Sampling regimes for rare species with imperfect detection

Jens Åström

2018-11-20

Intro

I here explore some effects of rare occurrence and imperfect detection in species observation and explore alternative survey regimes. This is mostly relevant for rare species, and not for the general survey of insects. Early detection of alien species is another use case.

The objective is to identify suitable sampling strategies for a couple of different scenarios, where we maximise the chances of detecting a species at least once while considering the economic costs. The basic principle here is that a species can be present in a sampling location with a given probability of occurrence. The observers then have a given chance (probability) of detecting a species on a visit, if it is present. The total probability of of detecting a species will be dependent on both the probability of occurrence and the probability of detection, how many locations are visited, and how many visits we make at each location.

If ψ represents the occurrence probability and θ is the detection probability, the probability of observing the species at least once in J locations visited K times is:

$$probObserv = 1 - (1 - \psi * (1 - (1 - \theta)^K))^J.$$

The costs can be assumed to scale roughly linearly to the total amount of visits, i.e. $totalCost = J * K * c$, where c is the cost of one visit/survey.

First look

We can explore how the overall probability of observing the species, and the associated costs depend on ψ, θ, J, K, c graphically. I have made a convenience function `obsProb` to calculate the values. A lot of the results seem pretty obvious in retrospect, but it is still worth plotting them to get a better feel for the possibilities. At the moment, we only consider one and the same cost for each survey visit. It may be worth while to code up using a higher initial cost of the first visit, and lower costs for subsequent visits.

```
require(InsectSurvPower)
## custom library, available through
## devtools::install_github('NINAnor/InsectSurvPower')
require(dplyr)
require(gridExtra)
require(ggplot2)
require(xtable)
```

We specify one or a range of values for occurrence probability in each location, detection probabilities, the number of locations visited, the number of visits per location, and the cost of each visit. To illustrate, we here use occurrence probabilities ranging from 0.01 to 1, two different detection probabilities as 0.5, and 0.8. We specify 30 locations, each visited 4 times, at an individual visit cost of 5000.

```
obsDf <- obsProb(occProb = seq(0.01, 1, by = 0.05), detectProb = c(0.5, 0.8),
    locations = 30, visits = 4, visitCost = 5000)

obsDf
```

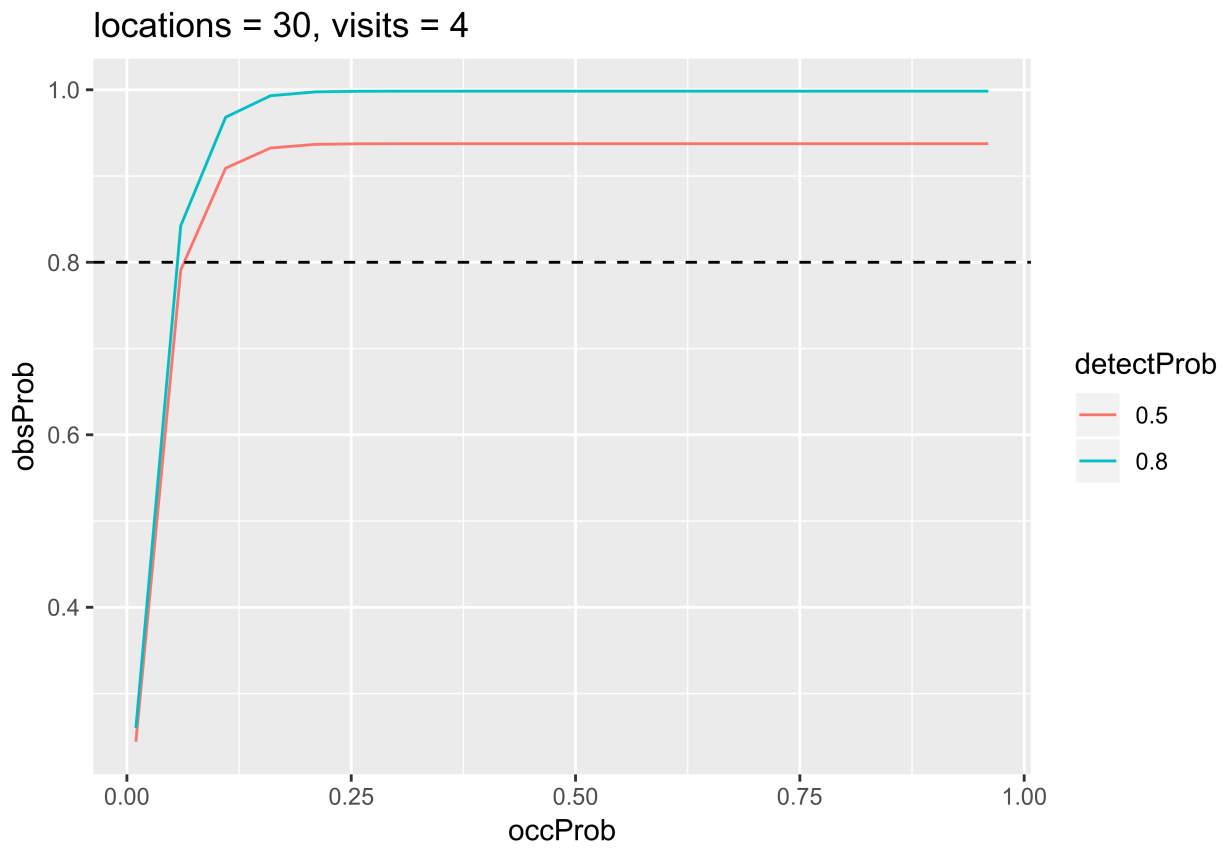


Figure 1: Observation probability for surveying a species with a detection probability of 0.5 in 30 locations, each visited 4 times.

```
## # A tibble: 40 x 7
##    occProb detectProb locations visits visitCost obsProb totCost
##      <dbl>      <dbl>     <dbl>  <dbl>     <dbl>   <dbl>   <dbl>
##  1  0.0100      0.500      30.0   4.00      5000   0.244  600000
##  2  0.0600      0.500      30.0   4.00      5000   0.791  600000
##  3  0.110       0.500      30.0   4.00      5000   0.909  600000
##  4  0.160       0.500      30.0   4.00      5000   0.932  600000
##  5  0.210       0.500      30.0   4.00      5000   0.937  600000
##  6  0.260       0.500      30.0   4.00      5000   0.937  600000
##  7  0.310       0.500      30.0   4.00      5000   0.937  600000
##  8  0.360       0.500      30.0   4.00      5000   0.937  600000
##  9  0.410       0.500      30.0   4.00      5000   0.937  600000
## 10  0.460       0.500      30.0   4.00      5000   0.937  600000
## # ... with 30 more rows
```

The probability of observing the species at least once is found in the `obsProb` column, and the total cost in the column `totCost`. The function returns an object of a specific class with a custom plotting function. This makes repeated plottings easier. We can specify a grouping variable to split up the lines.

```
plot(obsDf, group = "detectProb", xVar = "occProb", yVar = "obsProb", titleVar = c("locations",
    "visits"), hline = 0.8)
```

Figure 1 shows how the overall obervation probability is dependent on both occurrence and detection probability. A threshold of a total observation probability of 0.8 is added for comparison. The threshold is

reached in this case when the probability of occurrence in each location is 6%. With 30 locations, the overall observation probability rises quite sharply as a result of increased occurrence probability. With only 4 visits per location, the detection probability limits the overall achievable observation probability.

However, these occurrence probabilities are probably unreasonably high for rare species in the real world. If we assume that we search for a truly rare species with an occurrence probability of only 0.001, we find that reaching overall detectabilities above 80% is challenging (Figure 2).

```
rareLocDf <- obsProb(occProb = 0.001, detectProb = seq(0.4, 0.8, by = 0.2),
    locations = seq(30, 1000, by = 20), visits = seq(4, 16, by = 4), visitCost = 5000)

rareLocDf
```

```
## # A tibble: 588 x 7
##    occProb detectProb locations visits visitCost obsProb totCost
##      <dbl>      <dbl>     <dbl>  <dbl>     <dbl>   <dbl>   <dbl>
##  1 0.00100      0.400      30.0   4.00      5000  0.0257  600000
##  2 0.00100      0.600      30.0   4.00      5000  0.0288  600000
##  3 0.00100      0.800      30.0   4.00      5000  0.0295  600000
##  4 0.00100      0.400      50.0   4.00      5000  0.0425 1000000
##  5 0.00100      0.600      50.0   4.00      5000  0.0475 1000000
##  6 0.00100      0.800      50.0   4.00      5000  0.0487 1000000
##  7 0.00100      0.400      70.0   4.00      5000  0.0589 1400000
##  8 0.00100      0.600      70.0   4.00      5000  0.0659 1400000
##  9 0.00100      0.800      70.0   4.00      5000  0.0675 1400000
## 10 0.00100      0.400      90.0   4.00      5000  0.0750 1800000
## # ... with 578 more rows
```

```
plot(rareLocDf, group = "visits", xVar = "locations", yVar = "obsProb", hline = 0.8) +
    facet_grid(occProb ~ detectProb)
```

In figure 2, we see that observing a very rare species with a high certainty is difficult even with a very large number of visited locations. In these cases, it doesn't really help to visit each location many times, as the overall probability is limited by the number of locations we visit. Figure 3 shows the results of maximising the number of visits in fixed, but large number of locations, for a very rare species.

```
rareVisitDf <- obsProb(occProb = 0.001, detectProb = seq(0.2, 0.8, by = 0.2),
    locations = 250, visits = seq(1, 11, by = 5), visitCost = 5000)

rareVisitDf
```

```
## # A tibble: 12 x 7
##    occProb detectProb locations visits visitCost obsProb  totCost
##      <dbl>      <dbl>     <dbl>  <dbl>     <dbl>   <dbl>    <dbl>
##  1 0.00100      0.200       250   1.00      5000  0.0443  1250000
##  2 0.00100      0.400       250   1.00      5000  0.0885  1250000
##  3 0.00100      0.600       250   1.00      5000  0.133   1250000
##  4 0.00100      0.800       250   1.00      5000  0.177   1250000
##  5 0.00100      0.200       250   6.00      5000  0.163   7500000
##  6 0.00100      0.400       250   6.00      5000  0.211   7500000
##  7 0.00100      0.600       250   6.00      5000  0.220   7500000
##  8 0.00100      0.800       250   6.00      5000  0.221   7500000
##  9 0.00100      0.200       250  11.0       5000  0.202  13750000
## 10 0.00100      0.400       250  11.0       5000  0.220  13750000
## 11 0.00100      0.600       250  11.0       5000  0.221  13750000
## 12 0.00100      0.800       250  11.0       5000  0.221  13750000
```

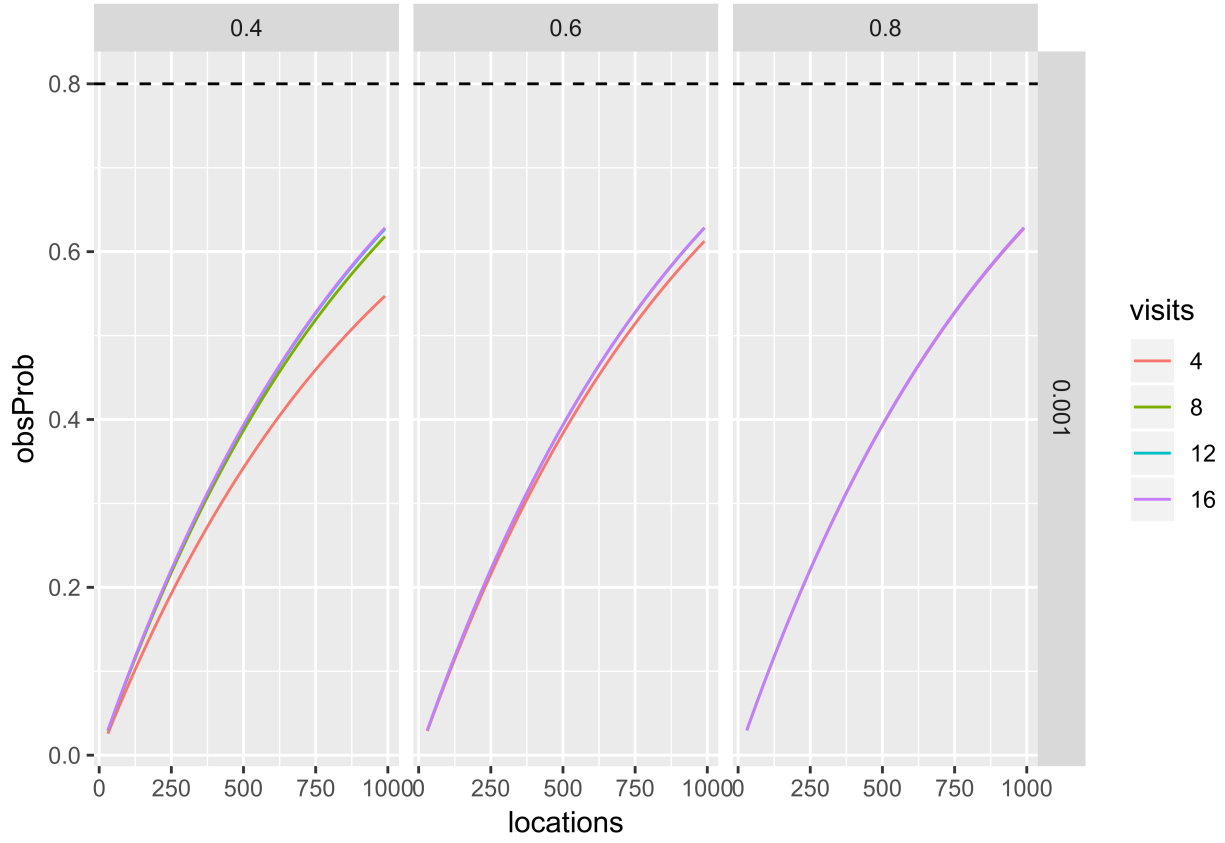


Figure 2: Probability of detecting a rare species (occurrence probability = 0.001) as a function of the number of visited locations and visits per location for different detection probabilities (detectProb = 0.4, 0.6, and 0.8).

