

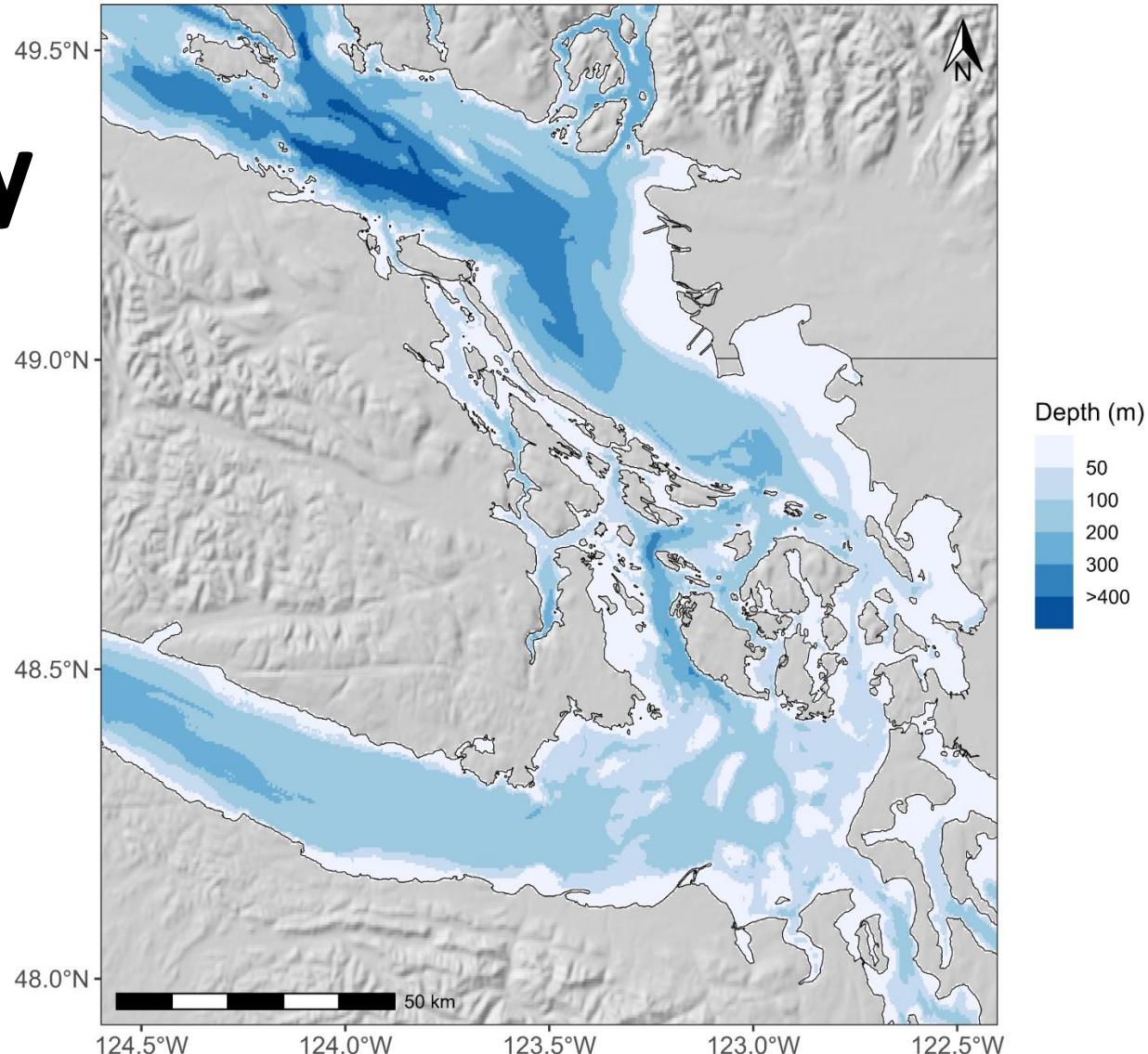
Making maps in R: applications for ecology and evolution

Wesley Greentree

Canadian Society for Ecology and Evolution

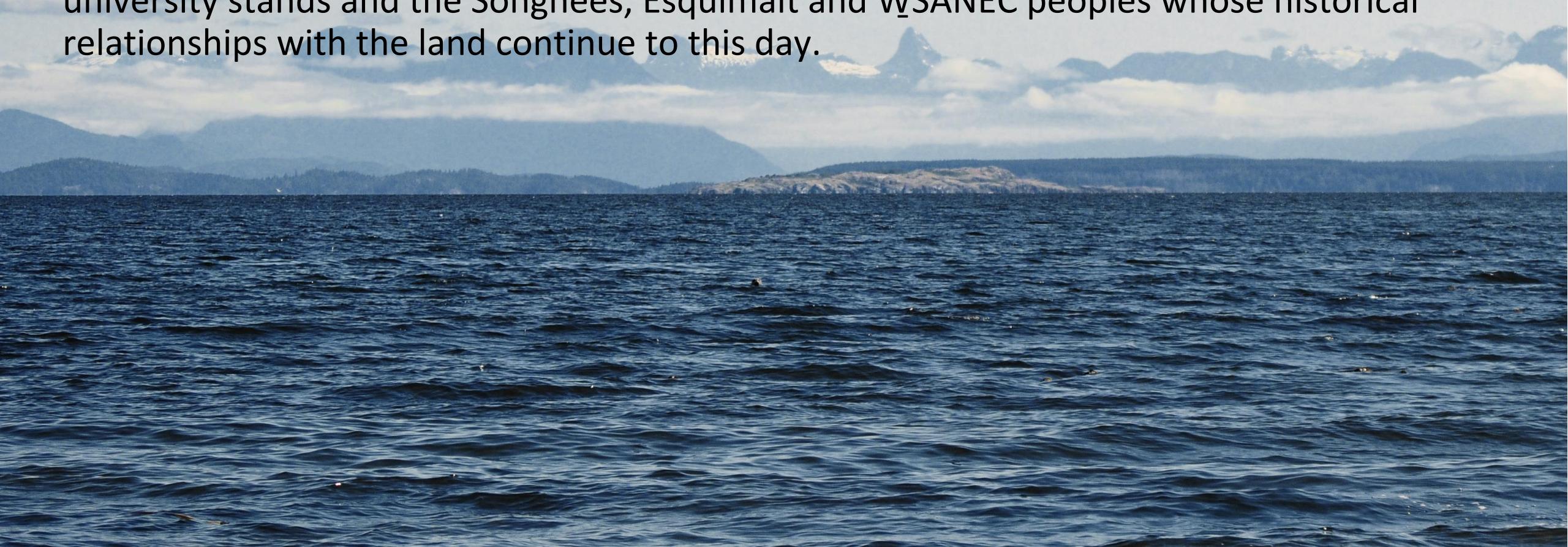
May 26, 2024

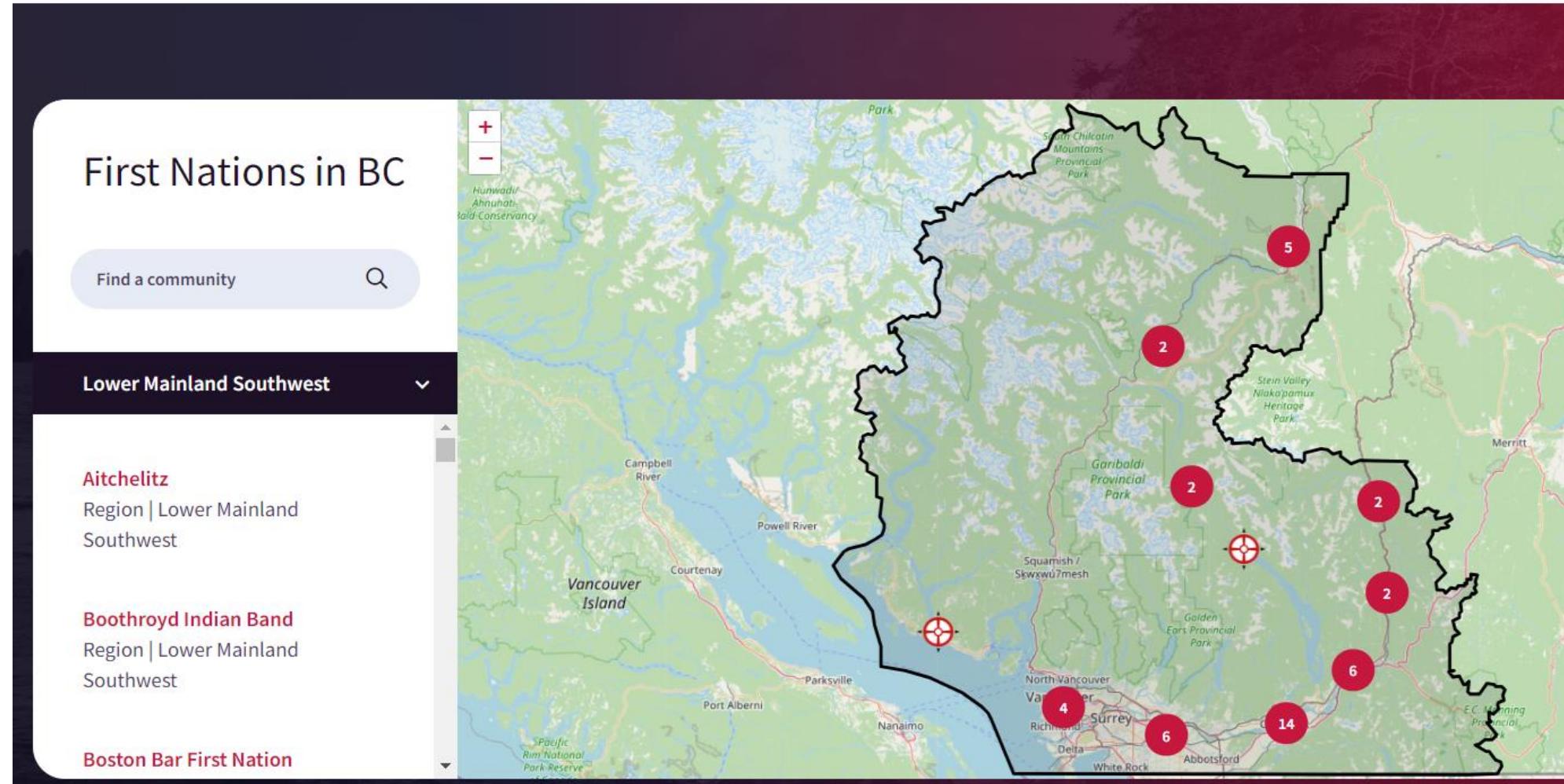
wgreentree@outlook.com

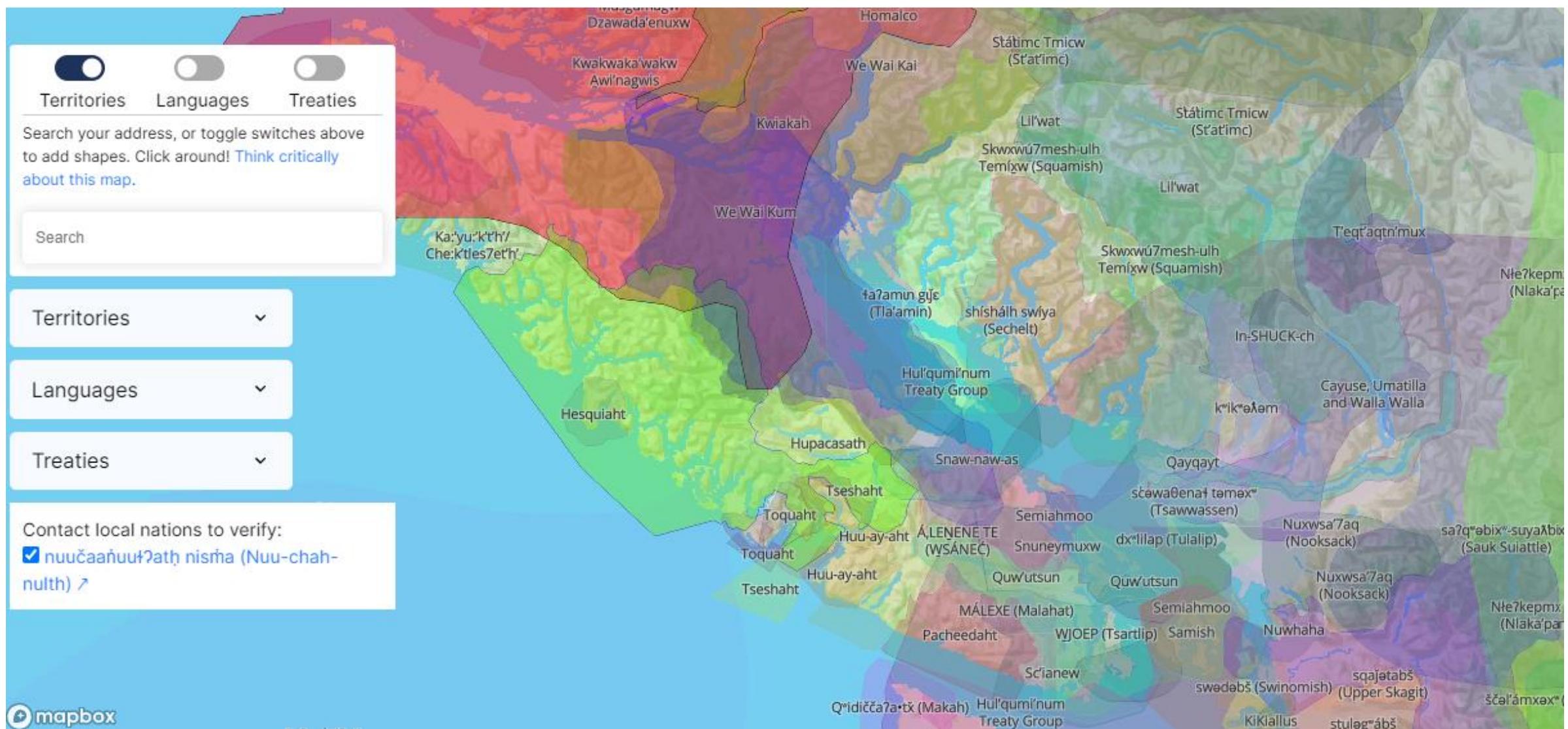


Territory acknowledgements

- **UBC** stands on the unceded territory of the Coast Salish Peoples, including the territories of the xwməθkwəyəm (Musqueam), Skwxwú7mesh (Squamish), Stó:lō and Sə̱l̓ílwətaʔ/Selilwitulh (Tsleil-Waututh) Nations
- **UVic:** I acknowledge and respect the lək'ʷəŋən 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.







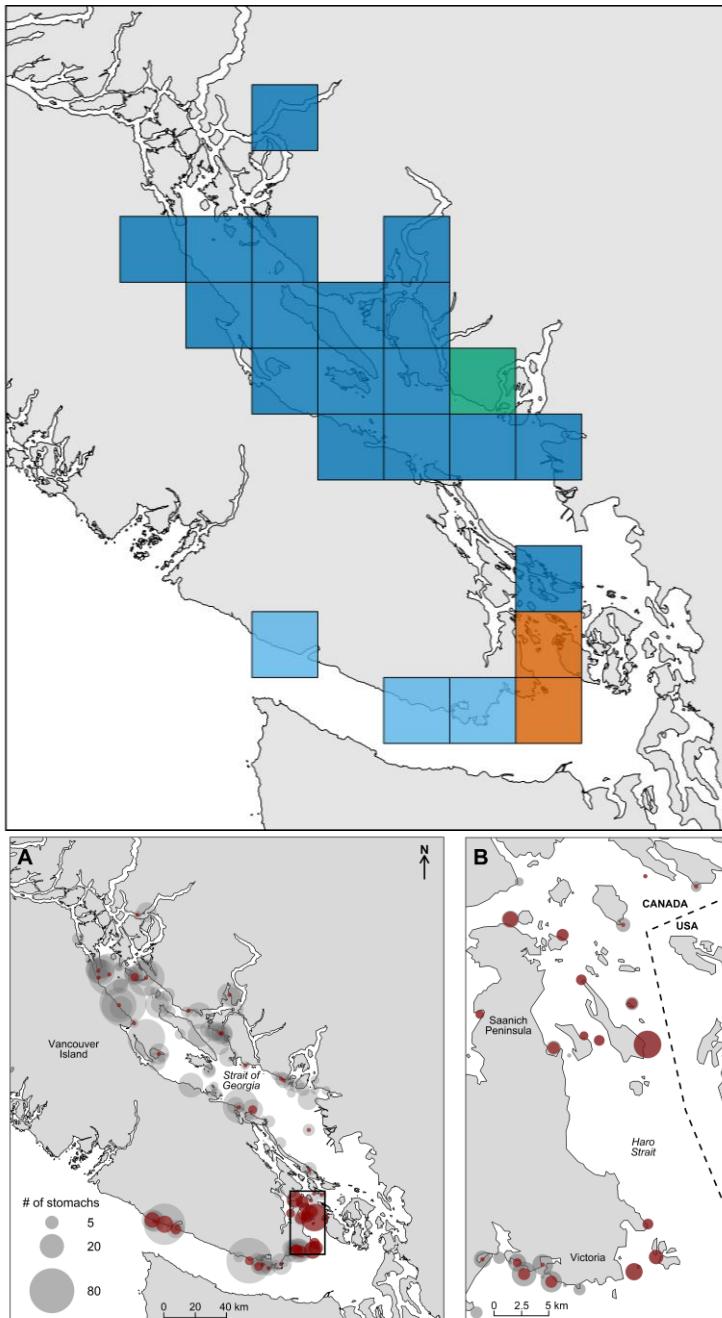
<https://native-land.ca/>



Wesley Greentree (he/him)
PhD student, UVic

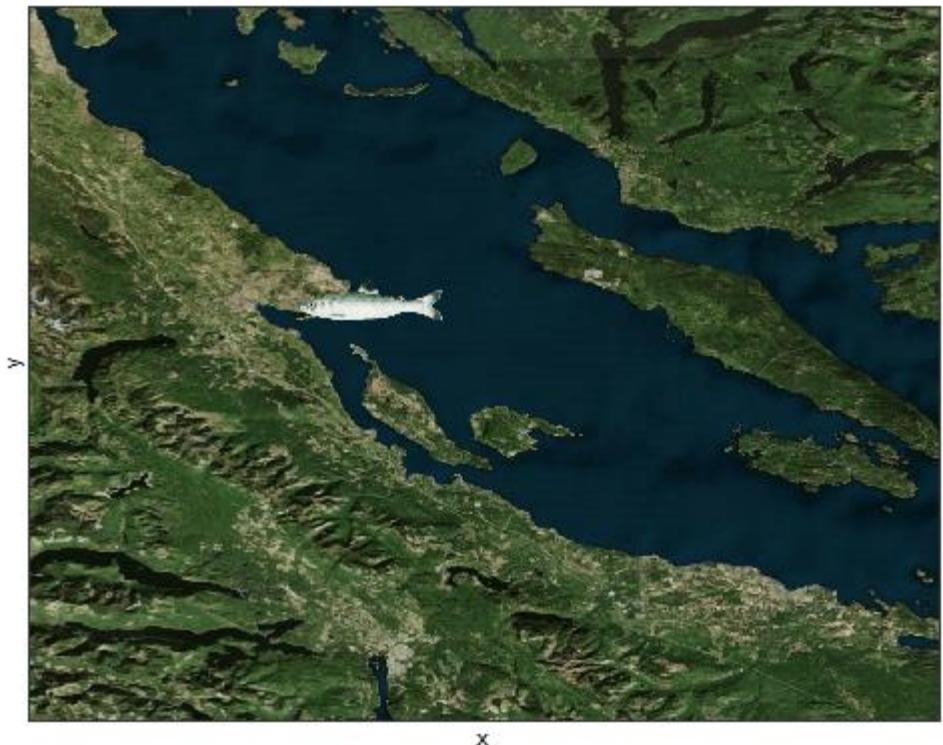


Wesley Greentree (he/him)
PhD student, UVic



Regional food webs

DateTime: 2022-10-15 13:31:01



Salmon migrations

Date: 2023-01-15



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/CSEE2024-R-maps>

The screenshot shows the GitHub repository page for 'CSEE2024-R-maps' owned by 'wesleygreentree'. The repository is public and contains 1 branch and 0 tags. The main file listed is 'README.md'. A context menu is open over the repository name, showing options like 'Unpin', 'Unwatch', 'Fork', and 'Star'. The 'Code' tab is selected in the menu. A detailed 'Clone' section is displayed, showing the HTTPS URL: <https://github.com/wesleygreentree/CSEE2024-R-maps>, along with 'Local' and 'Codespaces' tabs, and download options for 'GitHub Desktop' and 'ZIP'. The repository has 0 stars, 1 watching, and 0 forks. The 'About' section describes it as a 'Workshop on making maps with R'. The 'Releases' and 'Packages' sections are currently empty.

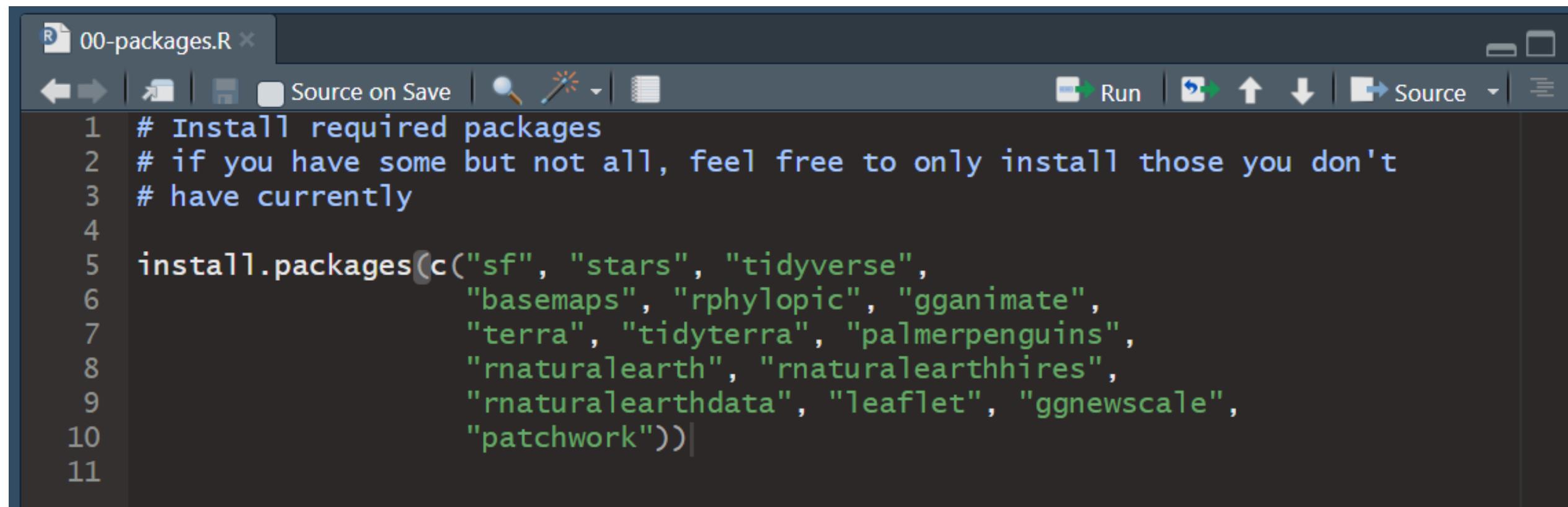
Workshop plan

Open **CSEE2024-R-maps.Rproj**, then open scripts with
File > Open File

 data		2024-05-25 11:51 PM	File folder	
 data/raw		2024-05-25 11:51 PM	File folder	
 figures		2024-05-25 11:51 PM	File folder	
 scripts		2024-05-25 11:51 PM	File folder	
 .gitignore		2024-05-25 11:51 PM	txtfile	1 KB
 CSEE2024-R-maps		2024-05-26 9:41 AM	R Project	1 KB
 README.md		2024-05-25 11:51 PM	MD File	2 KB

Run scripts/00-packages.R

Installs the necessary packages for the workshop

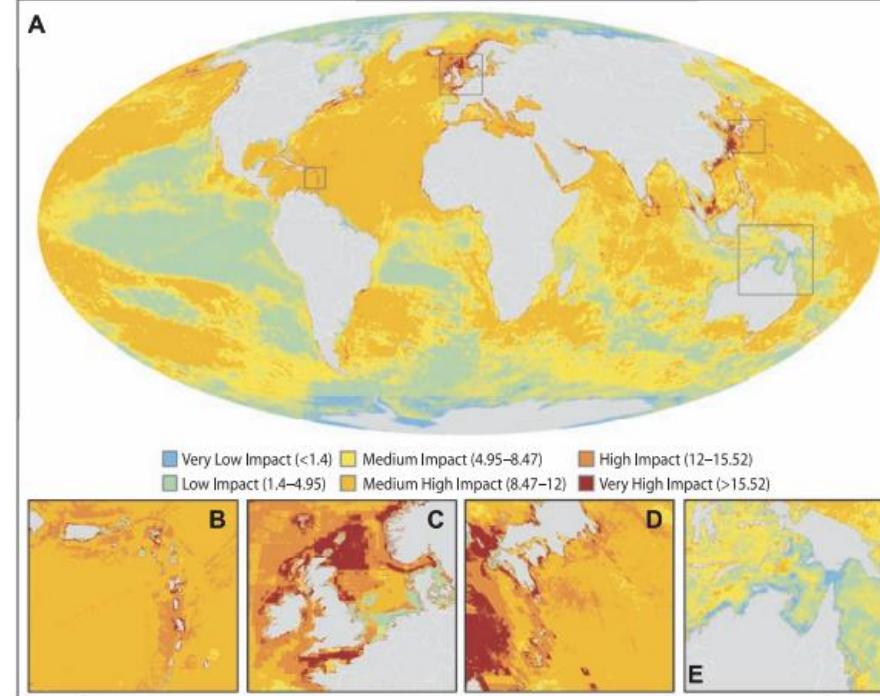


The screenshot shows the RStudio interface with the file "00-packages.R" open. The code in the editor is as follows:

