Kebersamaan Untuk Lautan RCT: Processing

This page has the following content:

1. Set up workspace and loading the raw data
2. Combining the catch and phase dates data
3. Aggregate at the boat level
4. Normalize the analysis data
5. Processing the release data
6. Save both datasets

By the end of this script, we will have created:

* “DF\_anly\_1.rds”: Which includes the landings and treatment exposure data, aggregated at by vessel-phase.
* “Release\_1.rds”: The release data by boat and date.

### 1) Set up workspace and loading the raw data

### Load packages ###  
library(rstudioapi)  
library(readr)  
library(naniar)  
library(readxl)  
library(GGally)  
library(dplyr)  
library(purrr)  
library(fuzzyjoin)  
  
### Load data as a dataframe ###  
rct\_catch\_raw <- read\_csv("../1 - Pre-processed Raw Data/rct\_catch\_raw.csv")  
Releases\_all <- read\_csv("../1 - Pre-processed Raw Data/Releases\_all.csv")  
phase\_date\_village <- read\_csv("../1 - Pre-processed Raw Data/phase\_date\_village.csv")  
phase\_date\_boat <- read\_csv("../1 - Pre-processed Raw Data/phase\_date\_boat.csv")

**Variable description for raw data**

***Variable description for ‘rct\_catch\_raw’***:

| Variaable | Description |
| --- | --- |
| Date | The date of the observation. |
| boat | The pseudonymized name of the boat. |
| landing | Whether the boat landed. |
| WF\_count | The count of wedgefish catches. |
| HH\_count | The count of hammerhead catches. |
| site | A code corresponding to the site. |
| regency | The name of the regency. |
| group | The treatment group in which the observation occurs. |
| group\_in | The group subjected to the treatment on the current date. |
| treatment | Whether the observation was subject to the treatment (NB. this variable is revised below, as there were minor differences in the exposure duration of boats in several villages for several phases). |
| village | The pseudonymized name of the village. |
| participation | Whether the boat received a payment within a given treatment phase. |
| season | The season. |
| Group 1 | Denoted as 1 if the boat was in treatment group 1. |
| Group 2 | Denoted as 1 if the boat was in treatment group 2. |

***Variable description for ‘Releases\_all’***:

| Variaable | Description |
| --- | --- |
| date | The date of the observation. |
| month | The month of the observation. |
| site | A code corresponding to the site. |
| village | The pseudonymized name of the village. |
| boat | The pseudonymized name of the boat. |
| species | The species for which a payment was given. |
| count | The count of the number of releases of that species. |
| payment | The total payment provided for that observation |
| phase | The phase in which the payment was given (NB. this variable is revised below, as there were minor differences in the exposure duration of boats in several villages for several phases) |
| group | The treatment group in which the observation occurs. |

***Variable description for ‘phase\_date\_village’***:

| Variaable | Description |
| --- | --- |
| Group | The code of the group. |
| Period | The name of the exposure period. |
| Village | The pseudonymized name of the village. |
| Regency | The name of the regency. |
| Start | The start date of the treatment exposure. |
| End | The end date of the treatment exposure. |
| Number of days | The number of days of the treatment exposure (exclusive of the last day, which is included in a revised version of the variable below). |
| Use | An indicator of whether all boats in this village have the same exposure dates and duration (where No, exposure dates from ‘phase\_date\_boat’ are used). |

***Variable description for ‘phase\_date\_boat’***:

| Variaable | Description |
| --- | --- |
| boat | The pseudonymized name of the boat. |
| Regency | The name of the regency. |
| Village | The pseudonymized name of the village. |
| Group | The code of the group. |
| Period | The name of the exposure period. |
| Start | The start date of the treatment exposure. |
| End | The end date of the treatment exposure. |
| Number of days | The number of days of the treatment exposure (exclusive of the last day, which is included in a revised version of the variable below). |

### 2) Combining the catch and phase dates data

We start by specifying the correct data types. First, inspect the data types:

### Data types ###  
str(rct\_catch\_raw)

## spc\_tbl\_ [29,479 × 15] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ Date : Date[1:29479], format: "2022-05-05" "2022-05-05" ...  
## $ boat : num [1:29479] 3 7 9 8 11 12 13 14 18 19 ...  
## $ landing : chr [1:29479] "Y" "N" "Y" "N" ...  
## $ WF\_count : num [1:29479] 0 NA 1 NA NA NA NA NA NA NA ...  
## $ HH\_count : num [1:29479] 2 NA 0 NA NA NA NA NA NA NA ...  
## $ site : chr [1:29479] "TL" "TL" "TL" "TL" ...  
## $ regency : chr [1:29479] "East Lombok" "East Lombok" "East Lombok" "East Lombok" ...  
## $ group : chr [1:29479] "TL2" "TL2" "TL1" "TL1" ...  
## $ group\_in : chr [1:29479] "TL1" "TL1" "TL1" "TL1" ...  
## $ treatment : num [1:29479] 0 0 1 1 1 0 1 0 0 0 ...  
## $ village : num [1:29479] 5 5 5 5 5 5 5 5 5 5 ...  
## $ participation: chr [1:29479] "N" "N" "N" "N" ...  
## $ season : chr [1:29479] "west" "west" "west" "west" ...  
## $ Group 1 : num [1:29479] 0 0 1 1 1 0 1 0 0 0 ...  
## $ Group 2 : num [1:29479] 1 1 0 0 0 1 0 1 1 1 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Date = col\_date(format = ""),  
## .. boat = col\_double(),  
## .. landing = col\_character(),  
## .. WF\_count = col\_double(),  
## .. HH\_count = col\_double(),  
## .. site = col\_character(),  
## .. regency = col\_character(),  
## .. group = col\_character(),  
## .. group\_in = col\_character(),  
## .. treatment = col\_double(),  
## .. village = col\_double(),  
## .. participation = col\_character(),  
## .. season = col\_character(),  
## .. `Group 1` = col\_double(),  
## .. `Group 2` = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

