

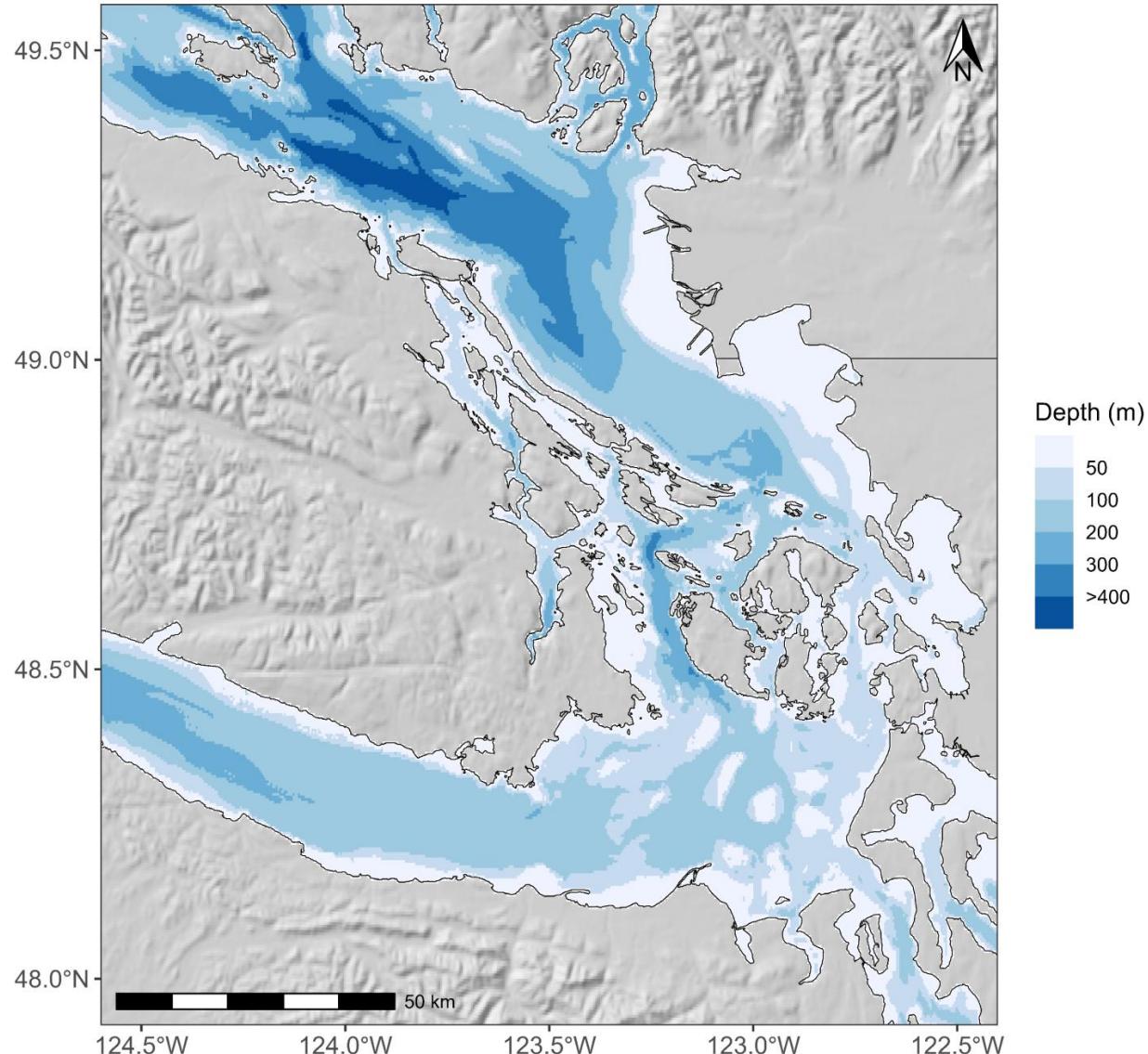
# Making maps in R: applications to ecology and evolution

Wesley Greentree

UVic Biology Undergraduate Society

March 18, 2025

wgreentree@outlook.com

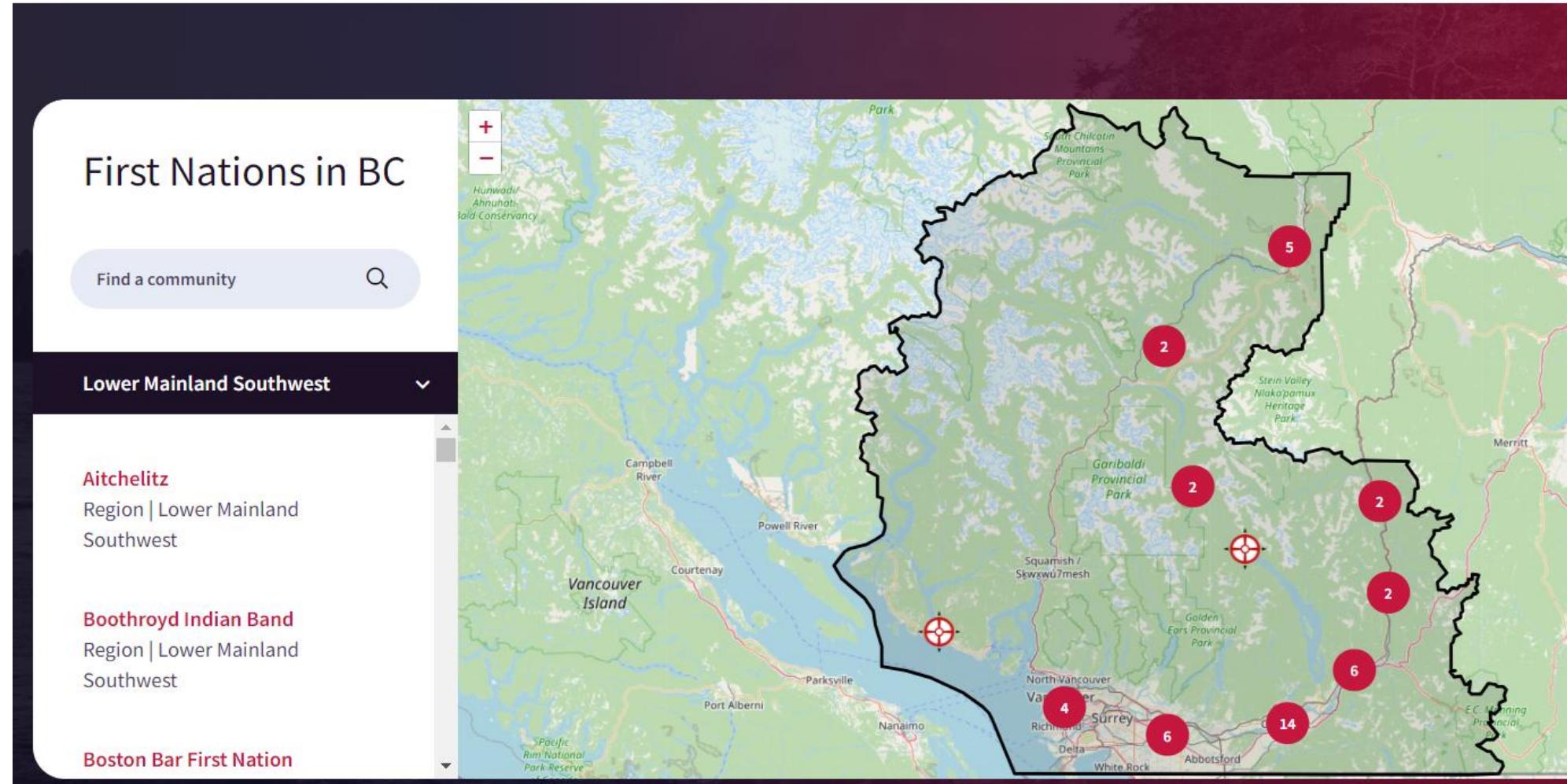


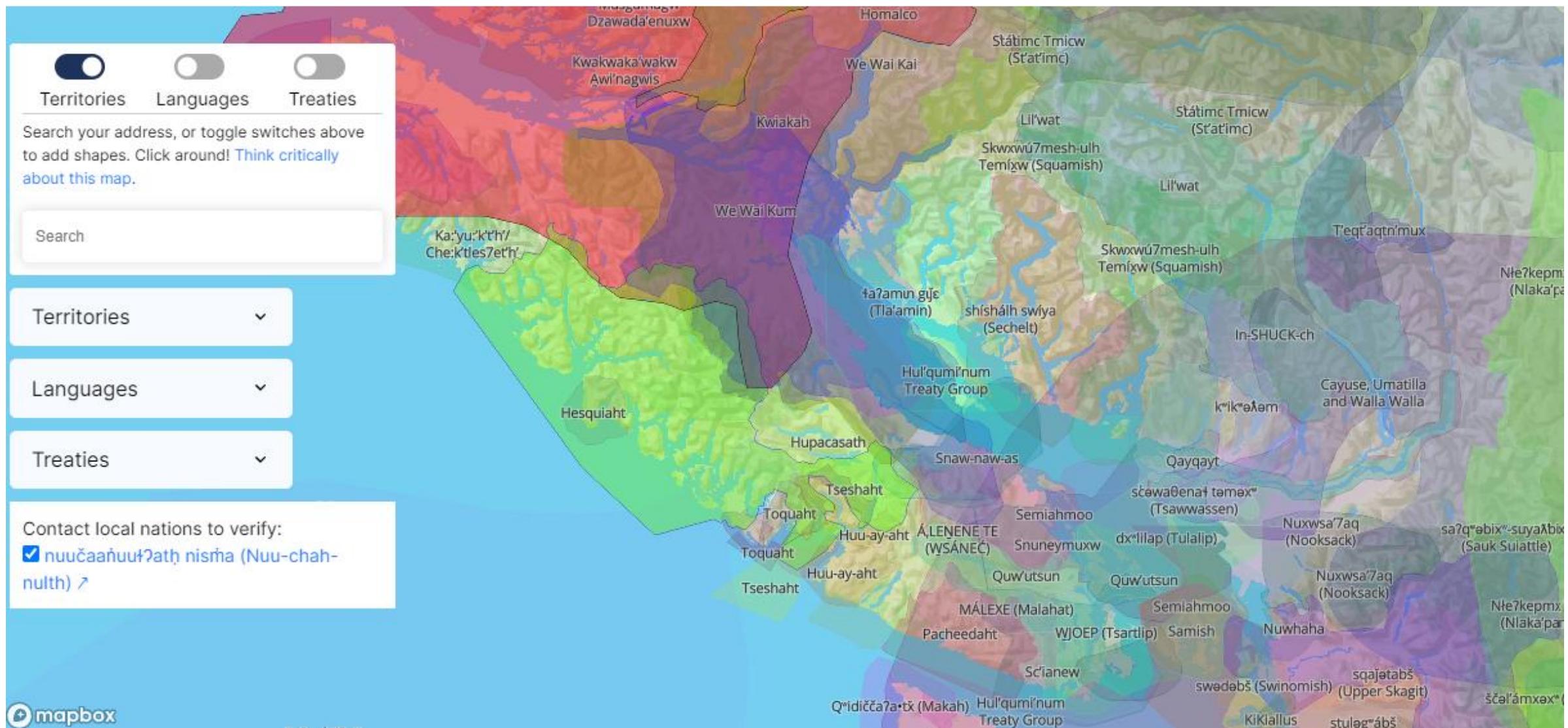
canadian institute of ecology and evolution  
institut canadien d'écologie et d'évolution

# Territory acknowledgements

- **UVic:** I acknowledge and respect the *lək'ʷəηən* peoples on whose traditional territory the university stands and the Songhees, Esquimalt and *WSÁNEĆ* peoples whose historical relationships with the land continue to this day.







<https://native-land.ca/>

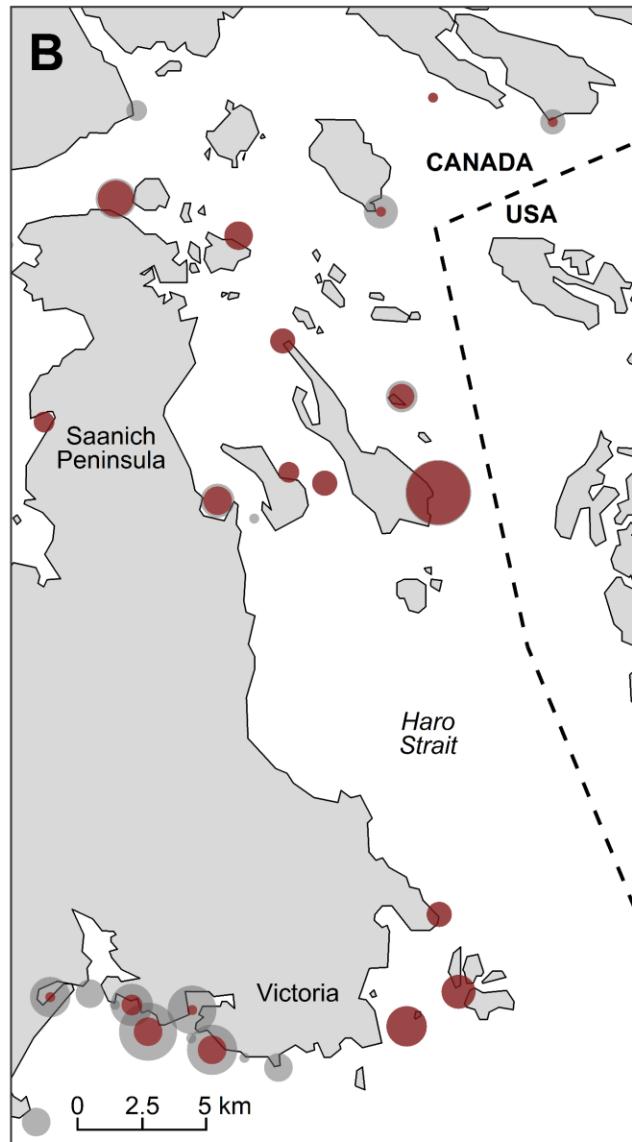
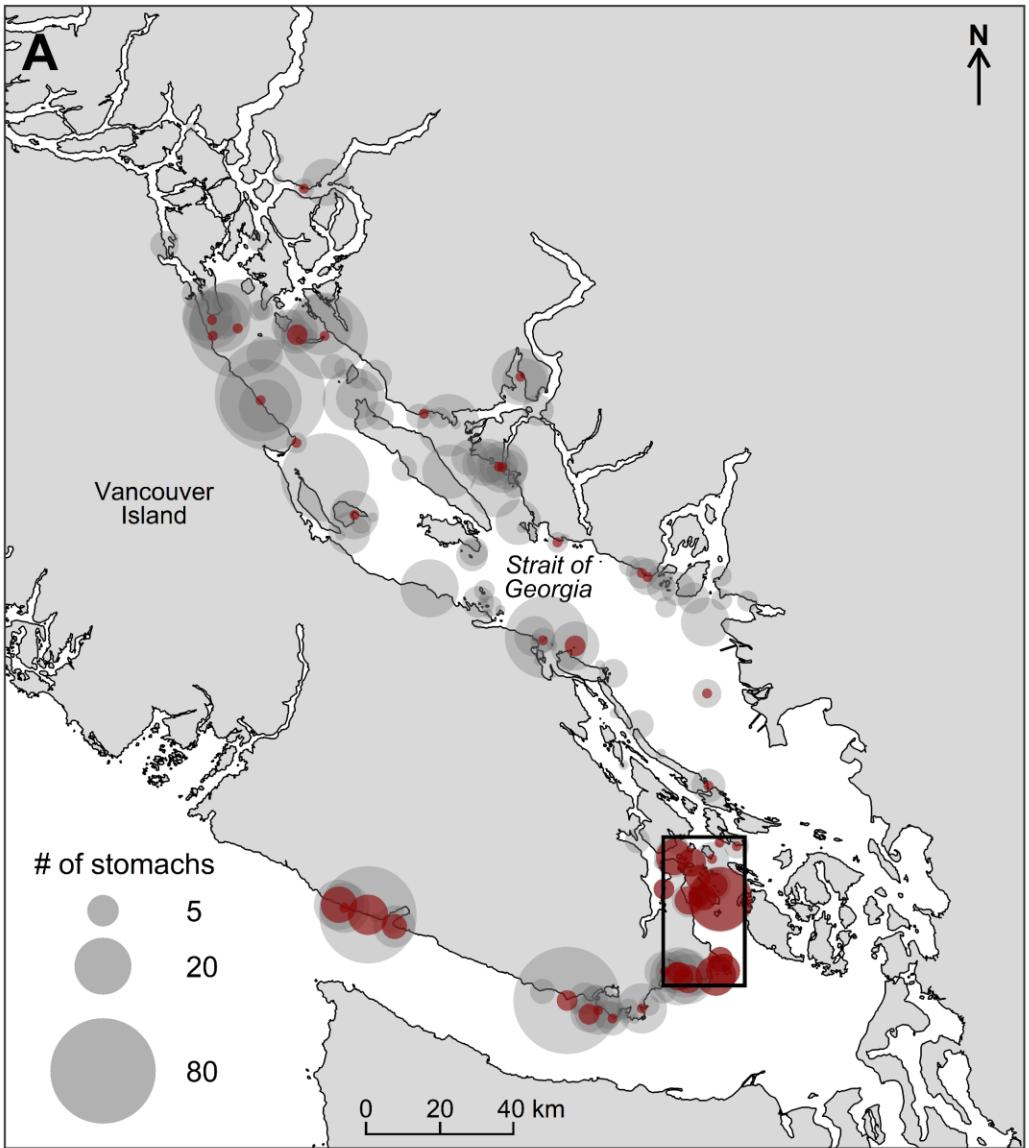


Wesley Greentree (he/him)  
PhD student, UVic





Wesley Greentree (he/him)  
PhD student, UVic



Regional food webs



Wesley Greentree (he/him)  
PhD student, UVic



DateTime: 2022-10-15



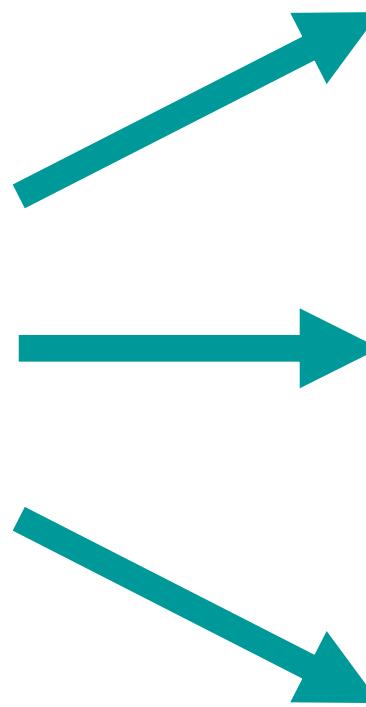
Salmon migrations

DateTime: 2022-10-15



Communicate your  
science through maps

Communicate your  
science through maps



All the relevant  
information, but  
not too much

Clean, beautiful  
maps

Easy for you!

# Workshop plan

Inclusive for beginner R users, with extra info for experienced coders

## Topics:

- Introduction to spatial data
- Introduction to ggplot()
- Simple study area maps
- Adding additional layers
- Advanced/fun topics

Please interrupt with questions (or raise hand)

Short breaks for discussion/practice

# Workshop plan

Download code and data from:

<https://github.com/wesleygreentree/UVicBUGS-R-maps>

The screenshot shows a GitHub repository page for 'UVicBUGS-R-maps'. The repository is public and has 1 branch and 0 tags. The main branch is 'main'. The repository was last updated 2 hours ago by 'wesleygreentree' with the commit message 'update workshop'. The repository contains several files and folders: 'data-raw', 'data', 'figures', 'scripts', '.gitignore', 'README.md', and 'UVicBUGS-R-maps.Rproj'. A modal window is open over the repository content, showing cloning options. The 'Clone' section includes 'Local' and 'Codespaces' tabs, and three cloning methods: 'HTTPS' (selected), 'SSH', and 'GitHub CLI'. The HTTPS URL is displayed as <https://github.com/wesleygreentree/UVicBUGS-R-maps>. Below the URL, there are links to 'Open with GitHub Desktop' and 'Download ZIP'. On the right side of the page, there is an 'About' section with a description: 'Workshop on making maps with R for UVic Biology Undergraduate Society'. It also shows statistics: 0 stars, 1 watching, and 0 forks. There is also a 'Releases' section indicating 'No releases published'.

UVicBUGS-R-maps Public

main 1 Branch 0 Tags

wesleygreentree update workshop

data-raw raw GEBCO data wa

data raw GEBCO data wa

figures update workshop

scripts update workshop

.gitignore upload workshop

README.md read me

UVicBUGS-R-maps.Rproj update workshop [No Title] 2 hours ago

Unpin Unwatch 1 Fork 0 Star 0

Go to file Add file Code Local Codespaces

Clone HTTPS SSH GitHub CLI

<https://github.com/wesleygreentree/UVicBUGS-R-maps>

Readme Activity 0 stars 1 watching 0 forks

About

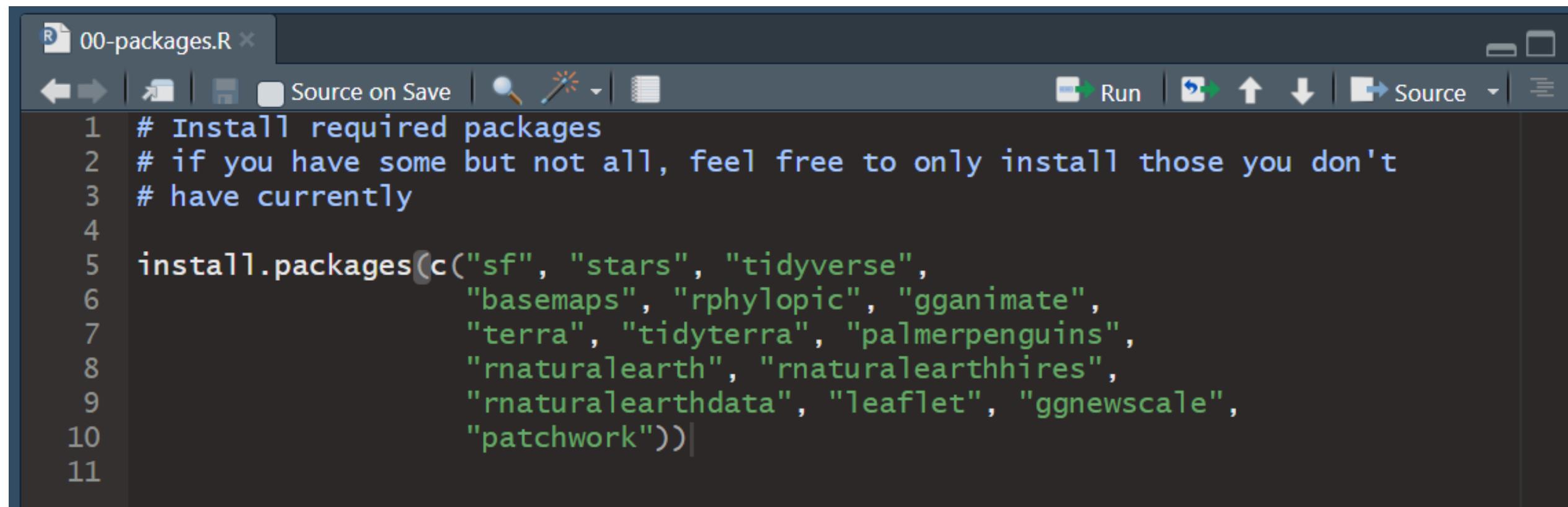
Workshop on making maps with R for UVic Biology Undergraduate Society

Releases

No releases published Create a new release

# Run scripts/00-packages.R

Installs the necessary packages for the workshop

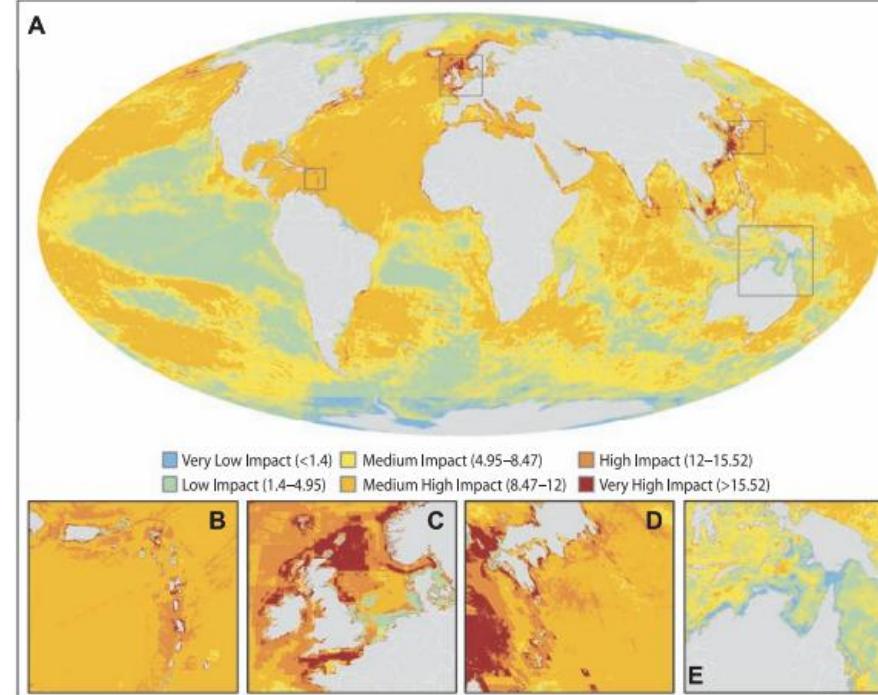


The screenshot shows the RStudio interface with the file '00-packages.R' open. The code in the editor is as follows:

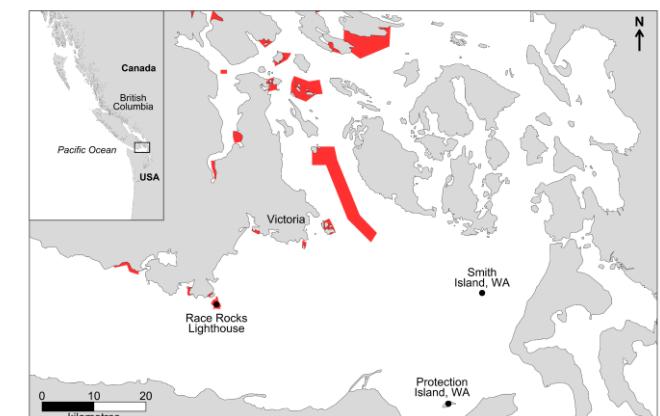
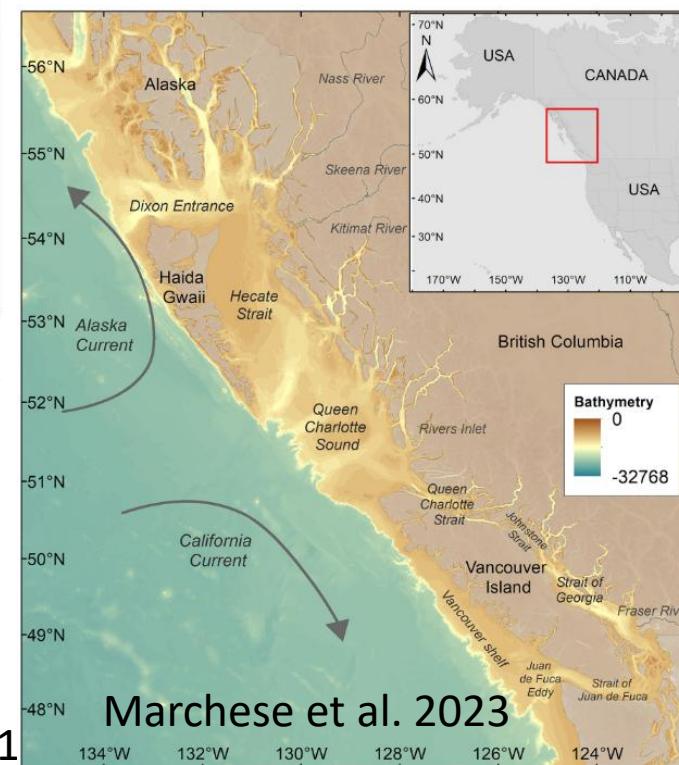
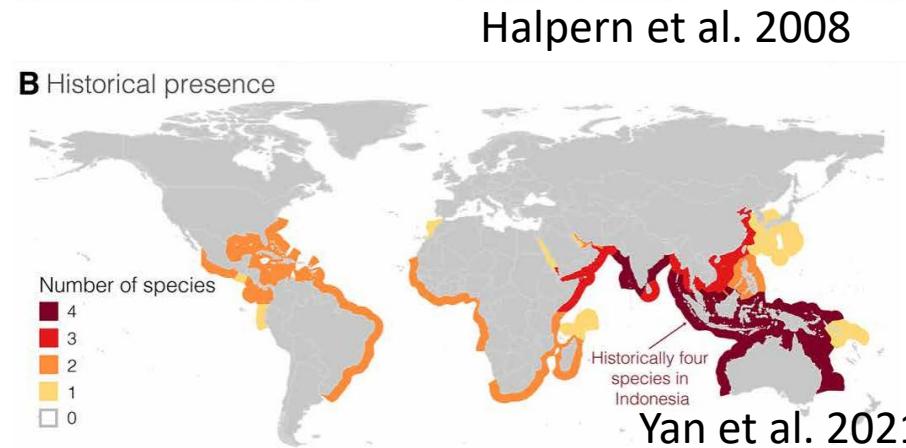
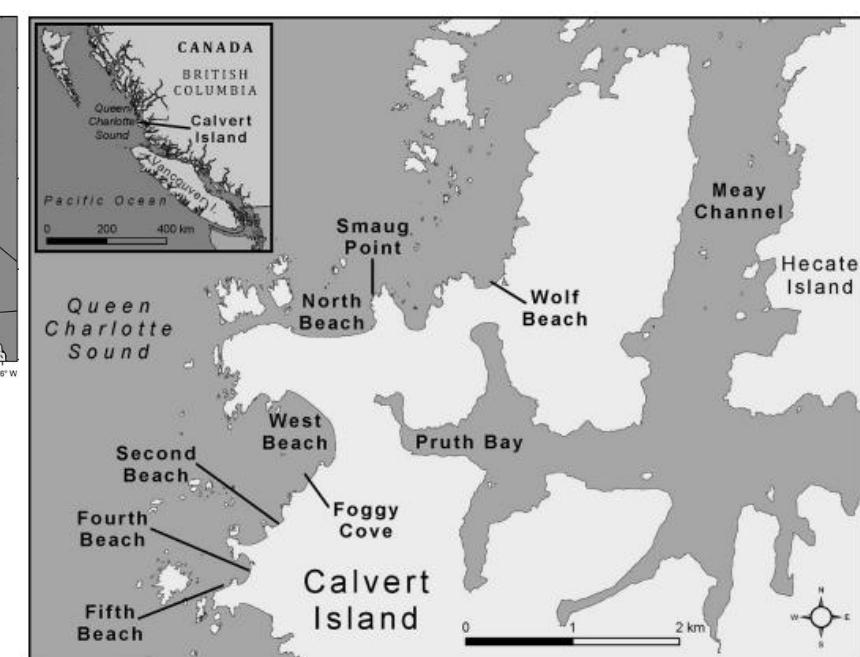
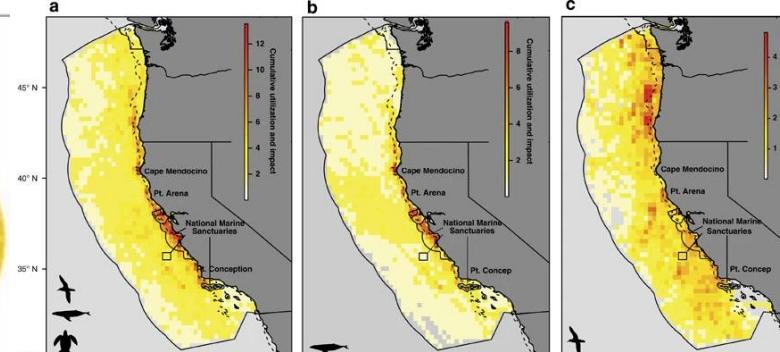
```
1 # Install required packages
2 # if you have some but not all, feel free to only install those you don't
3 # have currently
4
5 install.packages(c("sf", "stars", "tidyverse",
6                     "basemaps", "rphylopic", "ganimate",
7                     "terra", "tidyterra", "palmerpenguins",
8                     "rnaturalearth", "rnaturalearthhires",
9                     "rnaturalearthdata", "leaflet", "ggnewscale",
10                    "patchwork"))
```

# Spatial data are common in ecology

## Global



## Regional



# Types of spatial data

## Vectors

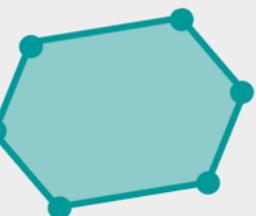
coordinates

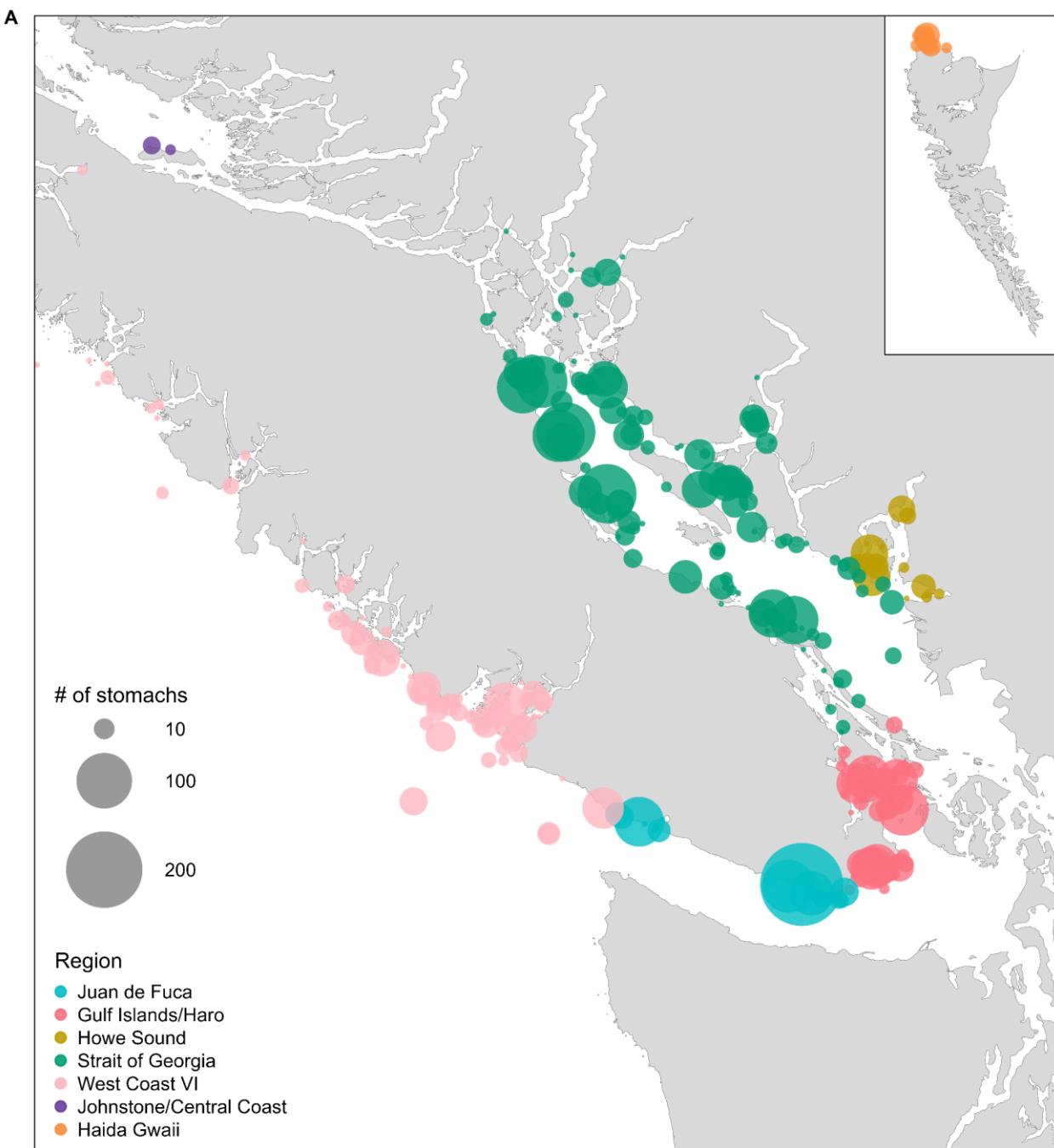
(can be joined together to make  
different vector types)

Points

Lines

Polygons





# Types of spatial data

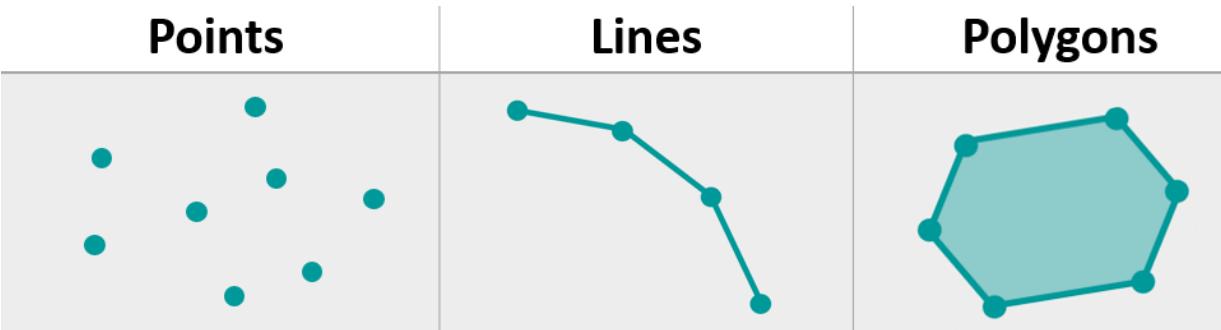
## Vectors

coordinates  
(can be joined together to make different vector types)

Points

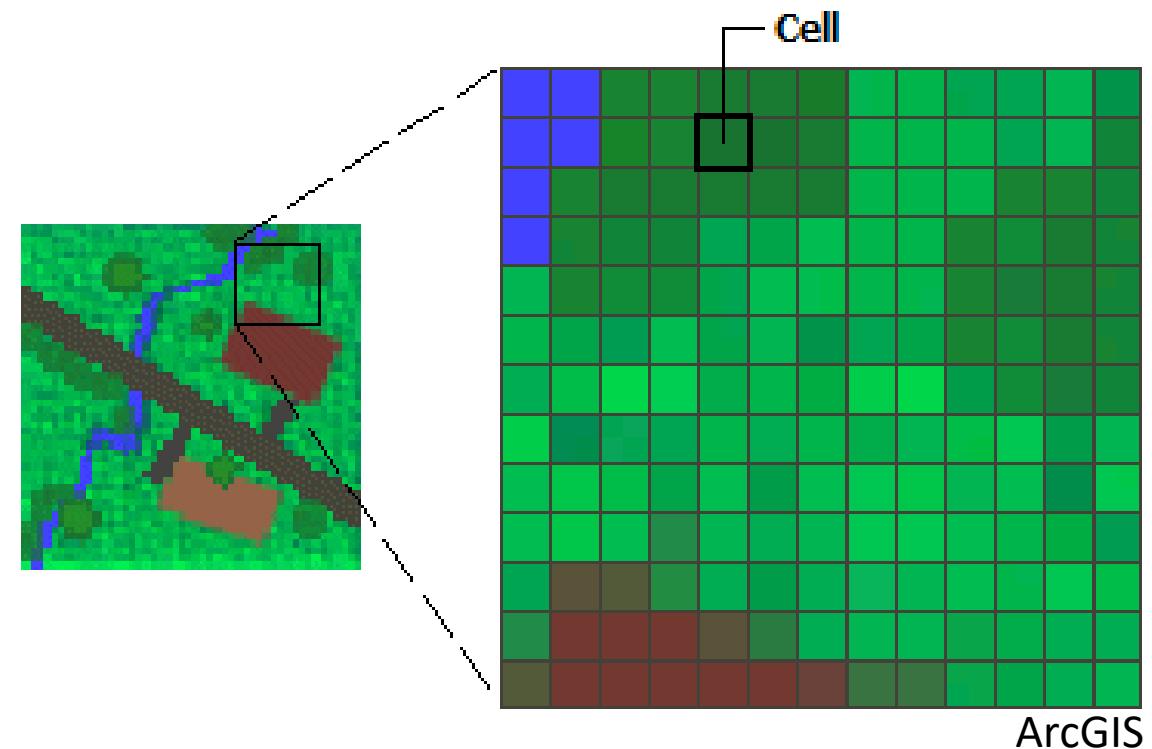
Lines

Polygons



## Rasters

pixels

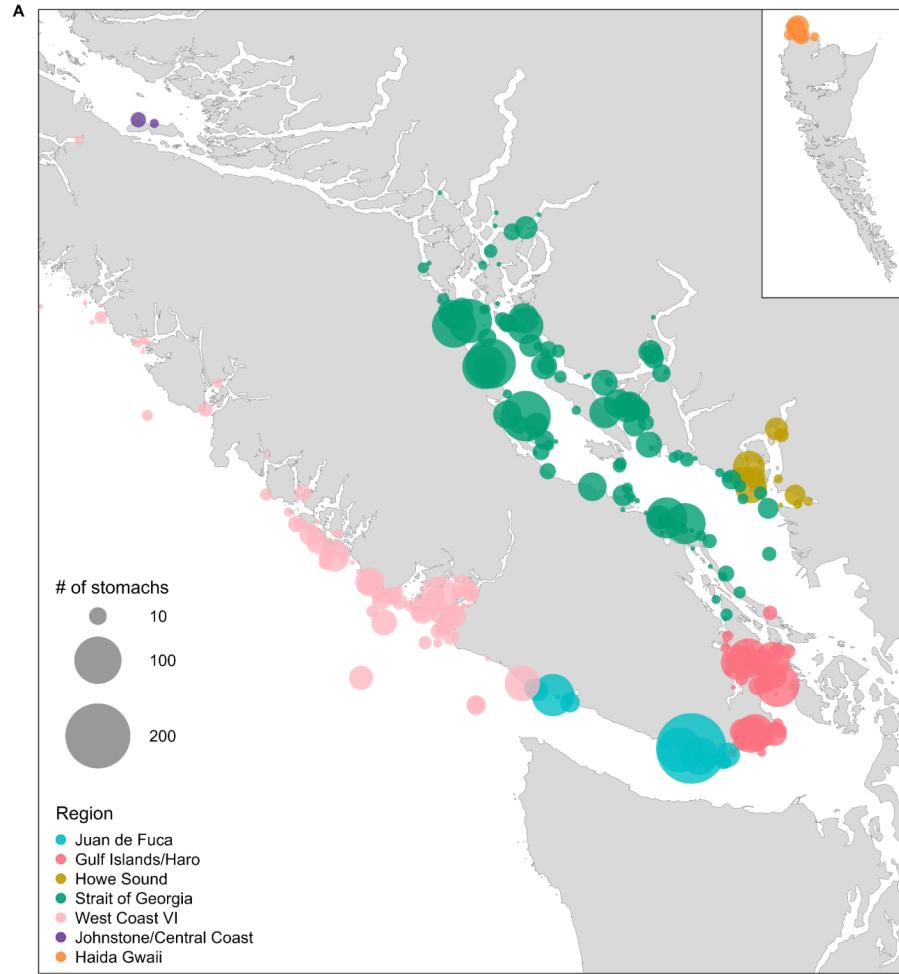




# Types of spatial data

## Vectors

polygons, points, lines

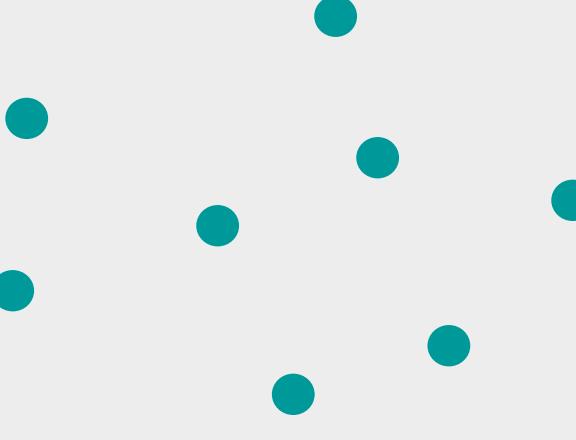
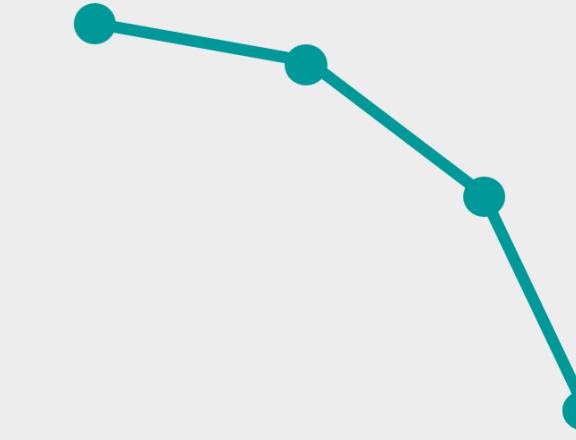
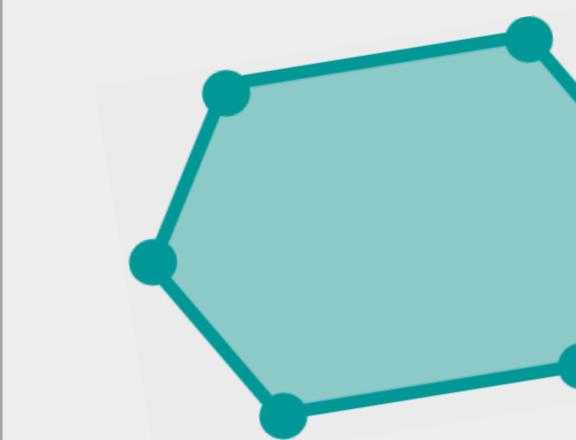


## Rasters

pixels  
e.g., satellite images

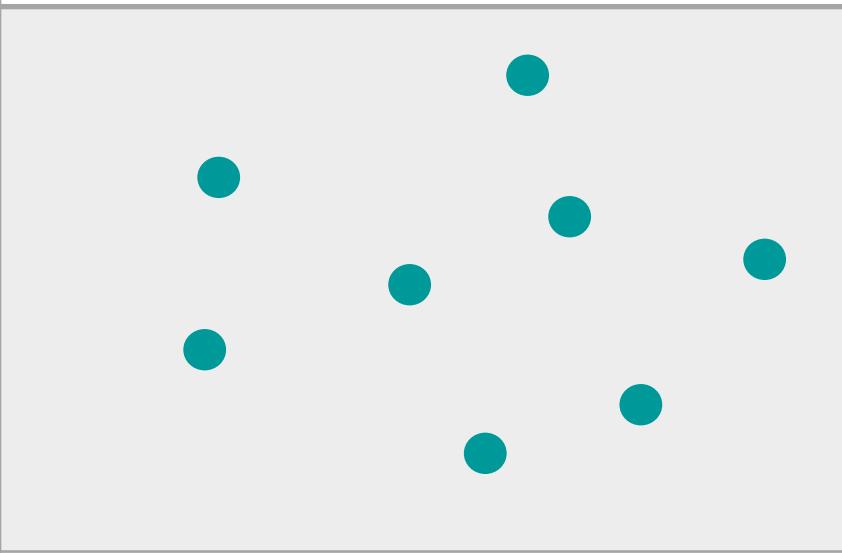


# Types of vector data

Points	Lines	Polygons
		
<p>individual x, y positions</p> <p><b>examples:</b></p> <ul style="list-style-type: none"><li>sampling locations</li><li>species presence</li></ul>	<p>lines connect 2+ points</p> <ul style="list-style-type: none"><li>migration paths</li><li>roads</li></ul>	<p>3+ points that connect and close</p> <ul style="list-style-type: none"><li>coastlines</li><li>protected areas</li></ul>

