:  package 'bbmle' was built under R version 3.2.5
## Warning:  replacing previous import by 'stats::na.omit' when loading
'bbmle'
```



The gray region shows the 95% LRT confidence intervals (B such that $-\log L < -\log L_{\min} + 1.92$). Note we seem to have a computational problem for these functions when evaluating the log-likelihood for $\log B > 33$, because $K/B \ll 1$. However, we really should never be dealing with host population sizes as large as e^{33}=2.1464358 $\times$ 10^{14}!

```
## for lower bound when K=100
lboundfun <- function(M) {
    uniroot(function(x) nllfun(x,K=M,M=M)-1.92,
            interval=c(log(M+0.001),25))$root
}
ffun <- function(K,M) {
    if (K<M) {
        f <- try(fitB(K,M),silent=TRUE)
        if (inherits(f,"try-error")) rep(NA,3) else unlist(f)
    } else {
        c(NA,lboundfun(M),NA)
    }
}
```

```
rfit <- laply(Kvec2,ffun,M=20)
```

(We get `NA` if $K = M$ — of course, since in this case $\hat{B} \to \infty$ — although we should still be able to get a lower bound in this case and hence include it in the coverage statistics, although if we include it in the bias calculation we will be in trouble. In the appropriate asymptotic case will the probability of this case go to zero fast enough??)

4.4 Doublets

This is based on Good (1979) (§10, "The repeat rate"): more generally these estimators are (apparently) known as "Good-Turing estimators", e.g. McAllester and Schapire (2000):

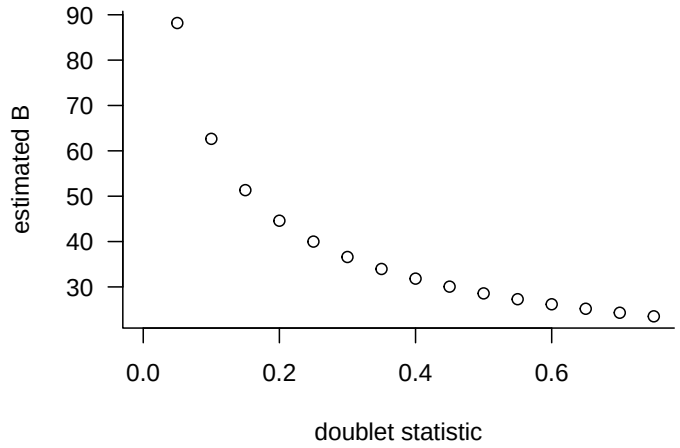
> The total probability mass of the words not in the sample is the so-called missing mass. Good showed that the fraction of the sample consisting of words that occur only once in the sample is a nearly unbiased estimate of the missing mass. Here, we give a PAC-style high-probability confidence interval for the actual missing mass. More generally, for $k > 0$, we give a confidence interval for the true probability mass of the set of words occuring k times in the sample.

There is also other relevant literature in ecology, mostly from the point of view of species distribution estimation (Good, 1953; Chao and Lee, 1992) — maybe a good place to go to get procedures for estimating confidence intervals ...

The sum of doublets, $W = \sum v_i(v_i - 1)$, (where $V = \{v_i\}$ = the unordered occupancy of birds) has an expected value

$\hat{B}^2 - \hat{B} - M(M-1)/W = 0$ hence $\hat{B} = 1 \pm \sqrt{1 + 4M(M-1)/W}/2$

```
Wtab <- table(Wvec)/nsim
Wvec2 <- as.numeric(names(Wtab))
Bhat_doublet <- 1 + sqrt(1+4*M*(M-1)/Wvec2)/2
plot(Wvec2,Bhat_doublet,xlab="doublet statistic",ylab="estimated B")
```



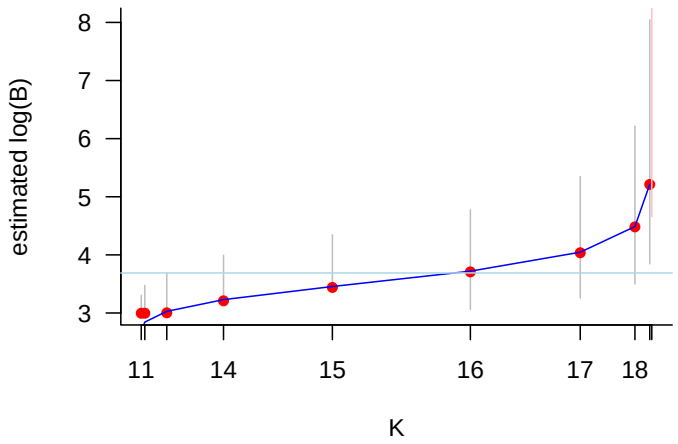
4.5 time to first collision