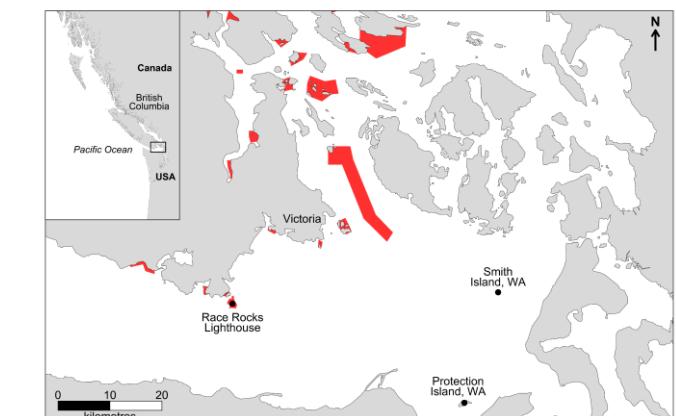
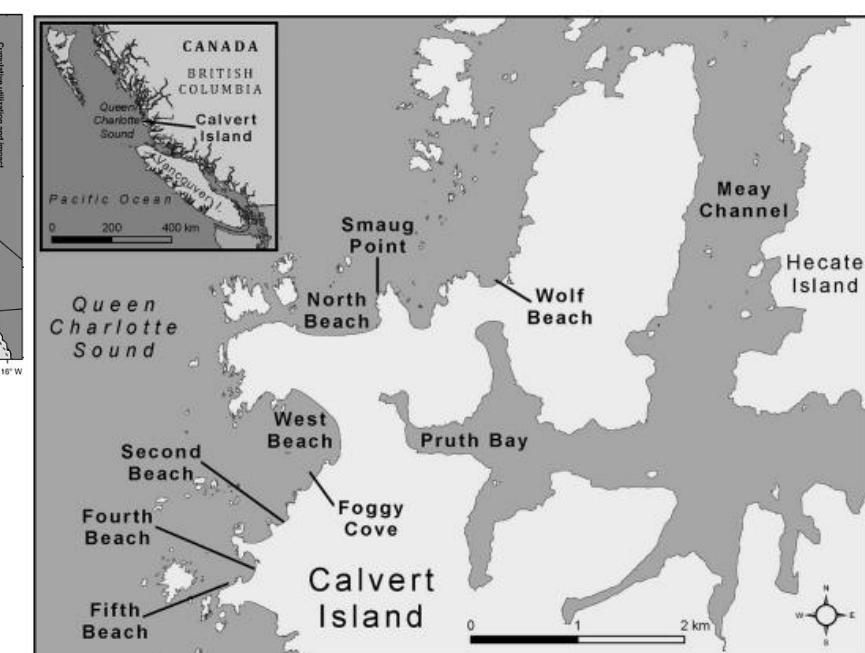
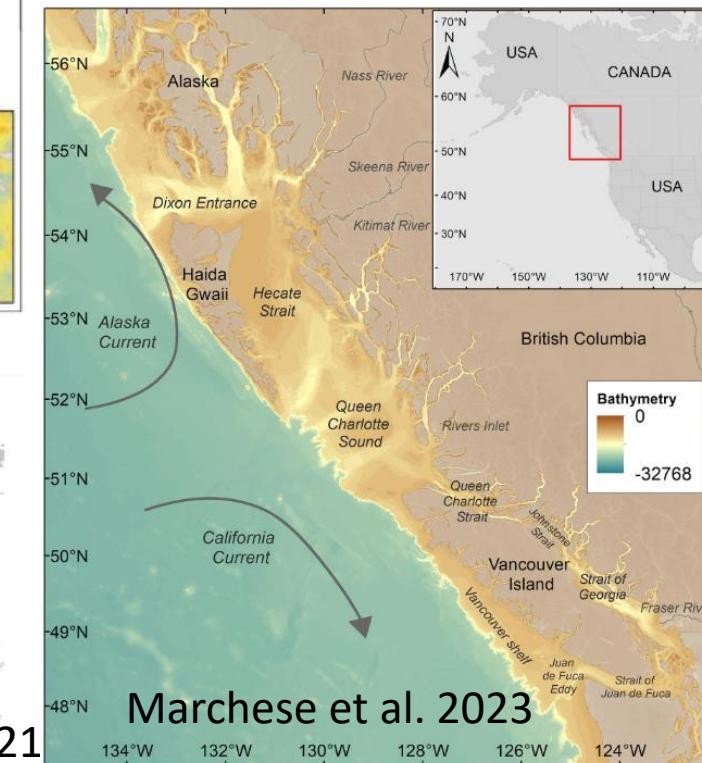
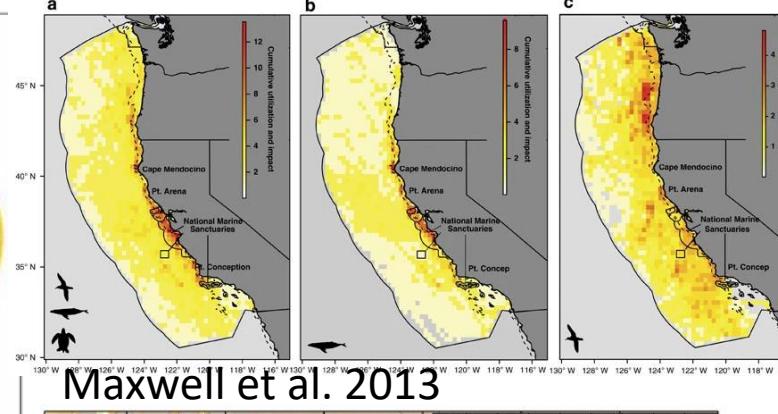
```
1 # Install required packages
2 # if you have some but not all, feel free to only install those you don't
3 # have currently
4
5 install.packages(c("sf", "stars", "tidyverse",
6                     "basemaps", "rphylopic", "ganimate",
7                     "terra", "tidyterra", "palmerpenguins",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

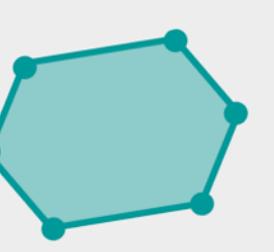
Points

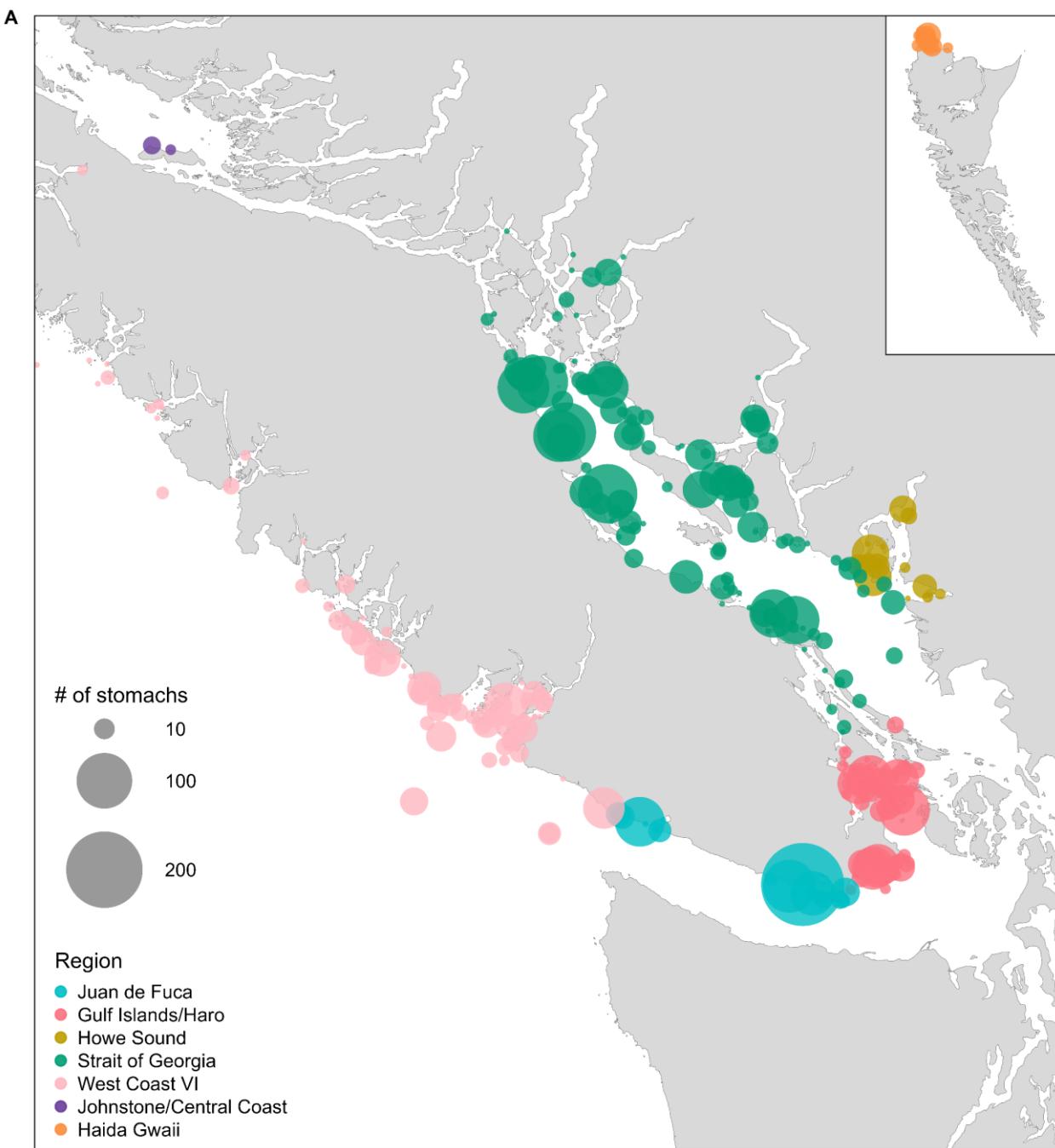


Lines



Polygons





Types of spatial data

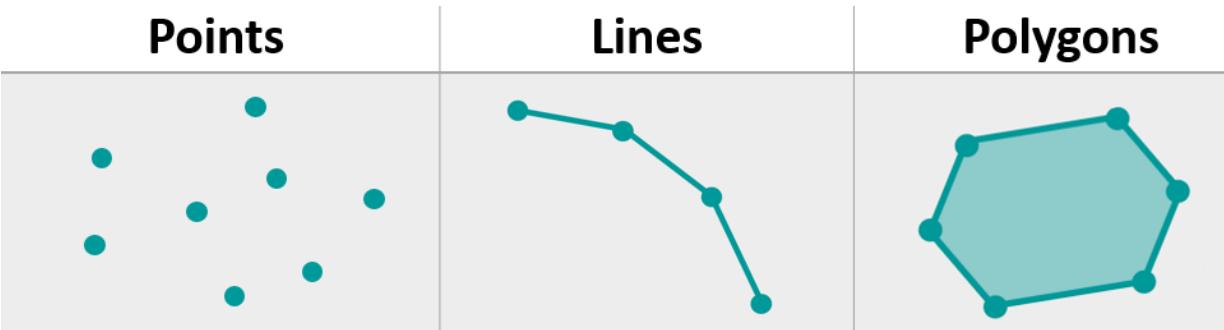
Vectors

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

Points

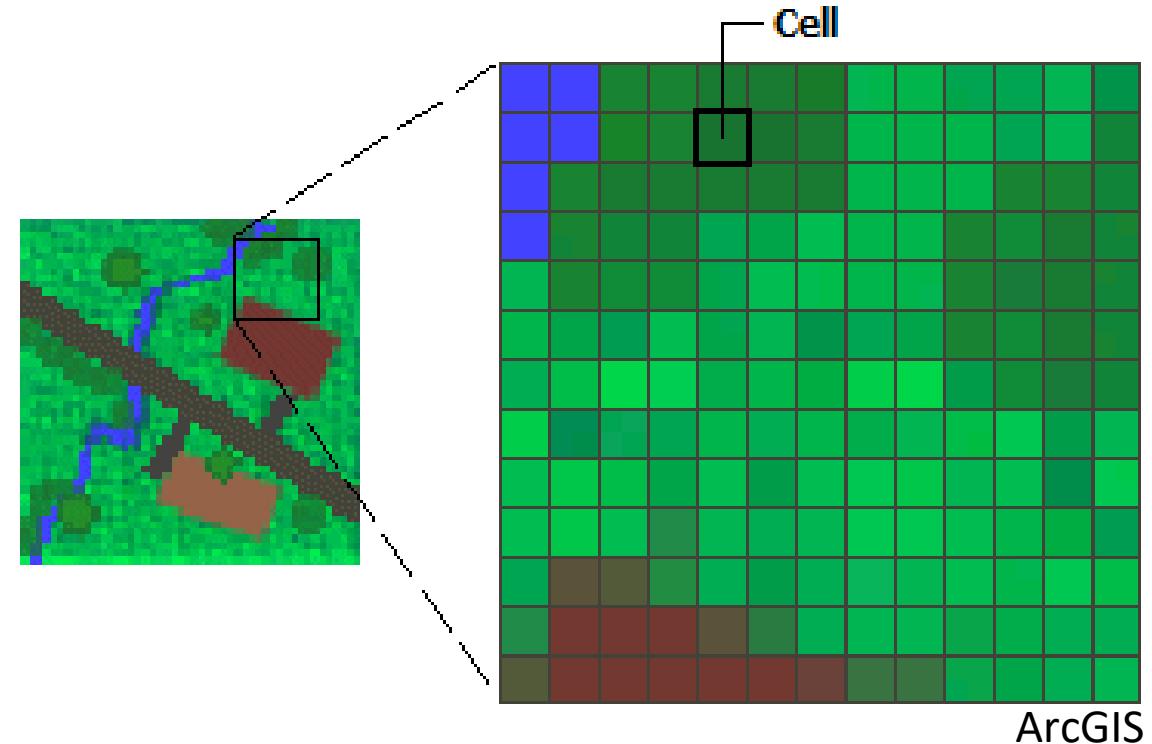
Lines

Polygons



Rasters

pixels

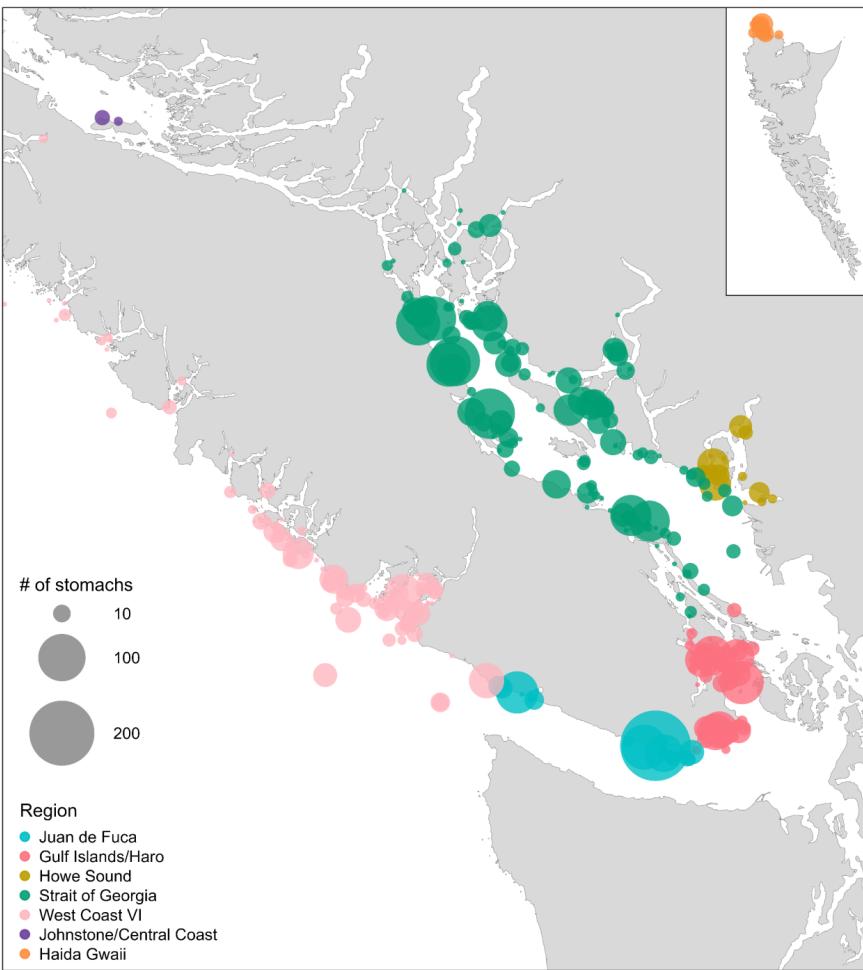




Types of spatial data

Vectors

polygons, points, lines

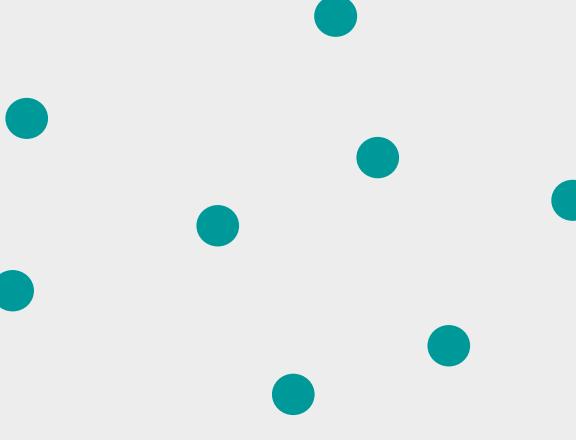
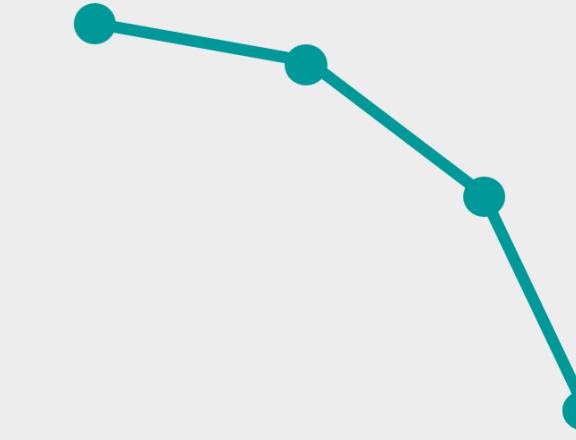
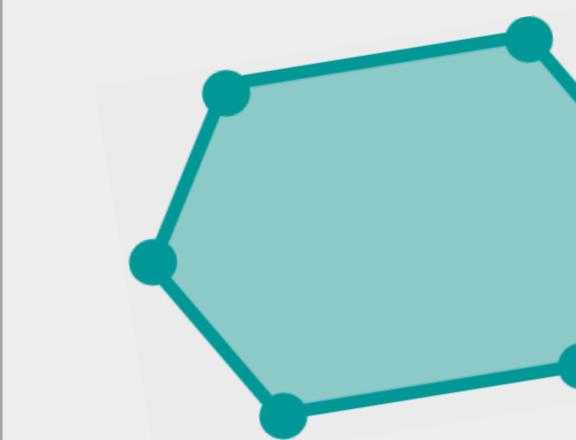


Rasters

pixels
e.g., satellite images



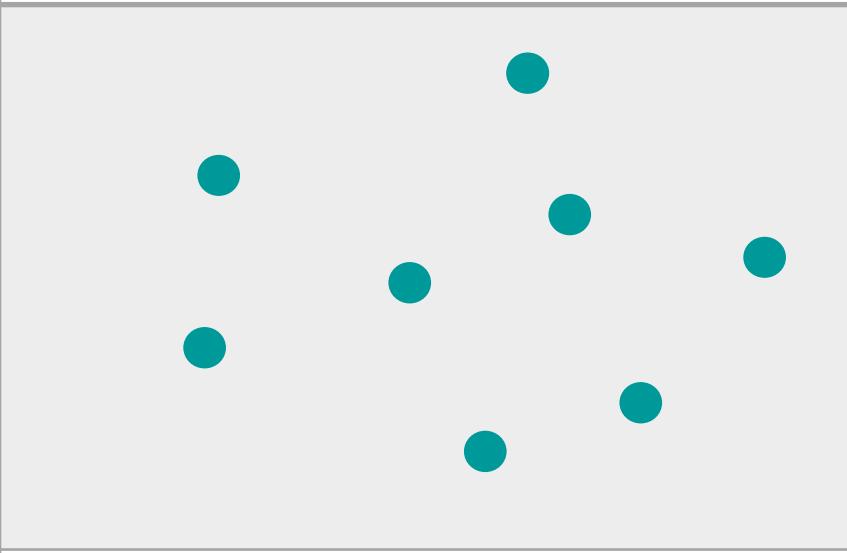
Types of vector data

Points	Lines	Polygons
		
<p>individual x, y positions</p> <p>examples:</p> <ul style="list-style-type: none">sampling locationsspecies presence	<p>lines connect 2+ points</p> <ul style="list-style-type: none">migration pathsroads	<p>3+ points that connect and close</p> <ul style="list-style-type: none">coastlinesprotected areas



Types of vector data

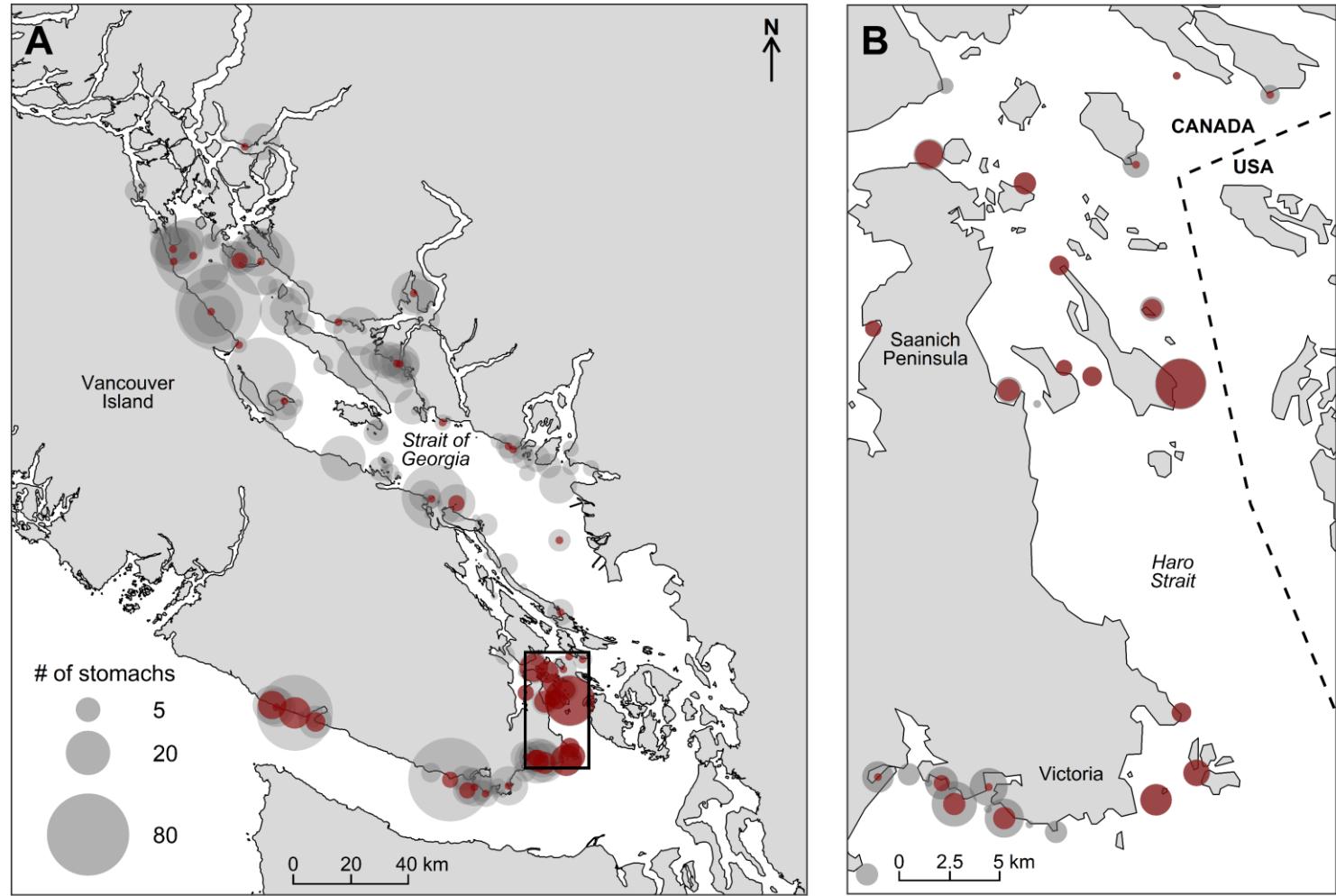
Points



individual x, y positions

examples:

sampling locations
species presence



Point size corresponds to number of salmon stomachs containing sand lance

Robinson et al. 2023

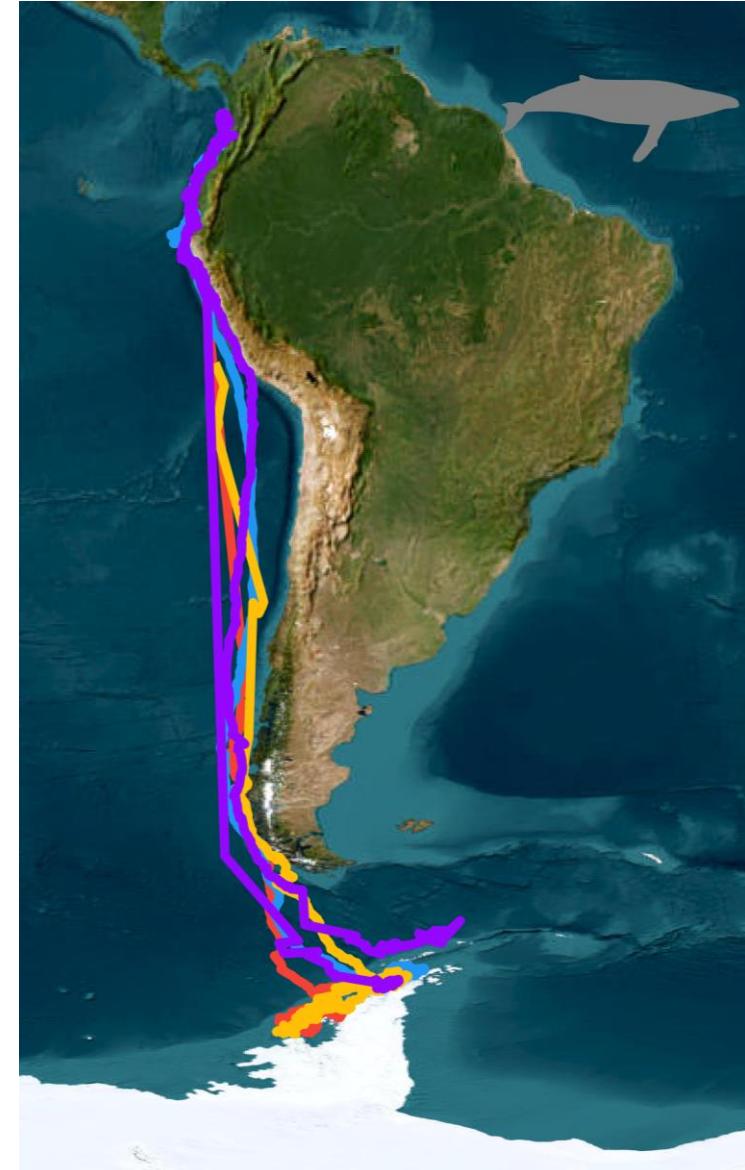
Types of vector data

Lines



lines connect 2+ points

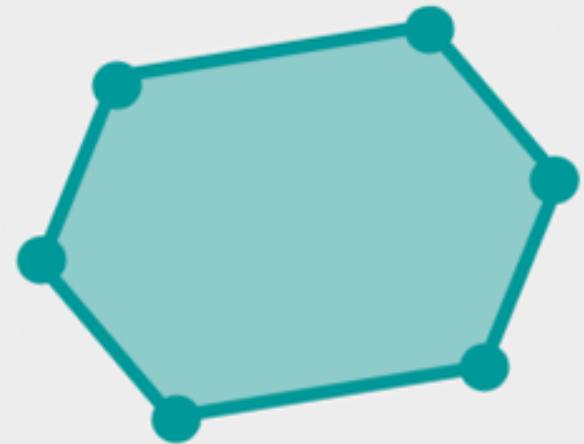
migration paths
roads



Humpback whale migration tracks
along South America

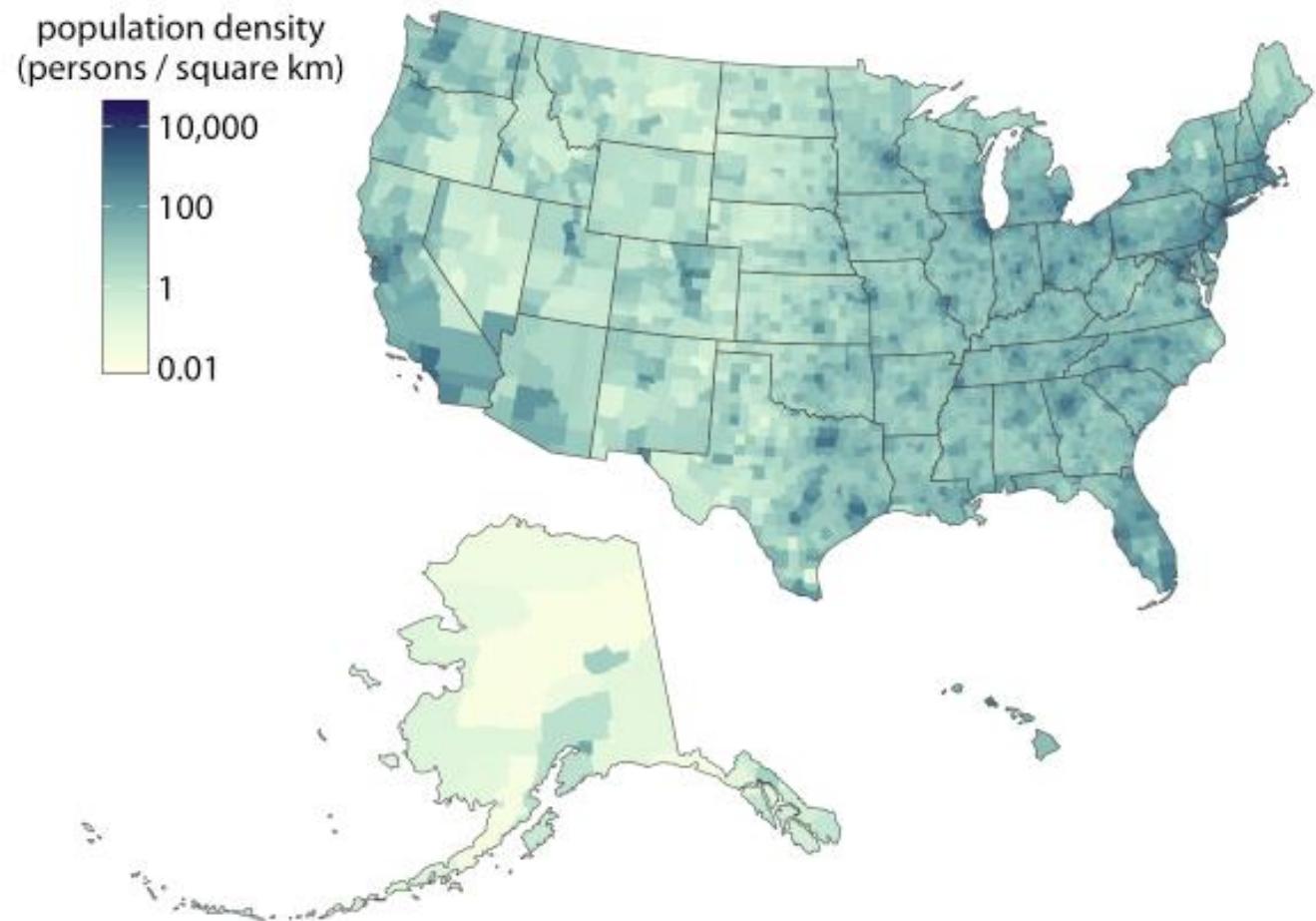
Types of vector data

Polygons



3+ points that connect
and close

coastlines
protected areas



Why use R for maps?

Challenges with GIS software:

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



ArcGIS
QGIS



Why use R for maps?

Challenges with GIS software:

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

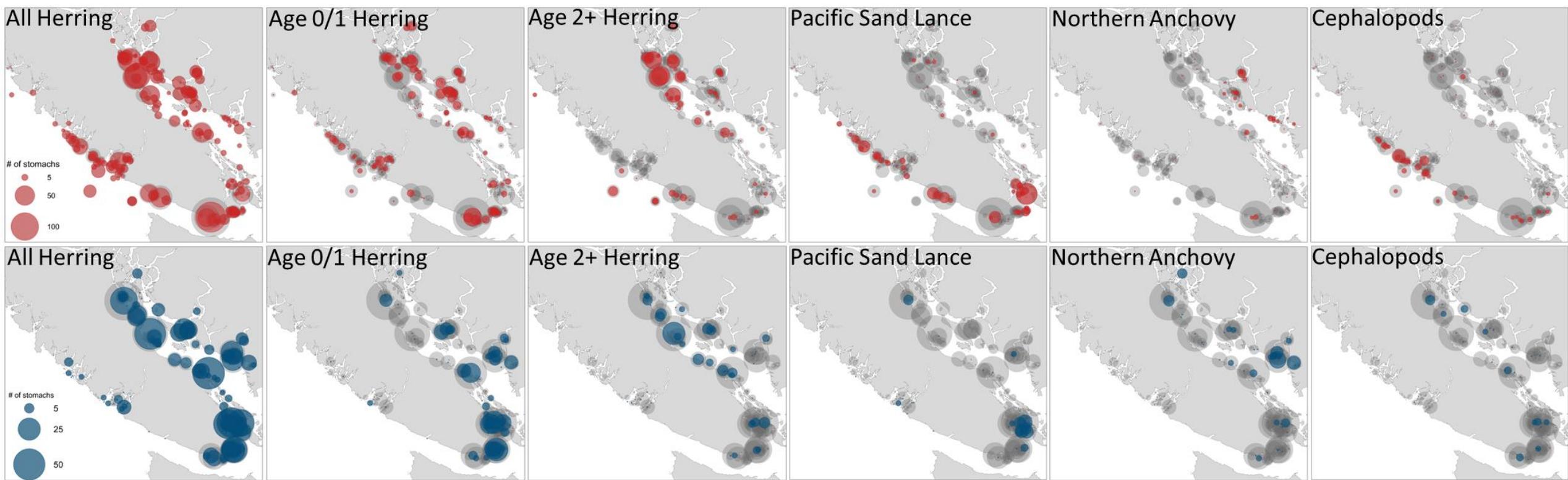
Making maps in R:

- **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)



ArcGIS
QGIS





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()
```

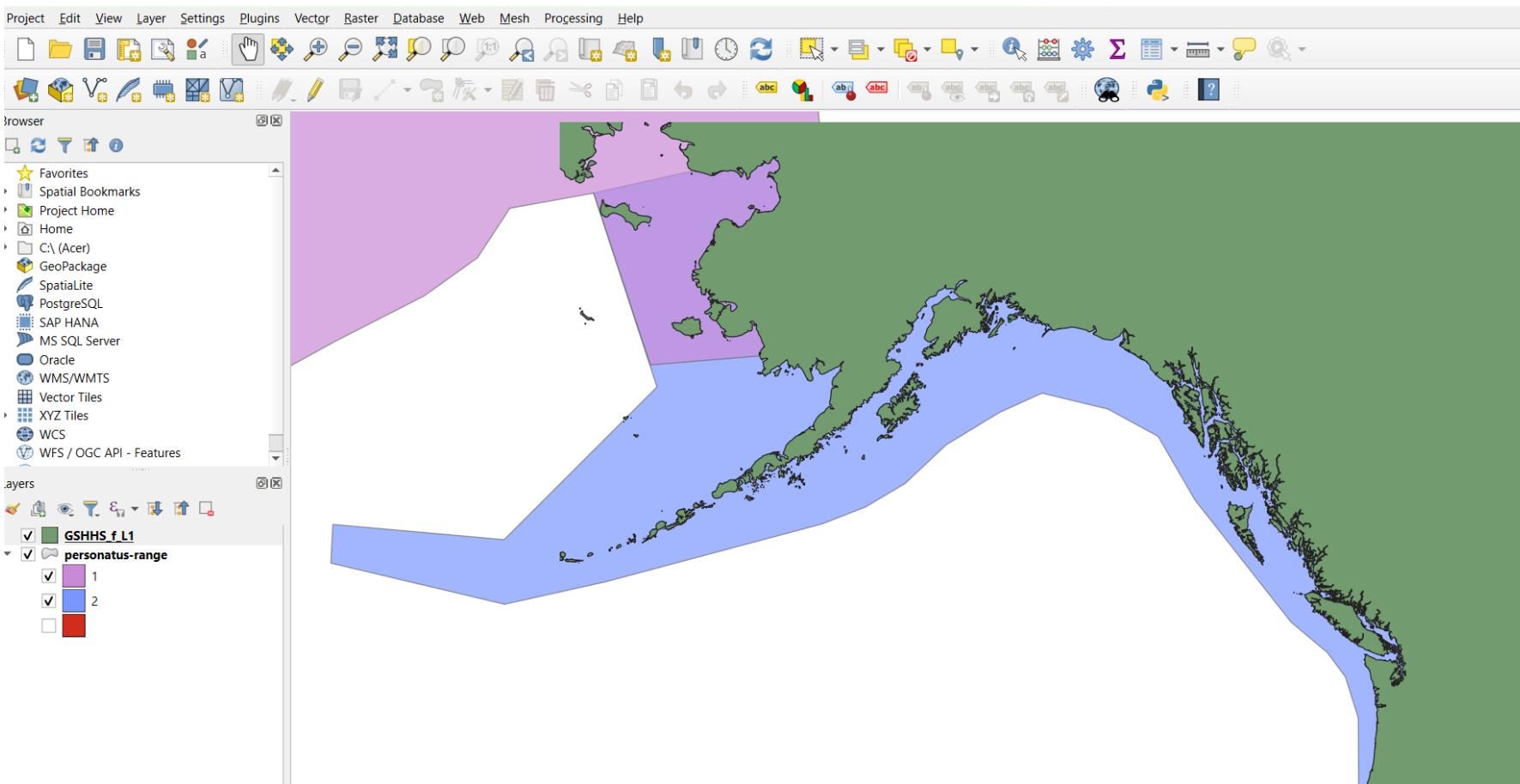
Data analysis

```
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

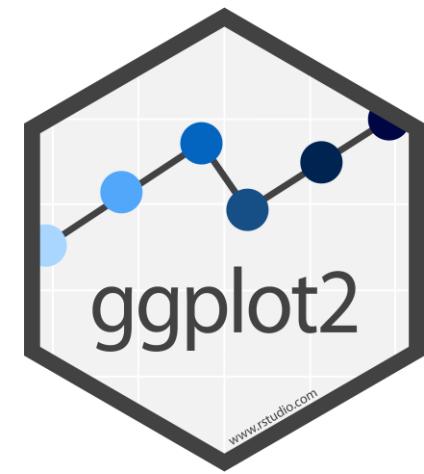
ggplot() makes plotting easier

Functions in R

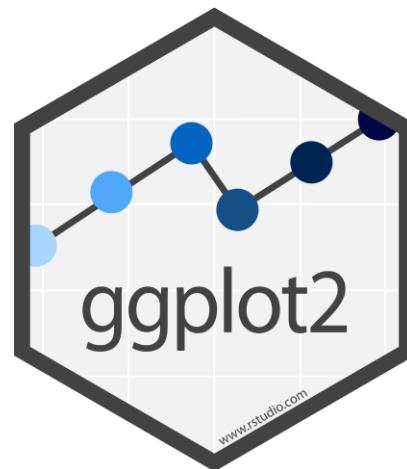
- R uses functions
 - plot() makes graphs
- Some functions are automatically loaded in R (**“base”**)
- R software developers build **packages**, which include their functions

```
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



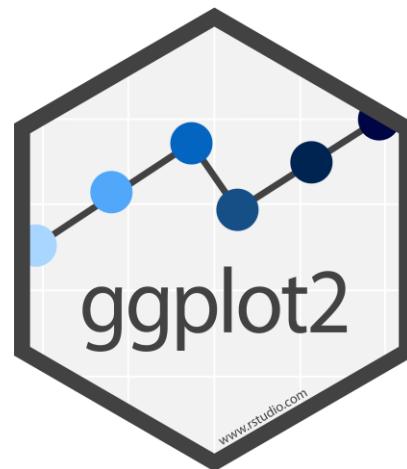
Create figures with multiple layers

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:

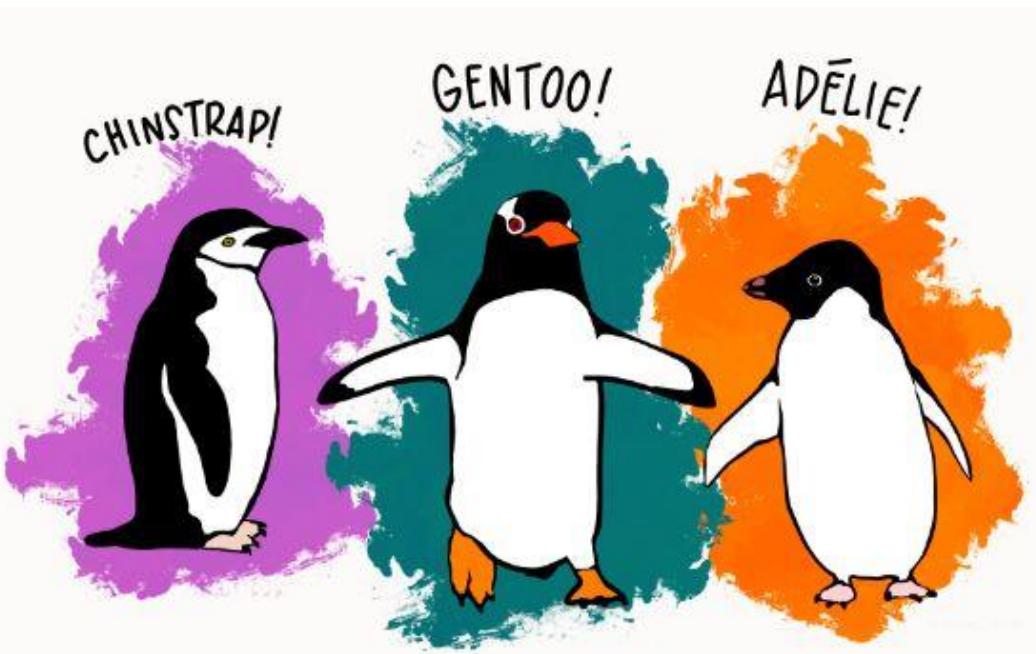
- `geom_point()` scatterplots, spatial points
- `geom_line()`, `geom_path()` line graphs, spatial lines
- `geom_sf()` shapefiles (spatial polygons)
- `geom_label()` print text labels on figure

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

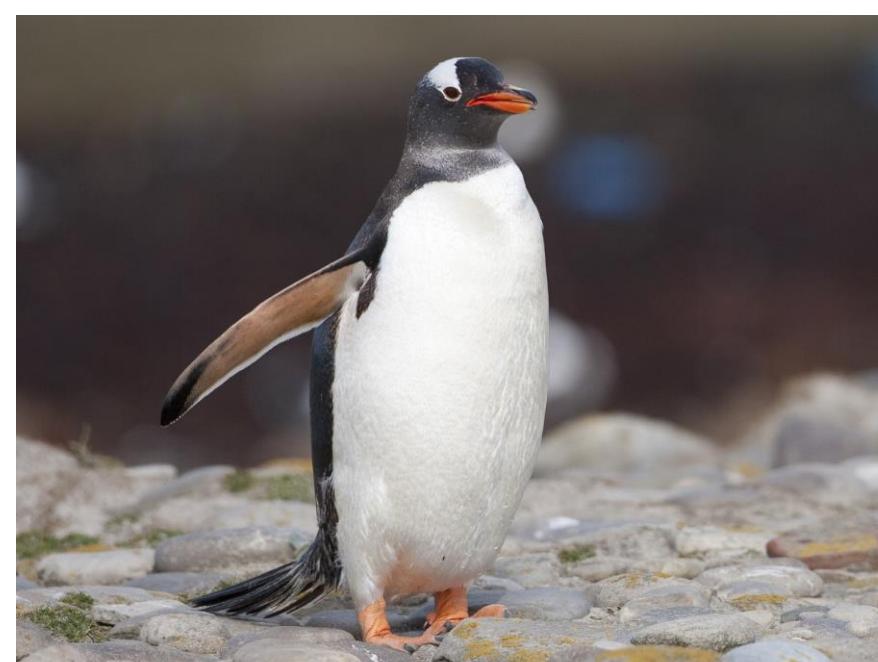
The code snippet shows the `geom_point()` function within a `ggplot()` call. Two specific arguments are highlighted: `data = df` and `aes(x = X, y = Y, color = group)`. An arrow points from the label "data" to the `data = df` argument, and another arrow points from the label "aesthetics" to the `aes(x = X, y = Y, color = group)` argument.

ggplot() example

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



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



eBird



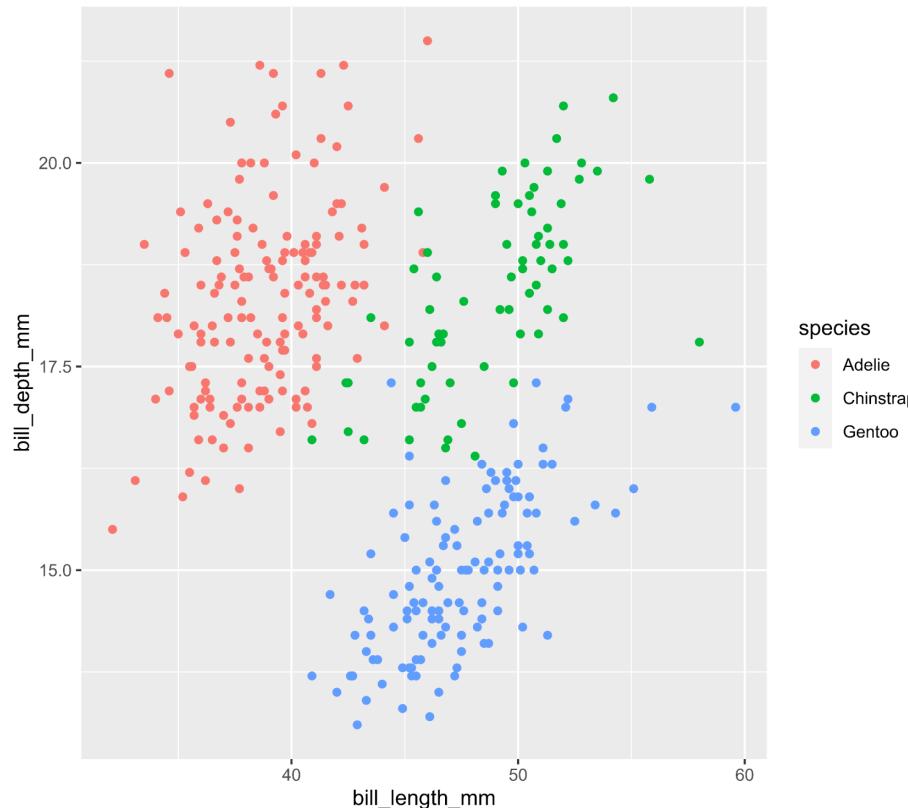
eBird



Australian Museum

ggplot() example

- palmerpenguins dataset
- scripts/01-ggplot-example.R
- morphological data for 3 penguin species
- example: bill depth vs bill length

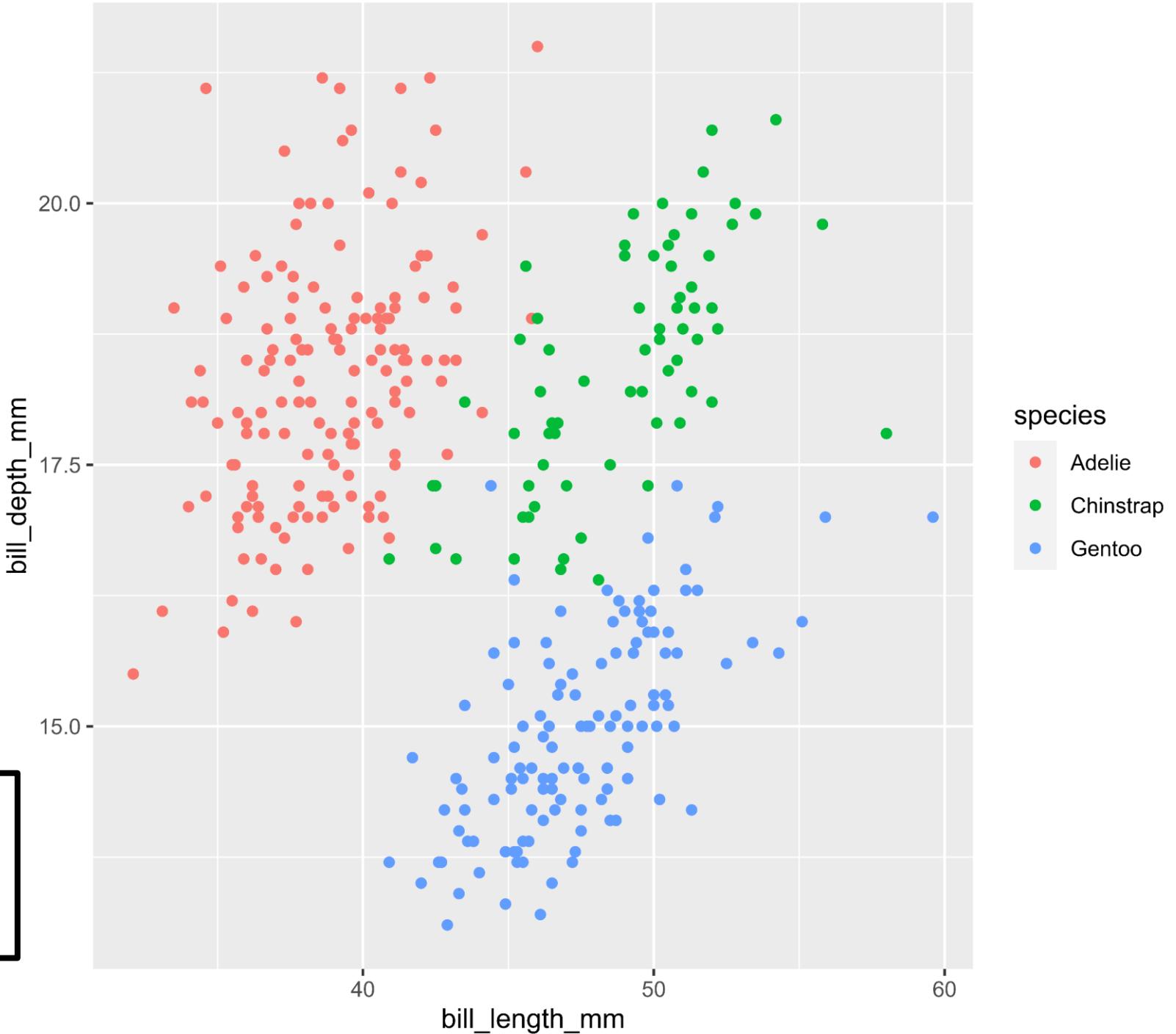


```
data(penguins)

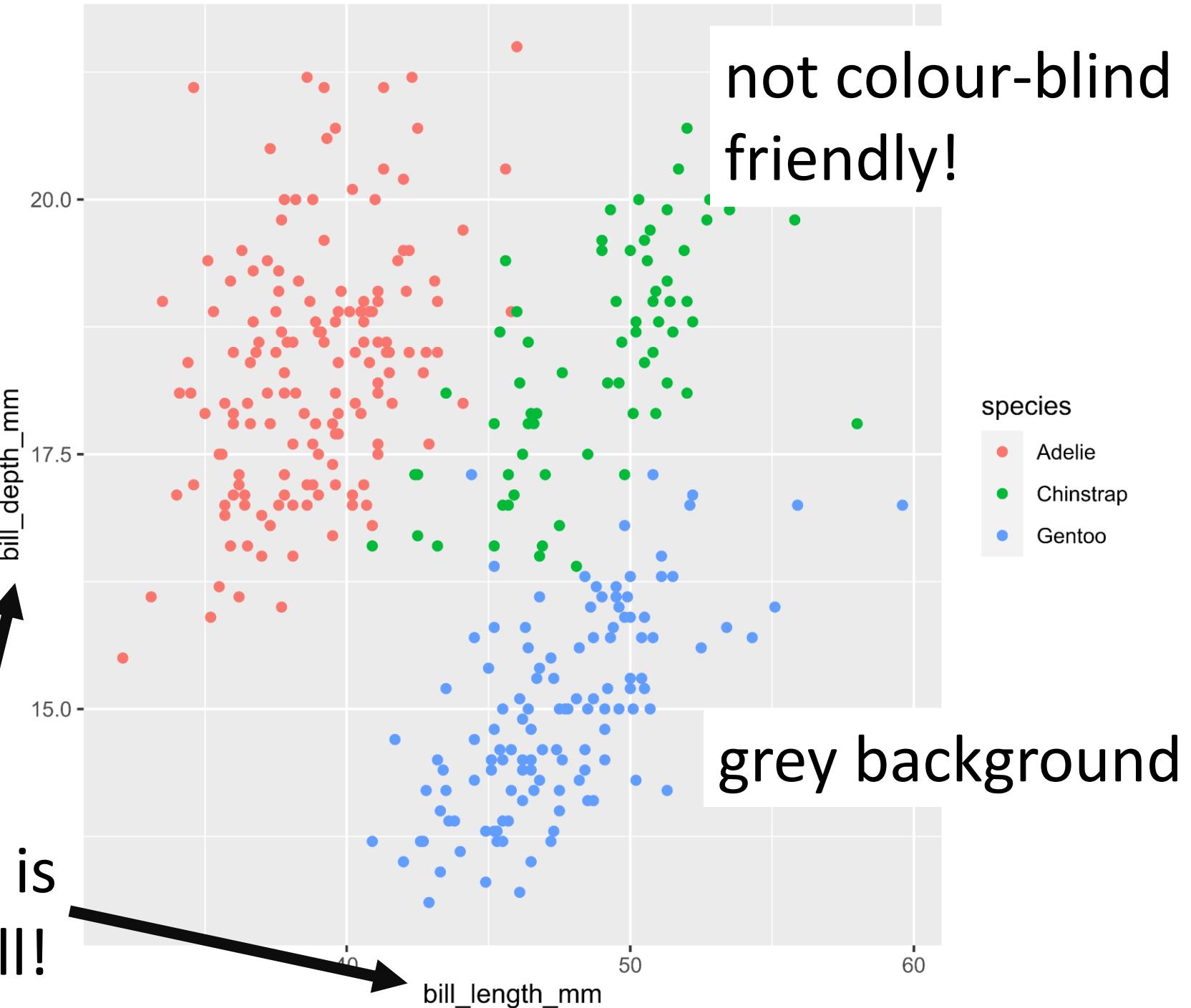
ggplot() +
  geom_point(data = penguins,
             aes(x = bill_length_mm, y = bill_depth_mm, color = species))
```

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

What do you think
should be changed?