Now convert characters to numeric and factors (where needed).

### Convert character columns to numeric ###  
num\_vars <- c("WF\_count", "HH\_count") # "village"   
rct\_catch\_raw[num\_vars] <- lapply(rct\_catch\_raw[num\_vars], as.numeric)  
str(rct\_catch\_raw[num\_vars])

## tibble [29,479 × 2] (S3: tbl\_df/tbl/data.frame)  
## $ WF\_count: num [1:29479] 0 NA 1 NA NA NA NA NA NA NA ...  
## $ HH\_count: num [1:29479] 2 NA 0 NA NA NA NA NA NA NA ...

rct\_catch <- rct\_catch\_raw  
  
### Convert character columns to factors ###  
char\_vars <- c("treatment", "boat", "landing", "site", "group", "group\_in", "village", "participation", "season", "regency")   
rct\_catch[char\_vars] <- lapply(rct\_catch[char\_vars], as.factor)  
str(rct\_catch[char\_vars])

## tibble [29,479 × 10] (S3: tbl\_df/tbl/data.frame)  
## $ treatment : Factor w/ 2 levels "0","1": 1 1 2 2 2 1 2 1 1 1 ...  
## $ boat : Factor w/ 86 levels "1","2","3","4",..: 3 7 9 8 11 12 13 14 18 19 ...  
## $ landing : Factor w/ 2 levels "N","Y": 2 1 2 1 1 1 1 1 1 1 ...  
## $ site : Factor w/ 2 levels "AJ","TL": 2 2 2 2 2 2 2 2 2 2 ...  
## $ group : Factor w/ 4 levels "AJ1","AJ2","TL1",..: 4 4 3 3 3 4 3 4 4 4 ...  
## $ group\_in : Factor w/ 4 levels "AJ1","AJ2","TL1",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ village : Factor w/ 5 levels "1","2","3","4",..: 5 5 5 5 5 5 5 5 5 5 ...  
## $ participation: Factor w/ 4 levels "AJ1","AJ2","N",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ season : Factor w/ 2 levels "east","west": 2 2 2 2 2 2 2 2 2 2 ...  
## $ regency : Factor w/ 2 levels "Aceh Jaya","East Lombok": 2 2 2 2 2 2 2 2 2 2 ...

Also, we make sure that dates are recognized as such, and create a variable describing the calendar month (‘month’) of each observation and chronological months (‘month\_start’).

# Make sure date is recognised as such   
rct\_catch$Date <- as.Date(rct\_catch$Date, format="%d %B %Y")  
  
### Convert date to month ###   
rct\_catch$month <- as.factor(format(rct\_catch$Date,"%m"))  
  
# Convert date to month since the start of the study   
rct\_catch$month\_start <- as.factor(format(rct\_catch$Date,"%Y-%m"))

Next, we want to remove observations where the boat did not land. However, in some cases, a boat appears to have landed (because there are recorded landings for one species) but were still marked as having not landed.

As a result, the following code indicates that if both WF and HH counts are NA (not 0), then there are no landings. But if either has non-NA (including 0), then they landed.

### Remove observations where boats did not land ###  
rct\_catch\_land <- rct\_catch  
rct\_catch\_land$landing2 <- factor(ifelse(is.na(rct\_catch\_land$WF\_count)==T & is.na(rct\_catch\_land$HH\_count)==T, "N","Y"))   
rct\_catch\_land <- rct\_catch\_land[rct\_catch\_land$landing2 %in% "Y",]  
nrow(rct\_catch\_land)

## [1] 6327

We check how many boats are in the raw vs the landings-only data.

# How many boats in the original and landings only data   
length(unique(rct\_catch$boat))

## [1] 86

length(unique(rct\_catch\_land$boat))

## [1] 86

setdiff(unique(rct\_catch$boat), unique(rct\_catch\_land$boat))

## character(0)

Some boats are recorded as NA for a species, even though they have retained catches for the other species during that landing. In other words, they had 0 retained catches, not NA (which is used when there are no landings). So, we convert these NAs to 0’s.