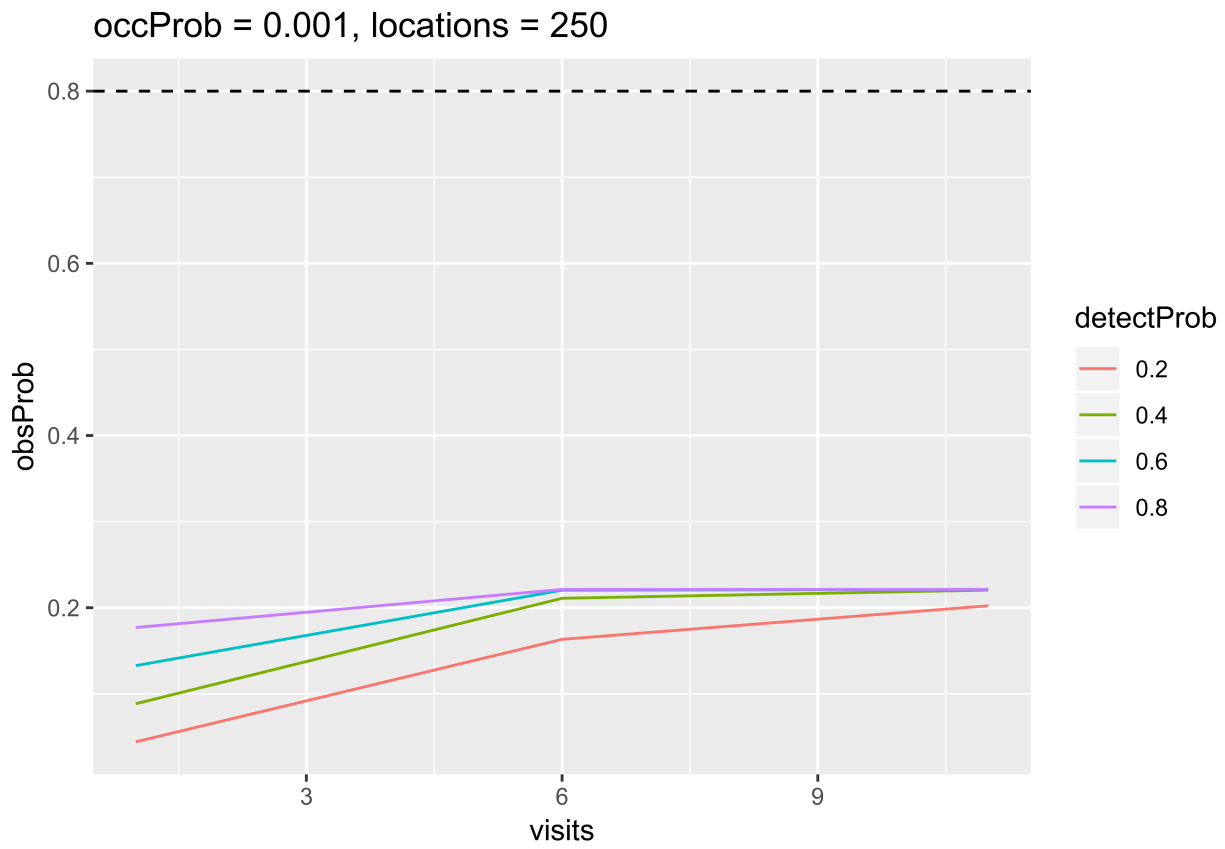


Figure 3: Observation probability for surveying a very rare species as function of sampled locations.

```
plot(rareVisitDf, group = "detectProb", xVar = "visits", yVar = "obsProb", titleVar = c("occProb",
    "locations"), hline = 0.8)
```

Although the overall cost increases linearly with the total number of samples (figure 4), in cases with very rare species, this doesn't mean that the overall detection probability continues to increase indefinitely (figure 3)

```
rareCostDf <- obsProb(occProb = 0.001, detectProb = 0.4, locations = seq(50,
    250, by = 50), visits = seq(1, 21, by = 5), visitCost = 5000)

rareCostDf
```

```
## # A tibble: 25 x 7
##    occProb detectProb locations visits visitCost obsProb totCost
##      <dbl>      <dbl>     <dbl>  <dbl>     <dbl>   <dbl>   <dbl>
##  1 0.00100      0.400      50.0   1.00      5000  0.0195  250000
##  2 0.00100      0.400     100     1.00      5000  0.0381  500000
##  3 0.00100      0.400     150     1.00      5000  0.0557  750000
##  4 0.00100      0.400     200     1.00      5000  0.0725 1000000
##  5 0.00100      0.400     250     1.00      5000  0.0885 1250000
##  6 0.00100      0.400      50.0   6.00      5000  0.0465 1500000
##  7 0.00100      0.400     100     6.00      5000  0.0908 3000000
##  8 0.00100      0.400     150     6.00      5000  0.133  4500000
##  9 0.00100      0.400     200     6.00      5000  0.173  6000000
```

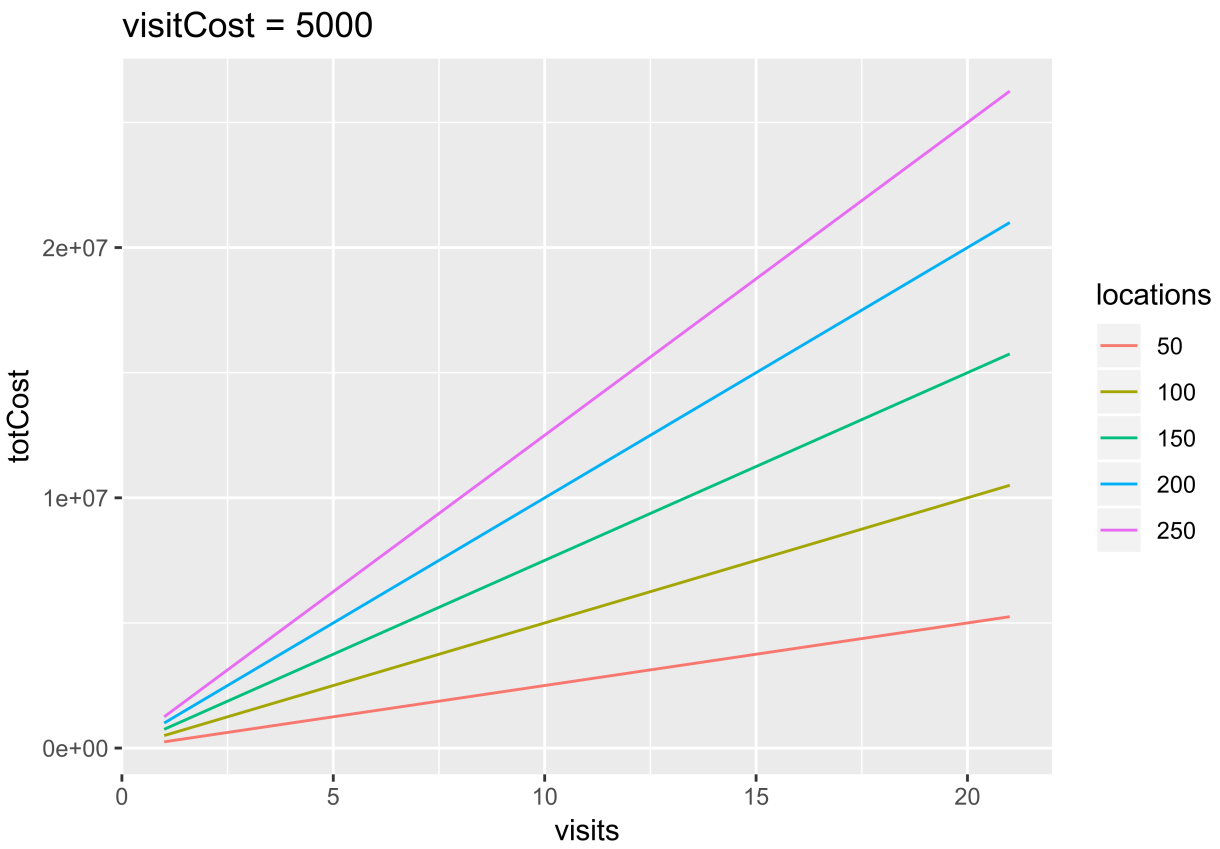


Figure 4: Total survey cost as a function of the total number of samples

```
## 10 0.00100      0.400     250     6.00      5000  0.211  7500000
## # ... with 15 more rows
```

```
plot(rareCostDf, group = "locations", xVar = "visits", yVar = "totCost", titleVar = "visitCost")
```

Some strategies

As seen in figure 5, when we deal with a very rare species, it is little use increasing the number of visits to each location (or to spend money maximizing the detection probability), if we can't at the same time span a very large number of locations. We must in these cases concentrate on increasing the number of visited locations. Still, for a very rare species such as displayed in figure 5, with an occurrence probability of 0.01%, reaching an overall observation probability of 80% requires more at least 1650 locations, which would be unfeasible for most survey programs.

Alternatively, if detectability is low but presence is relatively high, we should focus on increasing the number of revisits per location, instead of trying to cover many locations (figure 6).

```
lowOccurrDf <- obsProb(occProb = 0.001, detectProb = 0.8, locations = seq(50,
    2000, by = 50), visits = c(1, seq(5, 20, by = 5)), visitCost = 5000)

lowOccurrDf
```

```
## # A tibble: 200 x 7
```

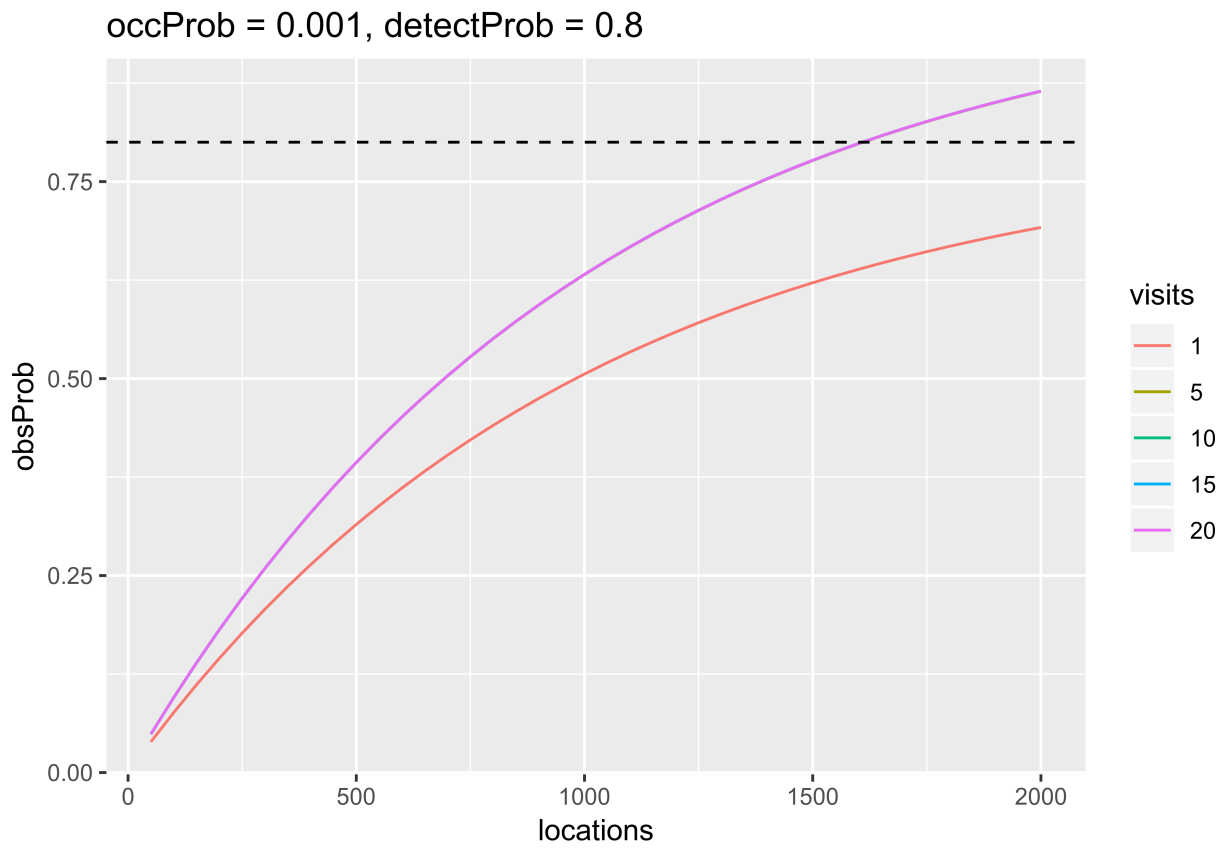


Figure 5: Observation probability for surveying a rare species as function of sampled locations.

```
##    occProb detectProb locations visits visitCost obsProb totCost
##      <dbl>      <dbl>     <dbl>  <dbl>     <dbl>   <dbl>   <dbl>
##  1 0.00100      0.800      50.0   1.00      5000  0.0390  250000
##  2 0.00100      0.800     100     1.00      5000  0.0762  500000
##  3 0.00100      0.800     150     1.00      5000  0.111   750000
##  4 0.00100      0.800     200     1.00      5000  0.145  1000000
##  5 0.00100      0.800     250     1.00      5000  0.177  1250000
##  6 0.00100      0.800     300     1.00      5000  0.207  1500000
##  7 0.00100      0.800     350     1.00      5000  0.236  1750000
##  8 0.00100      0.800     400     1.00      5000  0.264  2000000
##  9 0.00100      0.800     450     1.00      5000  0.290  2250000
## 10 0.00100      0.800     500     1.00      5000  0.315  2500000
## # ... with 190 more rows
```

```
plot(lowOccurrDf, group = "visits", xVar = "locations", yVar = "obsProb", titleVar = c("occProb",
    "detectProb"), hline = 0.8)
```

```
lowDetectDf <- obsProb(occProb = 0.05, detectProb = 0.2, locations = seq(50,
    250, by = 50), visits = c(1, seq(5, 20, by = 5)), visitCost = 5000)

lowDetectDf
```

```
## # A tibble: 25 x 7
##    occProb detectProb locations visits visitCost obsProb totCost
```