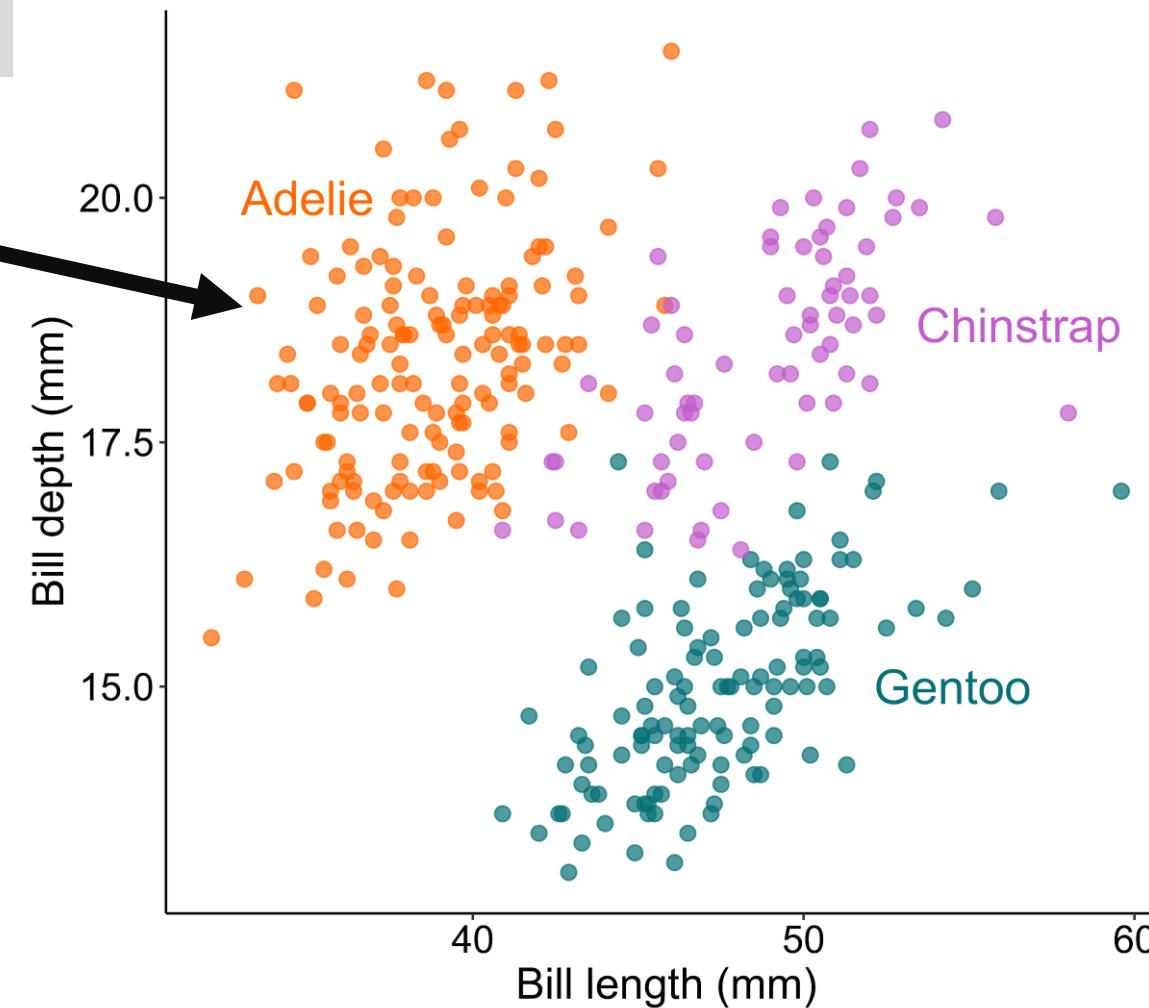
**DO NOT USE
ggplot()
defaults!**



```
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

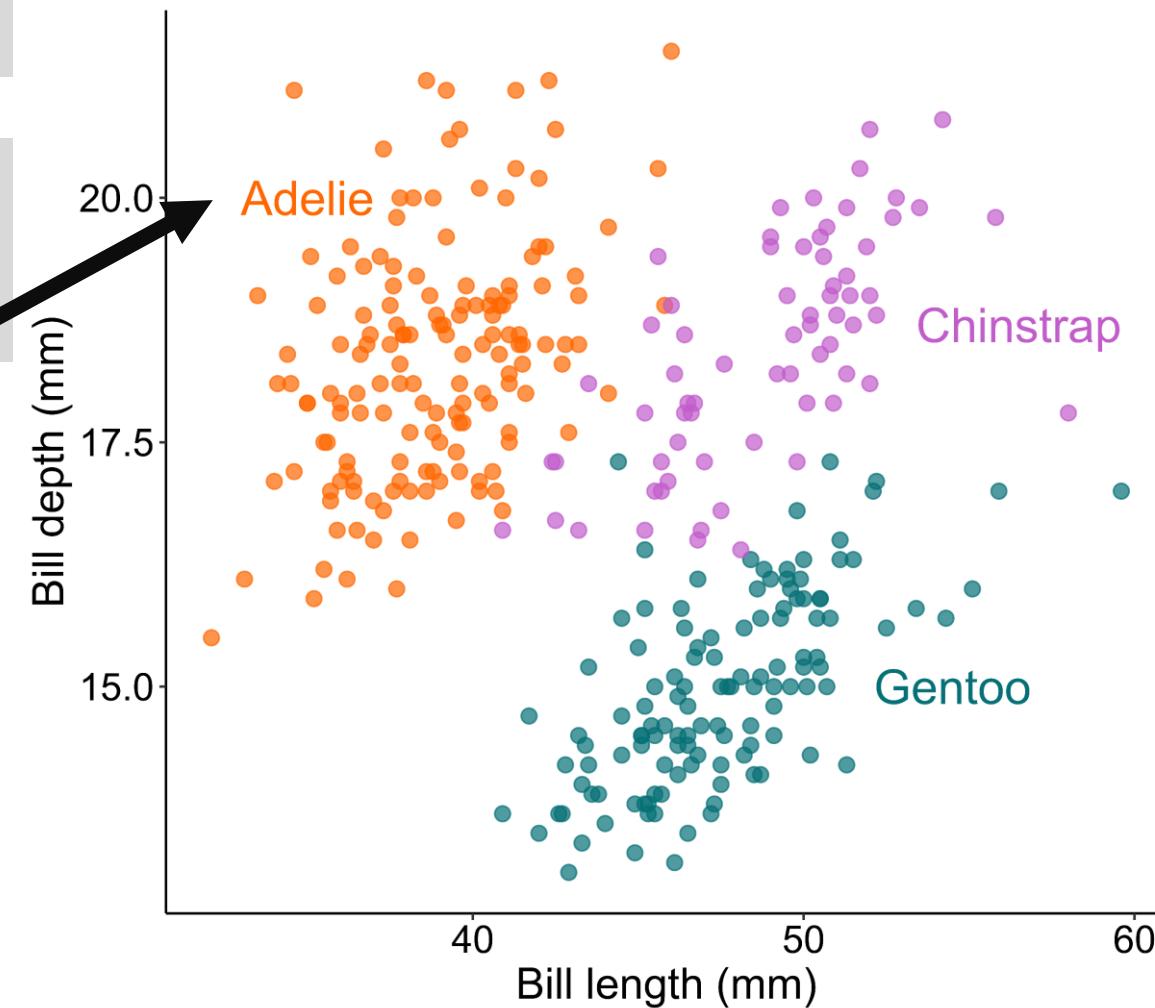


```
ggplot() +  
  geom_point(data = penguins,  
             aes(x = bill_length_mm, y = bill_depth_mm, color = species))
```

```
  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))
```

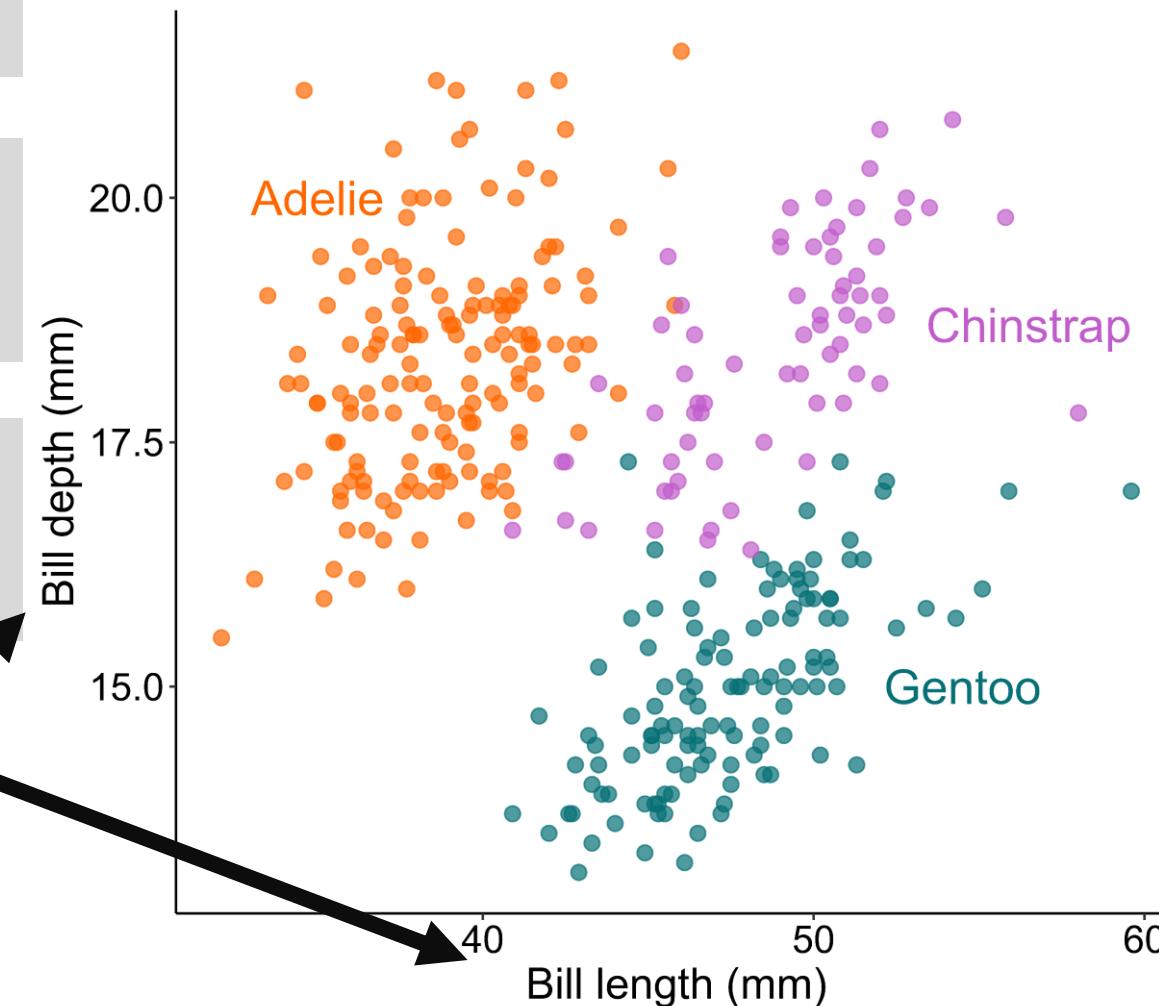
```
  geom_point(size = 2.7, alpha = 0.7)
```

```
  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))
```

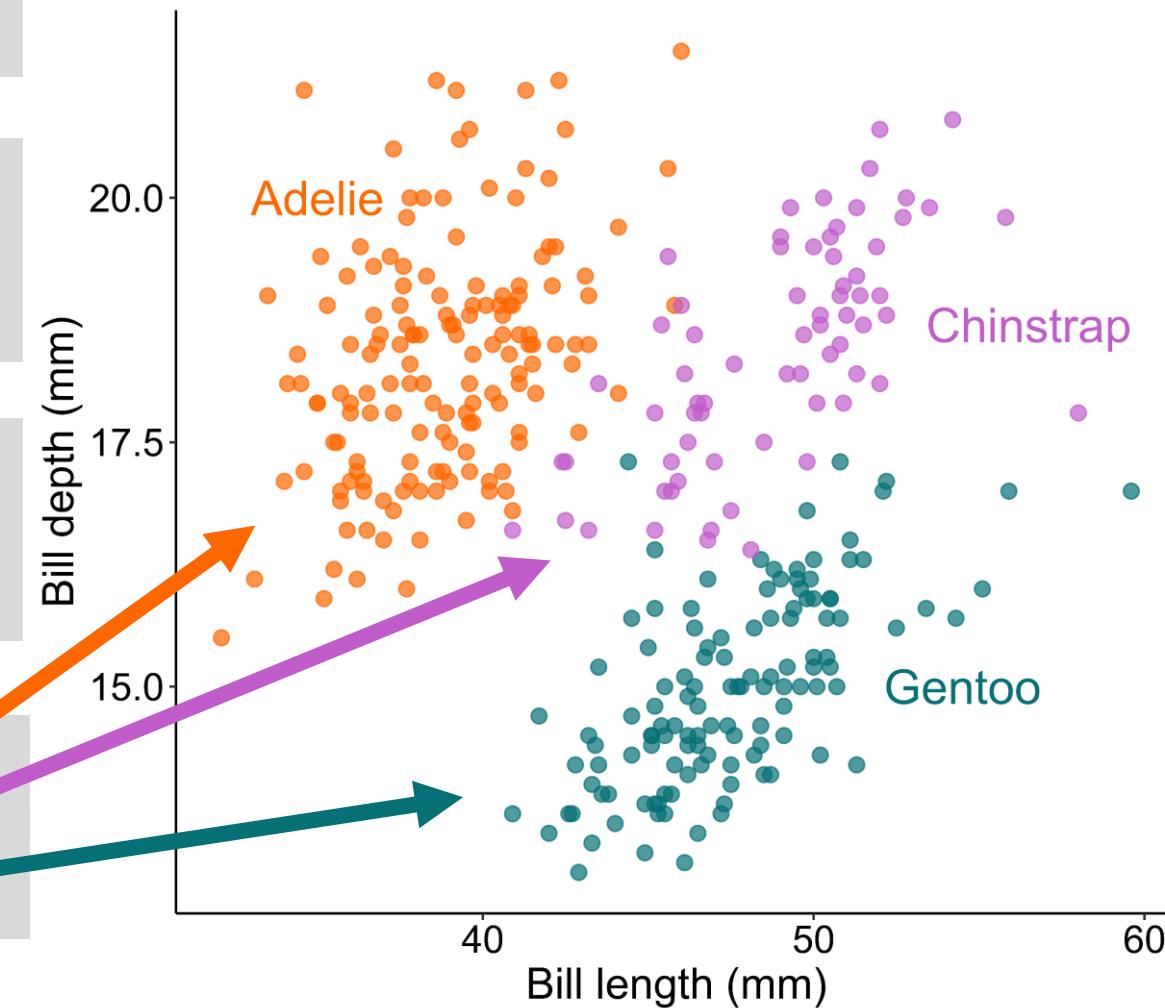
```
  geom_point(size = 2.7, alpha = 0.7)
```

