Analyses co-occurrence grand dauphin et activités humaines

Olivier Gimenez

28/09/2020

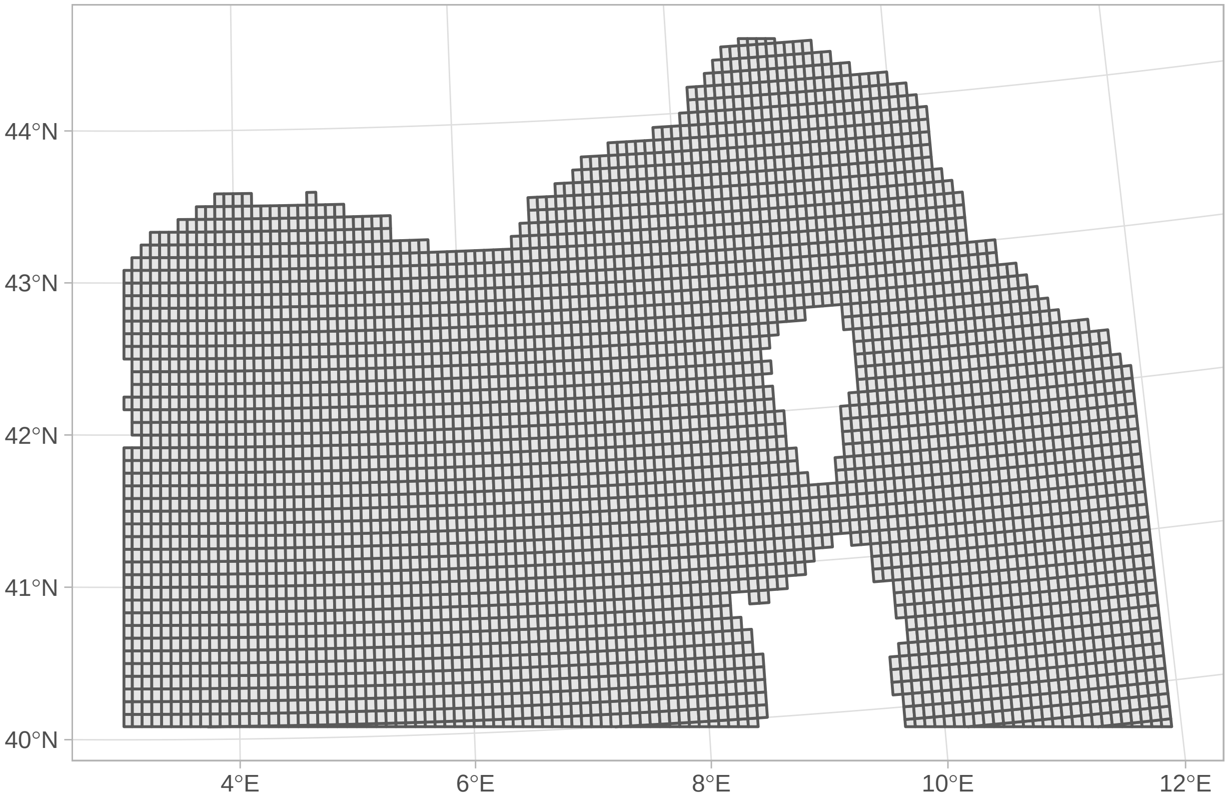
Lecture et nettoyage des données

La grille.

```
grid <- st_read("Grid/grid.shp")
```

```
## Reading layer ‘grid’ from data source ‘/Users/oliviergimenez/Dropbox/OG/GITHUB/human-tursiops-twospe
## Simple feature collection with 4356 features and 3 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: 701000 ymin: 5886622 xmax: 1467639 ymax: 6390000
## proj4string:    +proj=lcc +lat_1=44 +lat_2=49 +lat_0=46.5 +lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GI
```

```
grid %>%
  ggplot() +
  geom_sf()
```



Les dauphins.

```
load("20180914_SAMM_data_LauretValentin.RData")
```

Les données été et hiver.

```
dauphins_summer <- summer
dauphins_winter <- winter
```

Les données transect uniquement.

```
transect_summer <- dauphins_summer$segdata %>%
  as_tibble() %>%
  select(date = date,
         transect = Transect.Label,
         eastings = X,
         northings = Y,
         counts = n,
         effort = Effort,
         id = Sample.Label) %>%
  add_column(season = "summer")

transect_winter <- dauphins_winter$segdata %>%
  as_tibble() %>%
  select(date = date,
         transect = Transect.Label,
         eastings = X,
         northings = Y,
         counts = n,
         effort = Effort,
         id = Sample.Label) %>%
  add_column(season = "winter")

transect <- bind_rows(transect_summer, transect_winter)
```

