Naïve estimates of occupancy and abundance

Point count data analysis workshop 2025

Péter Sólymos

2025-11-12

Table of contents

Preamble

```
suppressPackageStartupMessages({
    library(dplyr)
    library(ggplot2)
    library(mefa4)
    library(detect)
})
```

Nuisance variables

Nuisance variables are covariates whose effects we want to control but we are not really interested in their effect. We just need to estimate them so that we can account for them as best as we can.

We have 3 such covariates in our example:

- `TSSR`: time since local sunrise, (survey time - sunrise time) / 24
- `DAY`: day of the year, ordinal day / 365
- `WindStart`: wind speed using Beaufort scale

```
x <- detect::josm$surveys |>
    select(
        Longitude,
        Latitude,
        WindStart,
        TSSR,
        DAY,
        Open,
        Water,
        Decid,
        OpenWet,
        Conif,
        ConifWet,
        Agr,
        UrbInd,
        SoftLin,
        Roads
    )
x[is.na(x$WindStart), "WindStart"] <- c(0, 3)

spp <- "OVEN" # change here if you want to use another species

y <- mefa4::Xtab(~ SiteID + SpeciesID, detect::josm$counts)[, spp, drop = FALSE]
x$Count <- y[rownames(x), ]
```

```
m1 <- glm(Count ~ Decid * ConifWet + TSSR + I(TSSR^2) + DAY + WindStart, data = x, family = p
summary(m1)
```

```
Call:
```

```
glm(formula = Count ~ Decid * ConifWet + TSSR + I(TSSR^2) + DAY +
    WindStart, family = poisson, data = x)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.56160    0.23637   2.376 0.017504 *
Decid           1.20208    0.07834  15.345  < 2e-16 ***
ConifWet       -2.31754    0.14921 -15.532  < 2e-16 ***
TSSR           -1.67851    0.87429  -1.920 0.054876 .
I(TSSR^2)       4.33185    4.12004   1.051 0.293070
DAY            -2.15802    0.50845  -4.244 2.19e-05 ***
WindStart      -0.05751    0.01629  -3.530 0.000416 ***
Decid:ConifWet  5.44652    0.35783  15.221  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 7447.9  on 4568  degrees of freedom
Residual deviance: 5368.0  on 4561  degrees of freedom
AIC: 10537

Number of Fisher Scoring iterations: 6
```

```r
m2 <- step(m1, trace = 0)
summary(m2)
```

```
Call:
glm(formula = Count ~ Decid + ConifWet + TSSR + DAY + WindStart +
    Decid:ConifWet, family = poisson, data = x)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.54389    0.23581   2.306  0.02108 *
Decid           1.19988    0.07828  15.328  < 2e-16 ***
ConifWet       -2.31977    0.14915 -15.553  < 2e-16 ***
TSSR           -0.79903    0.25558  -3.126  0.00177 **
DAY            -2.17646    0.50817  -4.283 1.84e-05 ***
WindStart      -0.05650    0.01626  -3.475  0.00051 ***
Decid:ConifWet  5.44127    0.35780  15.208  < 2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 7447.9  on 4568  degrees of freedom
Residual deviance: 5369.1  on 4562  degrees of freedom
AIC: 10536

Number of Fisher Scoring iterations: 6
```

```
MuMIn::model.sel(m1, m2)
```

```
Model selection table
    (Int)    CnW    DAY   Dcd    TSS TSS^2      WnS CnW:Dcd df    logLik
m2 0.5439 -2.320 -2.176 1.200 -0.799       -0.05650   5.441  7 -5261.024
m1 0.5616 -2.318 -2.158 1.202 -1.679 4.332 -0.05751   5.447  8 -5260.474
      AICc delta weight
m2 10536.1  0.00  0.611
m1 10537.0  0.91  0.389
Models ranked by AICc(x)
```

When predicting abundance and accounting for nuisance variables, we want to set the values that maximize the prediction. All 3 nuisance variables have a negative effect, so we need to pick the mim:

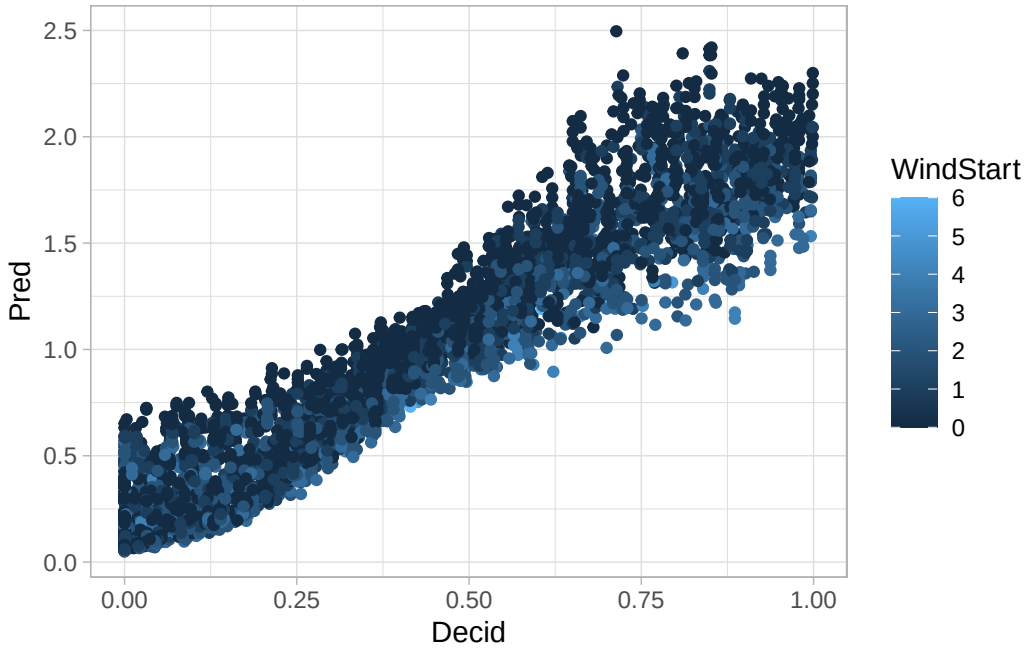
```
xnew <- x
xnew$TSSR <- min(xnew$TSSR)
xnew$DAY <- min(xnew$DAY)
xnew$WindStart <- min(xnew$WindStart)

x$Pred <- predict(m2, x, type = "response")
x$PredNew <- predict(m2, xnew, type = "response")
mean(x$Pred)
```

```
[1] 0.8859707
```