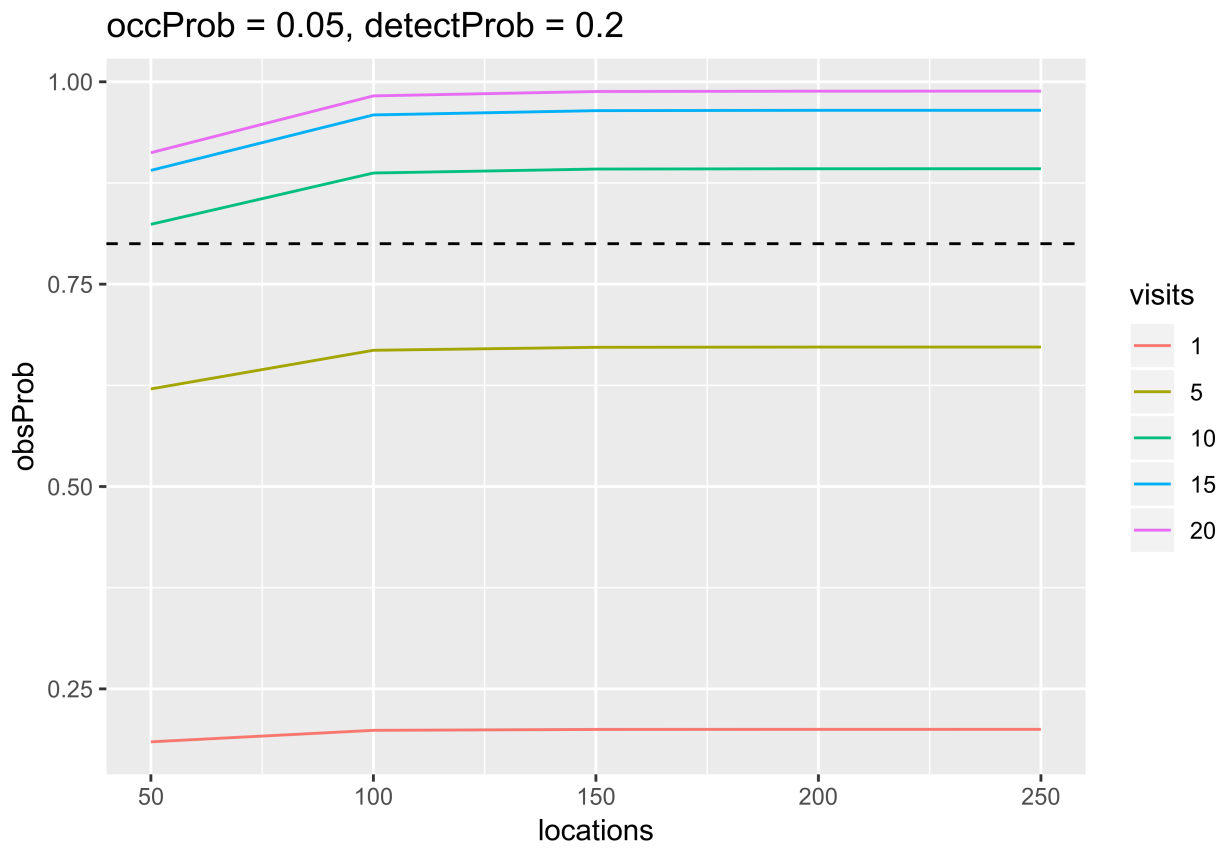


Figure 6: Observation probability for surveying a cryptic species as function of sampled locations.

```
##       <dbl>      <dbl>     <dbl>  <dbl>     <dbl>   <dbl>   <dbl>
##  1  0.0500       0.200      50.0   1.00      5000   0.185  250000
##  2  0.0500       0.200     100     1.00      5000   0.199  500000
##  3  0.0500       0.200     150     1.00      5000   0.200  750000
##  4  0.0500       0.200     200     1.00      5000   0.200 1000000
##  5  0.0500       0.200     250     1.00      5000   0.200 1250000
##  6  0.0500       0.200      50.0   5.00      5000   0.621 1250000
##  7  0.0500       0.200     100     5.00      5000   0.668 2500000
##  8  0.0500       0.200     150     5.00      5000   0.672 3750000
##  9  0.0500       0.200     200     5.00      5000   0.672 5000000
## 10  0.0500       0.200     250     5.00      5000   0.672 6250000
## # ... with 15 more rows
```

```
plot(lowDetectDf, group = "visits", xVar = "locations", yVar = "obsProb", titleVar = c("occProb",
    "detectProb"), hline = 0.8)
```

Plausible values

It is difficult to guess plausible values for occurrence and detectability for real world species, but it is reasonable to assume that we only have to consider rather low occurrence probabilities, since we are working on early detections. We can explore our possibilities of observing a species that occurr in between 0.1 to 1% of all studied locations. We can set the number of locations to 100 and with two visits, as a reasonable

possibility.

```
guessDf <- obsProb(occProb = c(0.001, 0.005, 0.01, 0.02, 0.04, 0.05, 0.1), detectProb = c(0.1,
    seq(0.2, 0.8, by = 0.2)), locations = c(50, seq(100, 300, by = 100)), visits = seq(2,
    6, by = 2), visitCost = 5000)

guessDf
```

```
## # A tibble: 420 x 7
##    occProb detectProb locations visits visitCost obsProb totCost
##      <dbl>      <dbl>     <dbl>  <dbl>     <dbl>   <dbl>   <dbl>
##  1 0.00100      0.100      50.0   2.00      5000 0.00927  500000
##  2 0.00500      0.100      50.0   2.00      5000 0.0421   500000
##  3 0.0100       0.100      50.0   2.00      5000 0.0750   500000
##  4 0.0200       0.100      50.0   2.00      5000 0.121    500000
##  5 0.0400       0.100      50.0   2.00      5000 0.165    500000
##  6 0.0500       0.100      50.0   2.00      5000 0.175    500000
##  7 0.100        0.100      50.0   2.00      5000 0.189    500000
##  8 0.00100      0.200      50.0   2.00      5000 0.0176   500000
##  9 0.00500      0.200      50.0   2.00      5000 0.0798   500000
## 10 0.0100       0.200      50.0   2.00      5000 0.142    500000
## # ... with 410 more rows
```

```
plot(guessDf, group = "detectProb", xVar = "occProb", yVar = "obsProb", hline = 0.8) +
    facet_grid(visits ~ locations) + theme(axis.text.x = element_text(size = 5))
```

```
plot(guessDf, group = "detectProb", xVar = "occProb", yVar = "obsProb", hline = 0.8) +
    facet_grid(visits ~ locations) + scale_y_log10() + scale_x_log10() + theme(axis.text.x = element_te
    ylab("Observasjonssannsynlighet") + xlab("Forekomstsannsynlighet") + guides(color = guide_legend(ti
```

Unequally distributed probabilities

** There is no definition for early detection. The earlier we want to detect (no occurrences is low) the more costly to detect. We can use the alien/native map as a relative risk map and use that to "model" a weighted occurrence map. This we visit in a weighted fashion as well, a number of times. Need to work out the math. We can then calculate the costs for different number of alien occurrences.**

In the equation $probObserv = (1 - (1 - \psi)^J) * (1 - (1 - \theta)^K))$, ψ designates the probability that a site we visit contains a certain species. So far, we have only considered situations where the occurrence probabilities (ψ) are the same for all sites. In reality, however, the probability that a specific site that you visit contains a certain species will vary between sites. It will depend both on the probability of a site containing a species, and the probability that you visit that site. We can view the probability of a specific site containing a species as a weighted sampling without replacement. For example, if we know there are a 100 sites containing species x, we can calculate the probability that each site contains species x if we know the probability weights. This probability can be written, following Eframidis & Spirakis 2006,as $p_i(k) = \frac{w_i}{\sum_{S_j \in V-S} w_j}$. For brevity, however, we will simply designate this probability as $Pr[w_i, n]$, where the weights w_i sums to 1, and n is the number of sites with the species. Similarly, we choose which sites to visit as a sampling without replacement, so that the probability of visiting a site i is $Pr[u_i, v]$, where u_i is the visitation weights, which sums to 1, and v is the number of sites you visit. The probability of visiting a site with an alien species then becomes $\psi_i = Pr[w_i, n] * Pr[u_i, v]$, and the probability of visiting any location inhabited by an alien species is $PoccurrVisit = (1 - \prod_i (1 - (Pr[w_i, n] * Pr[u_i, v])))$.

For simplicity, we can assume that the probability of detection is equal for each site and visit. The probability of detecting an alien species is then $Pdetect = (1 - \prod_i (1 - (Pr[w_i, n] * Pr[u_i, v]))) * (1 - (1 - d)^K$, where d

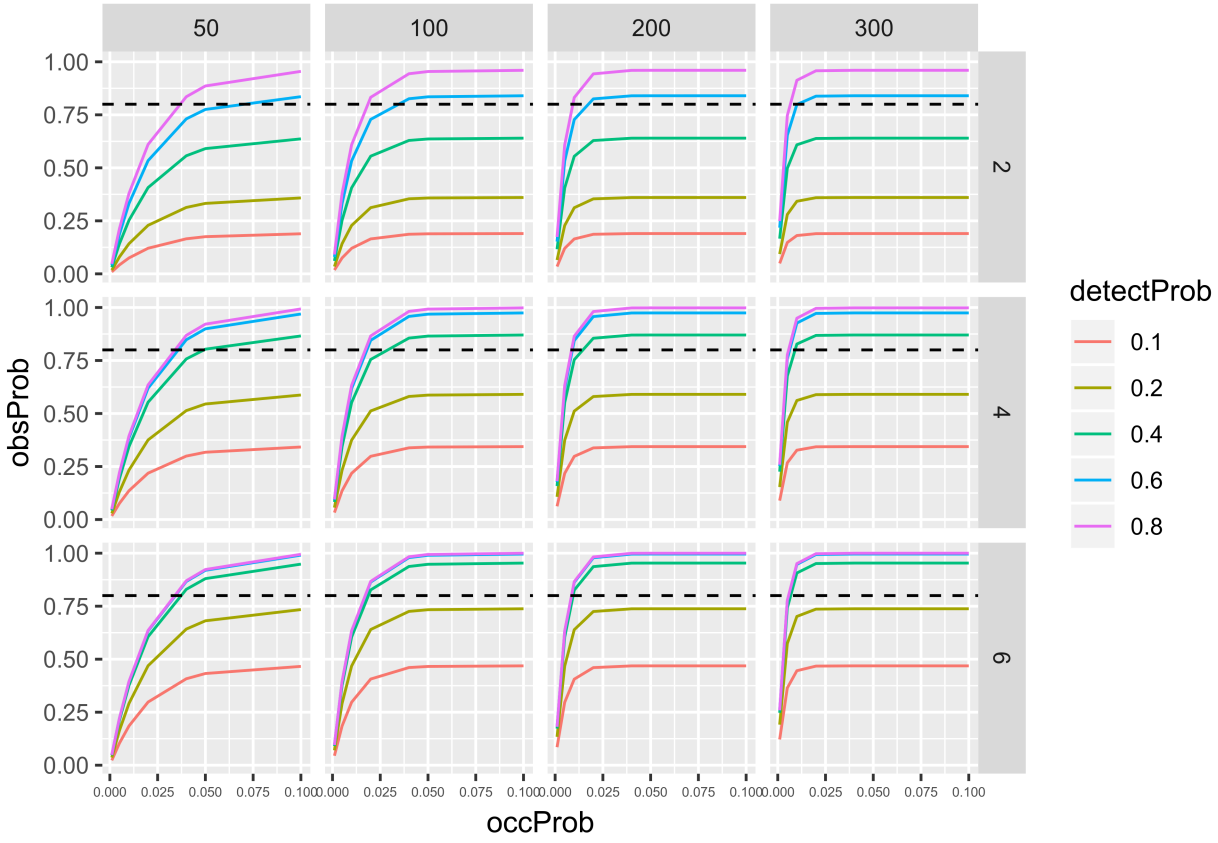


Figure 7: Observation probability for a plausible range of values. The plots are divided by number of sample locations in columns, and by number of visits in rows.

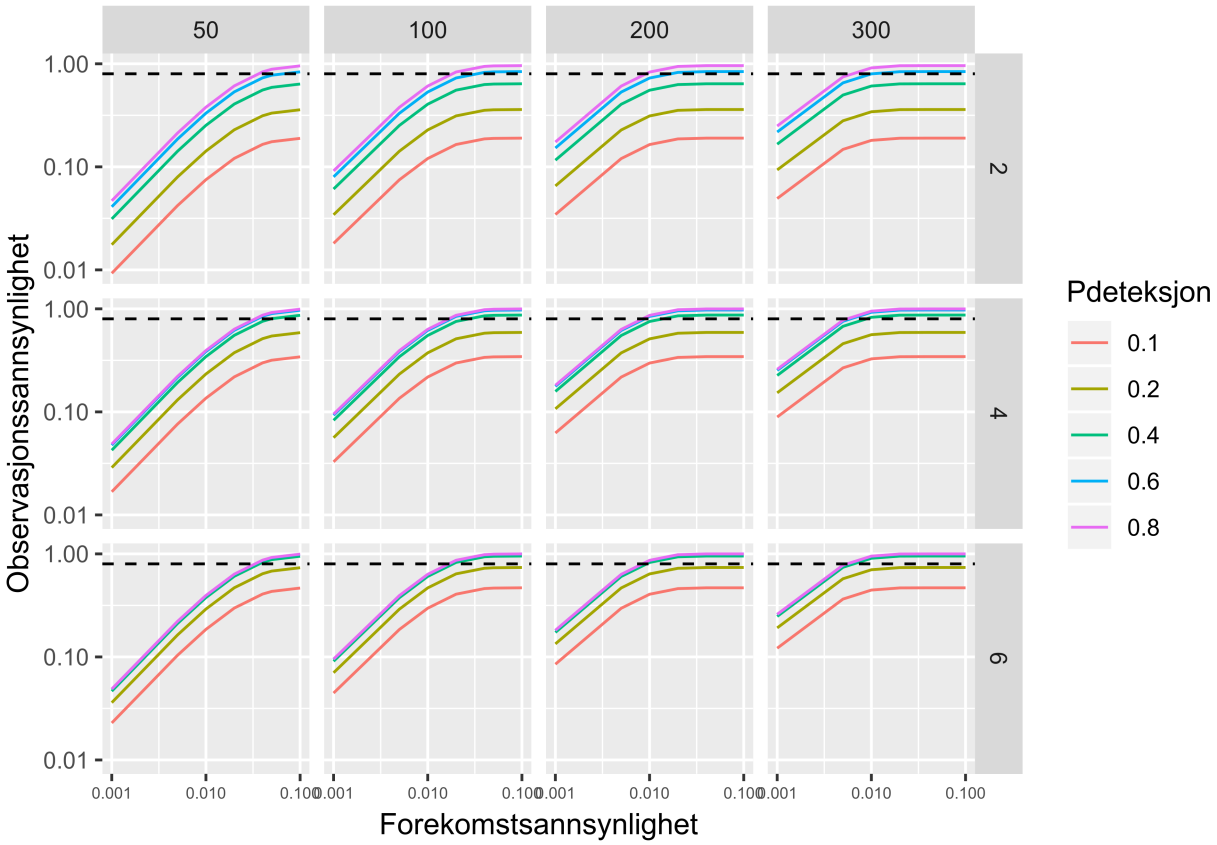


Figure 8: Observation probability for a plausible range of values. The plots are divided by number of sample locations in columns, and by number of visits in rows. Note the logarithmic scale.

is the detection probability, and K is again the number of visits per site.