Quelques statistiques, avec le nombre de détections par transect.

```
transect %>%
  count(transect, wt = counts, sort = TRUE)
```

```
## # A tibble: 1,780 x 2
##    transect     n
##    <chr>    <dbl>
##  1 522          5
##  2 4495         4
##  3 2846         3
##  4 3769         3
##  5 4278         3
##  6 5625         3
##  7 2025         2
##  8 2032         2
##  9 2059         2
## 10 2061         2
## # ... with 1,770 more rows
```

Le nombre total de dauphins.

```
transect %>%
  count(transect, wt = counts, sort = TRUE) %>%
  select(n) %>%
  sum()
```

```
## [1] 105
```

Et l'effort par transect.

```
transect %>%
  group_by(transect) %>%
  summarise(nb_detections = sum(counts),
            effort_total = mean(effort)) %>%
  arrange(desc(nb_detections))
```

```
## # A tibble: 1,780 x 3
##    transect nb_detections effort_total
##    <chr>            <dbl>        <dbl>
##  1 522                  5        12.6
##  2 4495                 4         7.21
##  3 2846                 3        10.3
##  4 3769                 3         9.78
##  5 4278                 3        10.6
##  6 5625                 3        10.2
##  7 2025                 2        13.6
##  8 2032                 2         9.37
##  9 2059                 2        10.4
## 10 2061                 2        10.0
## # ... with 1,770 more rows
```

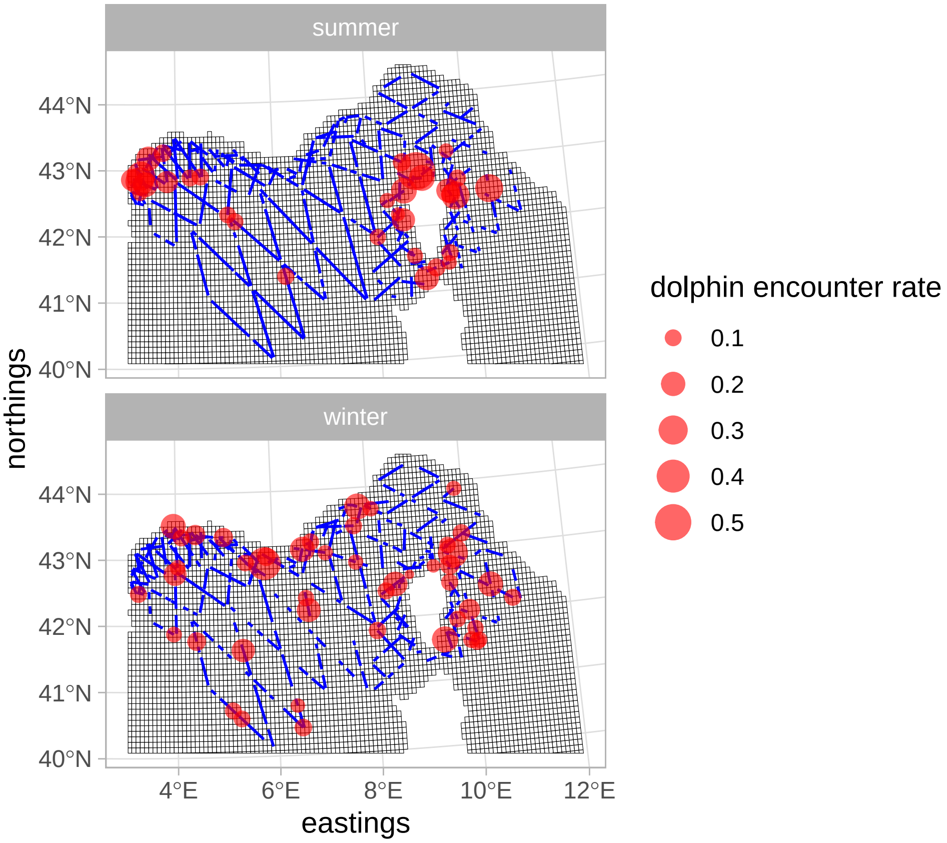
L'effort total.

```
transect %>%
  group_by(transect) %>%
  summarise(effort_total = max(effort)) %>%
  select(effort_total) %>%
  sum()
```

```
## [1] 15353.45
```

Visualisation.

```
grid %>%
  ggplot() +
  geom_sf(lwd = 0.1, color = "black", fill = "white")  +
  geom_line(data = transect, color = "blue",
            aes(x = eastings, y = northings, group = transect)) +
  coord_sf(xlim = st_bbox(grid)[c(1,3)],
           ylim = st_bbox(grid)[c(2,4)]) +
  geom_point(data = transect %>% filter(counts > 0),
             aes(x = eastings, y = northings, size = counts / effort),
             color = "red", alpha = 0.6) +
  labs(size = "dolphin encounter rate") +
  facet_wrap(~season, ncol = 1)
```