```
  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)
```

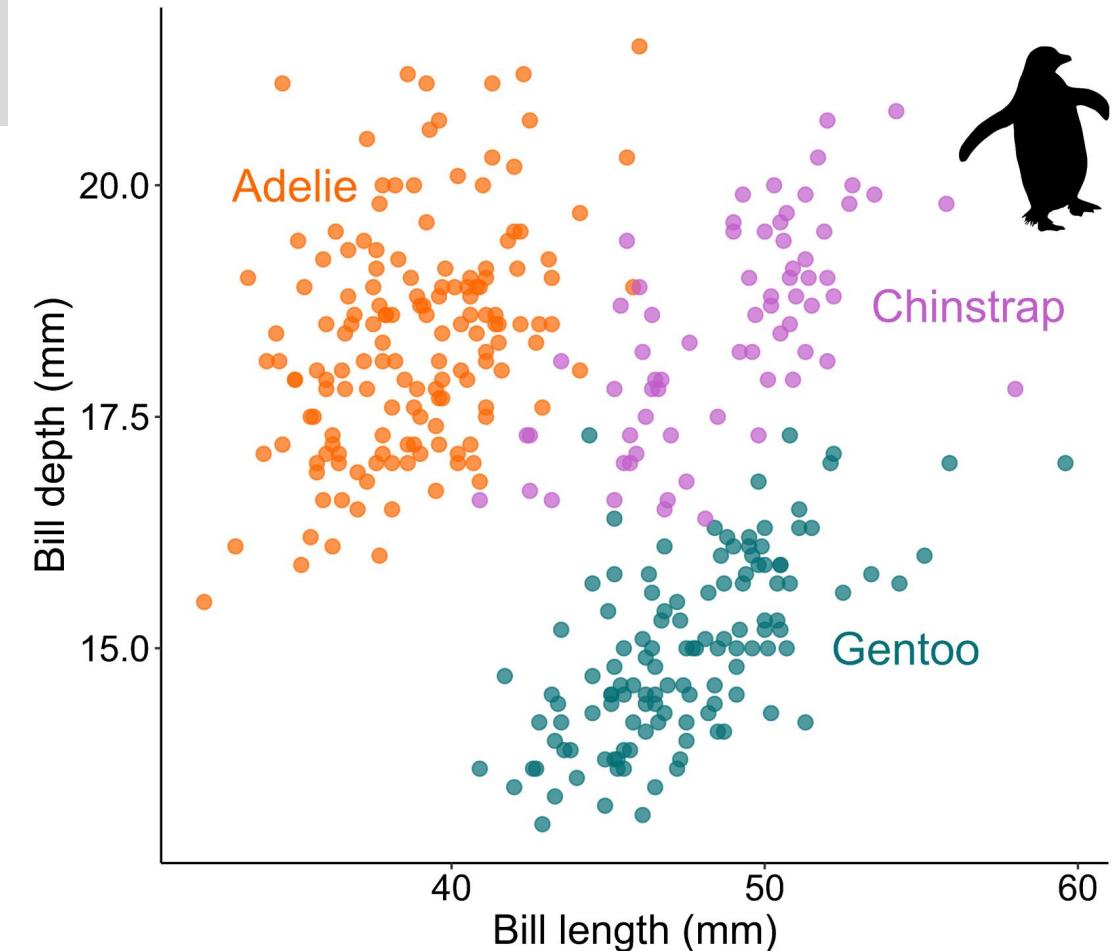
PHYLOPic Enter the name of a group of organisms.

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

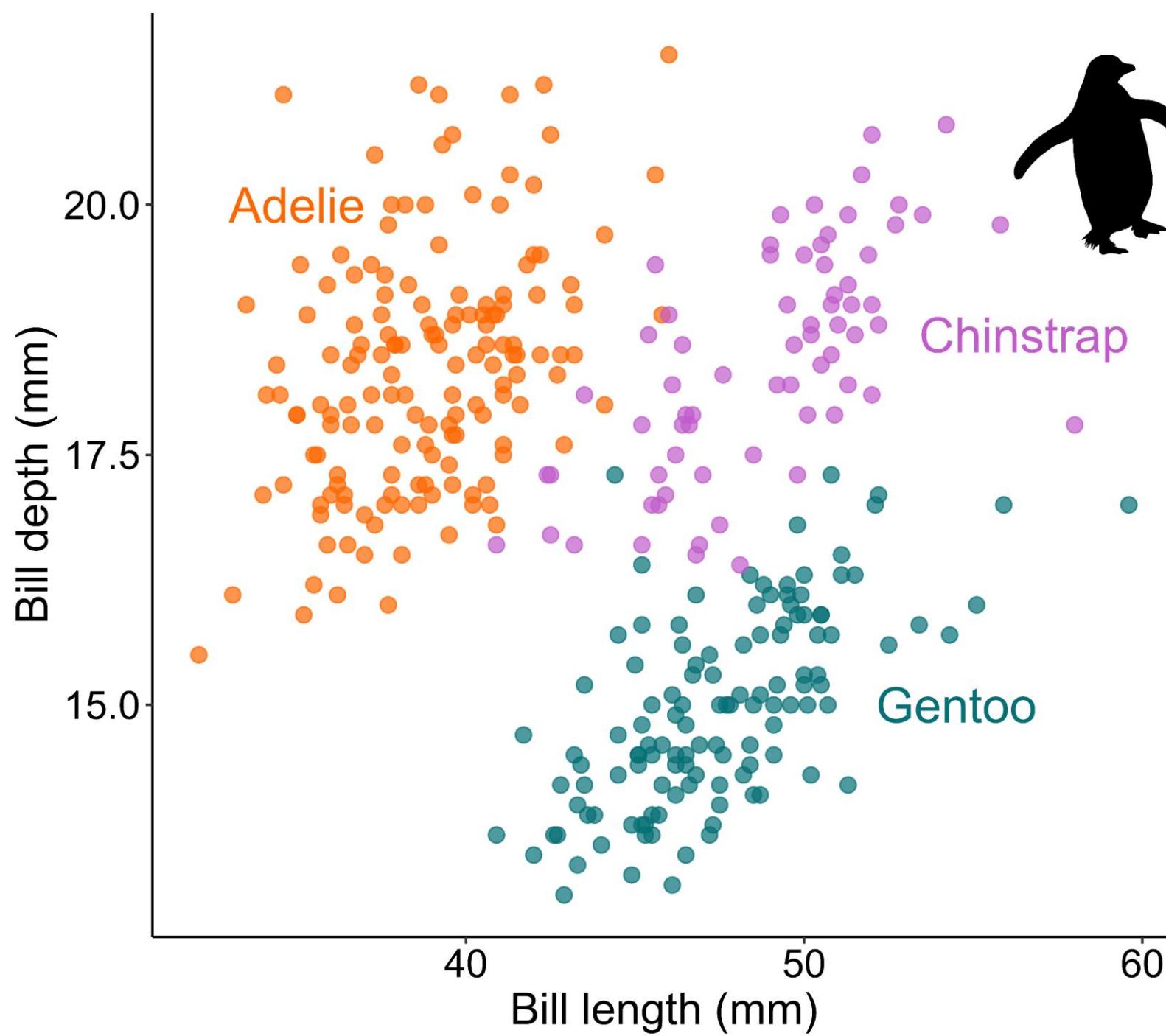
Latest Uploads

SEE MORE →

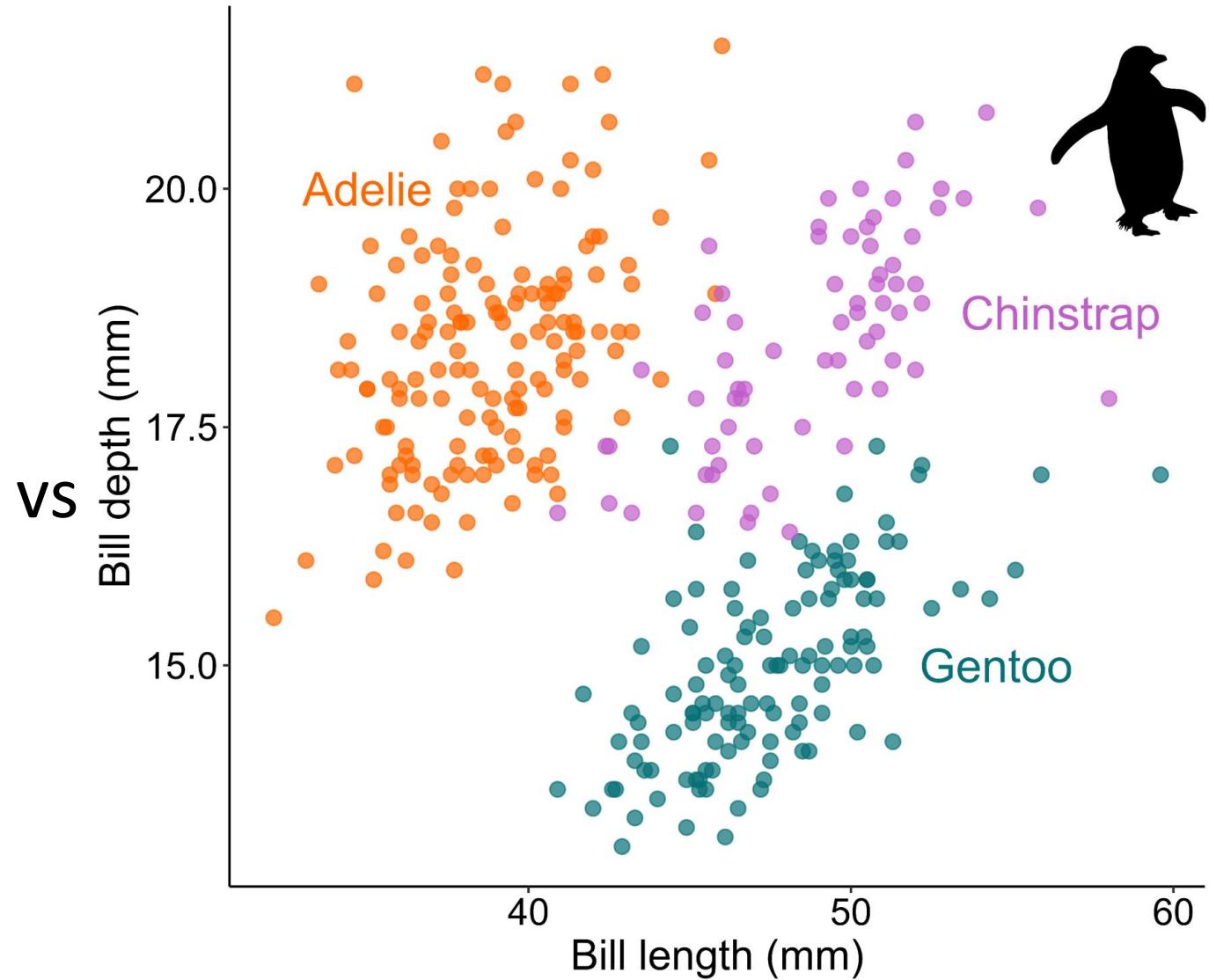
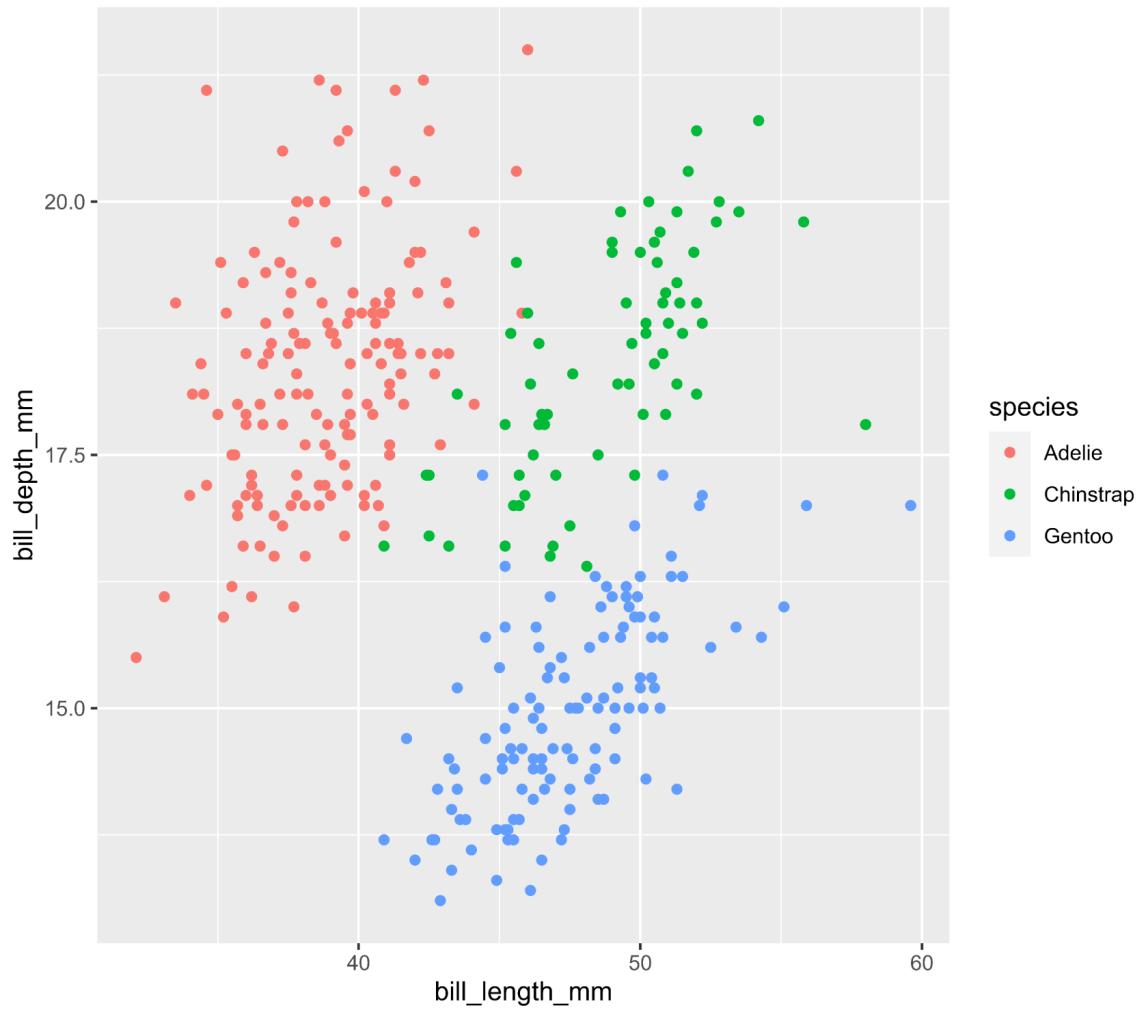
Gento penguin scientific name

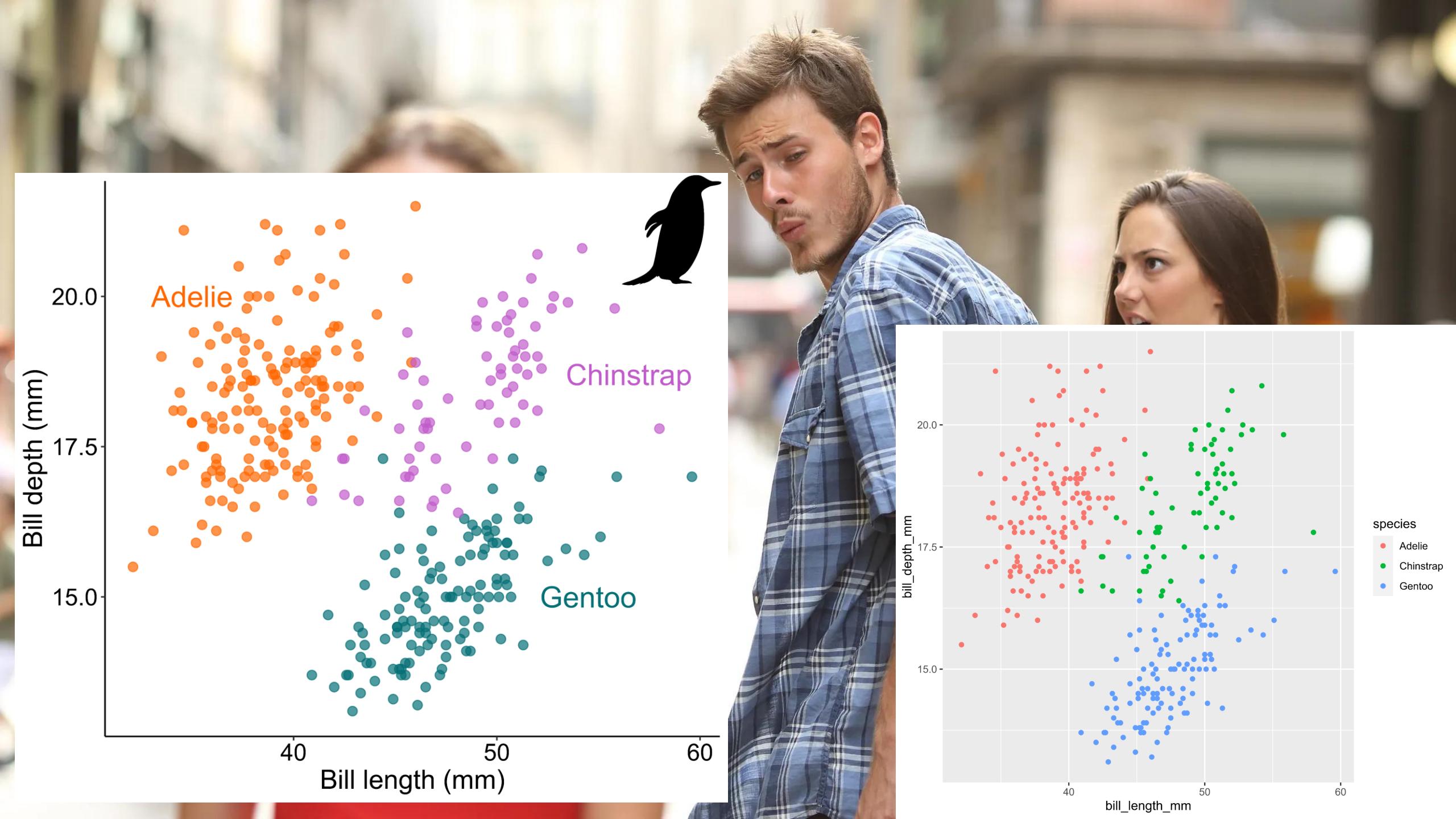


<https://www.phylopic.org/>



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()!

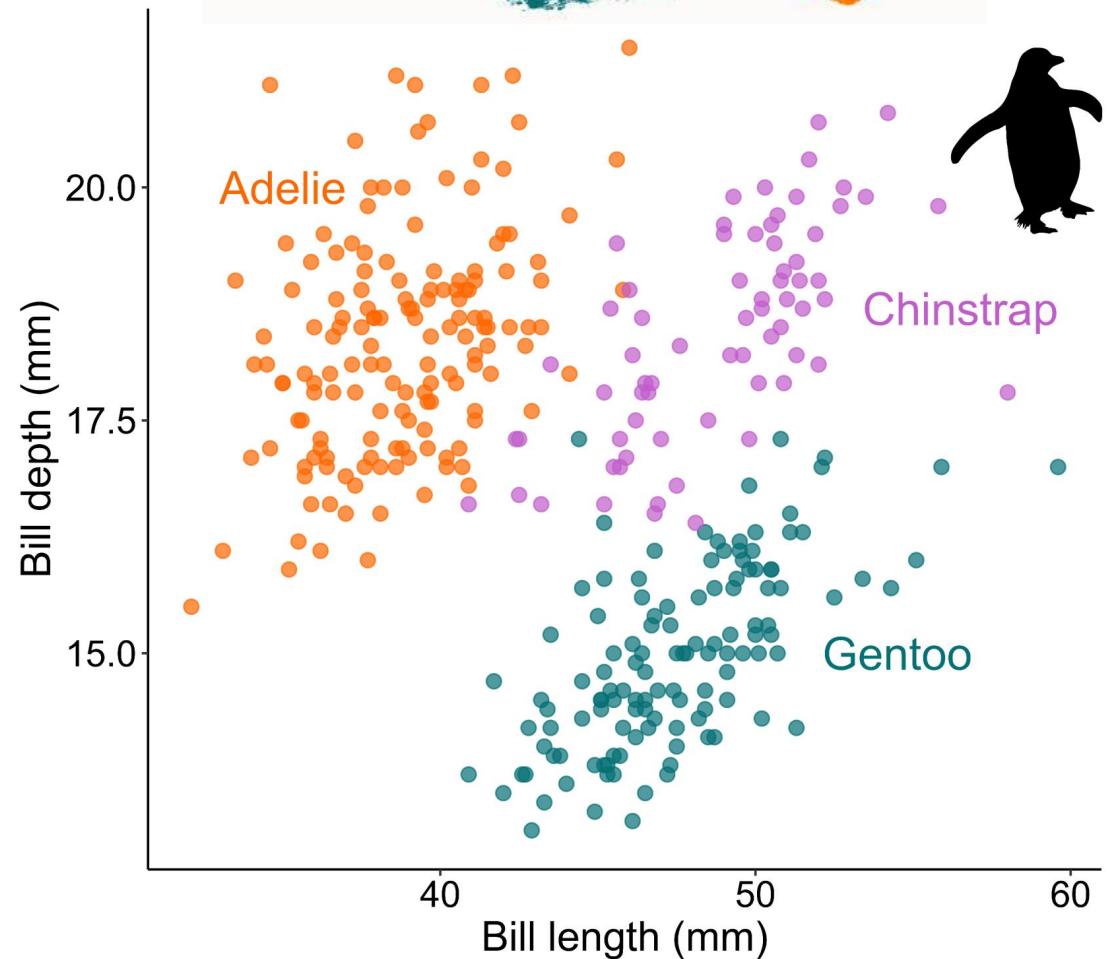
Additional resources

Hadley Wickham. R for data science

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

Hadley Wickham. ggplot2: Elegant graphics for data analysis

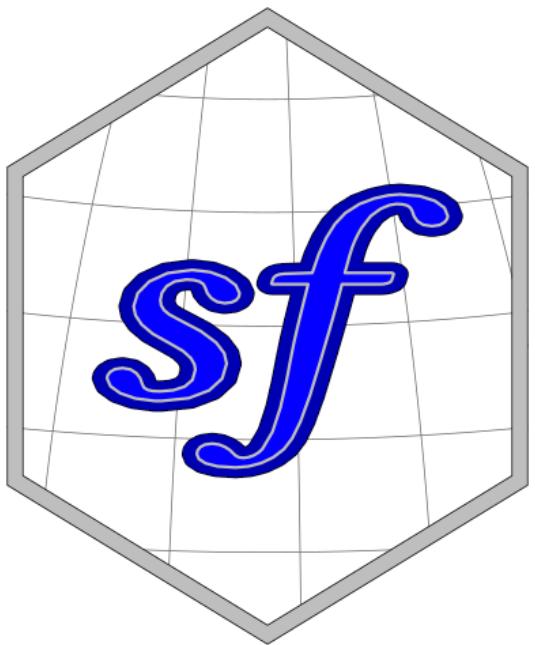
<https://ggplot2-book.org/>





Back to maps!

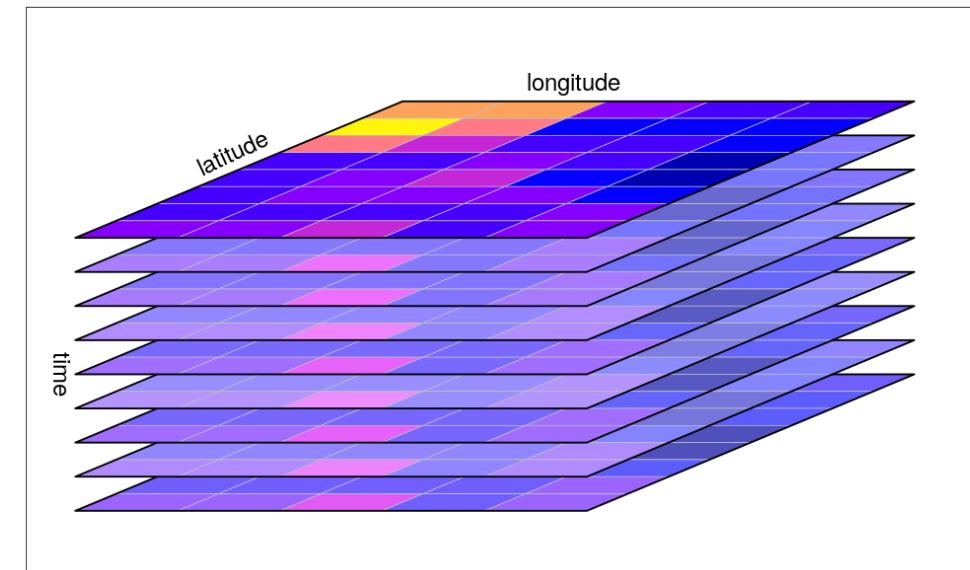
Which R packages to use?



vectors



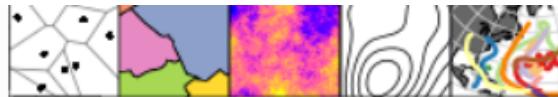
rasters



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](#)

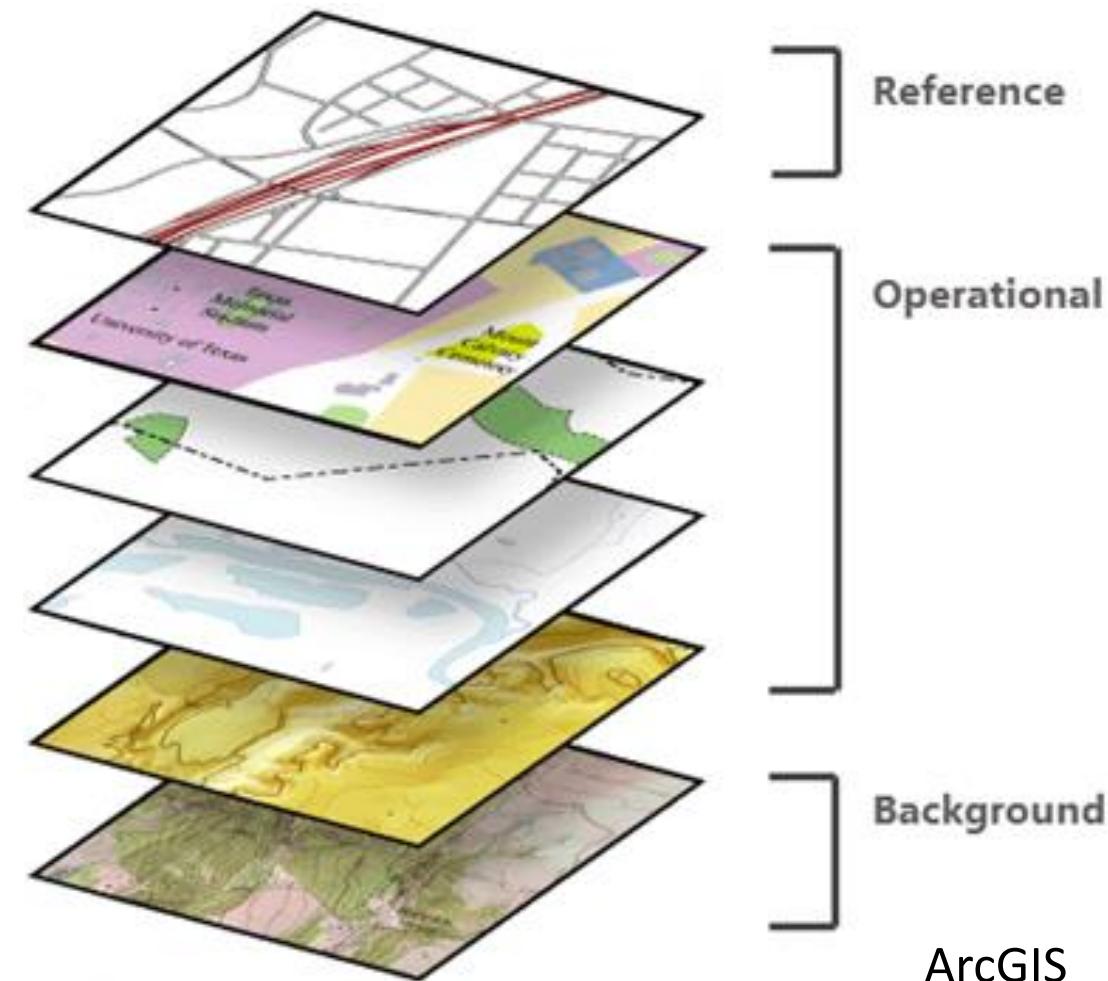
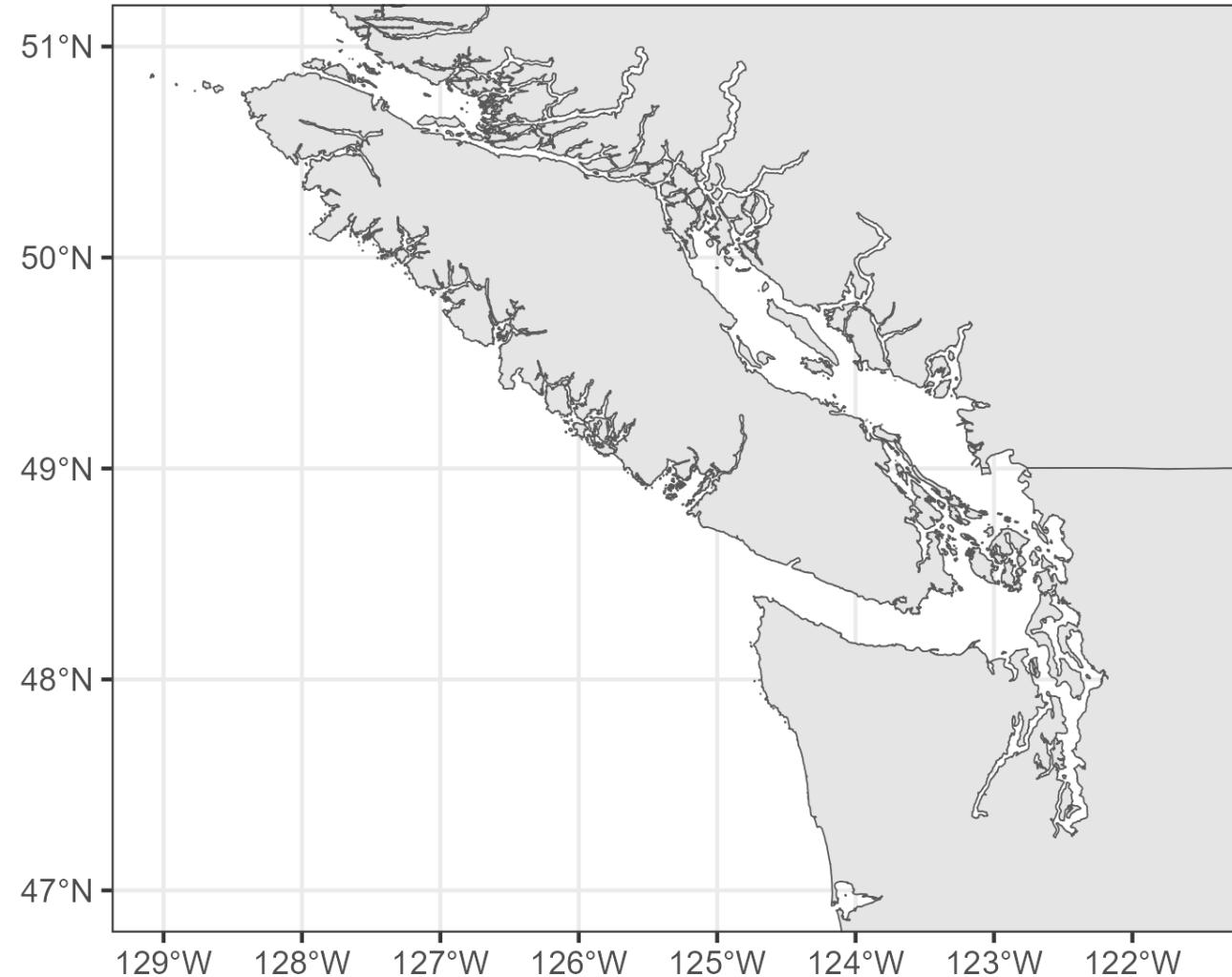
[\[view raw Rmd\]](#)

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

scripts/02-compare-basemaps.R

BC coast shapefile



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

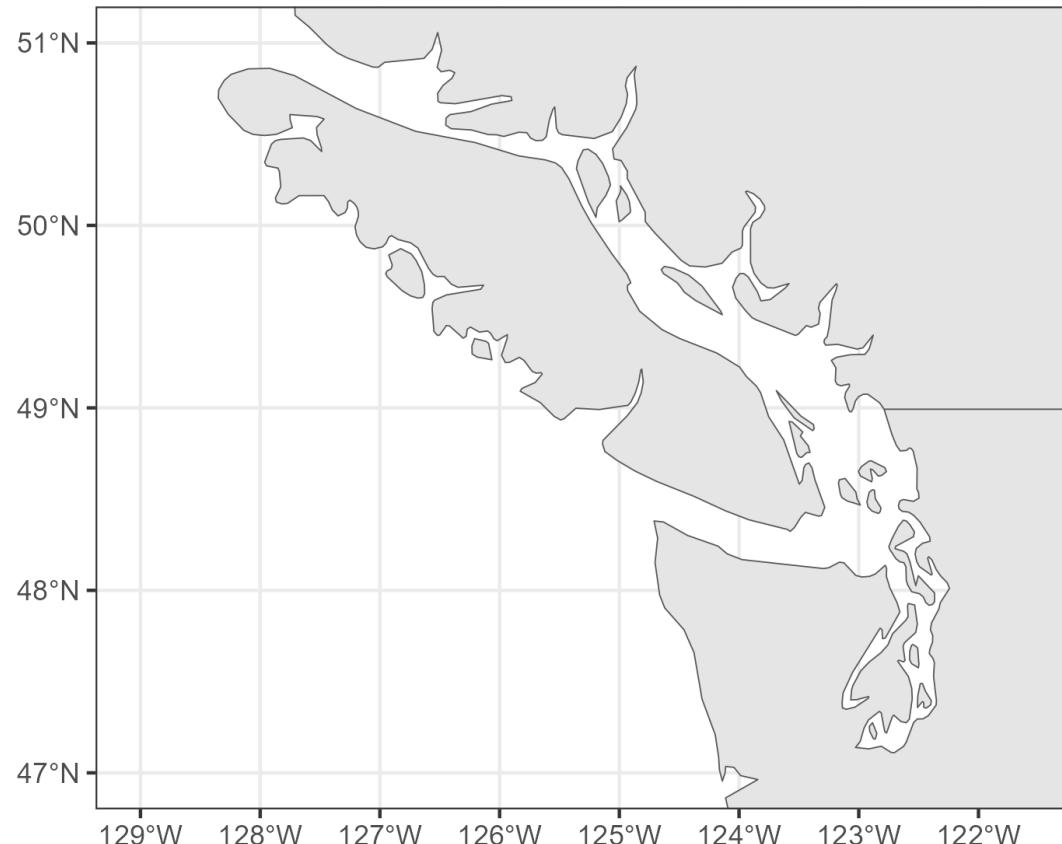
satellite and other basemaps

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

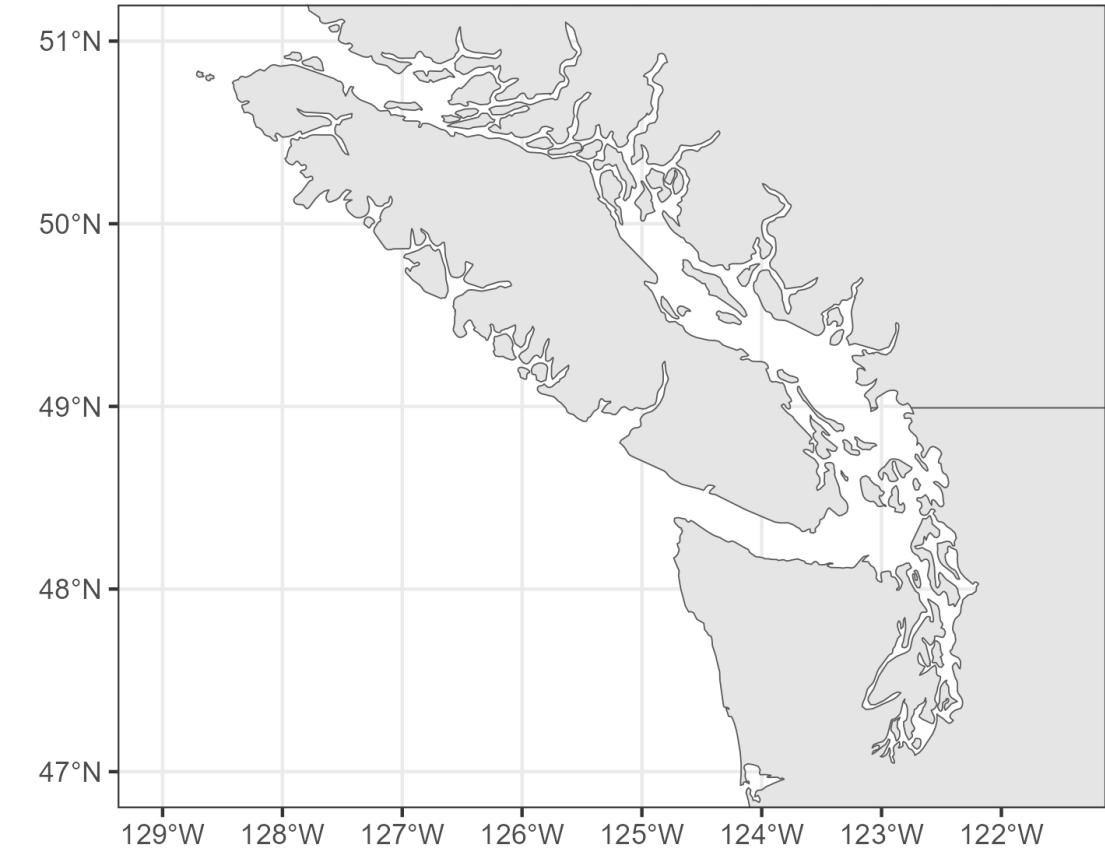
RNaturalEarth

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

RNaturalEarth 1:50M



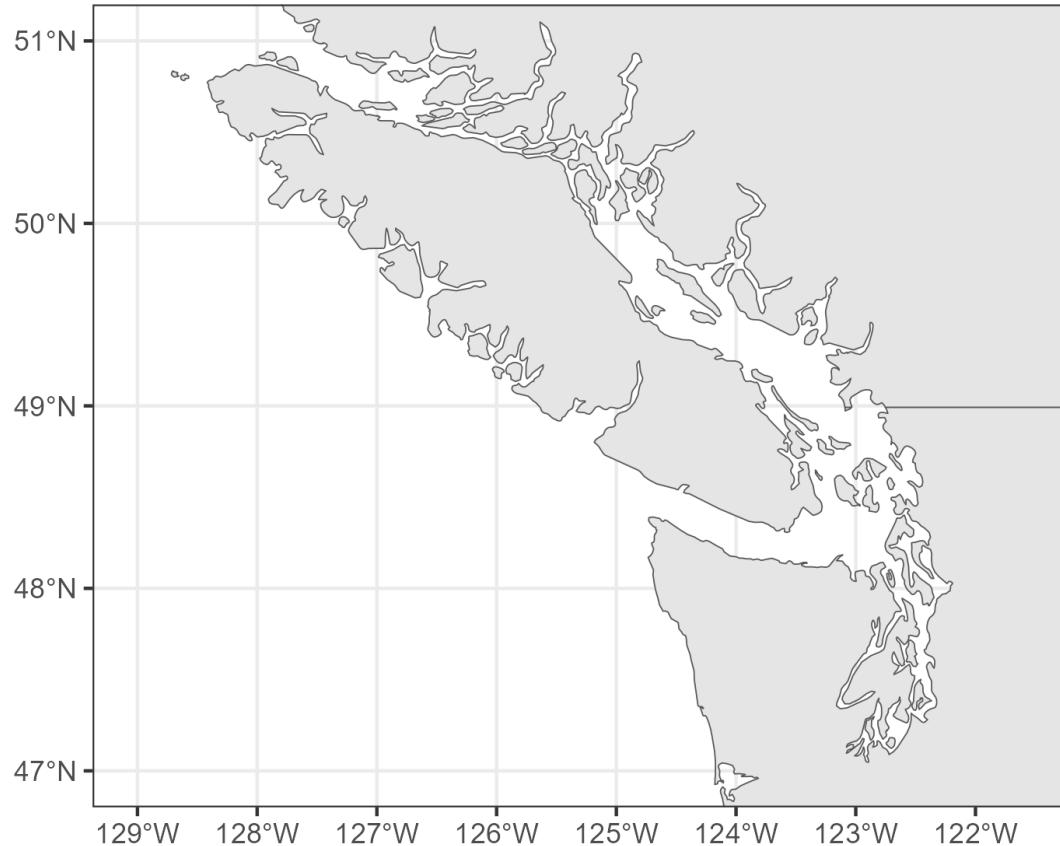
RNaturalEarth 1:10M



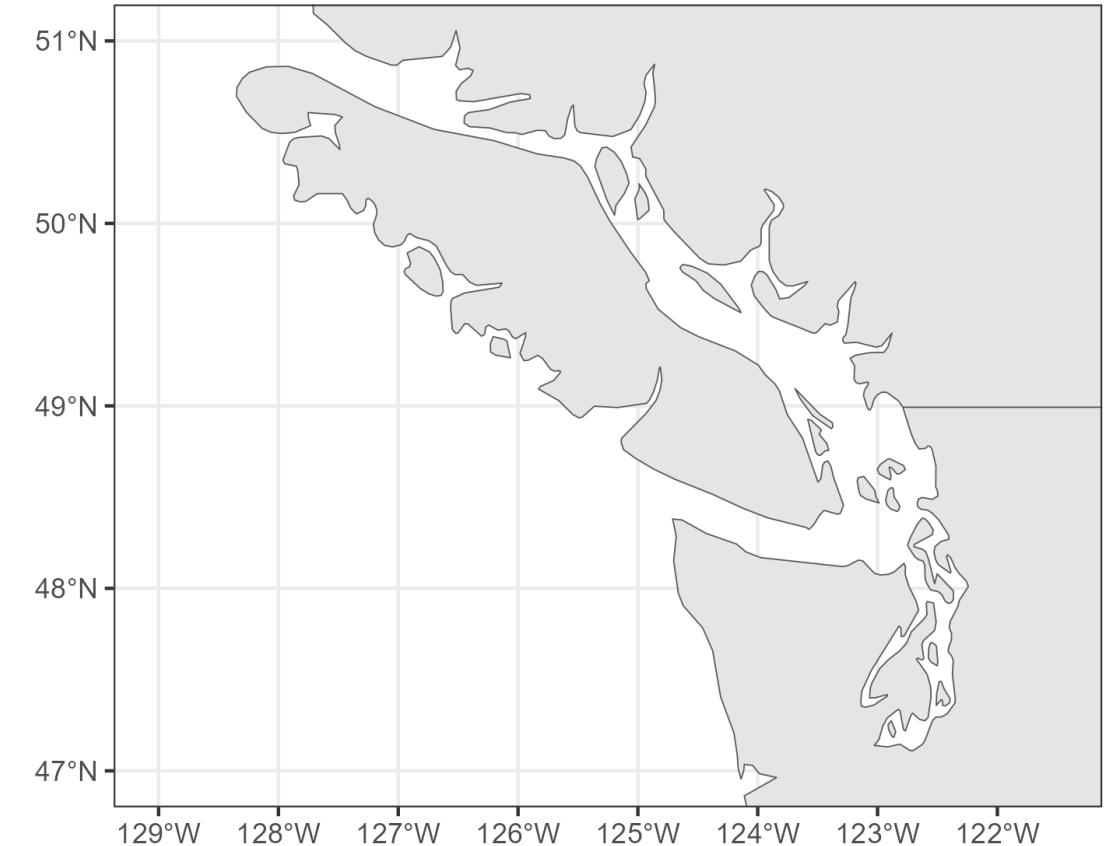
RNaturalEarth

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

RNaturalEarth 1:10M



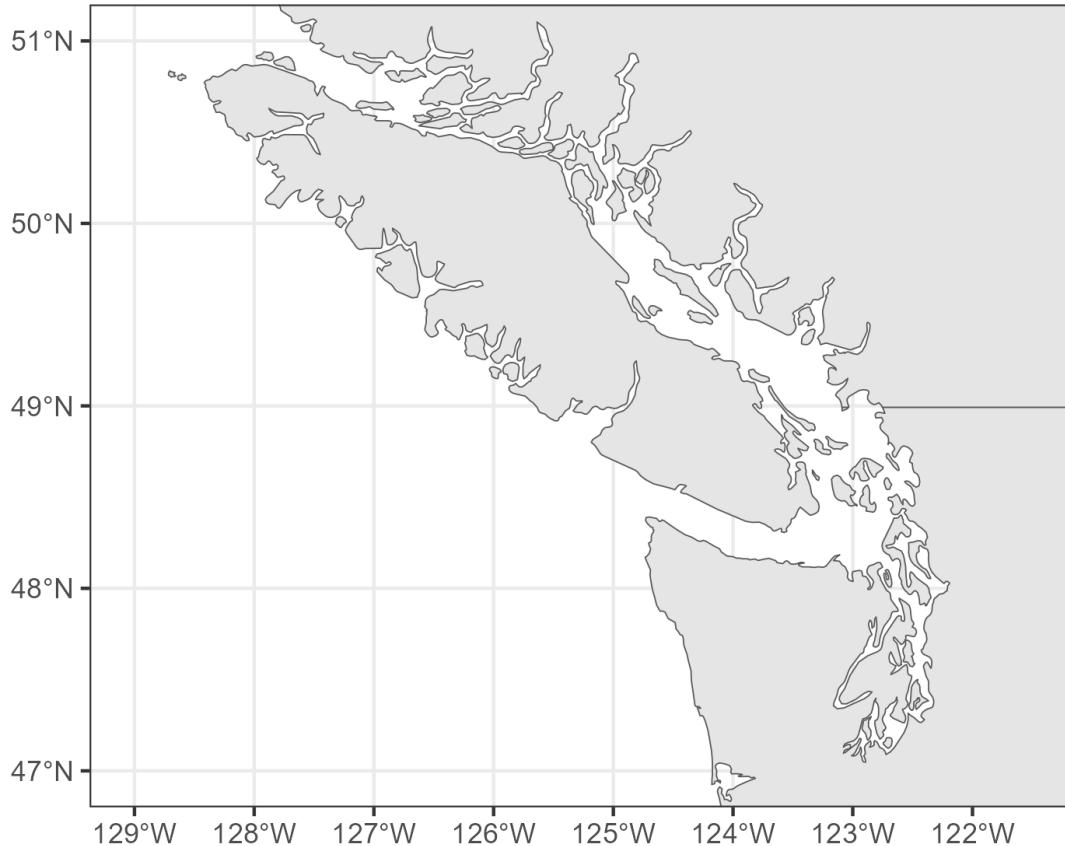
RNaturalEarth 1:50M



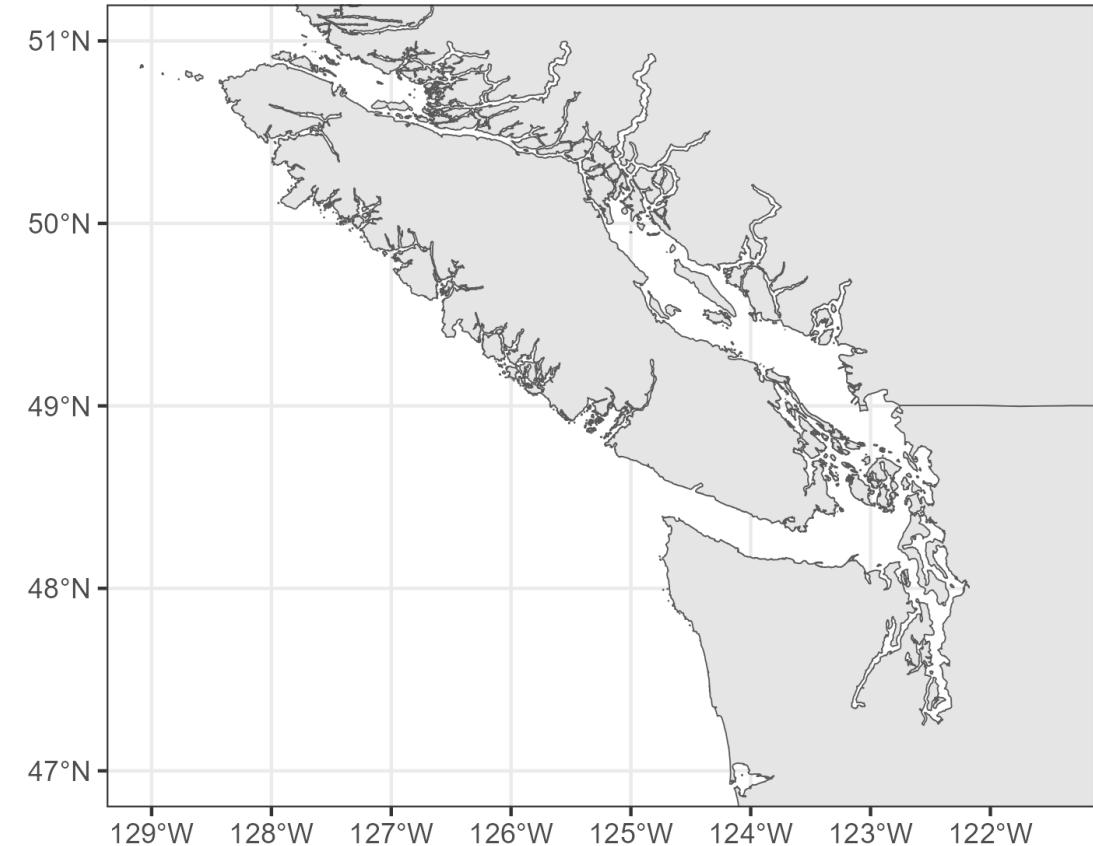
Compare RNaturalEarth to bc-coast.shp

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

RNaturalEarth 1:10M

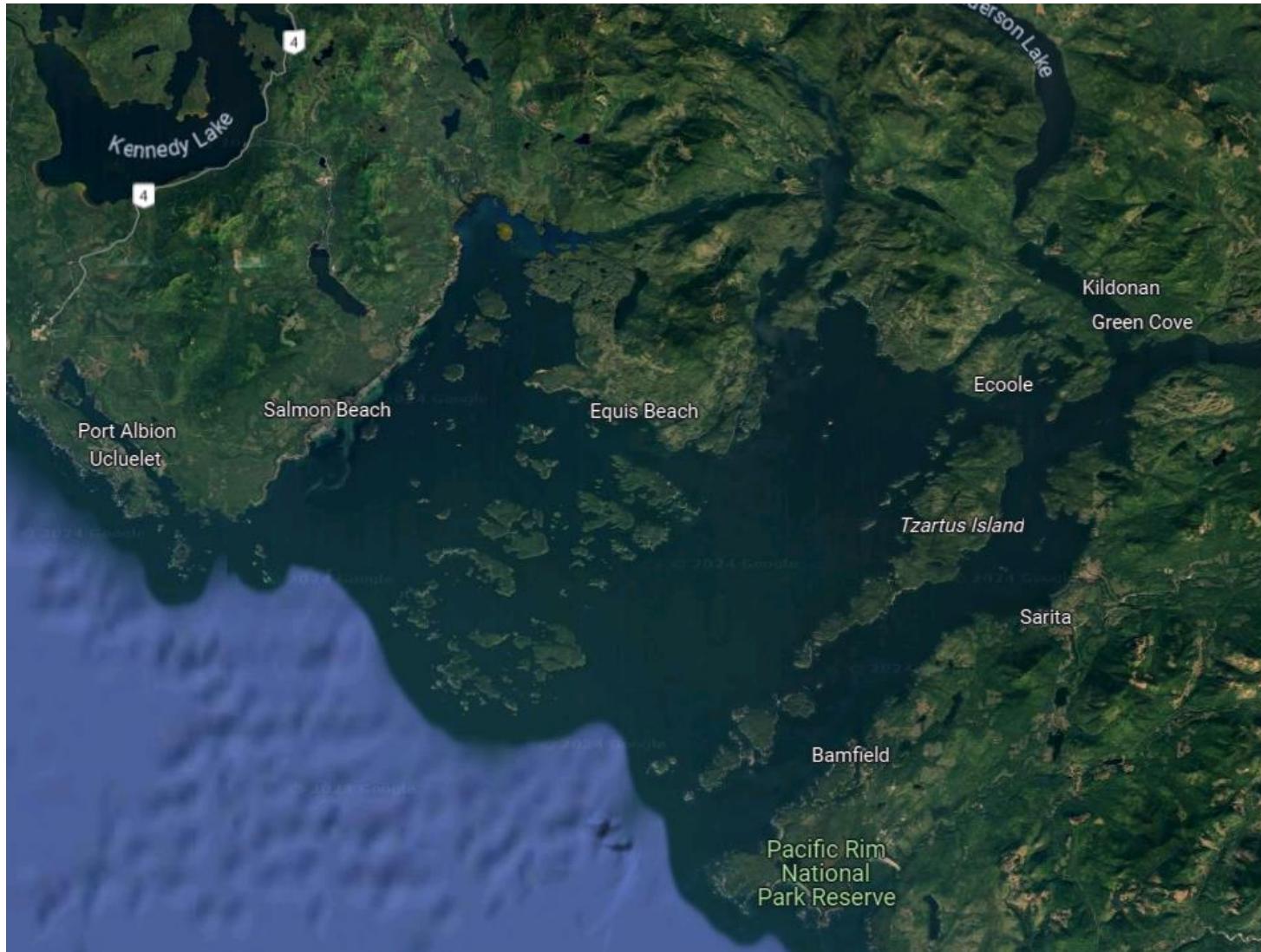


BC coast shapefile



Compare RNaturalEarth to bc-coast.shp

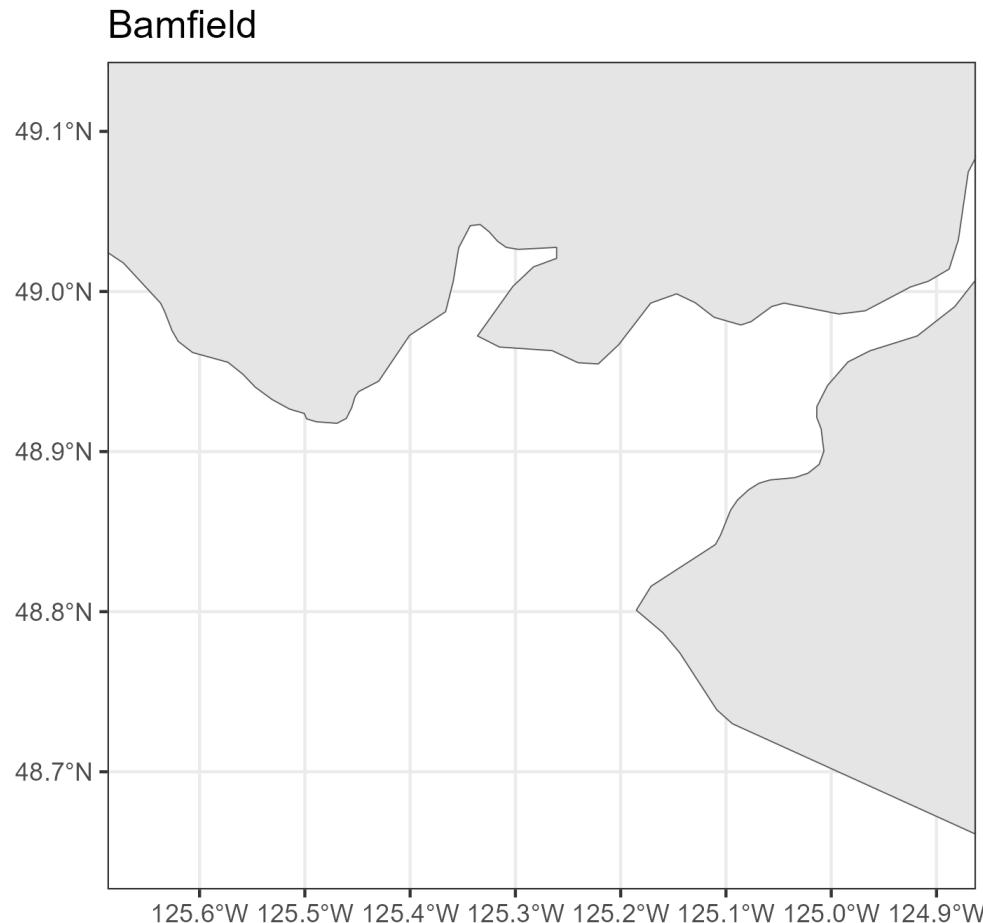
Bamfield, west coast of Vancouver Island



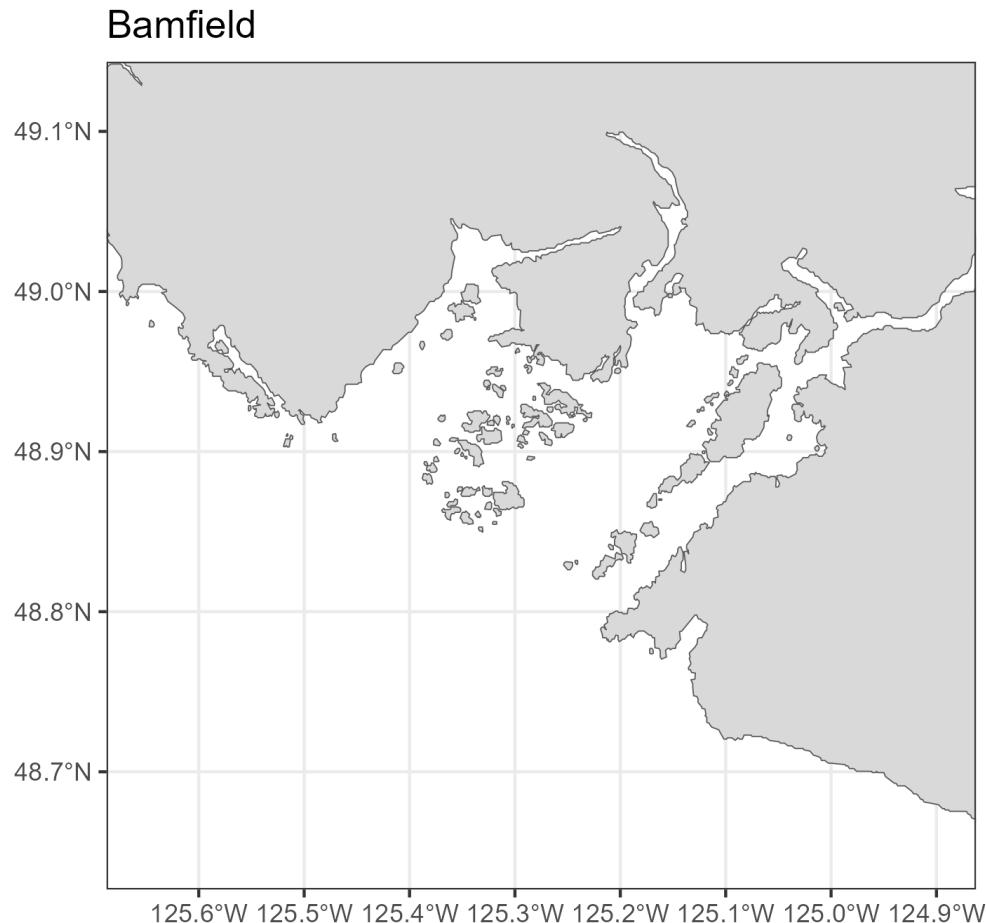
Compare RNaturalEarth to bc-coast.shp

Bamfield, west coast of Vancouver Island

RNaturalEarth 1:10M

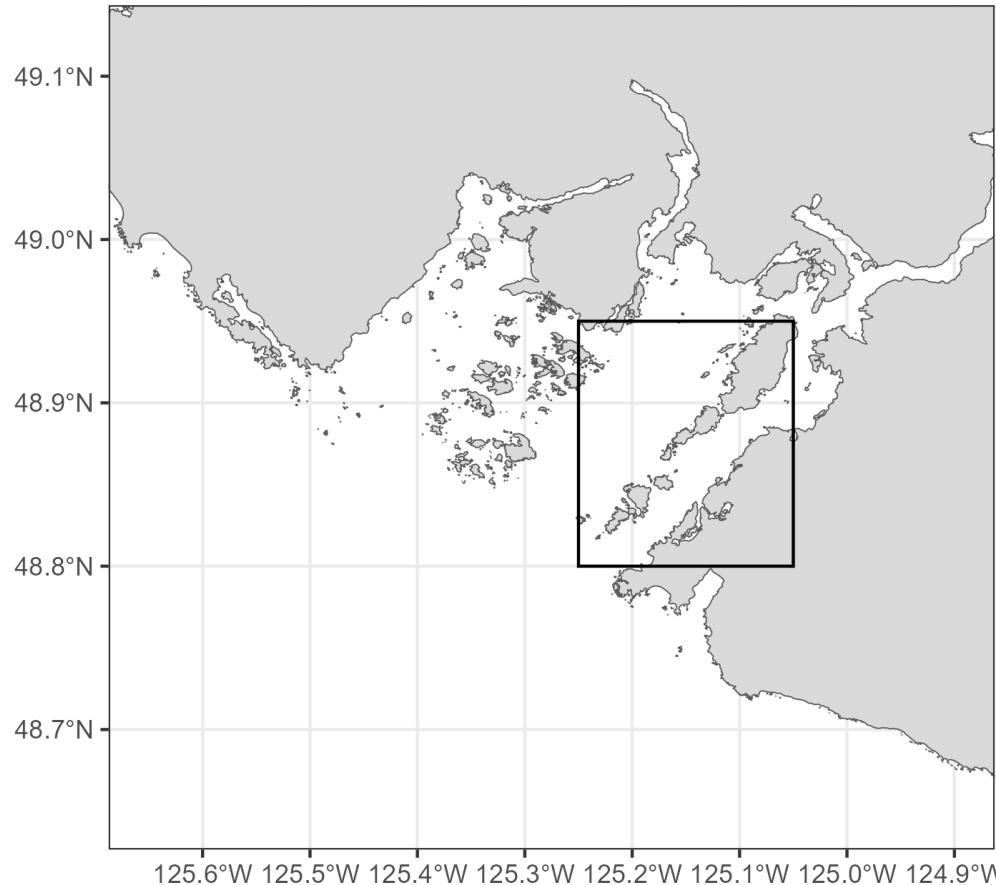


BC coast shapefile

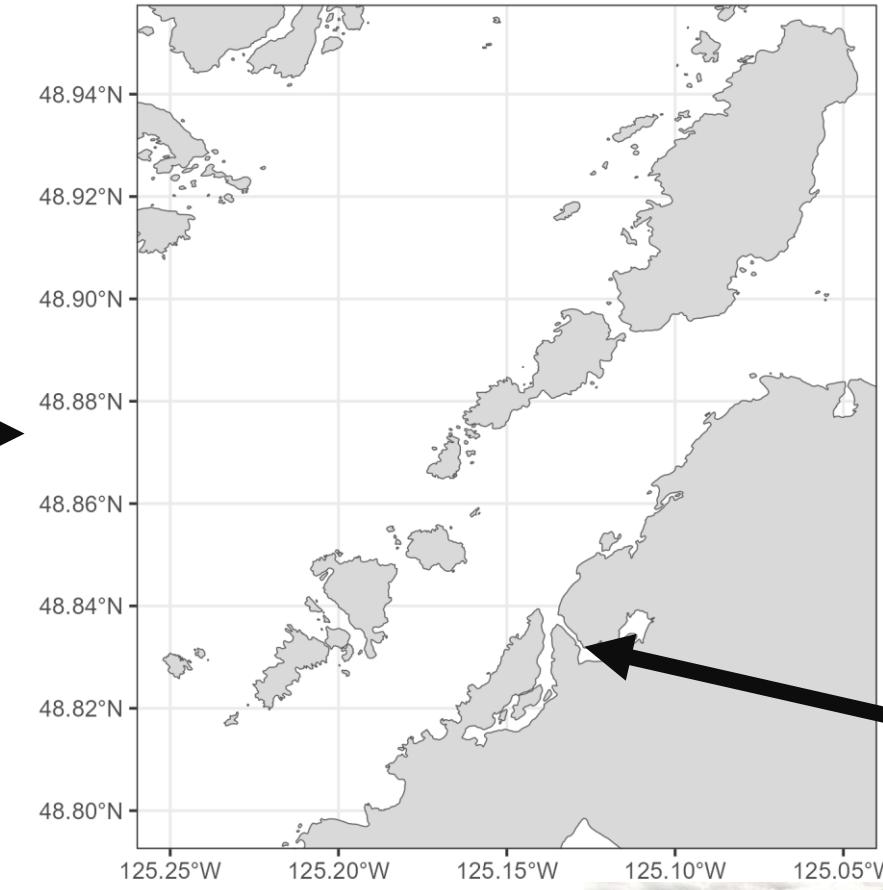


Higher-resolution for local maps

Even higher resolution is available with shapefiles from Hakai