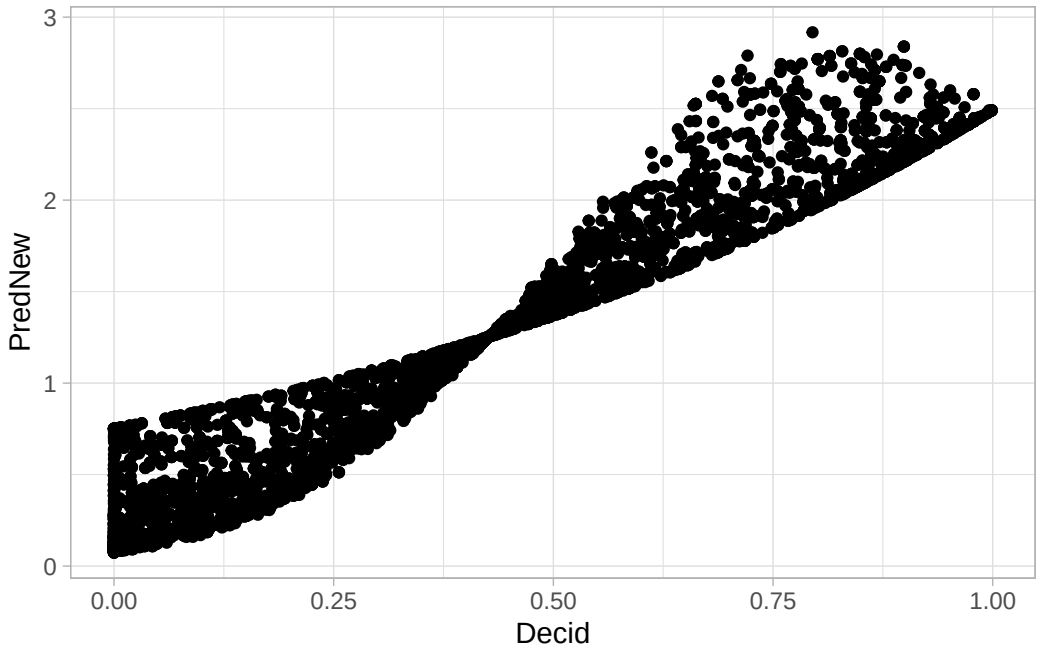
```
mean(x$PredNew)
```

```
[1] 1.159023
```

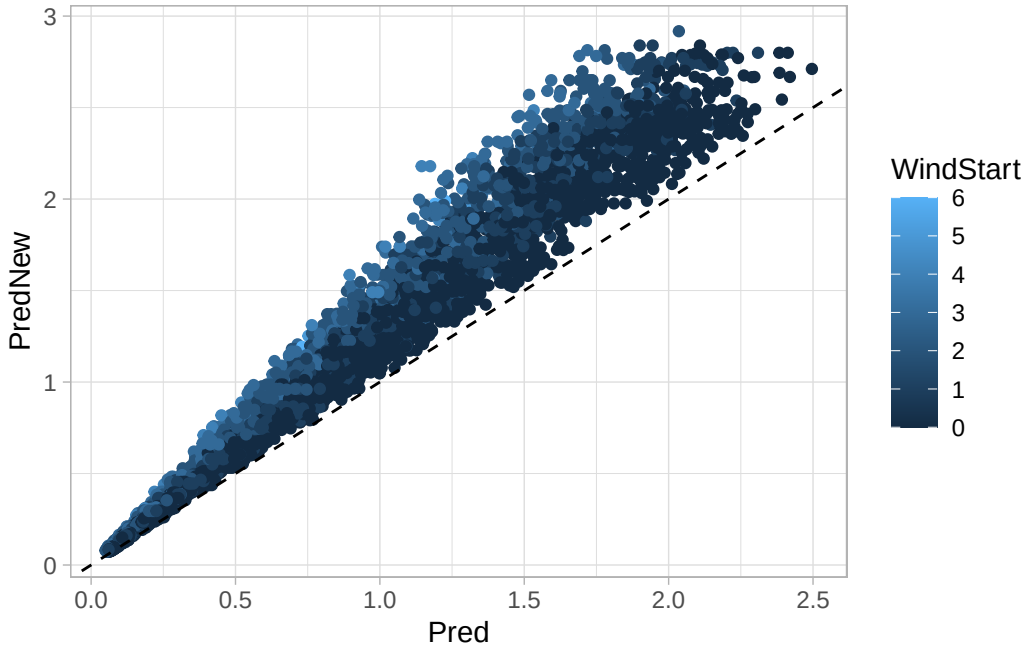
```
x |> ggplot(aes(x = Decid, y = Pred, col = WindStart)) +
    geom_point() +
    theme_light()
```




```
x |> ggplot(aes(x = Decid, y = PredNew)) +
    geom_point() +
    theme_light()
```




```
x |> ggplot(aes(x = Pred, y = PredNew, col = WindStart)) +
    geom_point() +
    geom_abline(intercept = 0, slope = 1, lty = 2) +
    theme_light()
```



Offsets

Offsets are constant terms in the linear predictor, e.g. $log(\lambda_i) = \beta_0 + \beta_1 x_{1i} + o_i$, where o_i is an offset.