```
Texpfun <- function(B,maxn=4*B) {
    nvec <- 2:maxn
    sum(exp(-(nvec*(nvec-1))/(2*B)))
}
Bhat_T <- function(tau) {
    uniroot(function(x) { tt <- Texpfun(x)
                          ## cat(tau,tt,"\n")
                          tt-tau },
            interval=c(tau,1e6))$root
}
Ctab <- table(Cvec)/nsim
Cvec2 <- as.numeric(names(Ctab))
```

```
Bhat_collision <- sapply(Cvec2,Bhat_T)
```

4.6 Results

Plot results for each value of K, with spacing on the horizontal axis corresponding to the probability distribution of K (the lower part of the confidence interval for $K = 100$ is drawn in light blue):



Coverage (nominal value is 0.95):

```
sum(Ktab*(rfit[,2]<log(B) & rfit[,3]>log(B)),na.rm=TRUE)

## [1] 0.592
```

Not bad (somewhat conservative).

Compare with fitting $P(B+1)/P(B) = (1+1/B)^M(1-K/(B+1)) = 1$ (probably faster: not really susceptible to closed-form solution either).

```
rKfun <- function(B,K,M) {
    (1+1/B)^M*(1-K/(B+1))-1
}
u1 <- uniroot(rKfun,interval=c(16,1e6),K=16,M=20)
log(u1$root)

## [1] 3.695131

fitB(16,20)