Zoom in
→



Sneak peak: satellite maps in R

Even higher resolution with satellite maps

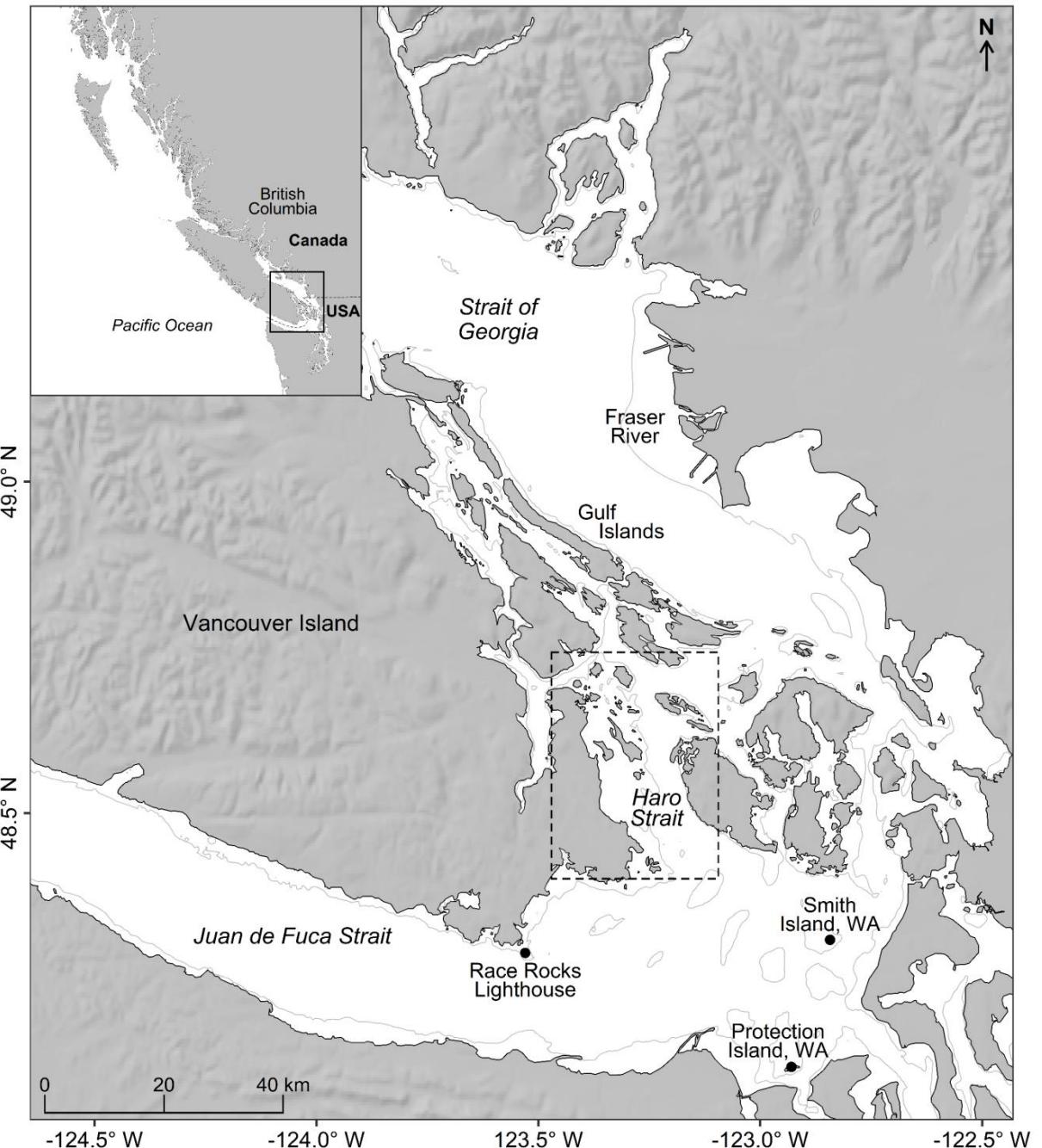


Zoom in
→



Study area maps

Goal: show sampling locations and important geographic/biological features

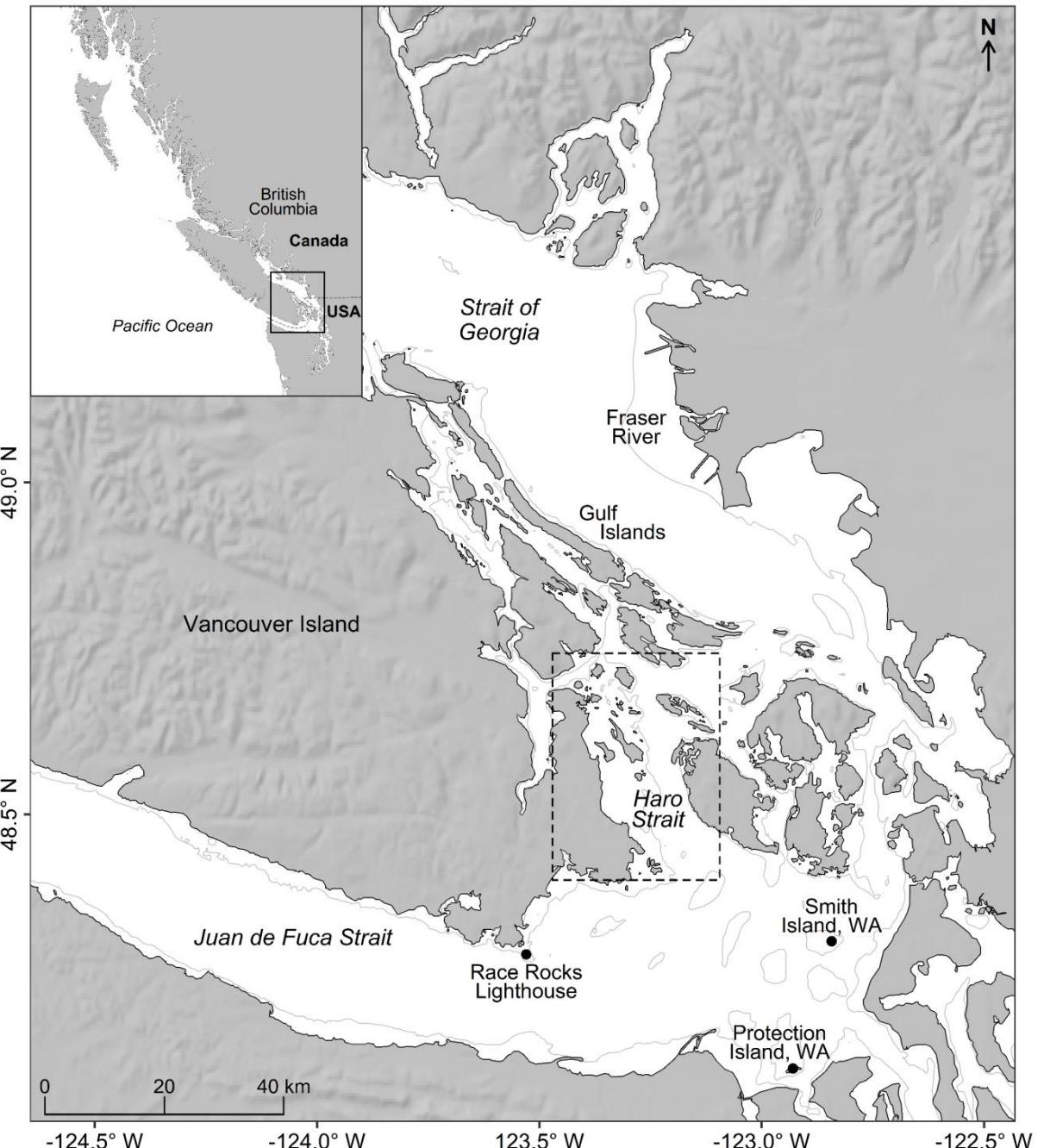


Study area maps

Goal: show sampling locations and important geographic/biological features

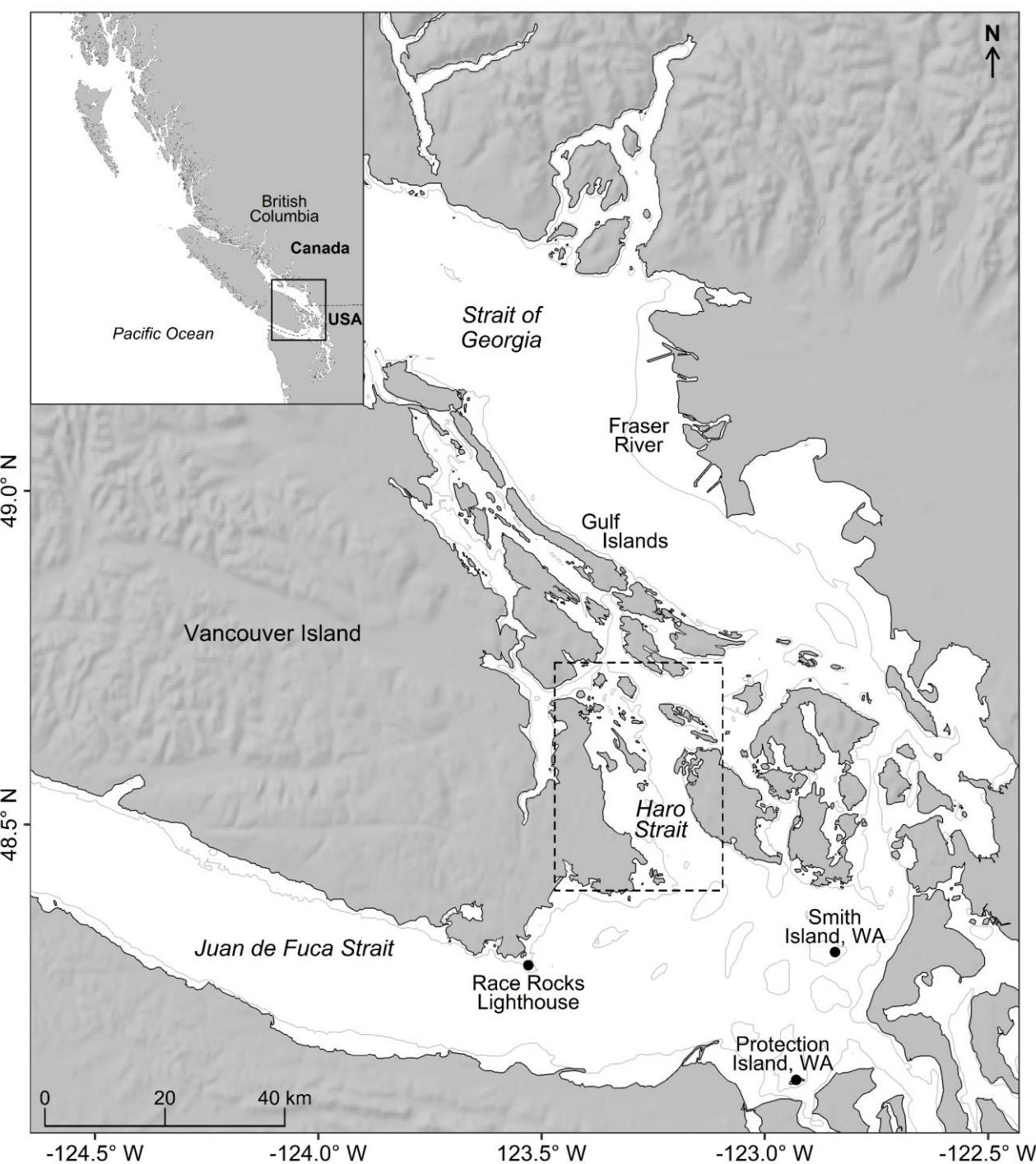
Often greyscale for simplicity

Include inset map showing larger geographic region



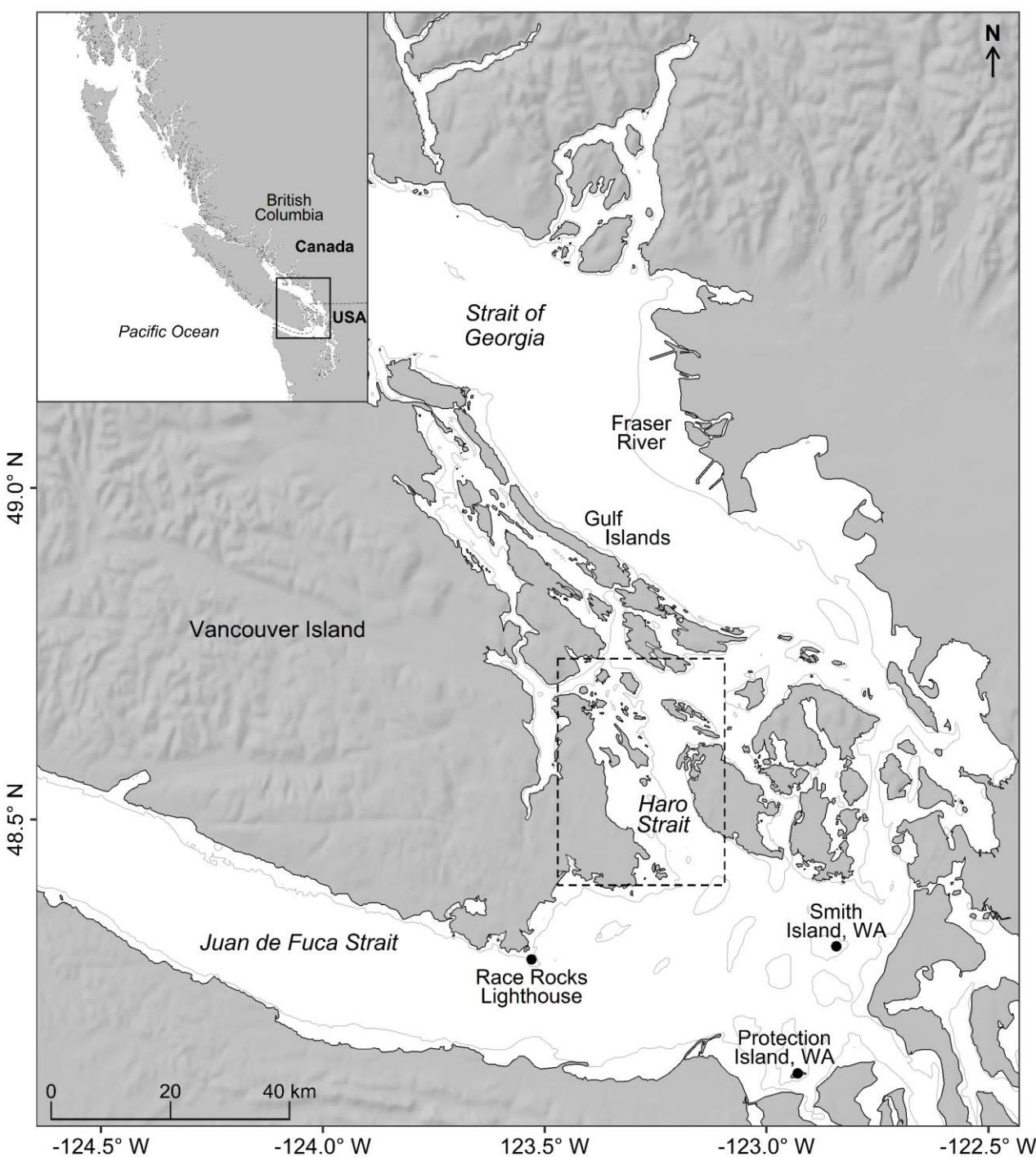
Important parts

**Good spatial
resolution is key**



Important parts

**What do you
think is
important?**

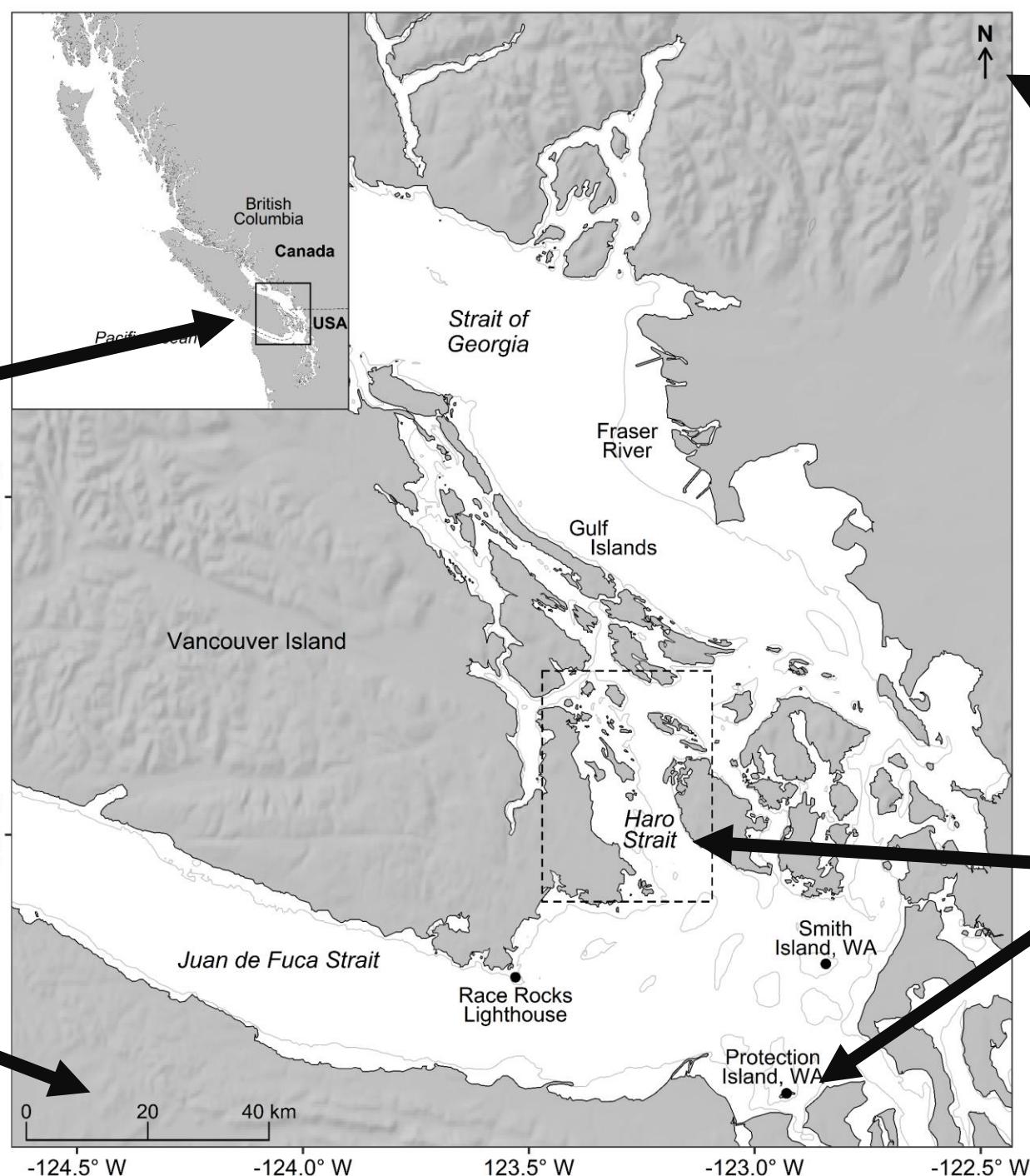


Important parts

Inset map showing where the study area is

Study area denoted by rectangle

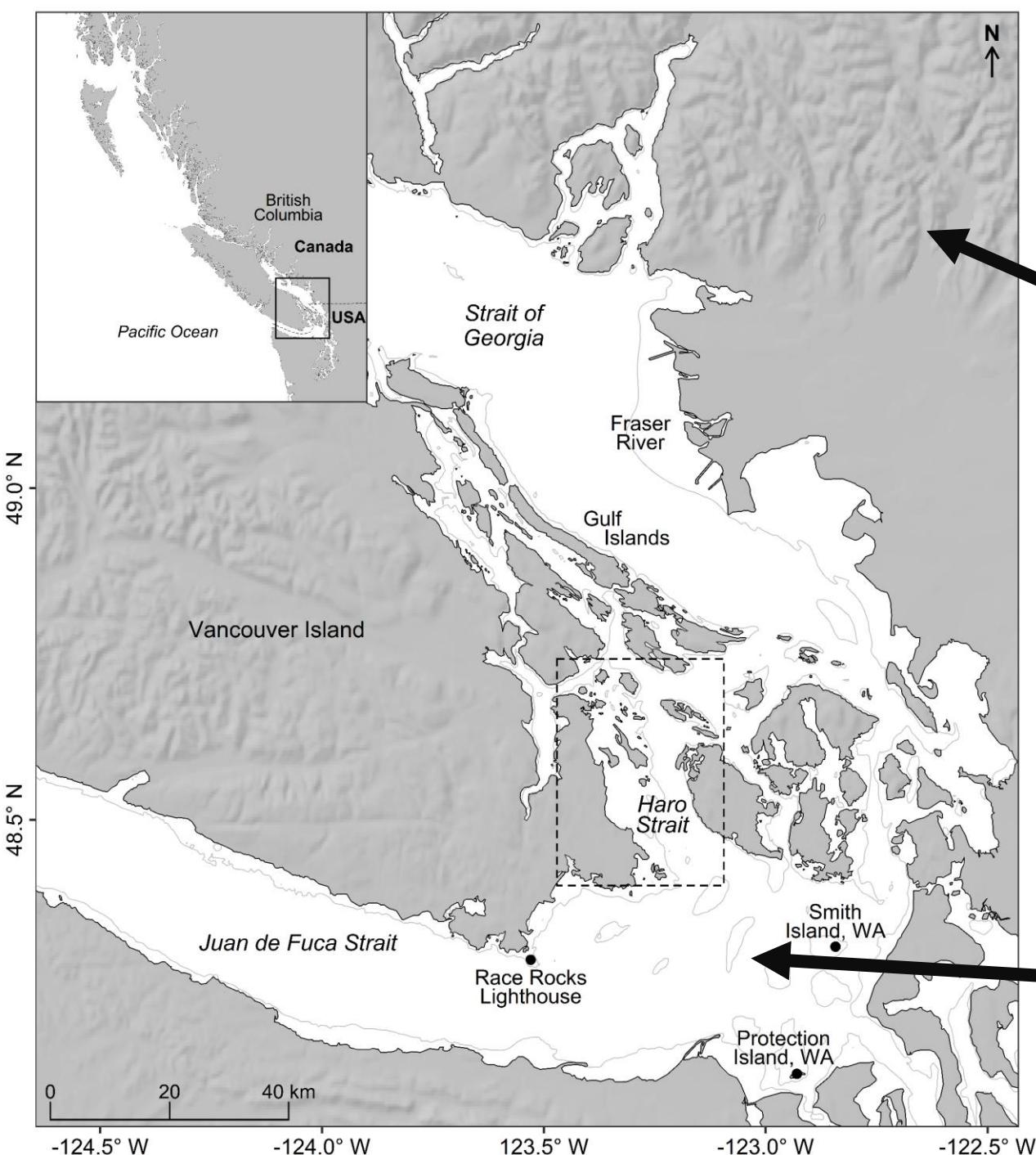
Scale bar



North arrow

Important sites labelled

Extra pieces



Hillshading to illustrate topography

50 metre depth contours

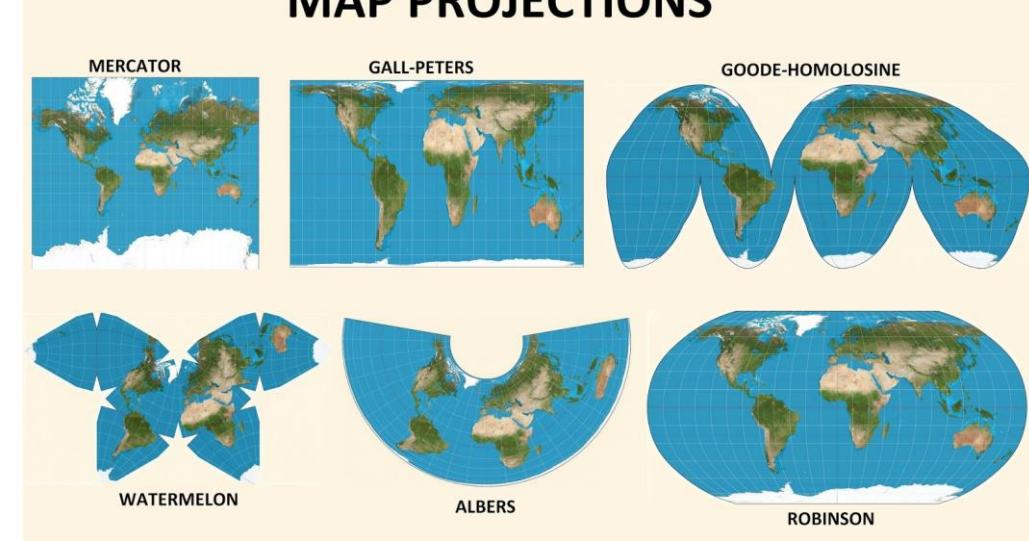
Projections

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

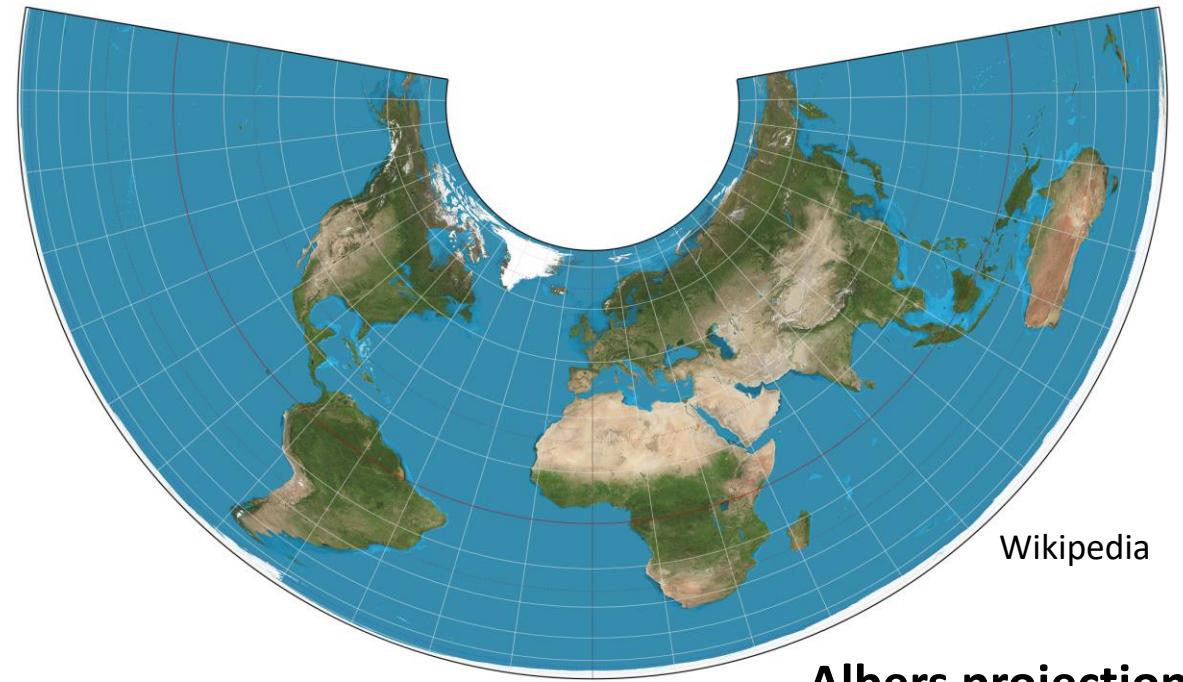
What projection to use?

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

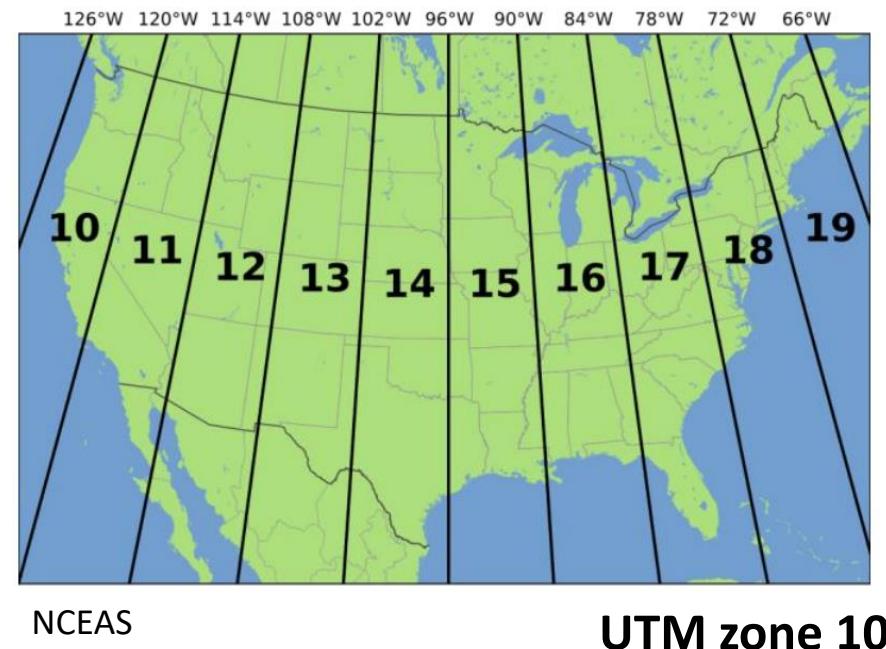
In BC, we want to use **BC Albers**

BC Albers: “equal area projection”

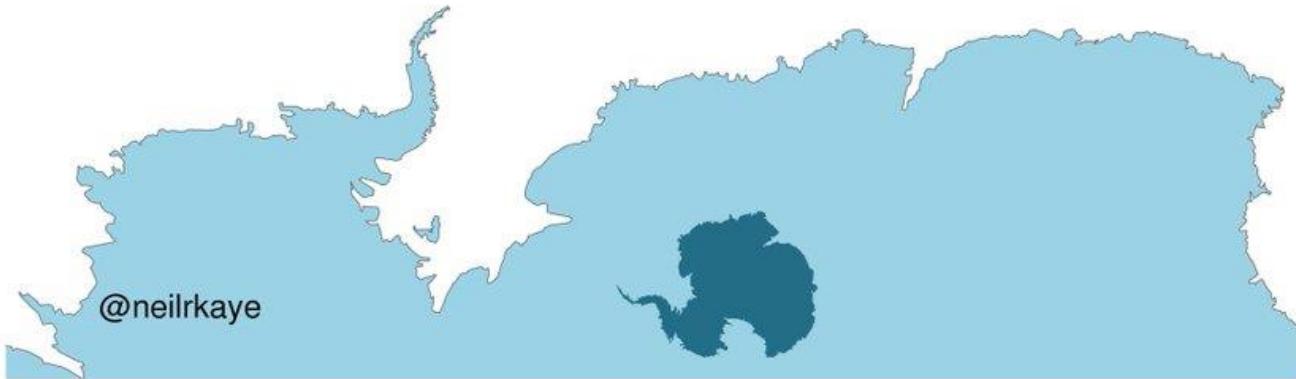
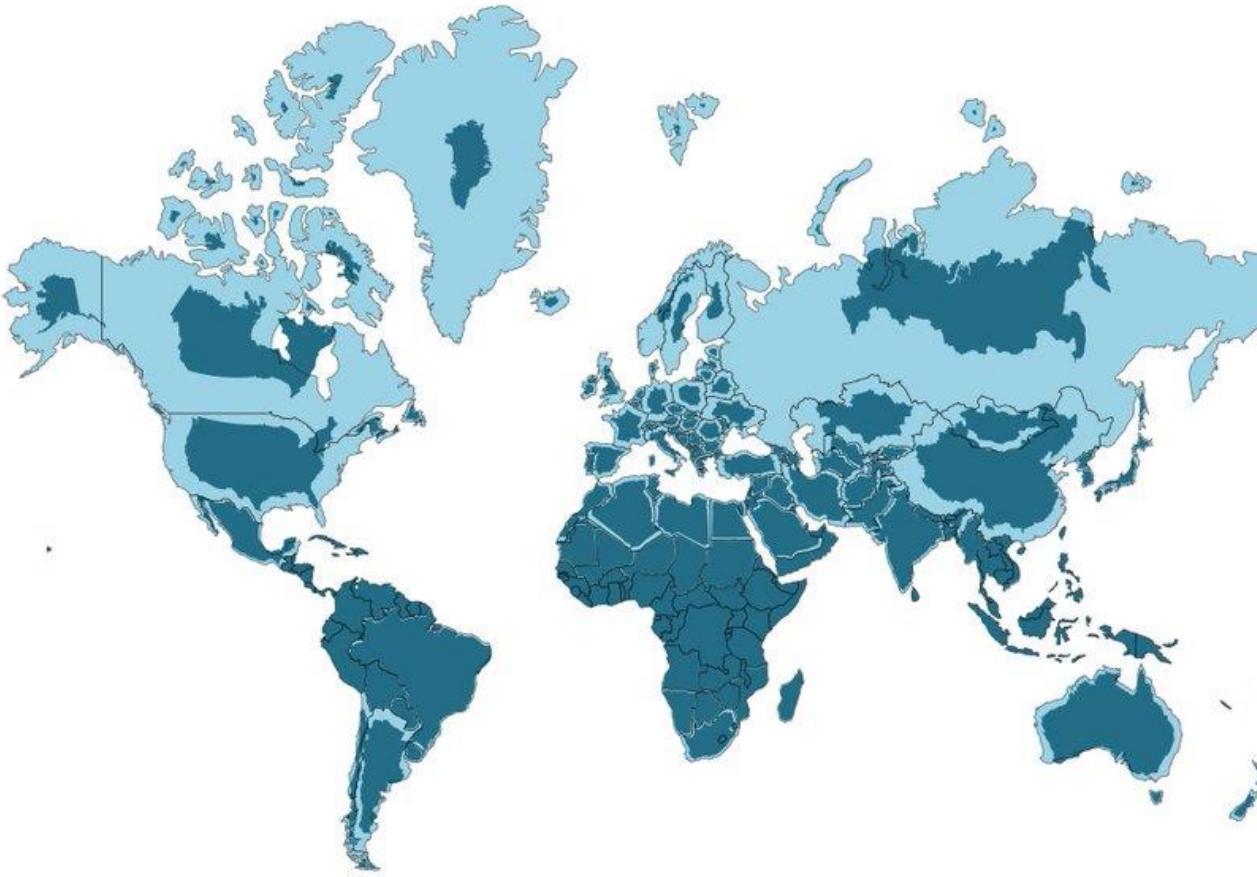
**Northings and Eastings
(metres/kilometres) instead of
latitude/longitude (degrees)**



Albers projection

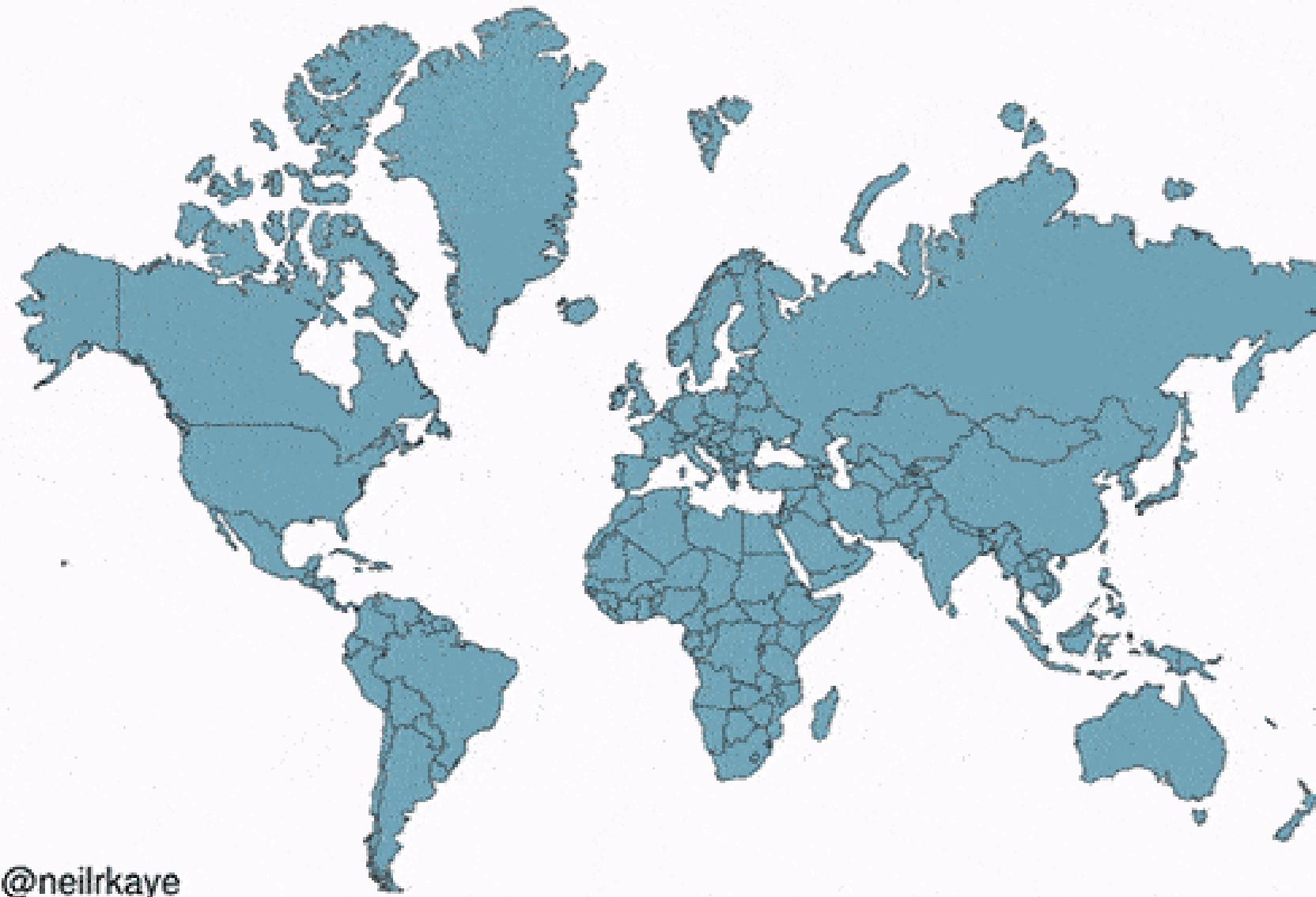


World Mercator projection with true country size and shape added



@neilrkaye

World Mercator projection with country going to true size

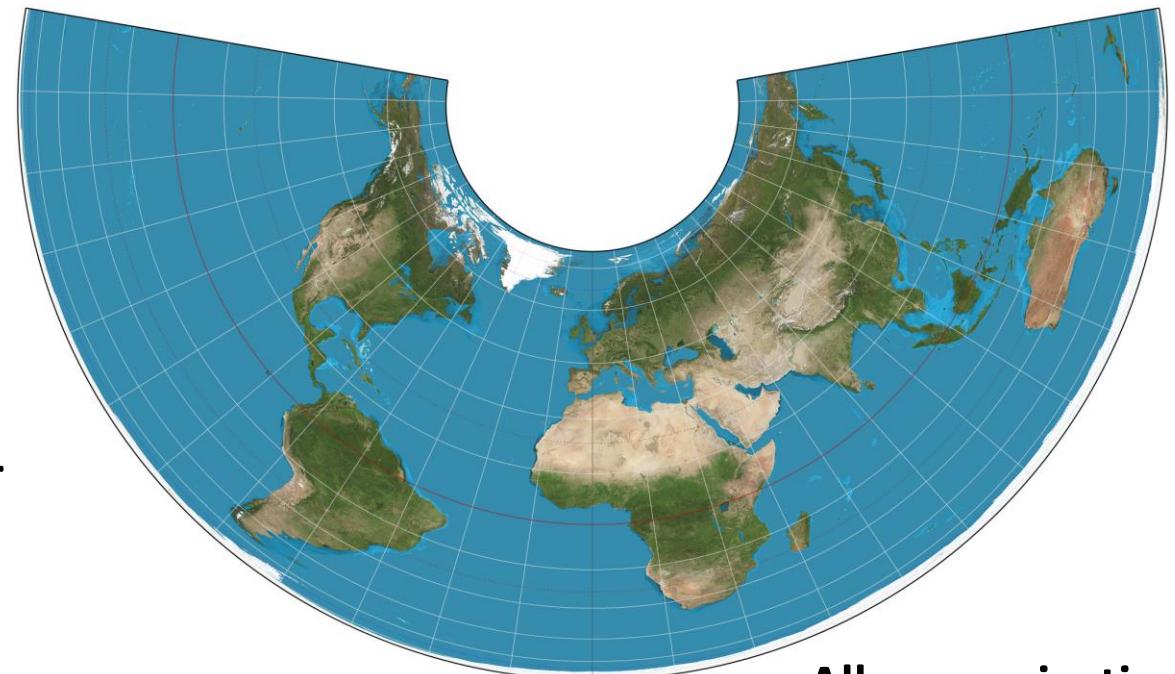


@neilrkaye

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

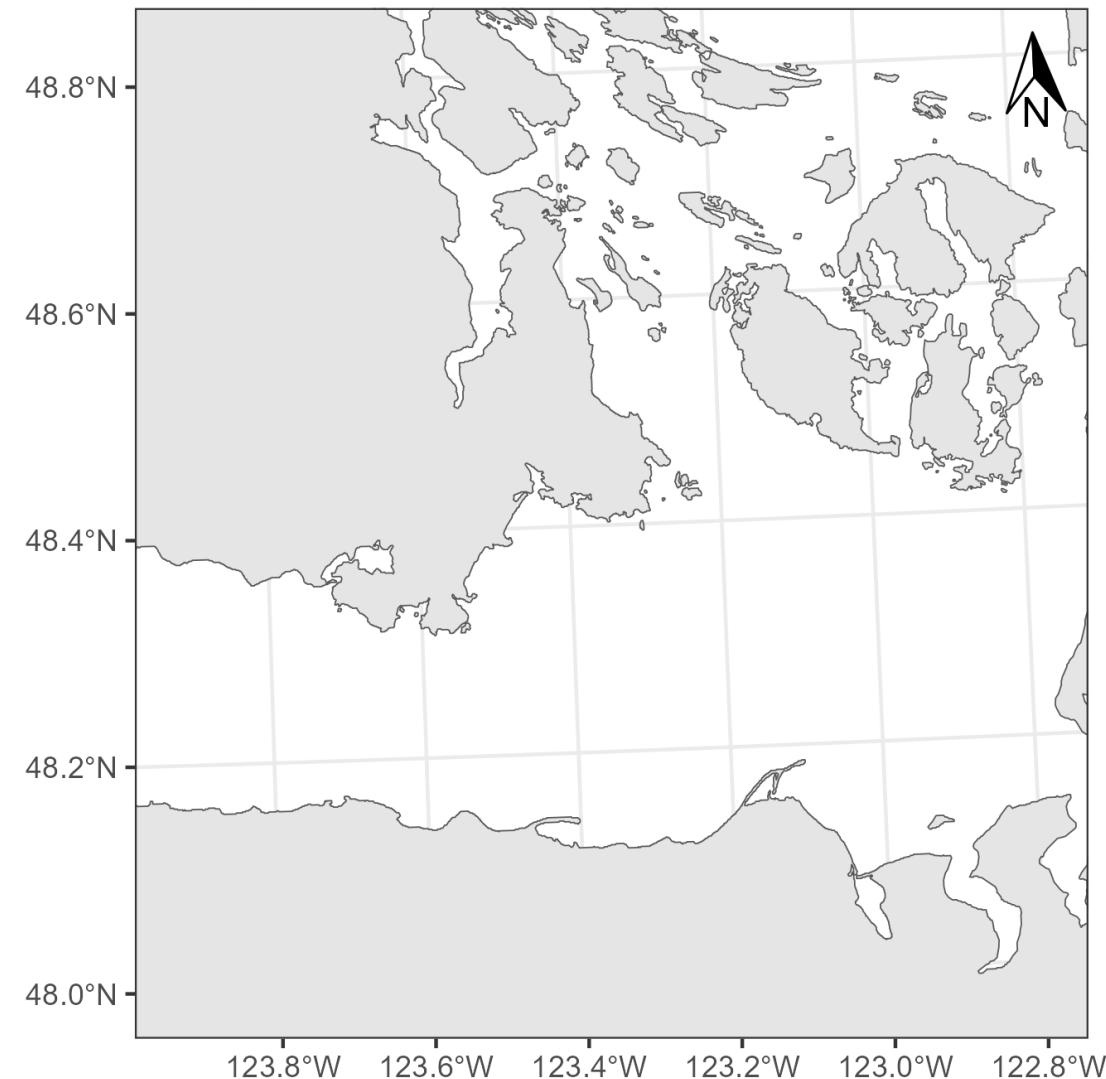
Projections

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

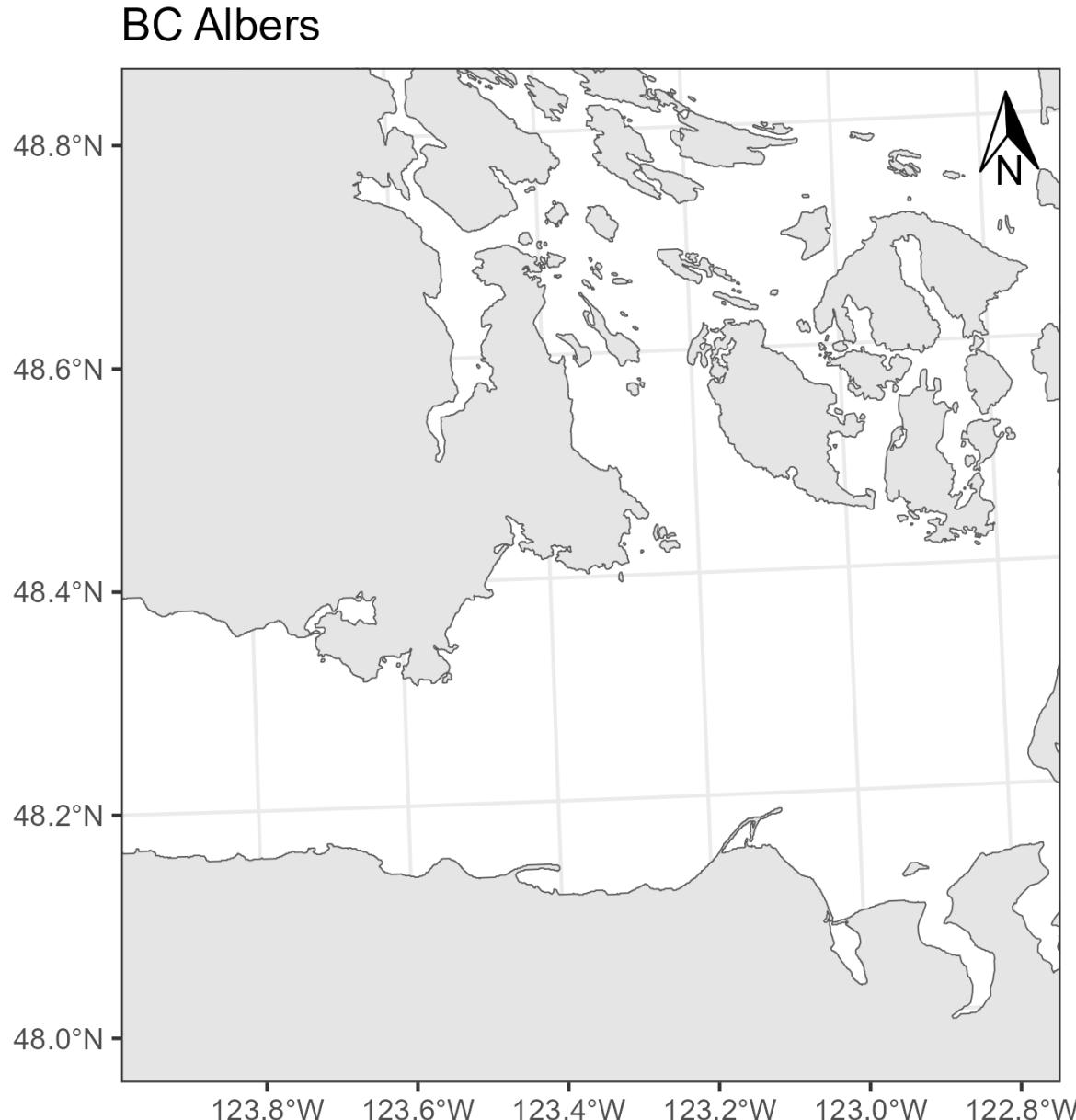
See 00-study-area-map.R, section 4 for easy code for setting the bounding box of maps in a projection that uses metres or kilometres

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

BC Albers



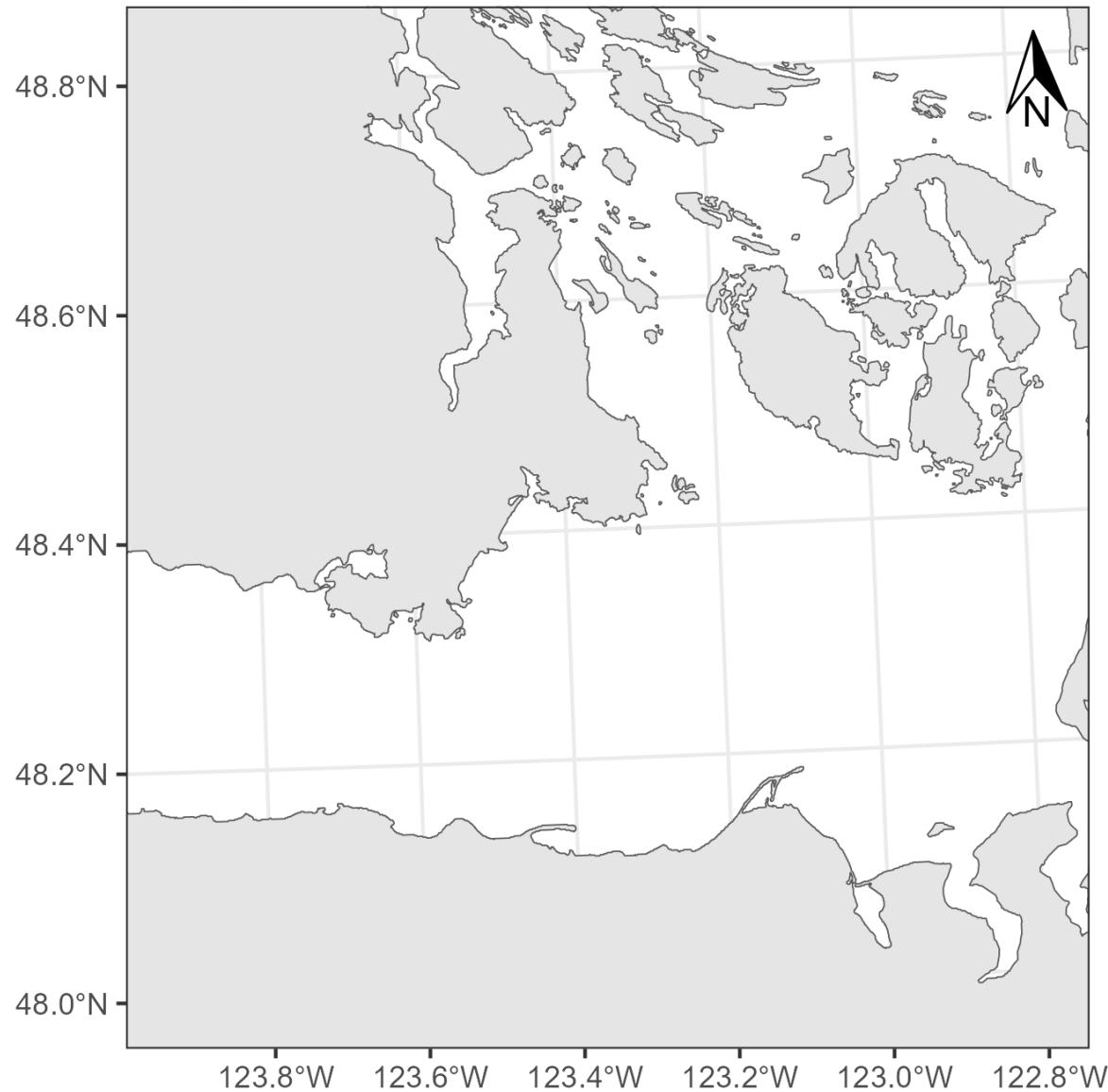
BC Albers study area map



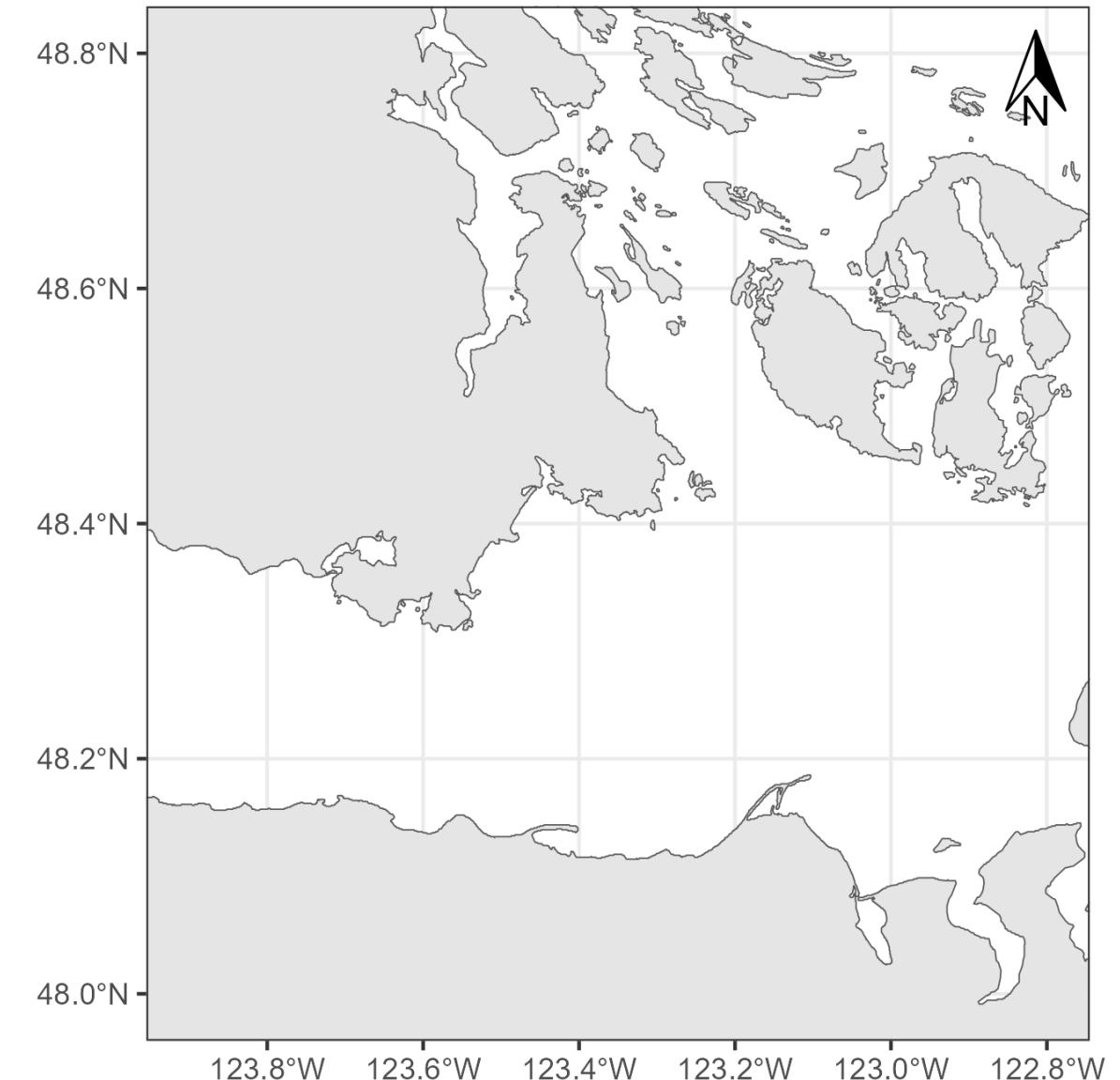
North arrow points
slightly to the left

BC Albers study area map – slightly different

BC Albers

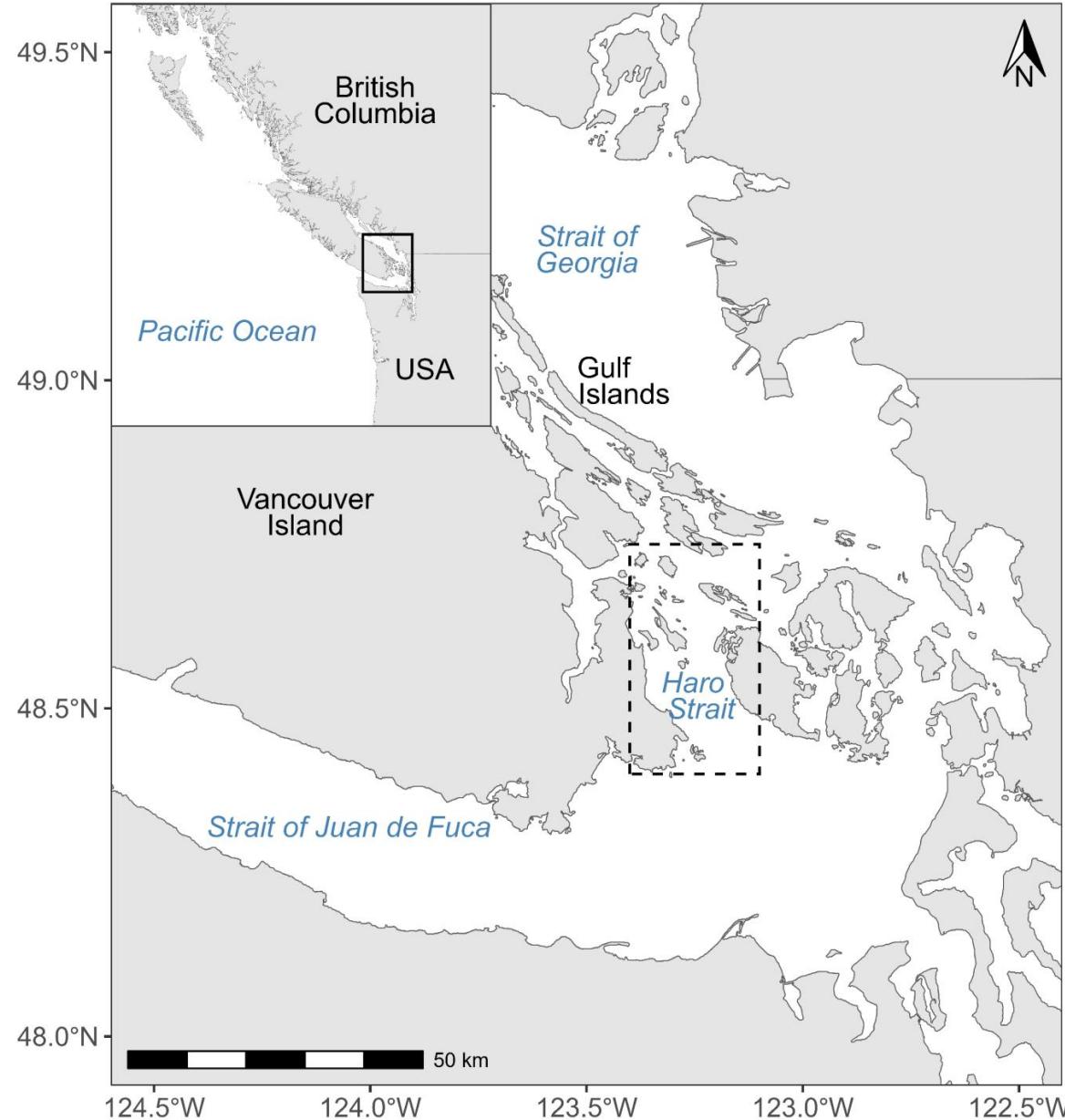


WGS 1984



Simple study area map

scripts/03-study-area-maps.R



Load basemap