### Convert NA to 0 ###  
rct\_catch\_land$WF\_count <- ifelse(is.na(rct\_catch\_land$WF\_count), 0, rct\_catch\_land$WF\_count)  
rct\_catch\_land$HH\_count <- ifelse(is.na(rct\_catch\_land$HH\_count), 0, rct\_catch\_land$HH\_count)  
  
### Drop unused levels ###  
rct\_catch\_land <- droplevels(rct\_catch\_land)

Now, we calculate what phase each boat is in. First, we look at the dates for each village. The phase\_date\_village object contains the dates for each village.

# First, make sure the date is recognized as such   
phase\_date\_village$Start <- as.Date(phase\_date\_village$Start, format="%d %B %Y")  
phase\_date\_village$End <- as.Date(phase\_date\_village$End, format="%d %B %Y")  
  
# Inspect the dates for the villages  
phase\_date\_village

## # A tibble: 18 × 8  
## Group Period Village Regency Start End `Number of days` Use   
## <chr> <chr> <dbl> <chr> <date> <date> <dbl> <chr>  
## 1 TL1 P1 1 East Lombok 2022-05-05 2022-07-10 66 Yes   
## 2 TL1 P1 5 East Lombok 2022-05-05 2022-07-10 66 Yes   
## 3 TL2 P2 1 East Lombok 2022-07-11 2022-11-15 127 Yes   
## 4 TL2 P2 5 East Lombok 2022-07-11 2022-11-15 127 Yes   
## 5 TL1 P3 1 East Lombok 2022-11-16 2023-03-31 135 Yes   
## 6 TL1 P3 5 East Lombok 2022-11-16 2023-03-31 135 Yes   
## 7 TL2 P4 1 East Lombok 2023-04-01 2023-05-21 50 Yes   
## 8 TL2 P4 5 East Lombok 2023-04-01 2023-05-21 50 Yes   
## 9 AJ1 P1 4 Aceh Jaya 2022-05-30 2022-08-31 93 Yes   
## 10 AJ2 P2 4 Aceh Jaya 2022-09-01 2023-01-13 134 Yes   
## 11 AJ2 P1 3 Aceh Jaya 2022-09-02 2023-01-12 132 Yes   
## 12 AJ2 P1 2 Aceh Jaya 2022-09-02 2023-01-11 131 Yes   
## 13 AJ1 P3 4 Aceh Jaya 2023-01-14 2023-05-31 137 No   
## 14 AJ1 P2 3 Aceh Jaya 2023-01-13 2023-06-01 139 No   
## 15 AJ1 P2 2 Aceh Jaya 2023-01-12 2023-06-01 140 Yes   
## 16 AJ2 P4 4 Aceh Jaya 2023-06-01 2023-08-17 77 No   
## 17 AJ2 P3 3 Aceh Jaya 2023-06-02 2023-08-17 76 No   
## 18 AJ2 P3 2 Aceh Jaya 2023-06-02 2023-08-17 76 Yes

However, in some of these villages, some boats entered or left the intervention at slightly different dates. In the above, the villages and phases with these cases are indicated as “No” in the use column. So, we can inspect the boats in the cases that have different dates.

# First, make sure the date is recognized as such   
phase\_date\_boat$Start <- as.Date(phase\_date\_boat$Start, format="%d %B %Y")  
phase\_date\_boat$End <- as.Date(phase\_date\_boat$End, format="%d %B %Y")  
  
# Inspect the dates for the boat  
phase\_date\_boat

## # A tibble: 24 × 8  
## boat Regency Village Group Period Start End `Number of days`  
## <dbl> <chr> <dbl> <chr> <chr> <date> <date> <dbl>  
## 1 28 Aceh Jaya 4 AJ2 P4 2023-06-01 2023-08-17 77  
## 2 30 Aceh Jaya 4 AJ2 P4 2023-06-01 2023-08-17 77  
## 3 31 Aceh Jaya 4 AJ2 P4 2023-06-01 2023-08-17 77  
## 4 33 Aceh Jaya 4 AJ2 P4 2023-06-01 2023-08-17 77  
## 5 35 Aceh Jaya 4 AJ2 P4 2023-06-02 2023-08-17 76  
## 6 32 Aceh Jaya 3 AJ1 P2 2023-01-14 2023-06-01 138  
## 7 43 Aceh Jaya 4 AJ2 P4 2023-06-02 2023-08-17 76  
## 8 34 Aceh Jaya 3 AJ1 P2 2023-01-14 2023-05-31 137  
## 9 46 Aceh Jaya 4 AJ2 P4 2023-06-02 2023-08-17 76  
## 10 37 Aceh Jaya 3 AJ1 P2 2023-01-14 2023-05-31 137  
## # ℹ 14 more rows

Then, we can calculate the number of days in each phase, which includes both the start and end date.