## $fit
##     logB
## 3.707357
##
## $confint
##    2.5 %   97.5 %
## 3.061193 4.775601
```

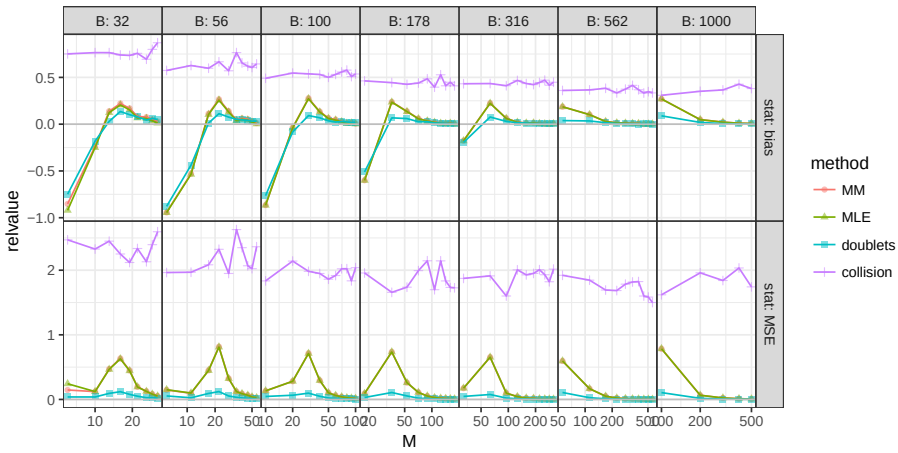
Actually not quite identical — precision problems?

Approximations:

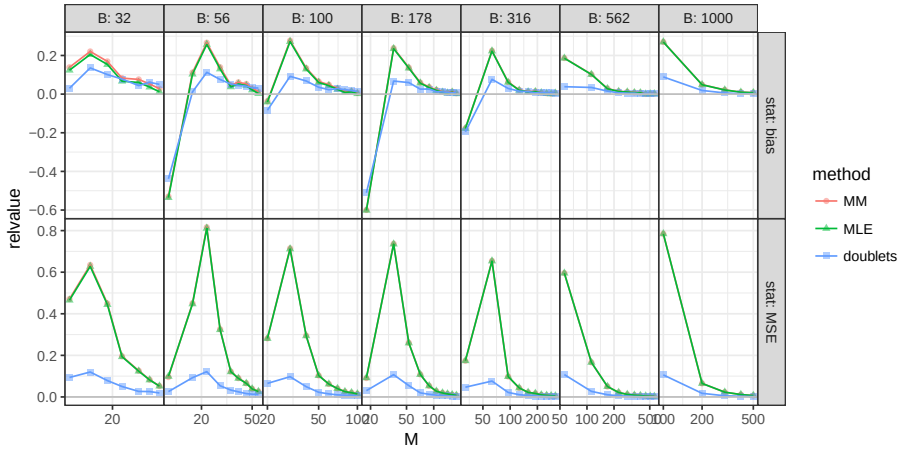
$$\begin{aligned}(1+1/B)^M\,(1-K/(B+1)) &\approx e^{M/B}(1-K/(B+1)) \qquad (B, M \gg 1)\\ &\approx e^{M/B}(1-K/B) \qquad (B \gg 1)\end{aligned}$$

Probably a fine approximation, but not sure that we can get much more out of this without a much more extreme approximation like $e^{M/B} \approx 1+(M/B)$ — would be nice if we could use Lambert W but I don't see how.

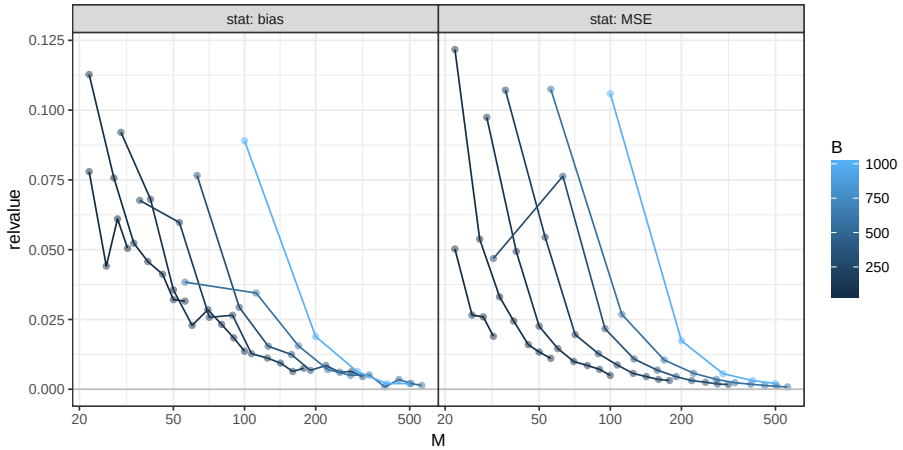
4.7 Simulation results



Take out first-collision method and $M > 10$:



Zoom in further (doublets only, value > 0, $M > 20$):



The time-to-first-collision method is terrible (although I can't rule out the possibility that I made a mistake). Somewhat to my surprise, the doublet method seems to dominate. There is some severe negative bias for particular combinations of low M and intermediate B, although I should again double-check and make sure that something funny isn't going on (also note that the $K = M$ results might be excluded from some of the calculations). I could look in more detail at the distributions of K and W for those cases, to see what's going on ...

