QFISH

Sumia Kabir Munia

2025-09-19

knitr::opts\_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)  
library(tidyverse)

## Warning: package 'ggplot2' was built under R version 4.5.1

## Warning: package 'tidyr' was built under R version 4.5.1

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(janitor)

## Warning: package 'janitor' was built under R version 4.5.1

##   
## Attaching package: 'janitor'  
##   
## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

data <- read.csv("C:/Users/sumia/OneDrive/Documents/MB5370/GitHub/MB5370\_Mod04\_Personal/data/export.csv")  
dim(data); data[1:5, 1:12]

## [1] 16 127

## X X.1 X.2 SpeciesGroup X.3 X.4 X.5 X.6 X.7  
## 1 CalenderYear 2001 2001 2001 2001 2001 Total 2002 2002 2002  
## 2 Area Mammal Other Shark Turtle Mammal Other Shark  
## 3 Bribie Island   
## 4 Bundaberg 1 62 12 75 3 62  
## 5 Cairns 1 16 79 4 100 18 46  
## X.8 X.9 X.10  
## 1 2002 2002 Total 2003  
## 2 Turtle Mammal  
## 3   
## 4 7 72   
## 5 6 70

view(data)

###———-Tidy Data————###

suppressPackageStartupMessages({  
 library(dplyr)  
 library(ggplot2)  
 library(stringr)  
})

# row 1 = Year headers, row 2 = Species headers, col 1 = Area, rows 3 = data  
# make everything character  
raw <- as.data.frame(data, stringsAsFactors = FALSE)  
raw[] <- lapply(raw, \(x) as.character(x))

# Helpers  
#Normalize species labels to the four groups we care about and drop totals/calendars  
ffill <- function(x) {   
 last <- NA\_character\_  
 for (i in seq\_along(x)) {  
 if (!is.na(x[i]) && nzchar(x[i])) last <- x[i]  
 if (is.na(x[i]) || !nzchar(x[i])) x[i] <- last  
 }  
 x  
}  
norm\_group <- function(g) {   
 g <- tolower(trimws(ifelse(is.null(g), "", g)))  
 g <- gsub("[^a-z0-9]+", " ", g)  
 g <- trimws(g)  
 if (g == "" || grepl("total|calendar", g)) return(NA\_character\_)  
 if (g %in% c("shark","sharks")) return("Shark")  
 if (g %in% c("turtle","turtles")) return("Turtle")  
 if (g %in% c("mammal","mammals")) return("Mammal")  
 if (g %in% c("other","others")) return("Other")  
 return(NA\_character\_) # anything else = ignore  
}

## Header rows & area vector  
hdr\_year <- as.character(raw[1, ])  
hdr\_group <- as.character(raw[2, ])  
area\_vec <- as.character(raw[-c(1,2), 1])  
# Forward-fill year row so every data column has a year  
yr\_ff <- ffill(hdr\_year)

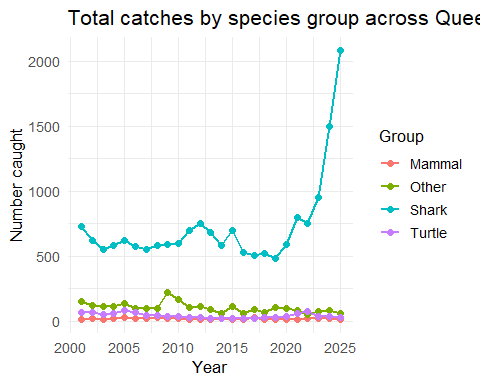
##–Build tidy rows column-by-column–##

parts <- list()  
ncol\_raw <- ncol(raw)  
for (j in 2:ncol\_raw) {  
 # extract year (4 digits anywhere in header cell)  
 yr <- sub(".\*?(\\b[0-9]{4}\\b).\*", "\\1", yr\_ff[j])  
 if (!grepl("\\b[0-9]{4}\\b", yr\_ff[j])) next  
 grp <- norm\_group(hdr\_group[j])  
 if (is.na(grp)) next  
  
 vals <- suppressWarnings(as.numeric(raw[-c(1,2), j]))  
 parts[[length(parts) + 1]] <- data.frame(  
 Area = area\_vec,  
 Year = as.integer(yr),  
 SpeciesGroup = grp,  
 Count = vals,  
 stringsAsFactors = FALSE  
 )  
}  
##Bind and clean: remove blanks/NA areas, keep only our 4 groups, drop "grand total" rows  
tidy <- dplyr::bind\_rows(parts)  
tidy <- subset(tidy, !is.na(Area) & Area != "" & !is.na(Year) & !is.na(Count))  
tidy <- subset(tidy, SpeciesGroup %in% c("Mammal","Shark","Turtle","Other"))  
tidy <- subset(tidy, !grepl("grand\\s\*total", Area, ignore.case = TRUE))  
  
print(dplyr::count(tidy, SpeciesGroup, sort = TRUE))