# Number of days in each phase (add 1 to include both the start and end dates)  
phase\_date\_village$day\_count <- as.numeric(phase\_date\_village$End - phase\_date\_village$Start) + 1  
phase\_date\_boat$day\_count <- as.numeric(phase\_date\_boat$End - phase\_date\_boat$Start) + 1  
  
# Take a look again   
phase\_date\_village[c( "Group","Period", "Village","Start","End", "day\_count" )]

## # A tibble: 18 × 6  
## Group Period Village Start End day\_count  
## <chr> <chr> <dbl> <date> <date> <dbl>  
## 1 TL1 P1 1 2022-05-05 2022-07-10 67  
## 2 TL1 P1 5 2022-05-05 2022-07-10 67  
## 3 TL2 P2 1 2022-07-11 2022-11-15 128  
## 4 TL2 P2 5 2022-07-11 2022-11-15 128  
## 5 TL1 P3 1 2022-11-16 2023-03-31 136  
## 6 TL1 P3 5 2022-11-16 2023-03-31 136  
## 7 TL2 P4 1 2023-04-01 2023-05-21 51  
## 8 TL2 P4 5 2023-04-01 2023-05-21 51  
## 9 AJ1 P1 4 2022-05-30 2022-08-31 94  
## 10 AJ2 P2 4 2022-09-01 2023-01-13 135  
## 11 AJ2 P1 3 2022-09-02 2023-01-12 133  
## 12 AJ2 P1 2 2022-09-02 2023-01-11 132  
## 13 AJ1 P3 4 2023-01-14 2023-05-31 138  
## 14 AJ1 P2 3 2023-01-13 2023-06-01 140  
## 15 AJ1 P2 2 2023-01-12 2023-06-01 141  
## 16 AJ2 P4 4 2023-06-01 2023-08-17 78  
## 17 AJ2 P3 3 2023-06-02 2023-08-17 77  
## 18 AJ2 P3 2 2023-06-02 2023-08-17 77

Now we have to create another dataset, with boats in the villages with the varying treatment exposure.

# Check village names match  
setdiff(phase\_date\_boat$Village, rct\_catch\_land$village) # Yes

## numeric(0)

# And check the boat names match  
setdiff(phase\_date\_boat$boat, rct\_catch\_land$boat) # Yes

## numeric(0)

# Second, merge rct\_catch\_land and phase\_date\_boat using fuzzyjoin ###  
# Fuzzy left join  
nrow(rct\_catch\_land)

## [1] 6327

rct\_catch\_date\_b <- fuzzyjoin::fuzzy\_left\_join(rct\_catch\_land, phase\_date\_boat,  
 by = c("boat" = "boat",  
 "village" = "Village",  
 "Date" = "Start",  
 "Date" = "End"),  
 match\_fun = list(`==`,`==`, `>=`, `<=`))  
  
# Drop the rows with NA for phase (this will be added later)  
rct\_catch\_date\_b <- subset(rct\_catch\_date\_b, !is.na(Period))  
  
# Drop the duplicated rows  
rct\_catch\_date\_b <- dplyr::select(rct\_catch\_date\_b, -ends\_with(".y"))  
colnames(rct\_catch\_date\_b) <- gsub("\\.x", "", colnames(rct\_catch\_date\_b))

Now, we match all catch data against the phase\_date\_village dates.

# Merge rct\_catch\_land and phase\_date\_village using fuzzyjoin ###  
# Fuzzy left join   
rct\_catch\_comb\_1 <- fuzzyjoin::fuzzy\_left\_join(rct\_catch\_land, phase\_date\_village,  
 by = c("village" = "Village",  
 "Date" = "Start",  
 "Date" = "End"),  
 match\_fun = list(`==`, `>=`, `<=`))  
  
# Do row counts match?  
nrow(rct\_catch\_comb\_1) == nrow(rct\_catch\_land)

## [1] TRUE

# Any missing data?  
table(is.na(rct\_catch\_comb\_1$Period))

##   
## FALSE   
## 6327

Now, all boats have been assigned phases. However, we know that some have the wrong count of days.

So, we want to substitute the day\_count, start and end date, period, and group variables in rct\_catch\_comb\_1 with the values from the same variables in rct\_catch\_date\_b where observations match, based on the variables date, boat, and village match. We keep the original values in rct\_catch\_comb\_1 for all other observations.

# Perform a left join to merge data based on 'Date', 'boat', and 'village'  
merged\_df <- rct\_catch\_comb\_1 %>%  
 left\_join(rct\_catch\_date\_b, by = c("Date", "boat", "village", "WF\_count","HH\_count", "treatment", "site"), suffix = c("\_comb1", "\_dateb"))  
  
# Update the variables conditionally  
final\_df <- merged\_df %>%  
 mutate(day\_count = ifelse(!is.na(day\_count\_dateb), day\_count\_dateb, day\_count\_comb1),  
 Start\_dat = ifelse(!is.na(Start\_dateb), Start\_dateb, Start\_comb1),  
 End\_date = ifelse(!is.na(End\_dateb), End\_dateb, End\_comb1),  
 Period = ifelse(!is.na(Period\_dateb), Period\_dateb, Period\_comb1),  
 Group = ifelse(!is.na(Group\_dateb), Group\_dateb, Group\_comb1),  
 Group\_in = ifelse(!is.na(group\_in\_dateb), group\_in\_dateb, group\_in\_comb1)) %>%  
 select(-matches("\_dateb"), -matches("\_comb1"))  
  
# Do the rows match   
nrow(final\_df) == nrow(rct\_catch\_comb\_1)

## [1] TRUE

# Are there duplicates?   
duplicates <- final\_df %>%  
 group\_by(boat, Date) %>%  
 filter(n() > 1) %>%  
 ungroup()  
  
# Duplicates?   
nrow(duplicates)

## [1] 0

Rename Period to exp\_phase.