Les activités.

```
load("20200928_SAMM_data_Pressure.RData")
```

On récupère les activités par saison en les regroupant dans une catégorie unique *pêche*. Il y a le détail : "Bouee de peche", Bateau art dormant (fileyeur, caseyeur)",”Bateau chalutier",”Bateau de peche pro",”Bateau senneur, bolincheur".

```
activ_summer <- transect %>%
  filter(season == "summer") %>%
  mutate(id = as.numeric(id),
         dolphins = if_else(counts>0, 1, 0)) %>%
  select(date, id, eastings, northings, effort, dolphins, transect) %>%
  full_join(summer_fishingactivities$obsdata, by =  c("id" = "Sample.Label")) %>%
  select(date,
         eastings,
         northings,
         dolphins,
         what,
         effort,
```

```r
         id,
         transect) %>%
  mutate(peche = if_else(!is.na(what), 1, 0)) %>%
  add_column(season = "summer") %>%
  select(date, eastings, northings, dolphins, effort, peche, season, id, transect)

activ_winter <- transect %>%
  filter(season == "winter") %>%
  mutate(id = as.numeric(id),
         dolphins = if_else(counts>0, 1, 0)) %>%
  select(date, id, eastings, northings, effort, dolphins, transect) %>%
  full_join(winter_fishingactivities$obsdata, by =  c("id" = "Sample.Label")) %>%
  select(date,
         eastings,
         northings,
         dolphins,
         what,
         effort,
         id,
         transect) %>%
  mutate(peche = if_else(!is.na(what), 1, 0)) %>%
  add_column(season = "winter") %>%
  select(date, eastings, northings, dolphins, effort, peche, season, id, transect)

activ <- bind_rows(activ_summer, activ_winter)
```

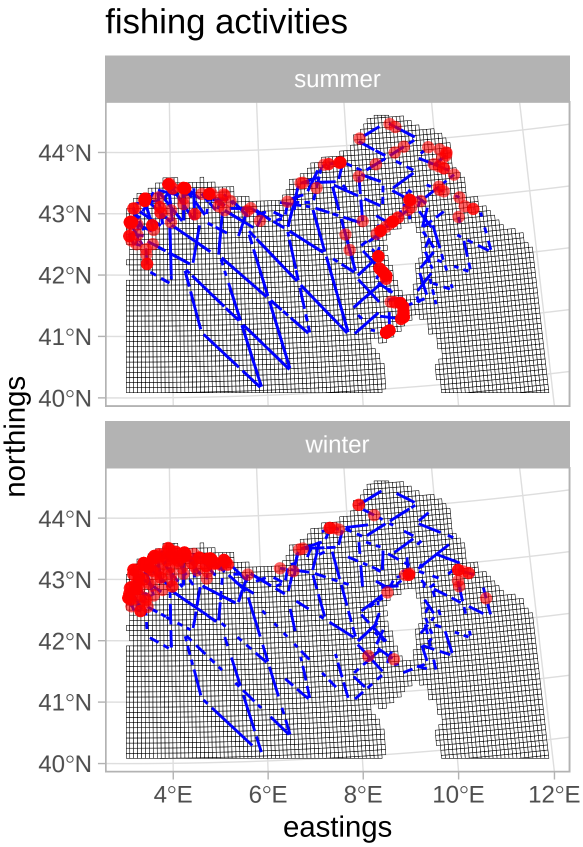
Quelques statistiques, avec le nombre d'activités par transect.

```r
activ %>%
  group_by(transect, season) %>%
  summarise(n_peche = sum(peche)) %>%
  filter(n_peche>0) %>%
  arrange(desc(n_peche))
```

```
## # A tibble: 201 x 3
## # Groups:   transect [201]
##    transect season n_peche
##    <chr>    <chr>    <dbl>
##  1 5828     summer      13
##  2 5568     summer      11
##  3 594      winter      10
##  4 1838     winter       9
##  5 5572     summer       9
##  6 1851     winter       7
##  7 5820     summer       7
##  8 1823     winter       6
##  9 1928     winter       6
## 10 2034     winter       6
## # ... with 191 more rows
```

Visualisation.

```
grid %>%
  ggplot() +
  geom_sf(lwd = 0.1, color = "black", fill = "white")  +
  geom_line(data = activ, color = "blue",
            aes(x = eastings, y = northings, group = transect)) +
  coord_sf(xlim = st_bbox(grid)[c(1,3)],
           ylim = st_bbox(grid)[c(2,4)]) +
  geom_point(data = activ %>% filter(peche > 0),
            aes(x = eastings, y = northings),
             color = "red", alpha = 0.6) +
  facet_wrap(~season, ncol = 1) +
  labs(title = "fishing activities")
```



Construction des chroniques de détection/non-détections des sites

Get transects and grid in same coordinates system.

```
transect_line <- activ %>%
  mutate(month = month(date)) %>%
  select(eastings, northings, date, dolphins, peche, season, id, transect, month) %>%
  as.data.frame() %>%
  st_as_sf(coords = c("eastings", "northings"),
           crs = "+proj=longlat +init=EPSG:3035")

st_crs(grid) <- "+proj=longlat +init=EPSG:3035"
```

Intersect the grid and the transects.