In the survey area case, an offset might be the log of area surveyed. Abundance (N) is population density (D) multiplied by survey area (A). The logic for this is based on point processes: intensity is a linear function of area under a homogeneous Poisson point process. So we can say that $E[Y_i] = N_i = D_i A_i$, thus $log(N_i) = log(D_i) + log(A_i)$ where $o_i = log(A_i)$ is the offset.

The problem is that we do not know the survey area. We used 10 minutes and unlimited distance counts.

Let's see if using area as offset makes our models comparable. Instead of mixing up different survey types, let's see if we can make them identical. We use distance in meters divided by 100, so the population density is estimated in ha.

Distance effects

We will inspect what happens if we used different survey radii. Our data set has 50 and 100 m distances. We will not use the unlimited (>100 m) radius because the area surveyed would be arbitrary. On Day 2 we will see how to estimate the effective area sampled instead.

First, the `y1` matrix will have columns for the counts in the 3 different distance bands that we used to add 2 new `Count` column:

```
y1 <- mefa4::Xtab(~ SiteID + Dis, detect::josm$counts, subset = detect::josm$counts$SpeciesI
head(y1)
```

```
6 x 3 sparse Matrix of class "dgCMatrix"
        0-50m 50-100m 100+m
CL10102     1       2     .
CL10106     .       .     .
CL10108     .       .     .
CL10109     1       2     .
CL10111     1       .     1
CL10112     .       2     .
```

```
x$Count_10min_50m <- y1[rownames(x), "0-50m"]
x$Count_10min_100m <- y1[rownames(x), "0-50m"] + y1[rownames(x), "50-100m"]
```

Not surprisingly, the mean count in the 0–50 m area and much smaller than the counts in the 0–100 m area:

```
mean(y1[, "0-50m"])
```

```
[1] 0.2952506
```

```
mean(y1[, "0-50m"] + y1[, "50-100m"])
```

```
[1] 0.7874808
```

The mean density (abundance per unit area), however, shows the reverse pattern. Here we used units of 100 m, e.g. 50 m becomes 0.5, and 100 m becomes 1. This way, the density is calculated over a 100 x 100 m^2 area, that is 1 ha, commonly used for expressing density in the bird literature:

```
mean(y1[, "0-50m"]) / (0.5^2 * pi)
```

```
[1] 0.3759247
```

```
mean(y1[, "0-50m"] + y1[, "50-100m"]) / (1^2 * pi)
```

```
[1] 0.2506629
```

Why would the density over the 100 m area be smaller than the density over the 50 m area? Take your guesses. We'll learn about this in the next 2 days.

We add the survey area is calculated as $A = r^2\pi$ and then take the log of if because we are using the log link function. We then update the formula to use `Count_10min_50m` on the left and `offset(logA_50m)` on the right:

```
x$logA_50m <- log(0.5^2 * pi)
m3 <- update(m2, Count_10min_50m ~ . + offset(logA_50m))
summary(m3)
```

```
Call:
glm(formula = Count_10min_50m ~ Decid + ConifWet + TSSR + DAY +
    WindStart + Decid:ConifWet + offset(logA_50m), family = poisson,
    data = x)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.39999    0.40872  -0.979   0.3278
Decid           0.91916    0.13258   6.933 4.13e-12 ***
ConifWet       -3.14537    0.28576 -11.007  < 2e-16 ***
TSSR           -0.24180    0.44221  -0.547   0.5845
DAY            -1.54572    0.88131  -1.754   0.0794 .
WindStart      -0.08638    0.02857  -3.023   0.0025 **
Decid:ConifWet  6.38313    0.67259   9.490  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 4162.0  on 4568  degrees of freedom
Residual deviance: 3421.7  on 4562  degrees of freedom
AIC: 5711

Number of Fisher Scoring iterations: 6
```

We follow the same steps for the 100 m radius model:

```
x$logA_100m <- log(1^2 * pi)
m4 <- update(m2, Count_10min_100m ~ . + offset(logA_100m))
summary(m4)
```

```
Call:
glm(formula = Count_10min_100m ~ Decid + ConifWet + TSSR + DAY +
    WindStart + Decid:ConifWet + offset(logA_100m), family = poisson,
    data = x)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.69146    0.25019  -2.764 0.005715 **
Decid           1.21568    0.08324  14.605  < 2e-16 ***
```

```
ConifWet       -2.55033    0.16601 -15.363  < 2e-16 ***
TSSR           -0.70711    0.27107  -2.609 0.009091 **
DAY            -2.23644    0.53917  -4.148 3.35e-05 ***
WindStart      -0.06520    0.01732  -3.766 0.000166 ***
Decid:ConifWet  5.80413    0.39144  14.828  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 7133.5  on 4568  degrees of freedom
Residual deviance: 5152.3  on 4562  degrees of freedom
AIC: 9871.3

Number of Fisher Scoring iterations: 6
```

As for predicting with an offset, we will create a copy of `xnew` and add the 2 offset terms. Notice that we don't use the same values as we used for modeling, because we want predictions over a unit area of 1 ha, which is $log(1) = 0$.

```
xnew2 <- xnew

xnew2$logA_50m <- 0
xnew2$logA_100m <- 0

xnew2$Pred_10min_50m <- predict(m3, xnew2, type = "response")
xnew2$Pred_10min_100m <- predict(m4, xnew2, type = "response")
```