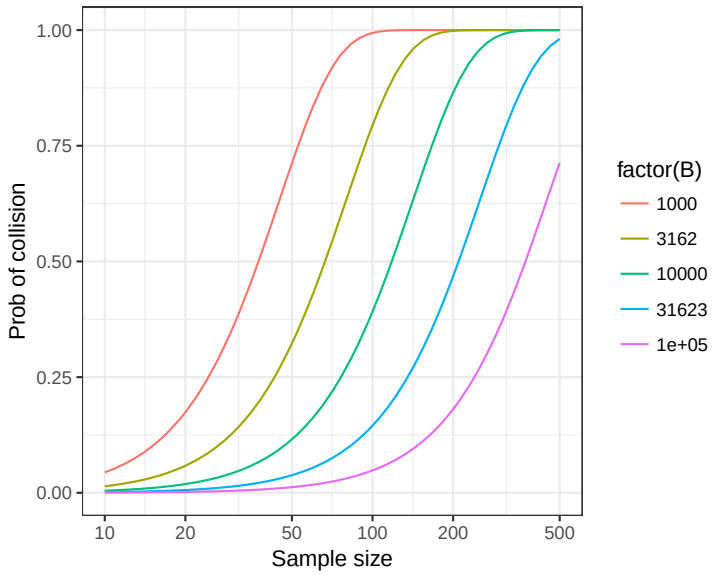
- Bias reduction methods??
- Rules of thumb for keeping MSE below some threshold (in terms of M or M/B)?
- Why do bias and MSE for MLE/MM have an intermediate peak?
- Should work out confidence intervals for the doublet method

5 Probability of collision

As before, we know that $P(K = M)$ is $\prod_{i=0}^{M-1}(B-i)/B$. This is useful in and of itself. If we have a preliminary estimate of the bird population size, how big do we have to make the sample size M to get a specified probability of collision?

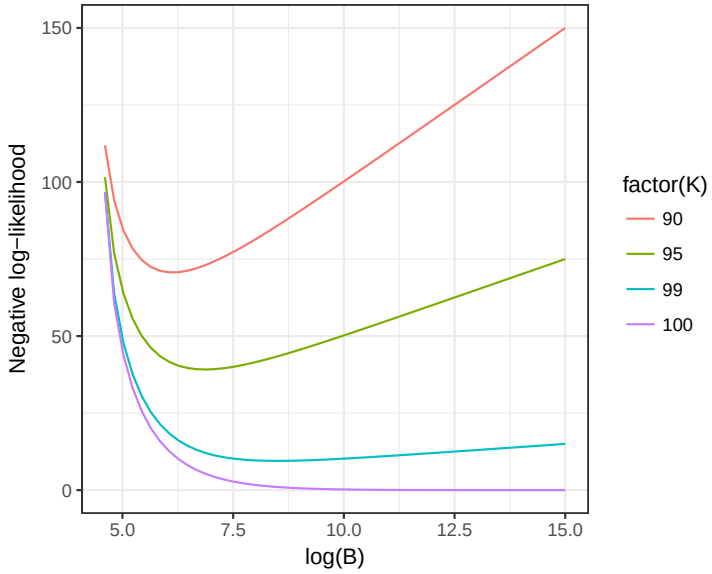
```
collision_prob <- function(M,B,inverse=FALSE,log=FALSE) {
    r <- sum(log(B-seq(0,M-1)))-M*log(B)
    if (inverse) {
        if (log) r else exp(r)
    } else {
        r2 <- 1-exp(r)
        if (log) log(r2) else r2
    }
}
```

Approximation for $M \ll B$?



6 Lower bound on B when $K = M$

This is a little dodgy, but: when $K = M$, the MLE $\hat{B}$ goes to infinity, because $-M \log B + \log B! - \log(B - K)!$ is maximized as $B \to \infty$.



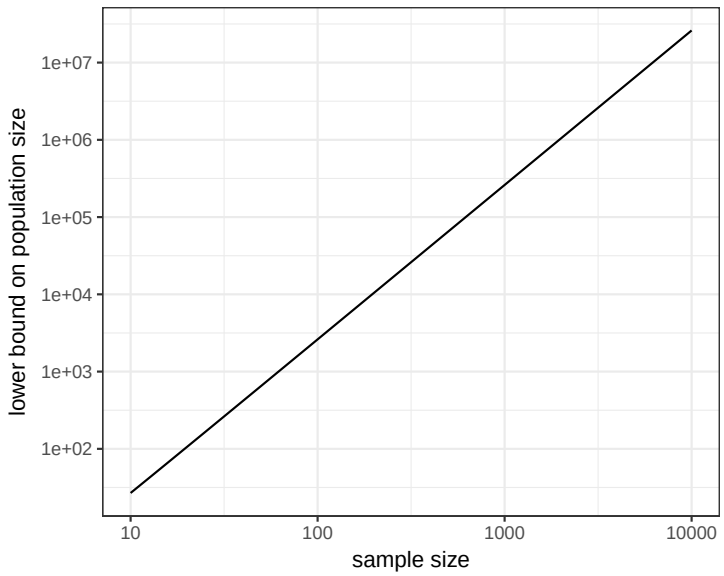
However, we can still try to solve for the case where $-\log L = \chi^2_1(0.975)/2 = 1.92$ to get a lower bound. For example, if $M = K = 100$,