# Types of vector data

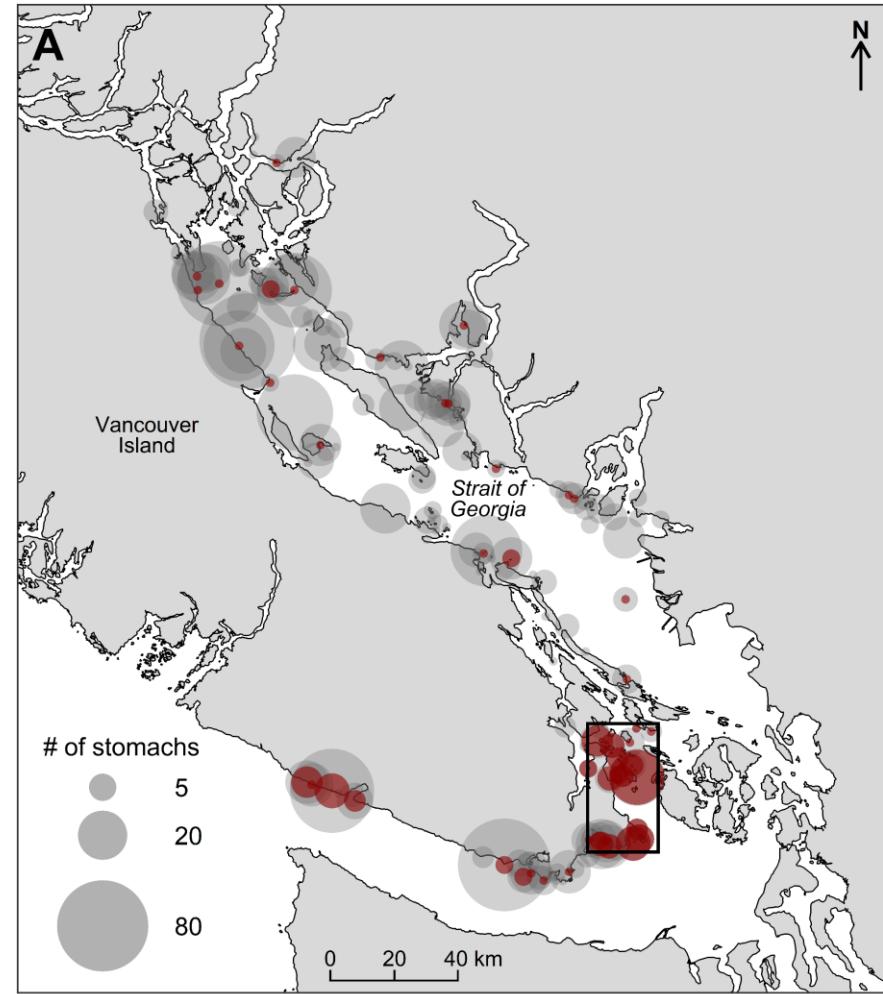
## Points



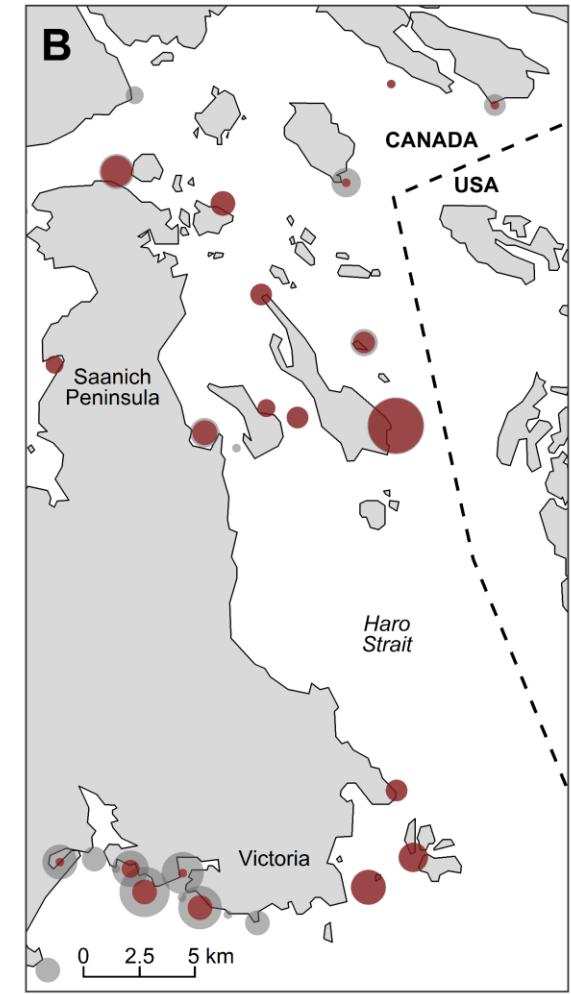
individual x, y positions

### examples:

sampling locations  
species presence



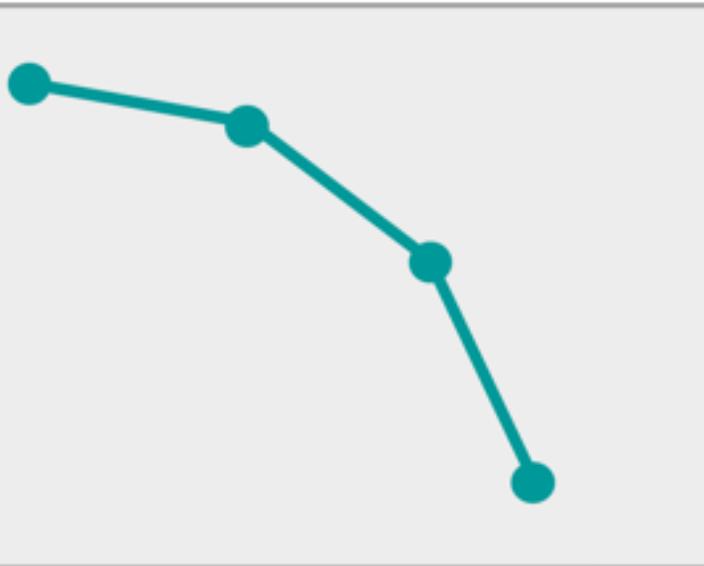
Point size corresponds to number of salmon stomachs containing sand lance



Robinson et al. 2023

# Types of vector data

## Lines



lines connect 2+ points

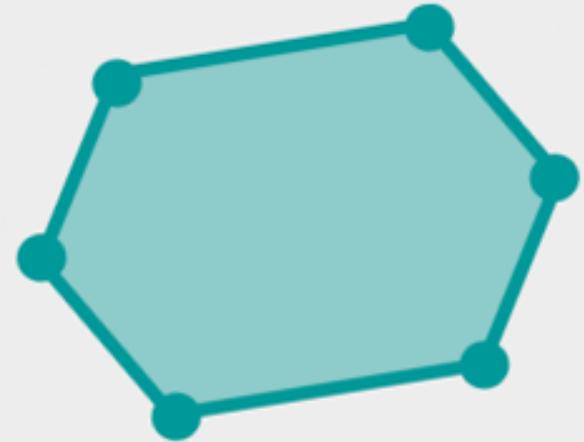
migration paths  
roads



Humpback whale migration tracks  
along South America

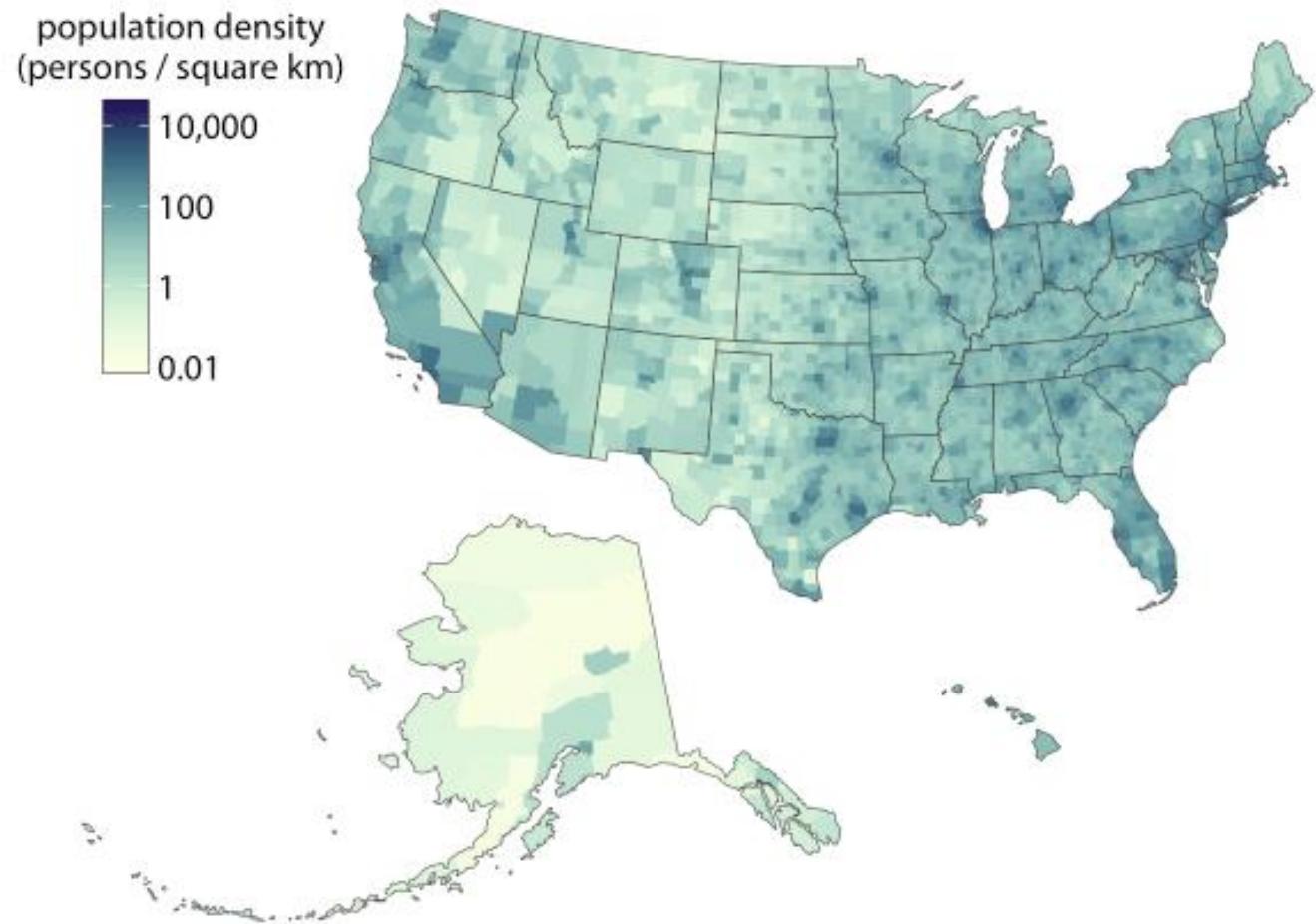
# Types of vector data

## Polygons

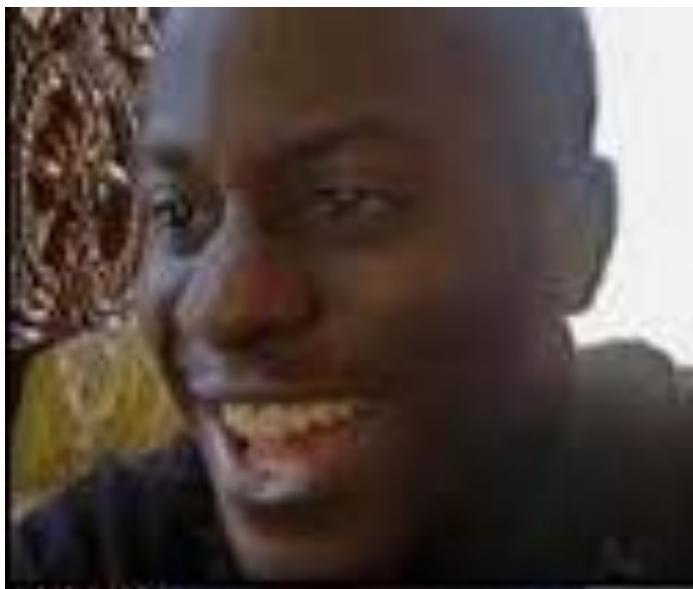


3+ points that connect  
and close

coastlines  
protected areas



# Why use R for maps?



Decides to learn ArcGIS



**ArcGIS**

---

Pay walls

Another complicated thing to learn!

Lack of training

Reproducibility crisis

Difficult to integrate with R workflows



# Why use R for maps?

Challenges with GIS software:

- Most ecologists lack GIS training
- Outside of R data/stats workflow

Making maps in R:

- **Fits in data processing/analysis scripts**
- Reproducibility (workflow saved in script)
- Nice basemaps are built into R packages
- Faceted maps (many maps in one plot)



# Example data analysis workflow

**Load data**  
`read.csv()`  
`read_sf`



**Data wrangling**  
`subset()`  
`merge()`  
`mutate()`



**Geoprocessing**  
`st_transform()`



All data processing,  
analysis, and  
visualization steps in  
one R script!

**Data visualization**  
`ggplot() +`  
`geom_sf() +`  
`coord_sf()`

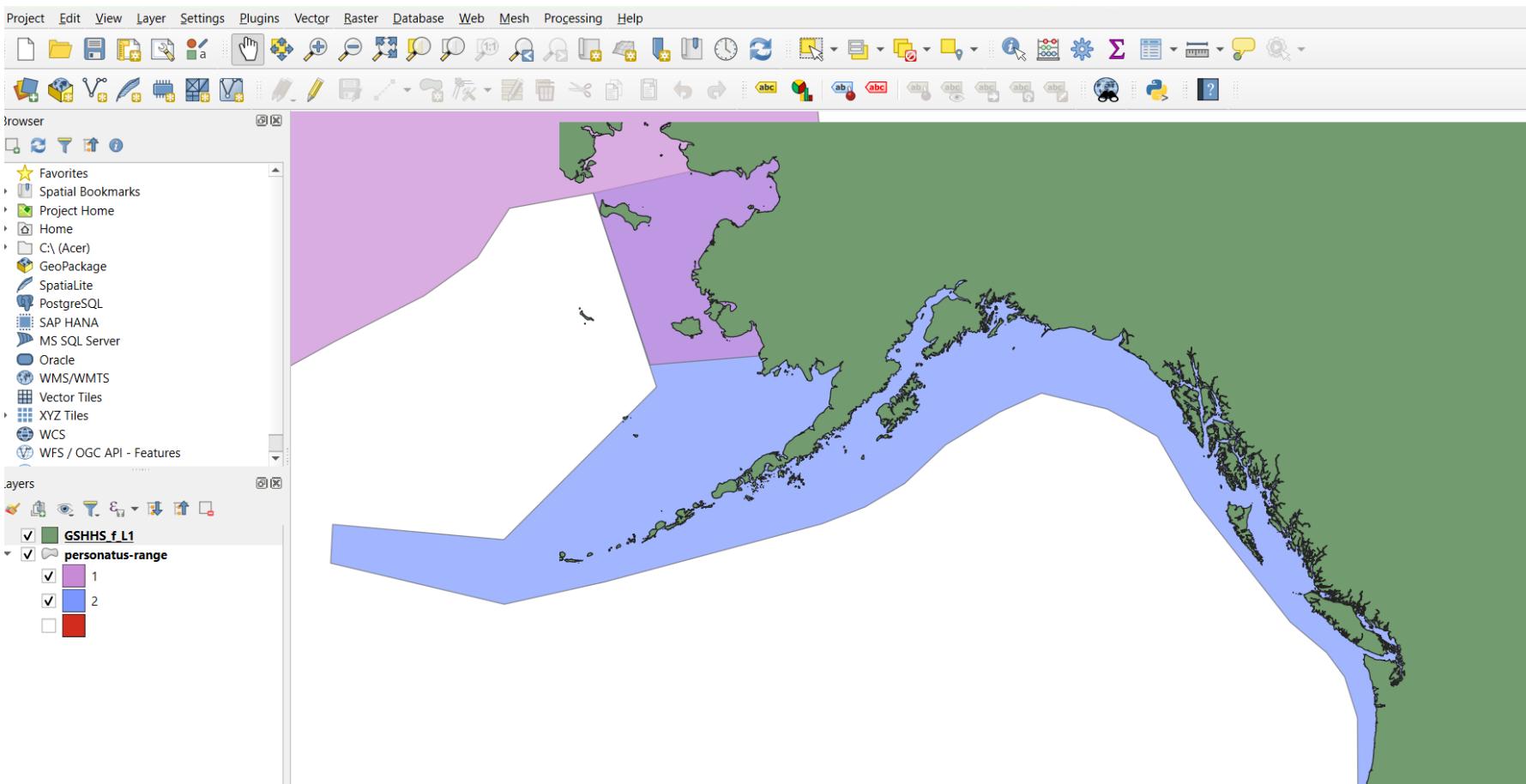


**Data analysis**  
`glm()`  
`nmds()`  
`adonis()`

# When do I use GIS software?



- Hand-drawing new polygons (e.g., species ranges)
- Producing files for a GPS device



# Functions in R

- R uses functions
  - plot() makes graphs
- Some functions are automatically loaded in R (“**base**”)
- R software developers build **packages**, which include their functions
  - base::plot()
    - ↑ package
    - ↑ function

# Functions in R

- R uses functions
  - plot() makes graphs
- Some functions are automatically loaded in R (**“base”**)
- R software developers build **packages**, which include their functions
  - base::plot()
  - ggplot2::ggplot()

Both plot() and ggplot() make plots

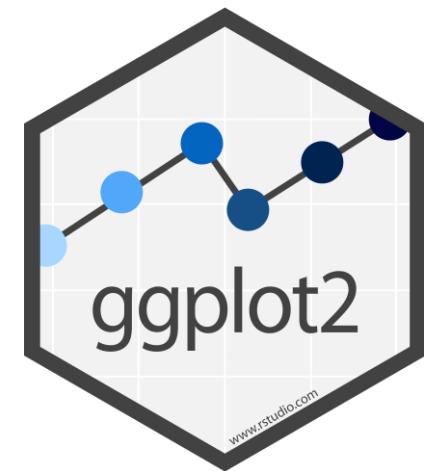
ggplot() makes plotting easier

# Functions in R

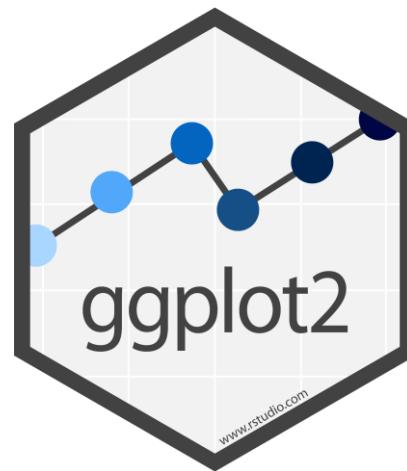
- R uses functions
  - plot() makes graphs
- Some functions are automatically loaded in R (**“base”**)
- R software developers build **packages**, which include their functions

```
install.packages("ggplot2") # first time installing package  
  
library(ggplot2) # load package that is already installed  
  
ggplot2::ggplot() # use a function without loading entire package
```

# ggplot() “grammar of graphics” in R



# ggplot() “grammar of graphics” in R



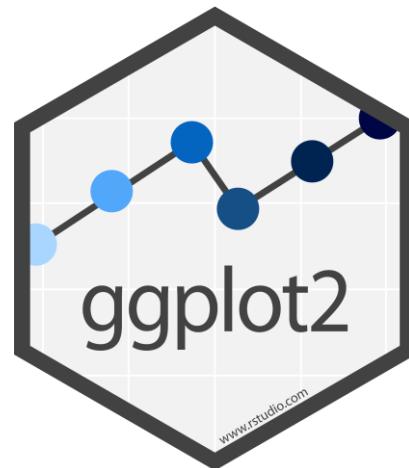
Create figures with multiple layers

Different types of “**geoms**” to make different plots:

- `geom_point()` scatterplots, spatial points
- `geom_line()`, `geom_path()` line graphs, spatial lines
- `geom_sf()` shapefiles (spatial polygons)
- `geom_label()` print text labels on figure

```
ggplot() +  
  geom_point(data = df,  
             aes(x = X, y = Y, color = group))
```

# ggplot() “grammar of graphics” in R



Create figures with multiple layers

Different types of “**geoms**” to make different plots:

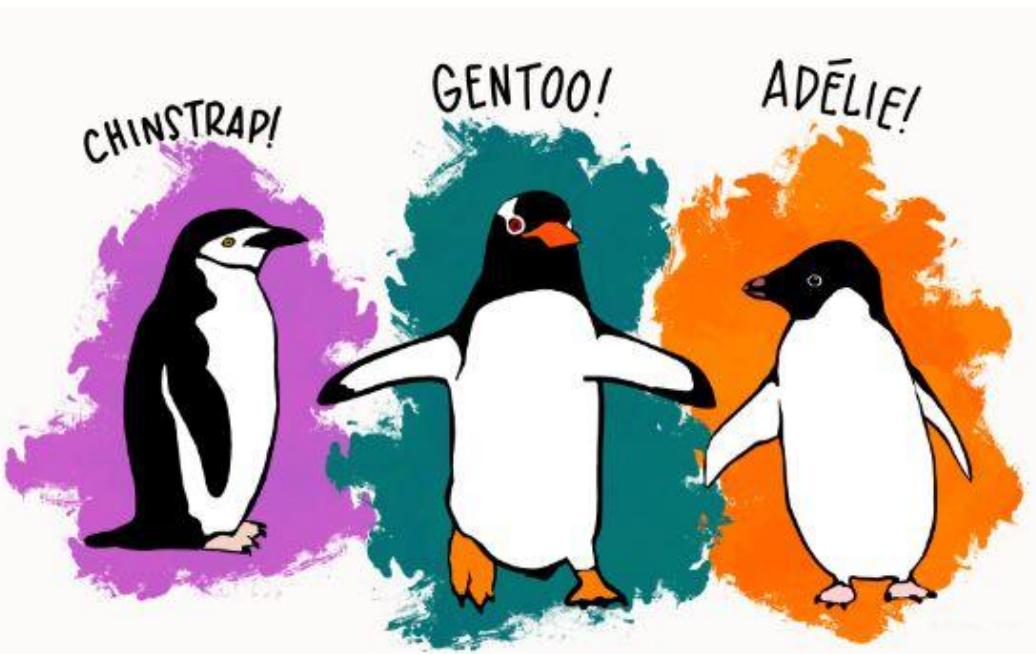
- `geom_point()` scatterplots, spatial points
- `geom_line()`, `geom_path()` line graphs, spatial lines
- `geom_sf()` shapefiles (spatial polygons)
- `geom_label()` print text labels on figure

```
ggplot() +  
  geom_point(data = df,  
             aes(x = X, y = Y, color = group))
```

The code snippet shows the `ggplot()` function followed by a plus sign, then `geom_point()`. The `data` argument is highlighted with a yellow background and has an arrow pointing to it from the word `data` above. The `aes` argument is highlighted with an orange background and has an arrow pointing to it from the word `aesthetics` above.

# ggplot() example

- palmerpenguins dataset
- morphological data for 3 penguin species
- example: bill depth vs bill length



<https://allisonhorst.github.io/palmerpenguins/>



eBird



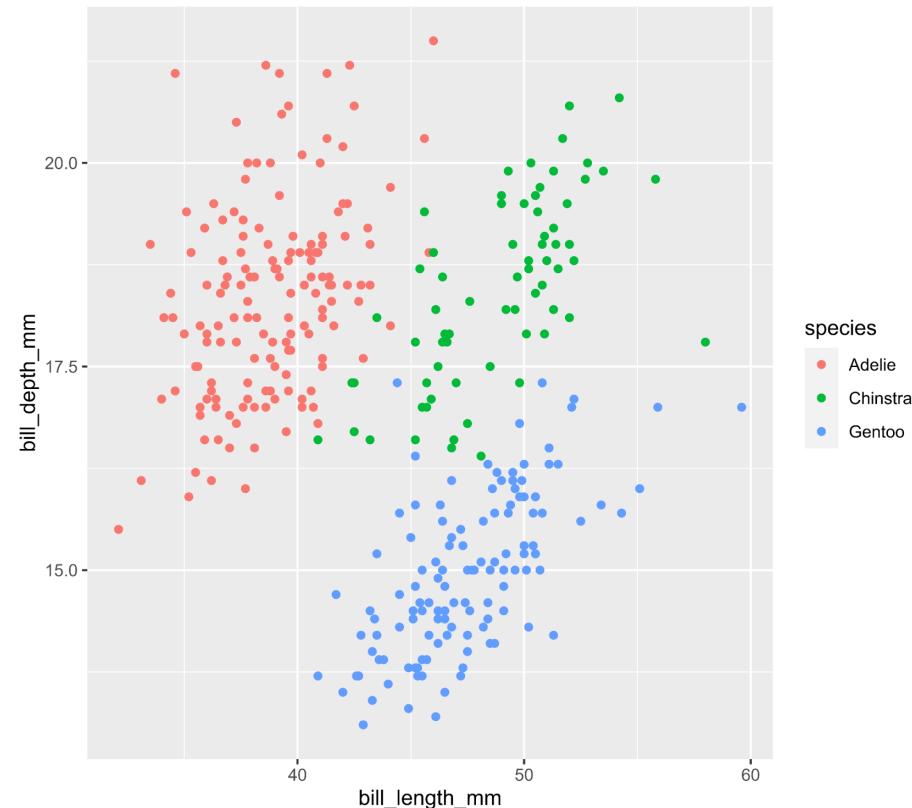
eBird



Australian Museum

# ggplot() example

- palmerpenguins dataset
- scripts/ggplot-example.R
- morphological data for 3 penguin species
- example: bill depth vs bill length

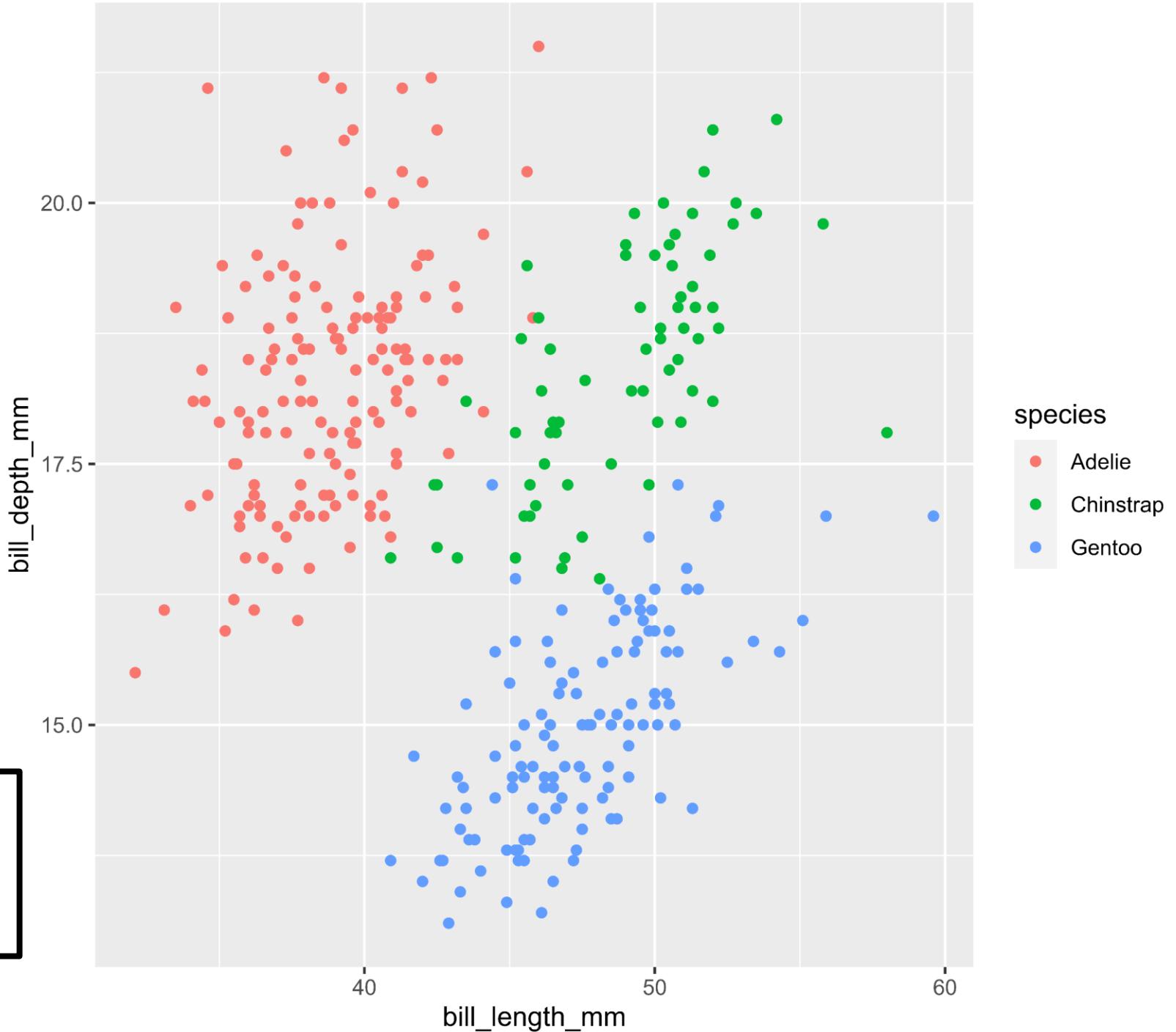


```
data(penguins)

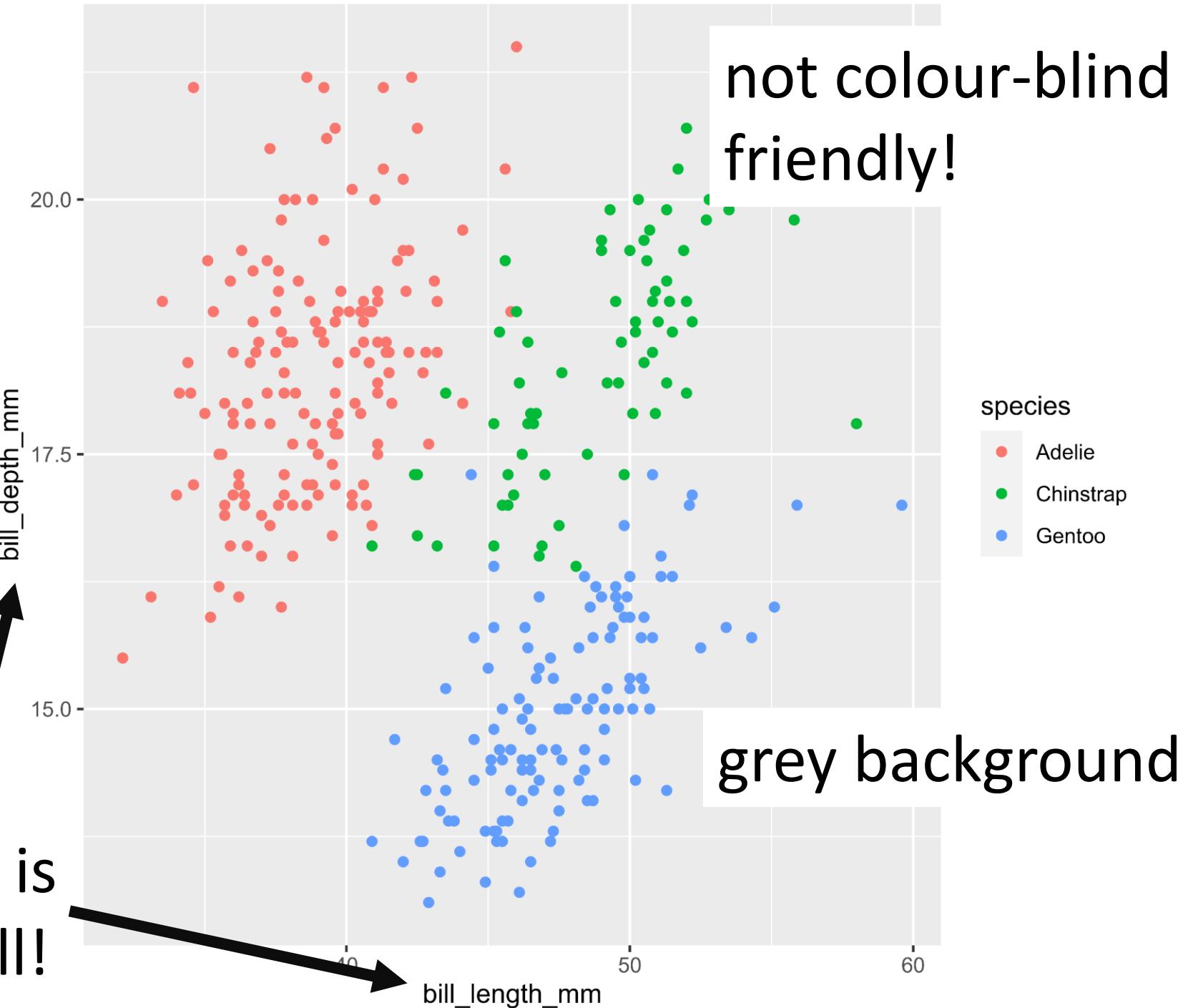
ggplot() +
  geom_point(data = penguins,
             aes(x = bill_length_mm, y = bill_depth_mm, color = species))
```

**DO NOT USE  
ggplot()  
defaults!**

What do you think  
should be changed?



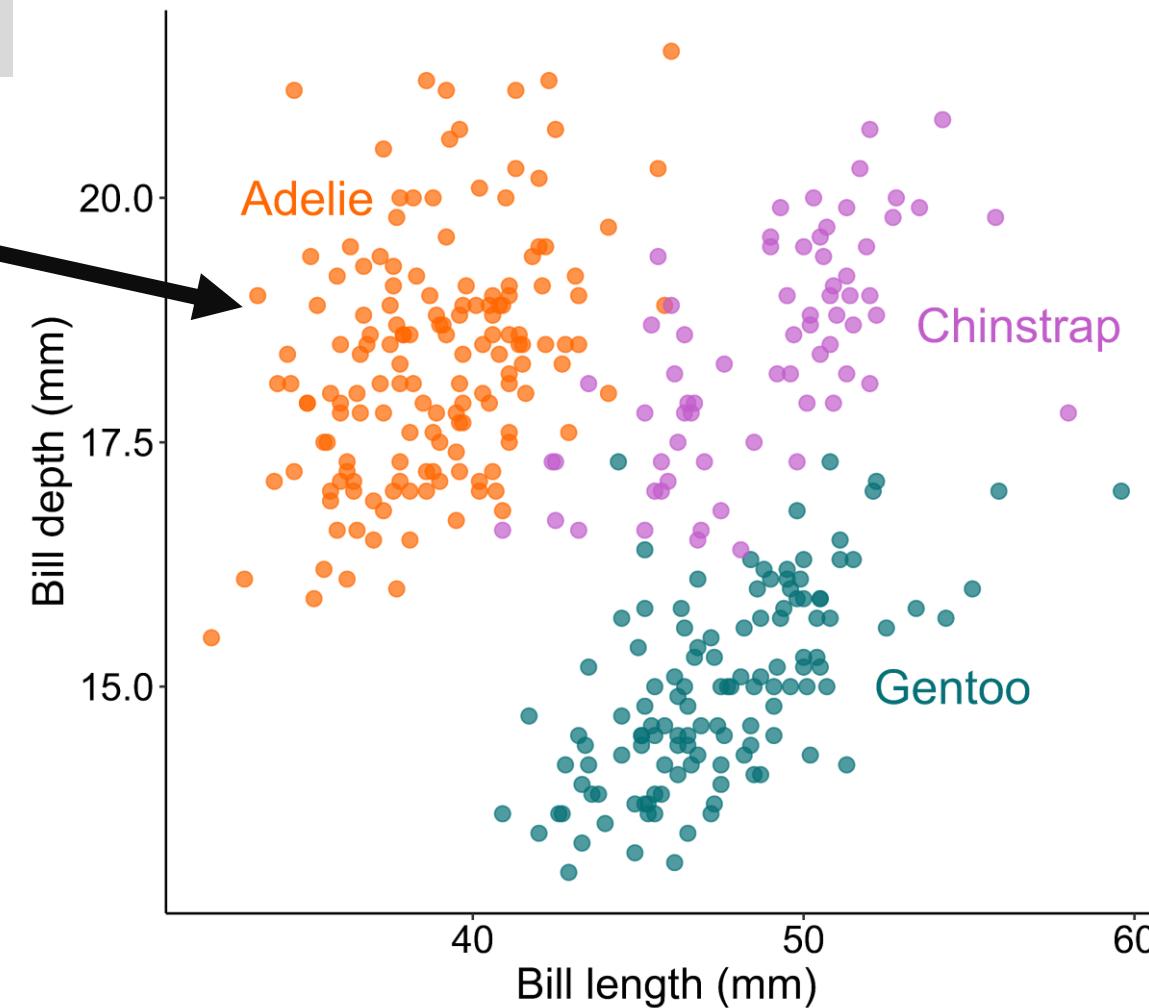
**DO NOT USE  
ggplot()  
defaults!**



```
ggplot() +  
  geom_point(data = penguins,  
             aes(x = bill_length_mm, y = bill_depth_mm, color = species))
```

```
geom_point(size = 2.7, alpha = 0.7)
```

increase point size

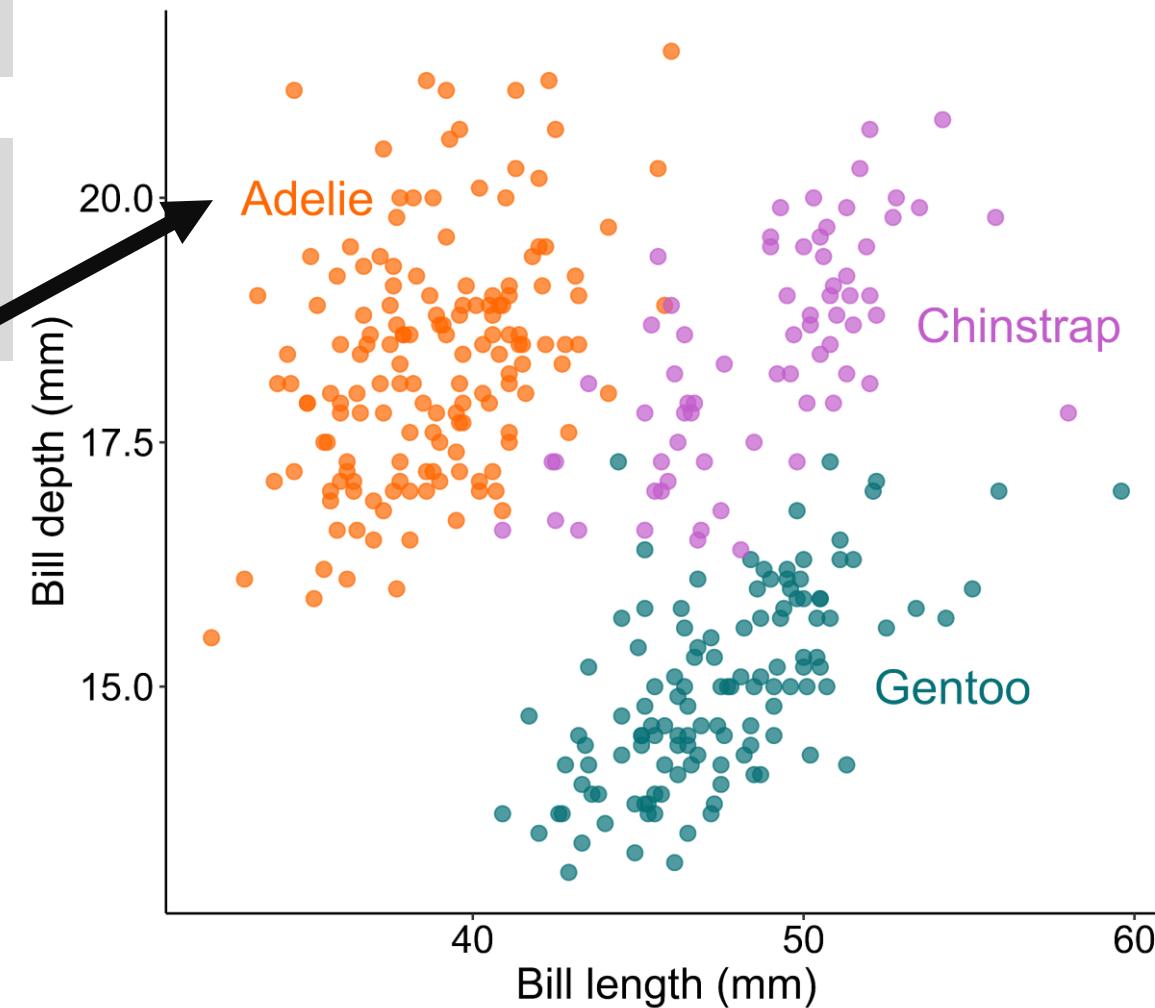


```
ggplot() +  
  geom_point(data = penguins,  
             aes(x = bill_length_mm, y = bill_depth_mm, color = species))
```

```
  geom_point(size = 2.7, alpha = 0.7)
```

```
  geom_text(aes(..., label = species,  
               color = species) +  
  theme(legend.position = "none")
```

species names on plot,  
instead of legend



```
ggplot() +  
  geom_point(data = penguins,  
             aes(x = bill_length_mm, y = bill_depth_mm, color = species))
```

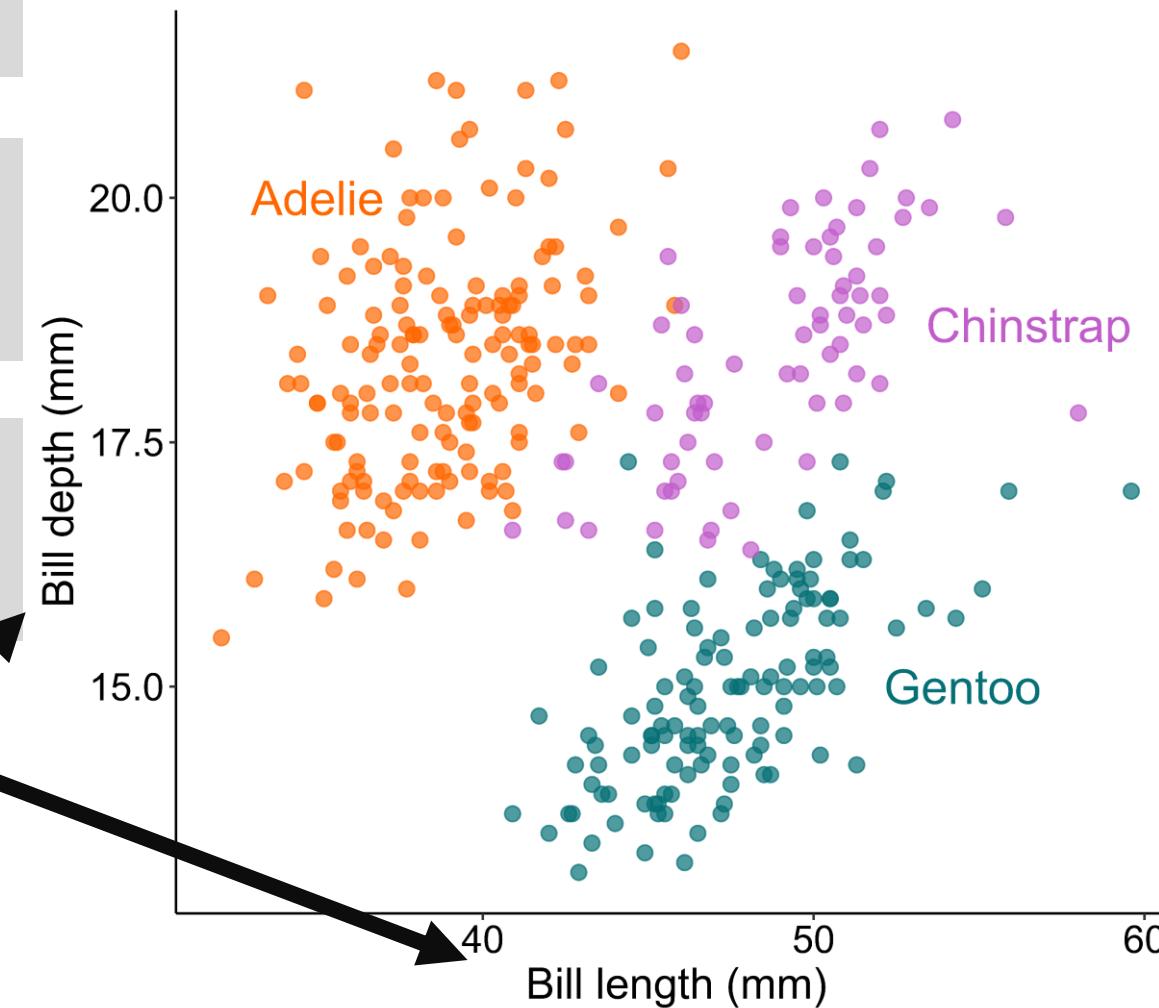
```
  geom_point(size = 2.7, alpha = 0.7)
```

```
  geom_text(aes(..., label = species,  
                color = species) +  
            theme(legend.position = "none")
```

```
  theme_classic() +  
  theme(axis.title = element_text(size = 18,  
                                   axis.text = element_text(size = 16)))
```

remove grey  
background

increase axis text



```
ggplot() +  
  geom_point(data = penguins,  
             aes(x = bill_length_mm, y = bill_depth_mm, color = species))
```

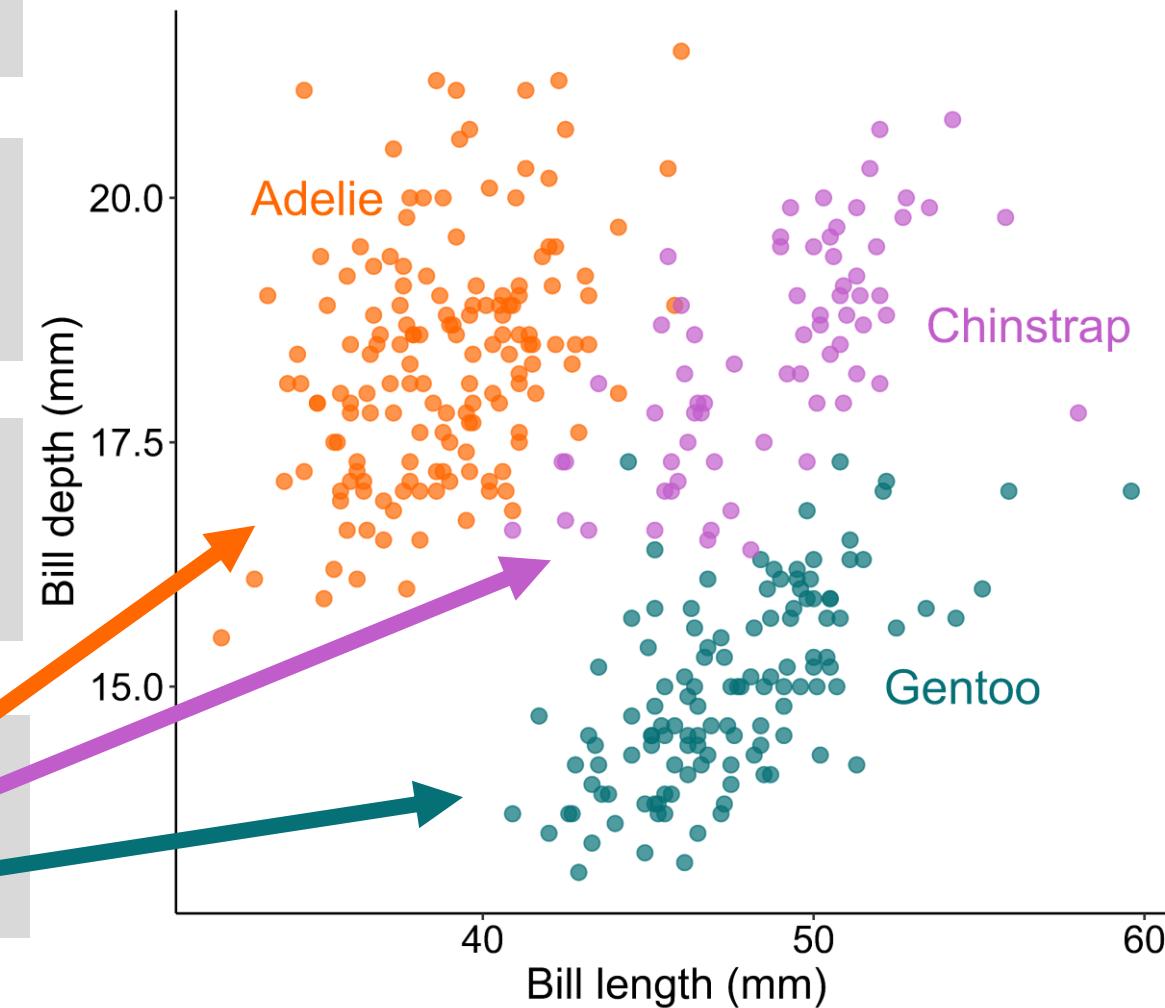
```
  geom_point(size = 2.7, alpha = 0.7)
```

```
  geom_text(aes(..., label = species,  
                color = species) +  
            theme(legend.position = "none")
```

```
  theme_classic() +  
  theme(axis.title = element_text(size = 18,  
                                   axis.text = element_text(size = 16))
```

```
  scale_colour_manual(values = c("#ff6700",  
                           "#c15ccb",  
                           "#057076"))
```

Custom colour palettes  
with hex codes



# Adding organism silhouettes with rphylopic

```
add_phylopic(name = "Pygoscelis papua,  
              x = 58.6, y = 20.5, ysize = 2)
```

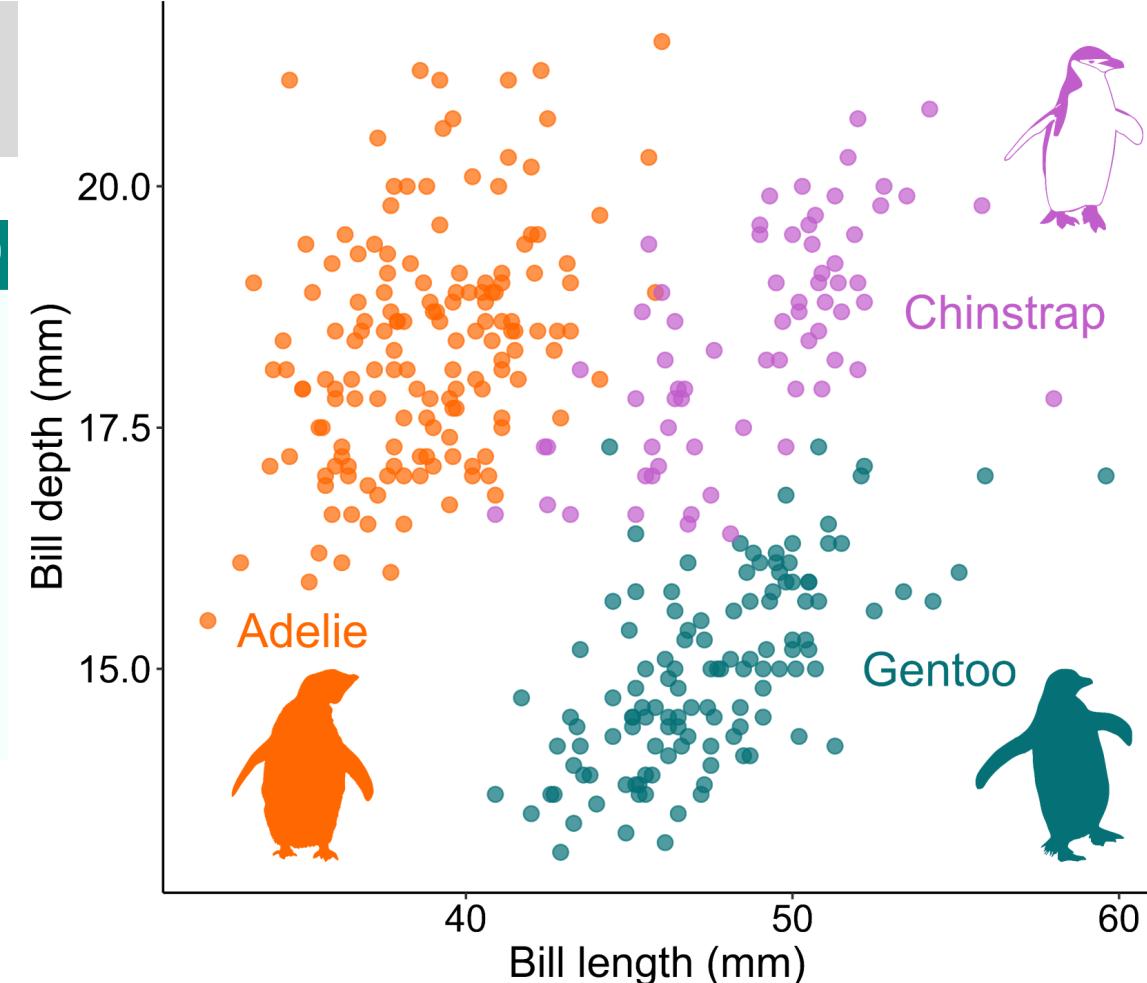
**PHYLOPic** Enter the name of a group of organisms.