# Rename  
final\_df <- final\_df %>% dplyr::rename(phase\_exp = Period)

### 3) Aggregate at the boat level

The next step is to aggregate by boat in each phase. We do this and check to see if any boats are duplicated in the same phase.

### Aggregate by boat in each phase ###  
group\_count <- function(df) {  
 df\_out <- df %>% group\_by(boat, phase\_exp, village, site) %>%  
 dplyr::summarise(WF\_count=sum(WF\_count),  
 HH\_count=sum(HH\_count),  
 Treatment = unique(treatment),  
 day\_count = unique(day\_count),  
 .groups = 'drop')  
 return(df\_out)  
}  
rct\_catch\_land\_agg <- group\_count(final\_df)  
  
# Check for duplicates based on the "boat" and "phase\_exp" columns  
duplicates <- rct\_catch\_land\_agg[duplicated(rct\_catch\_land\_agg[c("boat", "phase\_exp")]) | duplicated(rct\_catch\_land\_agg[c("boat", "phase\_exp")], fromLast = TRUE), ]  
  
# Duplicates?  
nrow(duplicates)

## [1] 28

head(duplicates)

## # A tibble: 6 × 8  
## boat phase\_exp village site WF\_count HH\_count Treatment day\_count  
## <fct> <chr> <fct> <fct> <dbl> <dbl> <fct> <dbl>  
## 1 33 P3 4 AJ 2 0 1 138  
## 2 33 P3 4 AJ 2 0 0 138  
## 3 39 P2 2 AJ 0 0 0 141  
## 4 39 P2 2 AJ 0 0 1 141  
## 5 40 P2 2 AJ 11 11 1 141  
## 6 40 P2 2 AJ 11 11 0 141

Above, we can see that there are duplicates (which indicate that some boats are recorded as both treatment and control within the same phase). This is expected, as we know that the Treatment variable is not 100% reliable. We look at the affected vessels.

# Unique rows of boats that are classified as both treatment and control within a given phase.   
duplicates\_unique <- duplicates %>% distinct(boat, .keep\_all = TRUE)  
  
# Which are the error rows   
error\_rows <- inner\_join(final\_df, duplicates\_unique, by = c("boat", "phase\_exp"))  
  
# Drop the erroneous Treatment  
error\_rows$Treatment <- NULL  
  
# Drop the duplicated rows   
error\_rows <- dplyr::select(error\_rows, -ends\_with(".y"))  
colnames(error\_rows) <- gsub("\\.x", "", colnames(error\_rows))  
  
# Look at the first 10  
head(error\_rows)

## # A tibble: 6 × 14  
## Date boat WF\_count HH\_count site treatment village Use day\_count  
## <date> <fct> <dbl> <dbl> <fct> <fct> <fct> <chr> <dbl>  
## 1 2023-01-12 61 0 0 AJ 1 2 Yes 141  
## 2 2023-01-12 58 0 0 AJ 1 2 Yes 141  
## 3 2023-01-12 40 0 0 AJ 1 2 Yes 141  
## 4 2023-01-12 62 0 0 AJ 0 2 Yes 141  
## 5 2023-01-12 45 0 0 AJ 0 2 Yes 141  
## 6 2023-01-12 55 0 0 AJ 0 2 Yes 141  
## # ℹ 5 more variables: Start\_dat <dbl>, End\_date <dbl>, phase\_exp <chr>,  
## # Group <chr>, Group\_in <int>

So, as the first step to help correct this, this following code performs a series of data transformation steps on the error\_rows dataframe to uniformly assign a Treatment value within each group defined by boat, phase\_exp, village, and site. If all other observations for that vessel in that phase were considered in one treatment or control group, but we have an outlier where on one day it is considered to be in the other, then we count the number observations by group and go with the majority.

error\_rows\_modified <- error\_rows %>%  
 group\_by(boat, phase\_exp, village, site) %>%  
 # Calculate the most common Treatment within each group  
 mutate(  
 common\_Treatment = if\_else(sum(treatment == 1, na.rm = TRUE) >= sum(treatment == 0, na.rm = TRUE), 1, 0)  
 ) %>%  
 # Apply the common Treatment to all rows in the group  
 mutate(treatment = common\_Treatment) %>%  
 # Selectively remove the helper column used for calculation  
 select(-common\_Treatment) %>%  
 ungroup()  
  