Density per unit area is higher for the 50 m model, similarly to what we have seen before.

```
mean(xnew2$Pred_10min_50m)
```

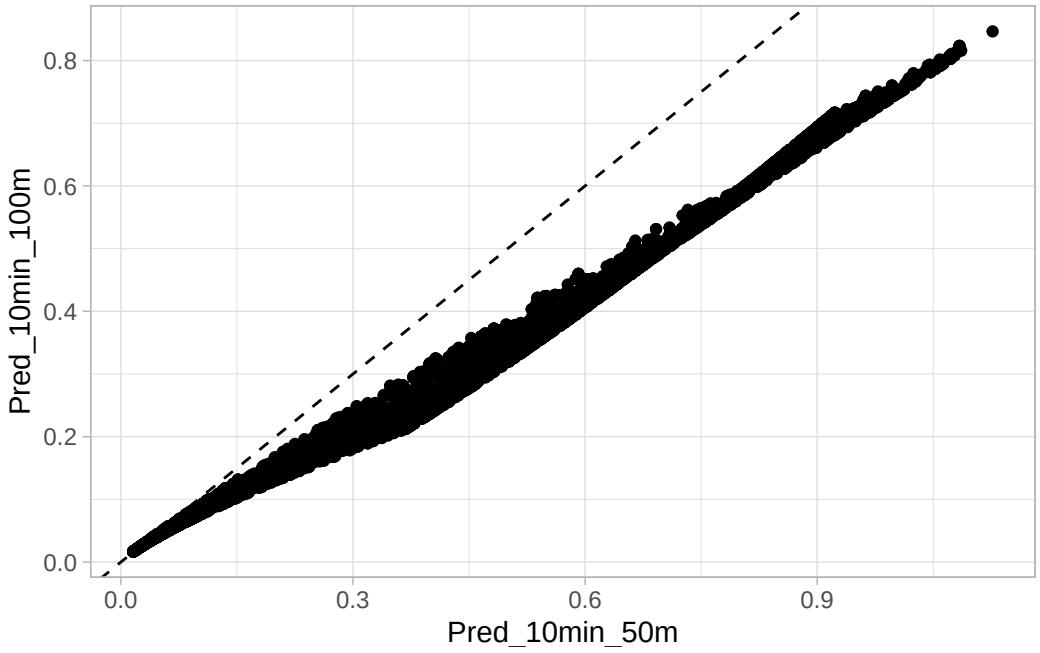
```
[1] 0.4520192
```

```
mean(xnew2$Pred_10min_100m)
```

```
[1] 0.327258
```

Let's compare the predictions:

```
xnew2 |> ggplot(aes(x = Pred_10min_50m, y = Pred_10min_100m)) +
    geom_point() +
    geom_abline(intercept = 0, slope = 1, lty = 2) +
    theme_light()
```



Duration effect

Next we will inspect what effect the count duration has on our estimates. Follow the same approach to get the counts by sites and distance intervals after subsetting for our species and for the 0–3 minutes time interval:

```
y2 <- mefa4::Xtab(~ SiteID + Dis, detect::josm$counts,
    subset = detect::josm$counts$SpeciesID == spp &
        detect::josm$counts$Dur == "0-3min"
)
x$Count_3min_50m <- y2[rownames(x), "0-50m"]
x$Count_3min_100m <- y2[rownames(x), "0-50m"] + y2[rownames(x), "50-100m"]
```

Same patterns for mean count and the empirical density as what we saw based on the 10 minutes counts:

```r
mean(y2[, "0-50m"])
```

```
[1] 0.2433793
```

```r
mean(y2[, "0-50m"] + y2[, "50-100m"])
```

```
[1] 0.6163274
```

```r
mean(y2[, "0-50m"]) / (0.5^2 * pi)
```

```
[1] 0.3098801
```

```r
mean(y2[, "0-50m"] + y2[, "50-100m"]) / (1^2 * pi)
```

```
[1] 0.1961831
```

The models:

```r
m5 <- update(m2, Count_3min_50m ~ . + offset(logA_50m))
summary(m5)
```

```
Call:
glm(formula = Count_3min_50m ~ Decid + ConifWet + TSSR + DAY +
    WindStart + Decid:ConifWet + offset(logA_50m), family = poisson,
    data = x)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.20050    0.44892  -0.447  0.65514
Decid           0.81383    0.14491   5.616 1.96e-08 ***
ConifWet       -3.19250    0.31129 -10.256  < 2e-16 ***
TSSR           -0.38761    0.48701  -0.796  0.42609
DAY            -2.21342    0.97017  -2.281  0.02252 *
WindStart      -0.10426    0.03182  -3.276  0.00105 **
Decid:ConifWet  6.39784    0.73980   8.648  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3710.3  on 4568  degrees of freedom
Residual deviance: 3114.2  on 4562  degrees of freedom
AIC: 5074.4

Number of Fisher Scoring iterations: 6
```

```r
m6 <- update(m2, Count_3min_100m ~ . + offset(logA_100m))
summary(m6)
```

```
Call:
glm(formula = Count_3min_100m ~ Decid + ConifWet + TSSR + DAY +
    WindStart + Decid:ConifWet + offset(logA_100m), family = poisson,
    data = x)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.74189    0.28256  -2.626 0.008649 **
Decid           1.21830    0.09436  12.912  < 2e-16 ***
ConifWet       -2.53029    0.18732 -13.507  < 2e-16 ***
TSSR           -0.79279    0.30647  -2.587 0.009685 **
DAY            -2.66570    0.60942  -4.374 1.22e-05 ***
WindStart      -0.07150    0.01965  -3.639 0.000273 ***
Decid:ConifWet  5.89165    0.44143  13.347  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 6276.2  on 4568  degrees of freedom
Residual deviance: 4720.1  on 4562  degrees of freedom
AIC: 8706.1

Number of Fisher Scoring iterations: 6
```