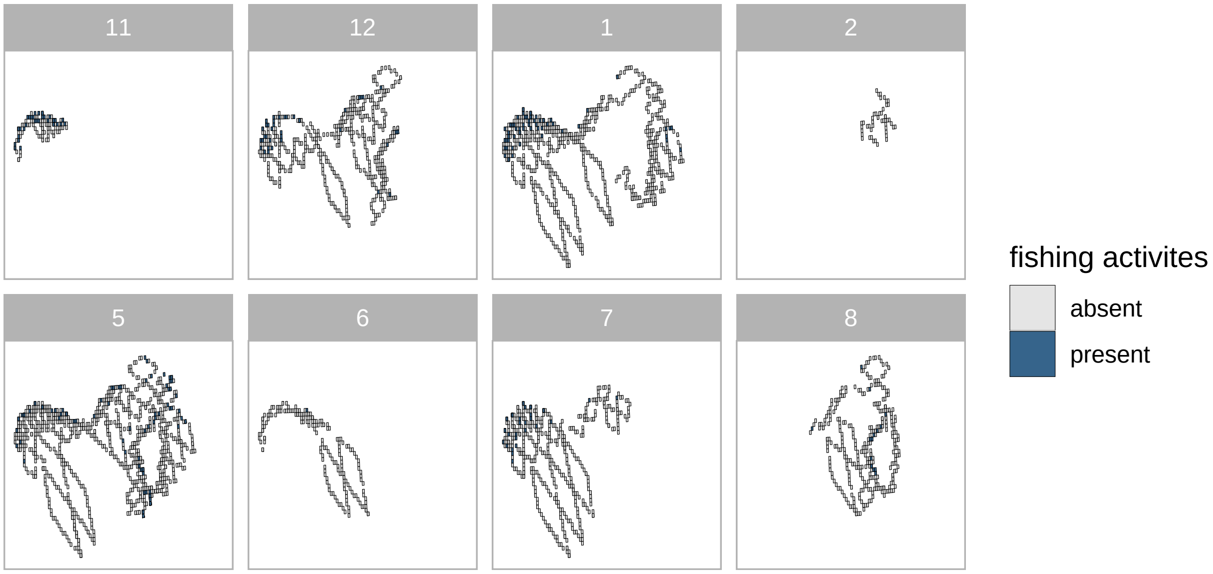
```r
int <- transect_line %>%
  st_intersection(grid)
```

Select those cells which intersect with the transects.

```r
all_df <- grid %>%
  filter(objectid %in% int$objectid) %>%
  st_join(int %>% mutate(peche = as_factor(peche),
                         dolphins = as_factor(dolphins))) %>%
  mutate(month = as_factor(month),
         month = fct_relevel(month, c("11", "12", "1", "2", "5", "6", "7", "8")))
```

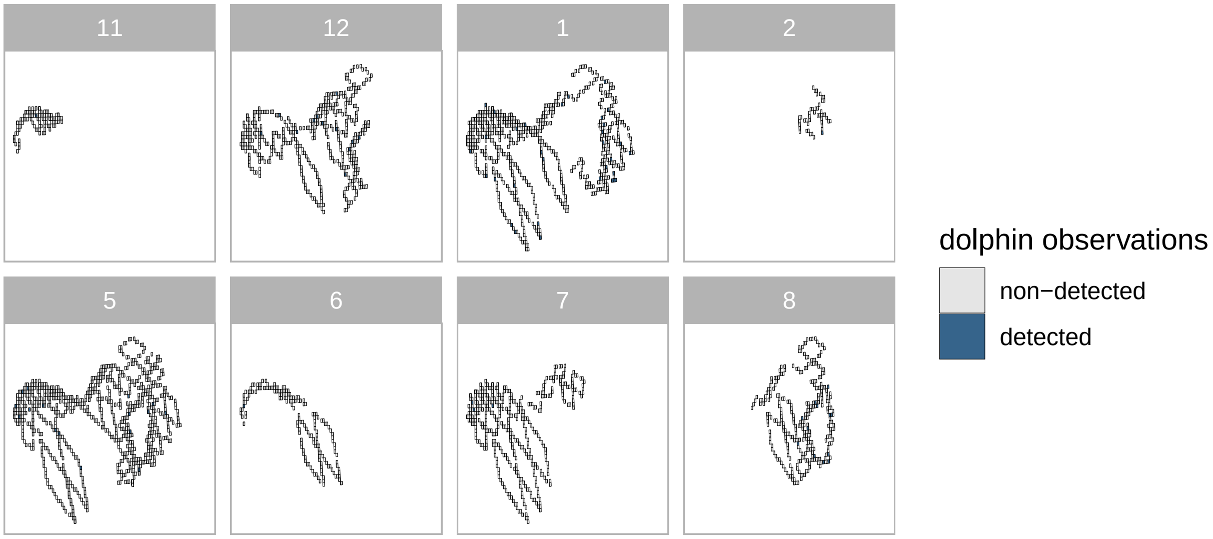
Fishing activities per month.

```r
all_df %>%
  ggplot() +
  geom_sf(lwd = 0.1, color = "black", aes(fill = peche))  +
  coord_sf(xlim = st_bbox(grid)[c(1,3)],
           ylim = st_bbox(grid)[c(2,4)]) +
  scale_fill_manual(values = c('gray90','steelblue4'),
                    name = "fishing activites",
                    labels = c("absent", "present")) +
  facet_wrap(~month, ncol = 4)
```