# View the modified dataframe  
head(error\_rows\_modified)

## # A tibble: 6 × 14  
## Date boat WF\_count HH\_count site treatment village Use day\_count  
## <date> <fct> <dbl> <dbl> <fct> <dbl> <fct> <chr> <dbl>  
## 1 2023-01-12 61 0 0 AJ 0 2 Yes 141  
## 2 2023-01-12 58 0 0 AJ 0 2 Yes 141  
## 3 2023-01-12 40 0 0 AJ 0 2 Yes 141  
## 4 2023-01-12 62 0 0 AJ 1 2 Yes 141  
## 5 2023-01-12 45 0 0 AJ 1 2 Yes 141  
## 6 2023-01-12 55 0 0 AJ 1 2 Yes 141  
## # ℹ 5 more variables: Start\_dat <dbl>, End\_date <dbl>, phase\_exp <chr>,  
## # Group <chr>, Group\_in <int>

The next step is to re-assign the treatment variable in final\_df, based on the corrected treatment value in error\_rows\_modified (only for those rows with the corrected treatment values).

# Code to assign the corrected treatment values based on error\_rows\_modified   
final\_df\_modified <- final\_df %>%  
 left\_join(error\_rows\_modified %>% select(Date, boat, village, treatment), by = c("Date", "boat", "village")) %>%  
 # This is needed, since without converting to a factor, the output includes 0, 1, and 2.   
 mutate(  
 treatment.x = as.factor(as.character(treatment.x)),  
 treatment.y = as.factor(as.character(treatment.y)),  
 # Determine final treatment value; use modified value if present, otherwise retain original  
 treatment = ifelse(is.na(treatment.y), treatment.x, treatment.y)  
 ) %>%  
 # Recode treatment levels from 1 and 2 to 0 and 1  
 mutate(  
 treatment = factor(recode(treatment, `1` = 0, `2` = 1))  
 )  
  
# Convert back to numeric   
final\_df\_modified$treatment <- as.numeric(as.character(final\_df\_modified$treatment))

How many WF and HH catches were moved between the treatment and control groups? First, we look at the number of WF and HH that were originally considered in the treatment group and are now in the control.

# Create a subset where treatment.x = 1 and treatment.y = 0  
subset\_tx1\_ty0 <- final\_df\_modified %>%  
 filter(as.character(treatment.x) == "1" & as.character(treatment.y) == "0")  
  
# Count the sum of WF and HH   
sum(subset\_tx1\_ty0$WF\_count)

## [1] 0

sum(subset\_tx1\_ty0$HH\_count)

## [1] 76

And the number who were originally considered in the control and are now in the treatment.

# Create a subset where treatment.x = 0 and treatment.y = 1  
subset\_tx0\_ty1 <- final\_df\_modified %>%  
 filter(as.character(treatment.x) == "0" & as.character(treatment.y) == "1")  
  
# Count the sum of WF and HH  
sum(subset\_tx0\_ty1$WF\_count)

## [1] 0

sum(subset\_tx0\_ty1$HH\_count)

## [1] 0

So the net difference is that there are 76 HH that were originally in the treatment group and are now in the control. Let’s double check this change.

Now we aggregate by boat and phase again. We find one duplicate, boat 51. The observation with 138 day\_count’s is the correct one, and we’re going to manually change this in the aggregated data.

### Aggregate by boat in each phase ###  
rct\_catch\_land\_agg2 <- group\_count(final\_df\_modified)  
  
# Check for duplicates based on the "boat" and "phase\_exp" columns  
duplicates <- rct\_catch\_land\_agg2[duplicated(rct\_catch\_land\_agg2[c("boat", "phase\_exp")]) | duplicated(rct\_catch\_land\_agg2[c("boat", "phase\_exp")], fromLast = TRUE), ]  
  
# Duplicates?  
nrow(duplicates)

## [1] 2

head(duplicates)

## # A tibble: 2 × 8  
## boat phase\_exp village site WF\_count HH\_count Treatment day\_count  
## <fct> <chr> <fct> <fct> <dbl> <dbl> <dbl> <dbl>  
## 1 51 P2 3 AJ 0 278 1 138  
## 2 51 P2 3 AJ 0 278 1 140

# Let's manually change the day\_count for km\_nasrol on the "2023-06-01"  
final\_df\_modified$day\_count <- ifelse(final\_df\_modified$Date == "2023-06-01" &  
 final\_df\_modified$boat == "51" &   
 final\_df\_modified$day\_count == 140, 138,   
 final\_df\_modified$day\_count)

Finally, we aggregate again and check for duplicates. There are no more duplicates (meaning that each boat-phase combination is only in the treatment or control (not both) and each has only one day\_count for that phase.

### Aggregate by boat in each phase ###  
rct\_catch\_land\_agg <- group\_count(final\_df\_modified)  
  
# Check for duplicates based on the "boat" and "phase\_exp" columns  
duplicates <- rct\_catch\_land\_agg[duplicated(rct\_catch\_land\_agg[c("boat", "phase\_exp")]) | duplicated(rct\_catch\_land\_agg[c("boat", "phase\_exp")], fromLast = TRUE), ]  
  
# Duplicates?  
nrow(duplicates)

## [1] 0

Check again that we have no lost or gained any catches.