In the best of worlds, w_i and u_i would match up well, so that we visit the most likely sites to contain a species of interest. The overall probability of detection will decrease as the difference between these variables increase, or in simpler terms we visit the wrong places. Also, the probability will go down the more spread out the probability weights are. In the extreme case, with no spread in these probabilities, there is 100 % certainty that a species will be present in location 1, and 100 % certainty that we will visit just that. In this case, the probabilities are 1 for both occurrence and visitation, so that we are certain we visit a site with a presence. In the other end, there might be no information in the weights, so that the occurrences are randomly spread out, and we visit random sites. In that case, w_i are all the same and u_i are all the same, and we end up with the first equation.

In reality, we don't know the true weights that the sites will contain a specific species, and we might choose to visit the wrong sites. For this example calculation, we will assume we know the occurrence weights, and visit them accordingly. In other words, that $w_i = u_i$. If we stipulate the desired detection probability (at e.g 0.8), we can calculate how many sites v we need to visit to be able to detect a species that occurr in n sites with weighted probabilities w_i, with a specific certainty.

So far, the best estimate for the occurrence weights are the occurrence modelling of alien vascular plants from Olsen et al. 2017. Using this as input, and some simple assumptions, we can calculate the needed number of sites.

To get test it out, we can use the 10km scale, which isn't so resource intensive. The 1km scale isn't really interactive since it takes to long time to calculate.

For now, we get rid of the geometry column to increase speed.

```
predWeights <- predMap %>% select(sites = ssbid, weights = pred) %>% sf::st_set_geometry(NULL)
```

But we start with a situation where the occupied patches are distributed randomly, and we visit the sites randomly. In other words, where the occurrence and visitation weights all are equal.

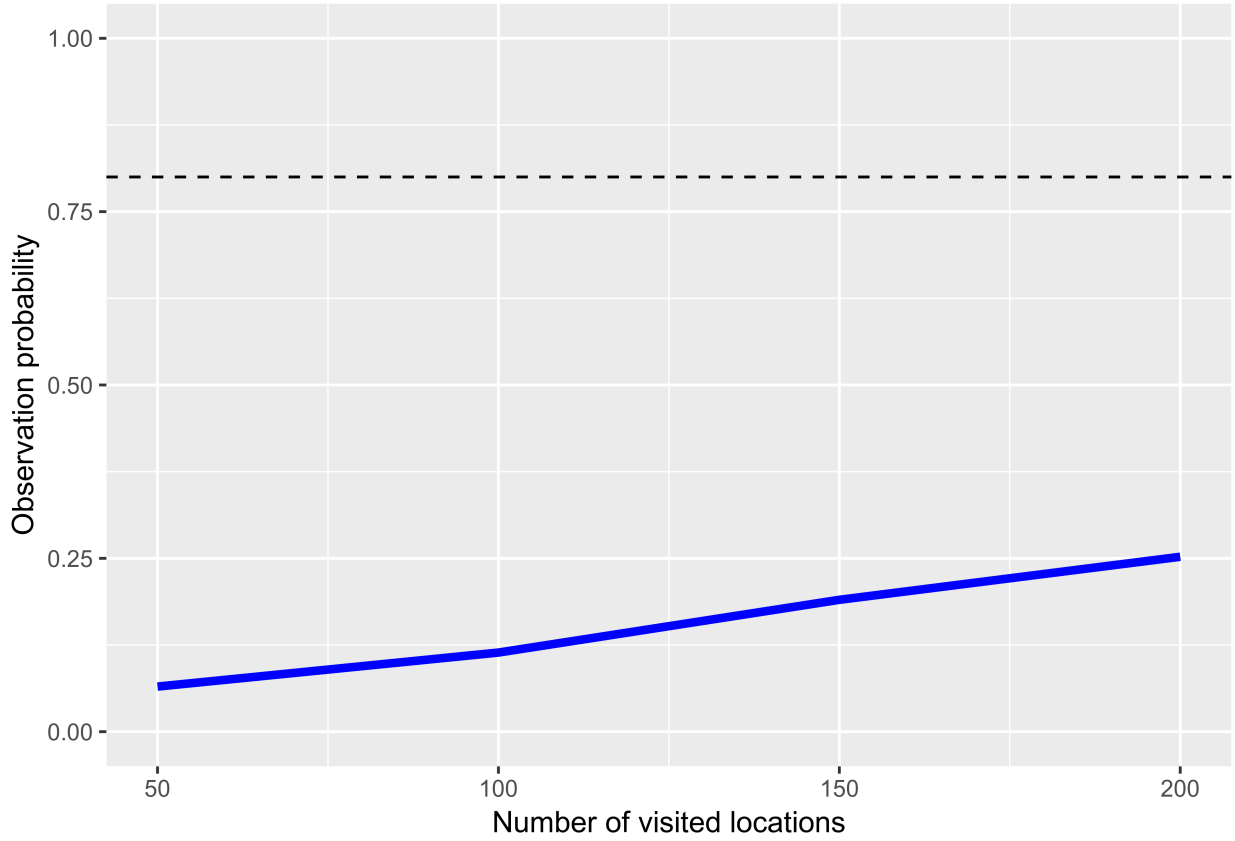
```
predWeightsZero <- predWeights
predWeightsZero$weights <- 1


system.time(predProb50Zero <- weightedDetection(occWeights = predWeightsZero,
    visWeights = predWeights, noOccur = 5, noLocations = seq(50, 200, by = 50),
    noVisits = 1, detectProb = 1))
```

```
##    user  system elapsed
##   2.724   0.008   2.733
```

```
predProb50Zero
```

```
## # A tibble: 4 x 2
##   noLocations probObs
##         <dbl>   <dbl>
## 1        50.0  0.0651
## 2       100    0.114
## 3       150    0.190
## 4       200    0.252
```

```r
plot(predProb50Zero, threshold = 0.8)
```



We can see that the probability of visiting a randomly occupied cell in this case starts from about 0.07 and approaches 0.2 as we increase our number of visited locations from 50 to 200. We can quality check this with a simpler function.

```r
test <- function(noOccur = 50, noLocations = 5, nIter = 999) {

    prop <- function(noLocations. = noLocations, noOccur. = noOccur) {
        visited <- sample(1:4057, noLocations., replace = F)  #number of 10km cells
        occupied <- sample(1:4057, noOccur, replace = F)

        any(visited %in% occupied)
    }


    sum(replicate(nIter, prop())/nIter)
}



test()
```

```
## [1] 0.07807808
```

We get the same results if the occurrence probabilities is distributed according to informative weights, and only the visitations are random (not shown).

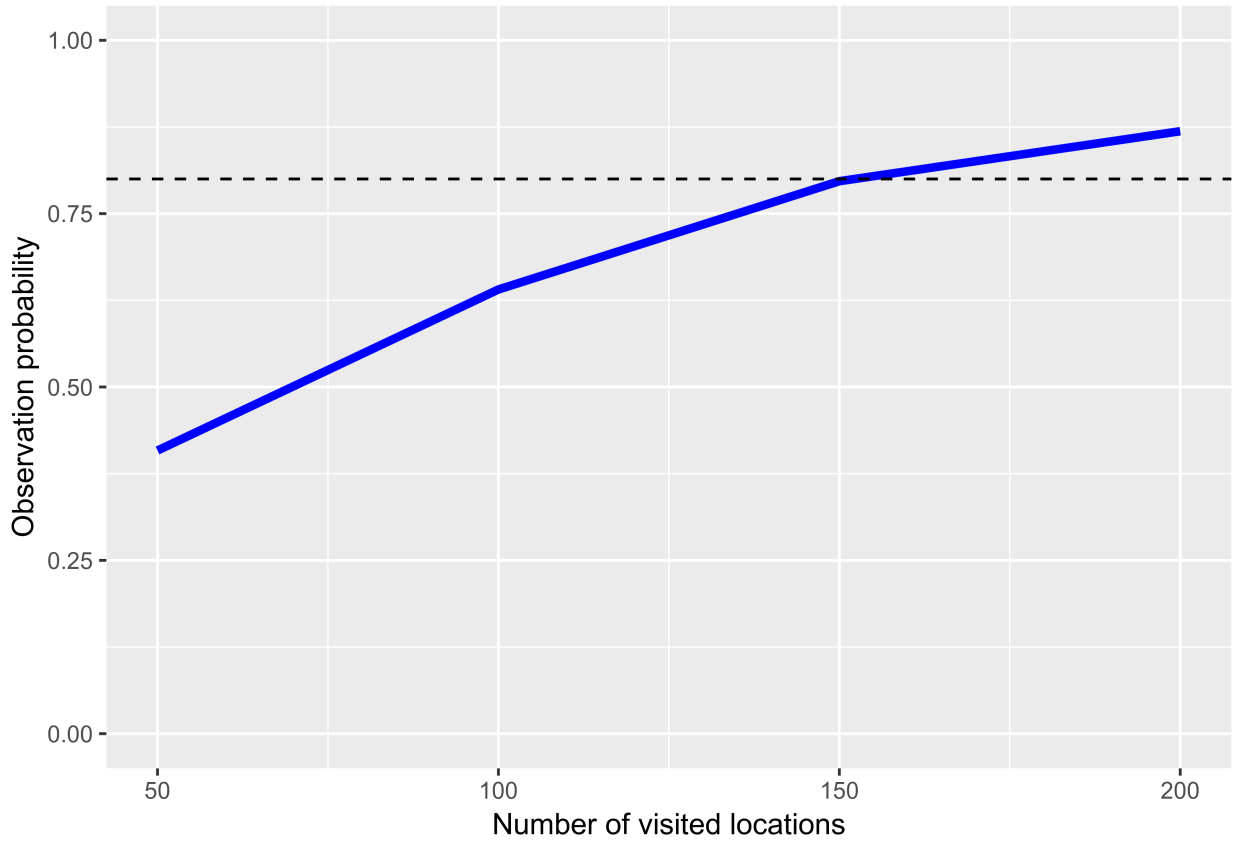


Figure 9: 50 out of 4057 grid cells occupied according to weights, 1 visit with 1 detection probability

But what happens when we have information about the occurence of the species? In effect, we limit the number of potential sites we visit to a smaller value, which have higher probability of housing the species. We use the prediction map to set the occurrences, and visitation probabilities. We continue with the detection probability set to 1, with just 1 visit per site.

```
system.time(predProb <- weightedDetection(occWeights = predWeights, visWeights = predWeights,
    noOccur = 5, noLocations = seq(50, 200, by = 50), noVisits = 1, detectProb = 1,
    nIter = 999))
```

```
##    user  system elapsed
##   3.720   0.004   3.725
```

```
predProb
```

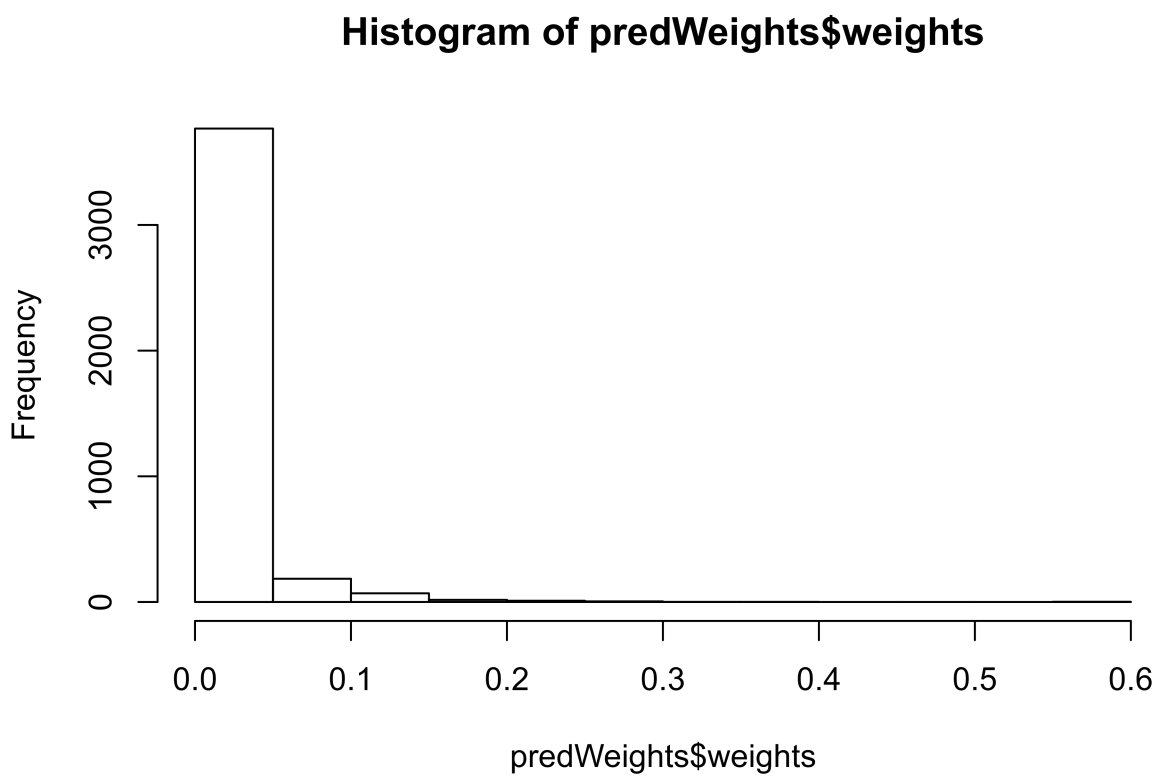
```
## # A tibble: 4 x 2
##   noLocations probObs
##         <dbl>   <dbl>
## 1        50.0   0.408
## 2       100     0.641
## 3       150     0.797
## 4       200     0.869
```

```
plot(predProb, threshold = 0.8)
```