Dolphin presence per month.

```r
all_df %>%
  ggplot() +
  geom_sf(lwd = 0.1, color = "black", aes(fill = dolphins))  +
  coord_sf(xlim = st_bbox(grid)[c(1,3)],
           ylim = st_bbox(grid)[c(2,4)]) +
  scale_fill_manual(values = c('gray90','steelblue4'),
                    name = "dolphin observations",
                    labels = c("non-detected", "detected")) +
  facet_wrap(~month, ncol = 4)
```



We have everything to build the cells detection-non/detection histories. Basically, the grey cells are gonna be 0's, the blue cells 1's and those cells that are not sampled in a month NA's.

```
all_df %>%
  add_count(transect, wt = as.numeric(dolphins)) %>%
  complete(transect) %>%
  mutate(tursiops = case_when(
  n == 1 ~ "NA",
  n == 2 ~ "non-detected",
  n > 2 ~ "detected"),
  tursiops = as_factor(tursiops)) %>%
  select(transect, dolphins, n, tursiops)
```

```
## # A tibble: 3,744 x 4
##    transect dolphins     n tursiops
##    <chr>    <fct>    <dbl> <fct>
##  1 1772     0            1 NA
##  2 1773     0            1 NA
##  3 1774     0            1 NA
##  4 1776     0            1 NA
##  5 1778     0            1 NA
##  6 1779     0            1 NA
##  7 1780     0            5 detected
##  8 1780     0            5 detected
##  9 1780     0            5 detected
## 10 1780     0            5 detected
## # ... with 3,734 more rows
```

```
#  head() %>%
#  mutate(occ1 = if_else(month == "11" | ))
#  pivot_wider(c(date, season),
#               values_from = dolphins,
#               names_from = month)
```

```
    #
    # mutate(dolphins = fct_recode(dolphins, "ND" = "0", "D" = "1"),
    #      month = fct_recode(month,
    #                          "dec" = "12",
    #                          "nov" = "11",
    #                          "jan" = "1",
    #                          "feb" = "2",
    #                          "may" = "5",
    #                          "jun" = "6",
    #                          "jul" = "7",
    #                          "aug"=  "8")) %>%
```

Build occupancy data.

```
occupancy_df <- all_df %>%
  mutate(dolphins = as.numeric(dolphins),
         peche = as.numeric(peche)) %>%
  group_by(transect, month) %>%
  summarise(dolphins = sum(dolphins),
            fishing = sum(peche)) %>%
  mutate(obs = case_when(
  dolphins == 1 && fishing == 1 ~ "NA",
  dolphins == 1 && fishing == 2 ~ "NA and fishing non-detected",
  dolphins == 2 && fishing == 1 ~ "dolphin non-detected and NA",
  dolphins == 1 && fishing > 2 ~ "NA and fishing detected",
  dolphins > 2 && fishing == 1 ~ "dolphin detected and NA",
  dolphins == 2 && fishing == 2 ~ "dolphin and fishing non-detected",
  dolphins > 2 && fishing == 2 ~ "dolphin detected and fishing non-detected",
  dolphins == 2 && fishing > 2 ~ "dolphin non-detected and fishing detected",
  dolphins > 2 && fishing > 2 ~ "dolphin and fishing detected"),
  obs = as_factor(obs)) %>%
  ungroup() %>%
  select(transect, obs, month)
```