Free silhouette images of animals, plants, and other life forms, available for reuse under Creative Commons licenses.

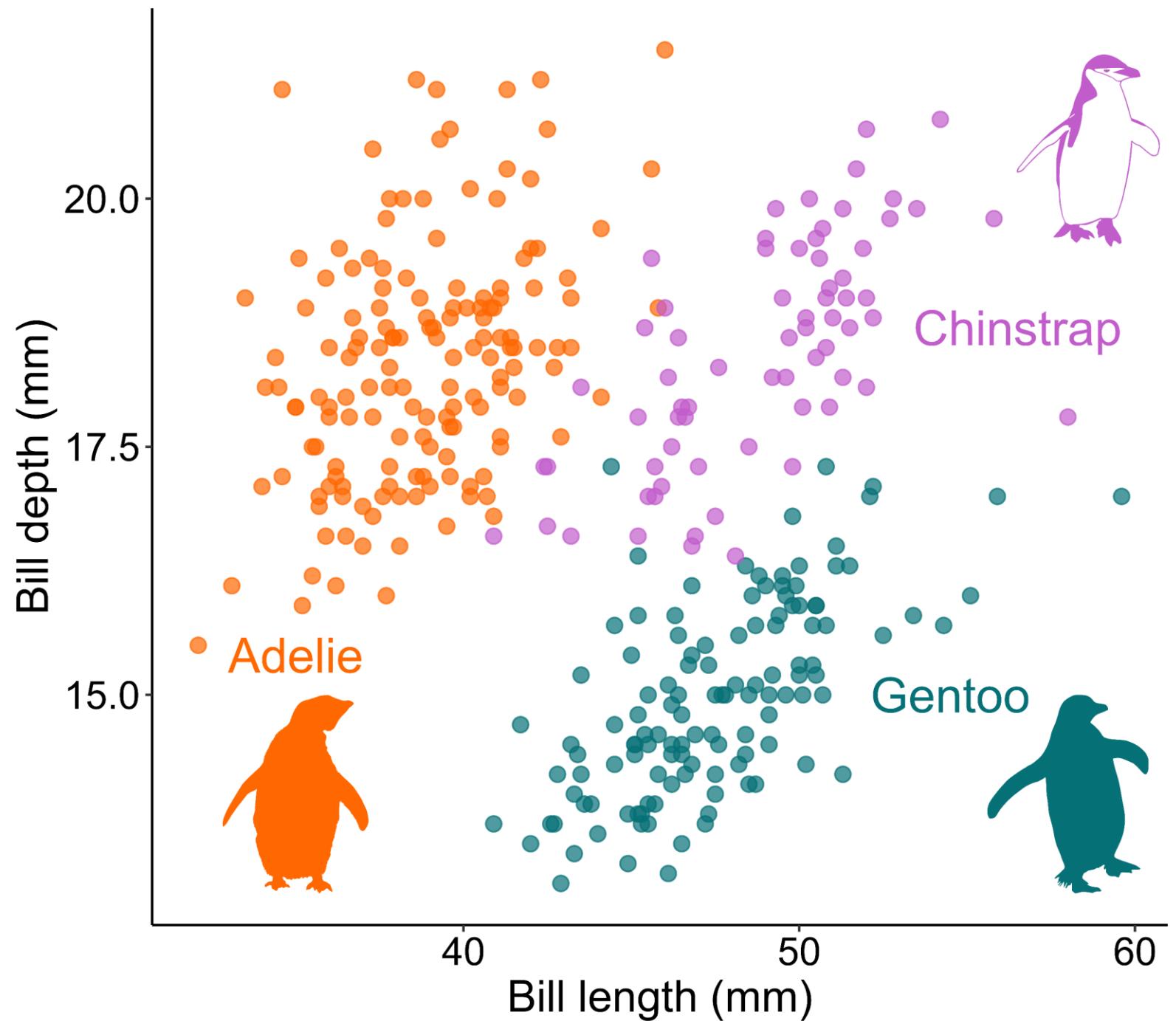
Latest Uploads

SEE MORE →

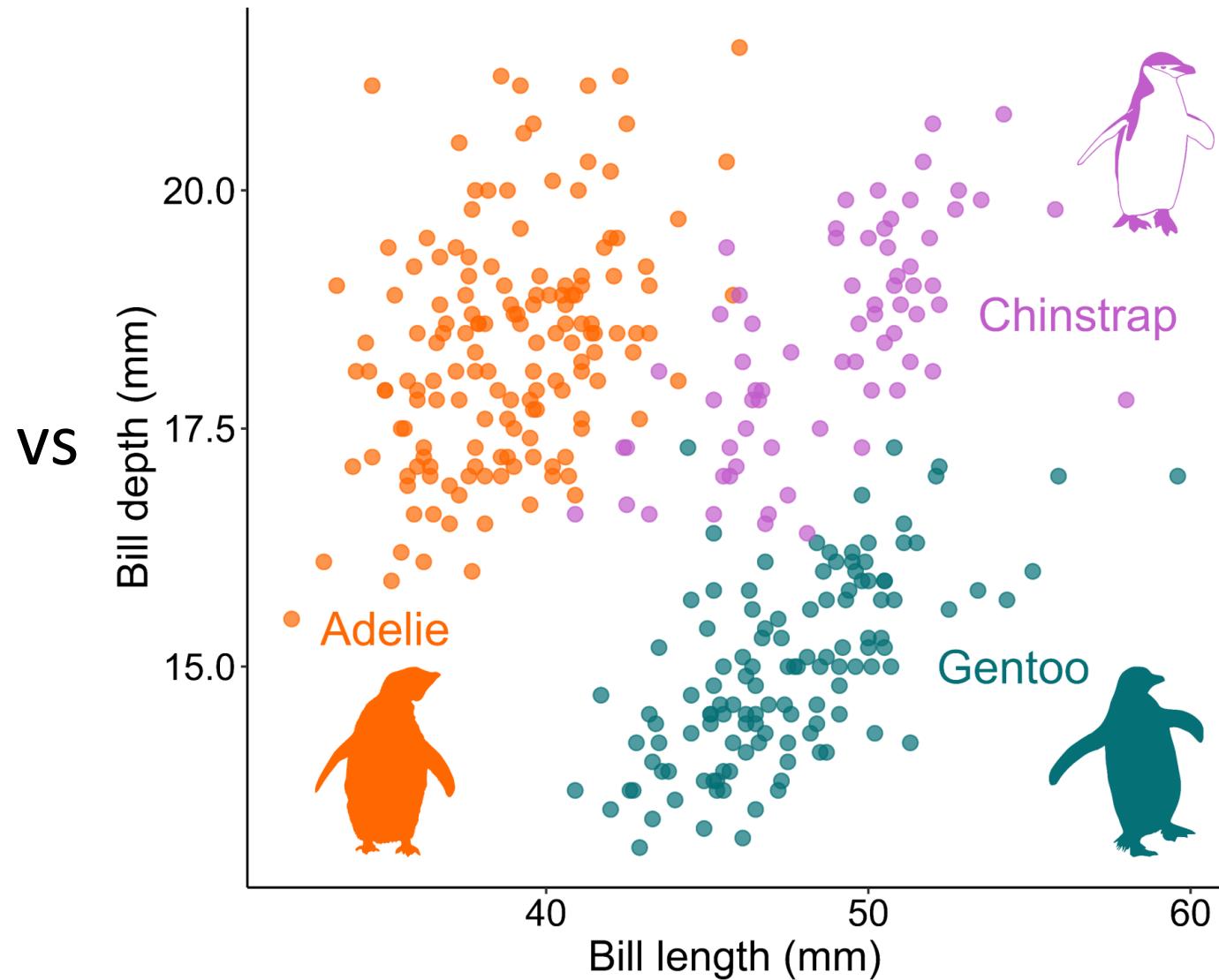
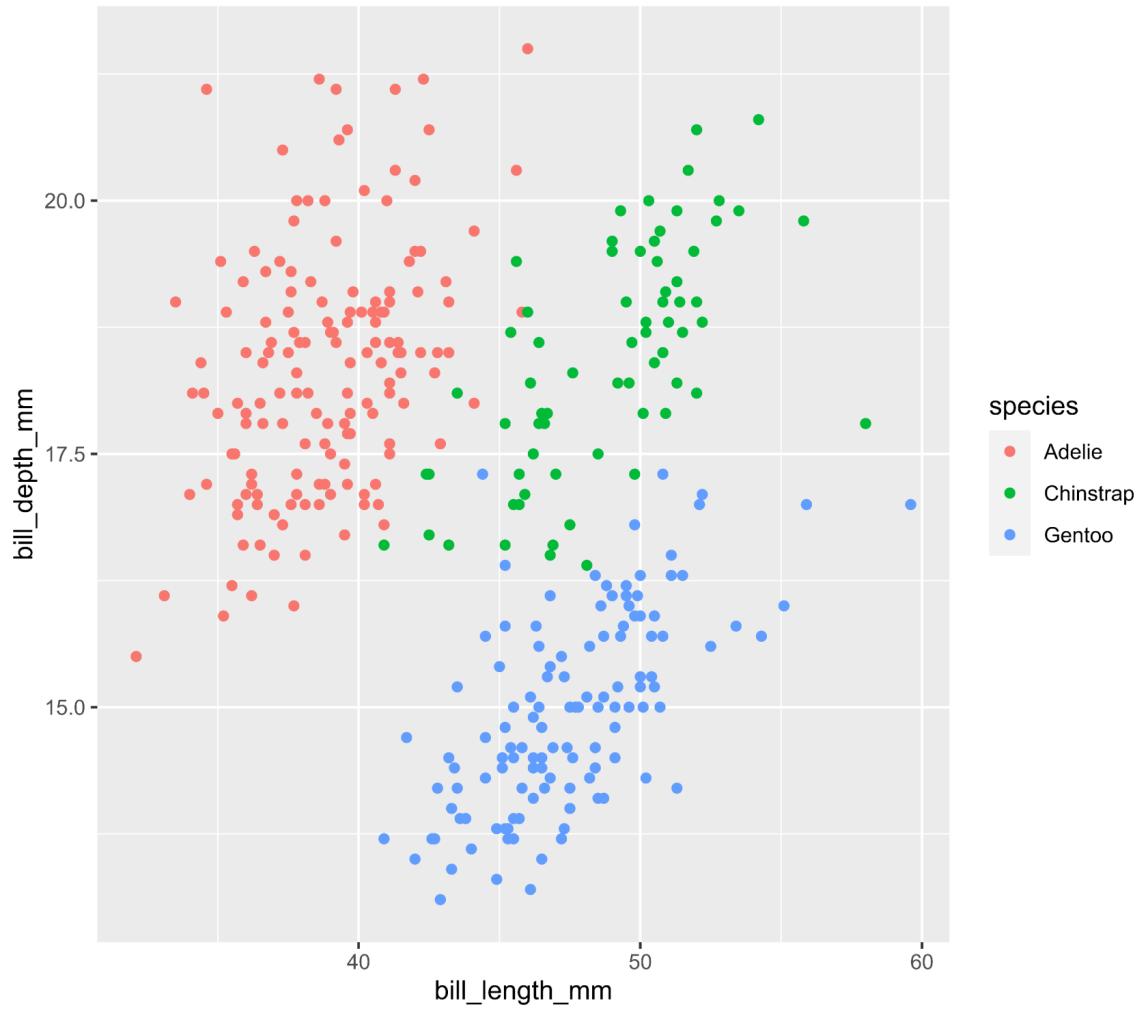
Gento penguin scientific name

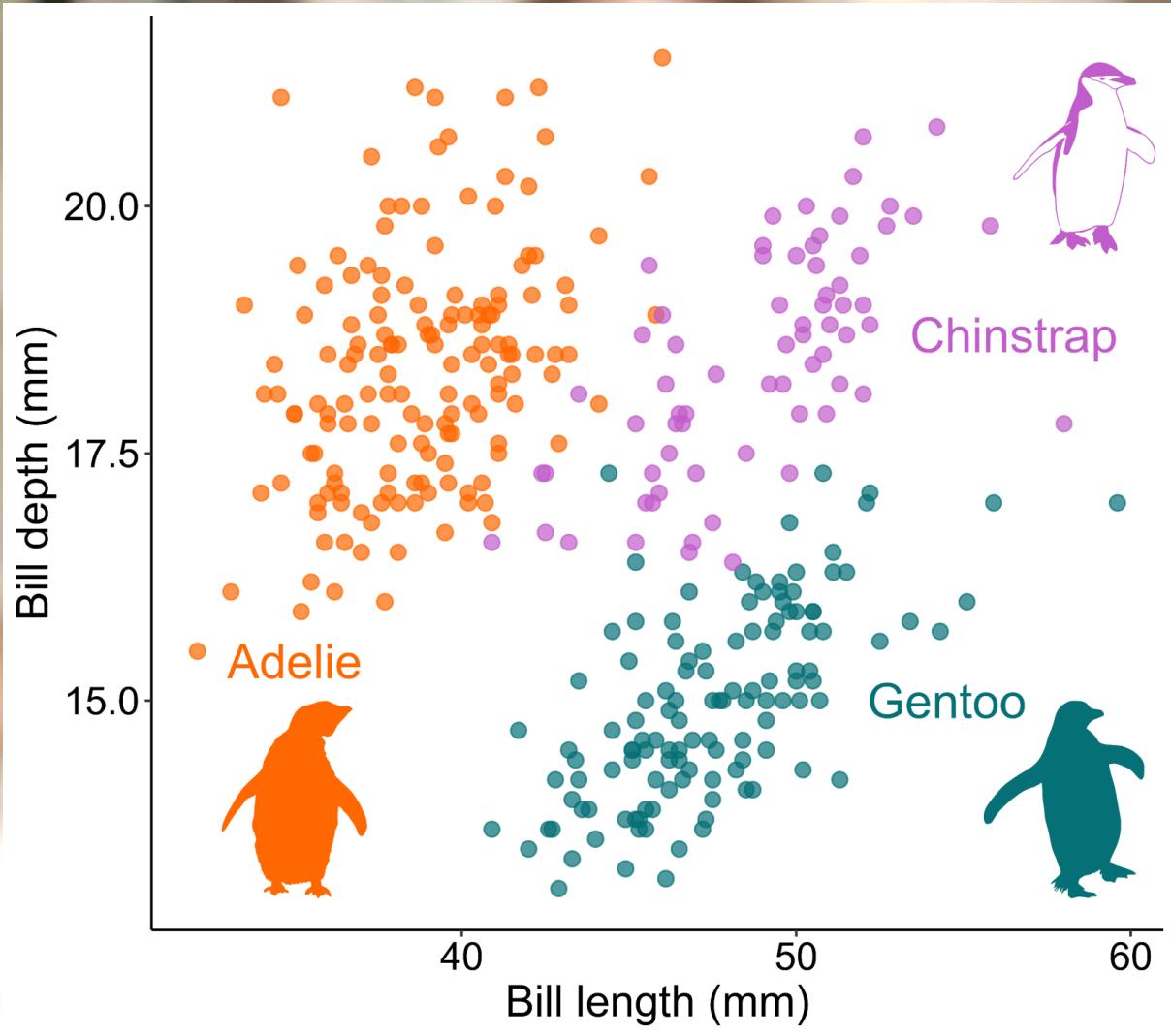


<https://www.phylopic.org/>



# Going beyond `ggplot()` defaults is easy and makes a big difference!





# ggplot() summary

Different geoms: points, lines, polygons, text, etc

Do not use ggplot() defaults!

Have fun/be creative with ggplot()!

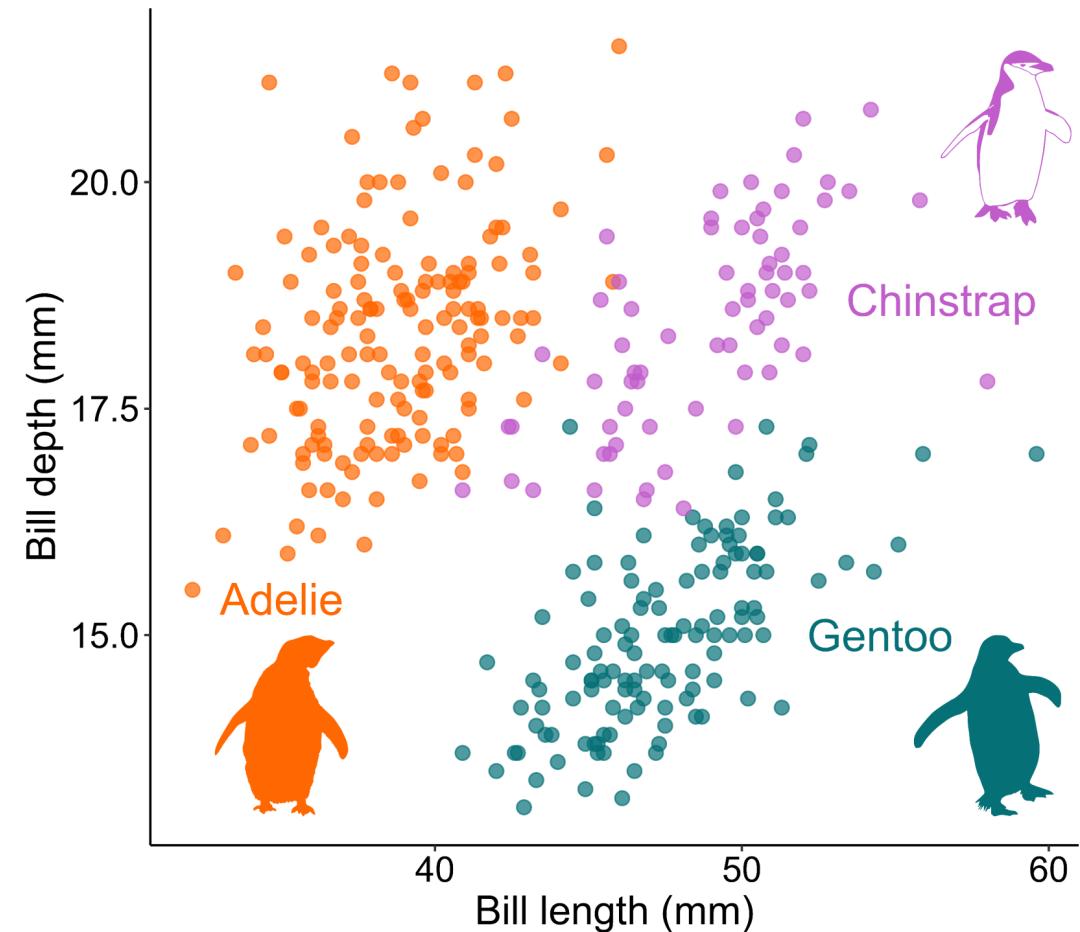
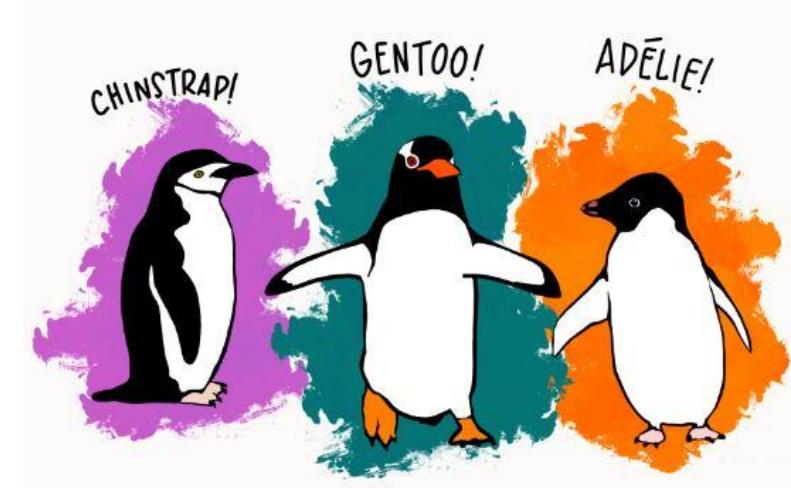
## Additional resources

Hadley Wickham. R for data science

<https://r4ds.had.co.nz/data-visualisation.html>

Hadley Wickham. ggplot2: Elegant graphics for data analysis

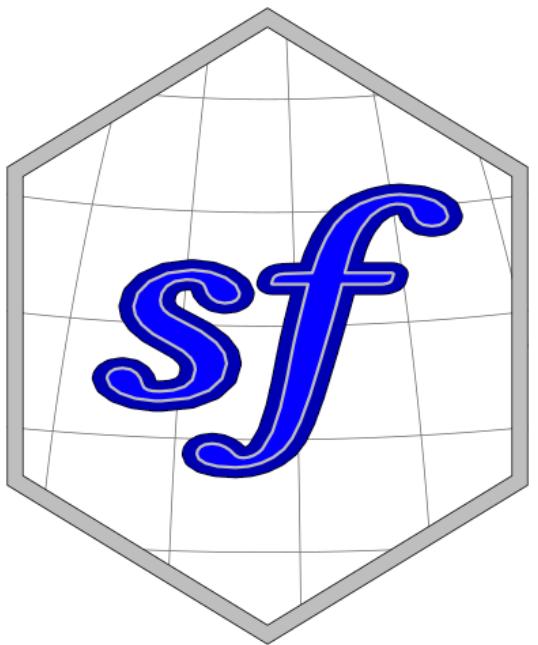
<https://ggplot2-book.org/>





Back to maps!

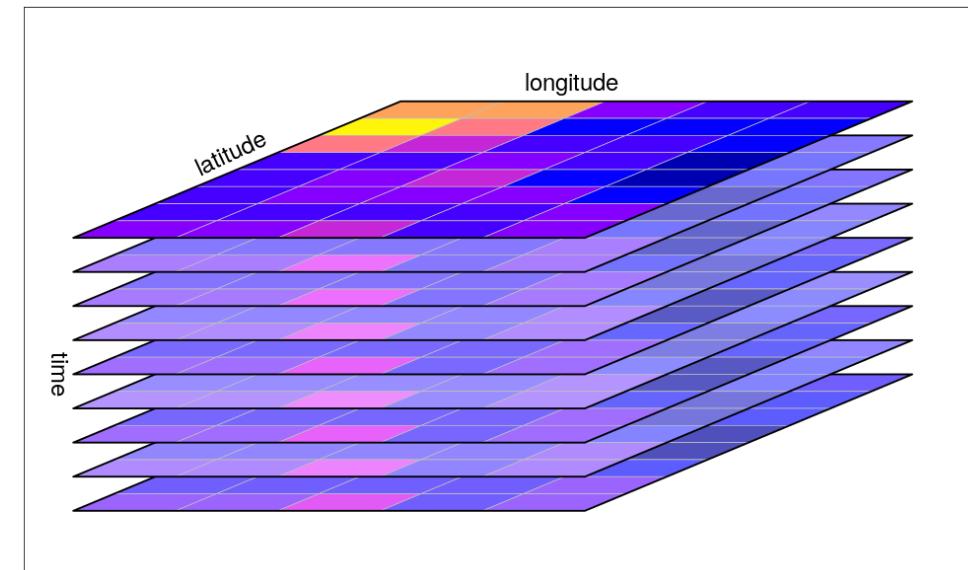
# Which R packages to use?



vectors



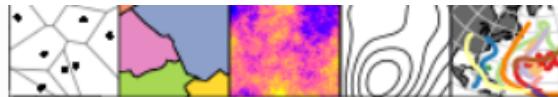
rasters



stars

“raster cubes” and other vector/raster spatiotemporal arrays

e.g., satellite data (composed of multiple bands at different wavelengths)



# R-spatial evolution: retirement of rgdal, rgeos and maptools

Apr 12, 2022 • Edzer Pebesma, Roger Bivand

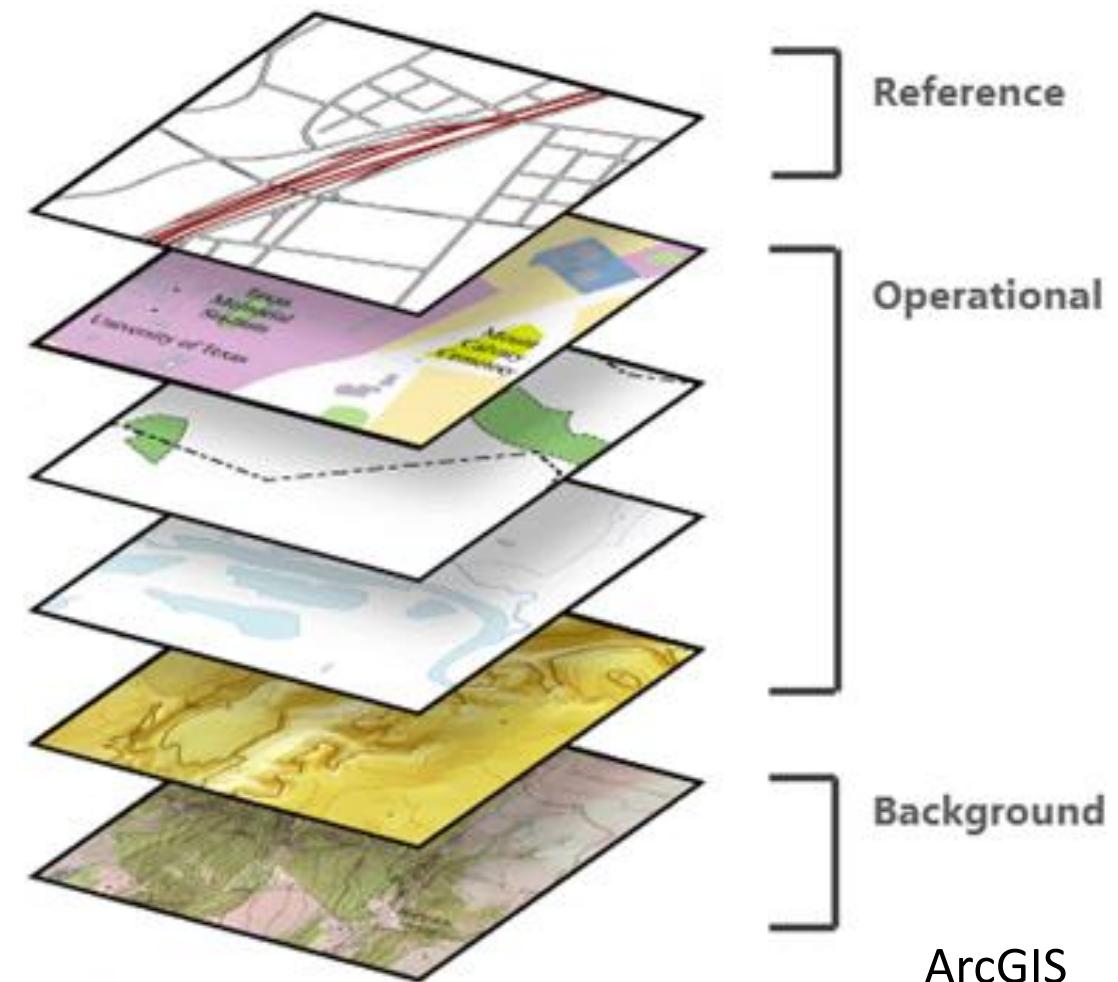
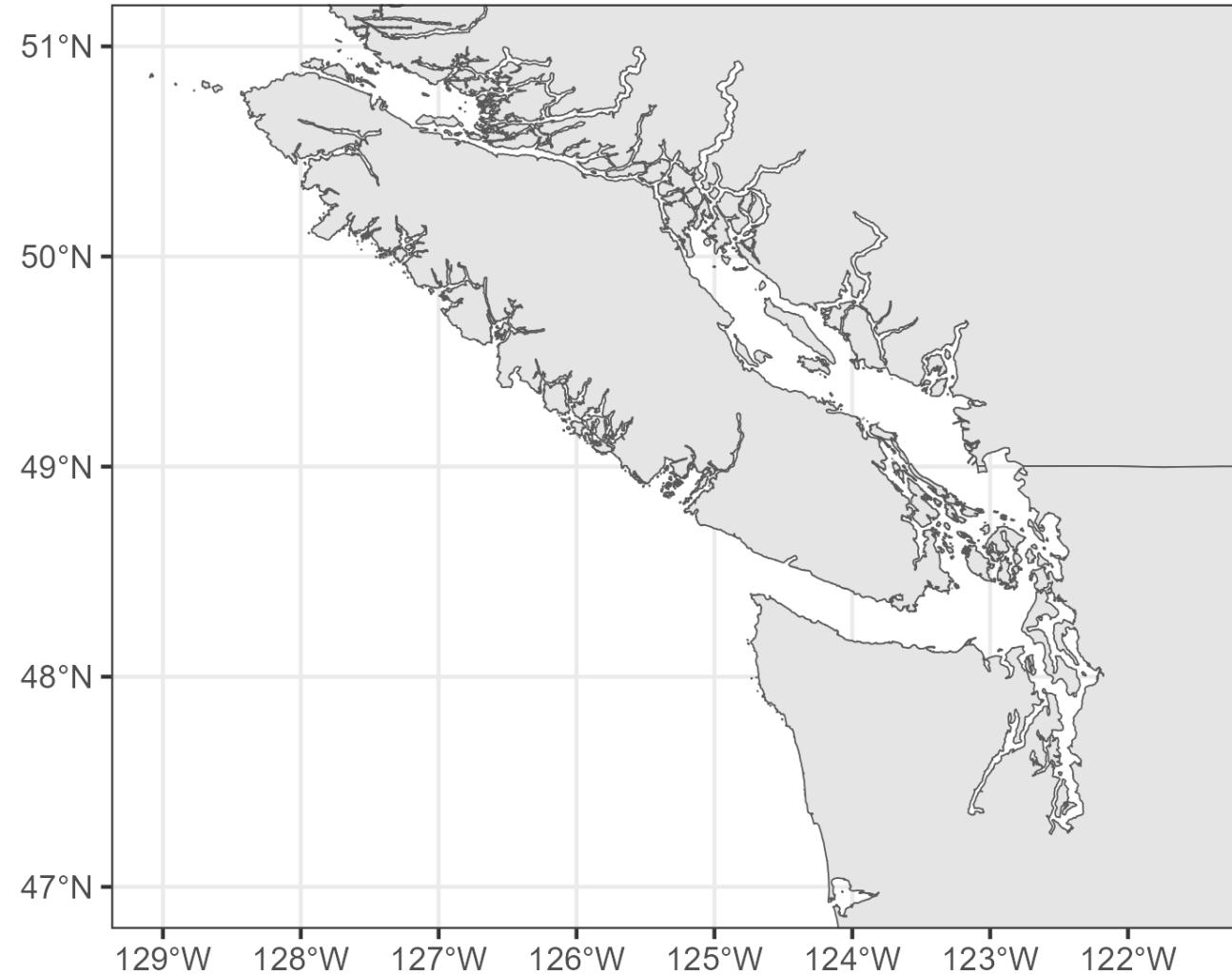
- [Why this blog?](#)
- [Why rgdal and rgeos will retire](#)
- [Packages depending on rgdal and rgeos](#)
- [The Plan](#)
- [Packages depending on sp and raster](#)

[\[view raw Rmd\]](#)

**Summary:** Packages rgdal, rgeos and maptools will retire by the end of 2023 . We describe where their functionality will go, what package maintainers can or should do, and which steps we will take to minimize the impact on dependent packages and on reproducibility in general.