We then reach quite high probabilities for observing the species. This of course depend on the quality of

the predictions, i.e. our weights. This prediction map is actually quite informative. From the histogram of weights, we can see that a small number of sites have a proportionally high weight. This limits the realised occurrences quite a bit.

```
hist(predWeights$weights)
```



Calculations based on 1km map

```
require(RPostgreSQL)
require(DBI)
require(rpostgis)
require(SurveyPower)
require(tidyverse)
data(map1km)


con <- dbConnect(RPostgreSQL::PostgreSQL(), host = "gisdata-db.nina.no", dbname = "gisdata",
    user = "postgjest", password = "gjestpost")
# pred <- pgGetRast(con, name = c('hotspot_ias', 'bigPred1km'), rast =
# 'rast', bands = 1, boundary = NULL)

predQ <- "SELECT ssbid, ST_Value(pred.rast, ST_Centroid(ssb.geom)) pred
FROM ssb_data_utm33n.ssb_1km ssb,
hotspot_ias.\"evenintbigpred1km\" pred
WHERE ST_Intersects(ssb.geom, pred.rast)
```

```
"

pred <- dbGetQuery(con, predQ)

pred <- pred %>% mutate(ssbid = as.character(ssbid))

predMap1km <- map1km %>% left_join(pred, by = c(ssbid = "ssbid"))

predMap1km$pred[is.na(predMap1km$pred)] <- 0

# plot(predMap1km['pred']) #Slow
```

```
predWeights1km <- predMap1km %>% select(sites = ssbid, weights = pred) %>% sf::st_set_geometry(NULL)

devtools::use_data(predWeights1km)
```

Since we now have about 100 times as many potential sites, we would need to increase the occurrences accordingly. But while 50 out of 4057 10x10km squares sounds reasonable for an "early" detection, multiplying this with 100 yields 5000 locations in a 1x1km grid. This sounds like a lot for an early detection. But it puts the 50 occurrences above into perspective. Surveying 10x10km cells with good detection probability is a tall order.

For the 1x1km analysis, the calculations takes to much time to do on the fly, so we pre-calculate them and load the results.

```
data("predWeights1km")
```

It is reasonable to assume that we won't reach a higher detection probability than 0.8 for a single visit, and even that is probably high for a truly novel species. But we can explore the range of occupied sites that are reasonable to handle with such a good detection probability.

```
system.time(predOcc500Det0.8 <- weightedDetection(occWeights = predWeights1km,
    visWeights = predWeights1km, noOccur = 500, noLocations = seq(50, 300, by = 50),
    noVisits = 1, detectProb = 0.8, nIter = 999))

save(predOcc500Det0.8, file = "predOcc500Det0.8.Rdata")
```

```
load(file = "predOcc500Det0.8.Rdata")
predOcc500Det0.8
```

```
## # A tibble: 6 x 2
##   noLocations probObs
##         <dbl>   <dbl>
## 1        50.0   0.343
## 2       100     0.538
## 3       150     0.691
## 4       200     0.798
## 5       250     0.866
## 6       300     0.913
```

```
plot(predOcc500Det0.8, threshold = 0.8, xlab = "Antall lokaliteter", ylab = "Observasjonssannsynlighet")
```

So for the case of 500 occupied cells, we reach the target observation probability after visiting about 200 sites. In similar fashion, we can explore the case with 200 and 100 occupied cells, respectively.

```
system.time(predOcc200Det0.8 <- weightedDetection(occWeights = predWeights1km,
    visWeights = predWeights1km, noOccur = 200, noLocations = seq(50, 600, by = 50),
```

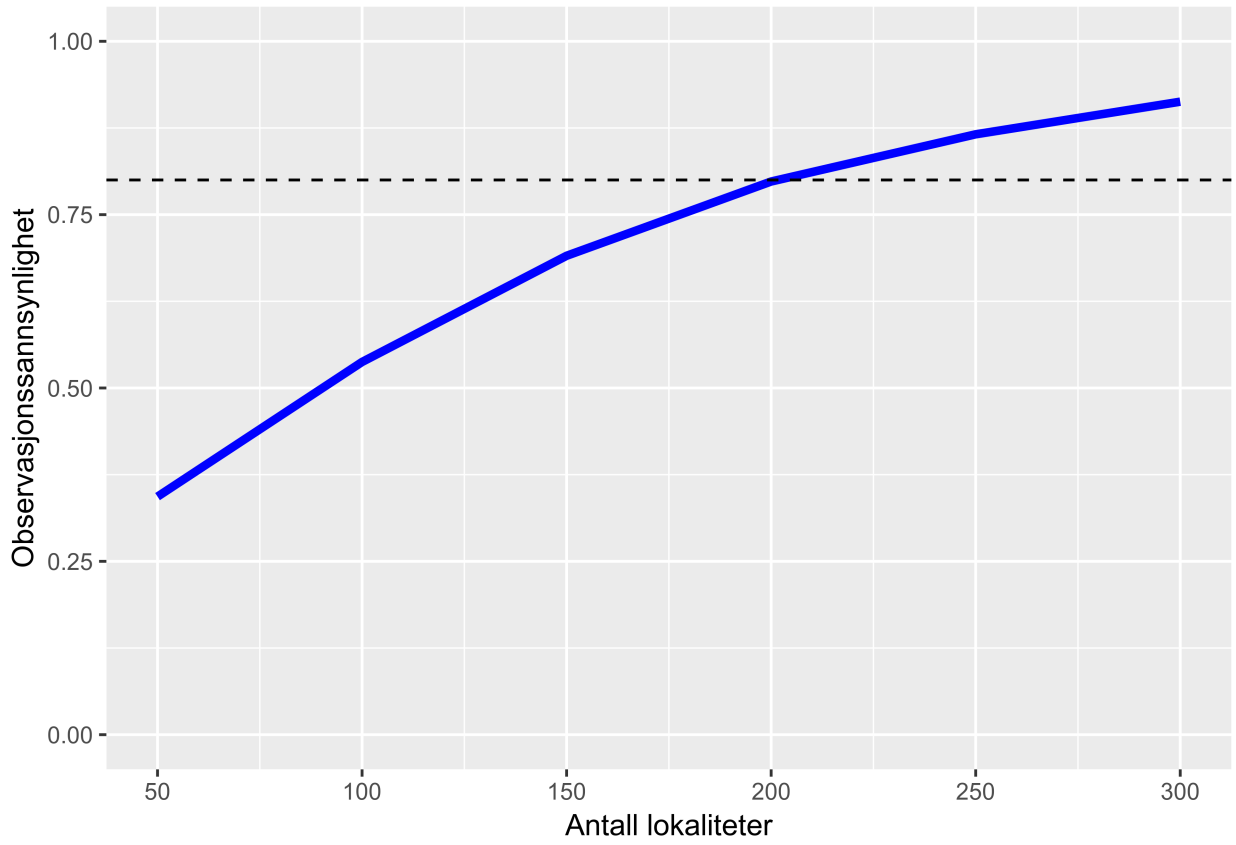


Figure 10: Estimated observation probability of an alien vascular plant species occurring in 500 1x1km grid cells as a function of the number of visited locations. Occurrences and location selection is based on the same weights, modelled from actual alien vascular plant species occurrences. Each visit has a 0.8 probability of detecting the species if present. The threshold of the desired observation probabilitiy of 0.8 is shown as a dashed line.

```
    noVisits = 1, detectProb = 0.8, nIter = 999))

save(predOcc200Det0.8, file = "predOcc200Det0.8.Rdata")
```

```
load(file = "predOcc200Det0.8.Rdata")
xtable(predOcc200Det0.8)
```

noLocations	probObs
50.00	0.13
100.00	0.28
150.00	0.40
200.00	0.48
250.00	0.52
300.00	0.63
350.00	0.68
400.00	0.73
450.00	0.76
500.00	0.81
550.00	0.83
600.00	0.88

```
plot(predOcc200Det0.8, threshold = 0.8, xlab = "Antall lokaliteter", ylab = "Observasjonssannsynlighet")
```

```
system.time(predOcc100Det0.8 <- weightedDetection(occWeights = predWeights1km,
    visWeights = predWeights1km, noOccur = 100, noLocations = seq(50, 600, by = 50),
    noVisits = 1, detectProb = 0.8, nIter = 999))

save(predOcc100Det0.8, file = "predOcc100Det0.8.Rdata")
```

```
load(file = "predOcc100Det0.8.Rdata")

xtable(predOcc100Det0.8)
```

noLocations	probObs
50.00	0.09
100.00	0.15
150.00	0.22
200.00	0.26
250.00	0.31
300.00	0.37
350.00	0.39
400.00	0.46
450.00	0.53
500.00	0.58
550.00	0.61
600.00	0.62

```
plot(predOcc100Det0.8, threshold = 0.8)
```

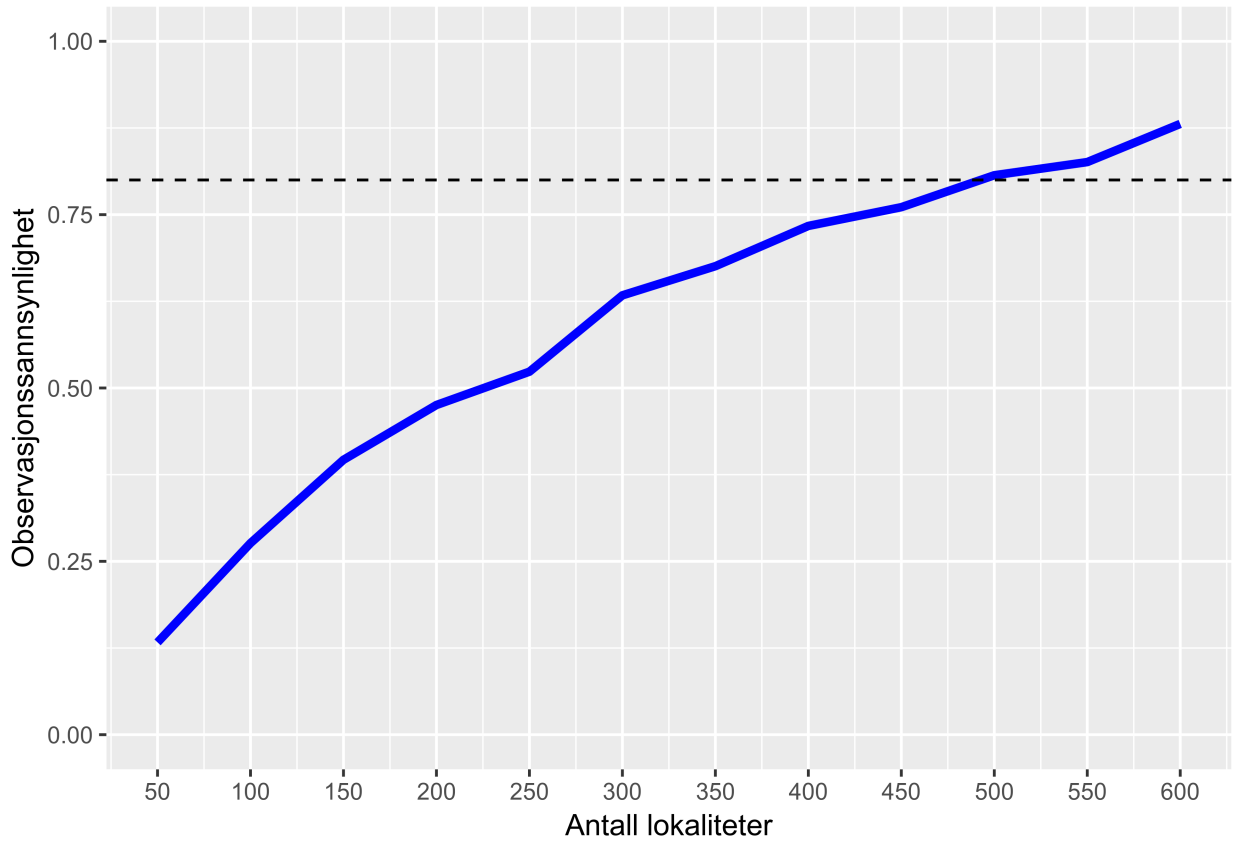
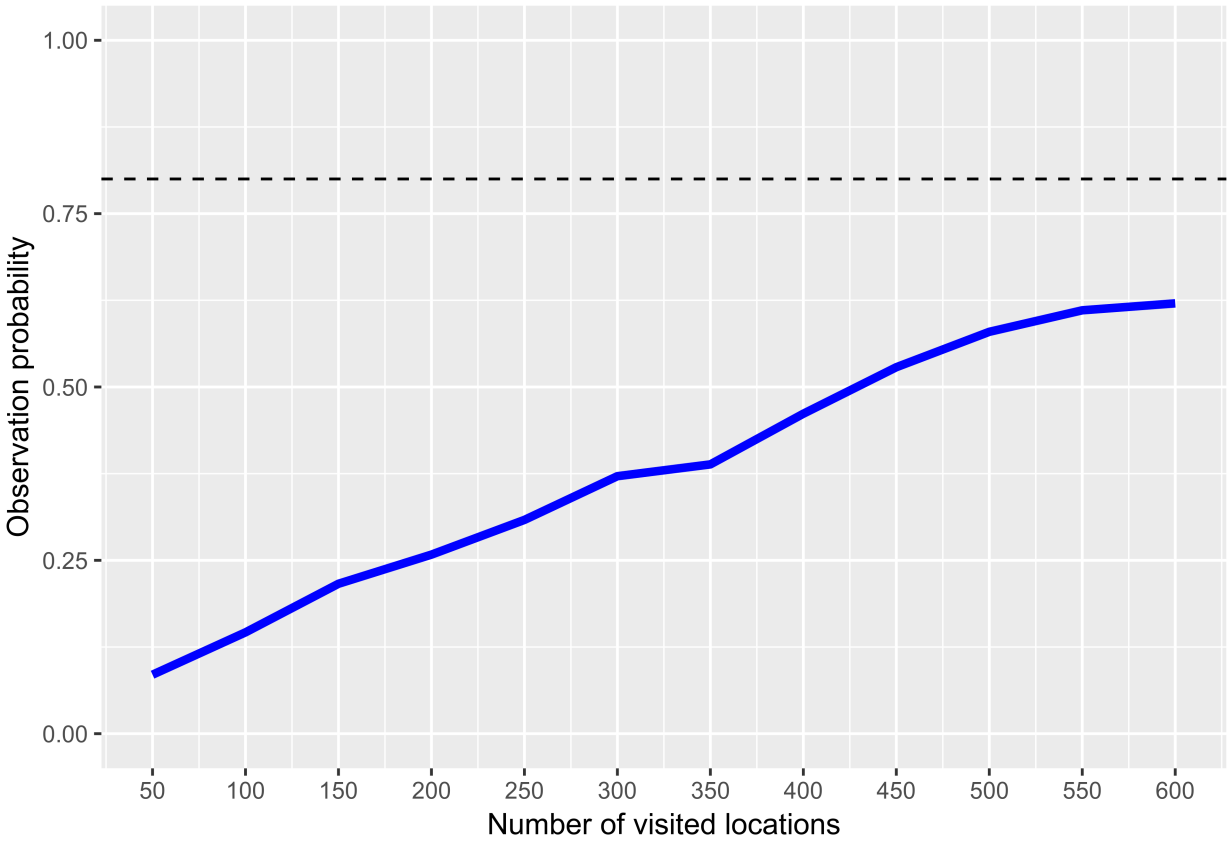