Some checks. I am not sure we can have NA and fishing (non-)detected or dolphin (non-)detected and NA, either a cell is sampled whatever the detection/non-detection, or it is not.

```
occupancy_df %>%
  count(obs)
```

```
## Simple feature collection with 7 features and 2 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 701000 ymin: 5890440 xmax: 1368980 ymax: 6378558
## CRS:            +proj=longlat +init=EPSG:3035
## # A tibble: 7 x 3
##   obs                        n                                           geometry
## * <fct>                  <int>                                 <MULTIPOLYGON [°]>
## 1 NA                       874 (((713739.2 6146319, 713719.7 6155590, 706859.8 ~
## 2 dolphin and fishing d~   460 (((1235908 6075337, 1235155 6084586, 1234402 609~
## 3 NA and fishing non-de~    44 (((1175381 6042771, 1174715 6052030, 1181684 605~
## 4 dolphin non-detected ~    46 (((1197705 6025793, 1197010 6035052, 1203997 603~
## 5 dolphin and fishing n~   325 (((942843.3 5890692, 942509.6 5899986, 942176.1 ~
## 6 dolphin non-detected ~    16 (((1228963 6074775, 1228220 6084025, 1235155 608~
## 7 dolphin detected and ~    15 (((982928.2 5957380, 982536.1 5966663, 989595.2 ~
```

Rebuild occupancy data.

```
occupancy_df <- all_df %>%
  mutate(dolphins = as.numeric(dolphins),
         peche = as.numeric(peche)) %>%
  group_by(transect, month) %>%
  summarise(dolphins = sum(dolphins),
            fishing = sum(peche)) %>%
  mutate(obs = case_when(
  dolphins == 1 && fishing == 1 ~ "NA",
  dolphins == 1 && fishing == 2 ~ "dolphin non-detected and fishing non-detected",
  dolphins == 2 && fishing == 1 ~ "dolphin non-detected and fishing non-detected",
  dolphins == 1 && fishing > 2 ~ "dolphin non-detected and fishing detected",
  dolphins > 2 && fishing == 1 ~ "dolphin detected and fishing non-detected",
  dolphins == 2 && fishing == 2 ~ "dolphin and fishing non-detected",
  dolphins > 2 && fishing == 2 ~ "dolphin detected and fishing non-detected",
  dolphins == 2 && fishing > 2 ~ "dolphin non-detected and fishing detected",
  dolphins > 2 && fishing > 2 ~ "dolphin and fishing detected"),
  obs = as_factor(obs)) %>%
  ungroup() %>%
  select(transect, obs, month)
```

```
occupancy_df %>%
  count(obs)
```

```
## Simple feature collection with 6 features and 2 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 701000 ymin: 5890440 xmax: 1368980 ymax: 6378558
## CRS:            +proj=longlat +init=EPSG:3035
## # A tibble: 6 x 3
##   obs                         n                                          geometry
## * <fct>                   <int>                                <MULTIPOLYGON [°]>
## 1 NA                        874 (((713739.2 6146319, 713719.7 6155590, 706859.8~
## 2 dolphin and fishing de~   460 (((1235908 6075337, 1235155 6084586, 1234402 60~
## 3 dolphin non-detected a~    60 (((1175381 6042771, 1174715 6052030, 1181684 60~
## 4 dolphin non-detected a~    46 (((1197705 6025793, 1197010 6035052, 1203997 60~
## 5 dolphin and fishing no~   325 (((942843.3 5890692, 942509.6 5899986, 942176.1~
## 6 dolphin detected and f~    15 (((982928.2 5957380, 982536.1 5966663, 989595.2~
```

Format the dolphin detections/non-detections for further analyses.

```
dolphin_df <- occupancy_df %>%
  as_tibble() %>%
  select(transect, obs, month) %>%
  mutate(obs = case_when(
  is.na(obs) ~ "NA",
  obs == "dolphin non-detected and fishing non-detected" ~ "0",
  obs == "dolphin non-detected and fishing detected" ~ "0",
  obs == "dolphin detected and fishing non-detected" ~ "1",
  obs == "dolphin and fishing non-detected" ~ "0",
  obs == "dolphin and fishing detected" ~ "1")) %>%
  pivot_wider(names_from = month, values_from = obs)
dolphin_df
```

```
## # A tibble: 1,780 x 9
##    transect ‘1‘   ‘11‘  ‘12‘  ‘2‘   ‘5‘   ‘8‘   ‘6‘   ‘7‘
##    <chr>    <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
##  1 1772     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  2 1773     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  3 1774     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  4 1776     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  5 1778     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  6 1779     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  7 1780     <NA>  1     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  8 1782     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  9 1783     0     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
## 10 1784     0     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
## # ... with 1,770 more rows
```