# Do they match?  
sum(rct\_catch\_land\_agg$WF\_count, na.rm = T) == sum(rct\_catch\_raw$WF\_count, na.rm = T)

## [1] TRUE

sum(rct\_catch\_land\_agg$HH\_count, na.rm = T) == sum(rct\_catch\_raw$HH\_count, na.rm = T)

## [1] TRUE

We see the expected change in 76 HH between the treatment and control groups.

# (Check catches among the treatment group)  
rct\_catch\_land\_agg\_treat <- rct\_catch\_land\_agg %>% filter(Treatment != 0)  
rct\_catch\_raw\_treat <- rct\_catch\_raw %>% filter(treatment != 0)  
sum(rct\_catch\_raw\_treat$WF\_count, na.rm = T) - sum(rct\_catch\_land\_agg\_treat$WF\_count, na.rm = T)

## [1] 0

sum(rct\_catch\_raw\_treat$HH\_count, na.rm = T) - sum(rct\_catch\_land\_agg\_treat$HH\_count, na.rm = T)

## [1] 76

### 4) Normalize and save the analysis data

Now we normalize the data and save DF\_anly\_1.

### Name as single DF  
DF\_anly\_1 <- rct\_catch\_land\_agg

First, we normalize the count data, meaning we want to work out the average WF and HH catches for each vessel-phase, by dividing the total catches that phase by the day\_count.

# Normalize   
DF\_anly\_1$WF\_count\_norm <- DF\_anly\_1$WF\_count/as.numeric(DF\_anly\_1$day\_count)  
DF\_anly\_1$HH\_count\_norm <- DF\_anly\_1$HH\_count/as.numeric(DF\_anly\_1$day\_count)

### 5) Processing the release data

Now we process the release data. First, we subset to the variables we are interested in.

# Subset variables   
release\_sub <- Releases\_all[,c("boat", "date", "species", "count", "village", "phase")]  
release\_sub <- subset(release\_sub, species!="GF")

We make a note of the total releases by species.

### Total releases by species? ###  
record <- release\_sub %>%  
 group\_by(species) %>%  
 summarise(count = sum(count))  
record

## # A tibble: 2 × 2  
## species count  
## <chr> <dbl>  
## 1 HH 364  
## 2 WF 475

Now we aggregate the releases by boat and date, to account for the fact that there are sometimes more than one recorded payment per boat in a given day.

### Aggregate by boat and date ###   
nrow(release\_sub)

## [1] 726

release\_sub <- release\_sub %>%  
 group\_by(boat, date, species, village, phase) %>%  
 summarise(count = sum(count)  
 )

## `summarise()` has grouped output by 'boat', 'date', 'species', 'village'. You  
## can override using the `.groups` argument.

nrow(release\_sub)

## [1] 214

We check that this aggregated data still matches the raw release data, which it does.

### Checking releases match ###  
release\_sub %>%  
 group\_by(species) %>%  
 summarise(count = sum(count)) == record

## species count  
## [1,] TRUE TRUE  
## [2,] TRUE TRUE

Now, we transform the shape of the dataset.

# Subset the data for 'WF' and 'HH' species  
wf\_df <- subset(release\_sub, species == "WF", select = c(date, boat, count, village, phase))  
hh\_df <- subset(release\_sub, species == "HH", select = c(date, boat, count, village, phase))  
  
# Merge the data frames based on 'date' and 'boat'  
merged\_df <- merge(wf\_df, hh\_df, by = c("date", "boat", "village", "phase"), all = TRUE, suffixes = c("\_WF\_R", "\_HH\_R"))  
head(merged\_df)

## date boat village phase count\_WF\_R count\_HH\_R  
## 1 2022-05-27 4 1 P1 1 NA  
## 2 2022-06-02 47 4 P1 1 NA  
## 3 2022-06-04 44 4 P1 NA 1  
## 4 2022-06-05 44 4 P1 1 NA  
## 5 2022-06-09 47 4 P1 1 NA  
## 6 2022-06-20 47 4 P1 1 NA

nrow(merged\_df)

## [1] 209

We replace NA’s with 0’s - this is where there a vessel has been paid on a given data for the release of one species but not the other, so we are indicating that for that date there was no releases for that species.