Predictions using `xnew2` as before:

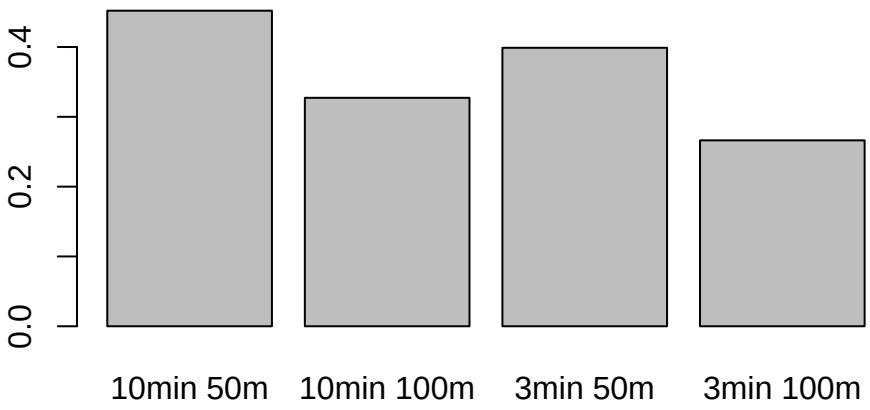
```
xnew2$Pred_3min_50m <- predict(m5, xnew2, type = "response")
xnew2$Pred_3min_100m <- predict(m6, xnew2, type = "response")
```

The 10 minutes estimates are higher due to accumulating more detections over the longer duration; the 50 m radius counts led to higher density estimates due to higher detectability compared to the 100 m radius counts:

```
D_est <- c(
    "10min 50m" = mean(xnew2$Pred_10min_50m),
    "10min 100m" = mean(xnew2$Pred_10min_100m),
    "3min 50m" = mean(xnew2$Pred_3min_50m),
    "3min 100m" = mean(xnew2$Pred_3min_100m)
)
D_est
```

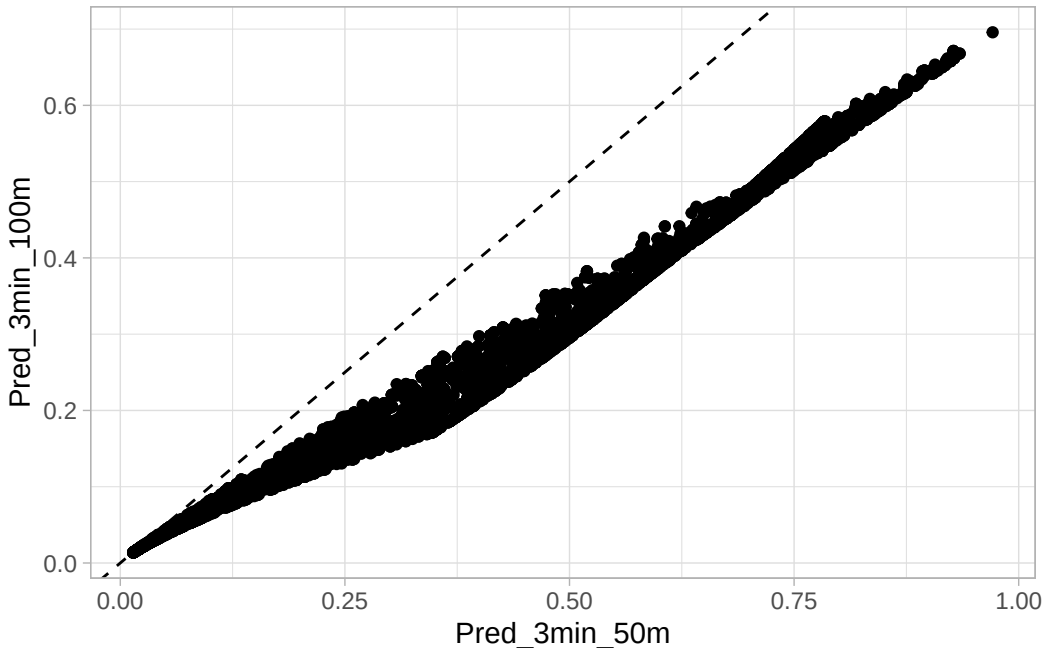
```
 10min 50m 10min 100m   3min 50m  3min 100m
 0.4520192  0.3272580  0.3988992  0.2663132
```

```
barplot(D_est)
```