# Where to start: basemaps

BC coast shapefile



Reference

Operational

Background

ArcGIS

# Built in basemaps in R

## **RNaturalEarth package**

coarse resolution, great for global maps, bad for regional/local maps

## **OpenStreetMap package**

load in satellite and other basemaps

## **basemaps package**

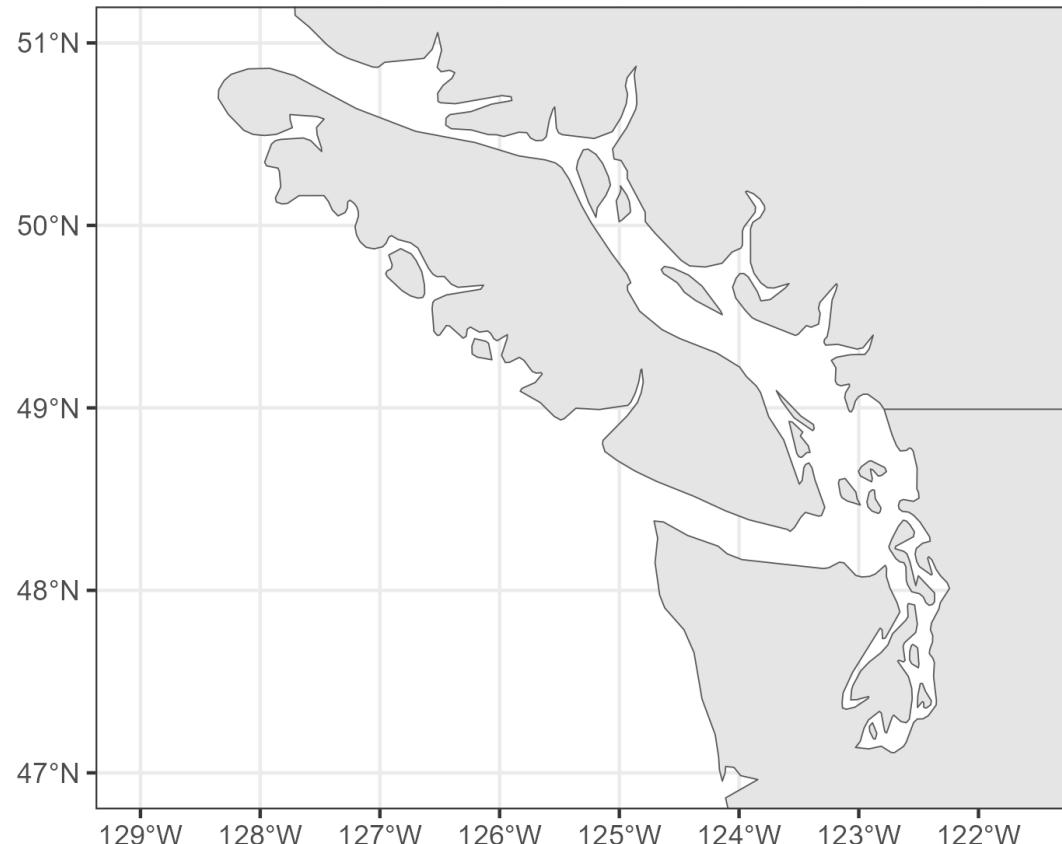
satellite and other basemaps

Can also load in your own basemaps with  
`geom_sf()`

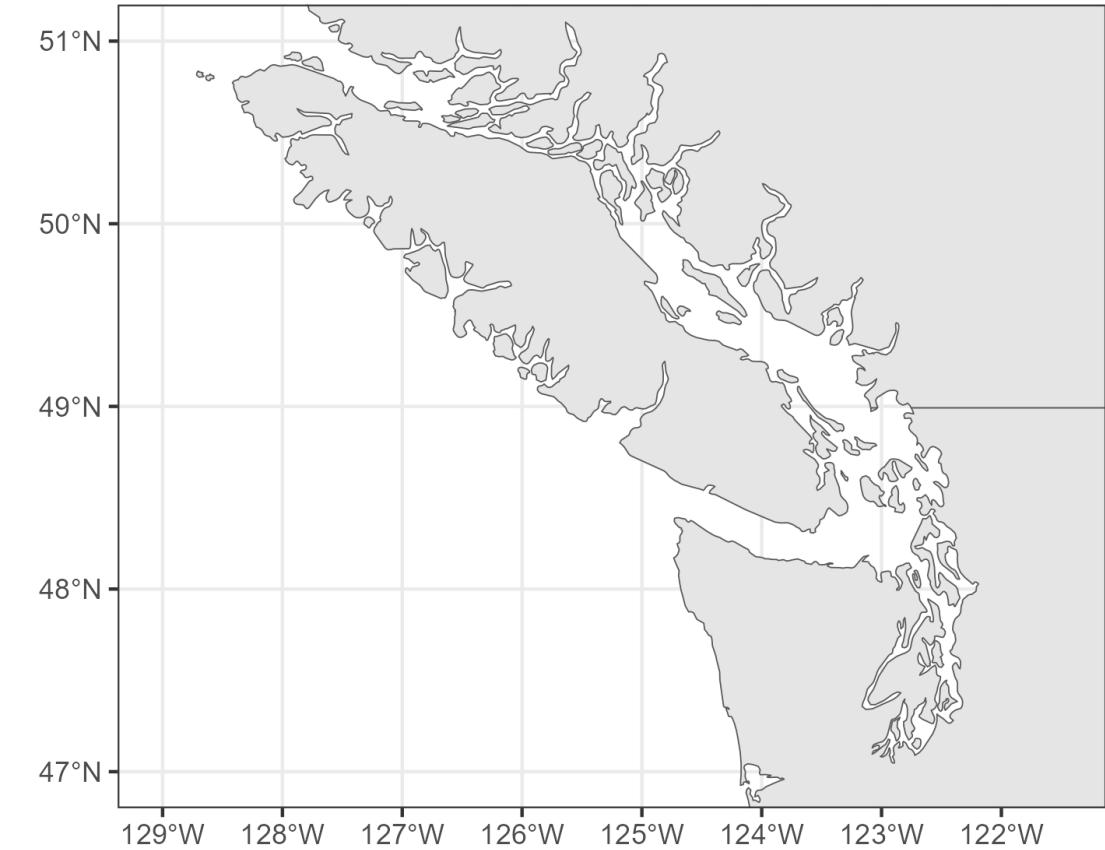
# RNaturalEarth

Includes different resolutions: **1:10M is best for regional-scale maps**

**RNaturalEarth 1:50M**



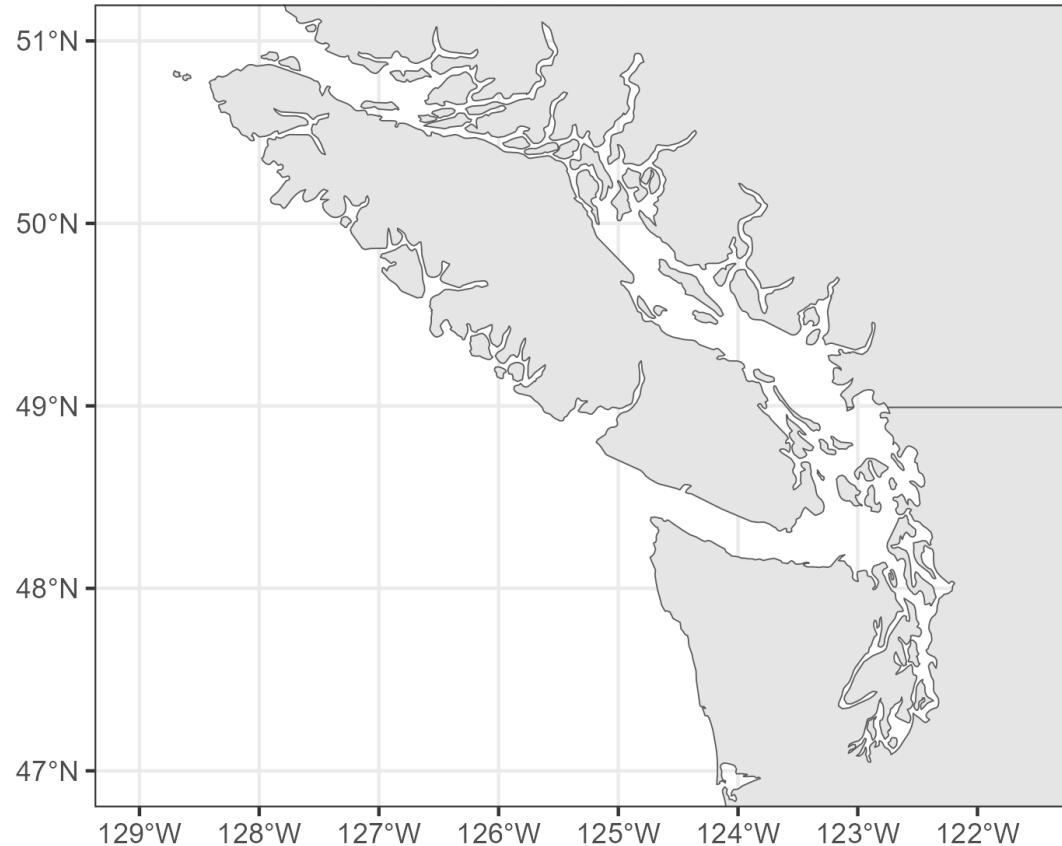
**RNaturalEarth 1:10M**



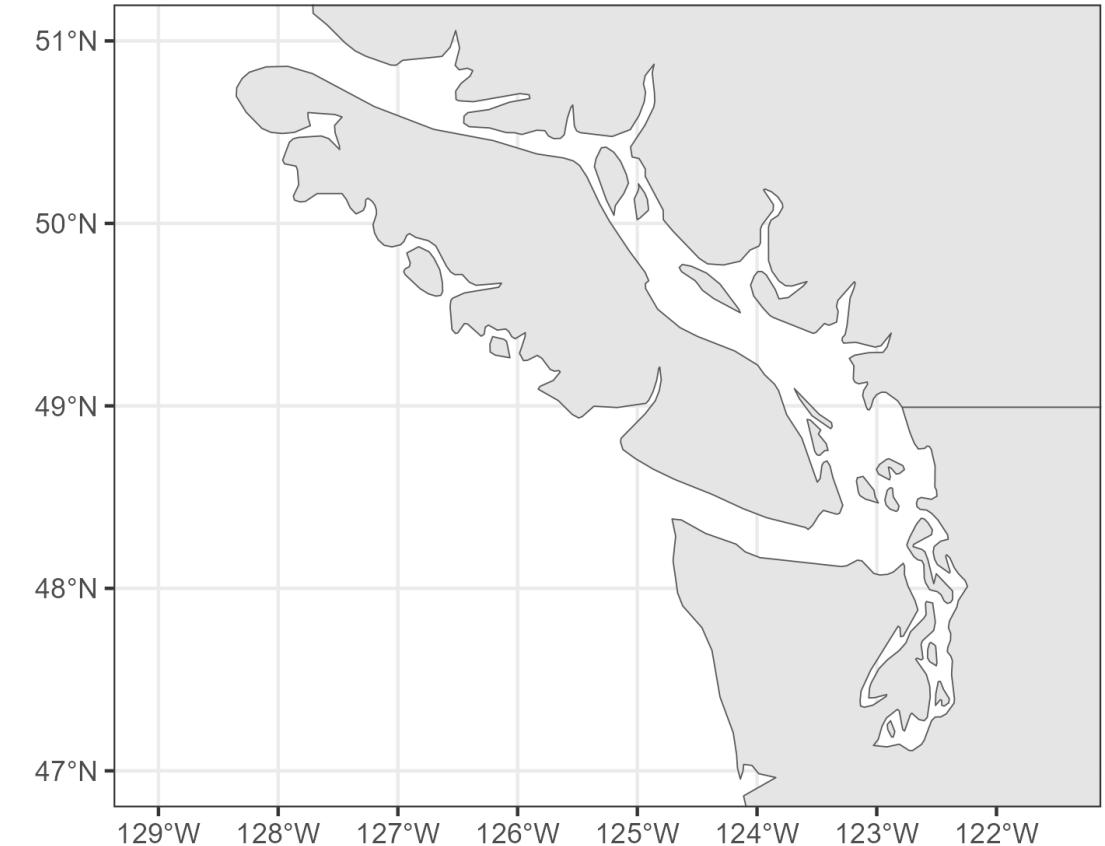
# RNaturalEarth

Includes different resolutions: **1:10M is best for regional-scale maps**

**RNaturalEarth 1:10M**



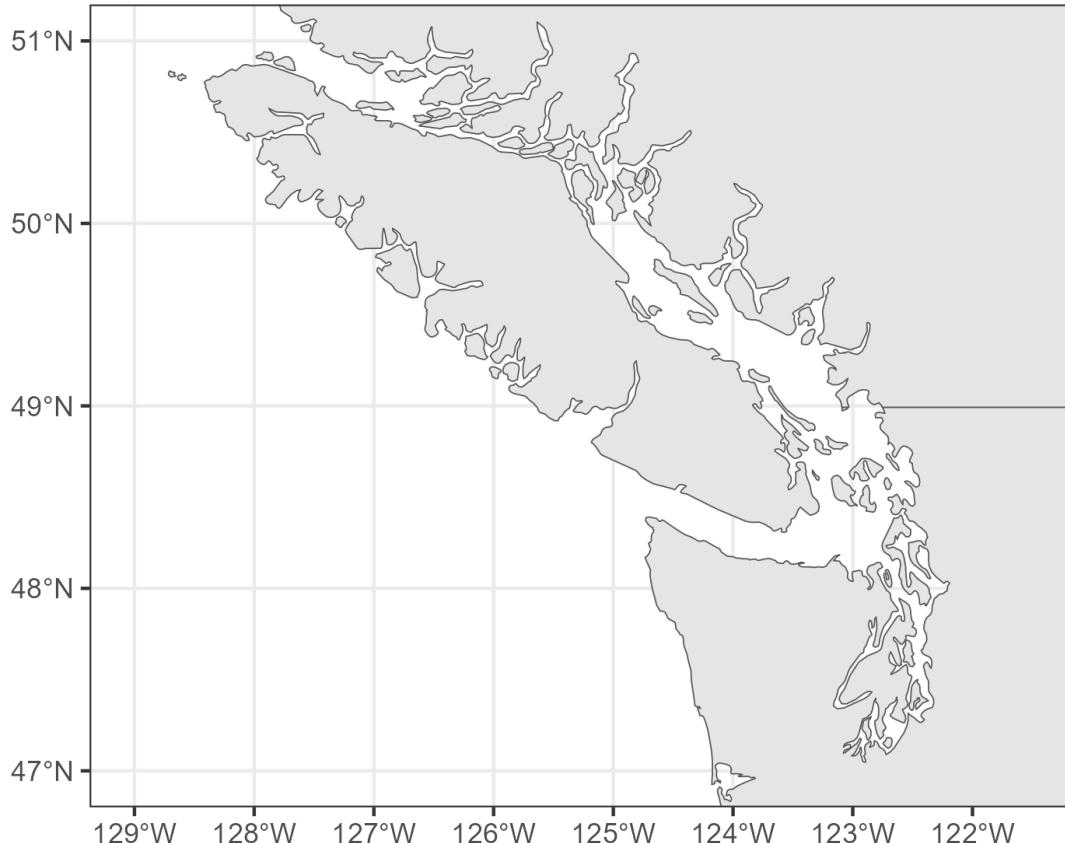
**RNaturalEarth 1:50M**



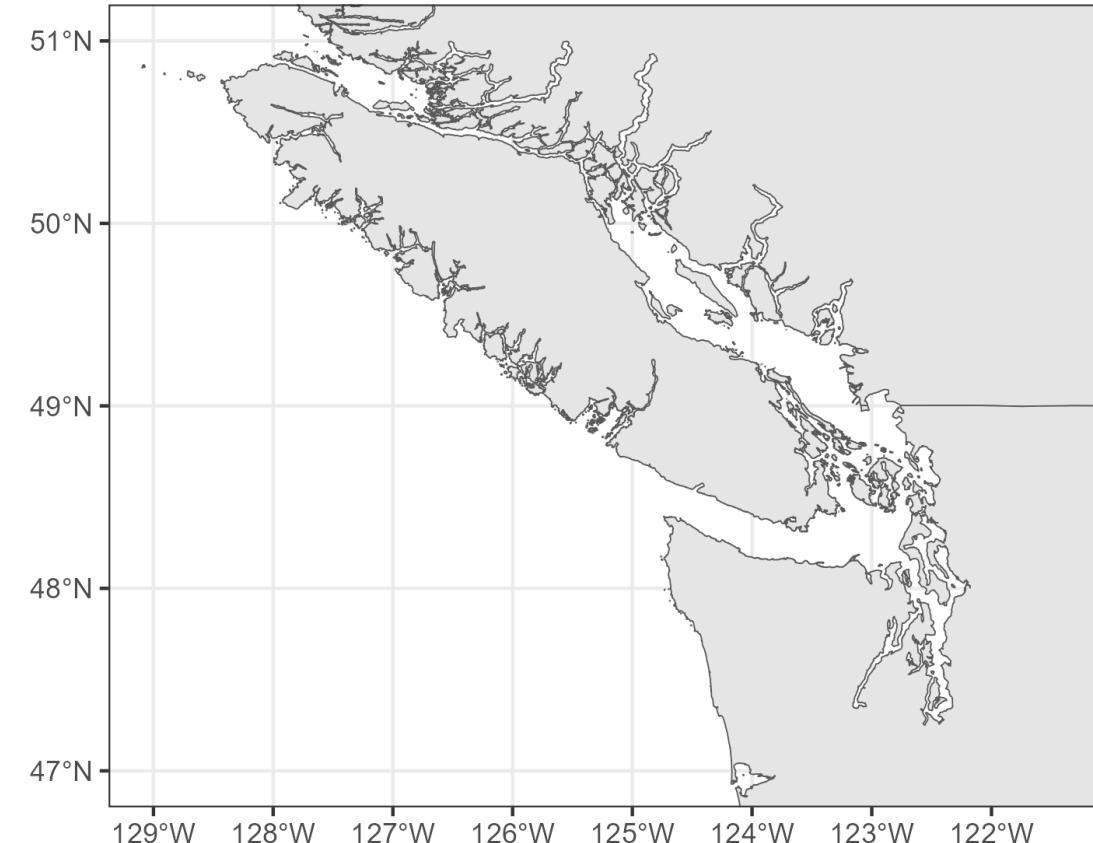
# Compare RNaturalEarth to bc-coast.shp

**bc-coast.shp:** shapefile provided in data folder of Github repository

**RNaturalEarth 1:10M**

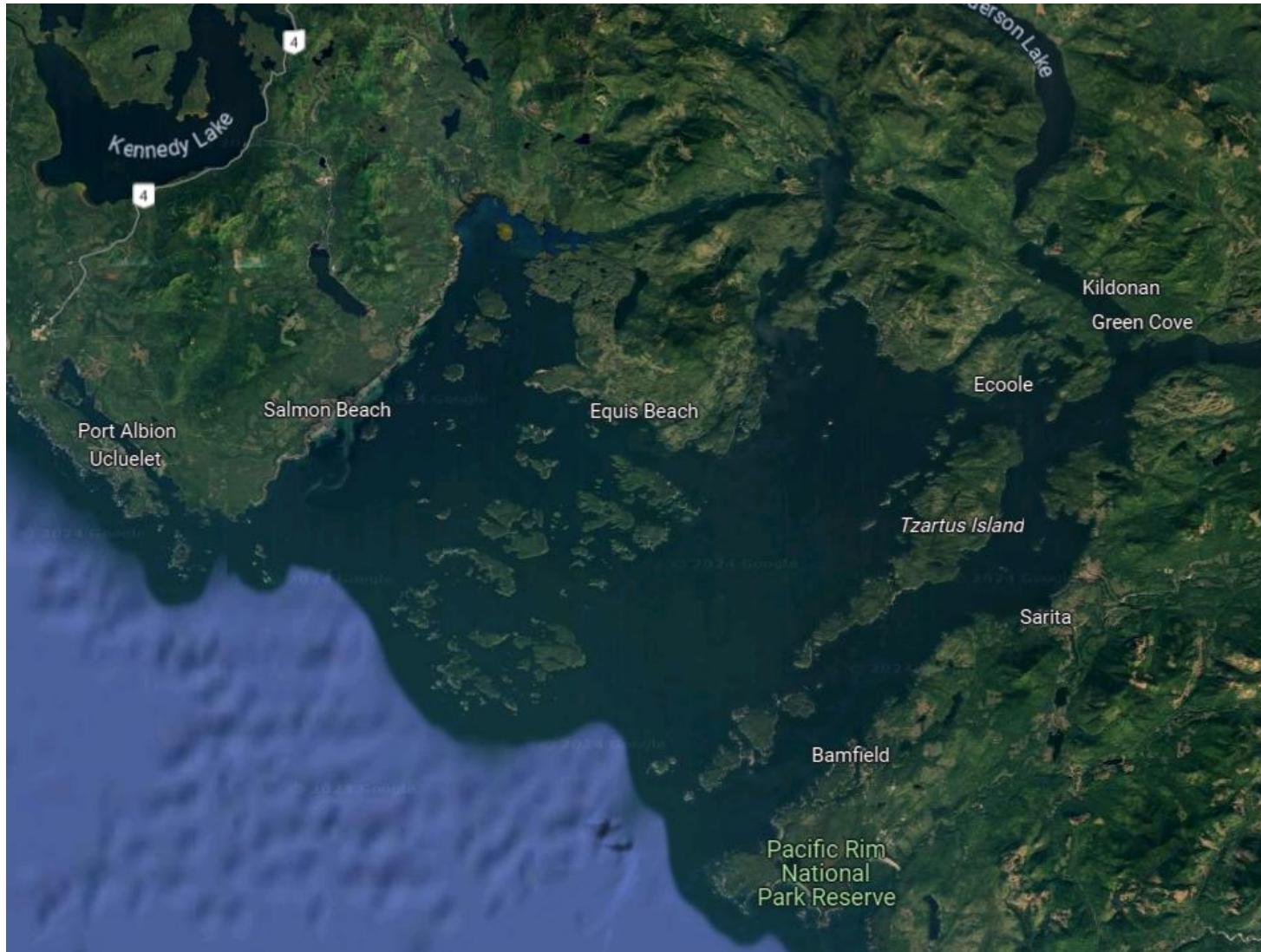


**BC coast shapefile**



# Compare RNaturalEarth to bc-coast.shp

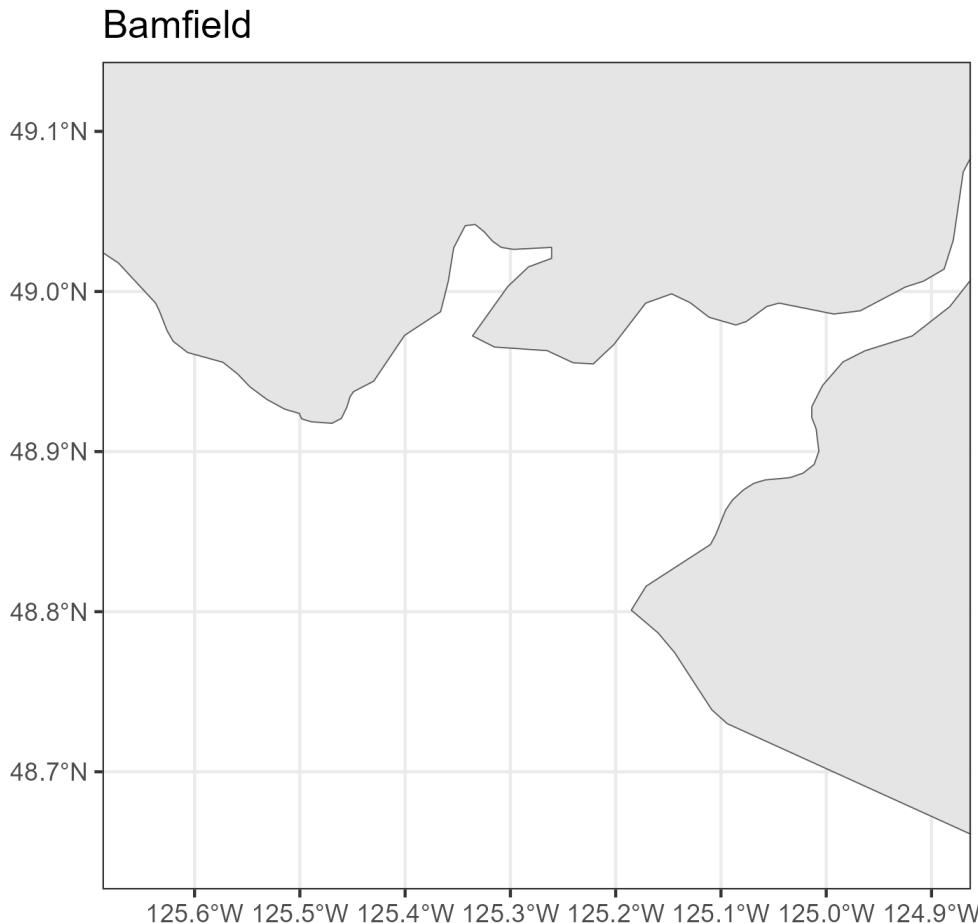
## Bamfield, west coast of Vancouver Island



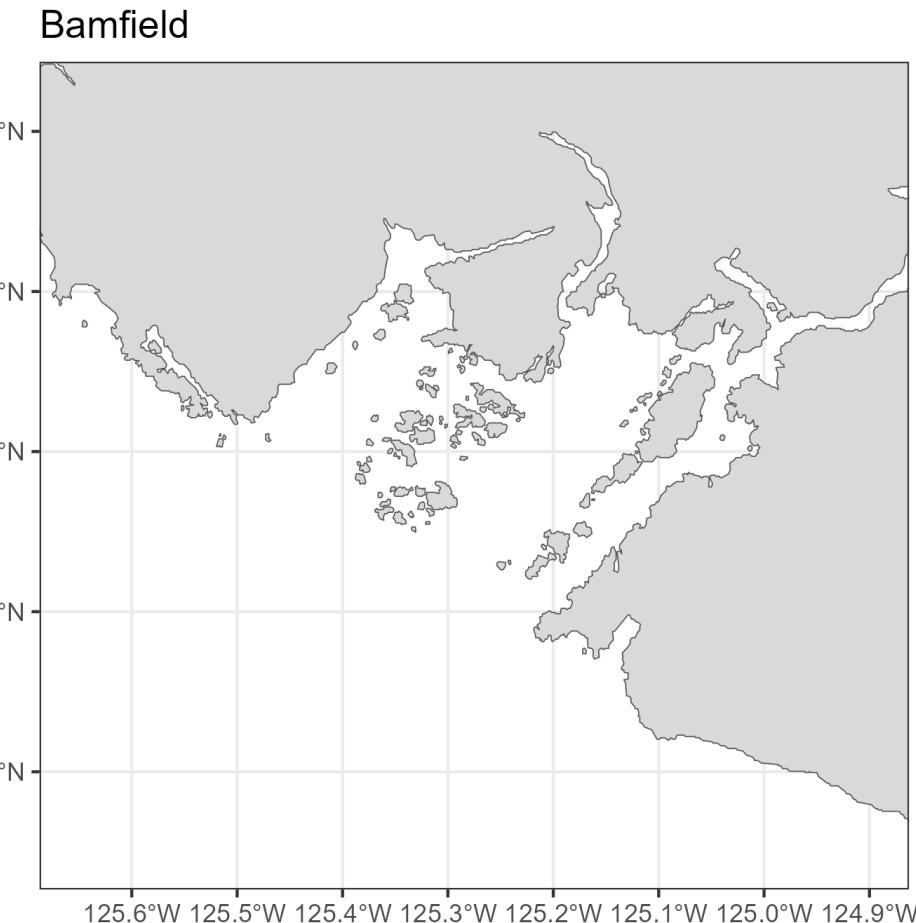
# Compare RNaturalEarth to bc-coast.shp

Bamfield, west coast of Vancouver Island

**RNaturalEarth 1:10M**

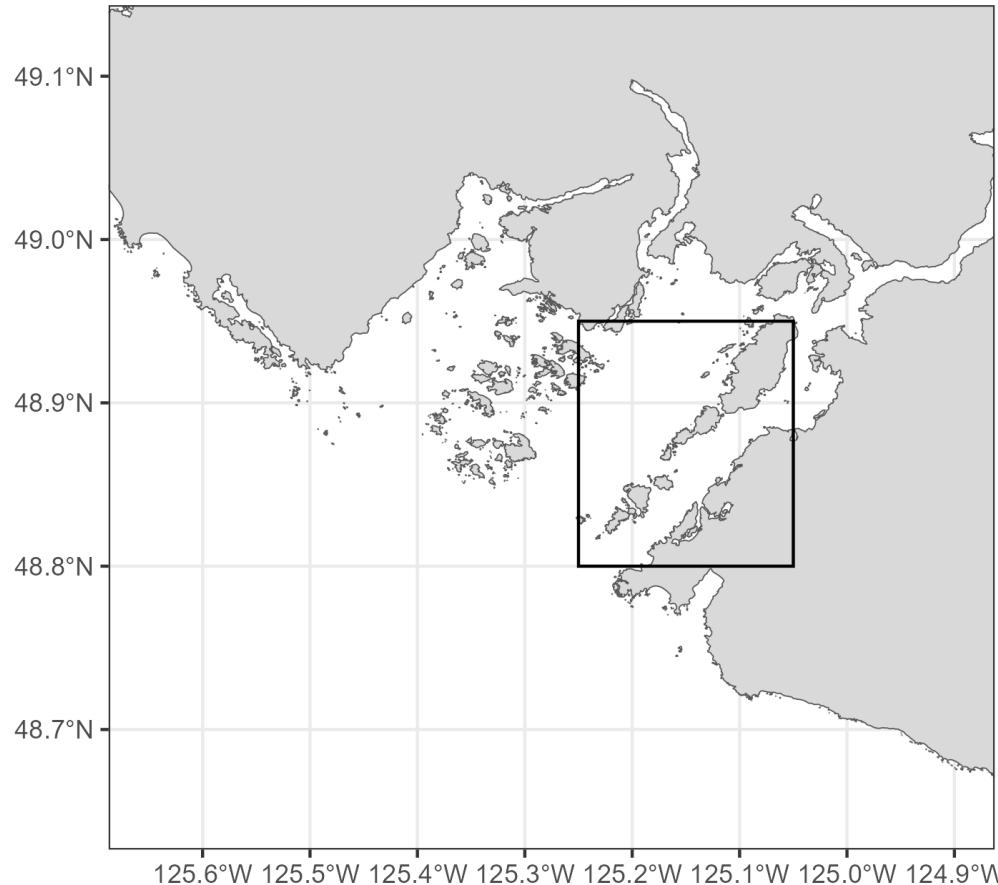


**BC coast shapefile**

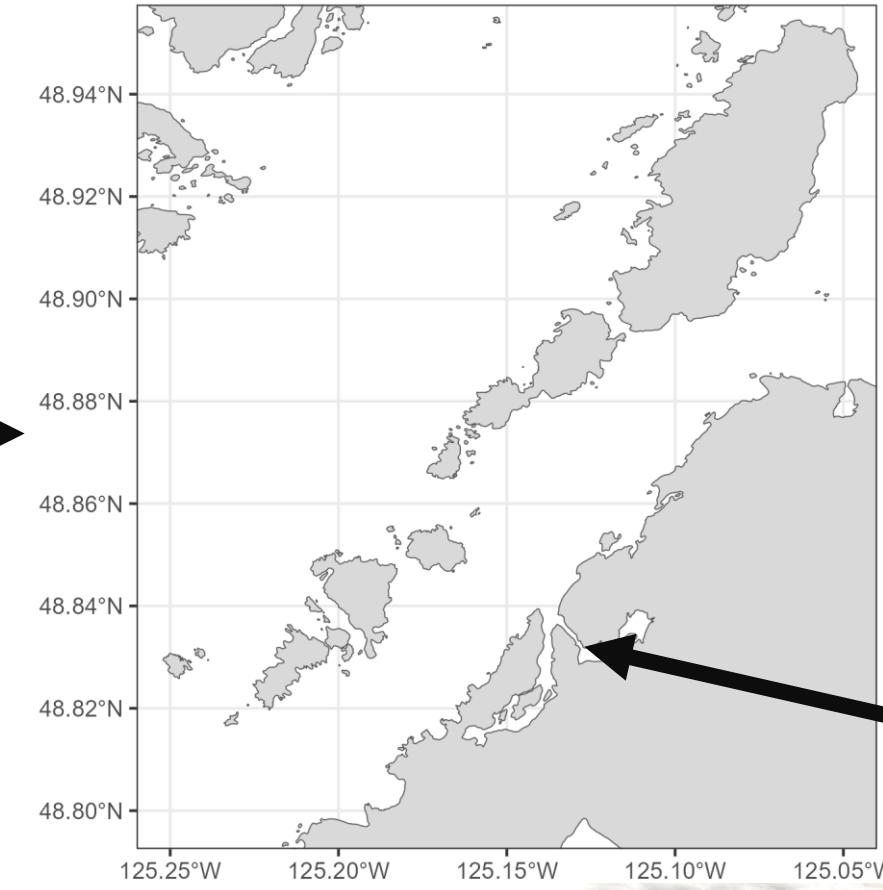


# Higher-resolution for local maps

**Even higher resolution is available with shapefiles from Hakai**



Zoom in  
→



# Sneak peak: satellite maps in R

**Even higher resolution with satellite maps**

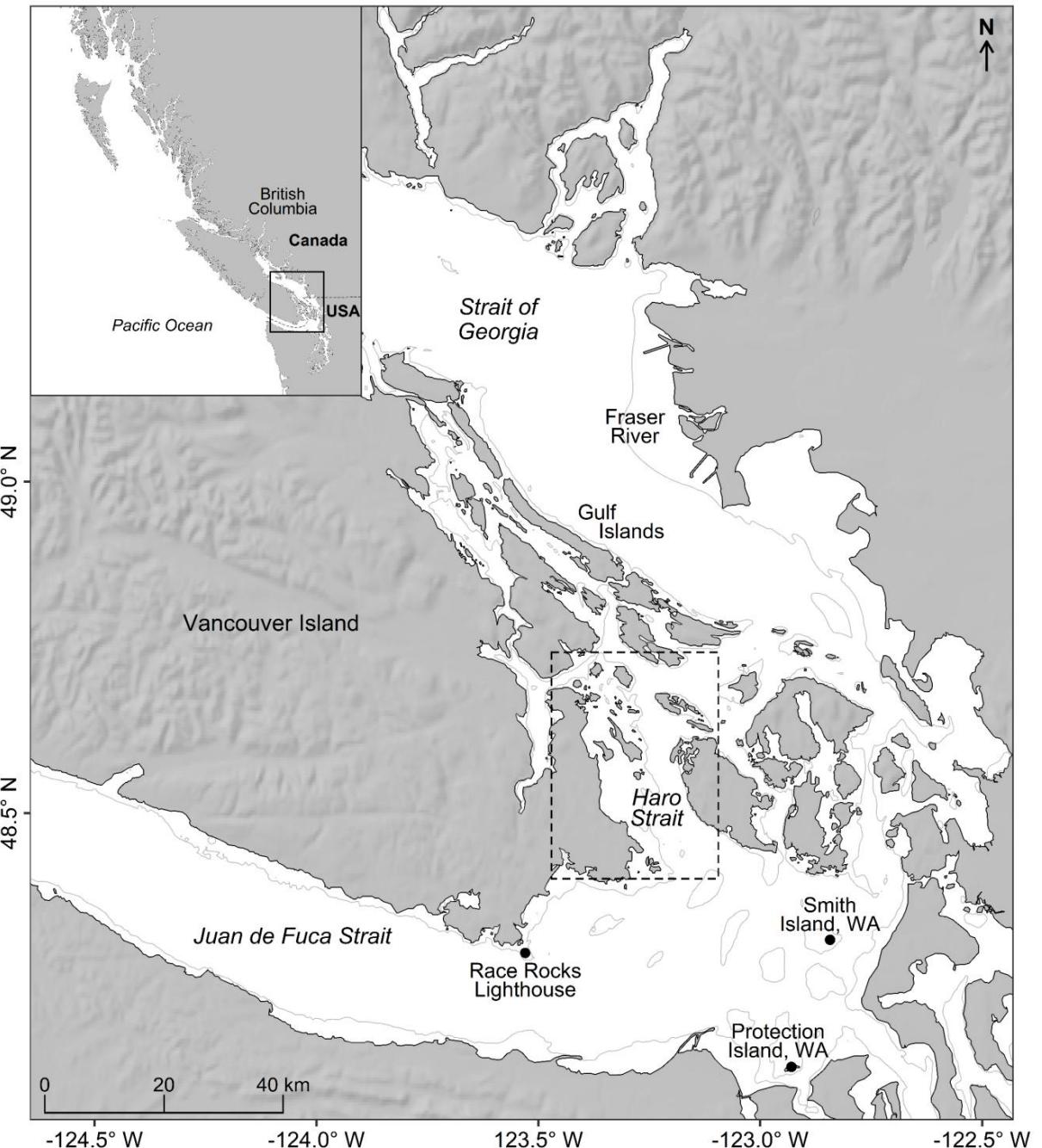


Zoom in  
→



# Study area maps

**Goal:** show sampling locations and important geographic/biological features

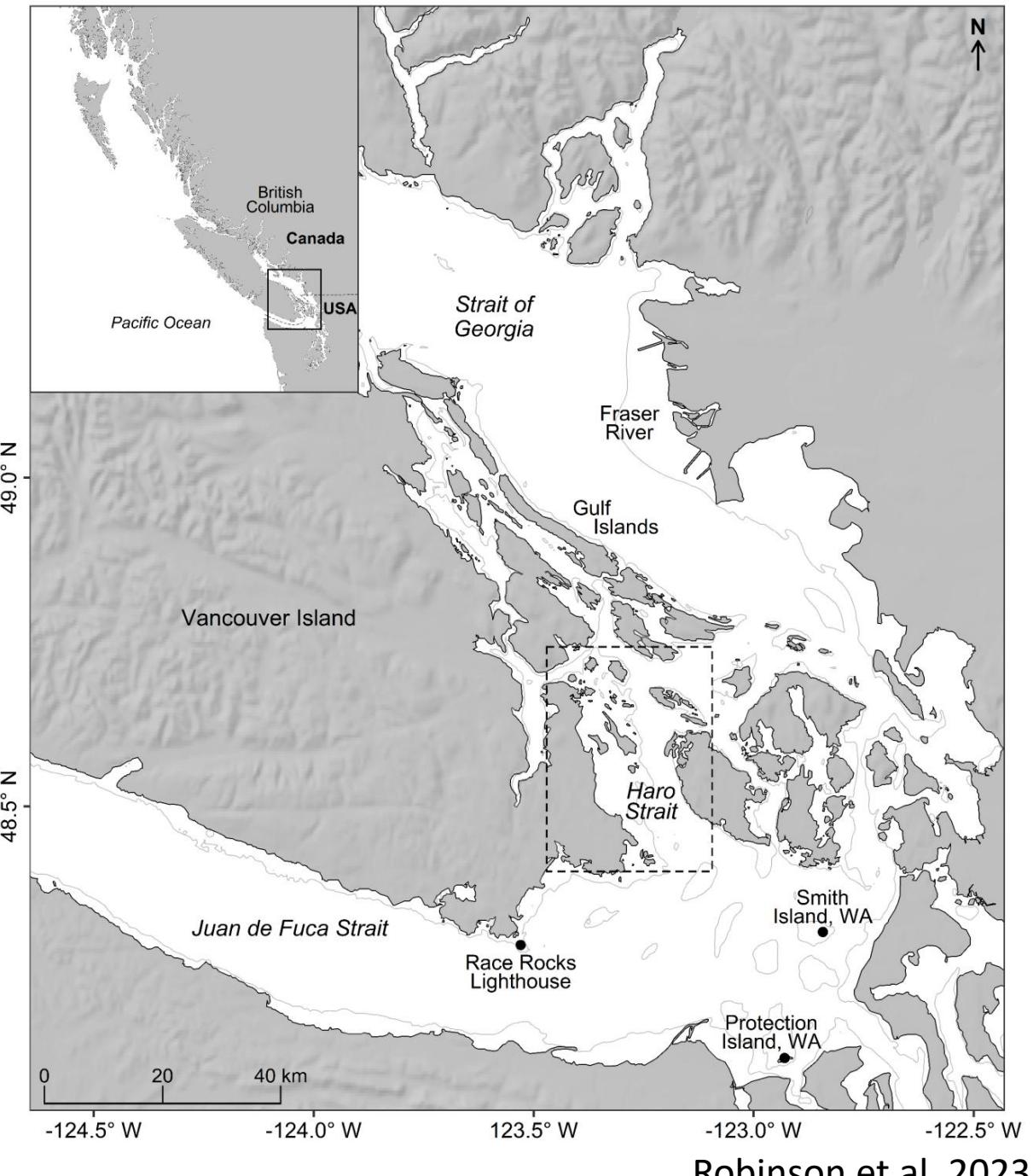


# Study area maps

**Goal:** show sampling locations and important geographic/biological features

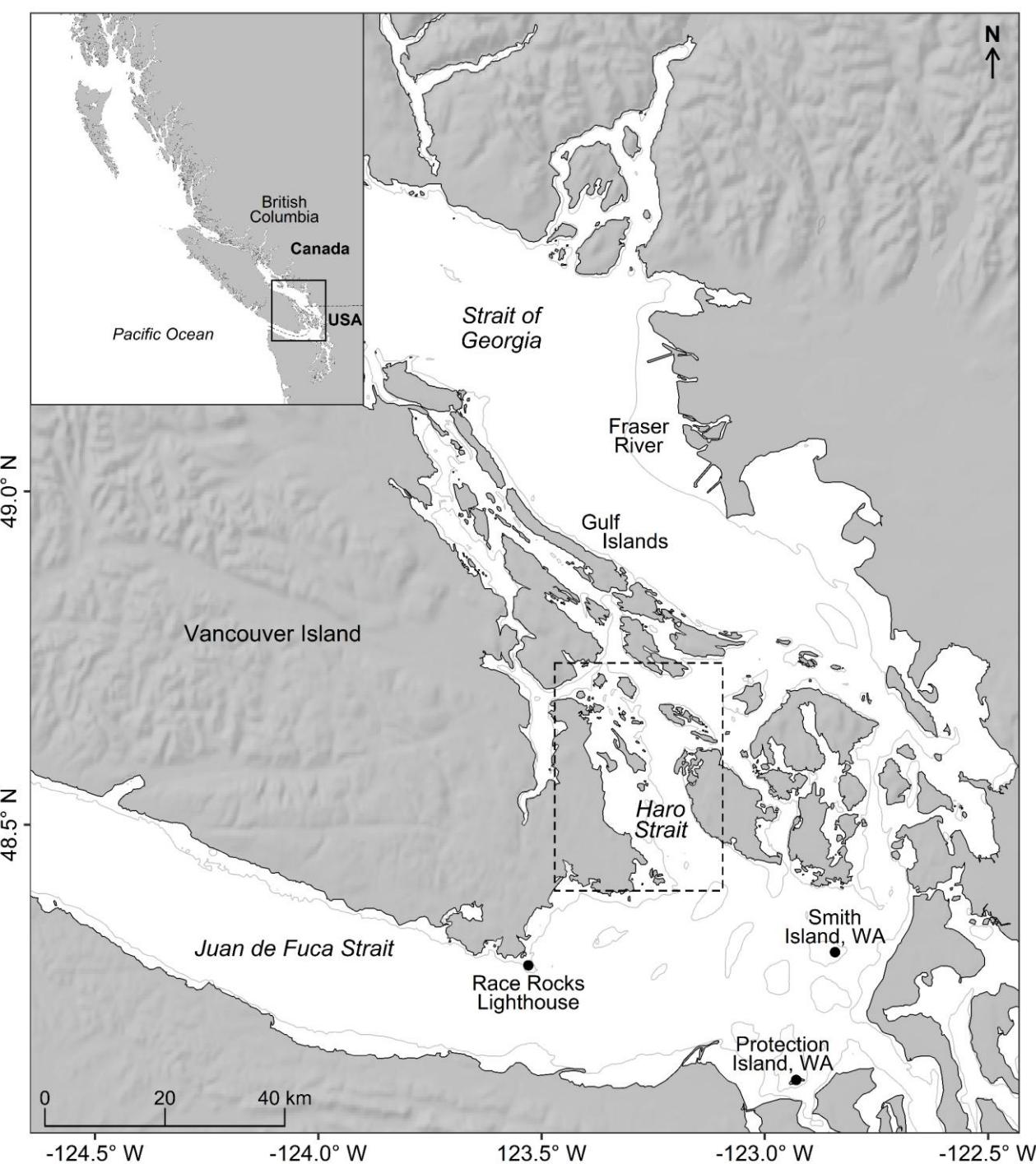
Often greyscale for simplicity

Include inset map showing larger geographic region



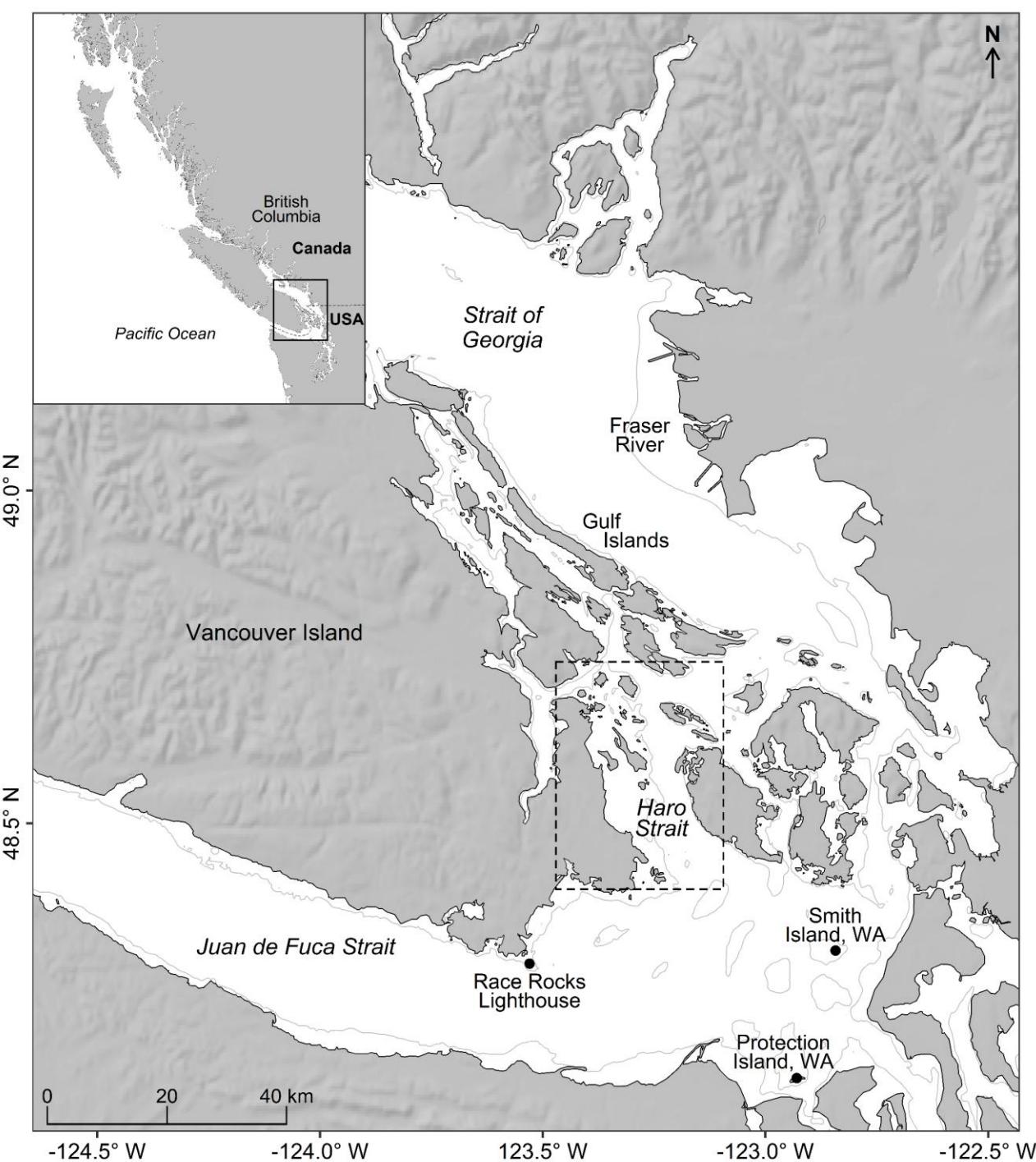
# Important parts

**Good spatial  
resolution is key**



# Important parts

**What do you  
think is  
important?**

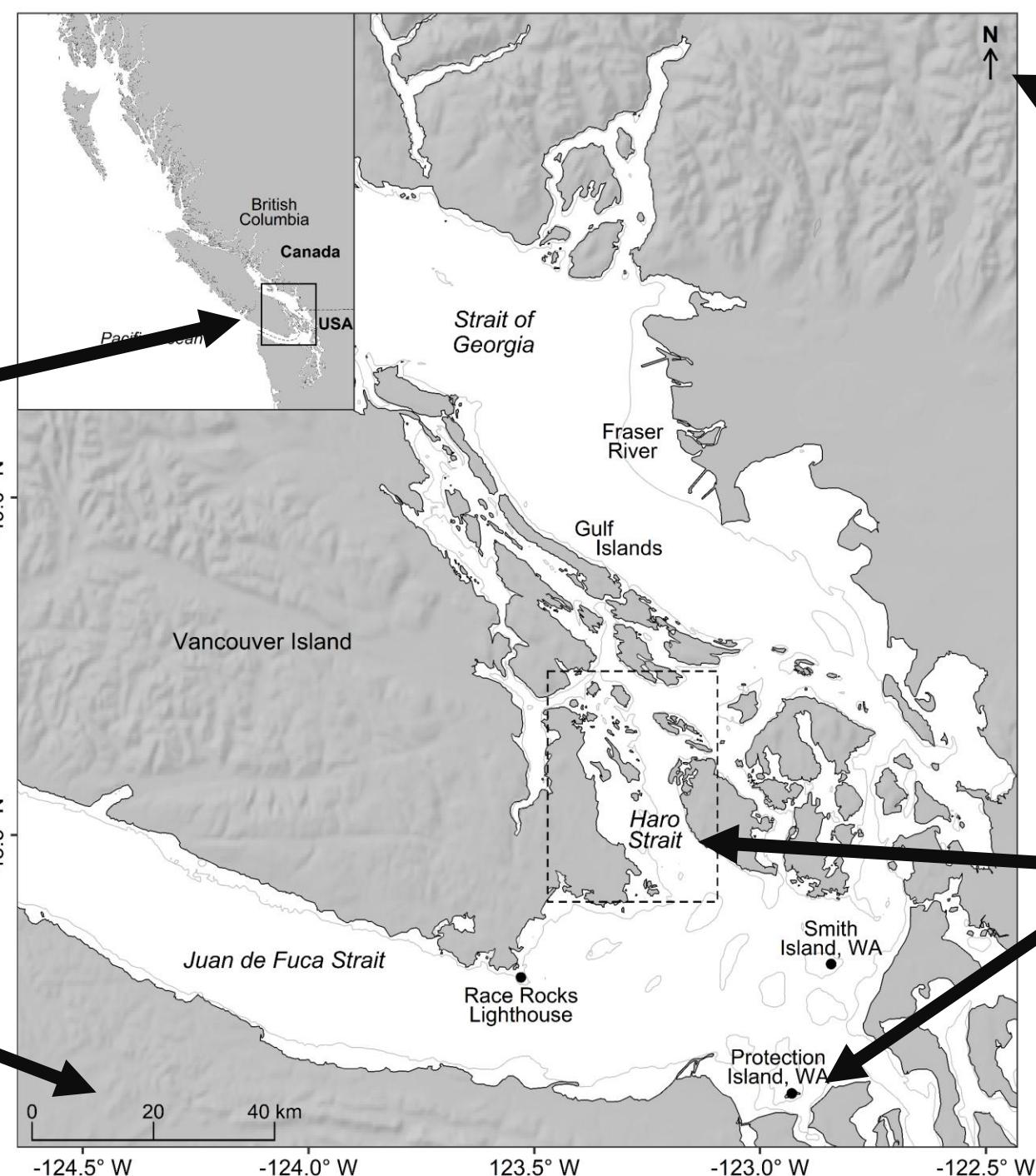


# Important parts

Inset map showing where the study area is

Study area denoted by rectangle

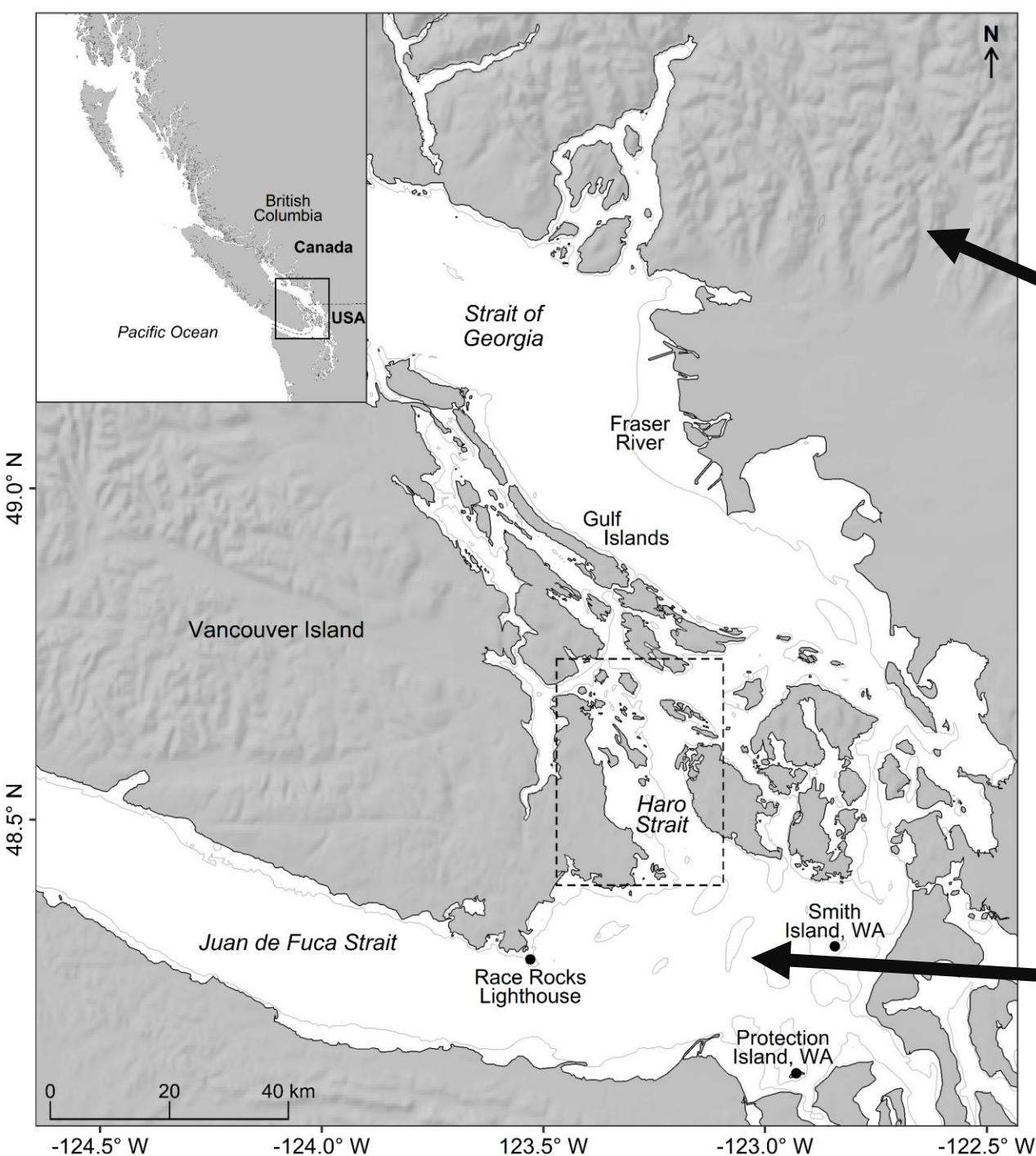
Scale bar



North arrow

Important sites labelled

# Extra pieces



Hillshading to illustrate topography

50 metre depth contours

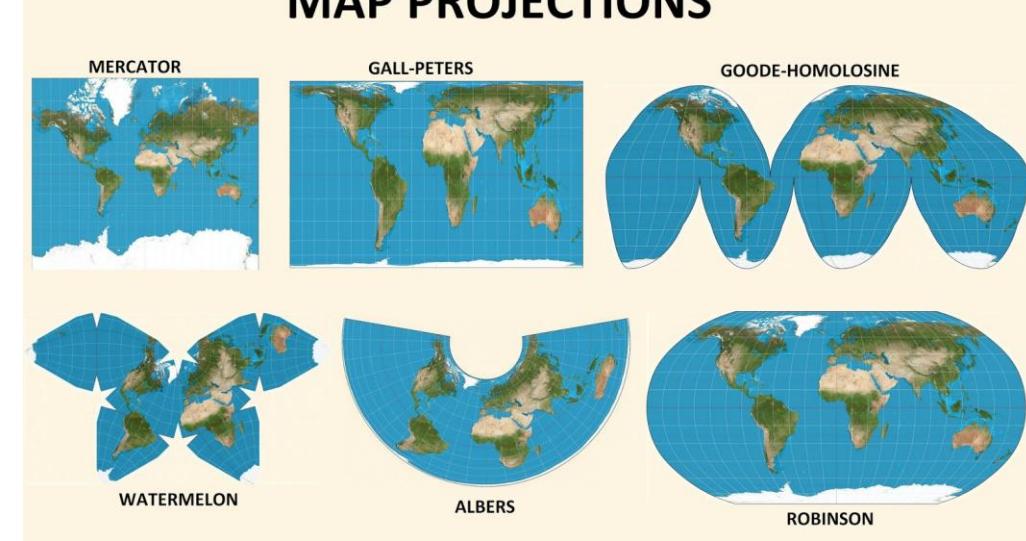
# Projections

## What projection to use?

## What is a projection?

Projections are used to map the Earth's 3D surface on a 2D plane (the map)

Either in degrees (e.g., WGS 84) or metres/kilometres (e.g., BC Albers, UTM Zone 10)



# Projections

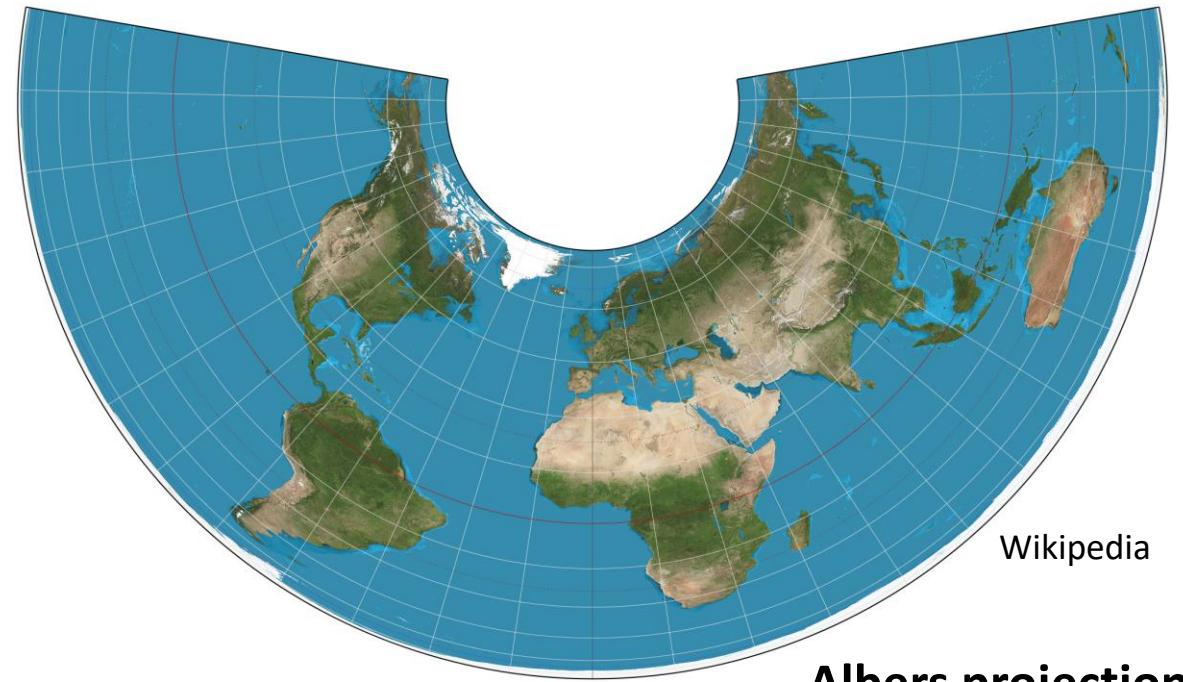
## What projection to use?