# Fill missing counts with 0  
merged\_df[is.na(merged\_df)] <- 0

# Set the correct date format  
merged\_df$date <- as.Date(merged\_df$date, format="%d %B %Y")  
  
# Merge   
merged\_df\_dt\_1 <- fuzzyjoin::fuzzy\_left\_join(merged\_df, phase\_date\_village,  
 by = c("village" = "Village",  
 "date" = "Start",  
 "date" = "End"),  
 match\_fun = list(`==`, `>=`, `<=`))  
  
# Check the number of rows match   
nrow(merged\_df\_dt\_1) == nrow(merged\_df)

## [1] TRUE

# Check for completeness  
merged\_df\_dt\_1 %>%  
 mutate(correct\_match = village == Village & date >= Start & date <= End) %>%  
 summarise(all\_matched = all(correct\_match, na.rm = TRUE))

## all\_matched  
## 1 TRUE

# Check that all observations have a phase   
merged\_df\_dt\_1 %>%  
 filter(is.na(Period))

## date boat village phase count\_WF\_R count\_HH\_R Group Period Village  
## 1 2022-09-01 29 3 P1 0 1 <NA> <NA> NA  
## Regency Start End Number of days Use day\_count  
## 1 <NA> <NA> <NA> NA <NA> NA

# Boat 29 did not have a correctly coded period - but it is P1, so we add this in manually   
merged\_df\_dt\_2 <- merged\_df\_dt\_1 %>%  
 mutate(Period = if\_else(date == as.Date("2022-09-01") & boat == 29 & village == 3 &   
 count\_WF\_R == 0 & count\_HH\_R == 1 & is.na(Period),   
 "P1", Period))  
  
# Check again that all observations have a phase   
merged\_df\_dt\_2 %>%  
 filter(is.na(Period))

## [1] date boat village phase count\_WF\_R   
## [6] count\_HH\_R Group Period Village Regency   
## [11] Start End Number of days Use day\_count   
## <0 rows> (or 0-length row.names)

# Drop unused variables   
keeps <- c("date","boat","village","Period","count\_WF\_R","count\_HH\_R")  
merged\_df <- merged\_df\_dt\_2[keeps]  
  
# Rename variable   
merged\_df <- merged\_df %>% dplyr::rename(phase\_exp = Period)

Finally, we want to add a ‘recipient’ variable that indicates if a given boat ever received a release payment.

# Check if 'boat' in DF\_anly\_1 is in 'boat' of merged\_df and assign the variable recipient a value of 1 if true, 0 if false  
DF\_anly\_1$recipient <- as.integer(DF\_anly\_1$boat %in% merged\_df$boat)  
  
# Check the new variable has been created as expected  
length(unique(DF\_anly\_1$boat[DF\_anly\_1$recipient == 1]))

## [1] 29

### 6) Save both datasets

**Variable description for processed data**

***Variable description for ‘DF\_anly\_1’***:

| Variaable | Description |
| --- | --- |
| boat | The pseudonymized name of the boat. |
| phase\_exp | The name of the treatment phase. |
| village | The pseudonymized name of the village. |
| site | A code corresponding to the site. |
| WF\_count | The aggregated count of wedgefish catches. |
| HH\_count | The aggregated count of hammerhead catches. |
| Treatment | Whether the boat was exposed to the treatment for that phase. |
| day\_count | The number of days the boat was exposed to the treatment. |
| WF\_count\_norm | The aggregated count of wedgefish catches divided by the number of exposure days. |
| HH\_count\_norm | The aggregated count of hammerhead catches divided by the number of exposure days. |
| recipient | Whether the boat received a payment during that phase. |

***Variable description for ‘merged\_df’ (renamed to ‘Release\_1’ below)***:

| Variaable | Description |
| --- | --- |
| date | The date of the observation. |
| boat | The pseudonymized name of the boat. |
| village | The pseudonymized name of the village. |
| phase\_exp | The name of the treatment phase. |
| count\_WF\_R | The count of wedgefish releases. |
| count\_HH\_R | The count of hammerhead releases. |

Now, we save the data.

### Data type ###  
DF\_anly\_1$village <- as.factor(DF\_anly\_1$village)  
DF\_anly\_1$boat <- as.factor(DF\_anly\_1$boat)  
str(DF\_anly\_1)

## tibble [261 × 11] (S3: tbl\_df/tbl/data.frame)  
## $ boat : Factor w/ 86 levels "1","2","3","4",..: 1 1 1 1 2 2 2 3 3 3 ...  
## $ phase\_exp : chr [1:261] "P1" "P2" "P3" "P4" ...  
## $ village : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 5 5 5 ...  
## $ site : Factor w/ 2 levels "AJ","TL": 2 2 2 2 2 2 2 2 2 2 ...  
## $ WF\_count : num [1:261] 0 1 0 2 1 1 3 0 0 0 ...  
## $ HH\_count : num [1:261] 3 11 2 0 0 1 7 3 9 12 ...  
## $ Treatment : num [1:261] 1 0 1 0 0 1 0 0 1 0 ...  
## $ day\_count : num [1:261] 67 128 136 51 67 128 136 67 128 136 ...  
## $ WF\_count\_norm: num [1:261] 0 0.00781 0 0.03922 0.01493 ...  
## $ HH\_count\_norm: num [1:261] 0.0448 0.0859 0.0147 0 0 ...  
## $ recipient : int [1:261] 0 0 0 0 0 0 0 0 0 0 ...

### Save an object to a file ###  
saveRDS(DF\_anly\_1, file = "../3 - Post-processed Data for Analysis/DF\_anly\_1.rds")

### Save an object to a file ###  
saveRDS(merged\_df, file = "../3 - Post-processed Data for Analysis/Release\_1.rds")