```
lobound <- uniroot(function(x) nllfun(x,K=100,M=100)-1.92,
   interval=c(log(101),25))$root
```

So in this case (for $M = 100$, $K = 100$) our 95% lower bound on the population size is $B = 2611$ (this leaves open the question of whether profile confidence limits are reasonable in this case).

Here's a more general result:

```
lboundfun <- function(M) {
    uniroot(function(x) nllfun(x,K=M,M=M)-1.92,
            interval=c(log(M+0.001),25))$root
}
Mvec <- round(10^seq(1,4,length=31))
dd3 <- data.frame(M=Mvec,lo=exp(sapply(Mvec,lboundfun)))
ggplot(dd3,aes(M,lo))+geom_line()+
    scale_y_log10(breaks=10^(2:7))+
    scale_x_log10(breaks=10^(1:4))+
    labs(x="sample size",y="lower bound on population size")
```



This is a perfectly boring graph, with log-log slope nearly exactly 2.0 — a quadratic relationship between the lower bound and the size of M (this should be easy to work out analytically, or to argue heuristically?)

7 To do

- Theoretical justification for using MLE, and profile likelihood CI: in what limit are we working — what gets large (i.e. B, M, K, s_0)? In general

we can expect $B \approx s_0 \gg M \approx K$ ... Can we show that something converges appropriately to give us asymptotic consistency, χ^2 distribution of deviance, etc.?

- Get the theoretical distribution of K (hypergeometric??), which might allow us to compute bias for particular cases by brute force, and make a case about asymptotic stuff? (If we know the distribution of K then we also know the distribution of $\hat{B}$, by inversion ...)
- Could probably get confidence limits considerably quicker by root-finding rather than using the general `mle2` machinery?
- Might be able to do more exact/more justifiable confidence limits on B by evaluating the spectrum of probabilities of no-collision as a function of B?

8 Data (!!)

OK, now we actually have a little bit of information about plausible sizes of B, M ...

The actual B values (number of American robins at a site) are thought to be in the range 10–40 (much smaller than I was imagining).

Some values of M from three sites across a range of years:

```
##           site year  M
## 1 Foggy Bottom 2004 19
## 2 Foggy Bottom 2006 11
## 3 Foggy Bottom 2008 13
## 4 Foggy Bottom 2011 17
## 5    Baltimore 2008 40
## 6    Baltimore 2010 18
## 7         NMNH 2004 14
```

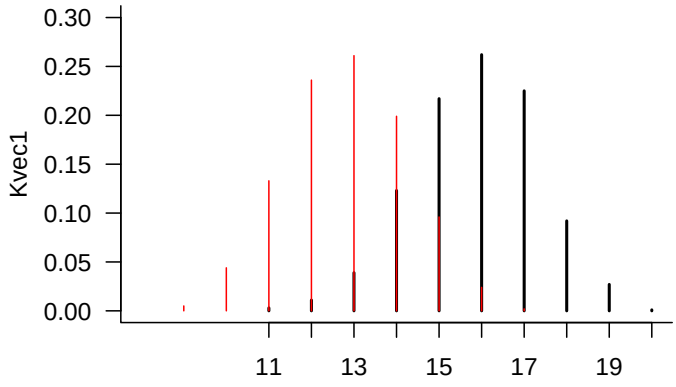
So if the local bird populations are really as small as they are thought to be, we should be very surprised if there are no collisions: here are the collision probabilities (i.e. the probability that at least one bird is sampled by more than one mosquito) for $B = 40$:

```
round(sapply(mdat$M,collision_prob,B=40),3)

## [1] 0.994 0.780 0.888 0.982 1.000 0.989 0.925
```

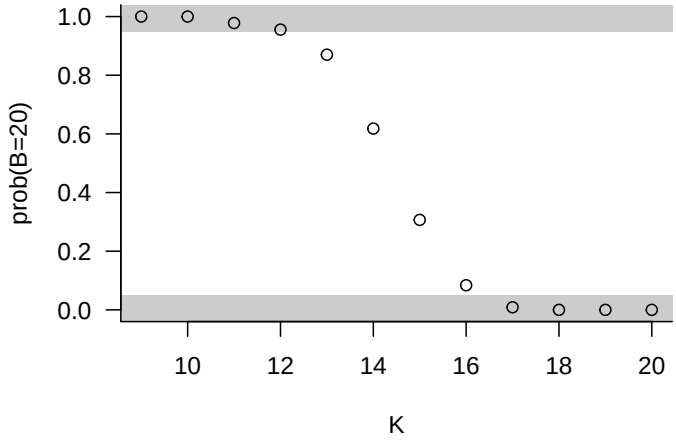
(The probability of a collision with $B = M = 40$ is not exactly 1, but it's very close: the probability of *not* having a collision is 6.749093×10^{-17}.)

It's not clear how much power we have to distinguish different effective population sizes from these data (i.e. we could tell if they were much larger, but not necessarily if they're much smaller). For example, here are the expected K

distributions from an effective population size of $B = 40$ (black) and $B = 20$ (red):



For example, if we were trying to distinguish the two hypotheses that $B = 20$ vs. $B = 40$:



If B were really 20, we would only have a power of 0.418 to detect the difference.

9 To do

- incorporate stuff from mbrs talk
- estimates
- confidence intervals for doublets?
- hierarchical models? combinations?
- stuff from Steve Walker, JD

10 Junk

10.1 Derivatives of log-likelihood

If we wanted to, we would in principle be able to use $dL/d(\log B) = -M + \psi(B)B - \psi(B-K)B$ where ψ is the digamma function (although this probably isn't worth it because we can likely solve all our problems faster by root-finding rather than minimization).

```
nllgrad <- function(logB,K,M) {
    B <- exp(logB)
    M - digamma(B)*B+digamma(B-K)*B
}
grad(function(x) nllfun(x,K=98,M=100),x=7)

## [1] -2.609517

nllgrad(7,98,100)

## [1] -2.707652

grad(function(x) nllfun(x,K=7,M=4),x=3)

## [1] -4.344651

nllgrad(3,7,4)

## [1] -4.879593
```

(Not quite right yet.)

10.2 Closed-form solution for Prob(K)

LB thinks this is difficult.

Gessel solution for K (Bose-Einstein??): $\left(\binom{B}{B-K}\binom{M-1}{K-1}\right) / \binom{B+M-1}{M}$

```r
## dhyper(B-K,B,M-1,M)
myhyper <- function(K,M,B,log=FALSE) {
    r <- lchoose(B,B-K)+lchoose(M-1,K-1)-lchoose(B+M-1,M)
    if (log) r else exp(r)
}
## R notation:
## H(x,m,n,k) -> choose(m,x) choose(n,k-x) / choose(m+n,k)
## K=B-s_0
## x=s0=B-K; m=B; n=M-1; k=M; m+n=B+M-1; k-x =M-s0
```

11 More on Good-Turing

Wikipedia, etc.

References

Chao, A. and S. Lee (1992, March). Estimating the number of classes via sample coverage. *Journal of the American Statistical Association 87*(417), 210–217.

Good, I. J. (1953, December). The population frequencies of species and the estimation of population parameters. *Biometrika 40*(3-4), 237–264.

Good, I. J. (1979). Studies in the history of probability and statistics. XXXVII A. M. Turing's statistical work in World War II. *Biometrika 66*(2), 393–396.

McAllester, D. and R. E. Schapire (2000). On the convergence rate of Good-Turing estimators. In *Proceedings of the Thirteenth annual conference on computational learning theory*, pp. 16.