## SpeciesGroup n  
## 1 Shark 285  
## 2 Other 230  
## 3 Turtle 195  
## 4 Mammal 120

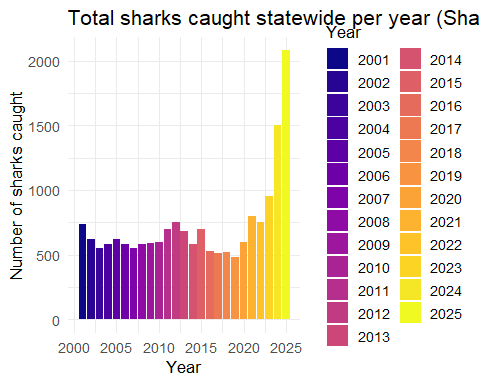
##——-PLOT——-## #Statewide trends for all four species groups

p\_species\_trend <- tidy %>%  
 group\_by(Year, SpeciesGroup) %>%  
 summarise(Total = sum(Count, na.rm = TRUE), .groups = "drop") %>%  
 ggplot(aes(x = Year, y = Total, color = SpeciesGroup)) +  
 geom\_line(linewidth = 0.8) +  
 geom\_point(size = 2) +  
 labs(  
 title = "Total catches by species group across Queensland",  
 x = "Year", y = "Number caught", color = "Group"  
 ) +  
 theme\_minimal(base\_size = 13)  
  
p\_species\_trend



## Statewide sharks by year (bar chart)

p\_sharks\_statewide <- tidy |>  
 dplyr::filter(SpeciesGroup == "Shark") |>  
 dplyr::group\_by(Year) |>  
 dplyr::summarise(Total = sum(Count, na.rm = TRUE), .groups = "drop") |>  
 ggplot(aes(x = Year, y = Total, fill = factor(Year))) +  
 geom\_col() +  
 scale\_fill\_viridis\_d(option = "plasma") +   
 labs(  
 title = "Total sharks caught statewide per year (Shark Control Program)",  
 x = "Year", y = "Number of sharks caught",  
 fill = "Year"  
 ) +  
 theme\_minimal(base\_size = 13)  
  
p\_sharks\_statewide



##————-Map————-##

library(sf)

lga <- st\_read("C:/Users/sumia/OneDrive/Documents/MB5370/GitHub/MB5370\_Mod04\_Personal/data/1259030001\_lga11aaust\_shape/lga11aaust.shp")

## Reading layer `LGA11aAust' from data source   
## `C:\Users\sumia\OneDrive\Documents\MB5370\GitHub\MB5370\_Mod04\_Personal\data\1259030001\_lga11aaust\_shape\LGA11aAust.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 565 features and 3 fields  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: 96.81694 ymin: -43.74051 xmax: 159.1092 ymax: -9.142176  
## Geodetic CRS: GDA94

names(lga)

## [1] "STATE\_CODE" "LGA\_CODE11" "LGA\_NAME11" "geometry"

head(lga, 3)

## Simple feature collection with 3 features and 3 fields  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: 146.8145 ymin: -36.11599 xmax: 152.4281 ymax: -30.23145  
## Geodetic CRS: GDA94  
## STATE\_CODE LGA\_CODE11 LGA\_NAME11 geometry  
## 1 1 10050 Albury (C) MULTIPOLYGON (((146.9138 -3...  
## 2 1 10110 Armidale Dumaresq (A) MULTIPOLYGON (((151.5358 -3...  
## 3 1 10150 Ashfield (A) MULTIPOLYGON (((151.1212 -3...

library(ggrepel)  
library(scales)

# --- Build the mapping data BEFORE plotting ---  
  
# 1) Keep only Queensland LGAs + name cleaner  
clean\_area <- function(x){  
 x %>%  
 stringr::str\_to\_lower() %>%  
 stringr::str\_replace\_all("&"," and ") %>%  
 stringr::str\_replace\_all("\\(.\*?\\)"," ") %>%  
 stringr::str\_replace\_all("regional|city|shire|council|lga"," ") %>%  
 stringr::str\_replace\_all("[^a-z0-9]+"," ") %>%  
 stringr::str\_squish()  
}  
  
qld\_lga <- lga %>%  
 dplyr::filter(STATE\_CODE == 3) %>%  
 dplyr::select(LGA\_NAME11, geometry) %>%  
 dplyr::mutate(LGA\_clean = clean\_area(LGA\_NAME11))  
  