```
library(ggplot2)
library(sf)

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

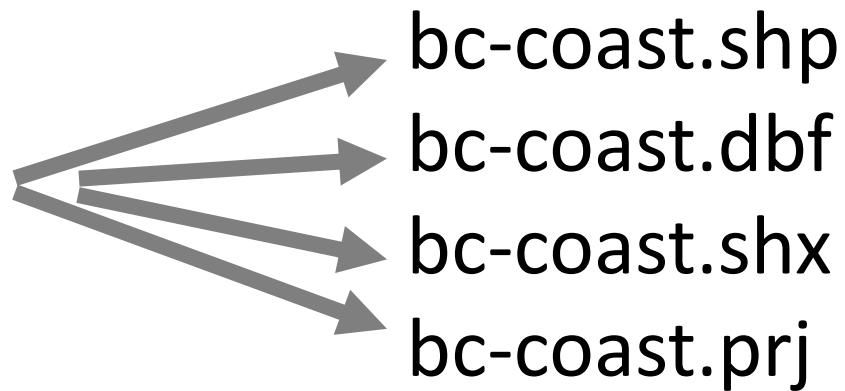
Load basemap

```
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]]
```

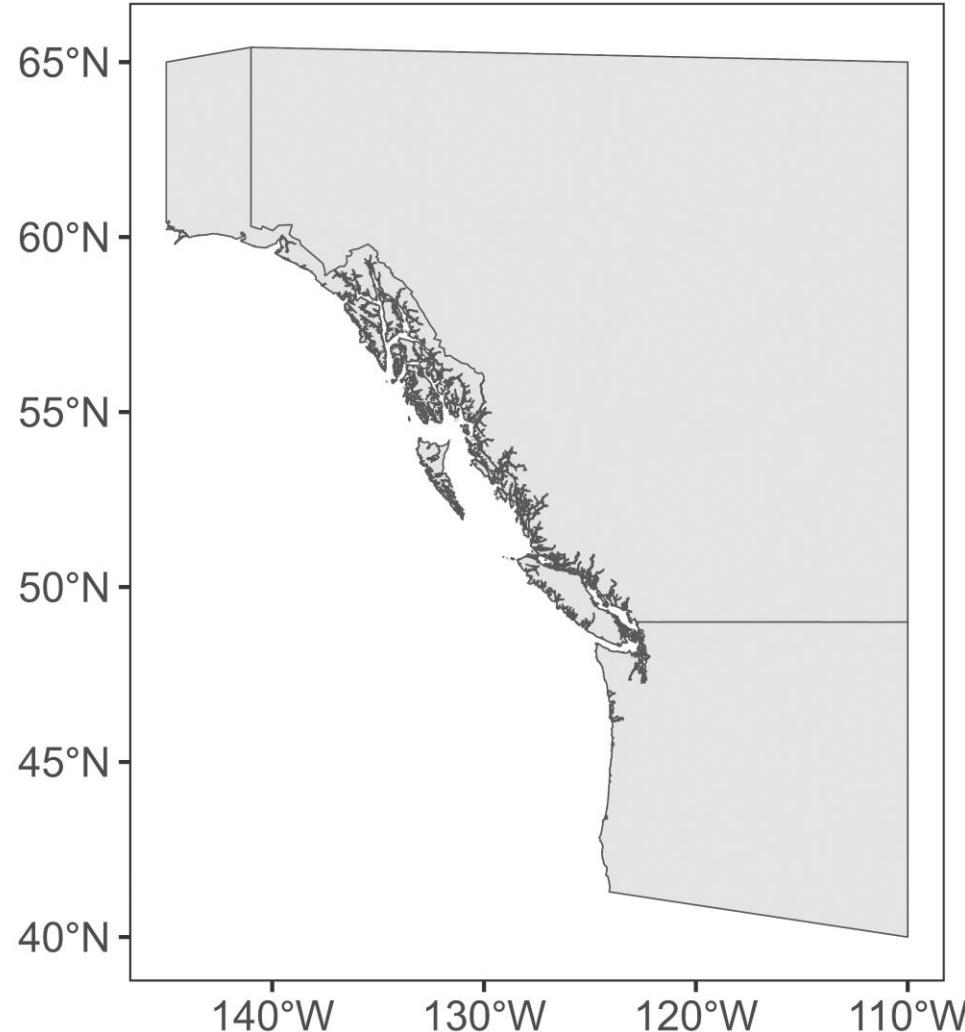
EPSG 4326: WGS 1984, a common global projection

We use EPSG numbers to refer to different projections/CRS

CRS: coordinate reference system

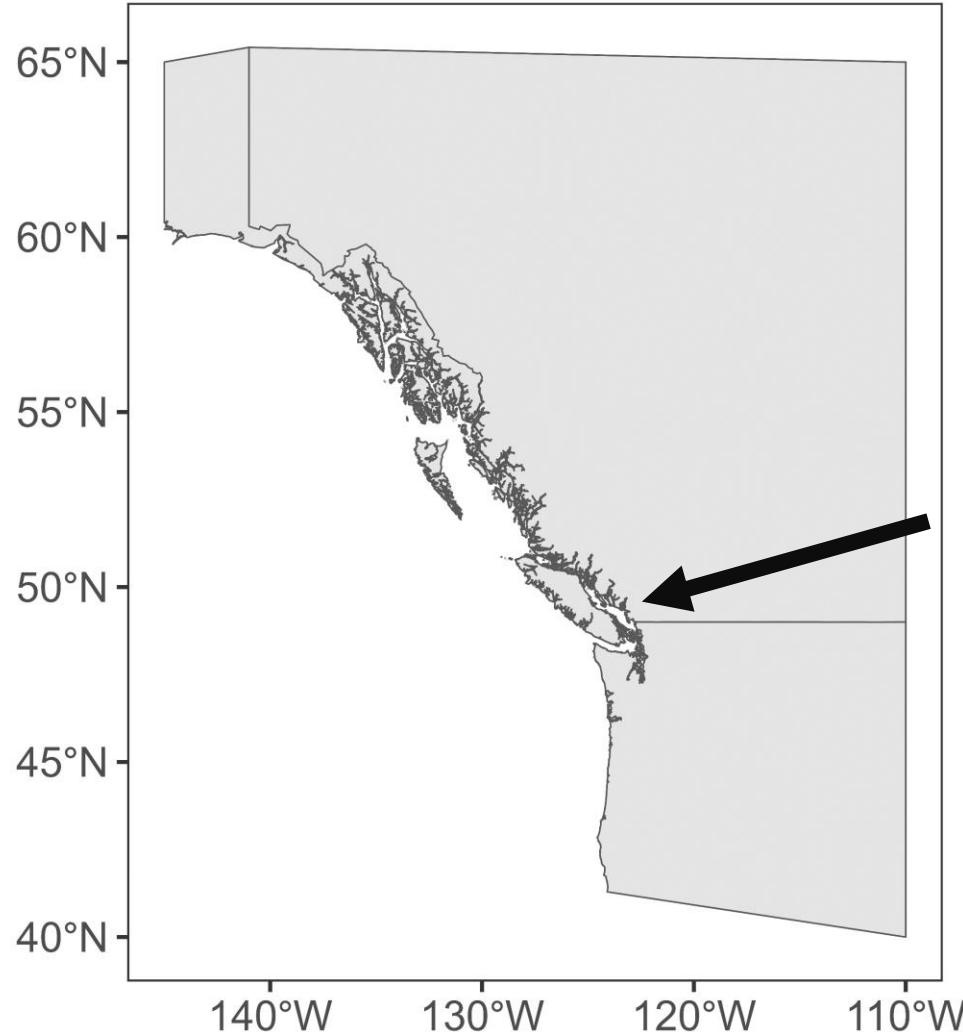
Plot basemap

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



Plot basemap

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

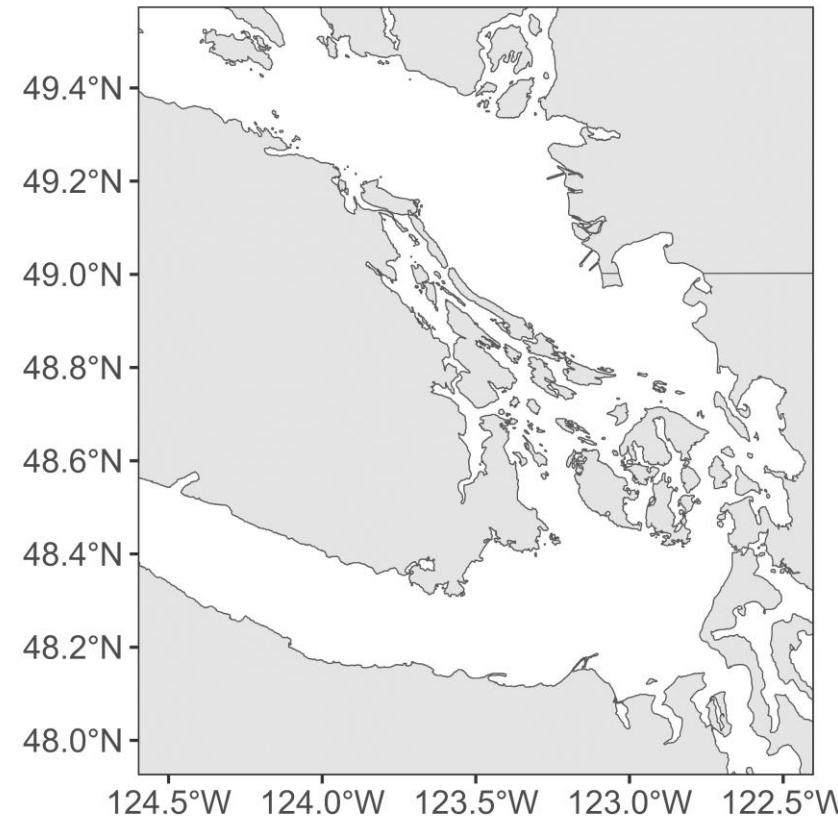
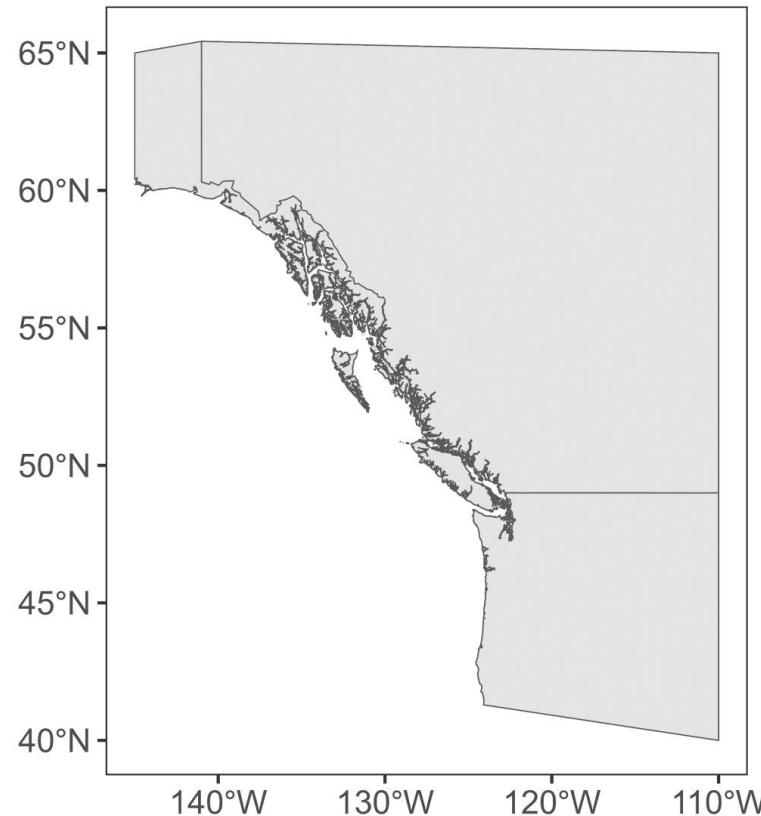