Figure 11: Estimated observation probability of an alien vascular plant species occurring in 200 1x1km grid cells as a function of the number of visited locations. Occurrences and location selection is based on the same weights, modelled from actual alien vascular plant species occurrences. Each visit has a 0.8 probability of detecting the species if present. The threshold of the desired observation probabilitiy of 0.8 is shown as a dashed line.



In the case of 200 occupied cells, we reach an overall observation probability of 0.8 after about 500 visited sites with an observation probability of 0.8. With only 100 occupied cells, we don't reach the target of 0.8 even after 600 visited locations.

We can see the effect of a lower detection probability.

Surveying 250x250m squares

In practice, it can be challenging to survey even a 1x1km grid with any respectable observation probability. We might therefore subdivide the squares in smaller units, with the result that our detection probability decreases from simply not covering the place where the species occupies. If a species just occurs in one out of 16 250x250 squares within a 1x1km square and we visit only one such smaller square, our detection probability drops to 1/16 of the former level. How much the detection probability drops of course depend on the aggregation pattern of the species, i.e. how much of the 1x1km cell it is present in. On a tangent, the number of occurrences we consider to be acceptable within an "early detection" framework depends on the scale of the grid cells considered, and the way that the species are aggregated. 500 out of ca 50 000 1x1km cells constitutes occurrences in 1% of the cells. If we accept a 1% occurrence in the about 800 000 250x250 cells, that amounts to 8000 occurrences. As long as these are not extremy aggregated, this could hardly be seen as an early establishment phase.

```
load("predOcc500Det0.05.Rdata")
load("predOcc500Det0.05Vis4.Rdata")
load("predOcc500Det0.05Vis10.Rdata")
load("predOcc500Det0.025Vis4.Rdata")
load("predOcc500Det0.025Vis10.Rdata")
load("predOcc100Det0.05Vis10.Rdata")
load("predOcc100Det0.05Vis4.Rdata")
```

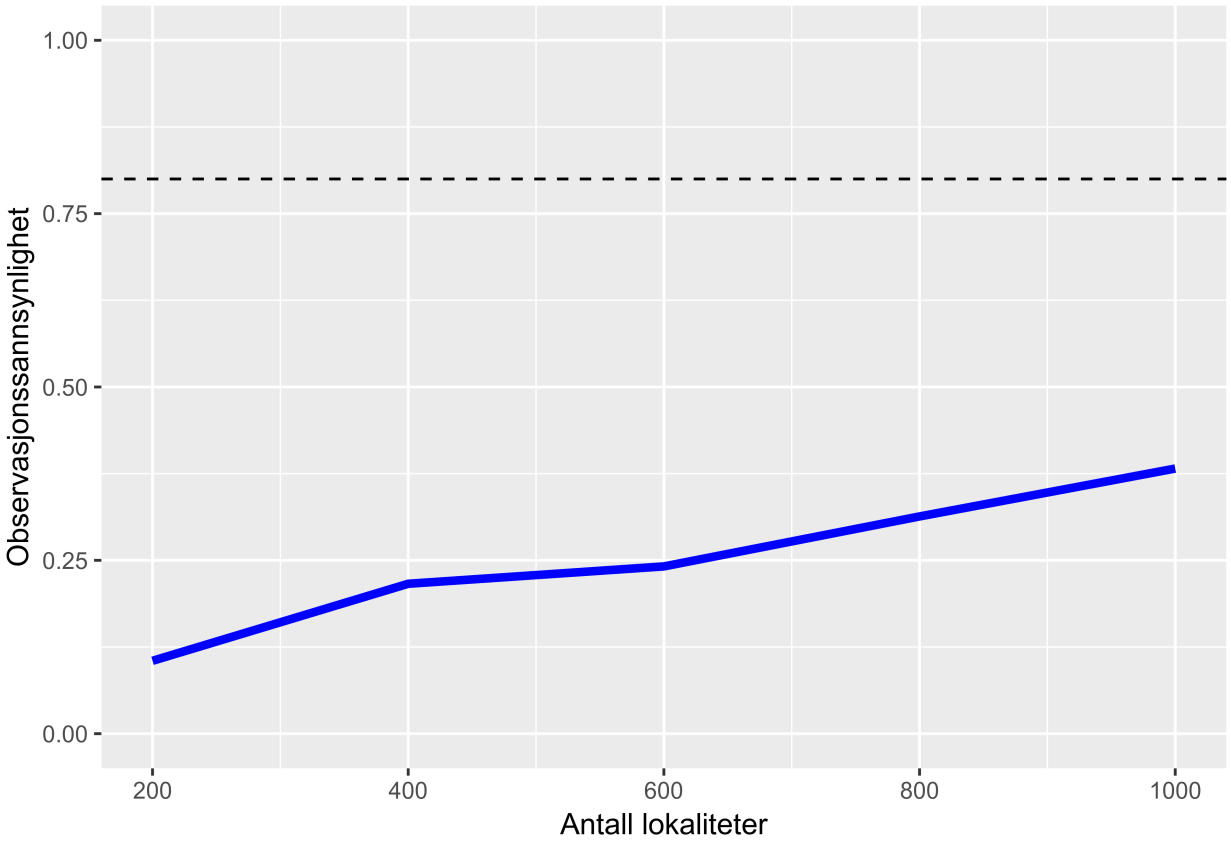


Figure 12: Probability of detecting a species at least once that is present in 500 of the 1x1km cells, but we survey only a 1/16 of the cell.

```
load("predOcc100Det0.05Vis1.Rdata")
load("predOcc500Det0.2.Rdata")
load("predOcc500Det0.5.Rdata")
```

Case of 500 1x1km cells occupied, but we survey only 250x250m subsquares

We start with the case of 500 occurrences spread out in the 50 000 1x1km cells, but when we visit just a 16th of these cells once.

```
plot(predOcc500Det0.05, threshold = 0.8, xlab = "Antall lokaliteter", ylab = "Observasjonssannsynlighet
```

It is clear that these conditions does not let us reach the desired detection probability of 0.8 even with a great number of sampled locations.

We can continue to explore the possibilities with a lower total occurrence, for example when a species is present in 100 out of the 50 000 1x1km grid cells.

```
plot(predOcc100Det0.05Vis1, threshold = 0.8, xlab = "Antall lokaliteter", ylab = "Observasjonssannsynli
```

This situation leaves us with very slim chances of detecting the species.

We can mitigate these low numbers by surveying the same 1x1km square multiple times. Note that I here assume that the species is stationary in only one place, and we return to different subplots within the 1x1km

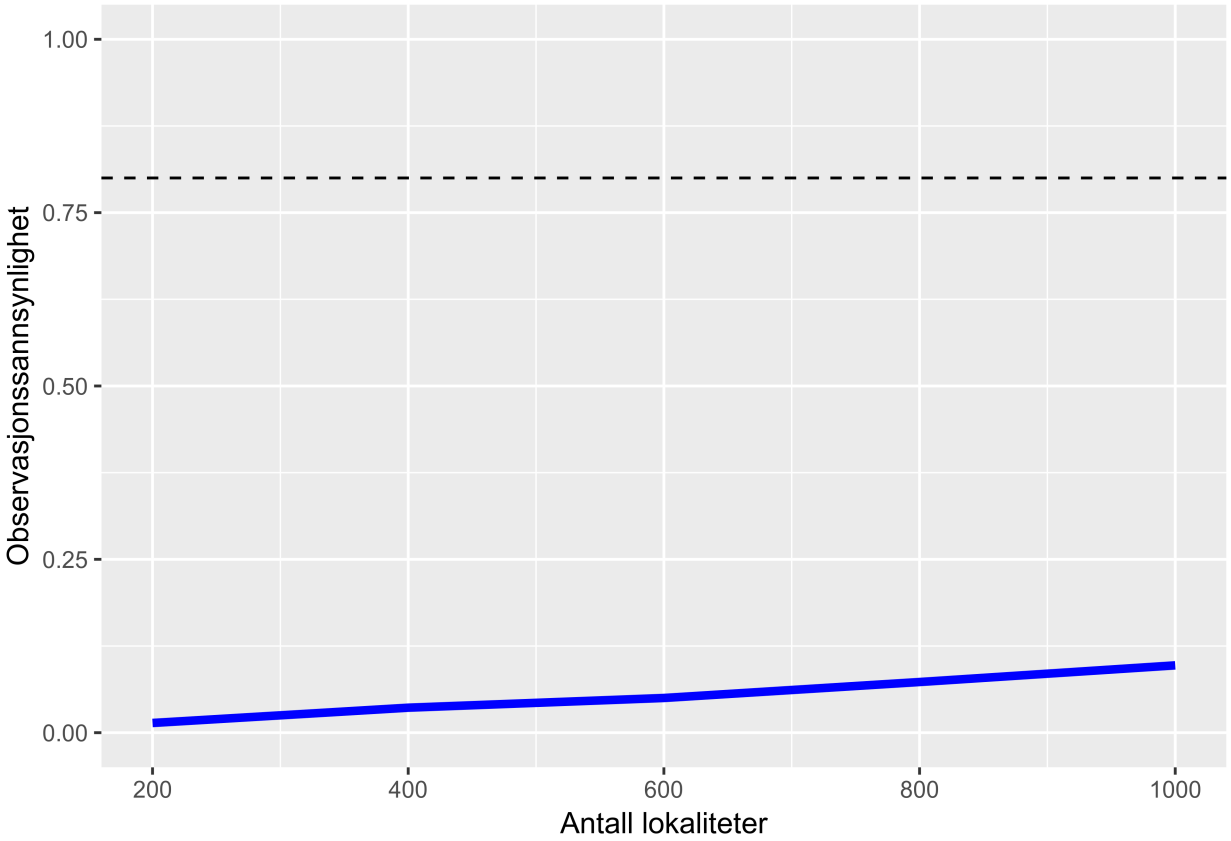


Figure 13: Probability of detecting a species at least once that is present in 100 of the 1x1km cells, but1 we survey only a 1/16 of the cell.

