Selection of locations for insect monitoring in oaks

Jens Å

19 April, 2023

Contents


```
connect_to_insect_db()
```

Plan

From Rannveig's notes.

Vi skal trekke 150 eiker (100 til overvåkingen, men må ha ekstra for å justere i forhold til logistikk, grunneiertillatelse og i felt feks om noen trær er borte) fra de 600 ARKO-eikene (657 minus «gone» og «not found» i 2019), etter følgende kriterier:

Doblet sannsynlighet for å trekke trær med omkrets over 200 cm.

Andel eiker i vårt utvalg speiler fordelingen blant alle ARKO-eiker; hvis det feks er 20% i Vestland, 30% i Agder, 30% i V/T, 20% i Viken, så skal overvåkingseikene fordeles etter samme andeler.

Maks 2 overvåkingstrær per ARKO-rute.

Da hule eiker i utgangspunktet er relativt jevnt fordelt mellom skog og åpent landskap, regner vi med at dette vil reflekteres i utvalget av overvåkingstrær uten å legge inn noe styrende kriterium for dette. Vi sjekker om vi har tilfredsstillende fordeling av eiker i skog og utenfor skog totalt og i hver region etter å ha trukket et sett overvåkingstrær.

Etter å ha gjort et utvalg sjekker vi fordeling i forhold til ulike parametre (feks hulrom, vedmuld, barktype, treform), særlig hvor mange A-eiker som er representert.

Jens interpretation

I see two approaches, and will test both: 1. We first select a fixed set of SSB-squares. These will be randomly drawn, and thereby they will be proportionally drawn from the regions depending on how many squares each region

contain. Then we select max 2 trees within each location. We do this so that we get double the amount of large trees to small trees. This way, we can set the amount of localities, and if we go for 2 trees in all locations, we get 50 locations. 2. We select trees randomly, with max 2 trees within each SSB-square. We will then get an unknown number of localities and an unknown number of localities with just a single tree.

Load the source excel-file

```
loc_raw <- openxlsx::read.xlsx("../rawData/Oak_2017data_2019resurveydata.xlsx") %>%
  as_tibble()

loc_raw
```

```
## # A tibble: 657 x 47
##    row_number RuteID RuteJA TreID Antall Verdi Omkrets Synlig_~1 Hulhe~2 Hulhe~3
##         <dbl>  <dbl> <chr>  <chr>  <dbl> <chr> <chr>   <chr>     <chr>   <chr>
##  1          1      7 Nei    7_01       1 A     332     JA        1615    2
##  2          2     23 Nei    23_1       1 C     225     NEI       <NA>    <NA>
##  3          3     23 Nei    23_2       1 C     205     NEI       <NA>    <NA>
##  4          4     24 Nei    24_10      1 B     300     JA        <NA>    2
##  5          5     24 Nei    24_2       1 B     237     JA        100     2
##  6          6     24 Nei    24_3       1 C     230     NEI       <NA>    <NA>
##  7          7     24 Nei    24_4       1 C     222     NEI       <NA>    <NA>
##  8          8     24 Nei    24_5       1 C     216     NEI       <NA>    <NA>
##  9          9     24 Nei    24_6       1 B     245     JA        2750    2
## 10         10     24 Nei    24_7       1 B     320     NEI       <NA>    <NA>
## # ... with 647 more rows, 37 more variables: Vedmuld <chr>, Treform <chr>,
## #   Barktype <chr>, Mosedekning <chr>, Vitalitet <dbl>, Kulturspor <chr>,
## #   Omgivelser <chr>, Renskog <chr>, Mestskog <chr>, Noeskog <chr>,
## #   PlasseringAR5 <chr>, Forskrift_gammel <dbl>, Forskrift <dbl>, Vern <chr>,
## #   Gjenvoksing <dbl>, Gjenvoksing2 <dbl>, Skjøtselsbehov <chr>,
## #   UTM32_X_koordinat <dbl>, UTM32_Y_koordinat <dbl>, Kommune <dbl>,
## #   Områdenavn <chr>, Nøyaktighetsklasse <dbl>, Utvalgt.Natur.type <chr>, ...
```

Filter out trees that is gone or not found in 2019.

```
loc <- loc_raw %>%
  filter(
    Gone != 1,
    Not_found != 1
  )

nrow(loc)
```

```
## [1] 600
```