Projections distort large parts of the Earth's surface – **need to choose the right projection for your map**

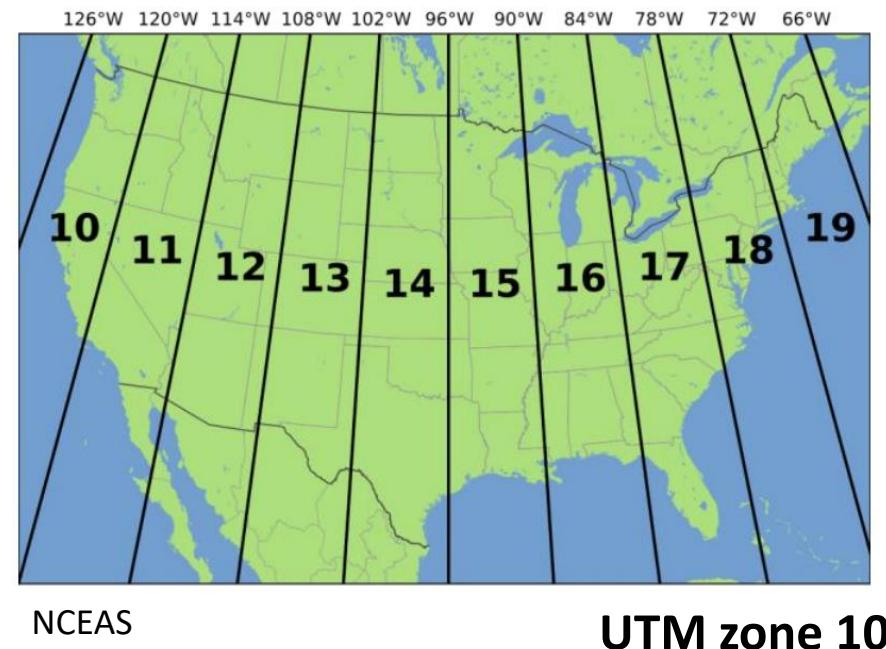
In BC, we want to use **BC Albers**

**BC Albers:** “equal area projection”

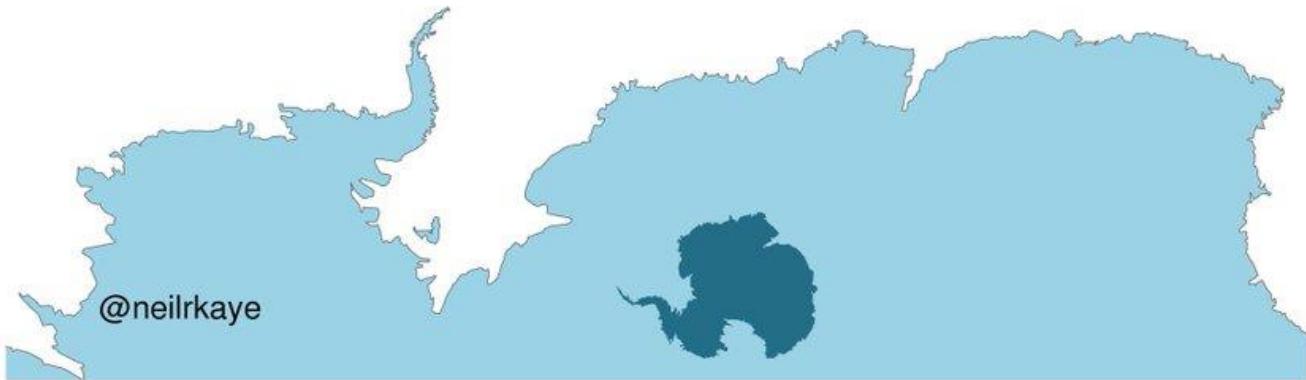
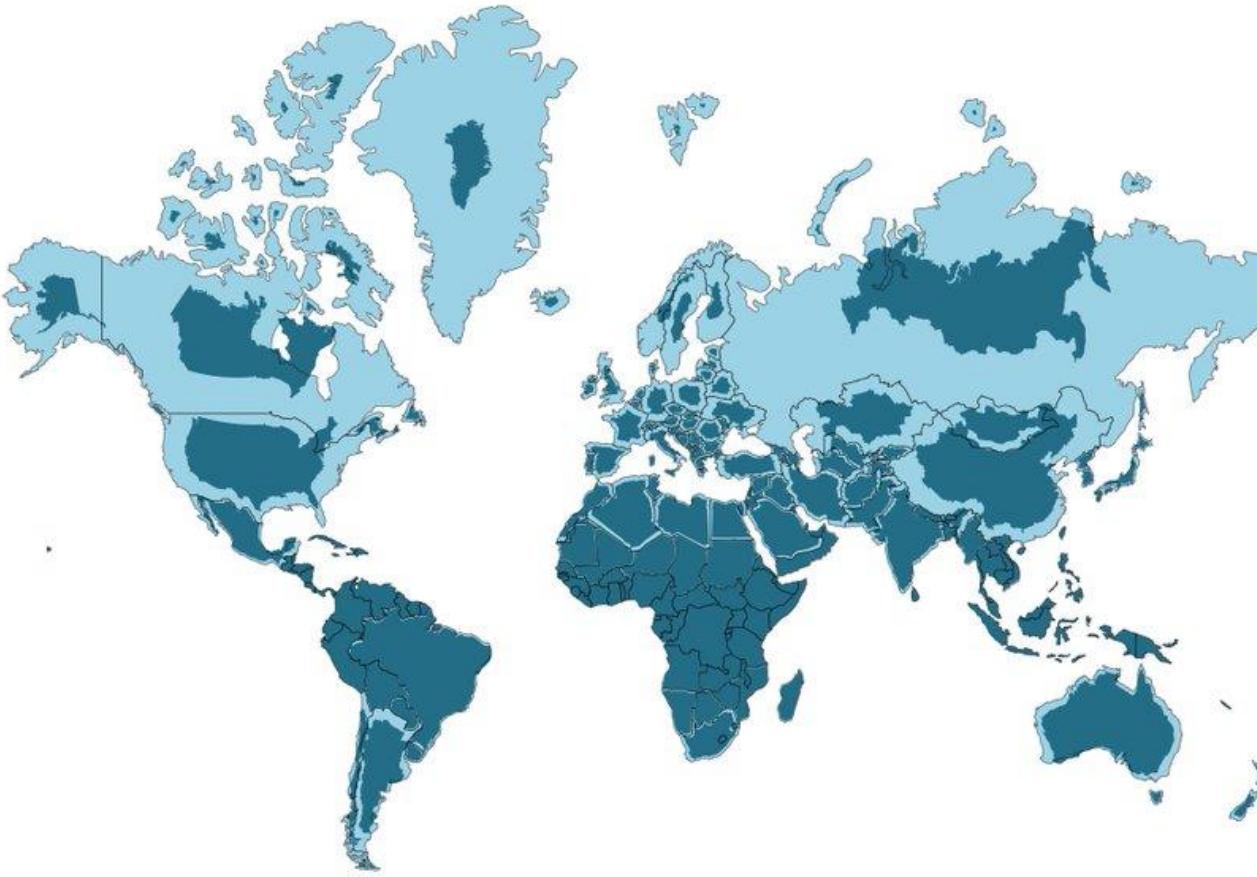
**Northings and Eastings  
(metres/kilometres) instead of  
latitude/longitude (degrees)**



Albers projection



World Mercator projection with true country size and shape added



@neilrkaye

# World Mercator projection with country going to true size

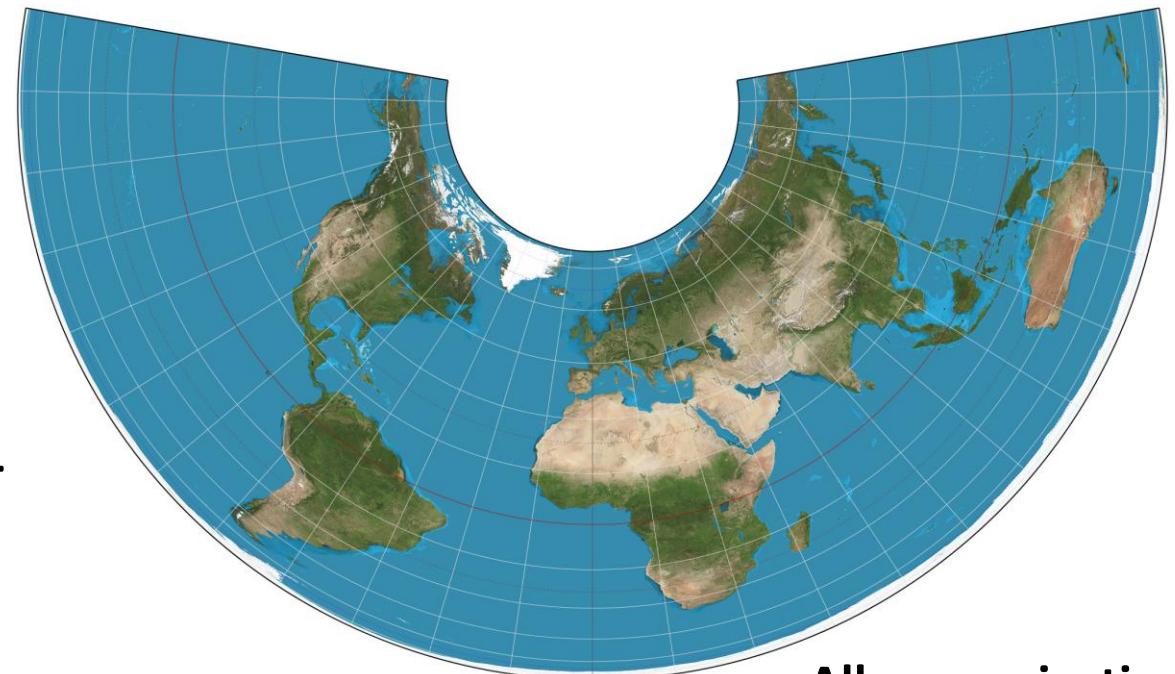


@neilrkaye

# Projections

What projection to use?

In this workshop, we will use WGS84 for simplicity



Albers projection

To transform to another projection:

```
albers <- st_transform(wgs), crs = st_crs(3005)
```



UTM zone 10

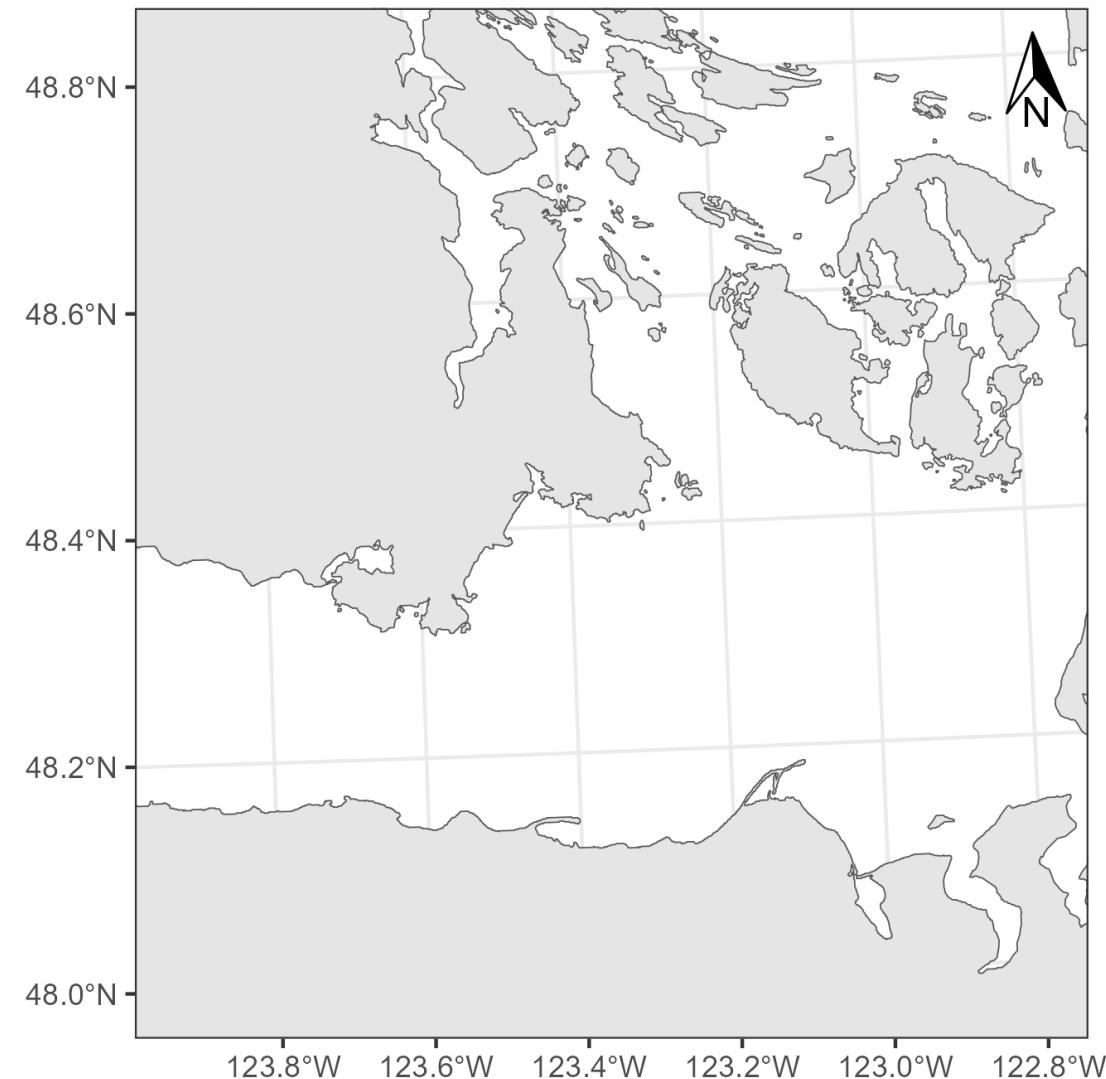
# Projections

```
albers <- st_transform(wgs), crs = st_crs(3005)
```

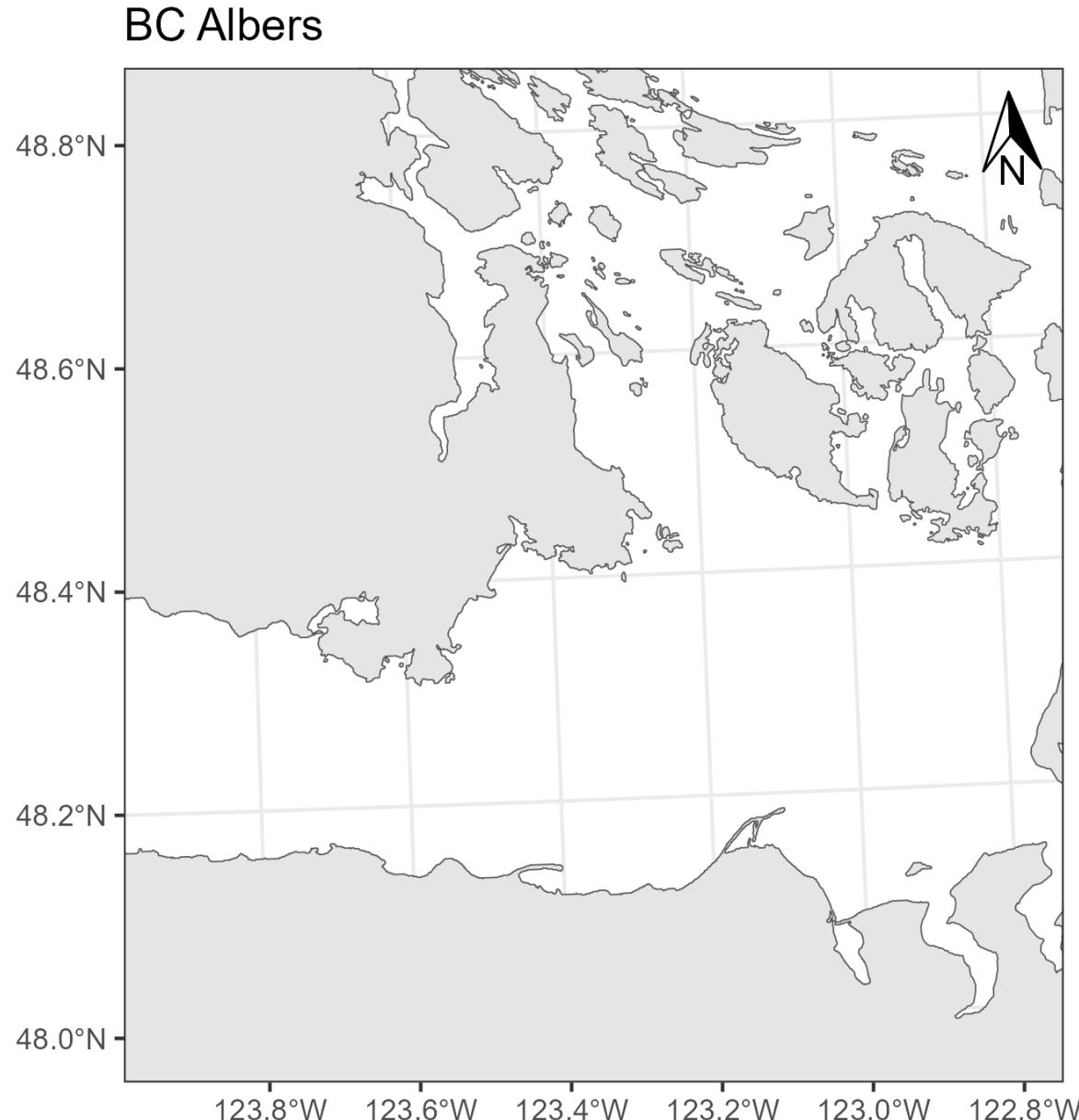
See 00-study-area-map.R, section 4 for easy code for setting the bounding box of maps in a projection that uses metres or kilometres

We are used to thinking in longitude/latitude – it can be difficult to define bounding boxes in metres

BC Albers



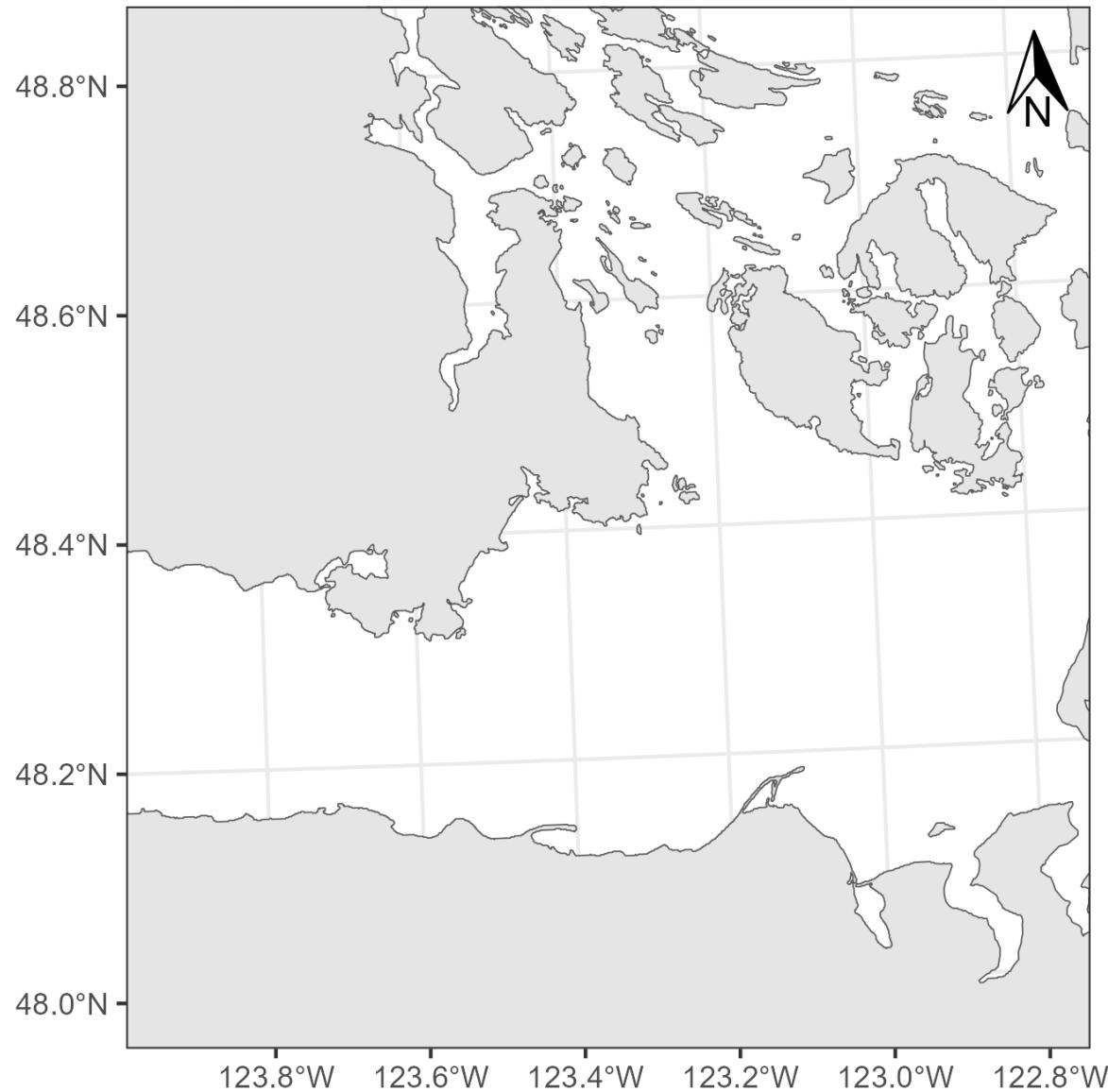
# BC Albers study area map



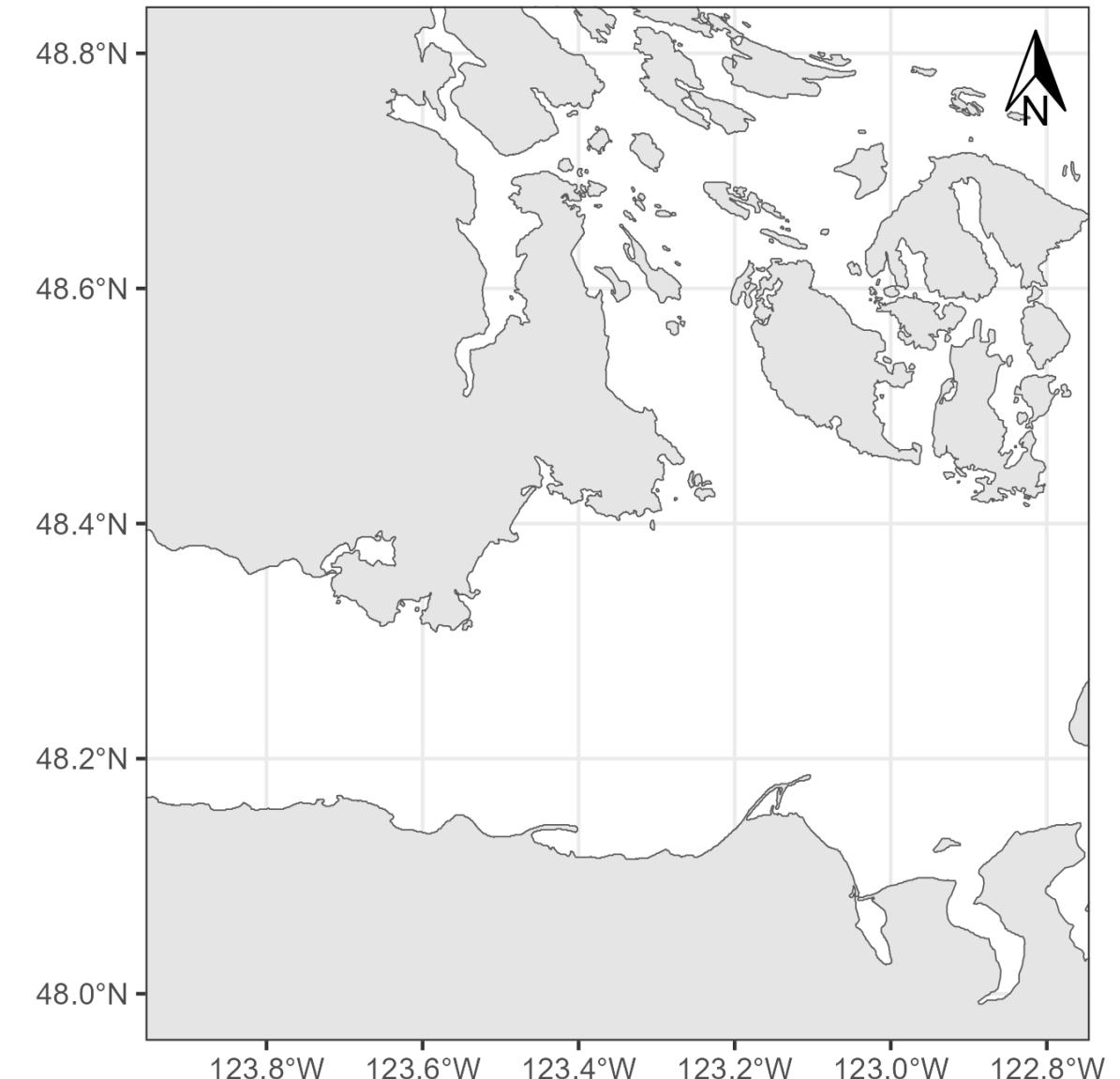
North arrow points  
slightly to the left

# BC Albers study area map – slightly different

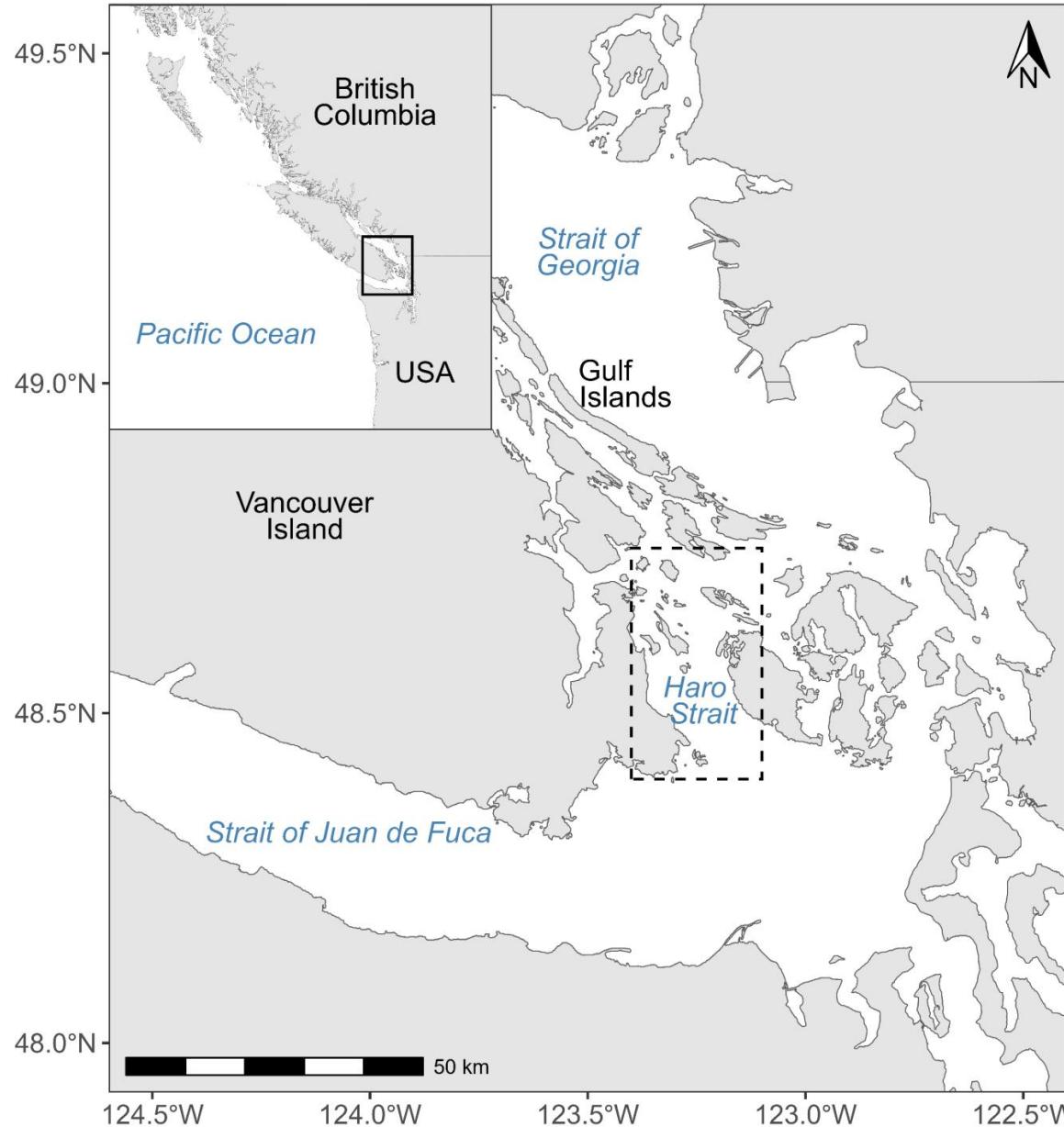
BC Albers



WGS 1984



# Simple study area map



# Load basemap

```
library(ggplot2)
library(sf)

bc.coast <- read_sf("data/bc-coast.shp")
```

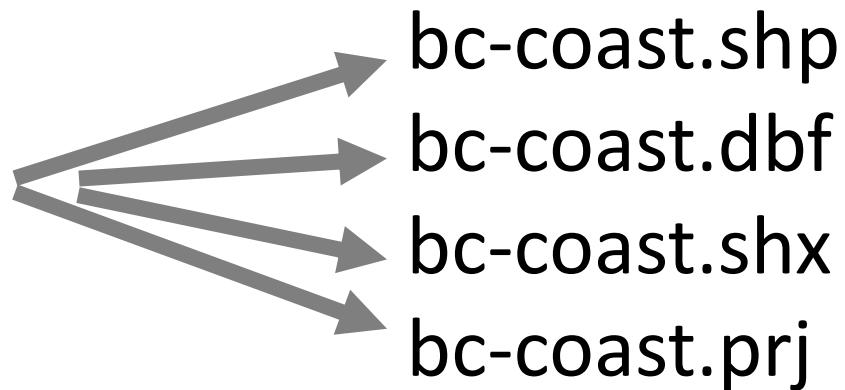
# Load basemap

```
bc.coast <- read_sf("data/bc-coast.shp")
```

A **shapefile** contains vector spatial data (e.g., points, lines, polygons).

A shapefile is NOT just one file!

bc-coast.shp



When sharing shapefiles,  
remember you need all 4 files!

# What projection?

```
library(ggplot2)
library(sf)

bc.coast <- read_sf("data/bc-coast.shp")
st_crs("data/bc-coast.shp")
```

`st_crs()` – what projection is the shapefile in?

```
> st_crs(bc.coast)
Coordinate Reference System:
  User input: WGS 84
  wkt:
GEOGCRS["WGS 84",
  DATUM["World Geodetic System 1984",
    ELLIPSOID["WGS 84",6378137,298.257223563,
      LENGTHUNIT["metre",1]]],
  PRIMEM["Greenwich",0,
    ANGLEUNIT["degree",0.0174532925199433]],
  CS[ellipsoidal,2],
  AXIS["latitude",north,
    ORDER[1],
    ANGLEUNIT["degree",0.0174532925199433]],
  AXIS["longitude",east,
    ORDER[2],
    ANGLEUNIT["degree",0.0174532925199433]],
  ID["EPSG",4326]]
```

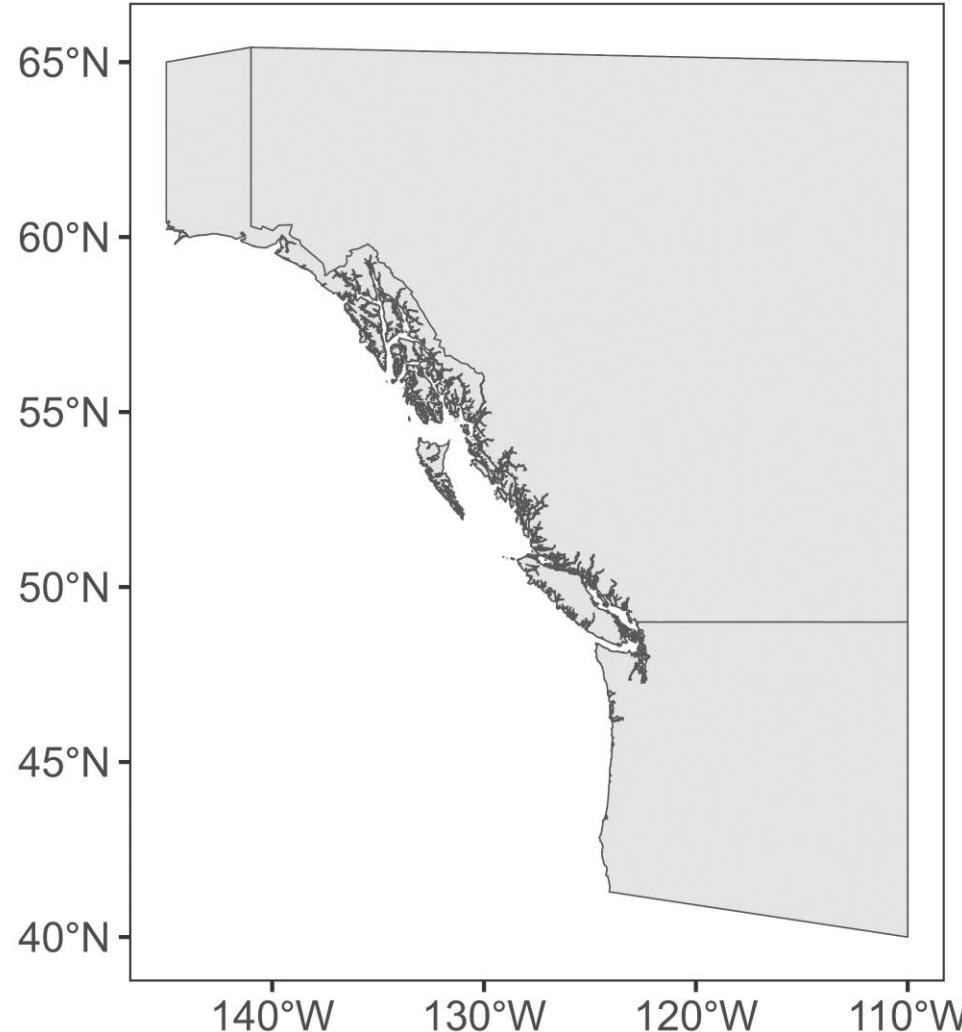
EPSG 4326: WGS 1984, a common global projection

We use EPSG numbers to refer to different projections/CRS

**CRS: coordinate reference system**

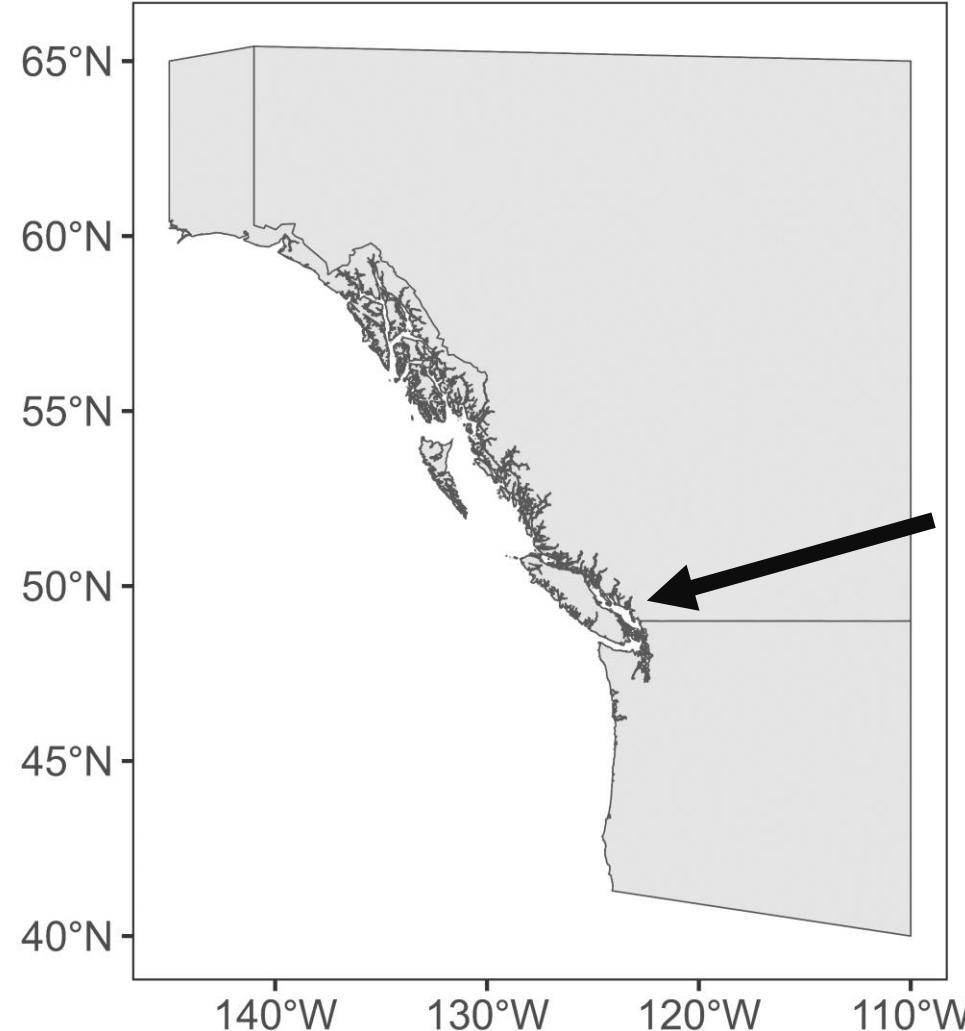
# Plot basemap

```
ggplot() +  
  geom_sf(data = bc.coast)
```



# Plot basemap

```
ggplot() +  
  geom_sf(data = bc.coast)
```

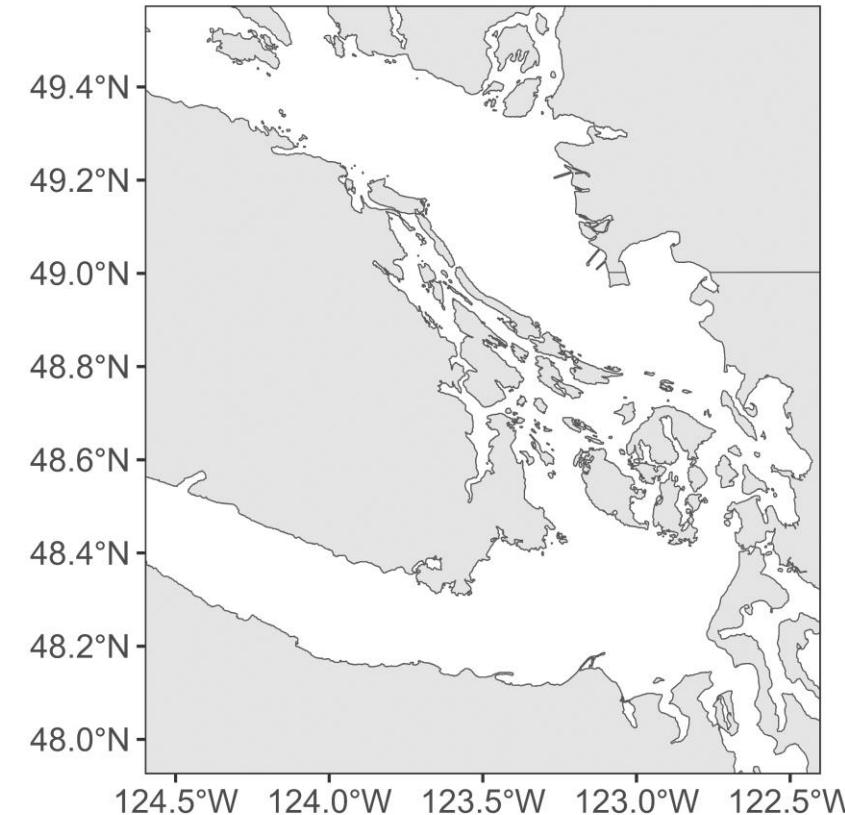
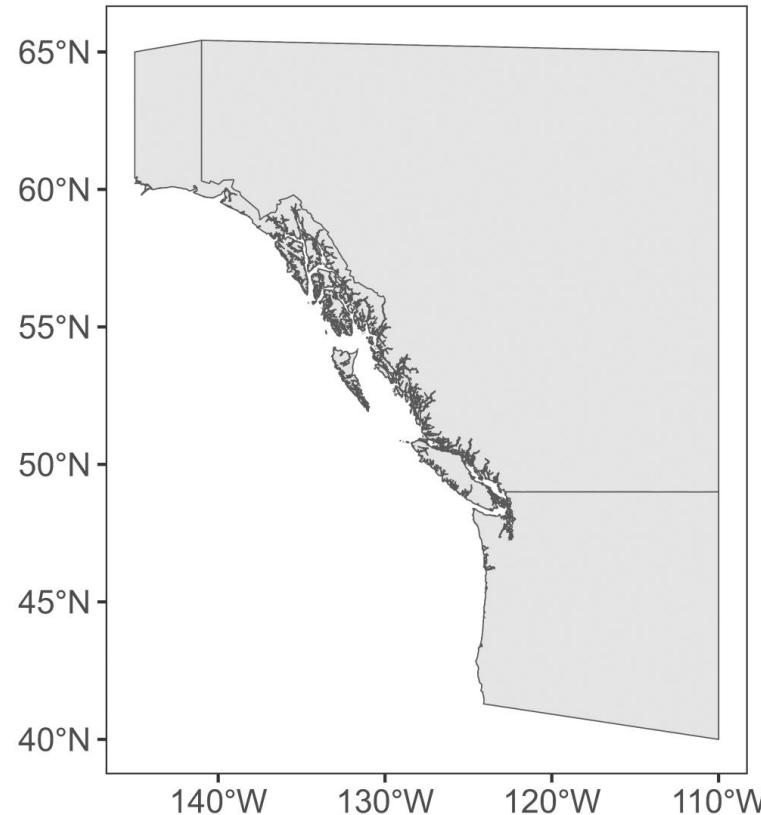


Need to zoom in on  
area of interest

# Plot basemap and crop to area of interest

```
ggplot() +  
  geom_sf(data = bc.coast) +  
  
  coord_sf(xlim = c(-124.5, -122.5),  
            ylim = c(48, 49.5))
```