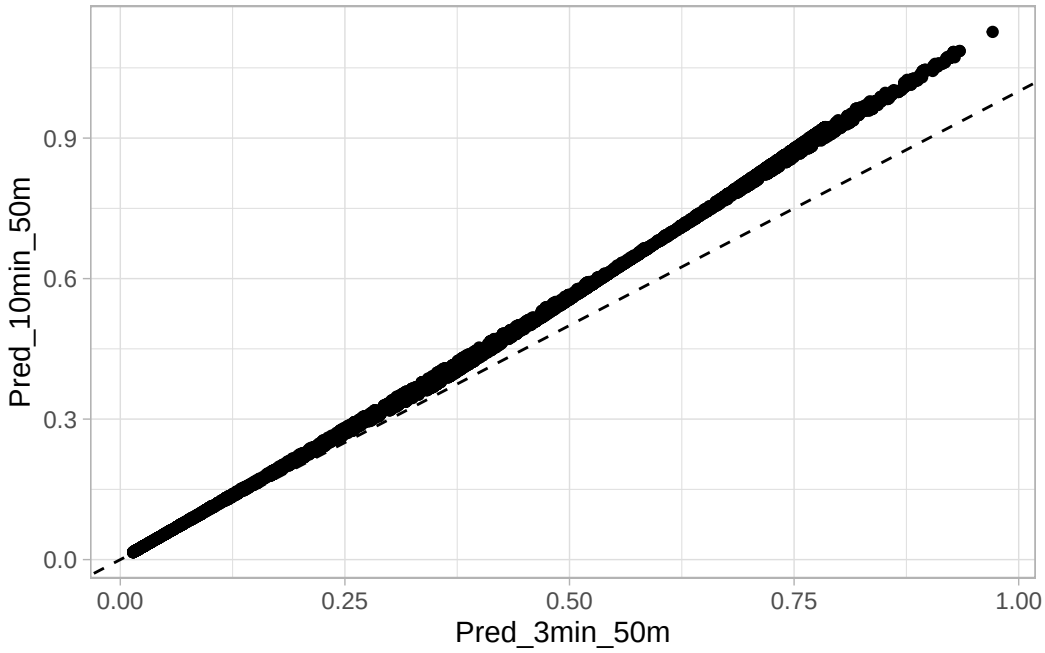


Make some plots to compare the site level predictions:

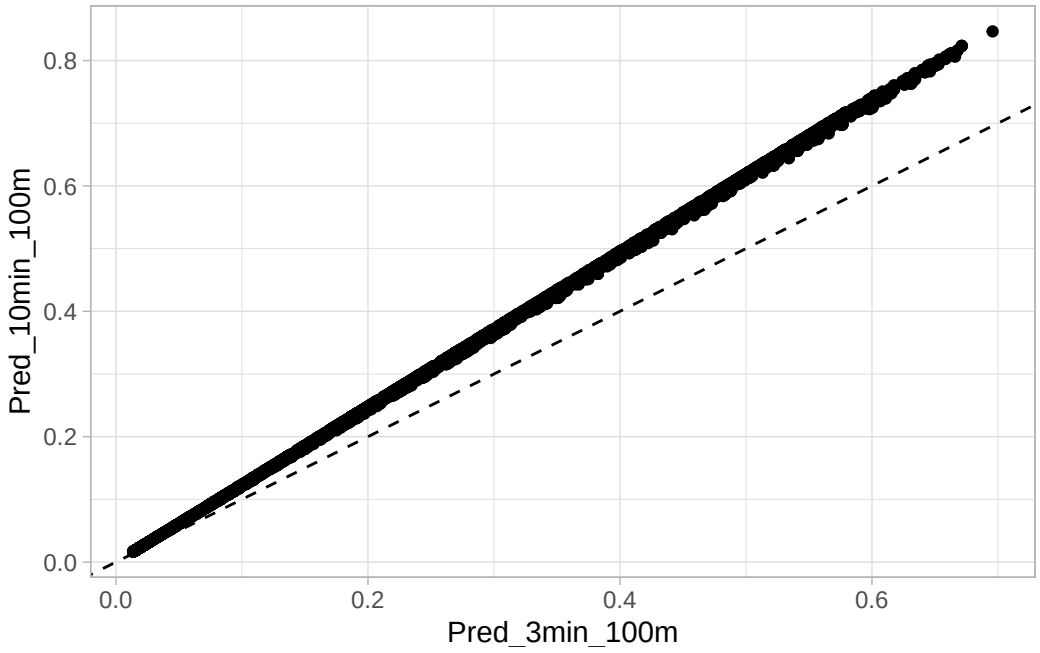
```
xnew2 |> ggplot(aes(x = Pred_3min_50m, y = Pred_3min_100m)) +
    geom_point() +
    geom_abline(intercept = 0, slope = 1, lty = 2) +
    theme_light()
```




```
xnew2 |> ggplot(aes(x = Pred_3min_50m, y = Pred_10min_50m)) +
    geom_point() +
    geom_abline(intercept = 0, slope = 1, lty = 2) +
    theme_light()
```



```
xnew2 |> ggplot(aes(x = Pred_3min_100m, y = Pred_10min_100m)) +
    geom_point() +
    geom_abline(intercept = 0, slope = 1, lty = 2) +
    theme_light()
```



Accounting for different field protocols

When we survey some areas, the survey protocol is the same for all sites. However, when we combine different datasets, we might find that each might have a slightly different sampling design.

Some surveys might be 10 min (3–5–100 min) and unlimited radius (0–100–∞ m), other might be 0–3 min and unlimited (0–∞ m) without any time and distance bands.

We will assign one of 4 survey protocols to each site and based on that:

- the `Count` values corresponding to the time and distance intervals
- set the log area for offset matching the protocol radii

```
Methods <- c("3min_50m", "3min_100m", "10min_50m", "10min_100m")
x$Method <- factor(sample(
    Methods,
    nrow(x),
```

```
    replace = TRUE
), levels = Methods)
x$Count_Rnd <- 0
for (i in Methods) {
    x$Count_Rnd[x$Method == i] <- x[[paste0("Count_", i)]][x$Method == i]
}
x$logA_Rnd <- log(ifelse(x$Method %in% c("3min_50m", "10min_50m"), 0.5, 1)^2 * pi)
```