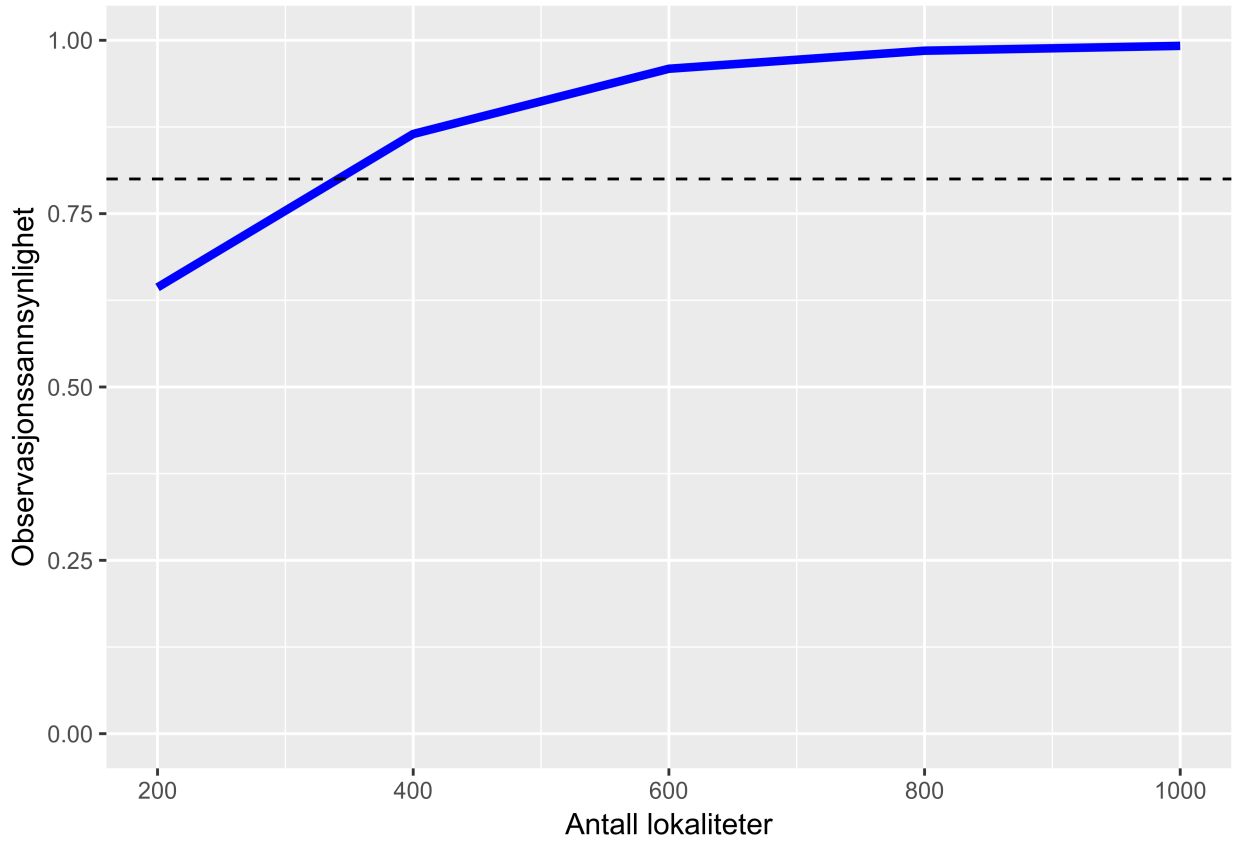


Figure 14: Probability of detecting a species at least once that is present in 500 of the 1x1km cells, but we survey 10 1/16th of the cell. Here we visit the 1x1km cell 10 times.

square, so that we cover an increasing portion of the sample location. The detectability thus increases linearly. For example, if we have a detection probability of 0.8 in a given location, but we only visit 10 1/16th of that location, we get a detection probability of $0.8/16*10 = 0.5$. *If we would instead visit 2 subsquares of 1/16, we would get* $0.8/16*4 = 0.2$

```
plot(predOcc500Det0.5, threshold = 0.8, xlab = "Antall lokaliteter", ylab = "Observasjonssannsynlighet"
```

```
plot(predOcc500Det0.2, threshold = 0.8, xlab = "Antall lokaliteter", ylab = "Observasjonssannsynlighet"
```

Surveys limited to a specific area

One strategy is to limit the surveyed areas to focus the efforts. We may thereby increase our chances to visit a location that is occupied by a rare species, but we also cannot find species that are not present in the area of interest.

This situation can be modelled using the same function `weightedDetection`, by setting the visitation probability to zero for the locations outside our focus area. Based on the prediction map, I have selected a set of kommunes around Oslo that seems to encompass the areas with highest predictions.

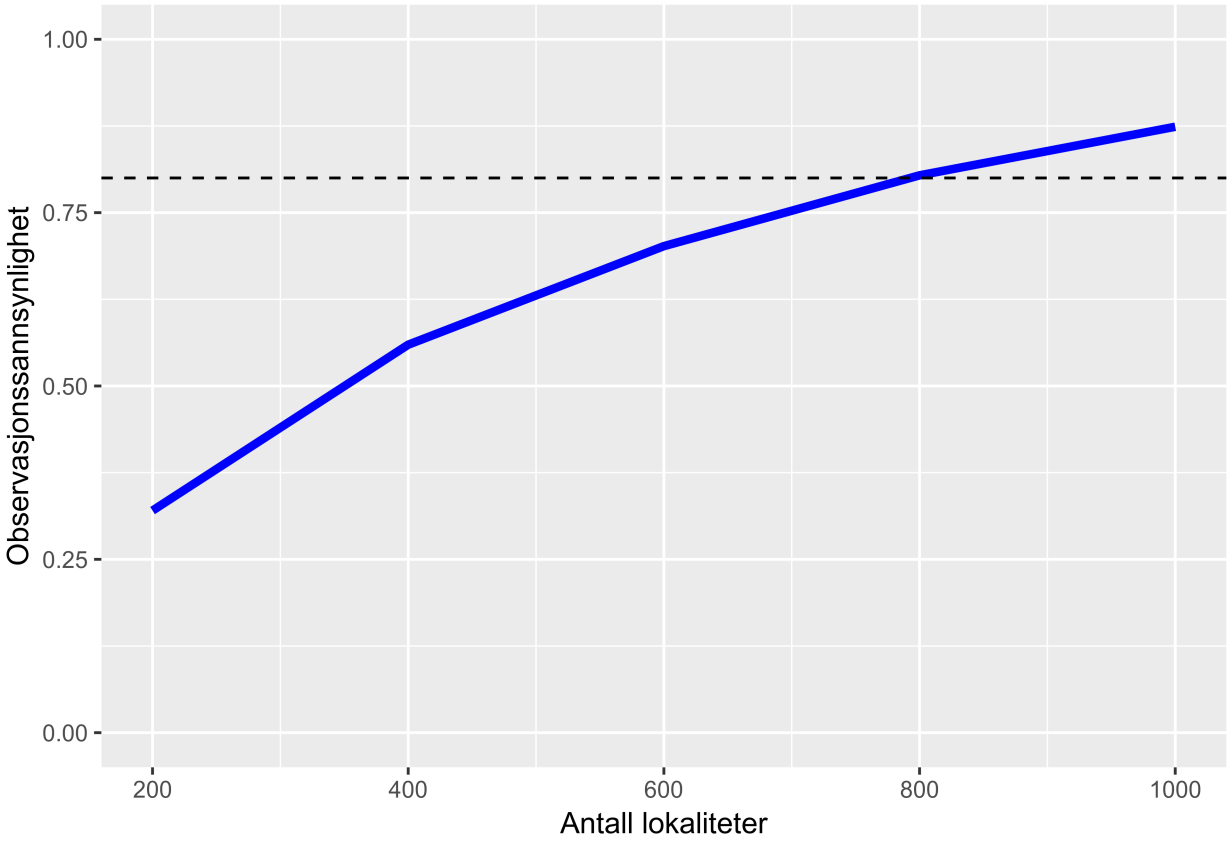
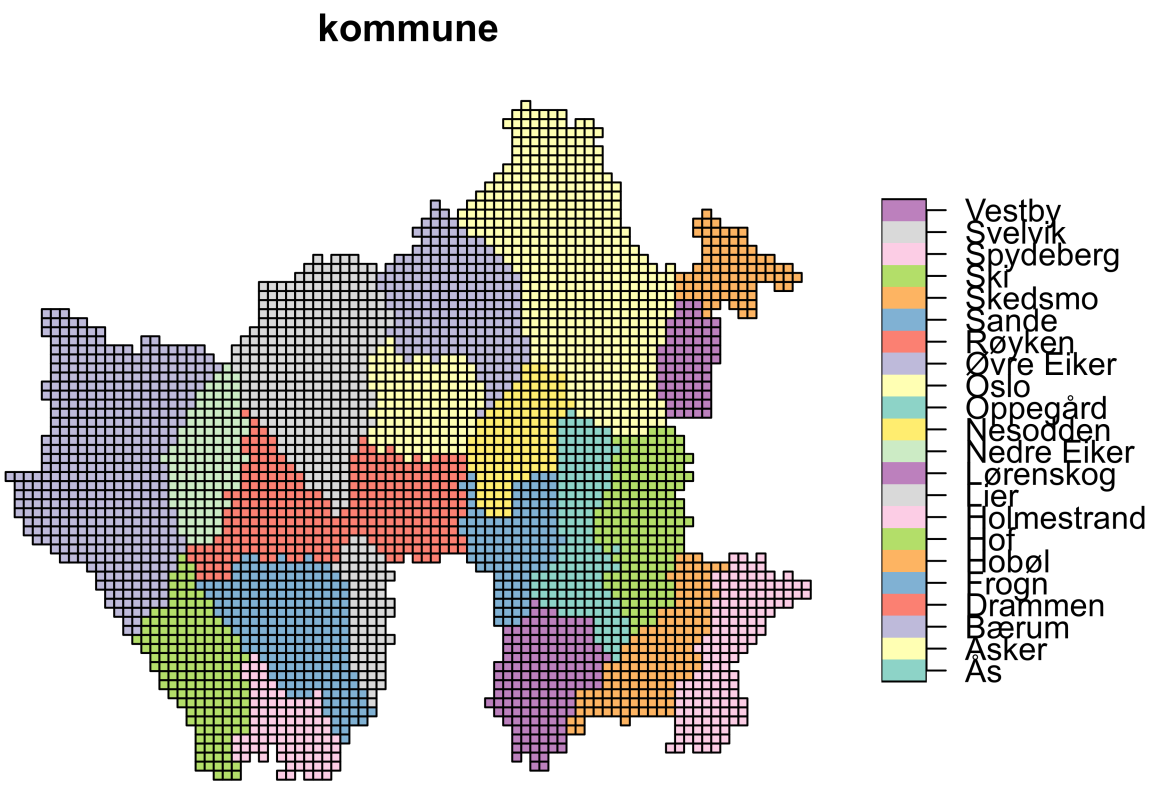


Figure 15: Probability of detecting a species at least once that is present in 500 of the 1x1km cells, but we survey 4 1/16th of the cell. Here we visit the 1x1km cell 10 times.

Oslo region

```
occWeights <- predWeights1km

osloArea <- c("Vestby", "Svelvik", "Lørenskog", "Frogn", "Ås", "Skedsmo",
    "Oppegård", "Ski", "Hobøl", "Spydeberg", "Sande", "Oslo", "Bærum", "Asker",
    "Lier", "Øvre Eiker", "Nedre Eiker", "Drammen", "Hof", "Holmestrand", "Nesodden",
    "Frogn", "Røyken")

osloAreaKommuneNr <- c("0211", "0711", "0230", "0215", "0214", "0231", "0217",
    "0213", "0138", "0123", "0713", "0301", "0219", "0220", "0626", "0624",
    "0625", "0602", "0714", "0702", "0216", "0215", "0627")

ssbidToVisit <- map1km %>% filter(KOMMUNENUMMER %in% osloAreaKommuneNr) %>%
    select(ssbid, kommune)


plot(ssbidToVisit["kommune"], key.width = lcm(4), key.pos = 4)
```




```
osloVisWeights <- occWeights
osloVisWeights$weights[!(osloVisWeights$sites %in% ssbidToVisit$ssbid)] <- 0
summary(osloVisWeights$weights)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## 0.0000000 0.0000000 0.0000000 0.0003481 0.0000000 3.0014784
```

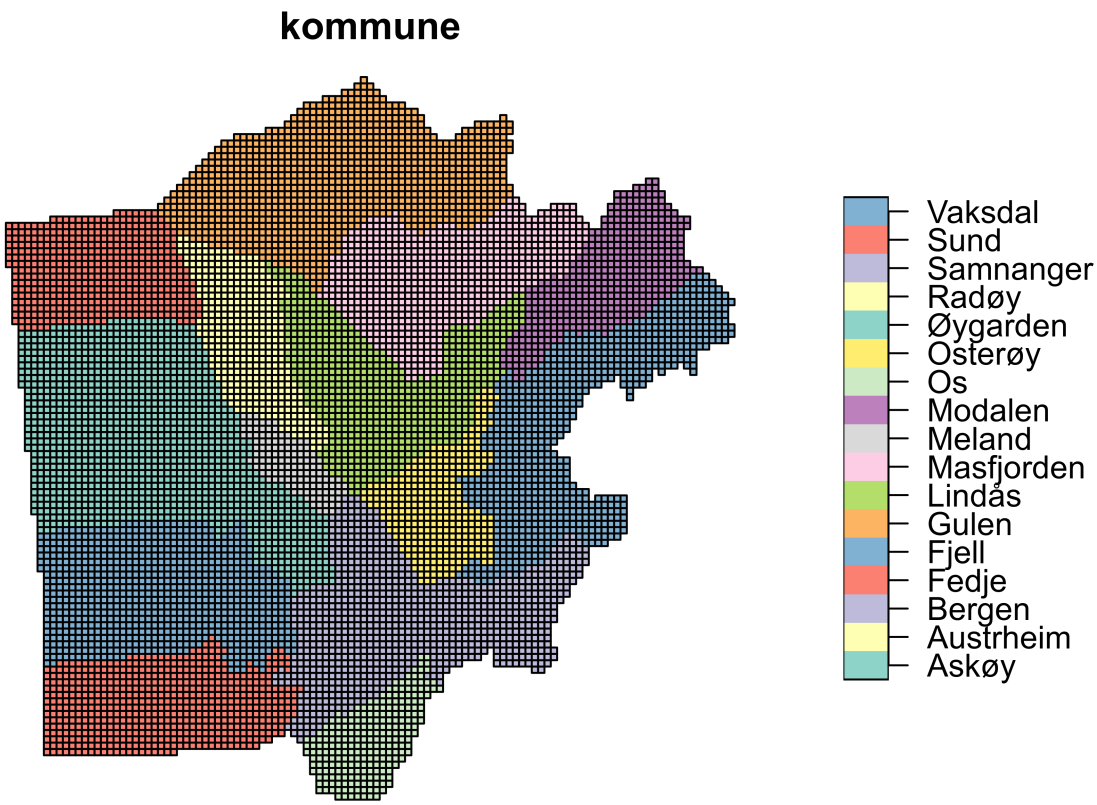
```
summary(occWeights$weights)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## 0.0000000 0.0000000 0.0000064 0.0089409 0.0041522 3.0014784
```

Bergen

One of several possible demarcations of "Bergen area"

```
bergenKommuneNr <- c("1265", "1259", "1253", "1260", "1263", "1251", "1411",
    "1252", "1256", "1266", "1264", "1242", "1247", "1201", "1243", "1246",
    "1245")

ssbidBergen <- map1km %>% filter(KOMMUNENUMMER %in% bergenKommuneNr) %>% select(ssbid,
    kommune)


plot(ssbidBergen["kommune"], key.width = lcm(4), key.pos = 4)
```




```
bergenVisWeights <- occWeights
bergenVisWeights$weights[!(bergenVisWeights$sites %in% ssbidBergen$ssbid)] <- 0
summary(bergenVisWeights$weights)
```