# 2) Shark totals across ALL years (use filter(Year == XXXX) if you need a single year)  
shark\_totals <- tidy %>%  
 dplyr::filter(SpeciesGroup == "Shark") %>%  
 dplyr::group\_by(Area) %>%  
 dplyr::summarise(TotalSharks = sum(Count, na.rm = TRUE), .groups = "drop") %>%  
 dplyr::mutate(Area\_clean = clean\_area(Area))  
  
# 3) Join polygons + totals (avoid the name 'map\_data' to not clash with ggplot2)  
shp\_join <- qld\_lga %>%  
 dplyr::left\_join(shark\_totals, by = c("LGA\_clean" = "Area\_clean"))  
  
# 4) Quick checks (optional, but helpful)  
print(names(shp\_join))

## [1] "LGA\_NAME11" "LGA\_clean" "Area" "TotalSharks" "geometry"

print(head(shp\_join %>% sf::st\_drop\_geometry() %>% dplyr::select(LGA\_NAME11, TotalSharks)))

## LGA\_NAME11 TotalSharks  
## 1 Aurukun (S) NA  
## 2 Balonne (S) NA  
## 3 Banana (S) NA  
## 4 Barcaldine (R) NA  
## 5 Barcoo (S) NA  
## 6 Blackall Tambo (R) NA

non\_na <- shp\_join %>% dplyr::filter(!is.na(TotalSharks))  
bb <- sf::st\_bbox(qld\_lga)  
  
# ... later in ggplot:  
geom\_sf(data = non\_na, aes(fill = TotalSharks), colour = "white", linewidth = 0.25)

## [[1]]  
## mapping: fill = ~TotalSharks   
## geom\_sf: na.rm = FALSE  
## stat\_sf: na.rm = FALSE  
## position\_identity   
##   
## [[2]]  
## <ggproto object: Class CoordSf, CoordCartesian, Coord, gg>  
## aspect: function  
## backtransform\_range: function  
## clip: on  
## crs: NULL  
## datum: crs  
## default: TRUE  
## default\_crs: NULL  
## determine\_crs: function  
## distance: function  
## expand: TRUE  
## fixup\_graticule\_labels: function  
## get\_default\_crs: function  
## is\_free: function  
## is\_linear: function  
## label\_axes: list  
## label\_graticule:   
## labels: function  
## limits: list  
## lims\_method: cross  
## modify\_scales: function  
## ndiscr: 100  
## params: list  
## range: function  
## record\_bbox: function  
## render\_axis\_h: function  
## render\_axis\_v: function  
## render\_bg: function  
## render\_fg: function  
## setup\_data: function  
## setup\_layout: function  
## setup\_panel\_guides: function  
## setup\_panel\_params: function  
## setup\_params: function  
## train\_panel\_guides: function  
## transform: function  
## super: <ggproto object: Class CoordSf, CoordCartesian, Coord, gg>

anti\_join(shark\_totals, qld\_lga %>% dplyr::select(LGA\_clean),   
 by = c("Area\_clean" = "LGA\_clean")) %>% dplyr::distinct(Area)

## # A tibble: 7 × 1  
## Area   
## <chr>   
## 1 Bribie Island   
## 2 Capricorn Coast   
## 3 Nth Stradbroke Is.   
## 4 Rainbow Beach   
## 5 Sunshine Coast North   
## 6 Sunshine Coast South   
## 7 Sunshine Coast South & Bribie Island

# --- empty overrides so knit never fails ---  
  
if (!exists("overrides")) {  
 overrides <- tibble::tibble(Area = character(), LGA\_clean = character())  
}

shark\_totals <- shark\_totals %>%  
 dplyr::left\_join(overrides, by = "Area") %>% # safe now  
 dplyr::mutate(Area\_clean = dplyr::coalesce(LGA\_clean, Area\_clean)) %>%  
 dplyr::select(-LGA\_clean)