Crop bc.coast to our area of interest



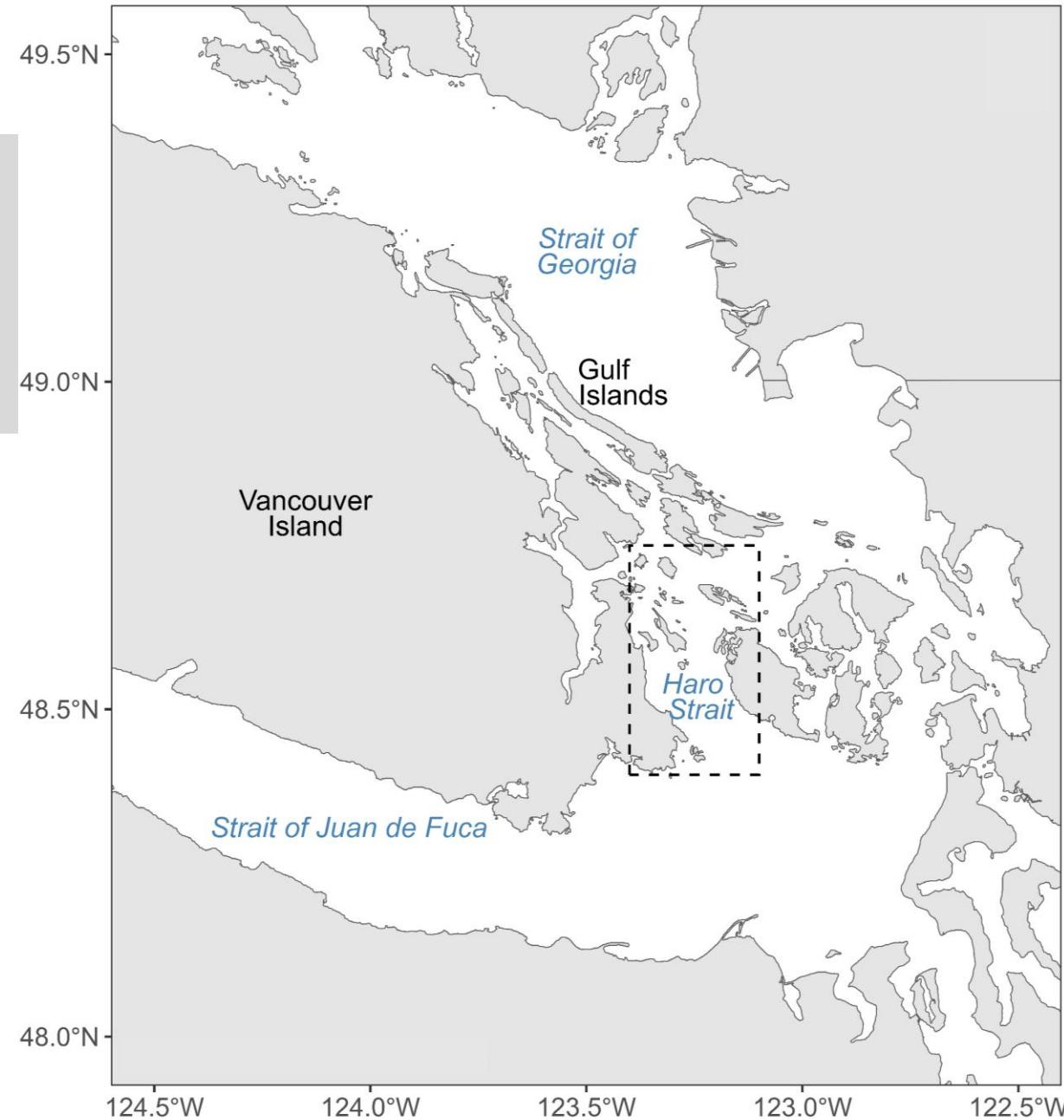
# Label areas of interest

```
# add scalebar (ggspatial package)  
  
annotate(geom = "text",  
        x = -124, y = 48.3,  
        label = "Strait of Juan de Fuca")
```

## Personal preferences

Water bodies: *blue italics*

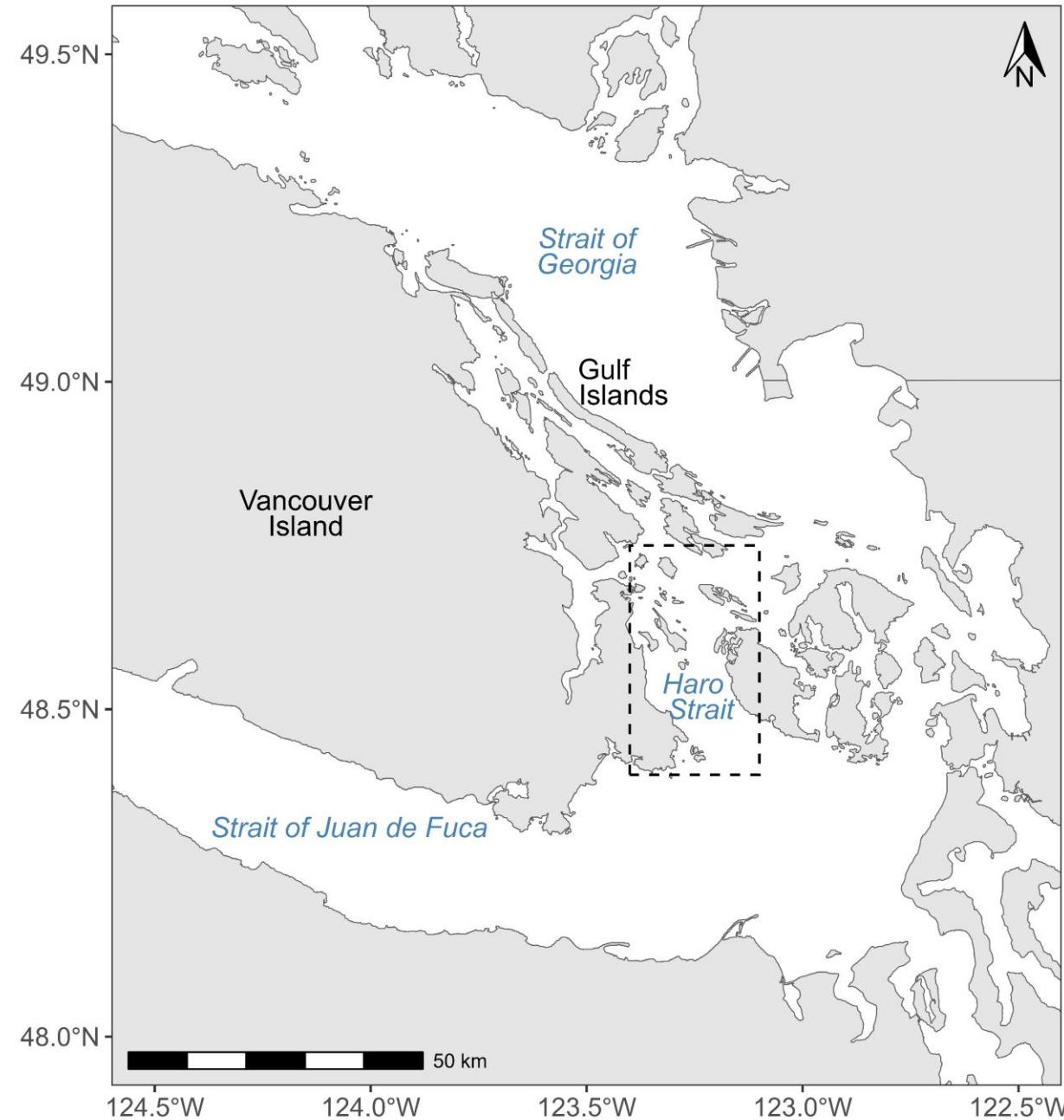
Land features: black



# Add scalebar, north arrow

```
# add scalebar (ggspatial package)  
  
annotation_scale(width_hint = 0.35) +  
  
annotation_north_arrow(location = "tr")
```

Note: the north arrow will point straight up for some projections but not others



# Add inset map

```
# inset map
inset <- ggplot() +
  geom_sf(data = bc.coast) +
  annotate(geom = "rect",
          xmin, xmax, ymin, ymax)
```

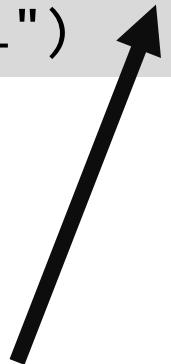
Add rectangle indicating the area  
of the larger map

Same coordinates as the coord\_sf()  
call from the main map

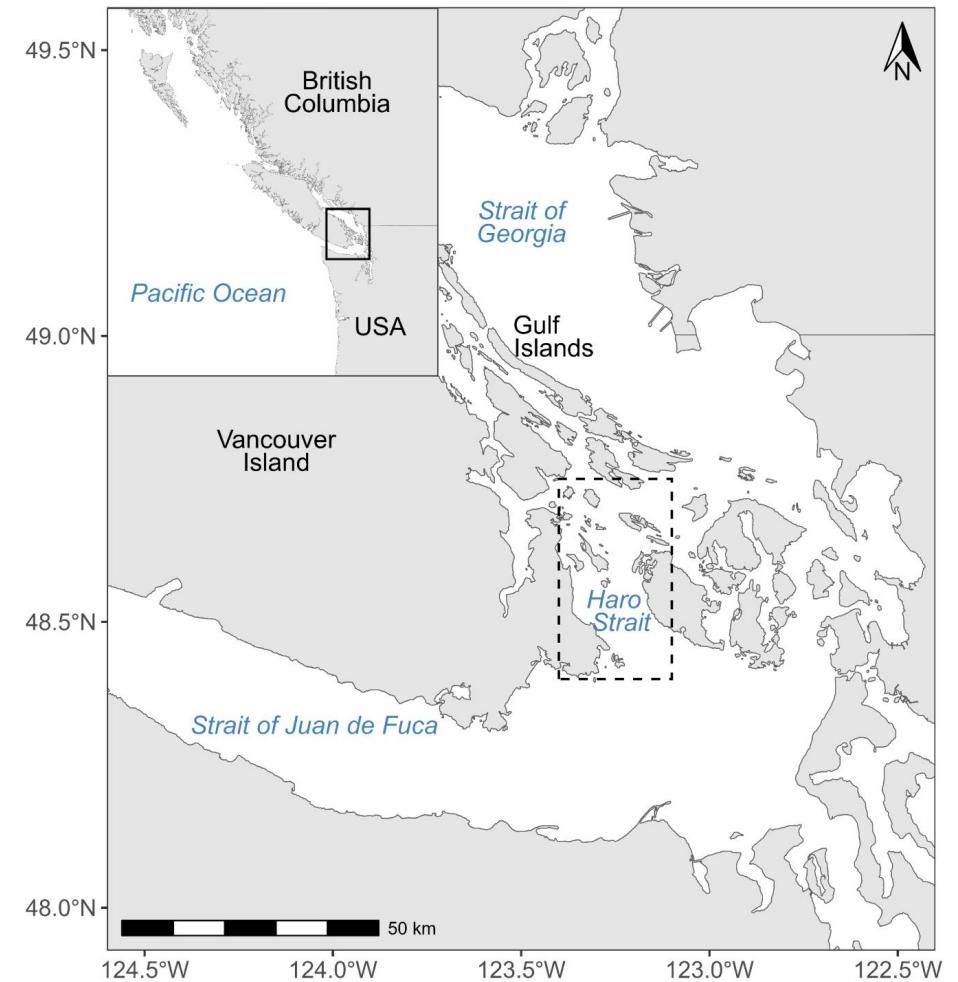


# Add inset map

```
# add inset map  
  
study.area +  
  inset_element(inset,  
  left = -0.007, bottom = 0.6,  
  right = 0.4, top = 1.008,  
  align_to = "panel")
```

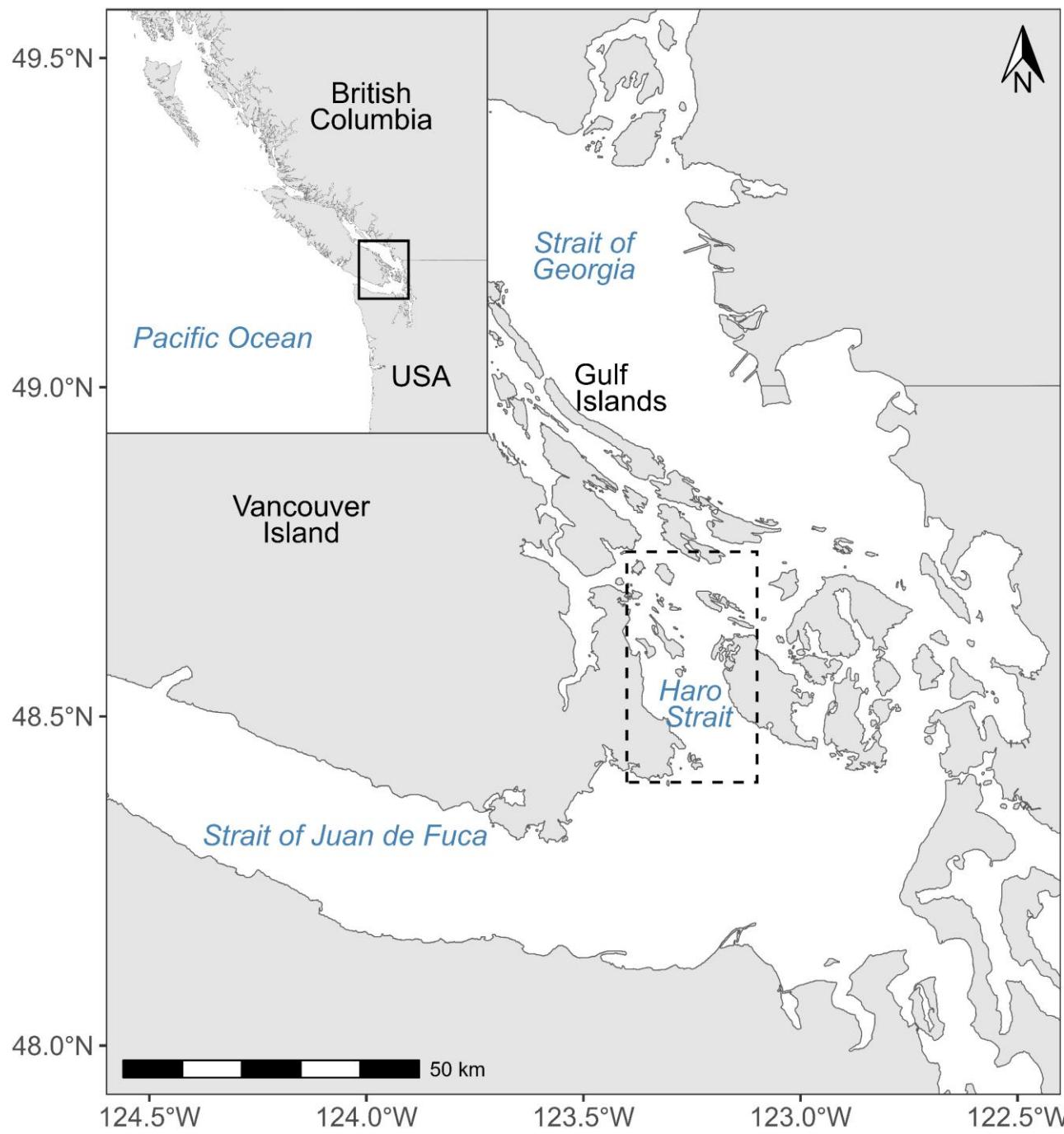


left, right, bottom, top:  
Determine by trial and  
error so that inset map is  
desired size and location



Top-left inset, start with:  
left = 0, bottom = 0.6, right = 0.4, top = 1

Top-right inset, start with:  
left = 0, bottom = 0.6, right = 0.6, top = 1

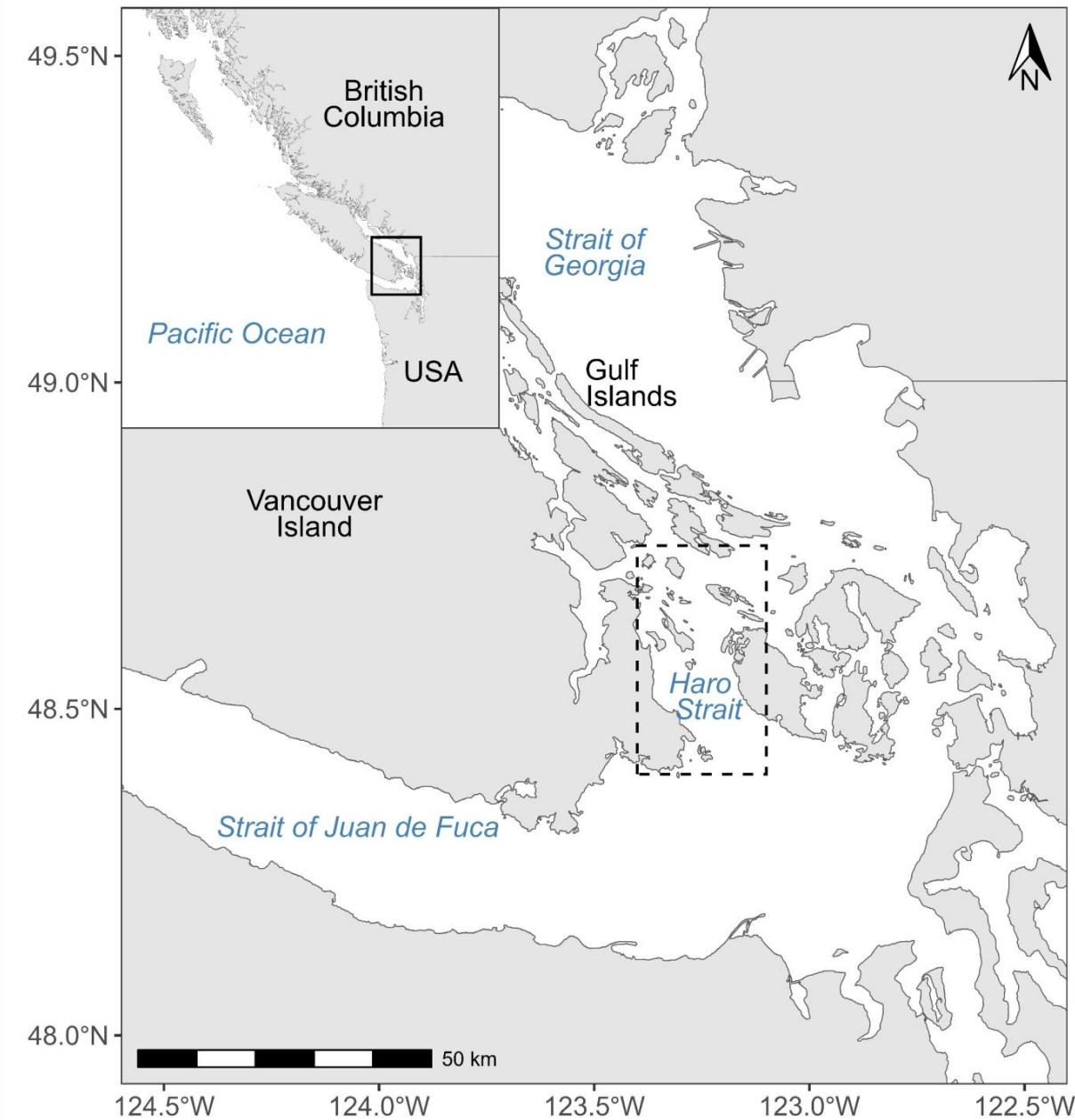


# Save the map

Print the map, then use `ggsave()`

```
study.area +  
  inset_element(inset,  
  left = -0.007, bottom = 0.6,  
  right = 0.4, top = 1.008,  
  align_to = "panel")  
  
ggsave("figures/study-area-map.PNG",  
  width = 17, height = 17,  
  units = "cm", dpi = 300)
```

Can increase resolution to `dpi = 1600`  
for very high quality images, if desired



# Where to find additional data layers?

Lots of open-source spatial data available

- BC Government Data Catalogue  
(especially terrestrial, freshwater)

<https://catalogue.data.gov.bc.ca/>

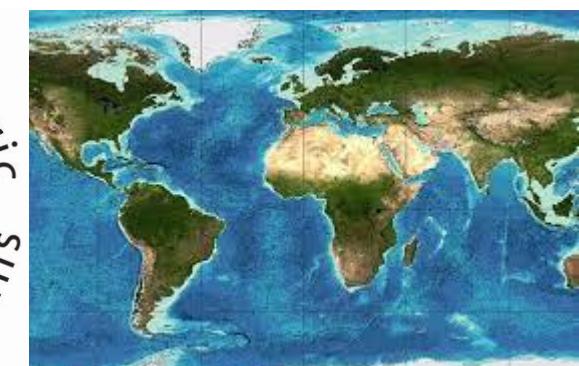
- PSF/UBC Marine Data Centre and  
Marine Reference Guide

<https://soggy2.zoology.ubc.ca/geonetwork/srv/eng/catalog.search#/home>

<https://gis.sogdatacentre.ca/sog-mrg/>

- GEBCO (bathymetry and topography)

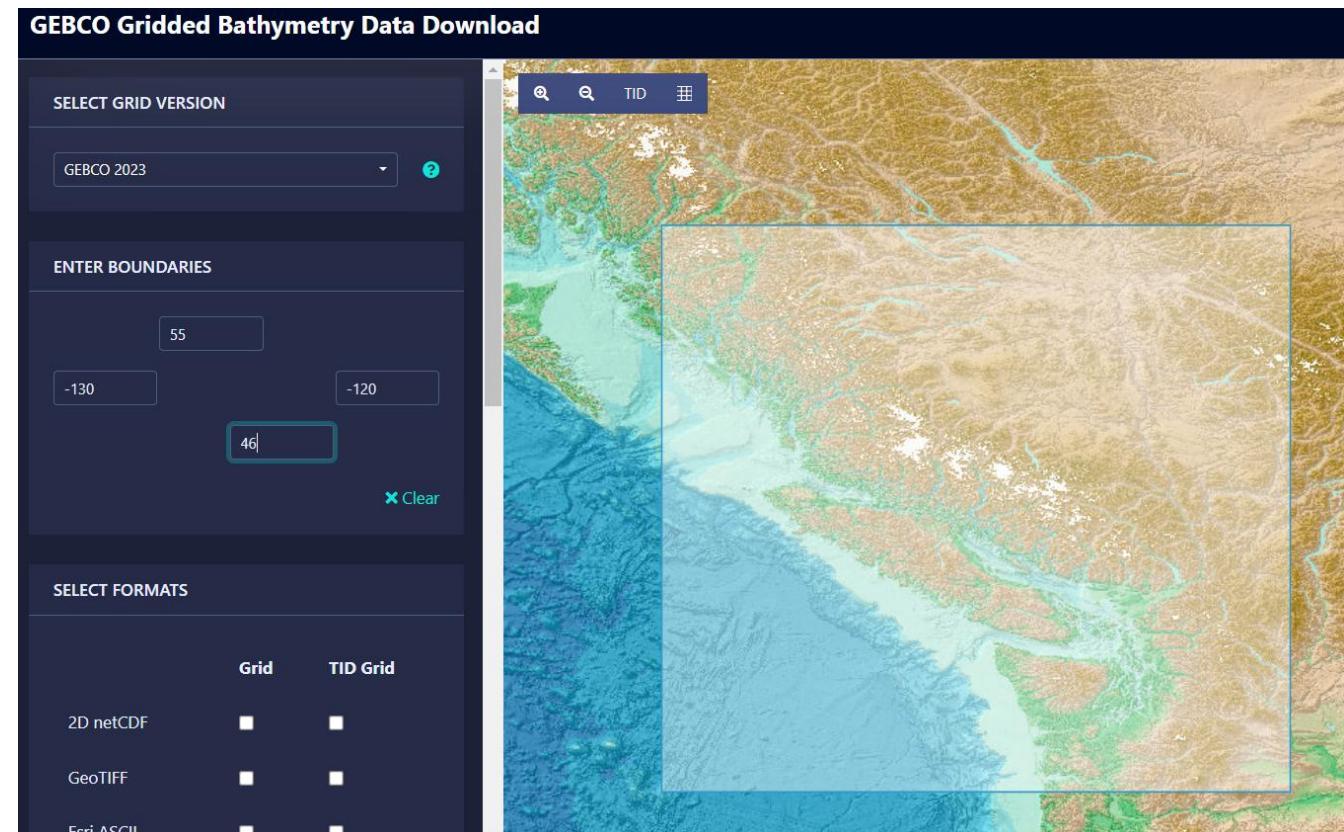
<https://www.gebco.net/>



# Add bathymetry and topography



- Using digital elevation model downloaded from GEBCO
- **Raster** of cells with a z-value indicating height below or above sea level
- **terra package: raster data**



<https://download.gebco.net/>

# GEBCO Gridded Bathymetry Data Download

SELECT GRID VERSION

GEBCO 2023 ?

ENTER BOUNDARIES

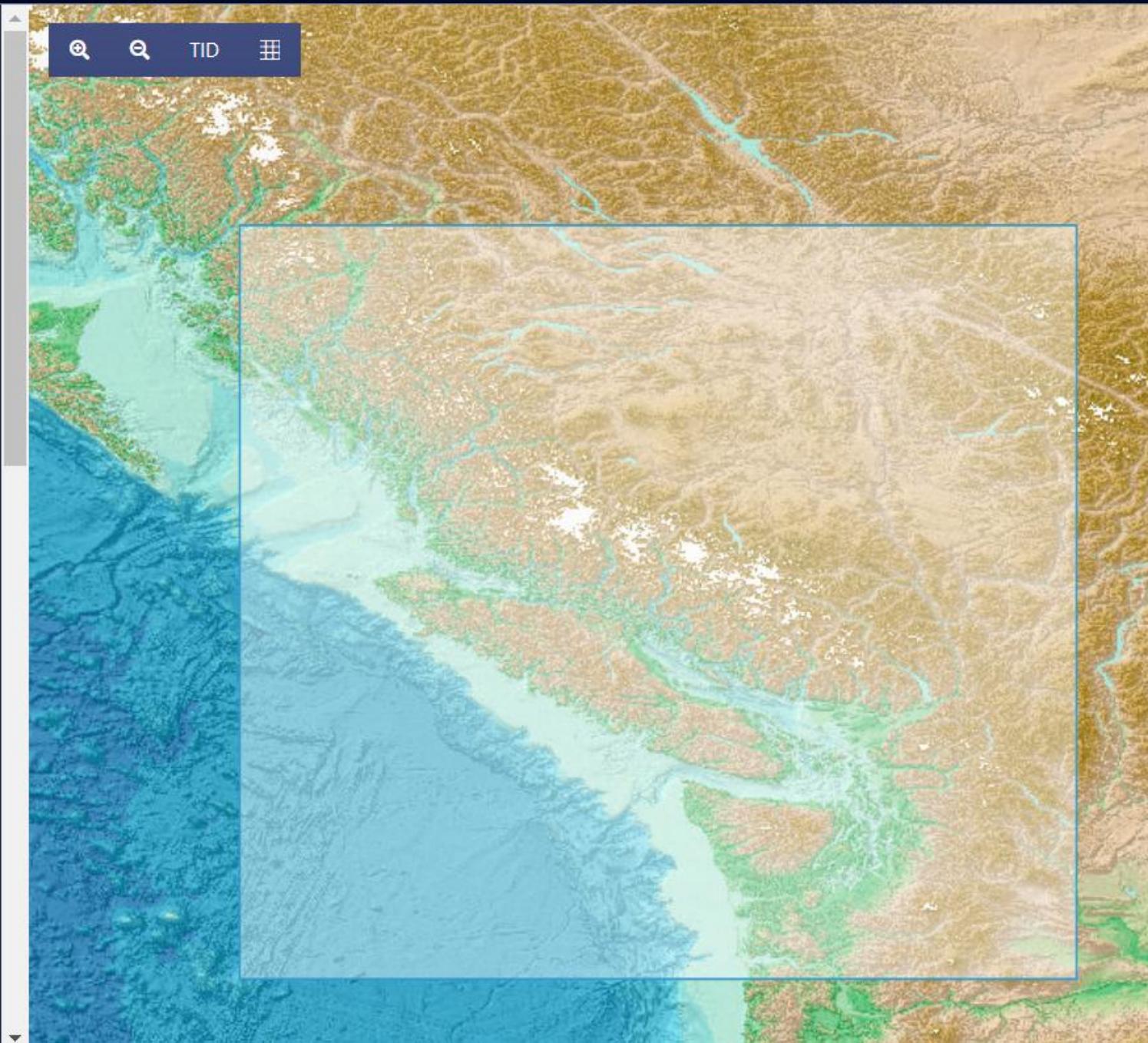
55

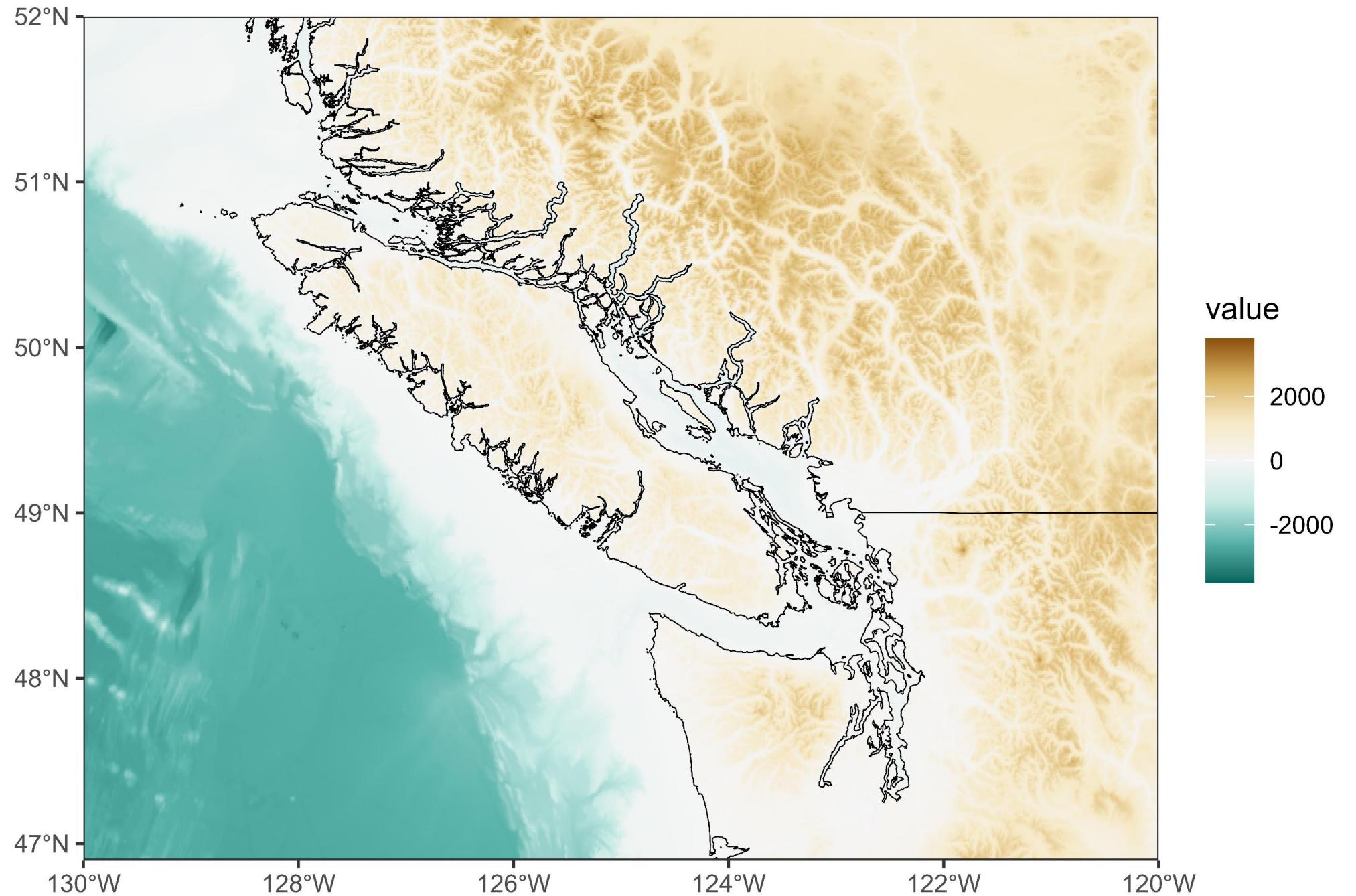
-130      -120

46 Clear

SELECT FORMATS

	Grid	TID Grid
2D netCDF	<input type="checkbox"/>	<input type="checkbox"/>
GeoTIFF	<input type="checkbox"/>	<input type="checkbox"/>
Esri ASCII	<input type="checkbox"/>	<input type="checkbox"/>





# Add bathymetry and topography

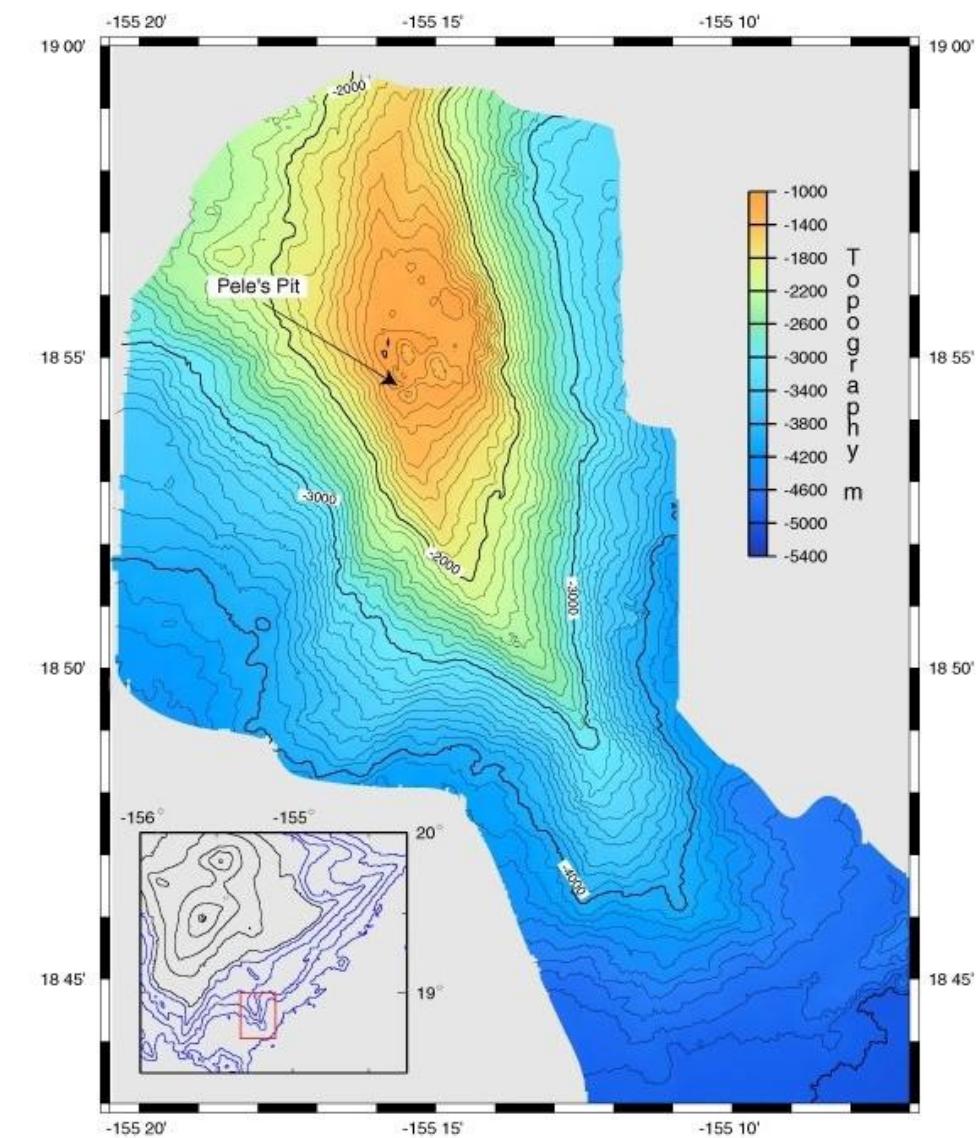
**terra** package: raster data

Plot rasters with **geom\_spatraster()** from **tidyterra** package

# Add bathymetry and topography

**Bathymetry layer:** depth of each grid cell

**Topography layer:** “hillshading” - the shadows cast by the digital elevation model, if the sun was shining from northwest at a  $45^\circ$  angle



Wikipedia

# Add bathymetry and topography

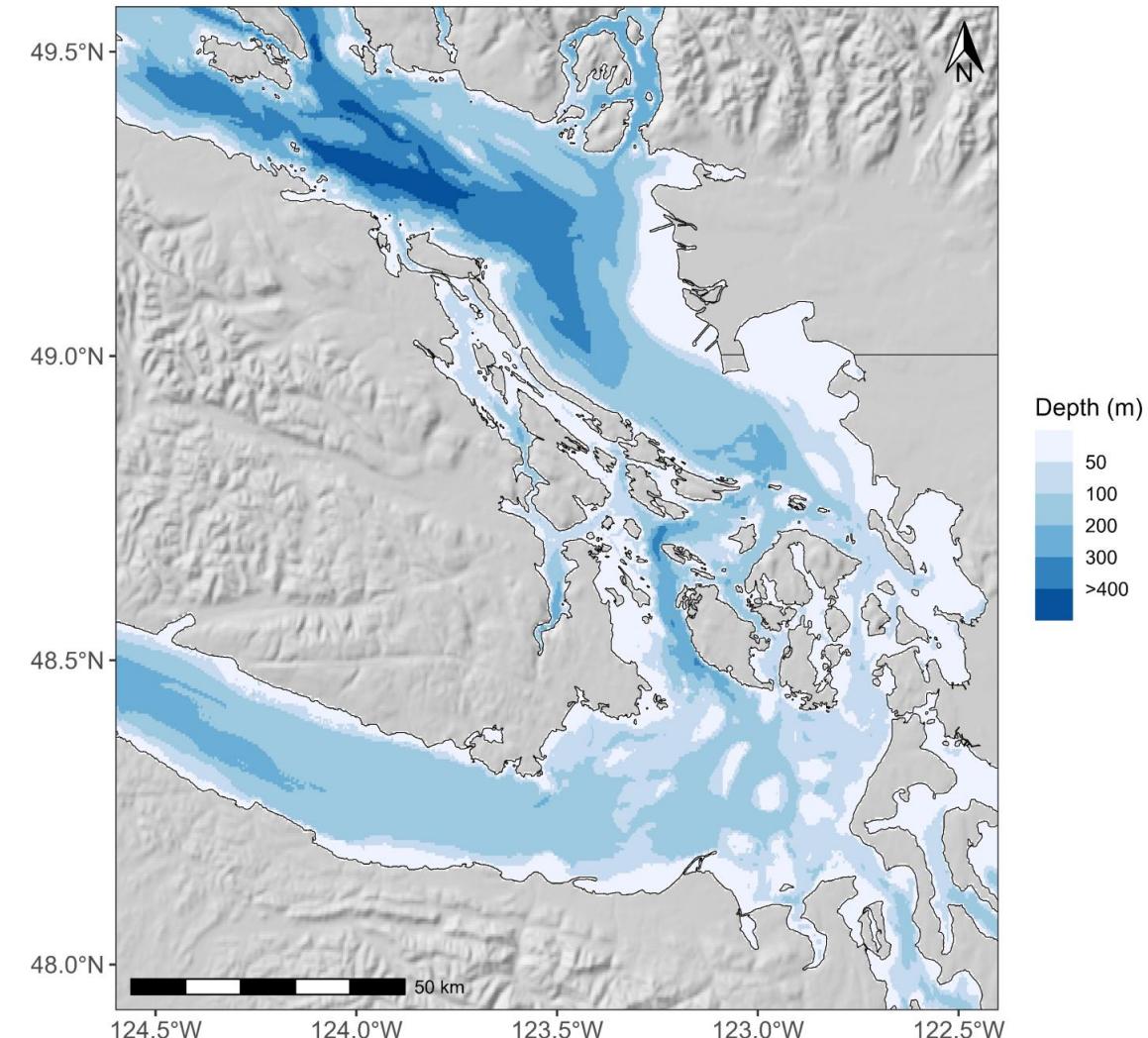
Bathymetry layer: depth of each grid cell, with land filtered out

Topography layer: “hillshading” - the shadows cast by the digital elevation model, if the sun was shining from northwest at a 45° angle

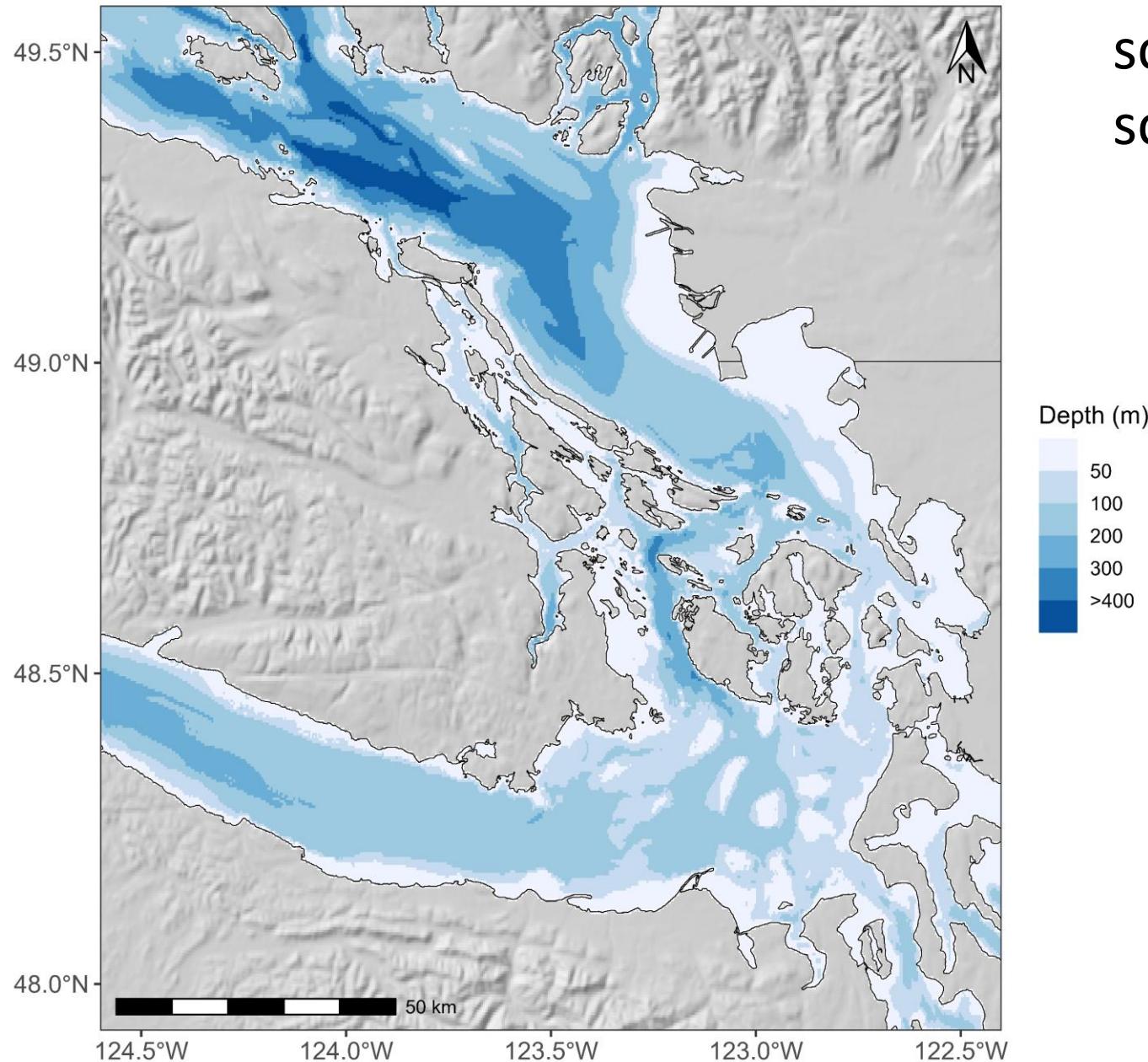


# Add bathymetry and topography

```
ggplot() +  
  geom_spatraster(data = hillshading,  
                   alpha = 0.7) +  
  
  geom_spatraster(data = bathymetry) +  
  
  geom_sf(bc.coast)
```



# Add bathymetry and topography



Full code:  
[scripts/bathymetry.R](#)  
[scripts/topography.R](#)

# Add external polygon and point data

**Rockfish conservation areas (RCAs):** marine protected areas with restricted fisheries to protect rockfishes



Hakai

# Add external polygon and point data

**Rockfish conservation areas (RCAs):** marine protected areas with restricted fisheries to protect rockfishes

**Question:** What is the distribution of RCAs relative to rockfish observations on iNaturalist around Victoria?

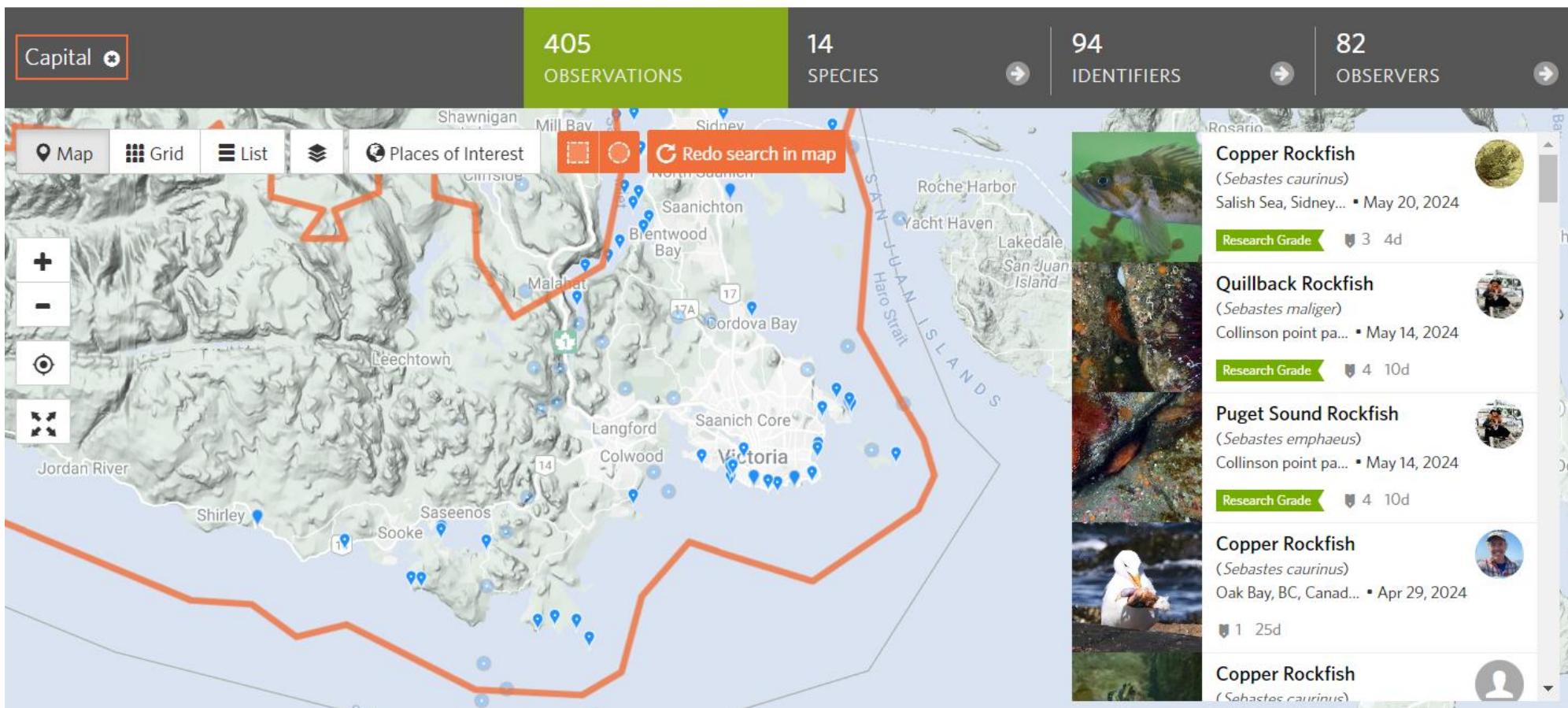


Hakai

# iNaturalist data

## Observations

 Rockfishes × Location Go Filters



[https://www.inaturalist.org/observations?place\\_id=49140&subview=map&taxon\\_id=47762](https://www.inaturalist.org/observations?place_id=49140&subview=map&taxon_id=47762)

# Add rockfish conservation areas

What projection is the RCA shapefile?



```
rca <- read_sf("data/rca.sf")  
  
st_crs(rca)
```

```
Coordinate Reference System:  
User input: Canada_Albers_Equal_Area_Conic  
wkt:  
PROJCRS["Canada_Albers_Equal_Area_Conic",  
    BASEGEOGCRS["NAD83",  
        DATUM["North American Datum 1983",  
            ELLIPSOID["GRS 1980",6378137,298.257222101,  
                LENGTHUNIT["metre",1]]],  
        PRTMF["Greenwich",0]
```

```
        AREA["Canada - onshore and offshore - Alberta; British Columbia; Manitoba; Ne  
Brunswick; Newfoundland and Labrador; Northwest Territories; Nova Scotia; Nunavut; On  
rio; Prince Edward Island; Quebec; Saskatchewan; Yukon."],  
        BBOX[38.21,-141.01,86.46,-40.73]],  
        ID["ESRI",102001]]
```