This next model will have `Count_Rnd` as the response, and we add the `Method` variable and the offset `offset(logA_Rnd)`:

```
m7 <- update(m2, Count_Rnd ~ . + Method + offset(logA_Rnd))
summary(m7)
```

```
Call:
glm(formula = Count_Rnd ~ Decid + ConifWet + TSSR + DAY + WindStart +
    Method + Decid:ConifWet + offset(logA_Rnd), family = poisson,
    data = x)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.27017    0.32292  -0.837  0.40279
Decid             1.13540    0.10753  10.559  < 2e-16 ***
ConifWet         -2.87012    0.22113 -12.979  < 2e-16 ***
TSSR             -0.78176    0.34289  -2.280  0.02261 *
DAY              -2.77702    0.68510  -4.053 5.05e-05 ***
WindStart        -0.06451    0.02211  -2.918  0.00352 **
Method3min_100m  -0.31468    0.07291  -4.316 1.59e-05 ***
Method10min_50m   0.26369    0.08379   3.147  0.00165 **
Method10min_100m -0.11500    0.07132  -1.612  0.10686
Decid:ConifWet    6.41037    0.51232  12.512  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5371.0  on 4568  degrees of freedom
Residual deviance: 4004.8  on 4559  degrees of freedom
AIC: 7241.2

Number of Fisher Scoring iterations: 6
```

When predicting, we pick the method that leads to highest counts (10 min) and smalles detection error (50 m radius), define the log area offset as 0:

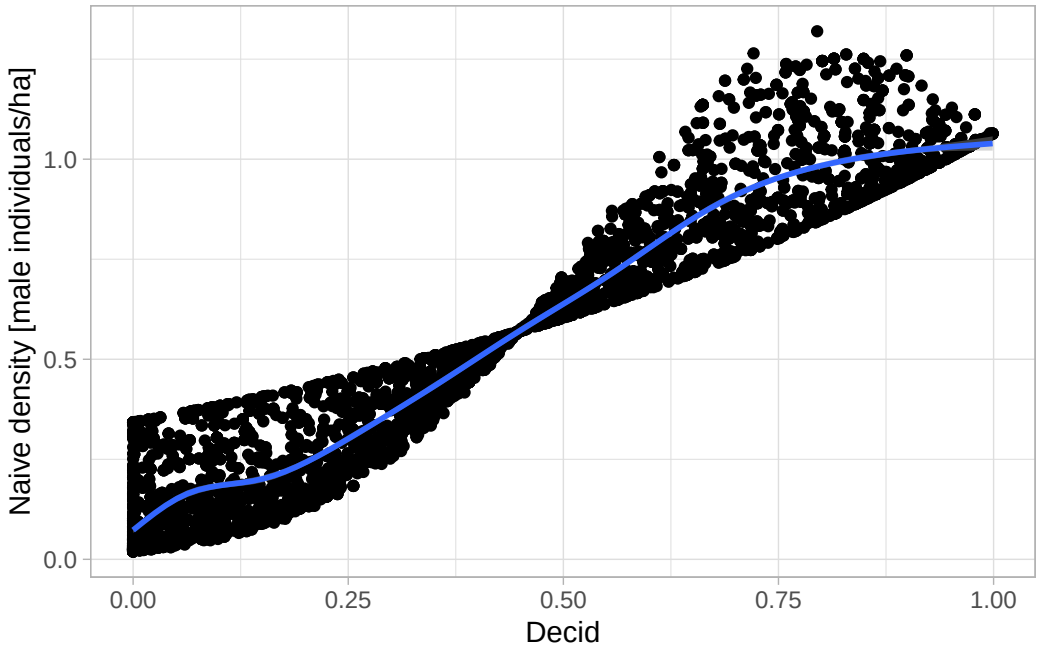
```
xnew2$logA_Rnd <- 0
xnew2$Method <- "10min_50m" # highest duration, highest detection
xnew2$Pred_Rnd <- predict(m7, xnew2, type = "response")

mean(xnew2$Pred_Rnd)
```

```
[1] 0.4972659
```

The mean density is 0.46 male birds per ha, but it can reach a maximum of 1.1 males per ha for fully deciduous habitats:

```
xnew2 |> ggplot(aes(x = Decid, y = Pred_Rnd)) +
    geom_point() +
    geom_smooth() +
    theme_light() +
    ylab("Naive density [male individuals/ha]")
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Mapping results

The habitat covariates came from detailed (vector based) habitat mapping, these were than summarized by 1 km^2 pixels using the area proportions of these land cover types. The raster file in the `data` folder has the corresponding layers.

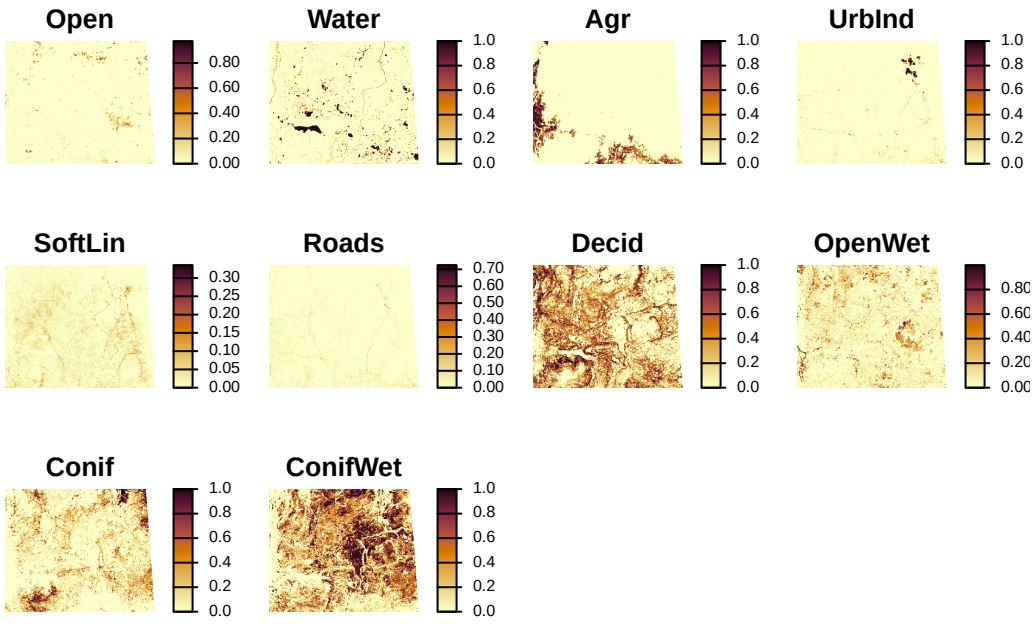
```
library(sf)
```