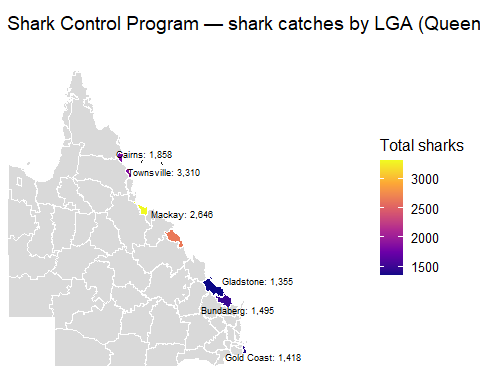
shark\_totals <- shark\_totals %>%  
 dplyr::left\_join(overrides, by = "Area") %>%  
 dplyr::mutate(Area\_clean = dplyr::coalesce(LGA\_clean, Area\_clean)) %>%  
 dplyr::select(-LGA\_clean)  
  
shp\_join <- qld\_lga %>% dplyr::left\_join(shark\_totals, by = c("LGA\_clean" = "Area\_clean"))  
  
# sanity check (should show some non-NA numbers)  
shp\_join %>% sf::st\_drop\_geometry() %>% dplyr::count(is.na(TotalSharks))

## is.na(TotalSharks) n  
## 1 FALSE 6  
## 2 TRUE 68

head(shp\_join %>% sf::st\_drop\_geometry() %>% dplyr::select(LGA\_NAME11, TotalSharks))

## LGA\_NAME11 TotalSharks  
## 1 Aurukun (S) NA  
## 2 Balonne (S) NA  
## 3 Banana (S) NA  
## 4 Barcaldine (R) NA  
## 5 Barcoo (S) NA  
## 6 Blackall Tambo (R) NA

non\_na <- shp\_join %>% dplyr::filter(!is.na(TotalSharks))  
# --- label data (top N) ---  
TOP\_N <- 8  
labs0 <- non\_na %>%  
 dplyr::arrange(dplyr::desc(TotalSharks)) %>%  
 dplyr::slice\_head(n = TOP\_N) %>%  
 sf::st\_point\_on\_surface() %>%  
 dplyr::mutate(  
 X = sf::st\_coordinates(geometry)[,1],  
 Y = sf::st\_coordinates(geometry)[,2],  
 label = paste0(stringr::str\_replace(LGA\_NAME11, "\\s\*\\(.\*\\)", ""), ": ",  
 scales::comma(TotalSharks))  
 ) %>% sf::st\_drop\_geometry()  
  
# --- tiny manual nudges for the crowded coastal councils ---  
# dx, dy are in degrees (small numbers).   
offsets <- tibble::tribble(  
 ~LGA\_NAME11, ~dx, ~dy,  
 "Cairns (R)", 0.85, 0.00,  
 "Townsville (C)", 1.00, 3.00,  
 "Mackay (R)", 0.18, 2.00,  
 "Rockhampton (R)", 0.85, 1.50,  
 "Gladstone (R)", 2.50, -0.10,  
 "Bundaberg (R)", 0.55, -1.10,  
 "Moreton Bay (R)", 2.00, 0.50,  
 "Gold Coast (C)", 0.80, -1  
)  
  
labs <- labs0 %>%  
 dplyr::left\_join(offsets, by = "LGA\_NAME11") %>%  
 dplyr::mutate(  
 dx = dplyr::coalesce(dx, 0),  
 dy = dplyr::coalesce(dy, 0),  
 Xlab = X + dx,  
 Ylab = Y + dy  
 )  
  
# --- more horizontal space for off-coast labels ---  
bb <- sf::st\_bbox(qld\_lga)  
xpad <- (bb["xmax"] - bb["xmin"]) \* 0.45   
  
p\_map <- ggplot() +  
 geom\_sf(data = qld\_lga, fill = "grey85", colour = "white", linewidth = 0.3) +  
 geom\_sf(data = non\_na, aes(fill = TotalSharks), colour = "white", linewidth = 0.25) +  
 scale\_fill\_viridis\_c(option = "plasma", na.value = "grey90") +  
 ggrepel::geom\_text\_repel(  
 data = labs, aes(Xlab, Ylab, label = label),  
 size = 2.5, colour = "black", seed = 123,  
 direction = "y",   
 nudge\_x = 0.4,   
 box.padding = 0.0, point.padding = 0.3,  
 force = 6, force\_pull = 0.6,  
 min.segment.length = 0, segment.size = 0.3, segment.alpha = 0.9  
 ) +  
 coord\_sf(  
 xlim = c(bb["xmin"], bb["xmax"] + xpad),  
 ylim = c(bb["ymin"], bb["ymax"]),  
 expand = FALSE, datum = NA, clip = "off"  
 ) +  
 labs(title = "Shark Control Program — shark catches by LGA (Queensland)",  
 fill = "Total sharks") +  
 theme\_minimal(base\_size = 12) +  
 theme(axis.text = element\_blank(), axis.ticks = element\_blank(), axis.title = element\_blank())  
  
p\_map



shark\_totals <- shark\_totals %>%  
 dplyr::left\_join(overrides, by = "Area") %>%  
 dplyr::mutate(Area\_clean = dplyr::coalesce(LGA\_clean, Area\_clean)) %>%  
 dplyr::select(-LGA\_clean)