**Canada Albers Equal Area Conic  
(102001)**

# Add rockfish conservation areas



## Transform RCA layer

```
rca.wgs <- st_transform(rca, crs = st_crs(4326))
```

```
Coordinate Reference System:
```

```
User input: EPSG:4326
```

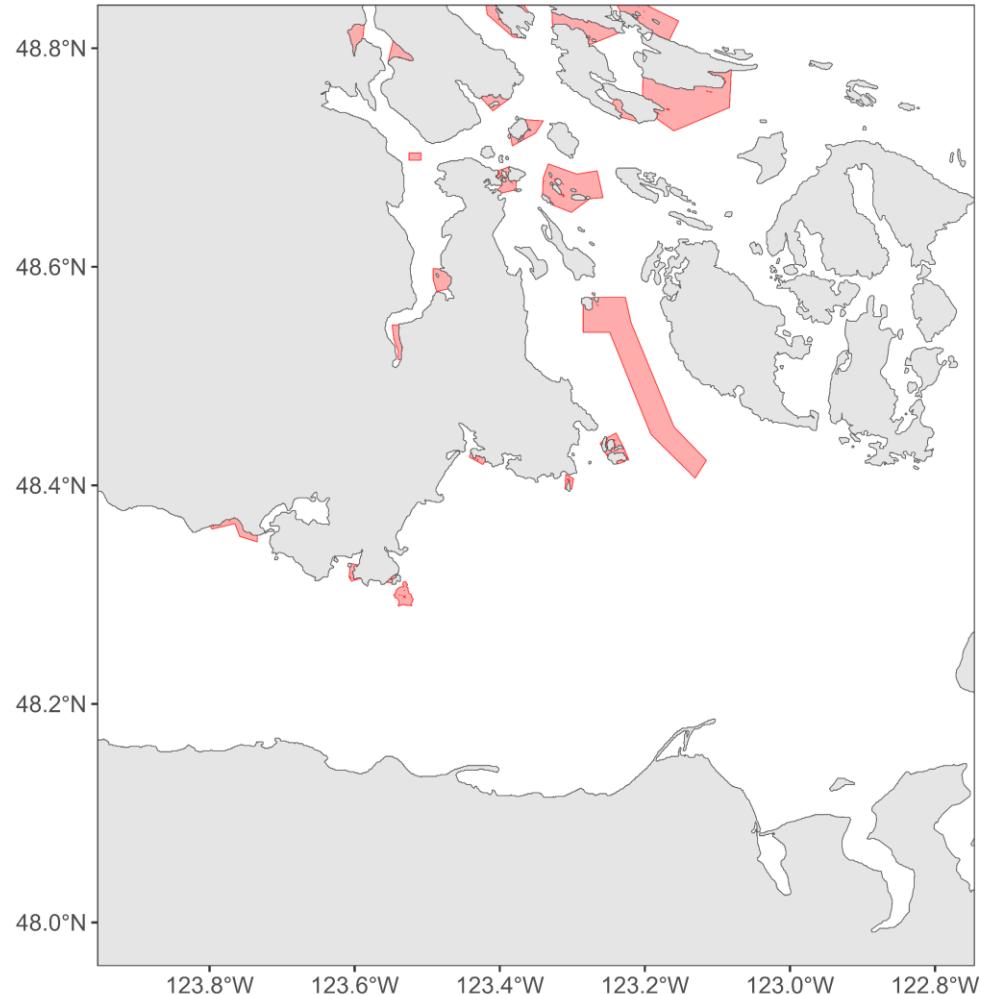
```
wkt:
```

```
GEOGCRS["WGS 84",  
    ENSEMBLE["World Geodetic System 1984 ensemble",
```

Now in WGS 1984 (4326)

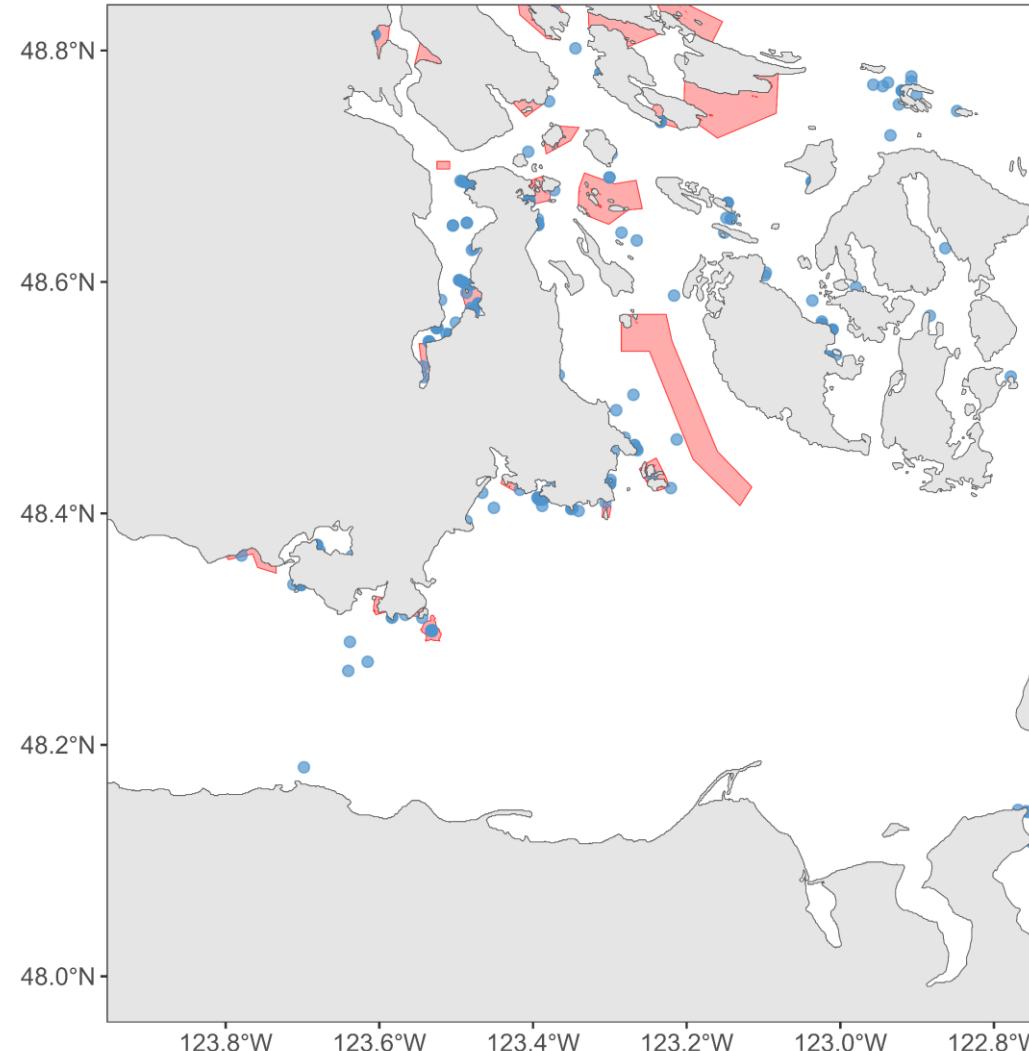
# Add rockfish conservation areas

```
ggplot() +  
  geom_sf(data = bc.coast) +  
  geom_sf(data = rca, fill = "firebrick1")
```

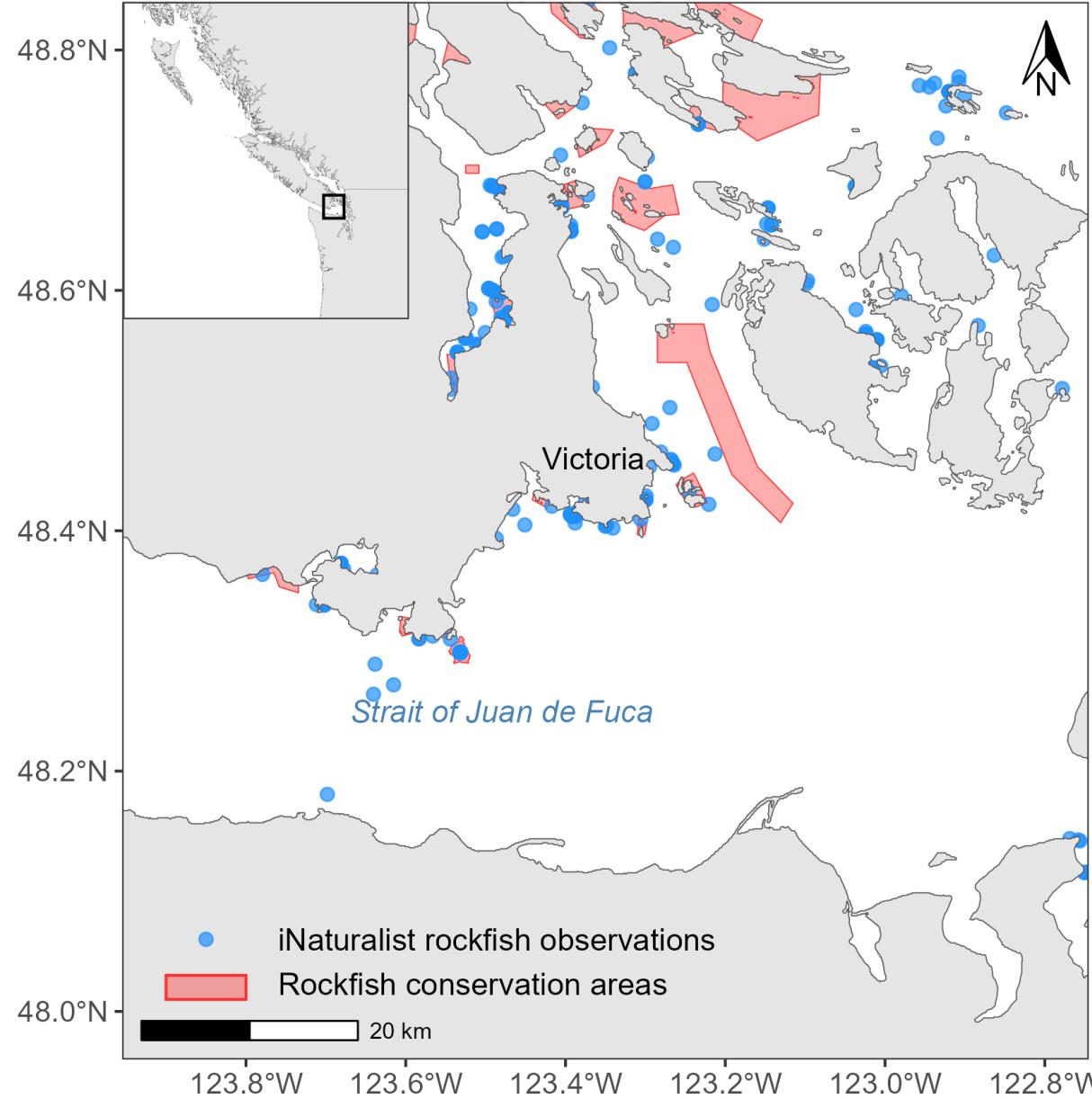


# Add iNaturalist observations

```
geom_point(data = iNaturalist.rockfish,  
           aes(x = longitude, y = latitude))
```



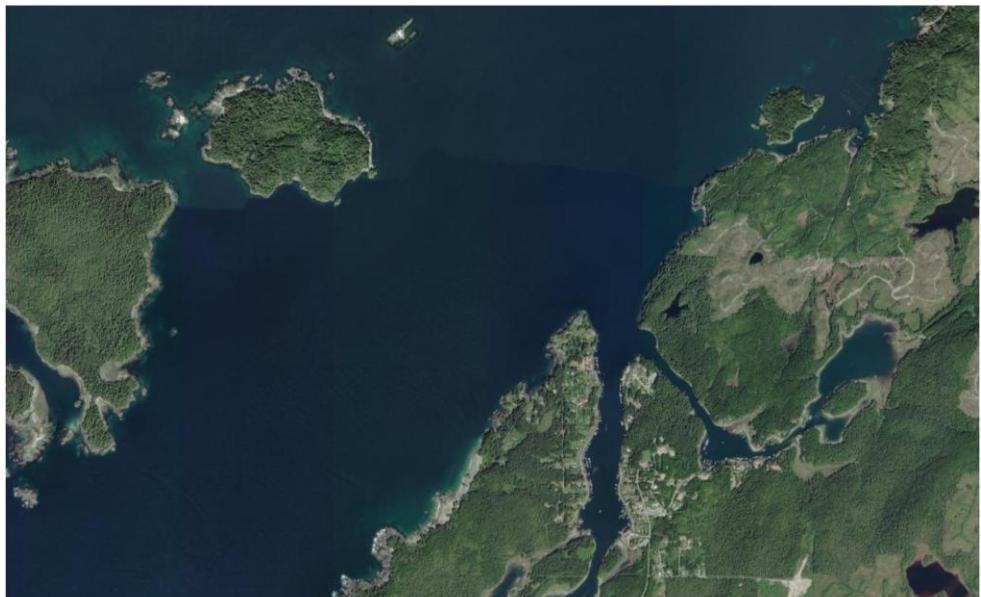
# Add inset map, labels



# Satellite maps

**basemaps** or **OpenStreetMap**  
packages

Cool way to show your study  
area!

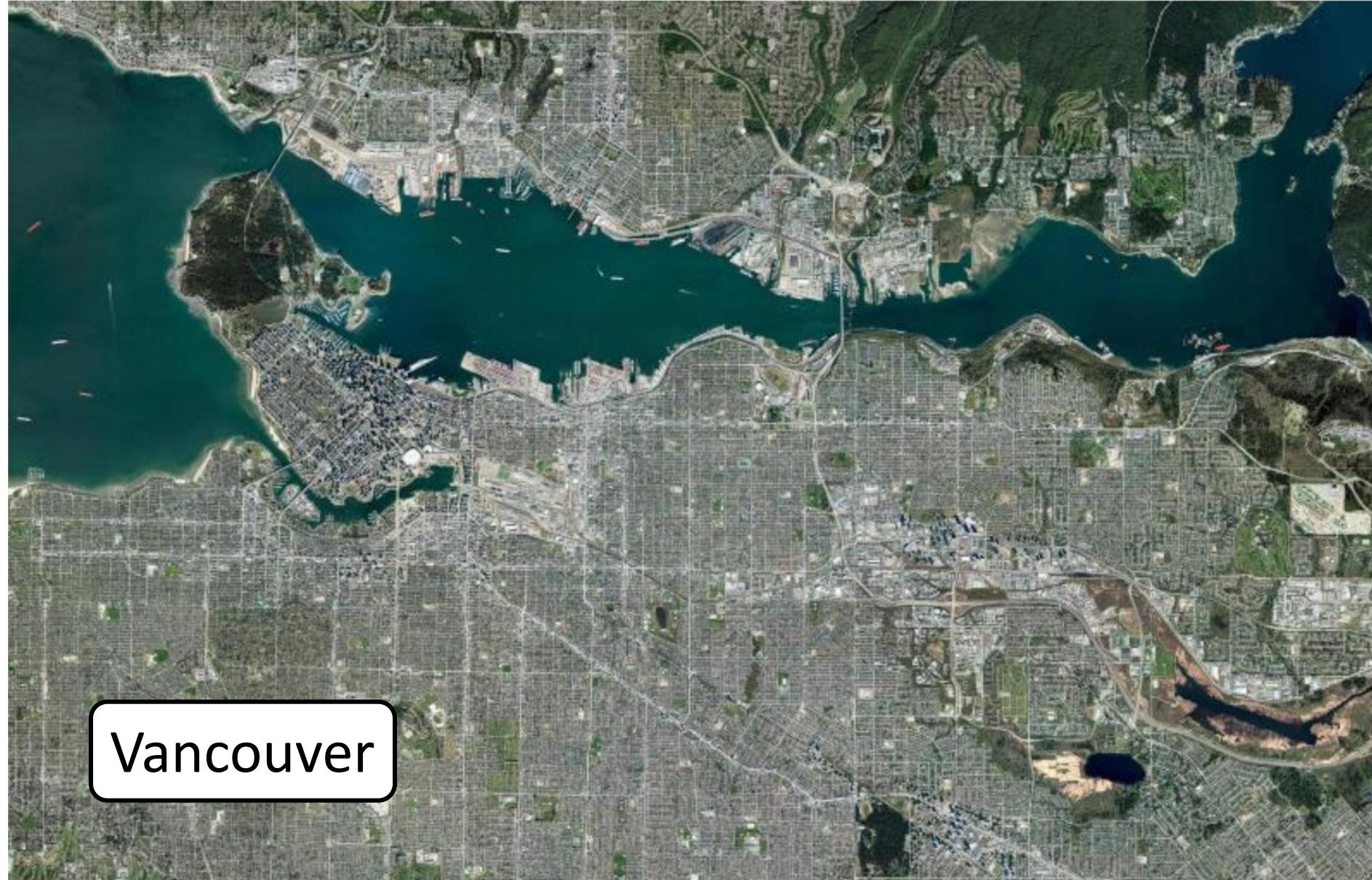


# Satellite maps



Cape Town

# Satellite maps



# Satellite maps

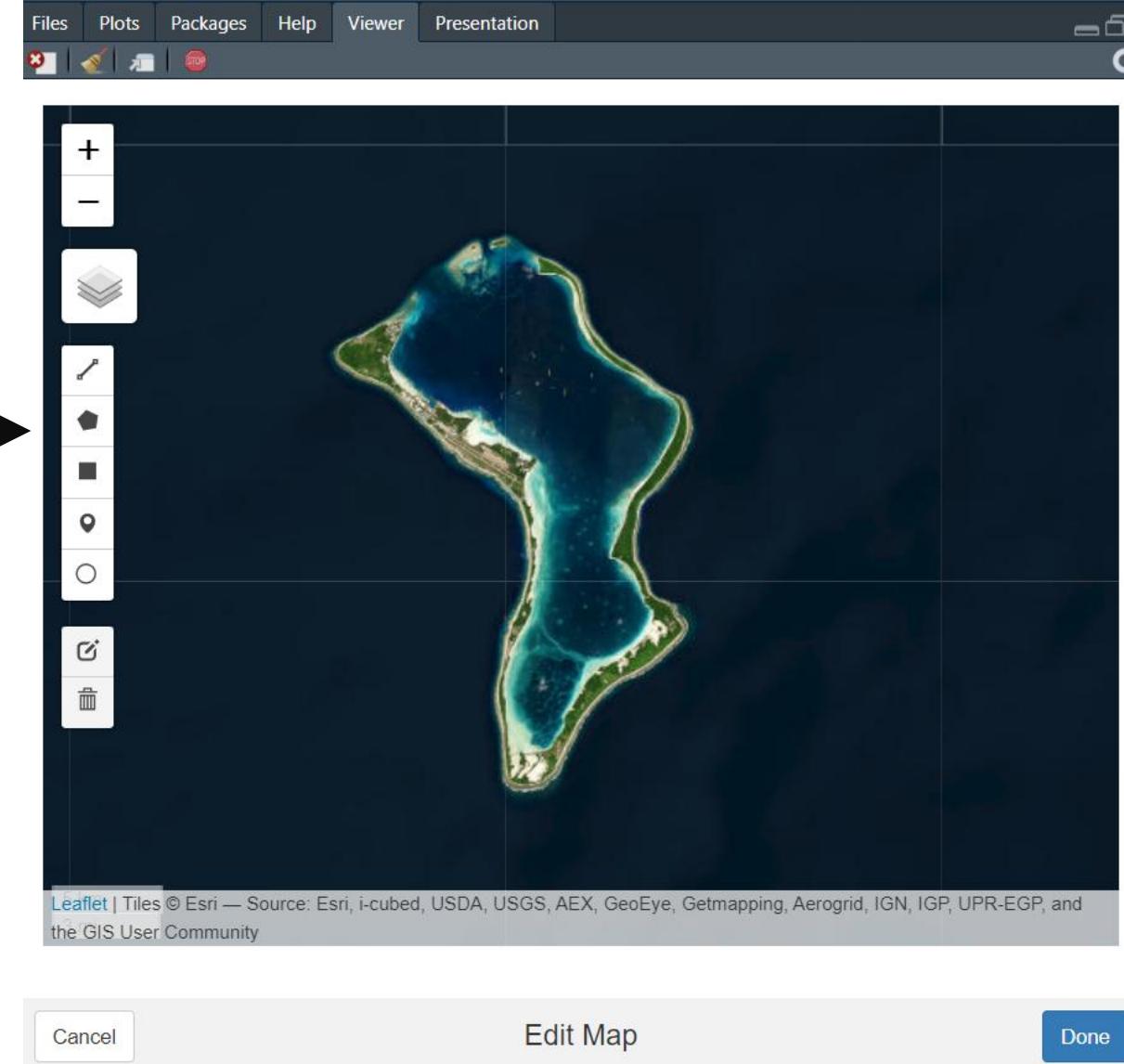
## basemaps package

1. Define bounding box, either with

`sf::st_bbox()` or `basemaps::draw_ext()` →



```
bc.box <- st_bbox(c(xmin = -142.3, ymin = 43,  
                     xmax = -110.5, ymax = 58),  
                     crs = st_crs(4326))
```

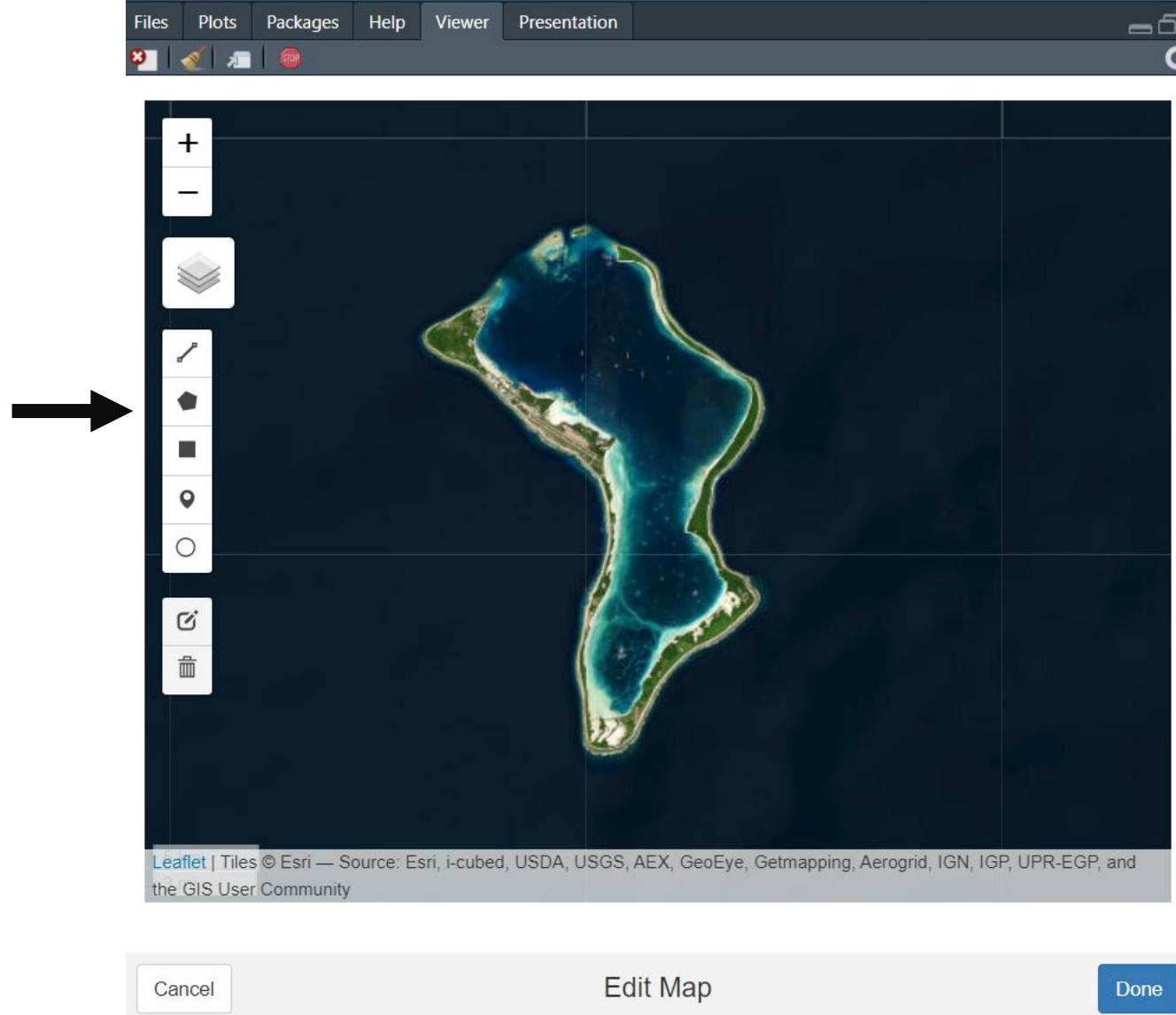


# Satellite maps

**basemaps** package

```
06-satellite-maps.R
```

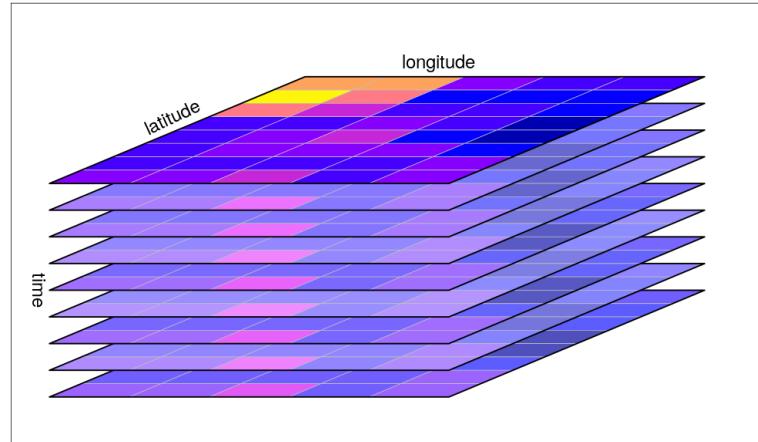
```
get_maptypes()  
basemaps::draw_ext()
```



# Satellite maps

```
bc.box <- st_bbox(c(xmin = -123.95, ymin = 48.1,  
                     xmax = -122.9, ymax = 48.7),  
                     crs = st_crs(4326))  
  
satellite <- basemap_stars(ext = bc.box,  
                           map_service = "esri",  
                           type = "world_imagery",  
                           map_res = 1)
```

stars package:  
raster cubes

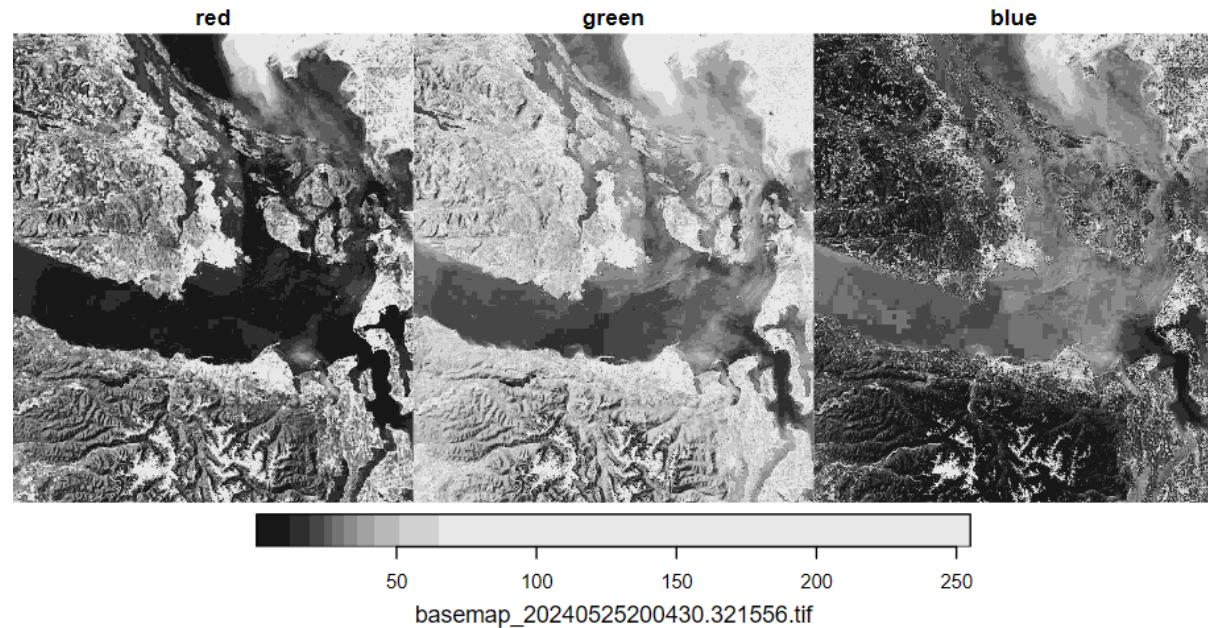


# Satellite maps

```
satellite <- basemap_stars(ext = bc.box,  
                           map_service = "esri",  
                           type = "world_imagery",  
                           map_res = 1)  
  
satellite.rgb <- st_rgb(satellite)
```

`st_rgb()` converts the satellite imagery into one RGB image

`st_rgb()` converts the satellite imagery into one RGB image



`st_rgb()`



# Satellite maps

```
satellite <- basemap_stars(ext = bc.box,  
                           map_service = "esri",  
                           type = "world_imagery",  
                           map_res = 1)  
  
st_crs(satellite)
```

Satellite basemap is in Web Mercator (EPSG 3857)

Can transform to WGS 84 but reduces quality.

# Satellite maps

```
satellite <- basemap_stars(ext = bc.box,  
                           map_service = "esri",  
                           type = "world_imagery",  
                           map_res = 1)  
  
satellite.rgb <- st_rgb(satellite)
```

The basemap was initially in Web Mercator (EPSG 3857), so we convert it to WGS 84 (EPSG 4326)

Same CRS: compatible with other layers

# Satellite maps

```
ggplot() +  
  geom_stars(data = satellite.rgb) +  
  geom_sf(data = rca.wm)
```



# Satellite maps

```
ggplot() +  
  geom_stars(data = satellite.rgb) +  
  geom_sf(data = rca.wm)
```

Transformed RCAs to  
Web Mercator

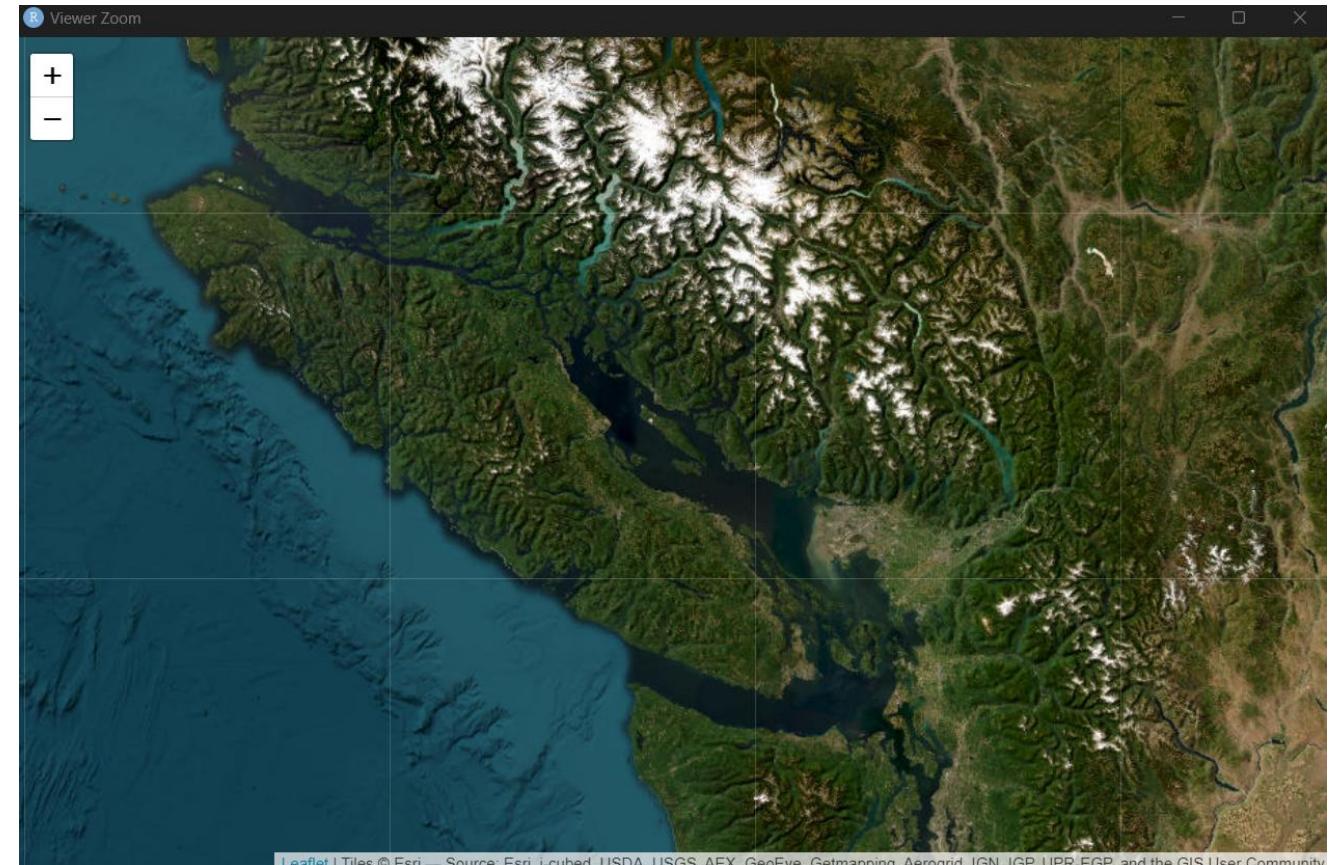




# Interactive maps with leaflet

leaflet is a Javascript library to make interactive maps

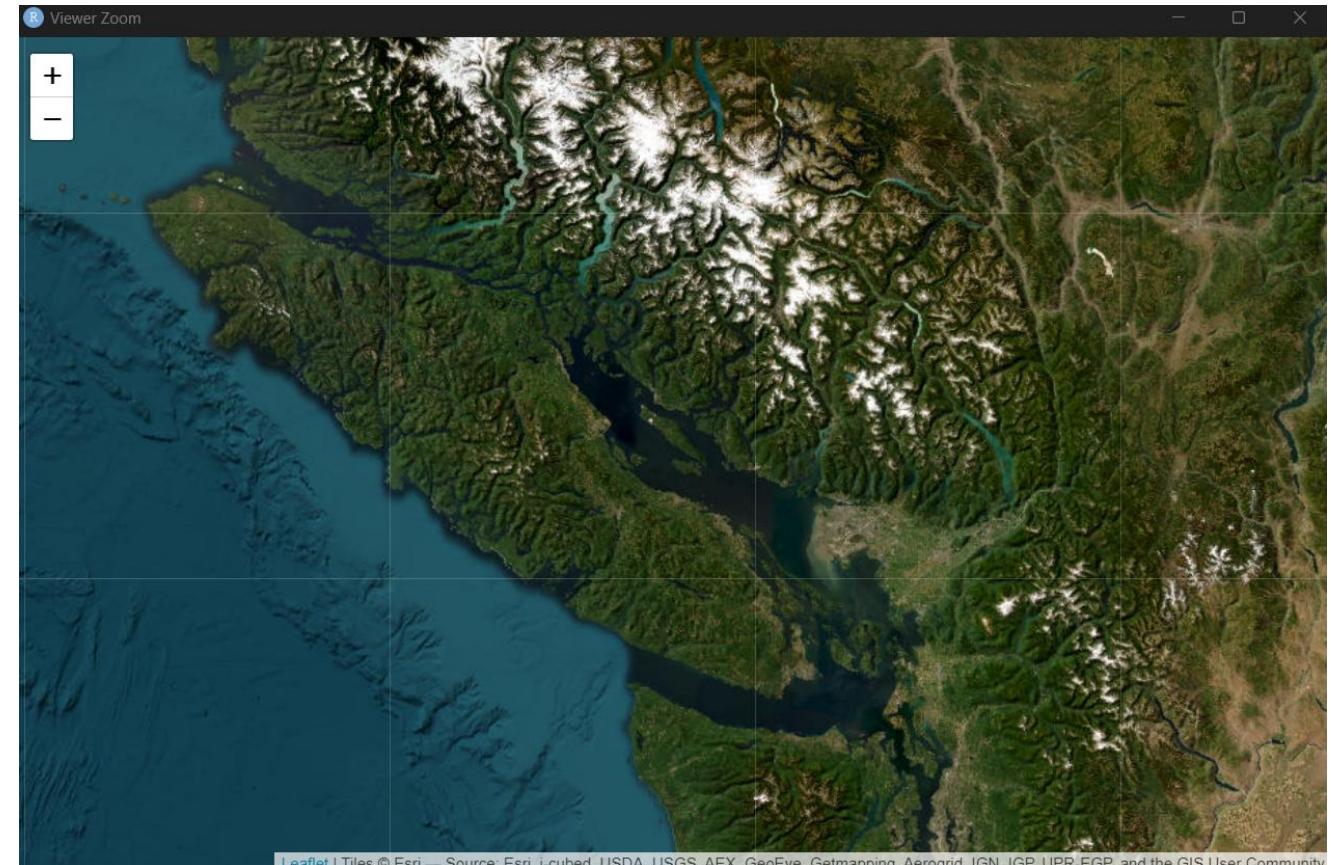
The **leaflet** package allows us to make these maps in R



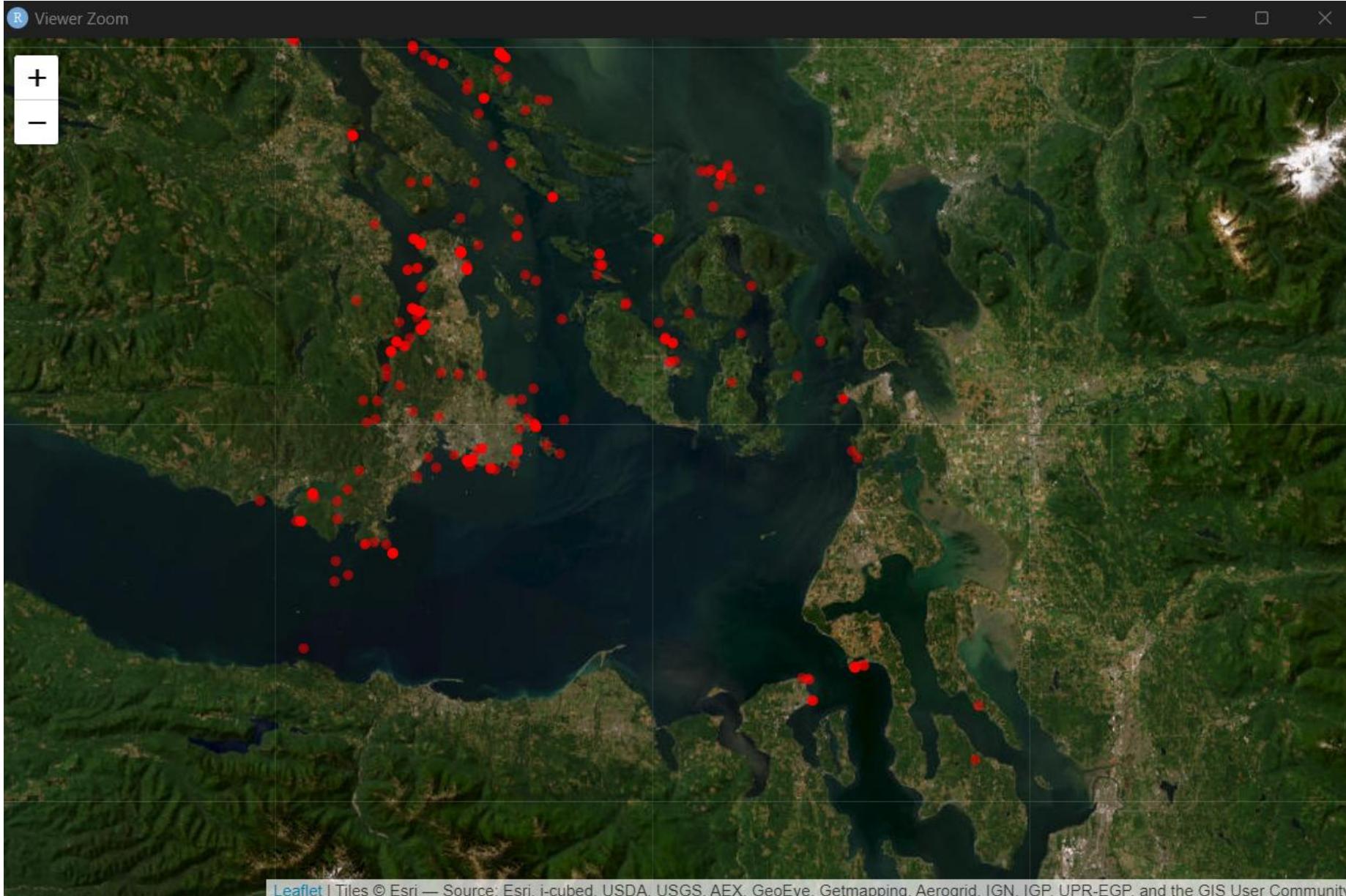
# Interactive maps with leaflet

leaflet is a Javascript library to make interactive maps

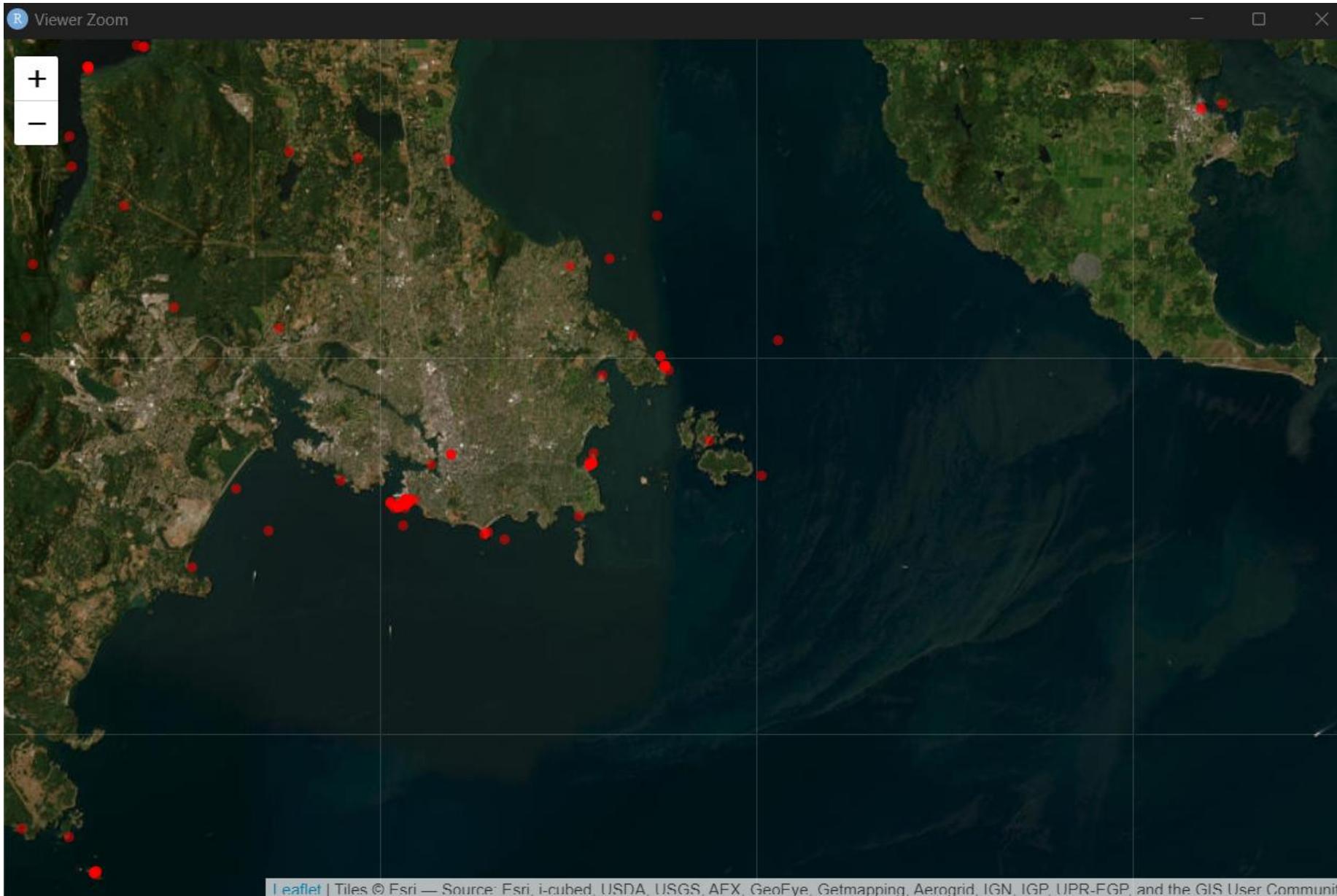
The **leaflet** package allows us to make these maps in R