Format the fishing activities detections/non-detections for further analyses.

```
fishing_df <- occupancy_df %>%
  as_tibble() %>%
  select(transect, obs, month) %>%
  mutate(obs = case_when(
  is.na(obs) ~ "NA",
  obs == "dolphin non-detected and fishing non-detected" ~ "0",
  obs == "dolphin non-detected and fishing detected" ~ "1",
  obs == "dolphin detected and fishing non-detected" ~ "0",
  obs == "dolphin and fishing non-detected" ~ "0",
  obs == "dolphin and fishing detected" ~ "1")) %>%
  pivot_wider(names_from = month, values_from = obs)
fishing_df
```

```
## # A tibble: 1,780 x 9
##    transect ‘1‘   ‘11‘  ‘12‘  ‘2‘   ‘5‘   ‘8‘   ‘6‘   ‘7‘
##    <chr>    <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
##  1 1772     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  2 1773     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  3 1774     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  4 1776     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  5 1778     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  6 1779     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  7 1780     <NA>  1     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  8 1782     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  9 1783     0     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
## 10 1784     1     <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
## # ... with 1,770 more rows
```

Put everything together.

```
y_dolphin <- dolphin_df %>%
  select(-transect) %>%
  mutate_if(is.character, as.numeric) %>%
  as.matrix()
ind_dolphin <- apply(y_dolphin, 1, function(x) all(is.na(x)))
y_dolphin <- y_dolphin[ !ind_dolphin, ]

y_fishing <- fishing_df %>%
  select(-transect) %>%
  mutate_if(is.character, as.numeric) %>%
  as.matrix()
ind_fishing <- apply(y_fishing, 1, function(x) all(is.na(x)))
y_fishing <- y_fishing[ !ind_fishing, ]

y <- list(y_dolphin, y_fishing)
names(y) <- c('dolphin','fishing')
```

Two-species occupancy analysis

Load `unmarked` awesome package.

```
library(unmarked)
```

Get help on the analysis.

```
?unmarked::occuMulti
```

I wrote a tutorial some time ago that might be useful, check out here.

Format data.

```
data <- unmarkedFrameOccuMulti(y = y) #,siteCovs=occ_covs,obsCovs=det_covs)
```

Summary stats.

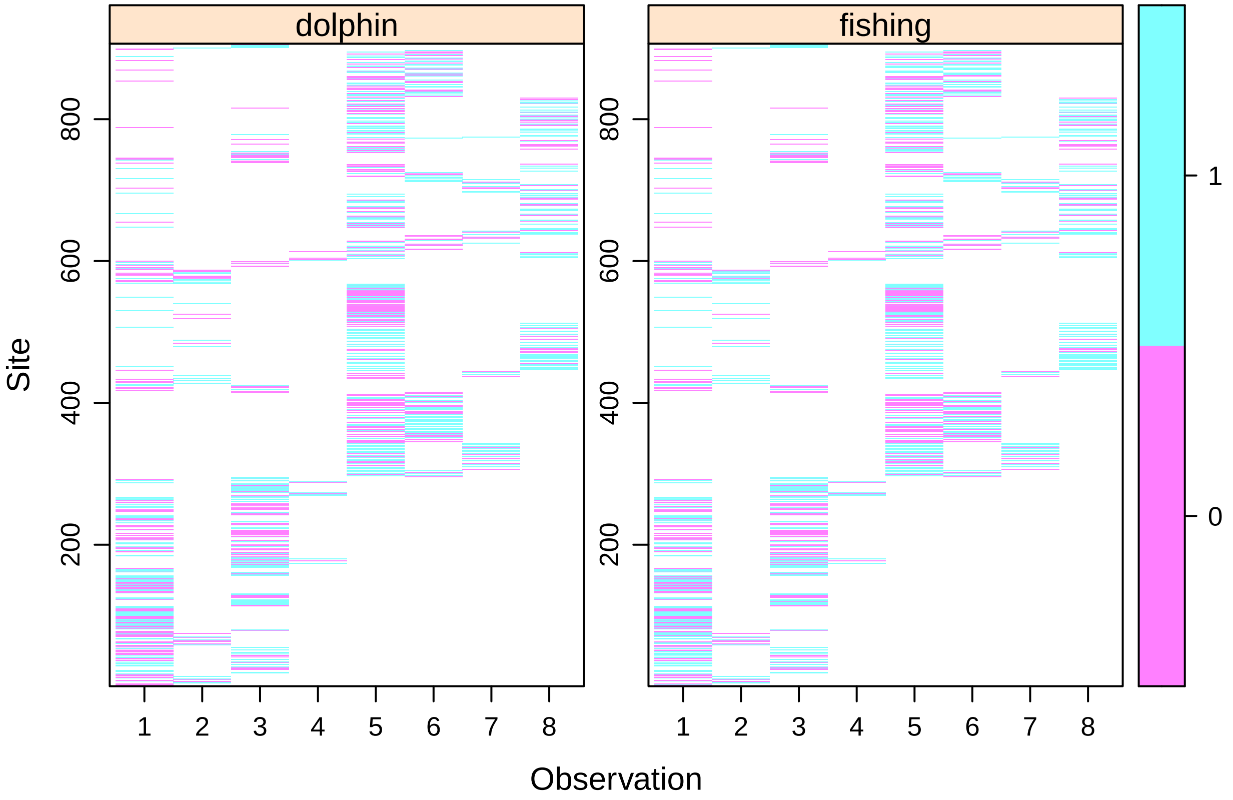
```
summary(data)
```

```
## unmarkedFrame Object
##
## 906 sites
## 2 species: dolphin fishing
## Maximum number of observations per site: 8
## Mean number of observations per site:
```

```
## dolphin: 1  fishing: 1
## Sites with at least one detection:
## dolphin: 475  fishing: 506
## Tabulation of y observations:
## dolphin:
##    0    1 <NA>
##  431  475 6342
## fishing:
##    0    1 <NA>
##  400  506 6342
```