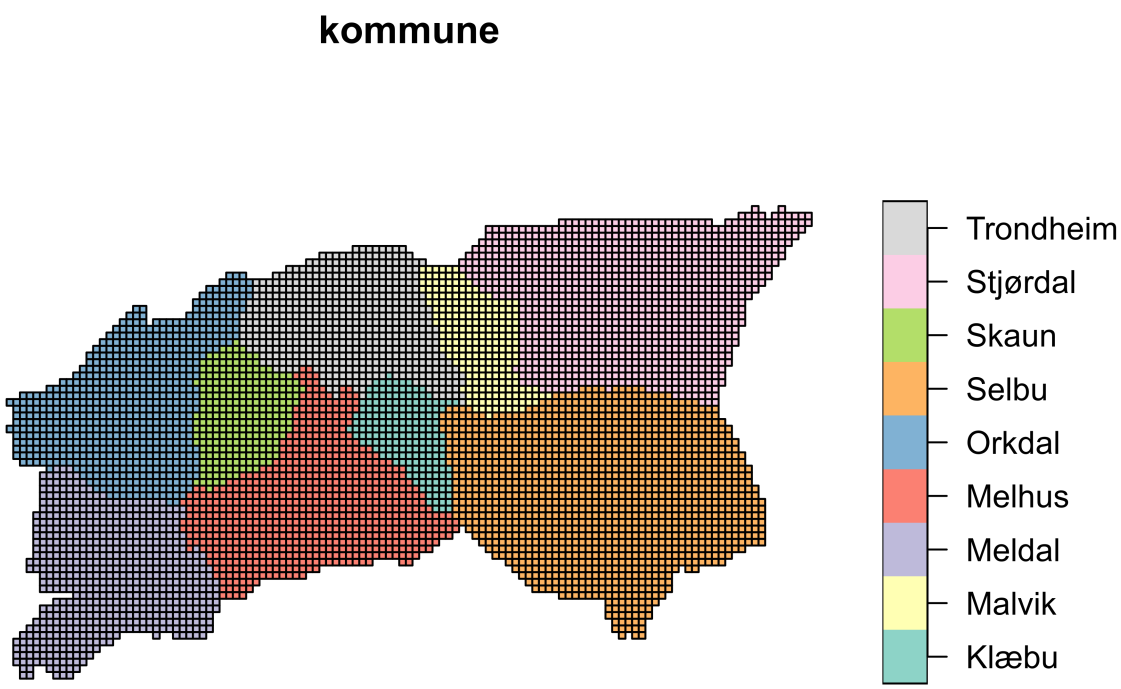
```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## 0.0000000 0.0000000 0.0000000 0.0004596 0.0000000 0.5579212
```

```
summary(occWeights$weights)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## 0.0000000 0.0000000 0.0000064 0.0089409 0.0041522 3.0014784
```

Trondheim

```
trondheimKommuneNr <- c("1638", "1657", "1636", "1664", "1662", "1601", "1663",
    "1714", "1653")


ssbidTrondheim <- map1km %>% filter(KOMMUNENUMMER %in% trondheimKommuneNr) %>%
    select(ssbid, kommune)


plot(ssbidTrondheim["kommune"], key.width = lcm(4), key.pos = 4)
```




```
trondheimVisWeights <- occWeights
trondheimVisWeights$weights[!(trondheimVisWeights$sites %in% ssbidTrondheim$ssbid)] <- 0
summary(trondheimVisWeights$weights)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## 0.0000000 0.0000000 0.0000000 0.0002464 0.0000000 0.3133284
```

```
summary(occWeights$weights)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0000000 0.0000064 0.0089409 0.0041522 3.0014784
```

Run calculations

```
oslo47Sites <- weightedDetection(occWeights = occWeights, visWeights = osloVisWeights,
    noOccur = 500, noLocations = 47, detectProb = 0.426, nIter = 999, noVisits = 1)

oslo47Sites16 <- weightedDetection(occWeights = occWeights, visWeights = osloVisWeights,
    noOccur = 500, noLocations = 47, detectProb = 0.426/16, nIter = 999, noVisits = 1)

oslo100Sites16 <- weightedDetection(occWeights = occWeights, visWeights = osloVisWeights,
    noOccur = 500, noLocations = 100, detectProb = 0.426/16, nIter = 999, noVisits = 1)


oslo66Sites16 <- weightedDetection(occWeights = occWeights, visWeights = osloVisWeights,
    noOccur = 500, noLocations = 66, detectProb = 0.426 * 1.5/16, nIter = 999,
    noVisits = 1)

oslo200Sites16 <- weightedDetection(occWeights = occWeights, visWeights = osloVisWeights,
    noOccur = 500, noLocations = 200, detectProb = 0.426/16, nIter = 999, noVisits = 1)

oslo140Sites16 <- weightedDetection(occWeights = occWeights, visWeights = osloVisWeights,
    noOccur = 500, noLocations = 140, detectProb = 0.426 * 1.5, nIter = 999,
    noVisits = 1)

## Bergen
bergen47Sites <- weightedDetection(occWeights = occWeights, visWeights = bergenVisWeights,
    noOccur = 500, noLocations = 47, detectProb = 0.426, nIter = 999, noVisits = 1)

bergen47Sites16 <- weightedDetection(occWeights = occWeights, visWeights = bergenVisWeights,
    noOccur = 500, noLocations = 47, detectProb = 0.426/16, nIter = 999, noVisits = 1)

bergen100Sites16 <- weightedDetection(occWeights = occWeights, visWeights = bergenVisWeights,
    noOccur = 500, noLocations = 100, detectProb = 0.426/16, nIter = 999, noVisits = 1)


bergen66Sites16 <- weightedDetection(occWeights = occWeights, visWeights = bergenVisWeights,
    noOccur = 500, noLocations = 66, detectProb = 0.426 * 1.5/16, nIter = 999,
    noVisits = 1)

bergen200Sites16 <- weightedDetection(occWeights = occWeights, visWeights = bergenVisWeights,
    noOccur = 500, noLocations = 200, detectProb = 0.426/16, nIter = 999, noVisits = 1)

bergen140Sites16 <- weightedDetection(occWeights = occWeights, visWeights = bergenVisWeights,
    noOccur = 500, noLocations = 140, detectProb = 0.426 * 1.5, nIter = 999,
    noVisits = 1)




## Trondheim

trondheim47Sites <- weightedDetection(occWeights = occWeights, visWeights = trondheimVisWeights,
    noOccur = 500, noLocations = 47, detectProb = 0.426, nIter = 999, noVisits = 1)
```

```
trondheim47Sites16 <- weightedDetection(occWeights = occWeights, visWeights = trondheimVisWeights,
    noOccur = 500, noLocations = 47, detectProb = 0.426/16, nIter = 999, noVisits = 1)

trondheim100Sites16 <- weightedDetection(occWeights = occWeights, visWeights = trondheimVisWeights,
    noOccur = 500, noLocations = 100, detectProb = 0.426/16, nIter = 999, noVisits = 1)


trondheim66Sites16 <- weightedDetection(occWeights = occWeights, visWeights = trondheimVisWeights,
    noOccur = 500, noLocations = 66, detectProb = 0.426 * 1.5/16, nIter = 999,
    noVisits = 1)

trondheim200Sites16 <- weightedDetection(occWeights = occWeights, visWeights = trondheimVisWeights,
    noOccur = 500, noLocations = 200, detectProb = 0.426/16, nIter = 999, noVisits = 1)

trondheim140Sites16 <- weightedDetection(occWeights = occWeights, visWeights = trondheimVisWeights,
    noOccur = 500, noLocations = 140, detectProb = 0.426 * 1.5, nIter = 999,
    noVisits = 1)



# Norge
norge47Sites <- weightedDetection(occWeights = occWeights, visWeights = occWeights,
    noOccur = 500, noLocations = 47, detectProb = 0.426, nIter = 999, noVisits = 1)

norge47Sites16 <- weightedDetection(occWeights = occWeights, visWeights = occWeights,
    noOccur = 500, noLocations = 47, detectProb = 0.426/16, nIter = 999, noVisits = 1)

norge100Sites16 <- weightedDetection(occWeights = occWeights, visWeights = occWeights,
    noOccur = 500, noLocations = 100, detectProb = 0.426/16, nIter = 999, noVisits = 1)


norge66Sites16 <- weightedDetection(occWeights = occWeights, visWeights = occWeights,
    noOccur = 500, noLocations = 66, detectProb = 0.426 * 1.5/16, nIter = 999,
    noVisits = 1)

norge200Sites16 <- weightedDetection(occWeights = occWeights, visWeights = occWeights,
    noOccur = 500, noLocations = 200, detectProb = 0.426/16, nIter = 999, noVisits = 1)

norge140Sites16 <- weightedDetection(occWeights = occWeights, visWeights = occWeights,
    noOccur = 500, noLocations = 140, detectProb = 0.426 * 1.5/16, nIter = 999,
    noVisits = 1)


## small expensive 2 visits, 1.5 detection prob

norge50Sites16Vis2 <- weightedDetection(occWeights = occWeights, visWeights = occWeights,
    noOccur = 500, noLocations = 50, detectProb = 0.426 * 1.5/16 * 2, nIter = 999,
    noVisits = 1)

oslo50Sites16Vis2 <- weightedDetection(occWeights = occWeights, visWeights = osloVisWeights,
    noOccur = 500, noLocations = 50, detectProb = 0.426 * 1.5/16 * 2, nIter = 999,
    noVisits = 1)
```

```
bergen50Sites16Vis2 <- weightedDetection(occWeights = occWeights, visWeights = bergenVisWeights,
    noOccur = 500, noLocations = 50, detectProb = 0.426 * 1.5/16 * 2, nIter = 999,
    noVisits = 1)


trondheim50Sites16Vis2 <- weightedDetection(occWeights = occWeights, visWeights = trondheimVisWeights,
    noOccur = 500, noLocations = 50, detectProb = 0.426 * 1.5/16 * 2, nIter = 999,
    noVisits = 1)


save(oslo47Sites, oslo47Sites16, oslo66Sites16, oslo100Sites16, oslo140Sites16,
    oslo200Sites16, oslo50Sites16Vis2, bergen47Sites, bergen47Sites16, bergen66Sites16,
    bergen100Sites16, bergen140Sites16, bergen200Sites16, bergen50Sites16Vis2,
    trondheim47Sites, trondheim47Sites16, trondheim66Sites16, trondheim100Sites16,
    trondheim140Sites16, trondheim200Sites16, trondheim50Sites16Vis2, norge47Sites,
    norge47Sites16, norge66Sites16, norge100Sites16, norge140Sites16, norge200Sites16,
    norge50Sites16Vis2, file = "survey_calc.Rdata")
```

```
load(file = "survey_calc.Rdata")
```

```
surTab <- tibble(Areal = rep(c("Norge", "Oslo", "Bergen", "Trondheim"), times = 6),
    `Antall lokaliteter` = rep(c(47, 66, 100, 140, 200, 50), each = 4), `Antall besøk` = rep(c(1,
        1, 1, 1, 1, 2), each = 4), `Deteksjonsrate 250x250m` = rep(c(0.426,
        0.426 * 1.5, 0.426, 0.426 * 1.5, 0.426, 0.426 * 1.5), each = 4))
```

```
surTab <- surTab %>% mutate(`Deteksjonerate 1x1km` = `Deteksjonsrate 250x250m`/16 *
    `Antall besøk`, Oppdagbarhet = c(norge47Sites16$probObs, oslo47Sites16$probObs,
    bergen47Sites16$probObs, trondheim47Sites16$probObs, norge66Sites16$probObs,
    oslo66Sites16$probObs, bergen66Sites16$probObs, trondheim66Sites16$probObs,
    norge100Sites16$probObs, oslo100Sites16$probObs, bergen100Sites16$probObs,
    trondheim100Sites16$probObs, norge140Sites16$probObs, oslo140Sites16$probObs,
    bergen140Sites16$probObs, trondheim140Sites16$probObs, norge200Sites16$probObs,
    oslo200Sites16$probObs, bergen200Sites16$probObs, trondheim200Sites16$probObs,
    norge50Sites16Vis2$probObs, oslo50Sites16Vis2$probObs, bergen50Sites16Vis2$probObs,
    trondheim50Sites16Vis2$probObs))

# surTab
```

```
xtable(surTab, caption = "Detection probabilities of a handful of survey regimes, with detection probabi
```

How much less is the prob using a subarea? (not much difference it seems)

References

Pavlos S. Efraimidis, Paul G. Spirakis, Weighted random sampling with a reservoir, Information Processing Letters, Volume 97, Issue 5, 16 March 2006, Pages 181-185, ISSN 0020-0190, 10.1016/j.ipl.2005.11.003.

Areal	Antall lokaliteter	Antall besøk	Deteksjonsrate 250x250m	Deteksjonerate 1x1km	Oppdagbarhet
Norge	47.00	1.00	0.43	0.03	0.01
Oslo	47.00	1.00	0.43	0.03	0.03
Bergen	47.00	1.00	0.43	0.03	0.02
Trondheim	47.00	1.00	0.43	0.03	0.01
Norge	66.00	1.00	0.64	0.04	0.03
Oslo	66.00	1.00	0.64	0.04	0.04
Bergen	66.00	1.00	0.64	0.04	0.04
Trondheim	66.00	1.00	0.64	0.04	0.02
Norge	100.00	1.00	0.43	0.03	0.02
Oslo	100.00	1.00	0.43	0.03	0.04
Bergen	100.00	1.00	0.43	0.03	0.03
Trondheim	100.00	1.00	0.43	0.03	0.02
Norge	140.00	1.00	0.64	0.04	0.05
Oslo	140.00	1.00	0.64	0.04	0.77
Bergen	140.00	1.00	0.64	0.04	0.70
Trondheim	140.00	1.00	0.64	0.04	0.52
Norge	200.00	1.00	0.43	0.03	0.05
Oslo	200.00	1.00	0.43	0.03	0.07
Bergen	200.00	1.00	0.43	0.03	0.07
Trondheim	200.00	1.00	0.43	0.03	0.05
Norge	50.00	2.00	0.64	0.08	0.04
Oslo	50.00	2.00	0.64	0.08	0.07
Bergen	50.00	2.00	0.64	0.08	0.05
Trondheim	50.00	2.00	0.64	0.08	0.04

Table 1: Detection probabilities of a handful of survey regimes, with detection probabilities from an empiric survey.