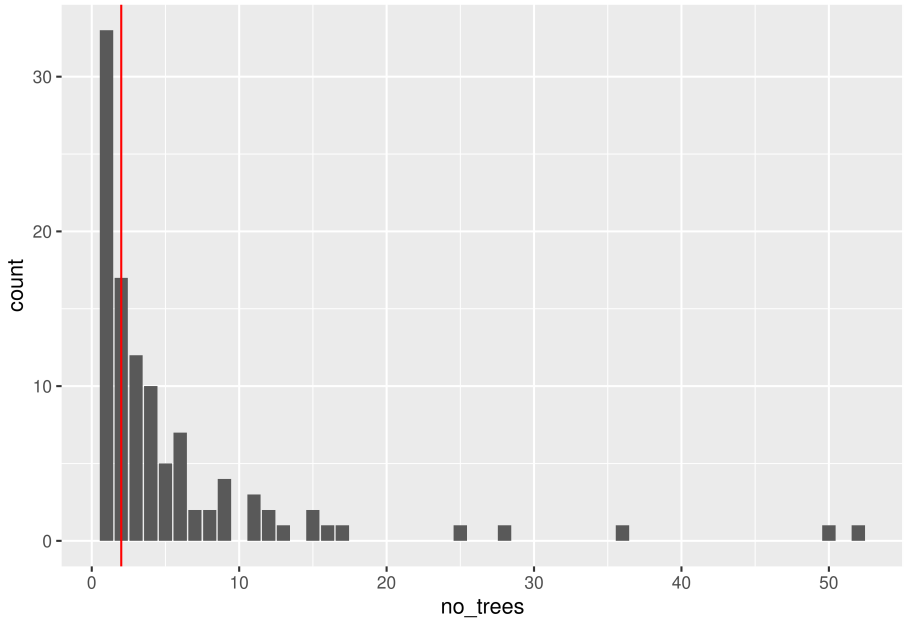
Selecting a fixed set of localities

How many locations have more than 2 trees anyway?

```
loc %>%
  group_by(RuteID) %>%
  summarise(no_trees = n()) %>%
  filter(no_trees > 1) %>%
  summarise(n_distinct(RuteID))
```

```
## # A tibble: 1 x 1
##   `n_distinct(RuteID)`
##                  <int>
## 1                   74
```

```
loc %>%
  group_by(RuteID) %>%
  summarise(no_trees = n()) %>%
  ggplot() +
  geom_bar(aes(x = no_trees)) +
  geom_vline(aes(xintercept = 2),
    col = "red"
  )
```




```
no_rute <- loc %>%
  summarise(no_rute = n_distinct(RuteID)) %>%
  pull()
```

```
squares <- loc %>%
  select(RuteID) %>%
  distinct() %>%
  pull()
```

We have 107 distinct survey squares (SSB) to choose from.

```
no_squares <- 80 # 50 + 30 extra to discard if needed

loc_square_subset <- loc %>%
  filter(RuteID %in% sample(squares,
    size = no_squares,
    replace = F
  ))
```

Apropos "double probability to draw a tree > 200 cm in circumference". This can be interpreted in several ways. There are more trees above the treshold, so a random sample will produce a higher probability anyway. Instead, I interpret it as drawing double the amount of large trees (>200cm) than smalller.

```
loc %>%
  group_by(Omkrets > 200) %>%
  summarise(no = n())
```

```
## # A tibble: 3 x 2
##   `Omkrets > 200`    no
##   <lgl>           <int>
## 1 FALSE             269
## 2 TRUE              329
## 3 NA                  2
```

Draw candidate trees

```
tree_selection <- loc %>%
  filter(!is.na(Omkrets > 200)) %>%
  mutate(sel_prob = ifelse(Omkrets > 200, 2 / 3, 1 / 3)) %>%
  group_by(RuteID) %>%
  filter(n() > 1) %>% # NB, only locations with >2 trees
  slice(sample(1:n(), 2, prob = sel_prob)) %>%
  mutate(large = Omkrets > 200)
```

```
tree_selection %>%
  ungroup() %>%
  summarise(no_ruter = n_distinct(RuteID))
```

```
## # A tibble: 1 x 1
##   no_ruter
##      <int>
```

```
## 1       74
```

```
tree_selection %>%
  group_by(Omkrets > 200) %>%
  summarise(no = n())
```

```
## # A tibble: 2 x 2
##   `Omkrets > 200`    no
##   <lgl>           <int>
## 1 FALSE              47
## 2 TRUE              101
```

```
tree_selection %>%
  ungroup() %>%
  summarise(no_rute = n_distinct(RuteID))
```

```
## # A tibble: 1 x 1
##   no_rute
##     <int>
## 1      74
```

Make an SF object

```
tree_sel_sf <- tree_selection %>%
  st_as_sf(
    coords = c(
      "UTM32_X_koordinat",
      "UTM32_Y_koordinat"
    ),
    crs = 25832
  )
```

Get the square geometries

```
ssb_500m <- read_sf(
  con,
  Id(
    schema = "backgrounds",
    table = "ssb_500m"
  )
) %>%
  st_transform(crs = 25832)

sel_ssb_500m <- ssb_500m %>%
  st_join(tree_sel_sf,
    left = FALSE
  ) %>%
  select(ssbid) %>%
  distinct()
```

```
regions <- read_sf(
  con,
  Id(
    schema = "backgrounds",
    table = "norway_terrestrial"
  )
)

south <- regions %>%
  filter(!(navn %in% c("Trøndelag", "Nordland", "Troms og Finnmark")))
```

```
# tmap_mode("view")
tm_shape(south) +
  tm_borders() +
  tm_shape(sel_ssb_500m) +
  tm_borders() +
  tm_shape(tree_sel_sf) +
  tm_dots(
    col = "large",
    size = 0.5,
    palette = ninaPalette()
  )
```

```
## Warning: palette colors names missing for FALSE, TRUE. Therefore, palette color
## names will be ignored
```