Visualize.

```
plot(data)
```



Specific effects on parameters.

```
occFormulas <- c('~1','~1','~1')
detFormulas <- c('~1','~1')
```

Fit model.

```
fit <- occuMulti(detFormulas, occFormulas, data)
```

Look at output

```
fit
```

```
##
## Call:
## occuMulti(detformulas = detFormulas, stateformulas = occFormulas,
##     data = data)
##
## Occupancy:
##                                 Estimate  SE   z P(>|z|)
## [dolphin] (Intercept)              -7.23 NaN NaN     NaN
## [fishing] (Intercept)              -2.79 NaN NaN     NaN
## [dolphin:fishing] (Intercept)      10.28 NaN NaN     NaN
##
## Detection:
##                         Estimate  SE   z P(>|z|)
## [dolphin] (Intercept)       2.99 NaN NaN     NaN
## [fishing] (Intercept)       3.44 NaN NaN     NaN
##
## AIC: 1689.777
```

Get the natural parameter and detection estimates:

```
mle <- fit@opt$par
names(mle) <- c('f1','f2','f12','lp1','lp2')
mle
```

```
##        f1        f2       f12       lp1       lp2
## -7.231012 -2.789498 10.283023  2.988637  3.440952
```

Get the occupancy estimates:

```
den <- 1 + exp(mle['f1'])+exp(mle['f2'])+exp(mle['f1']+mle['f2']+mle['f12'])
(psi11hat <- exp(mle['f1']+mle['f2']+mle['f12'])/den)
```

```
##       f1
## 0.550377
```

```
(psi10hat <- exp(mle['f1'])/den)
```

```
##           f1
## 0.0003063823
```

```
(psi01hat <- exp(mle['f2'])/den)
```

```
##         f2
## 0.02601288
```

I do it by hand to understand how `unmarked` works. The easy way is to use `predict(fit,'state')`.

Get the detection estimates:

```r
(p1hat <- plogis(mle['lp1']))
```

```
##       lp1
## 0.9520581
```

```r
(p2hat <- plogis(mle['lp2']))
```

```
##       lp2
## 0.9689602
```

Again I do it by hand, but `unmarked` can do it for you with `predict(fit,'det')`.

If we just want to get the parameter estimates directly:

```r
# detection
predict(fit,'det',species=1)[1,]
```

```
##   Predicted  SE lower upper
## 1 0.9520581 NaN   NaN   NaN
```

```r
predict(fit,'det',species=2)[1,]
```

```
##   Predicted  SE lower upper
## 1 0.9689602 NaN   NaN   NaN
```

```r
# marginal occupancy
predict(fit,'state',species=1)[1,]
```

```
## Bootstrapping confidence intervals with 100 samples
```

```
##   Predicted SE lower upper
## 1 0.5506833 NA    NA    NA
```

```r
predict(fit,'state',species=2)[1,]
```

```
## Bootstrapping confidence intervals with 100 samples
```

```
##   Predicted SE lower upper
## 1 0.5763898 NA    NA    NA
```

```r
# conditional occupancy
predict(fit,'state',species=1,cond='fishing')[1,] # species 1 | species 2 present
```

```
## Bootstrapping confidence intervals with 100 samples
```

```
##   Predicted SE lower upper
## 1 0.9548693 NA    NA    NA
```

```r
predict(fit,'state',species=1,cond='-fishing')[1,] # species 1 | species 2 absent
```

```
## Bootstrapping confidence intervals with 100 samples
```

```
##      Predicted SE lower upper
## 1 0.0007232647 NA    NA    NA
```

```r
predict(fit,'state',species=2,cond='dolphin')[1,] # species 2 | species 1 present
```

```
## Bootstrapping confidence intervals with 100 samples
```

```
##   Predicted SE lower upper
## 1 0.9994436 NA    NA    NA
```

```r
predict(fit,'state',species=2,cond='-dolphin')[1,] # species 2 | species 1 absent
```

```
## Bootstrapping confidence intervals with 100 samples
```

```
##    Predicted SE lower upper
## 1 0.05789431 NA    NA    NA
```