Need to zoom in on
area of interest

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



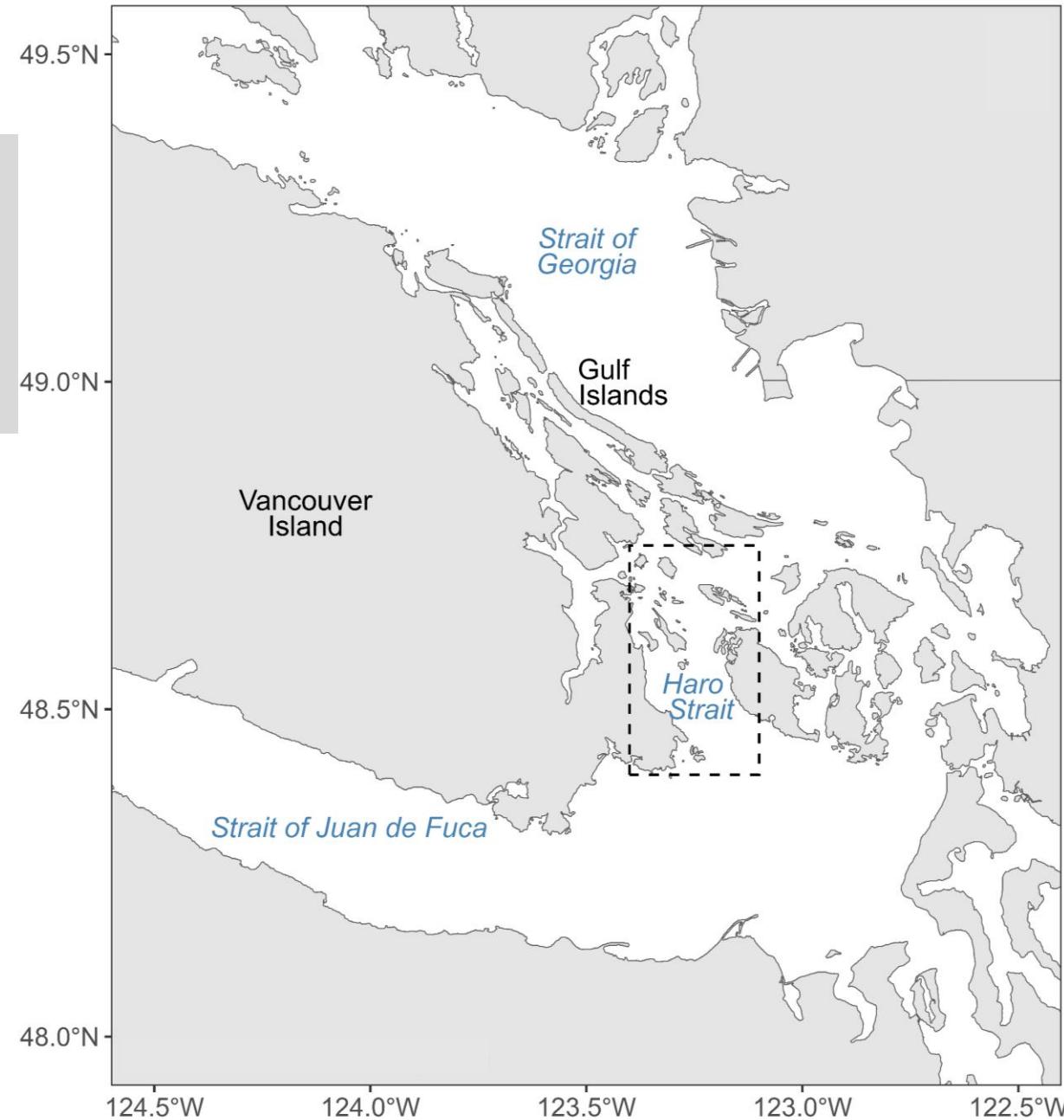
Label areas of interest

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

Personal preferences

Water bodies: *blue italics*

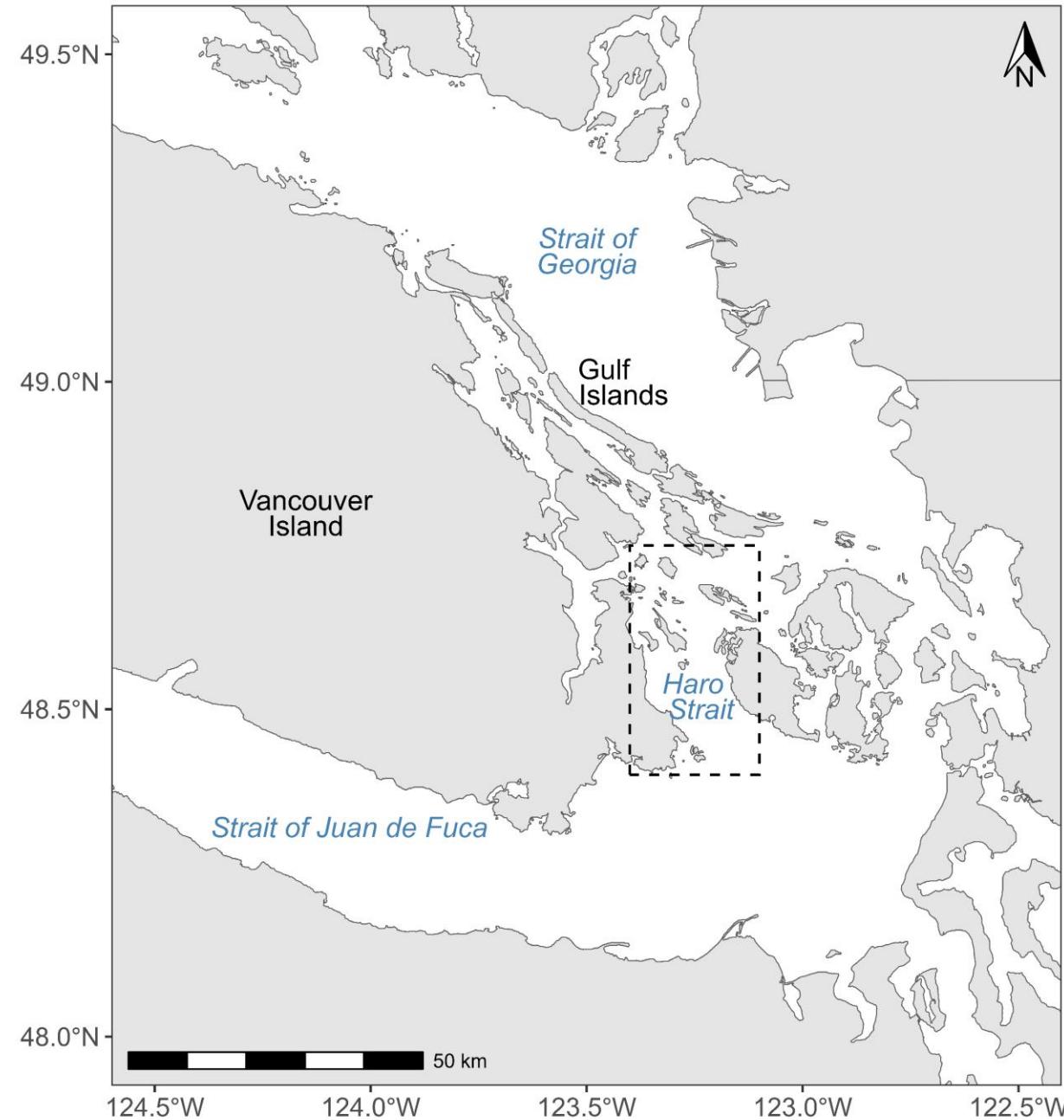
Land features: black



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)
```

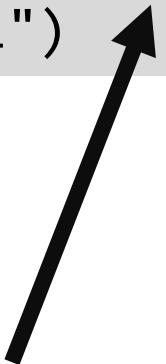
Add rectangle indicating the area
of the larger map

Same coordinates as the coord_sf()
call from the main map

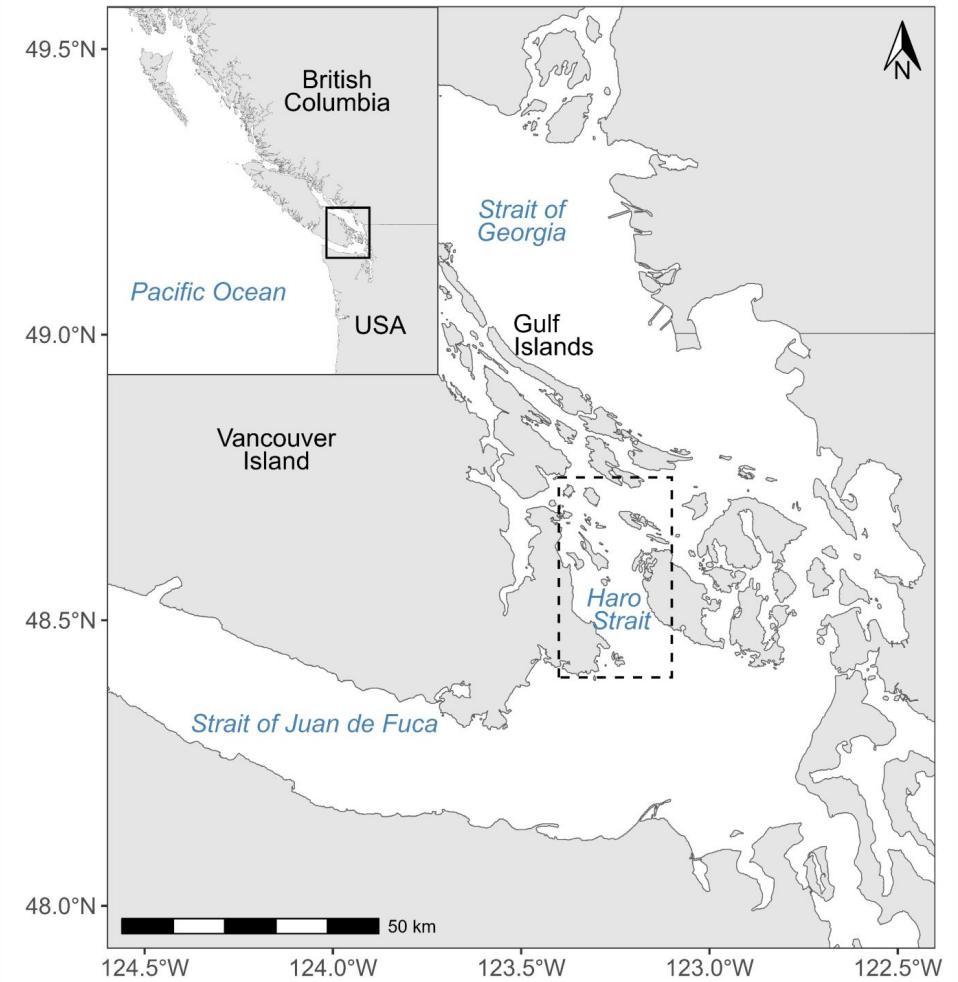


Add inset map

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

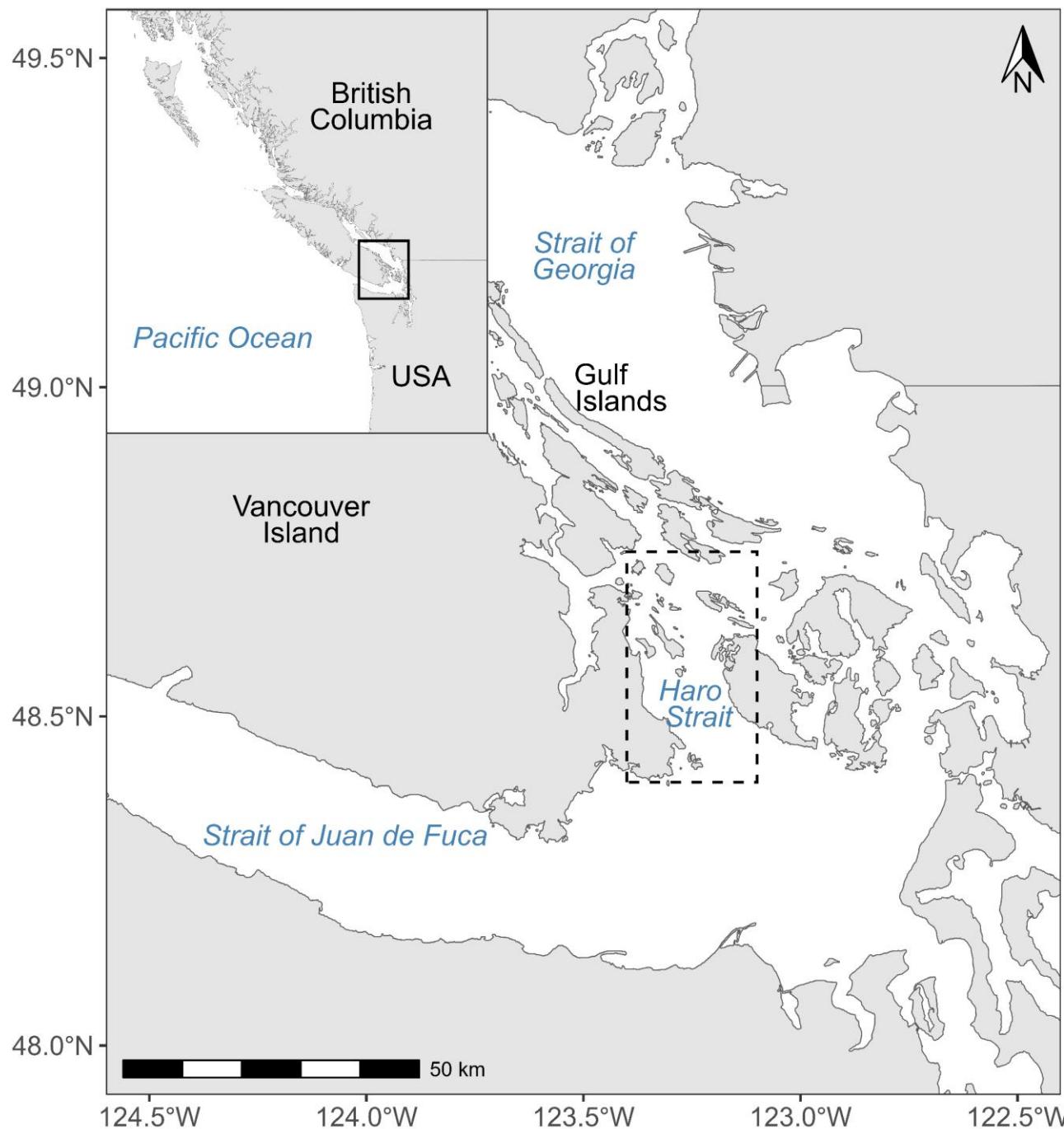


left, right, bottom, top:
Determine by trial and
error so that inset map is
desired size and location



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

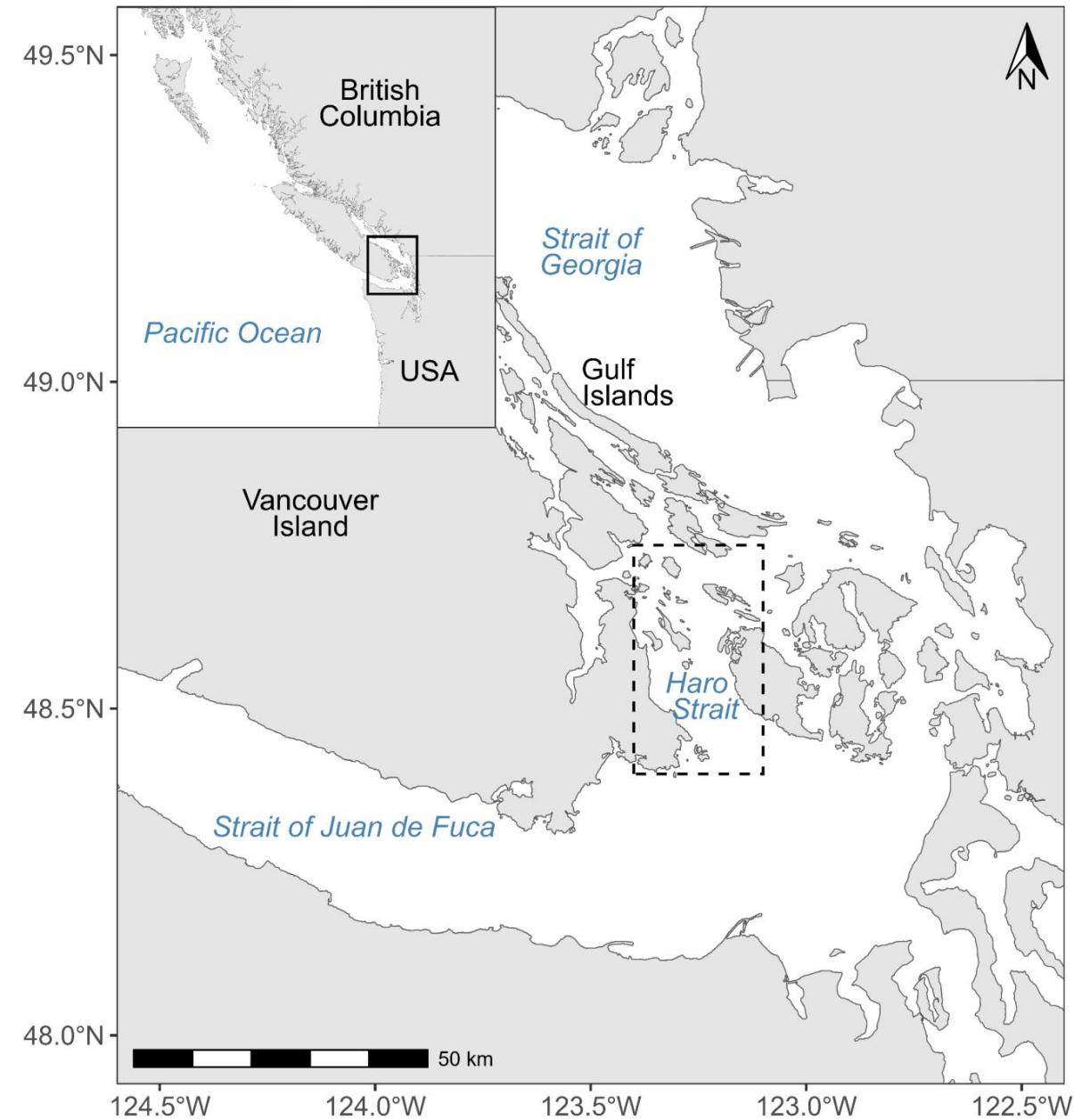


Save the map

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)
```

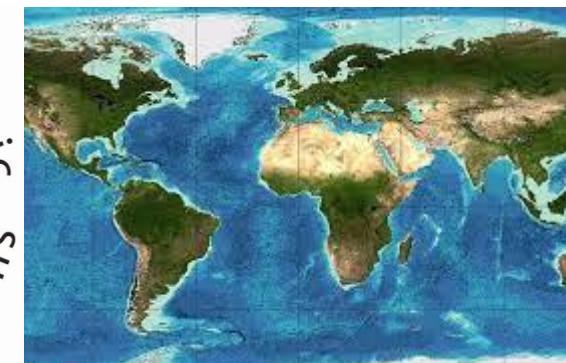
Can increase resolution to `dpi = 1600`
for very high quality images, if desired



Where to find additional data layers?

Lots of open-source spatial data available

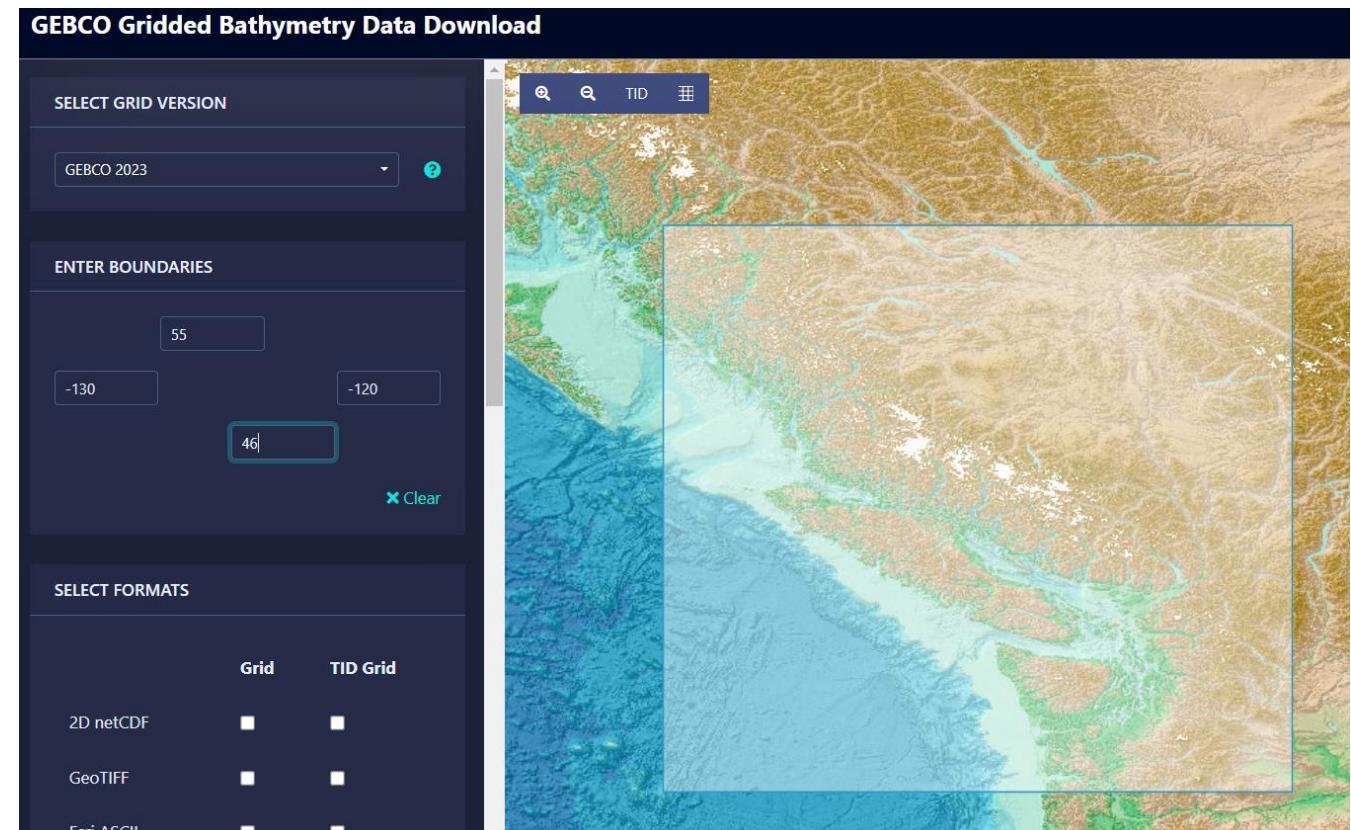
- BC Government Data Catalogue
(especially terrestrial, freshwater)
<https://catalogue.data.gov.bc.ca/>
- PSF/UBC Strait of Georgia Data Centre and Marine Reference Guide
<https://soggy2.zoology.ubc.ca/geonetwork/srv/eng/catalog.search#/home>
<https://gis.sogdatacentre.ca/sog-mrg/>
- GEBCO (bathymetry 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

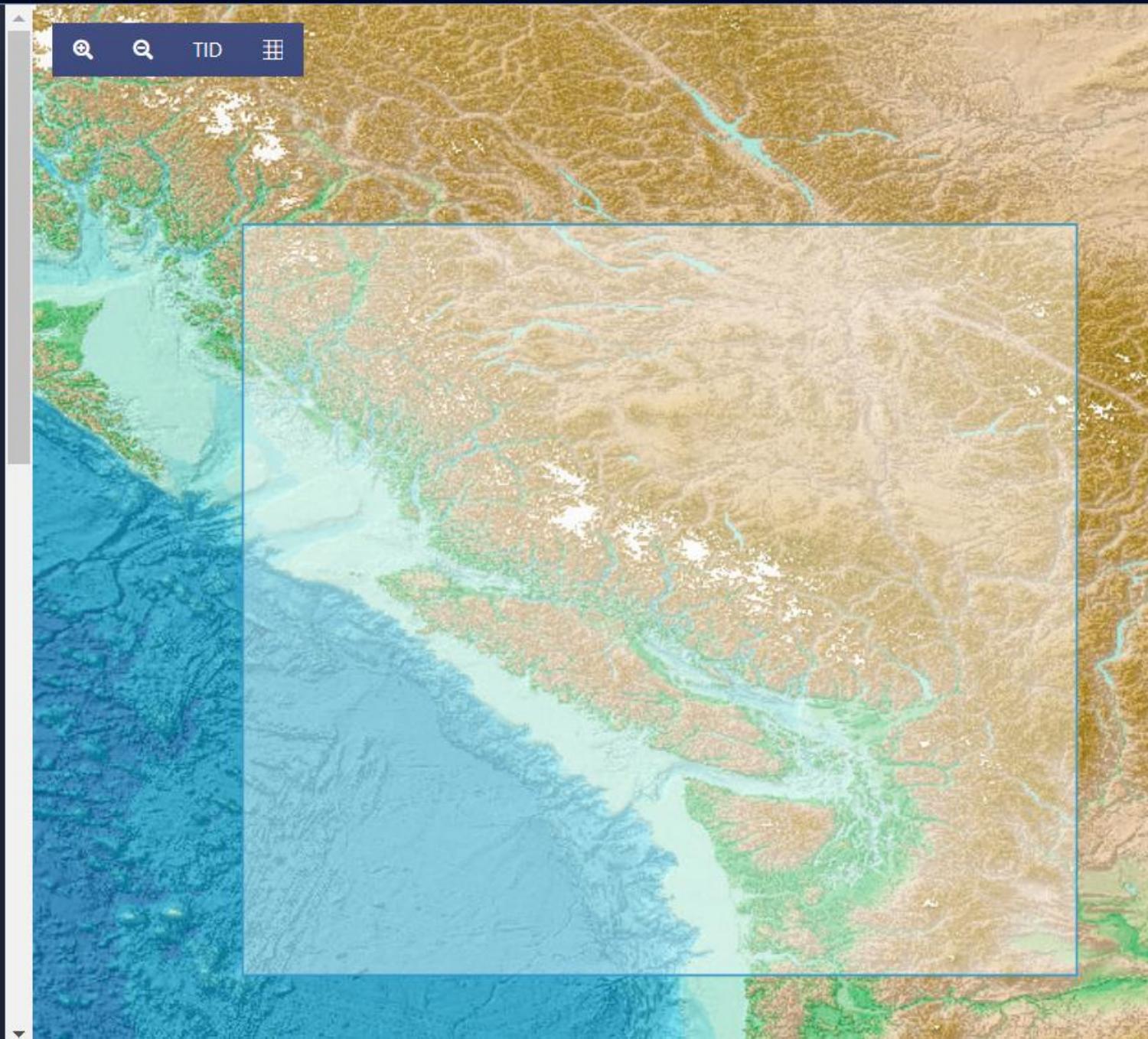
55

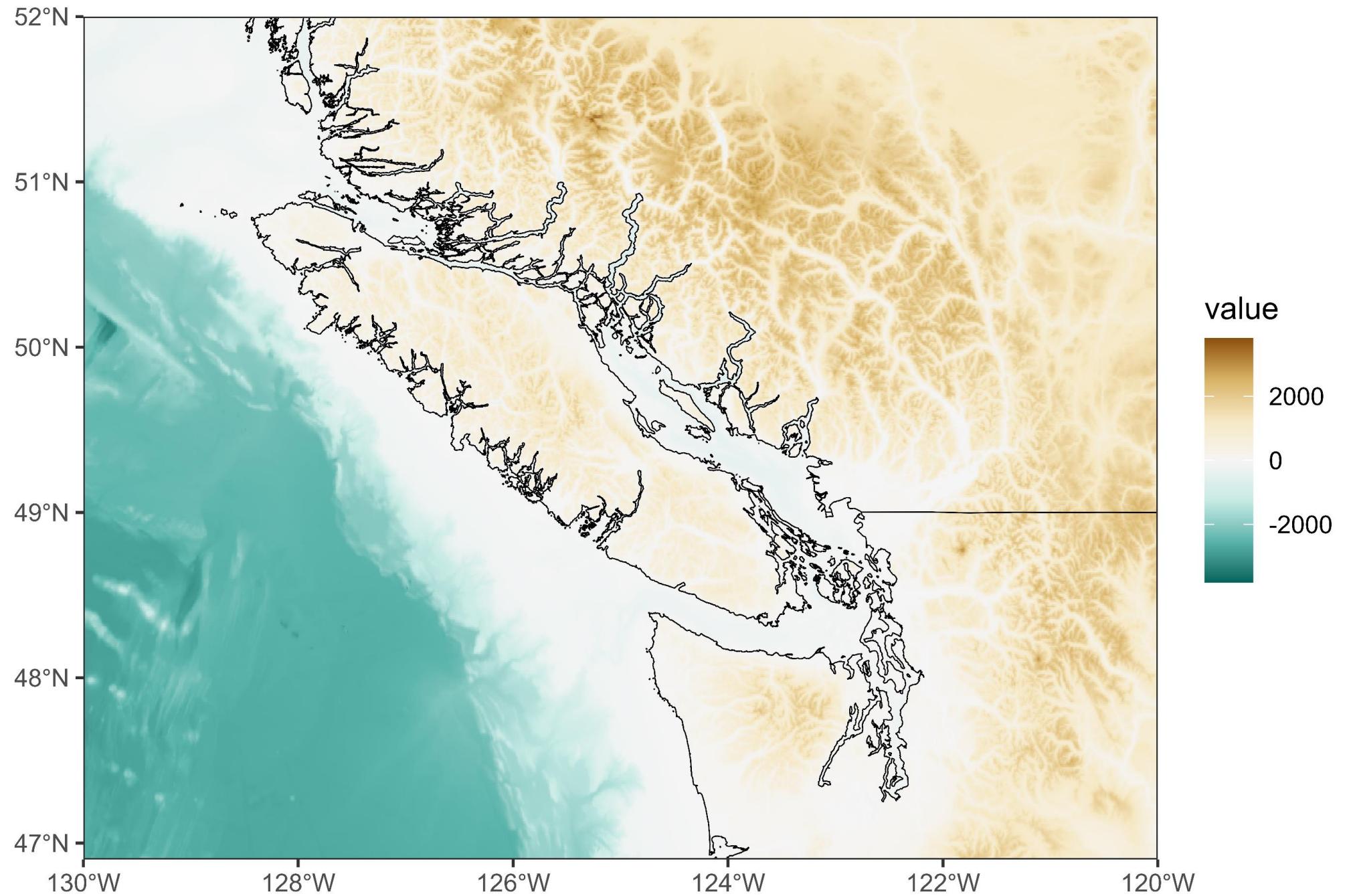
-130 -120

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

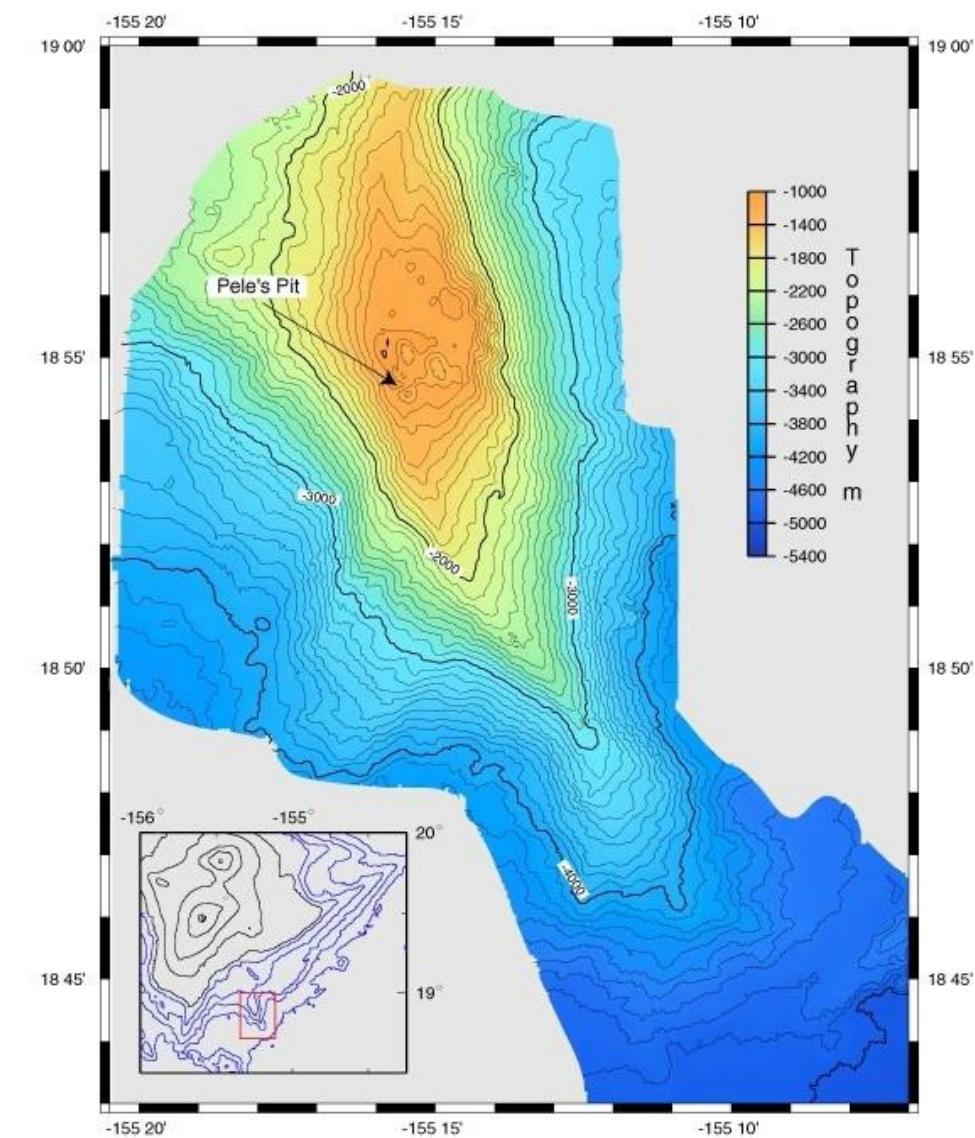
terra package: raster data

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° 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

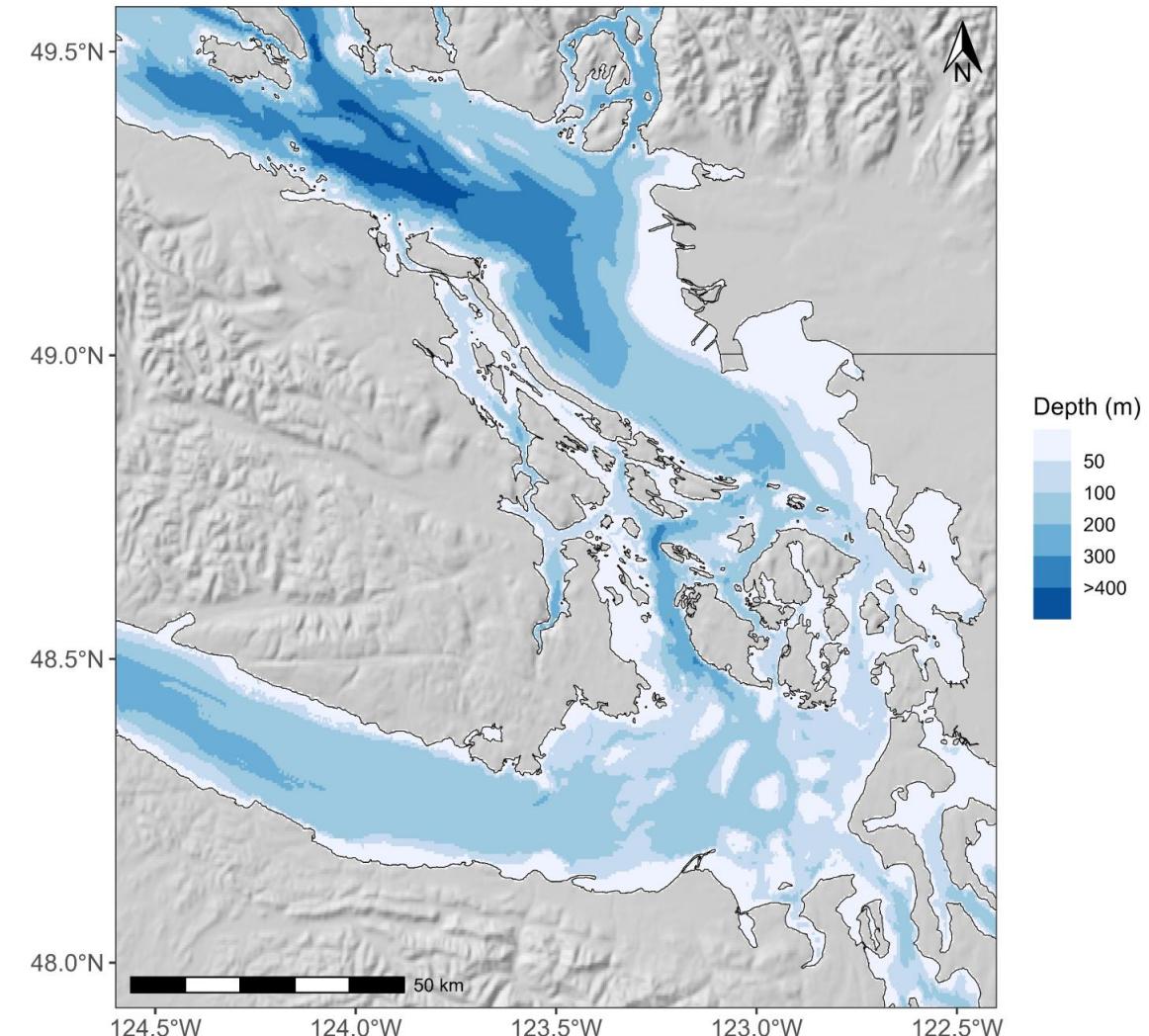


ArcGIS

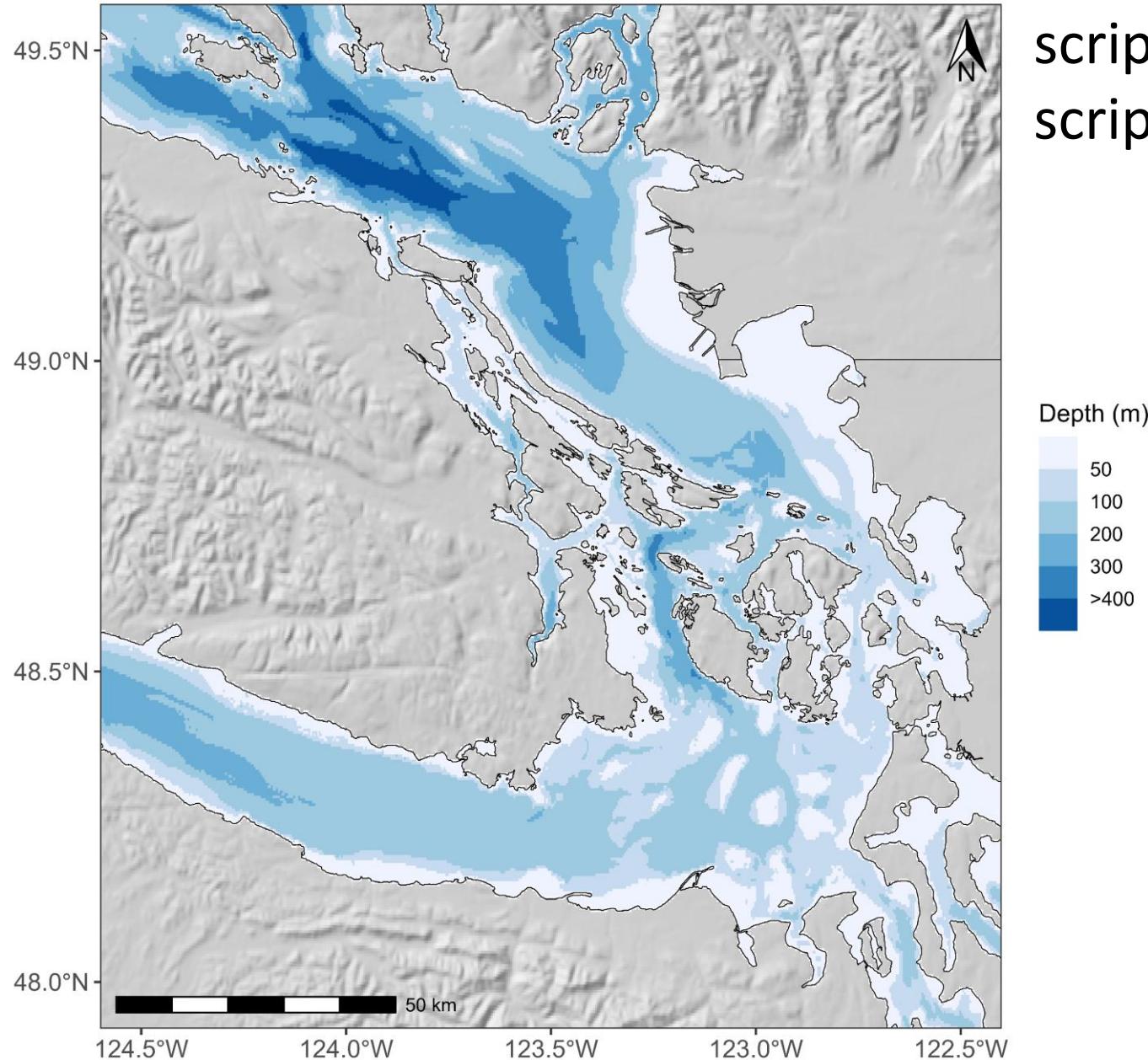
Add bathymetry and topography

scripts/03-study-area-maps.R

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



Add bathymetry and topography



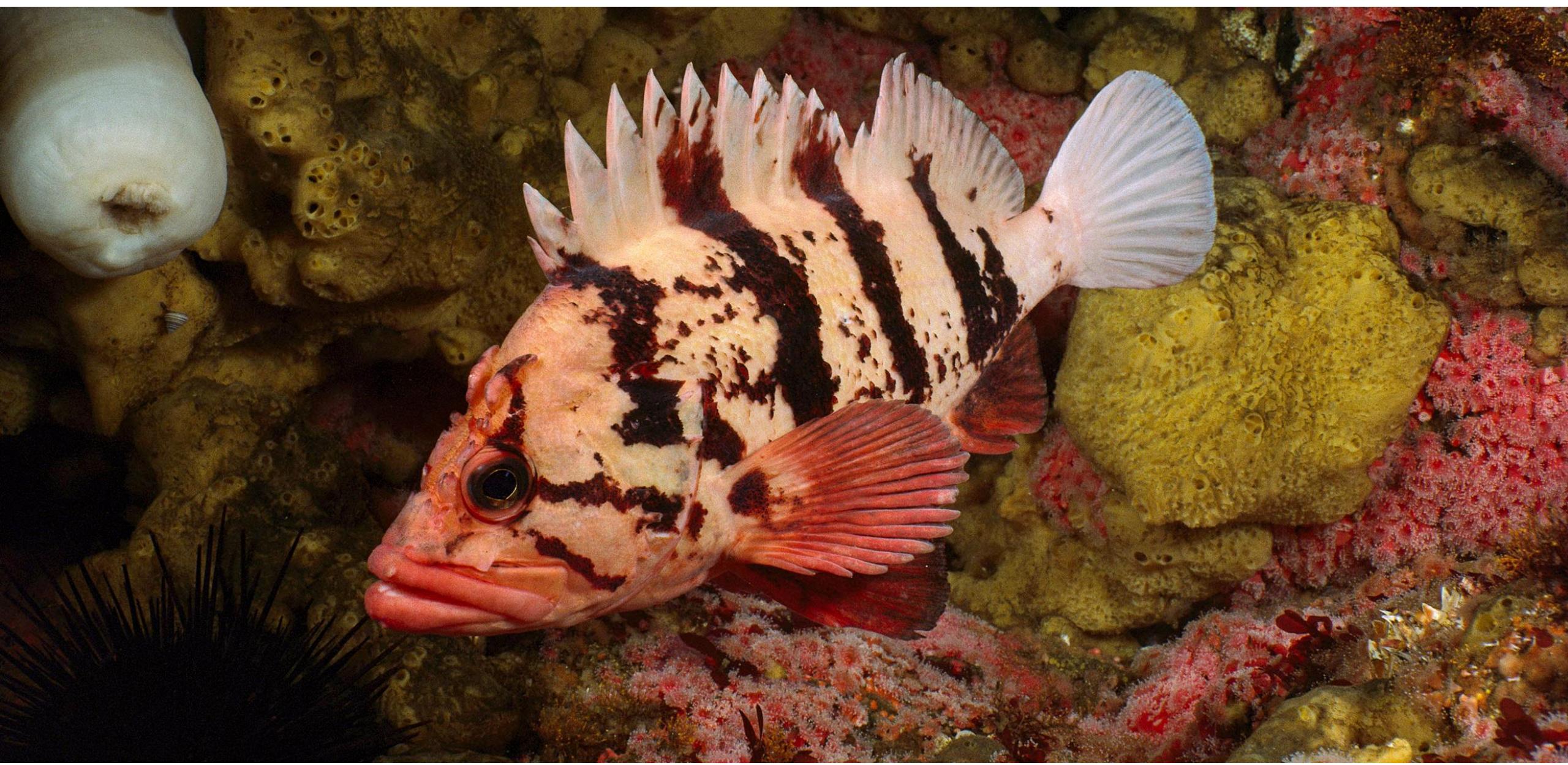
Full code:
scripts/04-bathymetry.R
scripts/05-topography.R

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

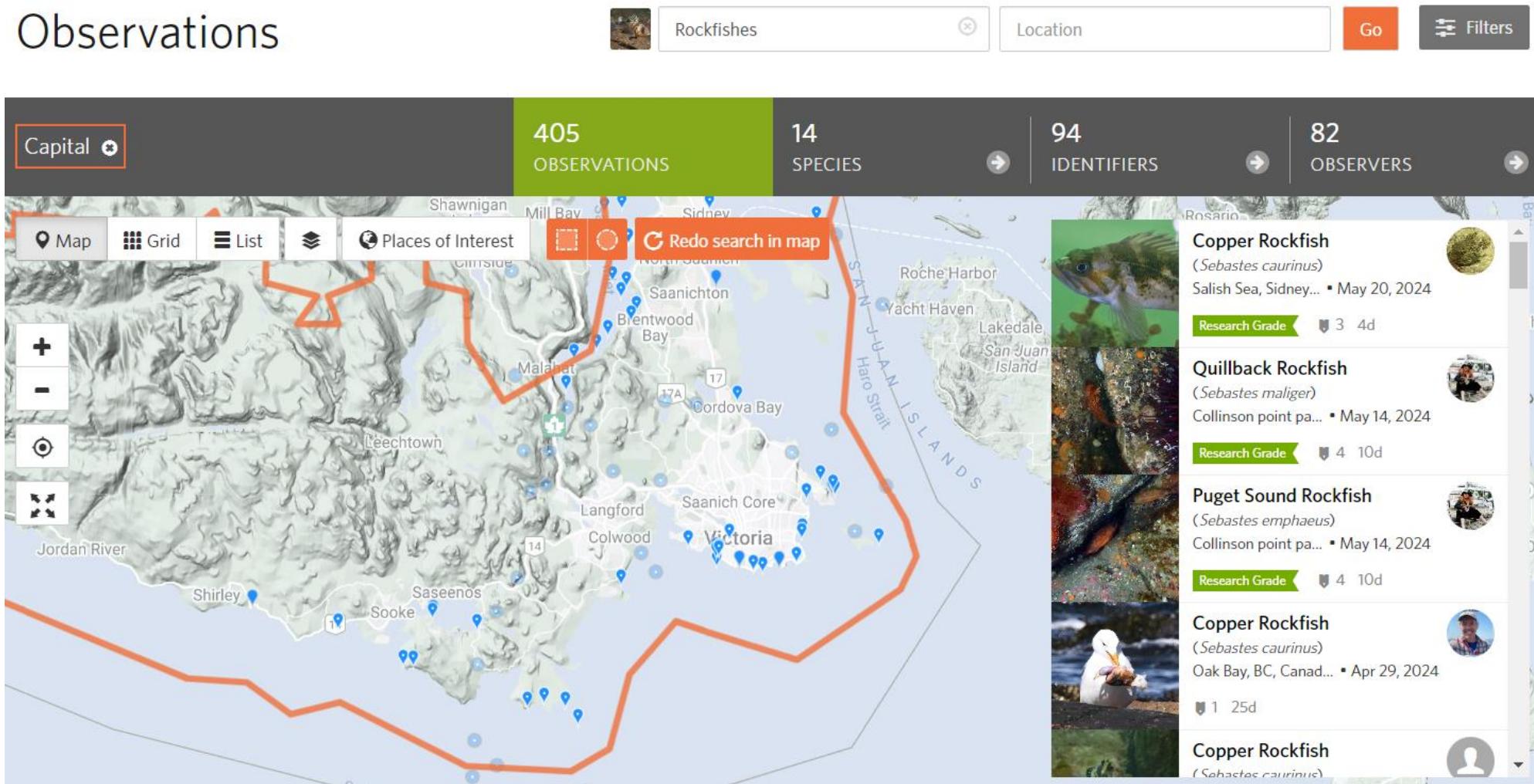
Question: What is the distribution of RCAs relative to rockfish observations on iNaturalist around Victoria?