# Interactive maps with leaflet

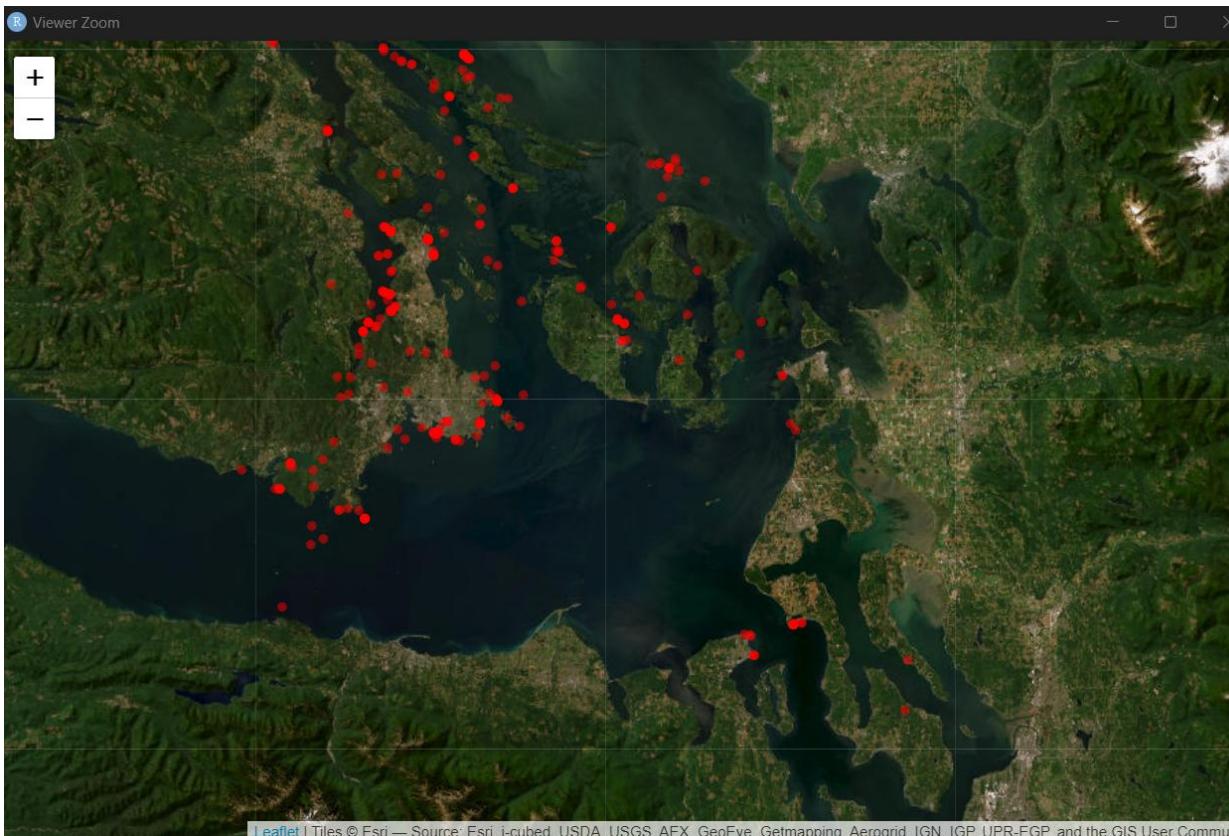


# Interactive maps with leaflet



# Interactive maps with leaflet

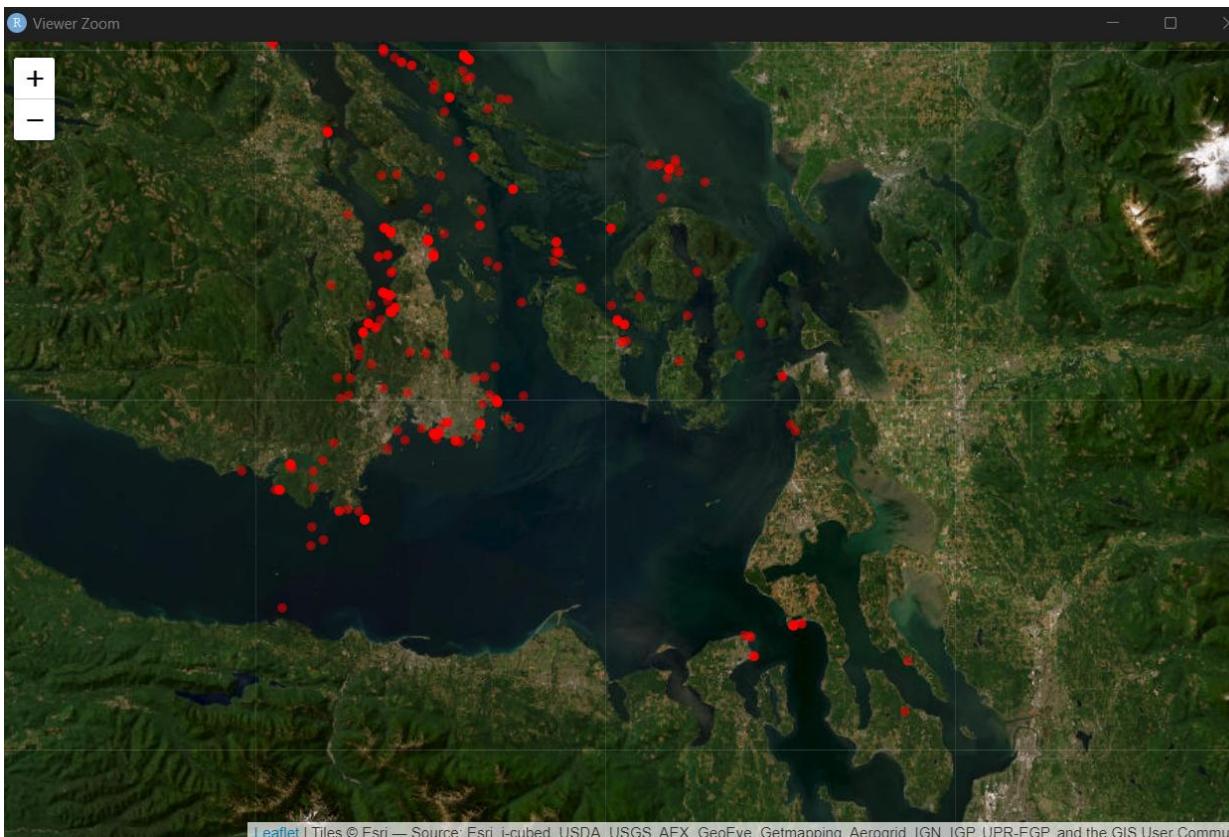
```
leaflet() |>  
  setView(lng = -123, lat = 48.4, zoom = 9) |>  
  addCircles(data = rockfish,  
             lng = rockfish$longitude,  
             lat = rockfish$latitude) |>  
  addProviderTiles(providers$Esri.WorldImagery)
```



# Interactive maps with leaflet

|> strings together  
multiple functions

```
leaflet() |>  
  setView(lng = -123, lat = 48.4, zoom = 9) |>  
  addCircles(data = rockfish,  
             lng = rockfish$longitude,  
             lat = rockfish$latitude) |>  
  addProviderTiles(providers$Esri.WorldImagery)
```



# Interactive maps with leaflet

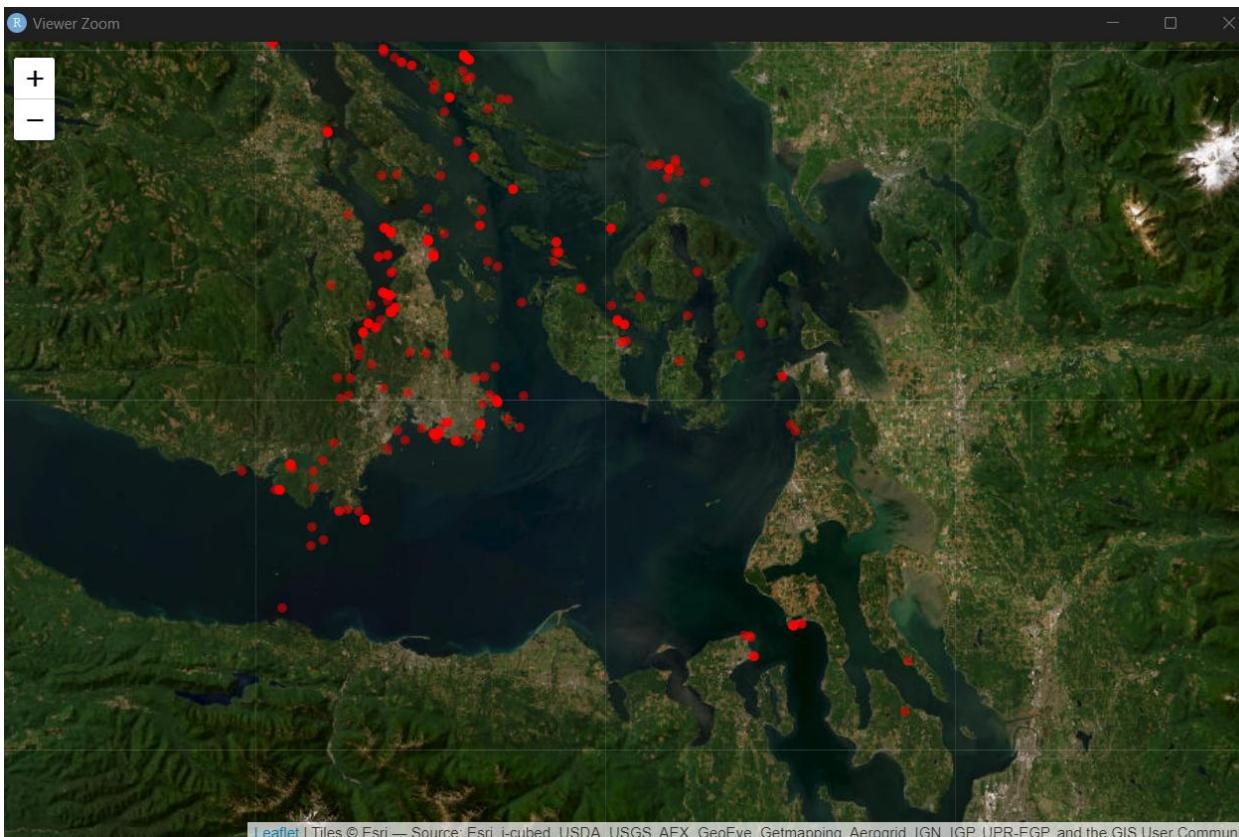
```
leaflet() |>  
  setView(lng = -123, lat = 48.4, zoom = 9) |>  
  addCircles(data = rockfish,  
             lng = rockfish$longitude,  
             lat = rockfish$latitude) |>  
  addProviderTiles(providers$Esri.WorldImagery)
```



|> strings together  
multiple functions

baseR version of %>

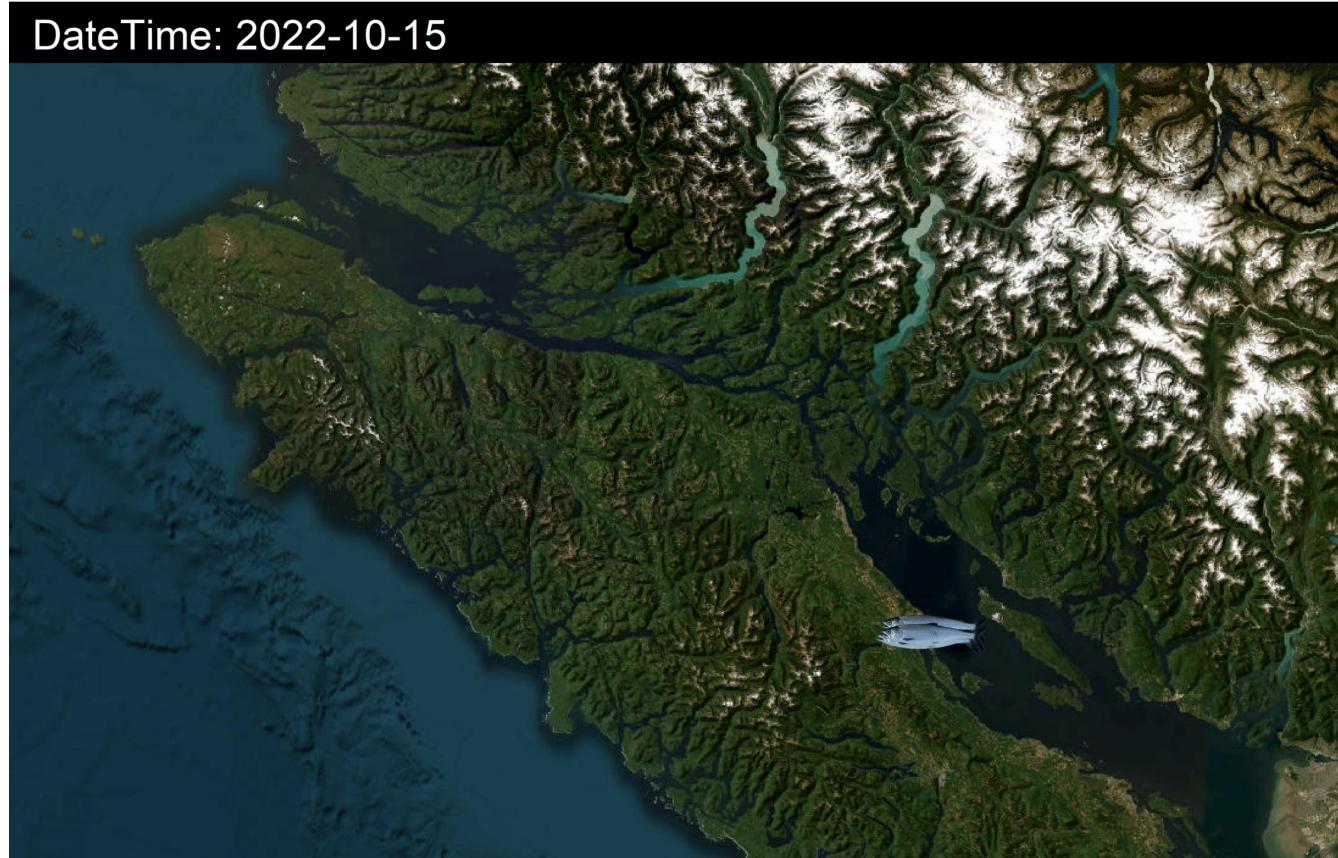
“pipe”



# Animated maps

**ganimate** package

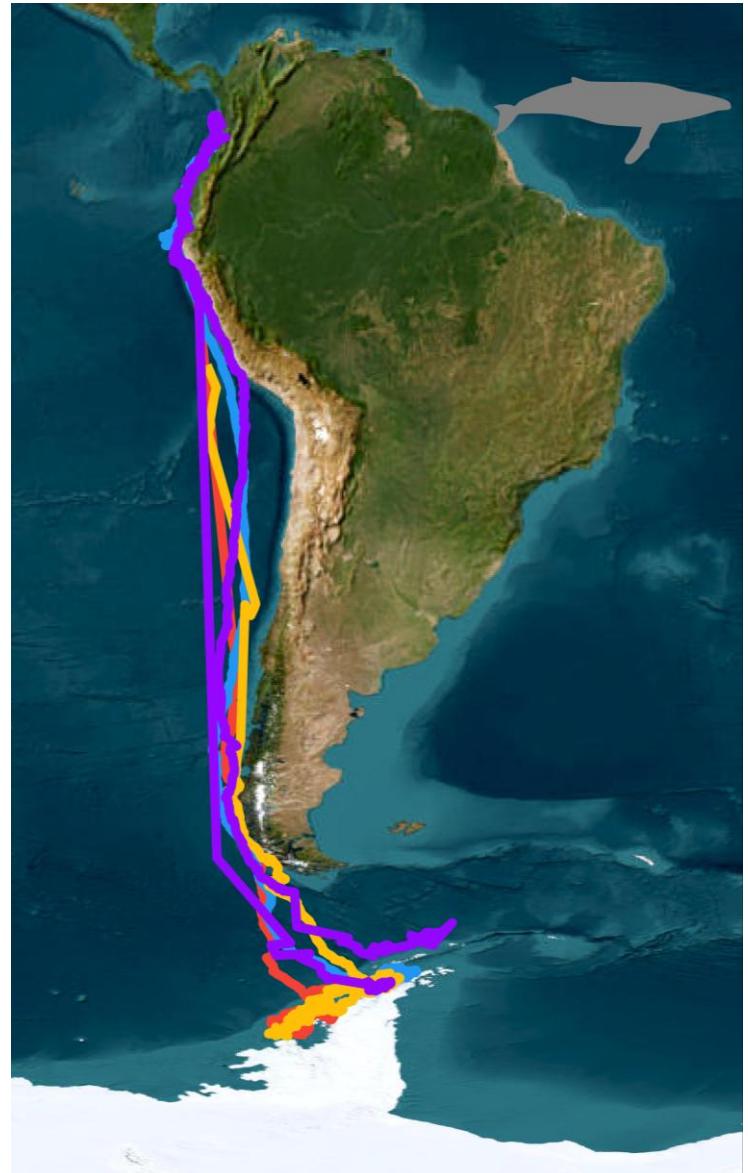
Exciting tool to visualize how a system changes over time, or track movements of tagged animals



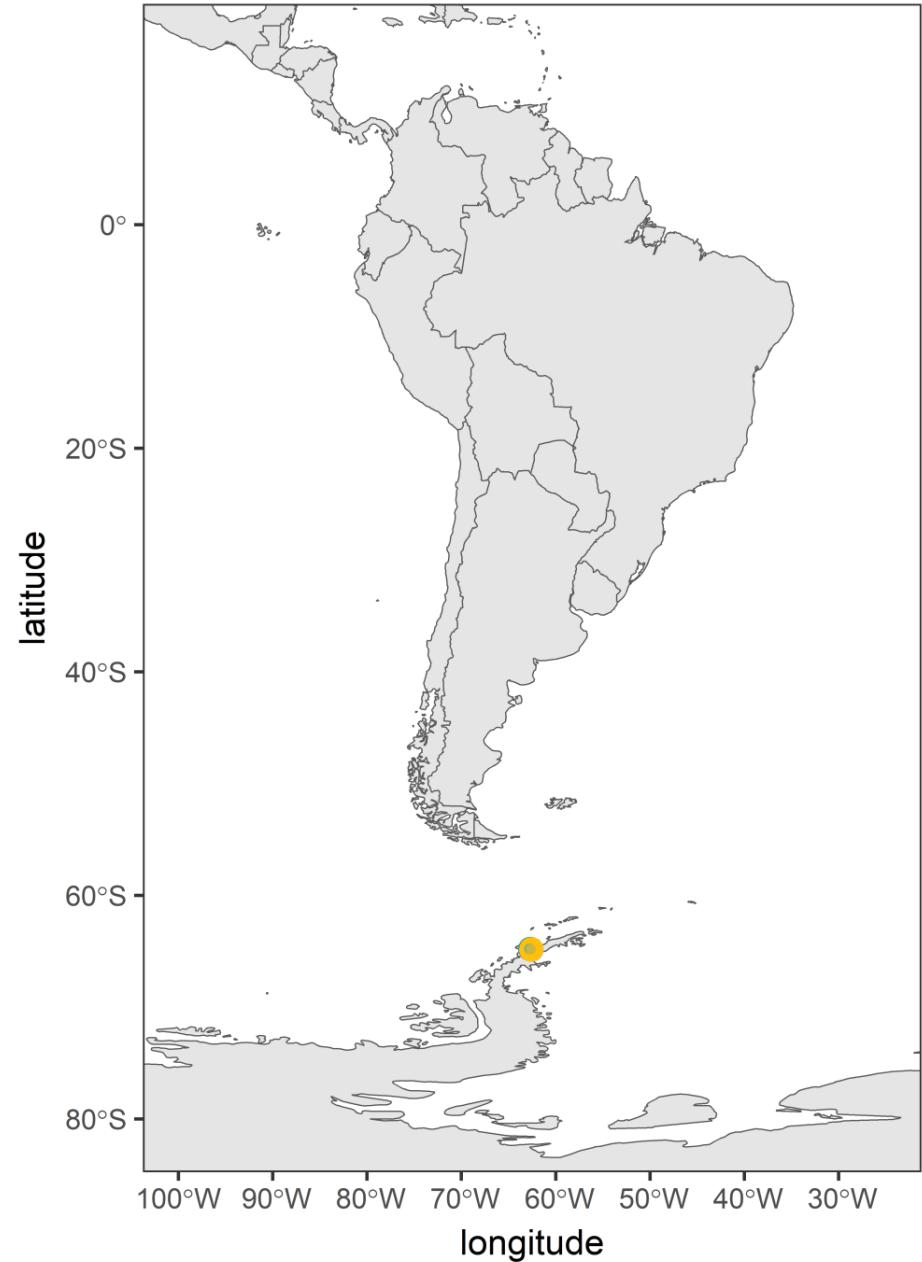
DateTime: 2022-10-15



# Humpback whale migrations



Date: 2013-02-05



# Humpback whale data

Satellite tags placed on humpback whales in 2013

RESEARCH

Open Access



CrossMark

Capturing foraging and resting behavior using nested multivariate Markov models in an air-breathing marine vertebrate

Ben G. Weinstein<sup>1\*</sup> , Ladd Irvine<sup>1</sup> and Ari S. Friedlaender<sup>1,2</sup>

RESEARCH

Open Access



First description of migratory behavior of humpback whales from an Antarctic feeding ground to a tropical calving ground

Michelle Modest<sup>1\*</sup> , Ladd Irvine<sup>2</sup>, Virginia Andrews-Goff<sup>3</sup>, William Gough<sup>4</sup>, David Johnston<sup>5</sup>, Douglas Nowacek<sup>5</sup>, Logan Pallin<sup>1</sup>, Andrew Read<sup>5</sup>, Reny Tyson Moore<sup>6</sup> and Ari Friedlaender<sup>1</sup>



NOAA Fisheries

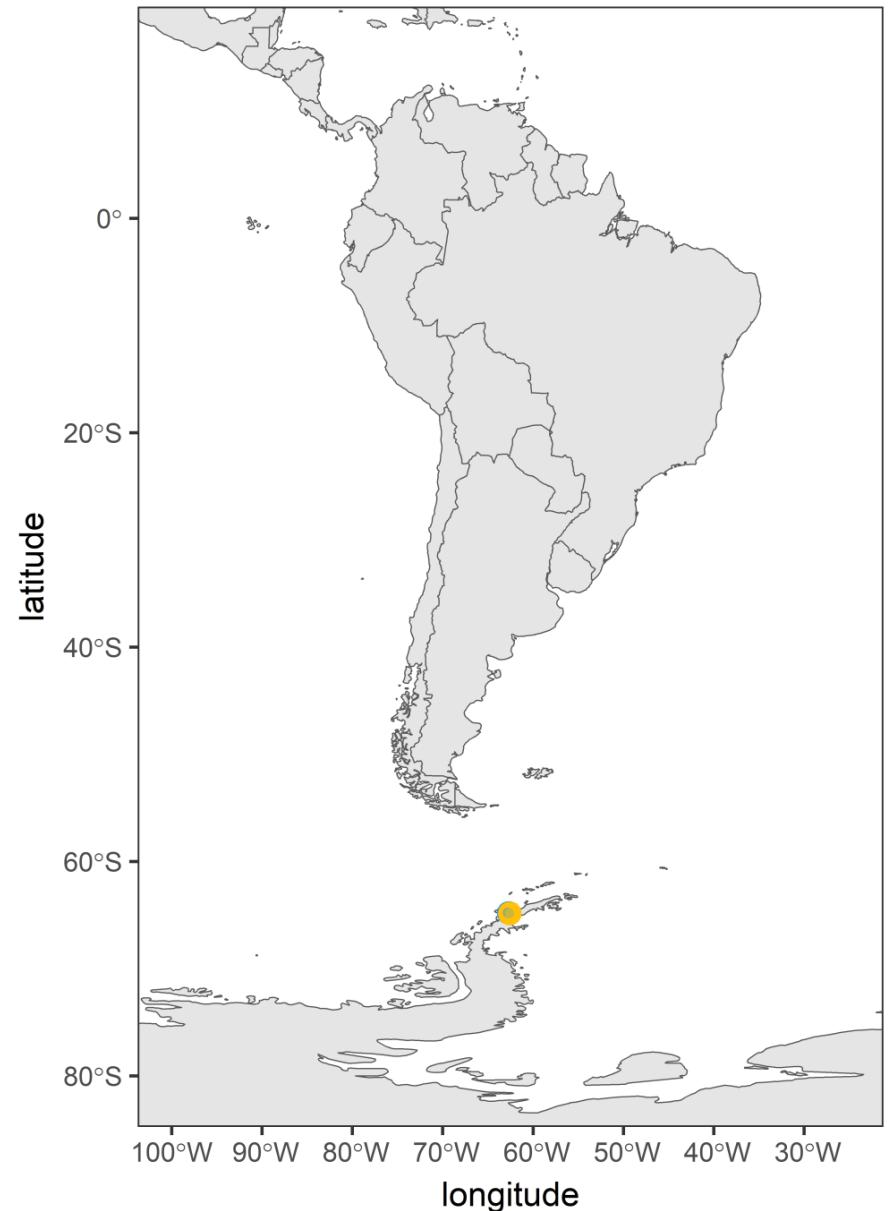


National Geographic

# Animated maps

Date: 2013-02-05

1. Make static map with `ggplot()`
2. Add animation with `ganimate` functions



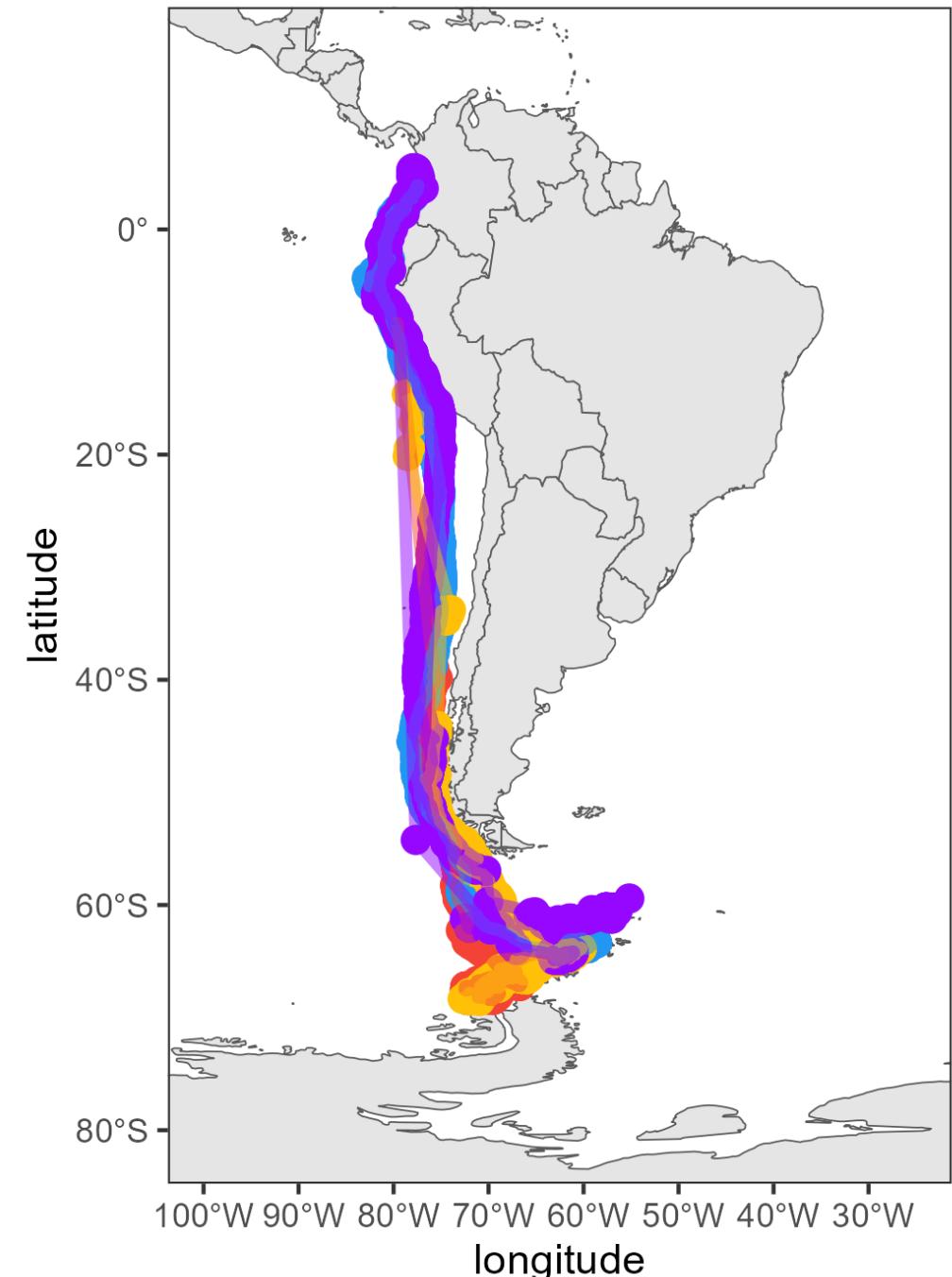
# Animated maps

## 1. Static map

```
static <- ggplot() +  
  geom_sf(data = south.america.antarctica) +  
  geom_point(data = humpbacks,  
             aes(x = longitude, y = latitude,  
                  group = animal_id)) +  
  geom_point(data = humpbacks,  
             aes(x = longitude, y = latitude,  
                  group = animal_id))
```

`geom_point()`: all satellite tag locations

`geom_path()`: connects successive  
locations

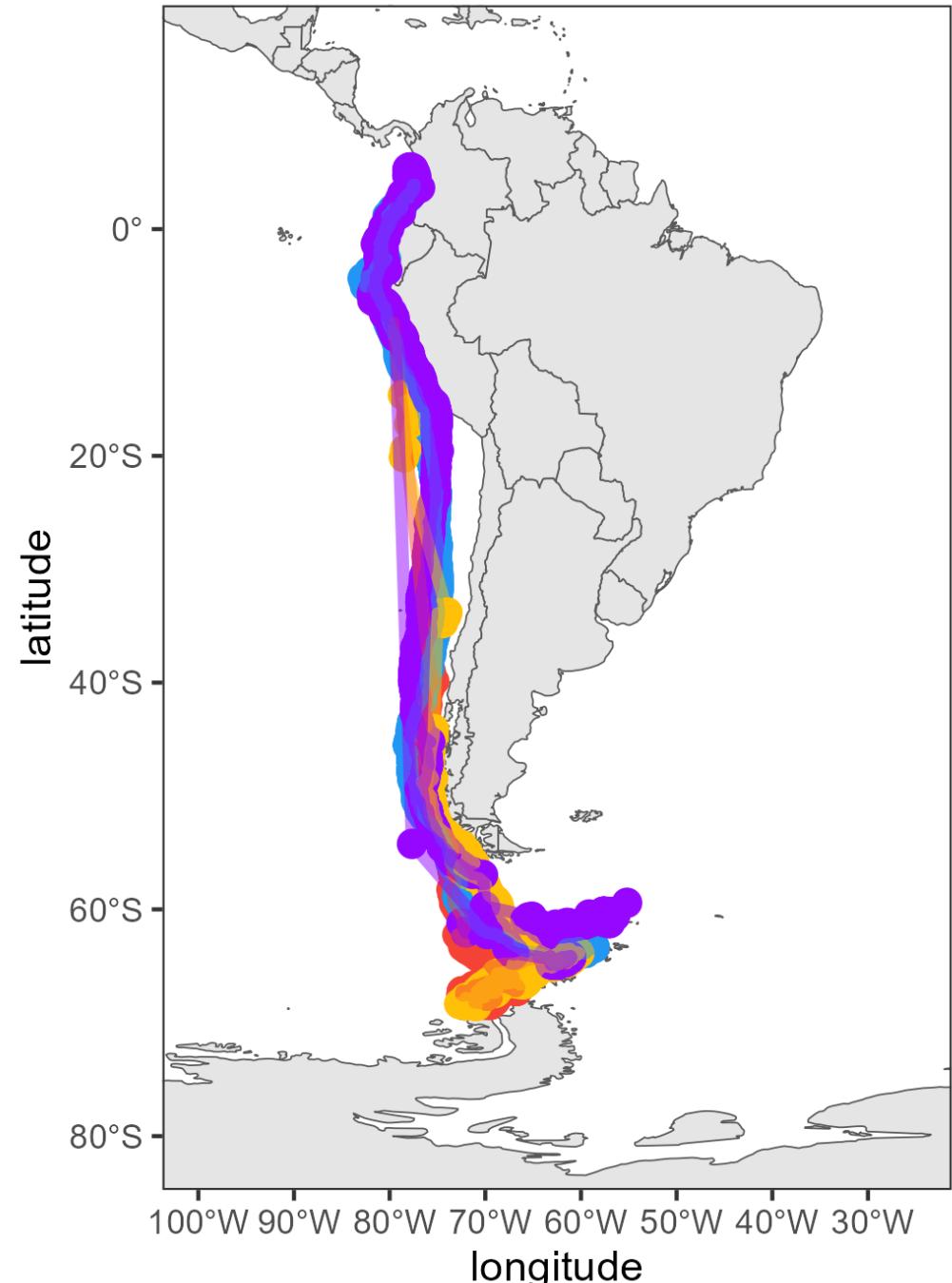


# Animated maps

## 1. Static map

```
static <- ggplot() +  
  geom_sf(data = south.america.antarctica) +  
  geom_point(data = humpbacks,  
             aes(x = longitude, y = latitude,  
                  group = animal_id)) +  
  geom_point(data = humpbacks,  
             aes(x = longitude, y = latitude,  
                  group = animal_id))
```

ganimate recognizes `geom_point()` and  
`geom_path()` together  
Plots the entire track path to date, but  
only the current point



# Animated maps

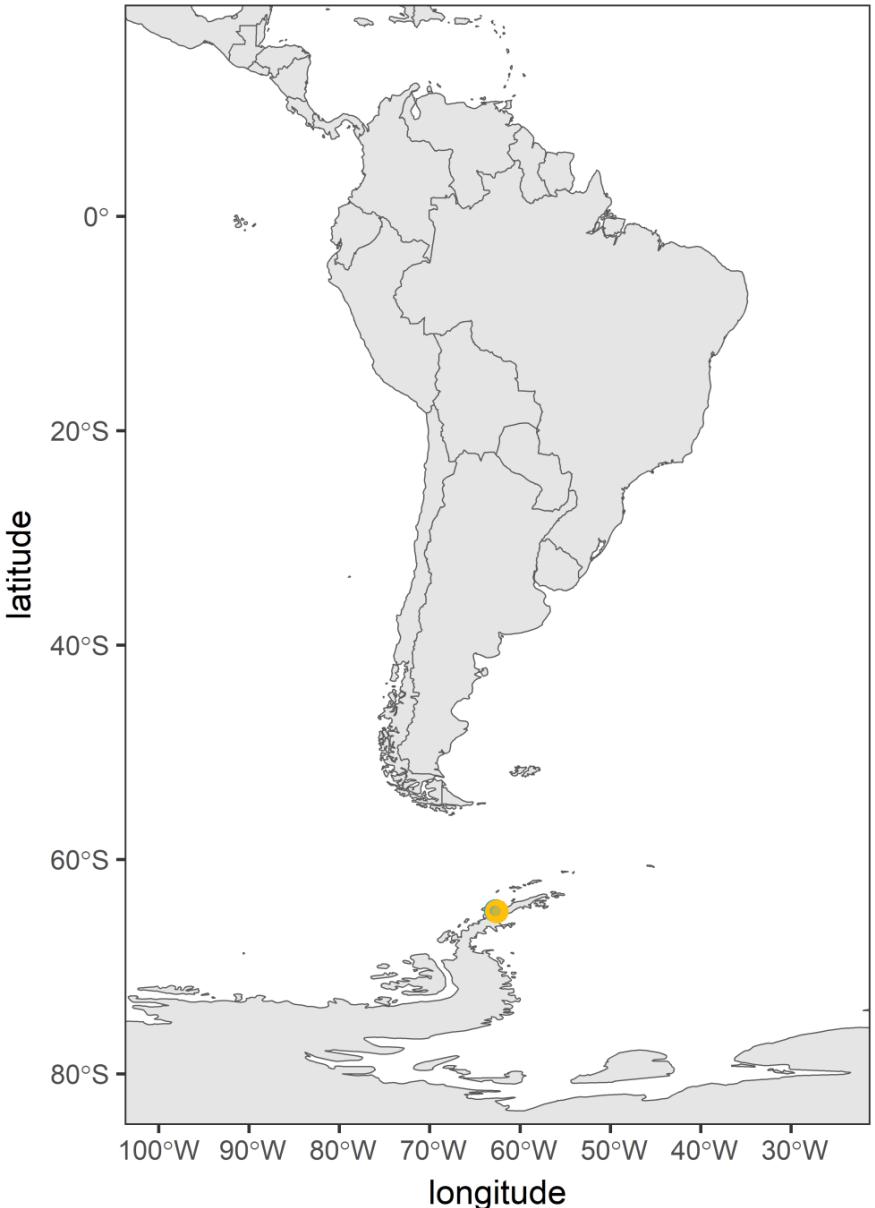
1. Make static map with ggplot()

2. Add animation with gganimate  
functions

```
dynamic <- static +  
  transition_reveal(datetime) +  
  labs(title = "Date:as_date(frame_along)}") +  
  ease_aes("linear")  
  
anim_save("figures/whale-animation.gif", dynamic)
```

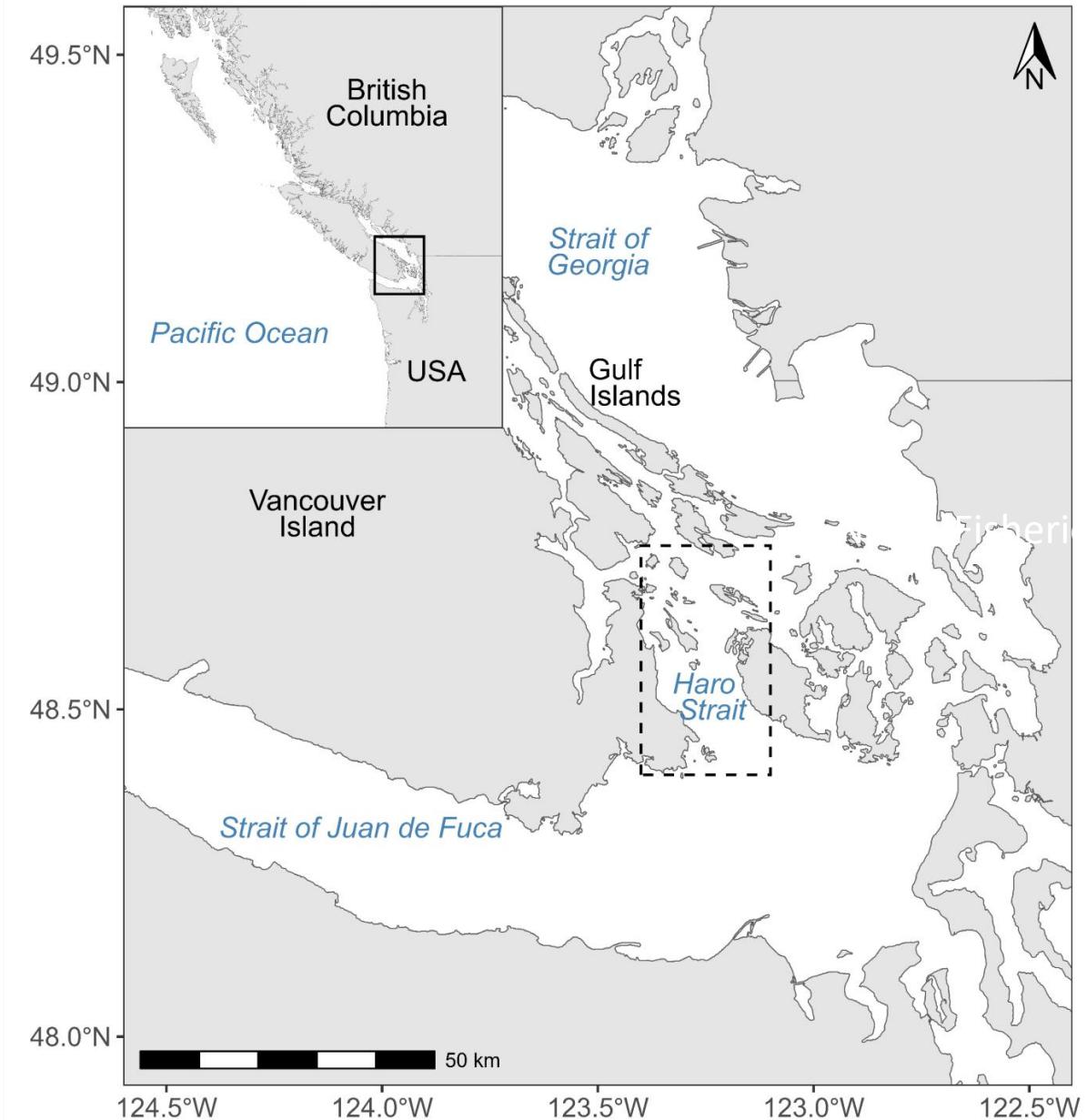
Single point for each whale, with path  
following behind the point

Date: 2013-02-05



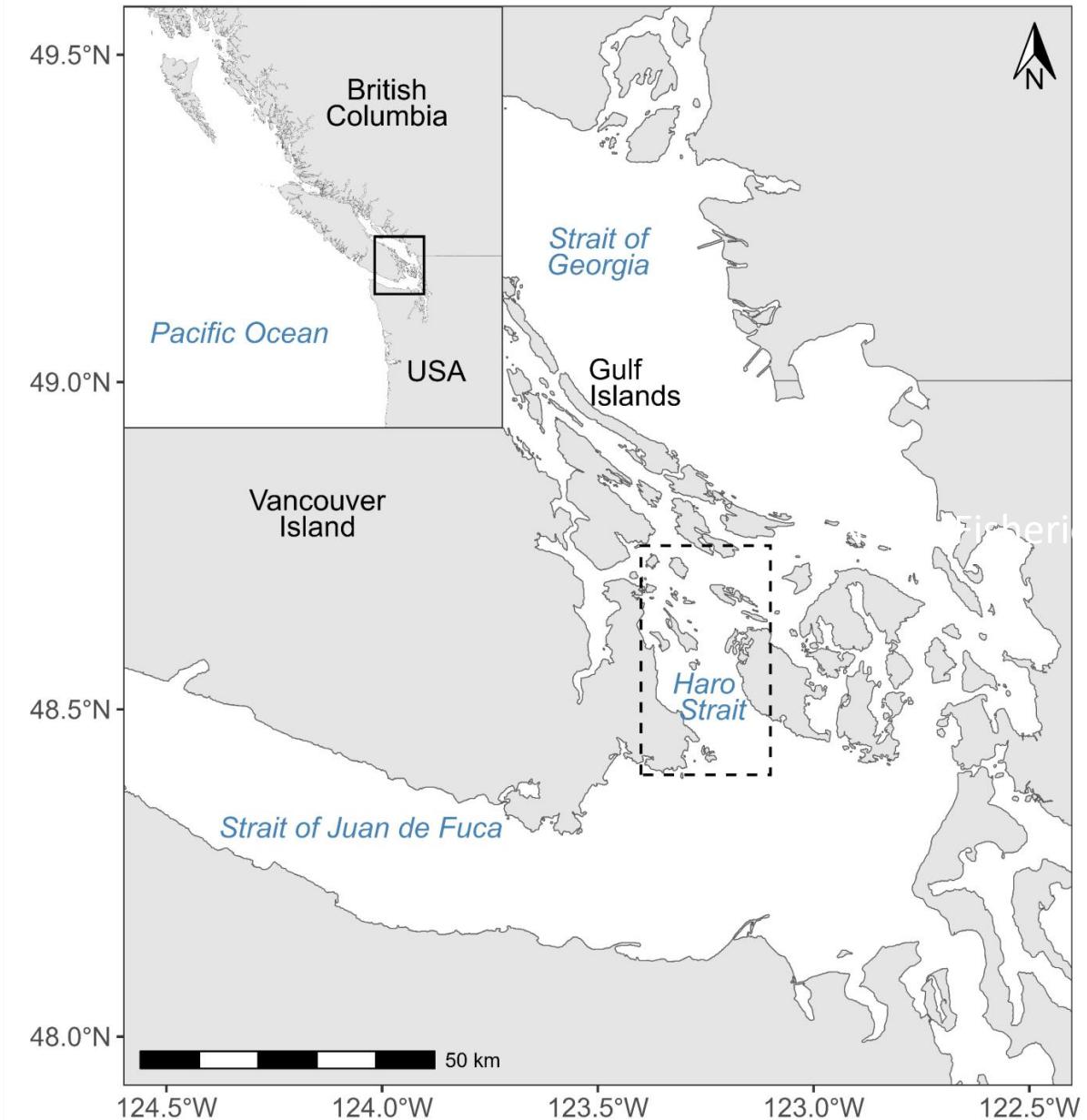
# Key Points

1. Study area maps:
  - a. high resolution
  - b. show what's important
  - c. clean

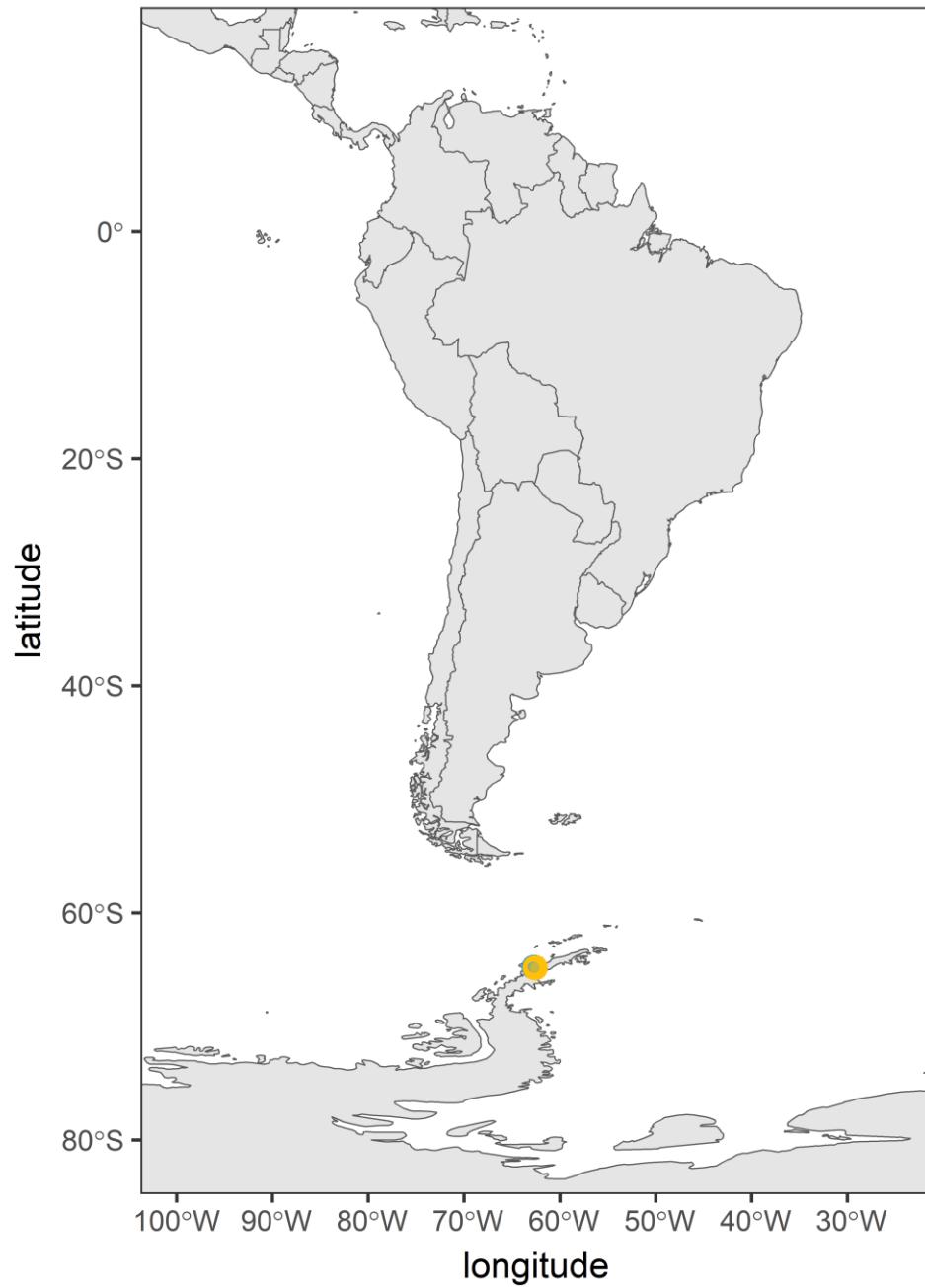


# Key Points

- 1. Study area maps:**
  - a. high resolution
  - b. show what's important
  - c. clean
  
- 2. R can make lots of different maps**
  - a. bathymetry, topography
  - b. satellite maps
  - c. interactive maps
  - d. animated maps



Date: 2013-02-05



# Acknowledgements

UVic BUGS

Canadian Institute of Ecology and Evolution

Francis Juanes, Will Duguid, and Juanes/Baum labmates

Meme support: Maddy Thomson, Maeve Corcoran, Tom Zhang



University  
of Victoria



canadian institute of ecology and evolution  
institut canadien d'écologie et d'évolution

# Thank you!

Questions/comments: **wgreentree@outlook.com**

Feel free to reach out with any R spatial questions you have in the future! Happy to give R spatial workshops to other groups!



Let's make our own maps!