# --- use only LGAs that actually have totals ---  
non\_na <- shp\_join %>% dplyr::filter(!is.na(TotalSharks))  
  
# --- which LGAs MUST be labelled even if crowded? ---  
MUST\_HAVE <- c("Gold Coast (C)", "Moreton Bay (R)", "Bundaberg (R)") # add any you want guaranteed  
  
TOP\_N <- 8  
  
# base top-N + force-add MUST\_HAVE  
labs0 <- non\_na %>%  
 dplyr::arrange(dplyr::desc(TotalSharks)) %>%  
 dplyr::slice\_head(n = TOP\_N)  
  
labs\_force <- non\_na %>% dplyr::filter(LGA\_NAME11 %in% MUST\_HAVE)  
  
labs\_raw <- dplyr::bind\_rows(labs0, labs\_force) %>%  
 dplyr::distinct(LGA\_NAME11, .keep\_all = TRUE) %>%  
 sf::st\_point\_on\_surface() %>%  
 dplyr::mutate(  
 X = sf::st\_coordinates(geometry)[,1],  
 Y = sf::st\_coordinates(geometry)[,2],  
 label = paste0(stringr::str\_replace(LGA\_NAME11, "\\s\*\\(.\*\\)", ""), ": ",  
 scales::comma(TotalSharks))  
 ) %>% sf::st\_drop\_geometry()  
  
# your manual nudges (keep/tweak small values so labels stay in frame)  
offsets <- tibble::tribble(  
 ~LGA\_NAME11, ~dx, ~dy,  
 "Cairns (R)", 0.85, 0.00,  
 "Townsville (C)", 0.80, 0.40, # was 3.00 (too large, could go off canvas)  
 "Mackay (R)", 0.18, 0.60, # was 2.00  
 "Rockhampton (R)", 0.85, 0.50, # was 1.50  
 "Gladstone (R)", 1.10, -0.10, # was 2.50  
 "Bundaberg (R)", 0.55, -0.60, # was -1.10  
 "Moreton Bay (R)", 1.20, 0.30, # was 2.00  
 "Gold Coast (C)", 0.80, -0.60 # was -1  
)  
  
labs <- labs\_raw %>%  
 dplyr::left\_join(offsets, by = "LGA\_NAME11") %>%  
 dplyr::mutate(  
 dx = dplyr::coalesce(dx, 0),  
 dy = dplyr::coalesce(dy, 0),  
 Xlab = X + dx,  
 Ylab = Y + dy  
 )  
  
# --- expand the canvas on BOTH axes so labels never clip ---  
bb <- sf::st\_bbox(qld\_lga)  
xpad <- (bb["xmax"] - bb["xmin"]) \* 0.55 # more ocean space to the right  
ypad <- (bb["ymax"] - bb["ymin"]) \* 0.06 # a little extra top/bottom  
  
p\_map <- ggplot() +  
 geom\_sf(data = qld\_lga, fill = "grey85", colour = "white", linewidth = 0.3) +  
 geom\_sf(data = non\_na, aes(fill = TotalSharks), colour = "white", linewidth = 0.25) +  
 scale\_fill\_viridis\_c(option = "plasma", na.value = "grey90") +  
 ggrepel::geom\_text\_repel(  
 data = labs, aes(Xlab, Ylab, label = label),  
 size = 2.5, colour = "black", seed = 123,  
 direction = "y", nudge\_x = 0.35,  
 box.padding = 0.0, point.padding = 0.3,  
 force = 8, force\_pull = 0.7,  
 min.segment.length = 0, segment.size = 0.3, segment.alpha = 0.9,  
 max.overlaps = Inf # <- never drop a label  
 ) +  
 coord\_sf(  
 xlim = c(bb["xmin"], bb["xmax"] + xpad),  
 ylim = c(bb["ymin"] - ypad, bb["ymax"] + ypad),  
 expand = FALSE, datum = NA, clip = "off"  
 ) +  
 labs(title = "Shark Control Program — shark catches by LGA (Queensland)",  
 fill = "Total sharks") +  
 theme\_minimal(base\_size = 12) +  
 theme(axis.text = element\_blank(), axis.ticks = element\_blank(), axis.title = element\_blank())  
  
p\_map