```
Linking to GEOS 3.13.0, GDAL 3.8.5, PROJ 9.5.1; sf_use_s2() is TRUE
```

```
library(terra)
```

```
terra 1.8.80


Attaching package: 'terra'

The following object is masked from 'package:knitr':

    spin
```

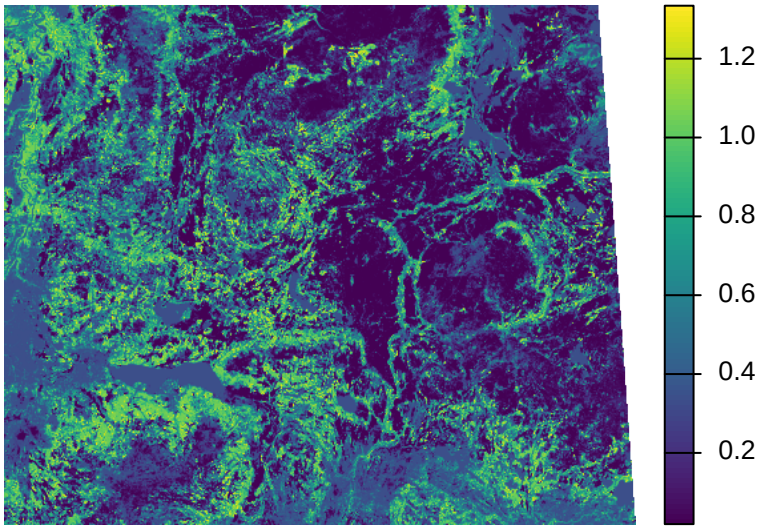
```
r <- rast("../data/josm-landcover.tif")

plot(r, col = hcl.colors(50, "Lajolla"), axes = FALSE)
```



By fixing the nuisance variables to a constant value and adding our offset term, we can make predictions.

Without detailing all the steps, here is how to make raster predictions using `m7`:

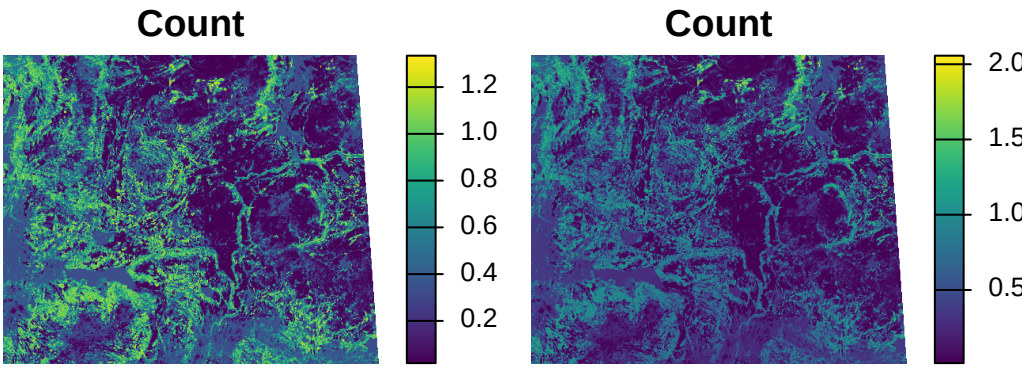
```
rd <- data.frame(
    as.matrix(r),
    WindStart = min(x$WindStart),
    TSSR = min(x$TSSR),
    DAY = min(x$DAY),
    Method = "10min_50m",
    logA_Rnd = 0
)

rp <- r[[1]]
names(rp) <- "Count"
values(rp) <- predict(m7, rd, type = "response")

plot(rp, axes = FALSE)
```



Next we add latitude and longitude (with interactions and quadratic terms) as new terms to better capture spatial patterns:

```
xy <- sf::st_as_sf(x, coords = c("Longitude", "Latitude"), crs = 4269) # NAD83 EPSG:4269
## NAD83 / Alberta 10-TM (Forest) EPSG:3400
xy <- sf::st_transform(xy, sf::st_crs(r))
coords <- sf::st_coordinates(xy)
```

```
x$X <- coords[, "X"]
x$Y <- coords[, "Y"]

m8 <- update(m7, . ~ . + X * Y + I(X^2) * I(Y^2))

rp2 <- rp
rxy <- crds(r, na.rm = FALSE)
colnames(rxy) <- c("X", "Y")
rd <- data.frame(rd, rxy)
values(rp2) <- predict(m8, rd, type = "response")

plot(rast(list(rp, rp2)), axes = FALSE)
```




```
mean(values(rp), na.rm = TRUE)
```

```
[1] 0.3999327
```

```
mean(values(rp2), na.rm = TRUE)
```

```
[1] 0.4171569
```

```
2 * sum(values(rp), na.rm = TRUE) * 100
```

```
[1] 14379340
```

```
2 * sum(values(rp2), na.rm = TRUE) * 100
```

```
[1] 14998627
```

```r
library(mapview)

ct <- xy[, "Count"]
ct$Count[ct$Count > 0] <- 1
ct$Count <- as.factor(ct$Count)

mapview(rp2) + mapview(ct)
```

Next

Multiple-visit occupancy and N-mixture models