Hakai

iNaturalist data

Observations



https://www.inaturalist.org/observations?place_id=49140&subview=map&taxon_id=47762

Add rockfish conservation areas

scripts/03-study-area-maps.R

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]]],  
        PRTMEM["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
```

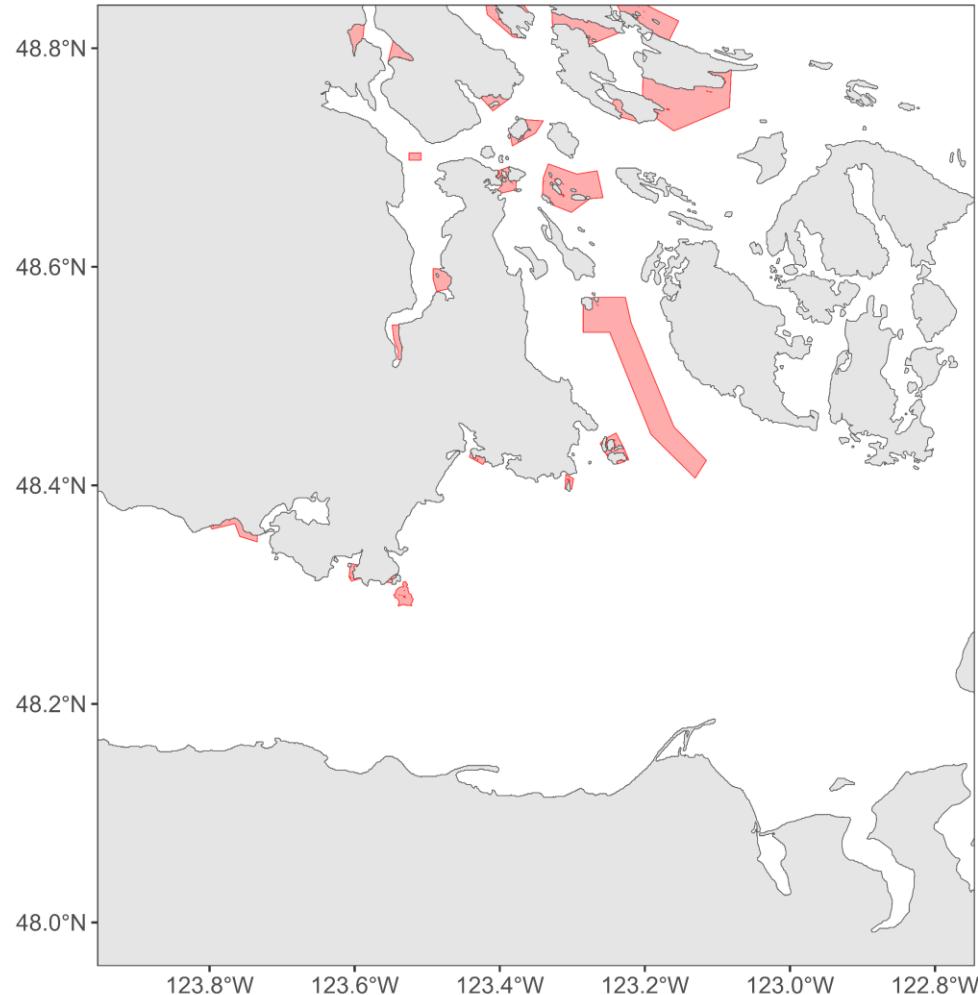
```
wkt:
```

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

Now in WGS 1984 (4326)

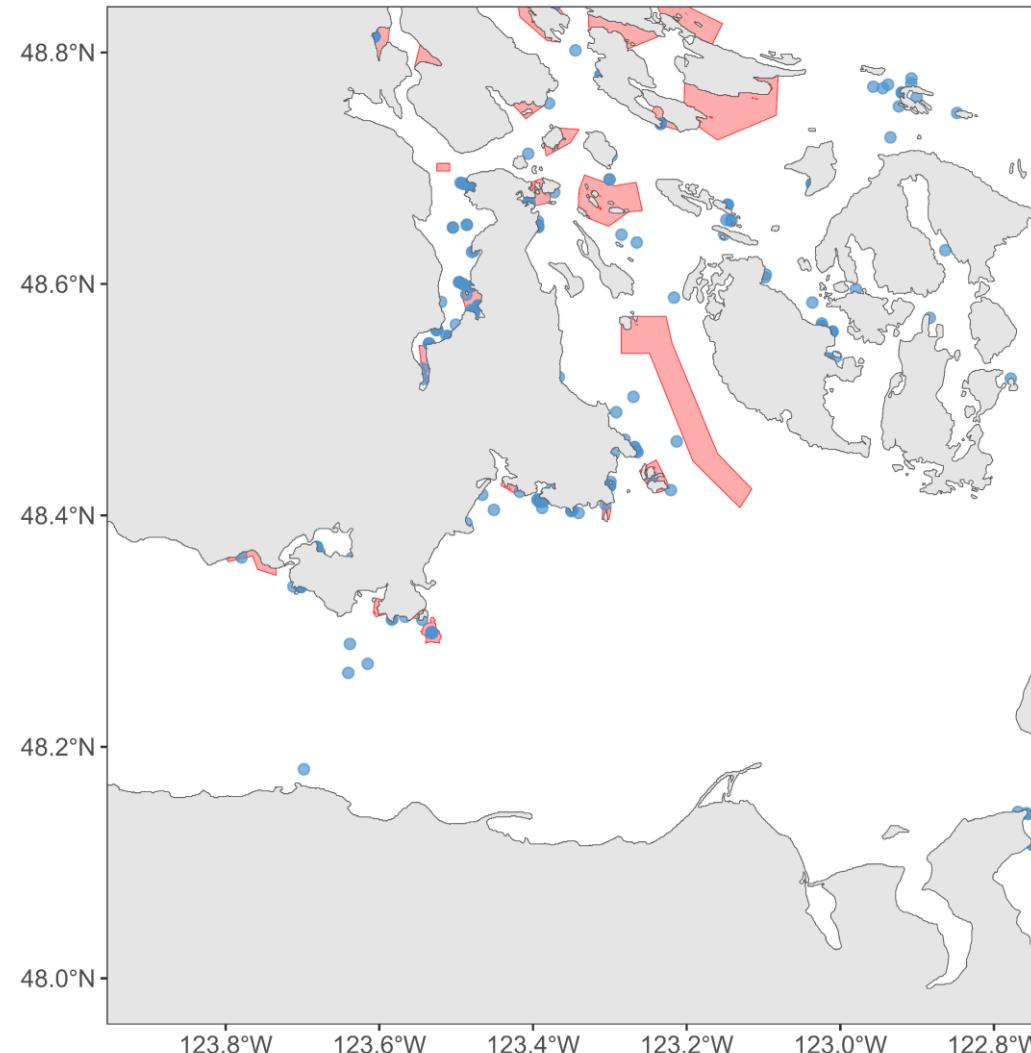
Add rockfish conservation areas

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

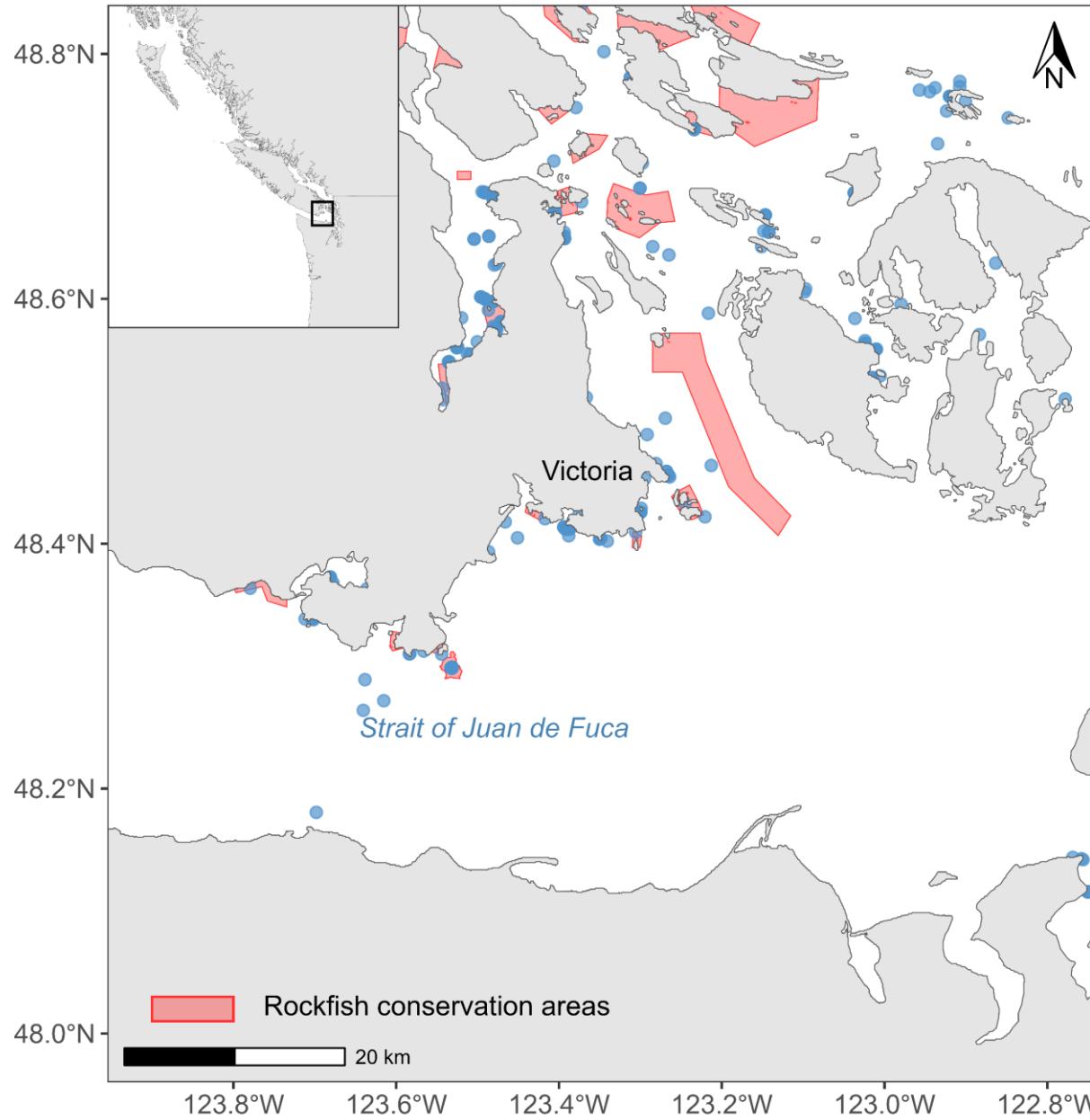


Add iNaturalist observations

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

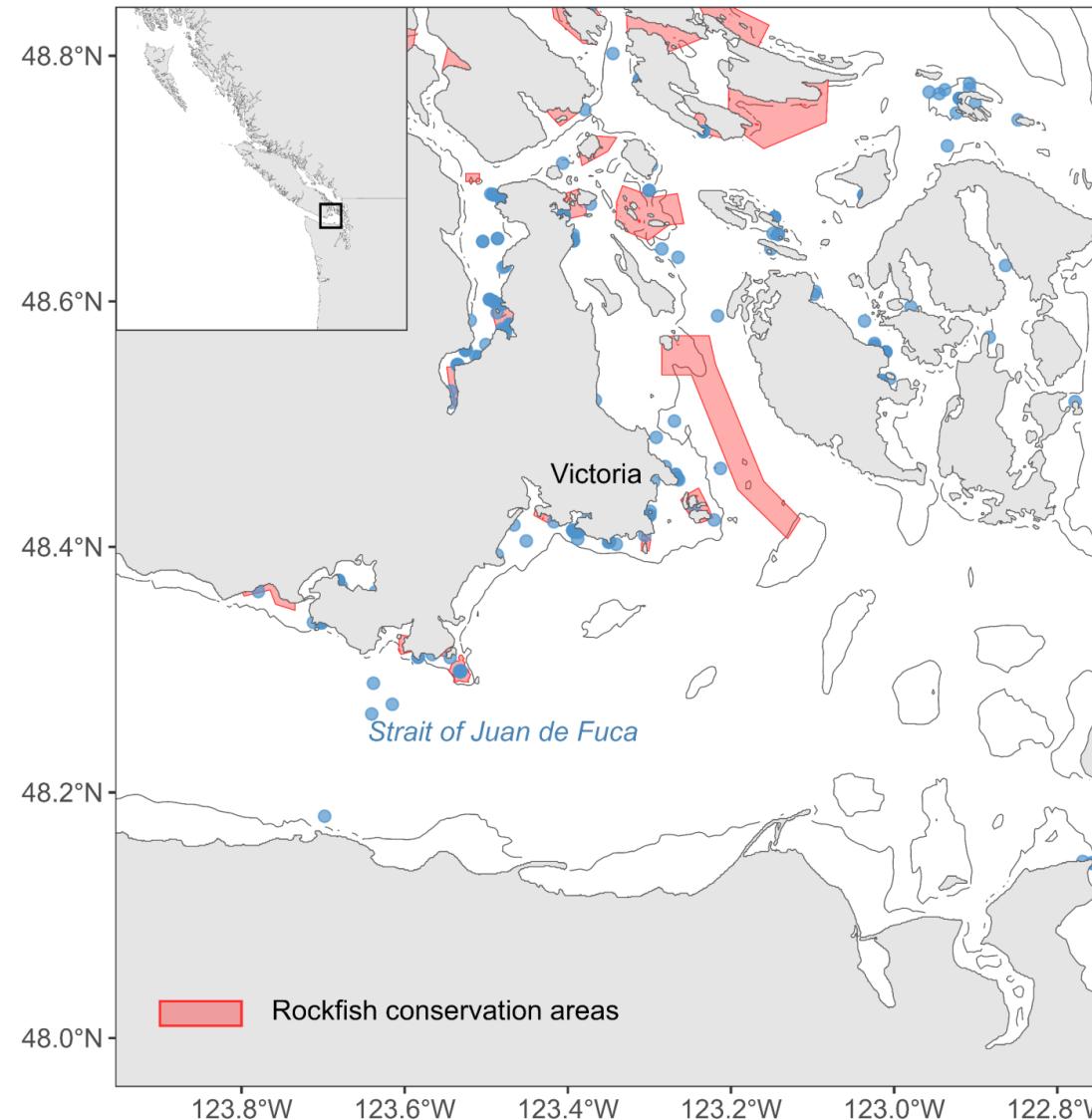


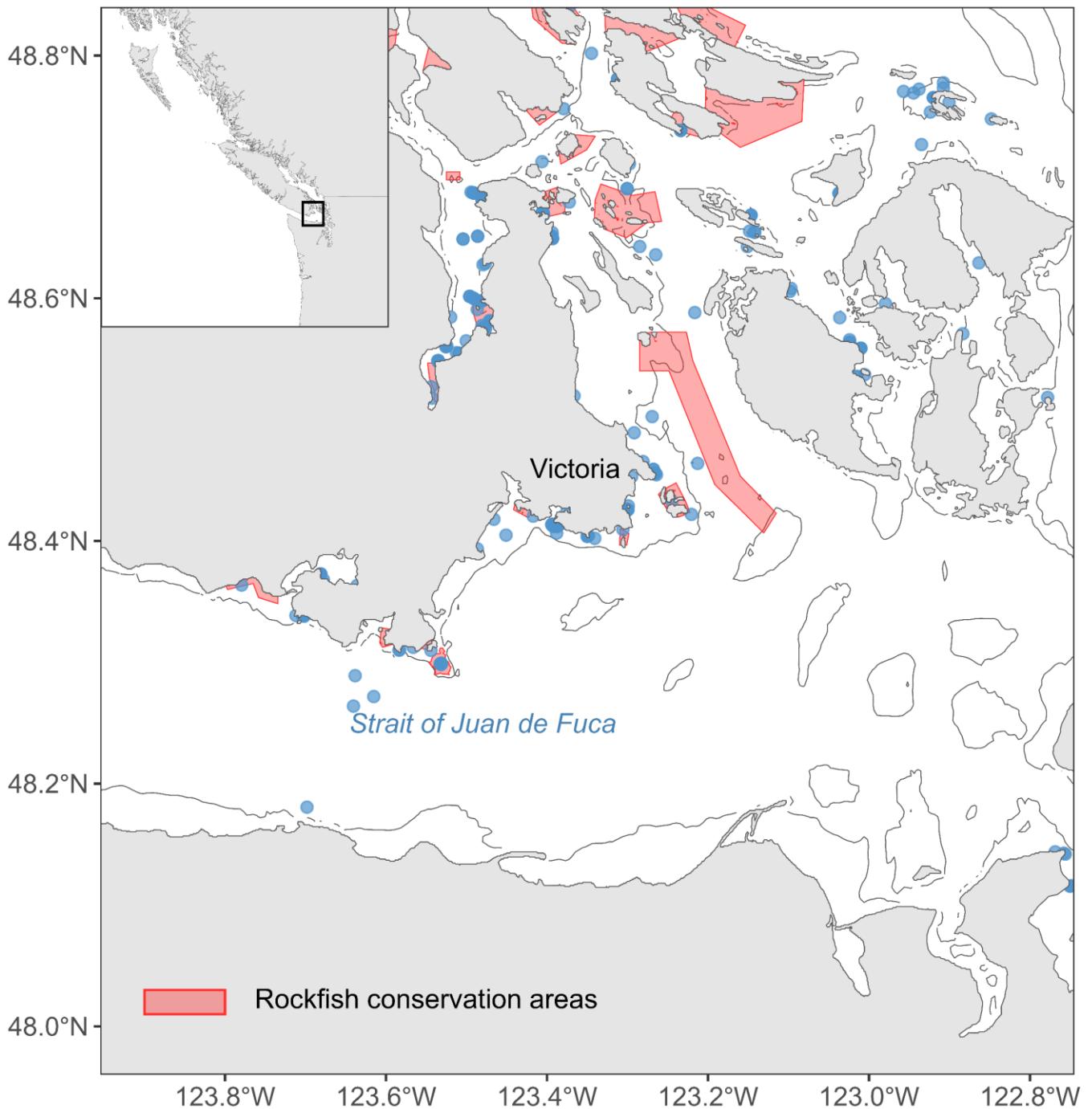
Add inset map, labels



Add 50 metre depth contour

```
geom_spatraster_contour(data = gebco, breaks = c(-50))
```

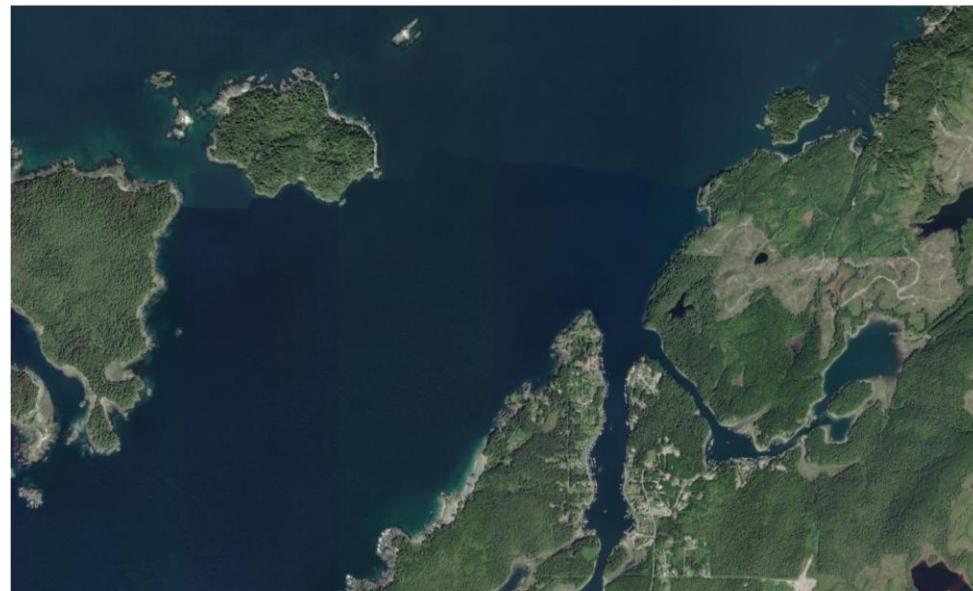




Satellite maps

basemaps or **OpenStreetMap**
packages

Cool way to show your study
area!



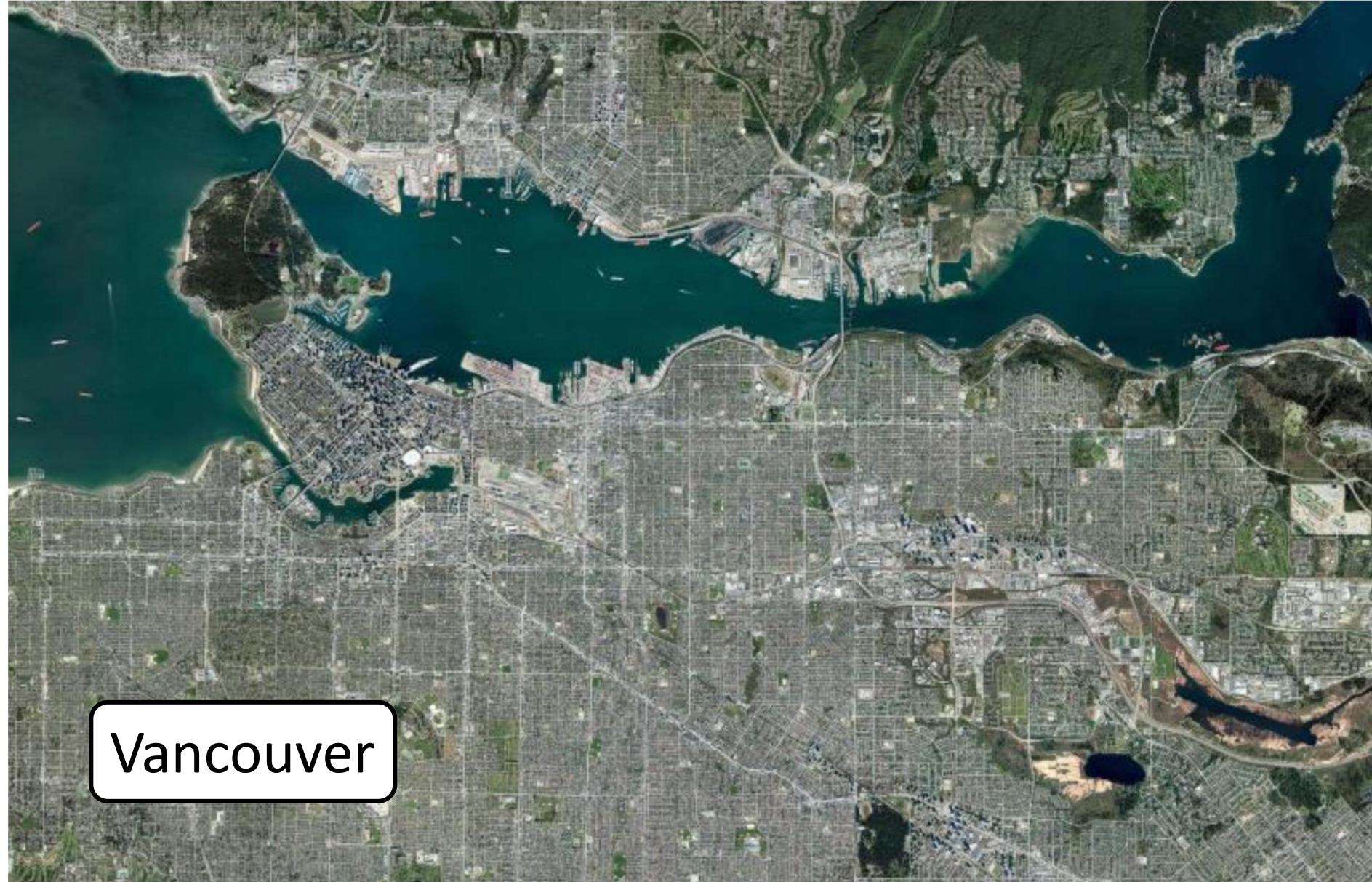
scripts/06-satellite-maps.R

Satellite maps



Cape Town

Satellite maps



Satellite maps

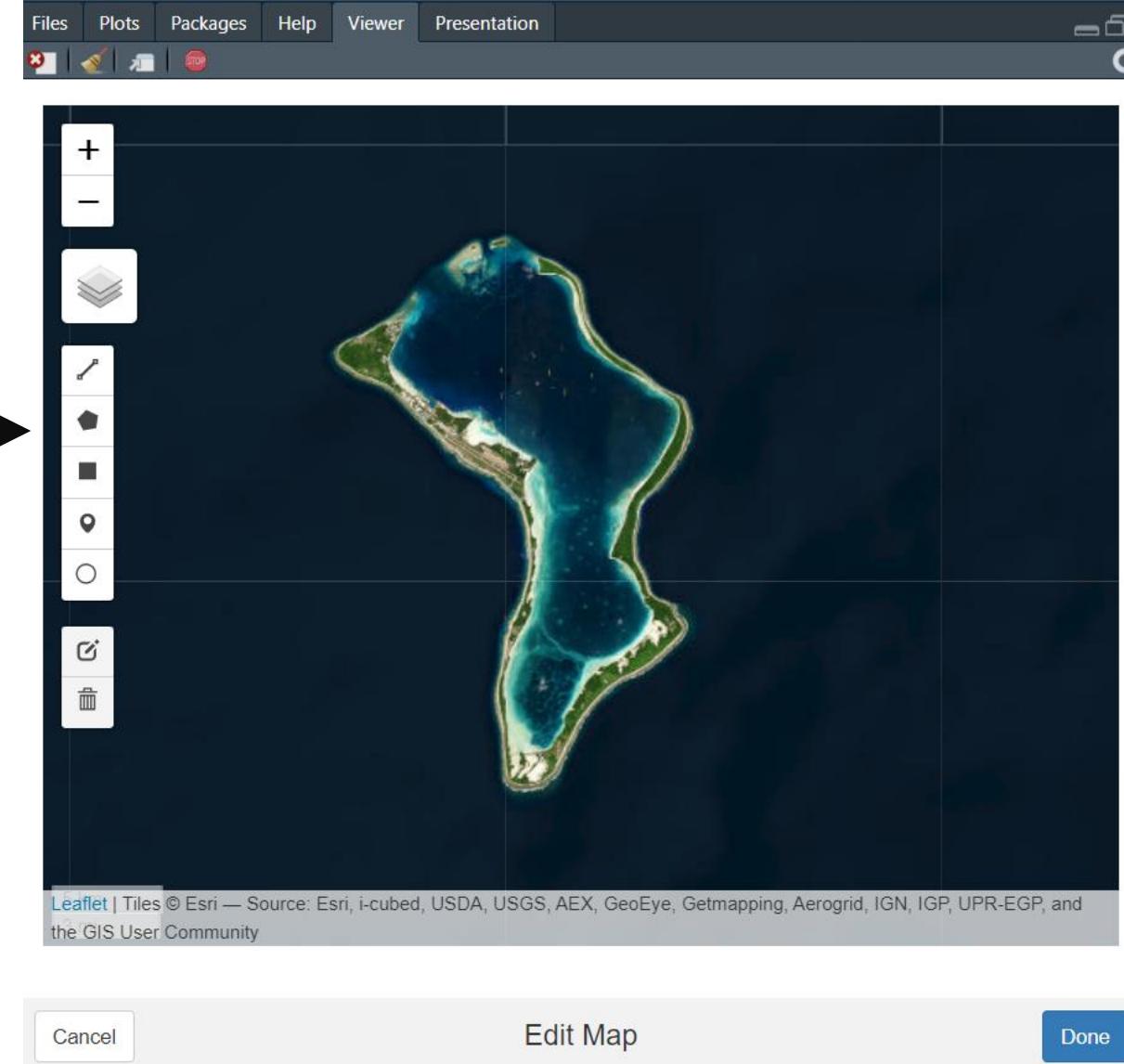
basemaps package

1. Define bounding box, either with

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



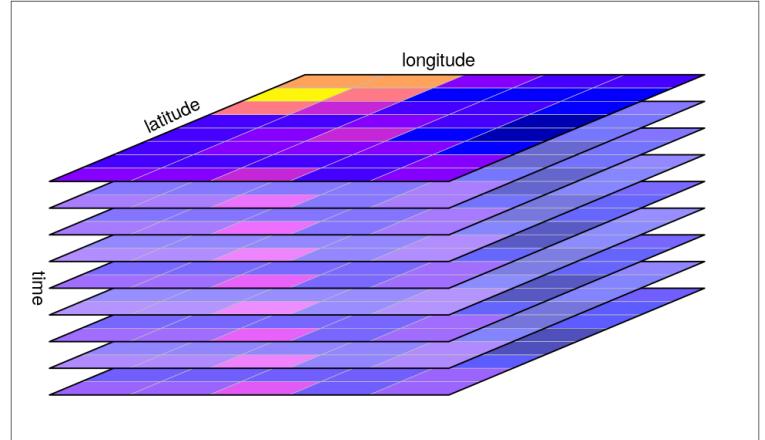
```
bc.box <- st_bbox(c(xmin = -142.3, ymin = 43,  
                     xmax = -110.5, ymax = 58),  
                     crs = st_crs(4326))
```



Satellite maps

```
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

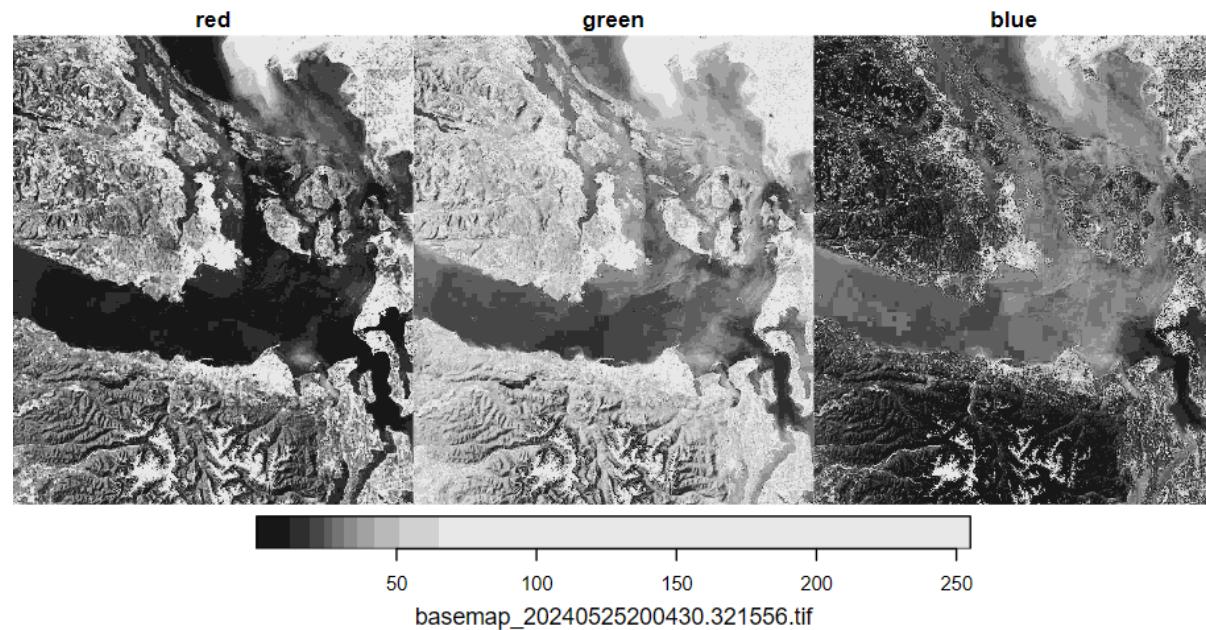


Satellite maps

```
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

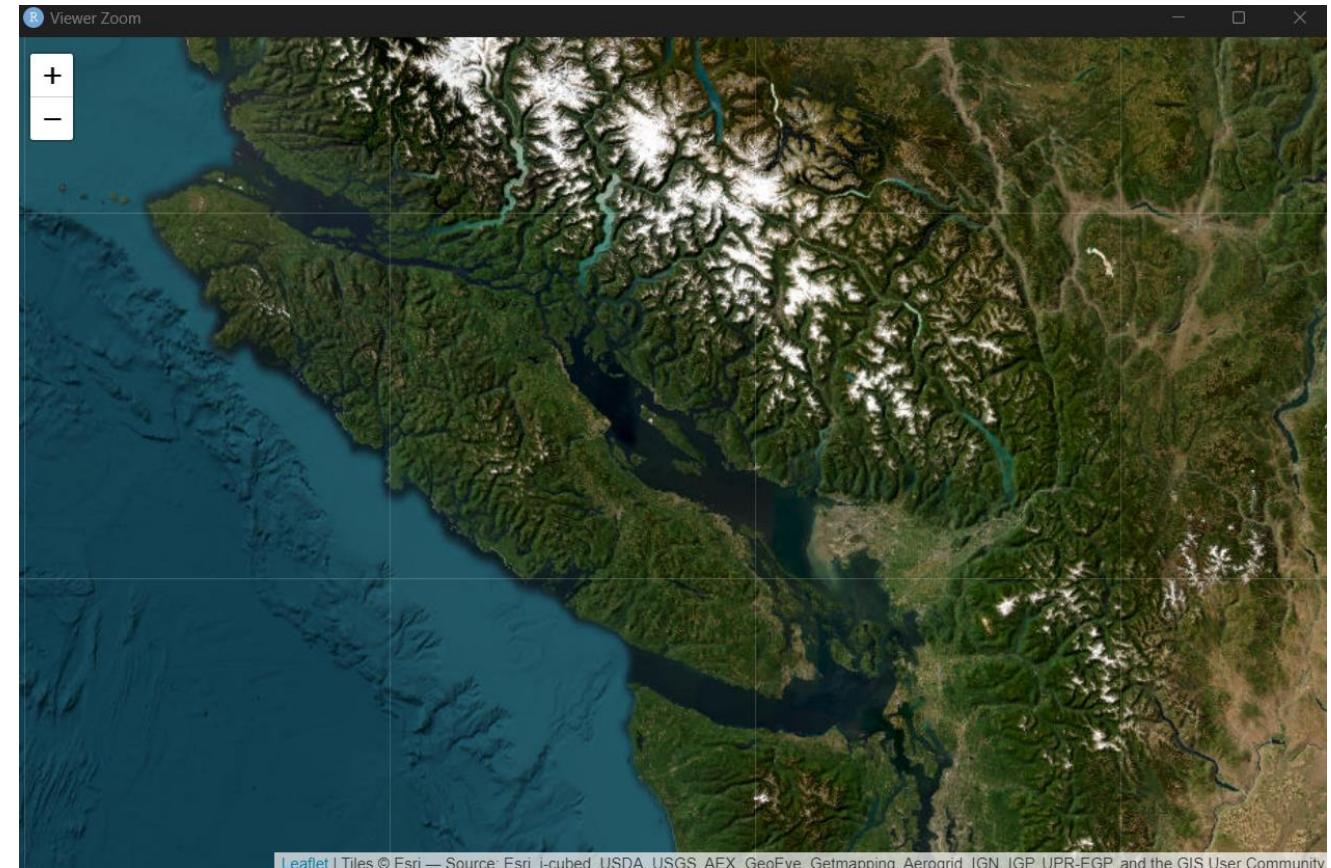




Interactive maps with leaflet

leaflet is a Javascript library to make interactive maps

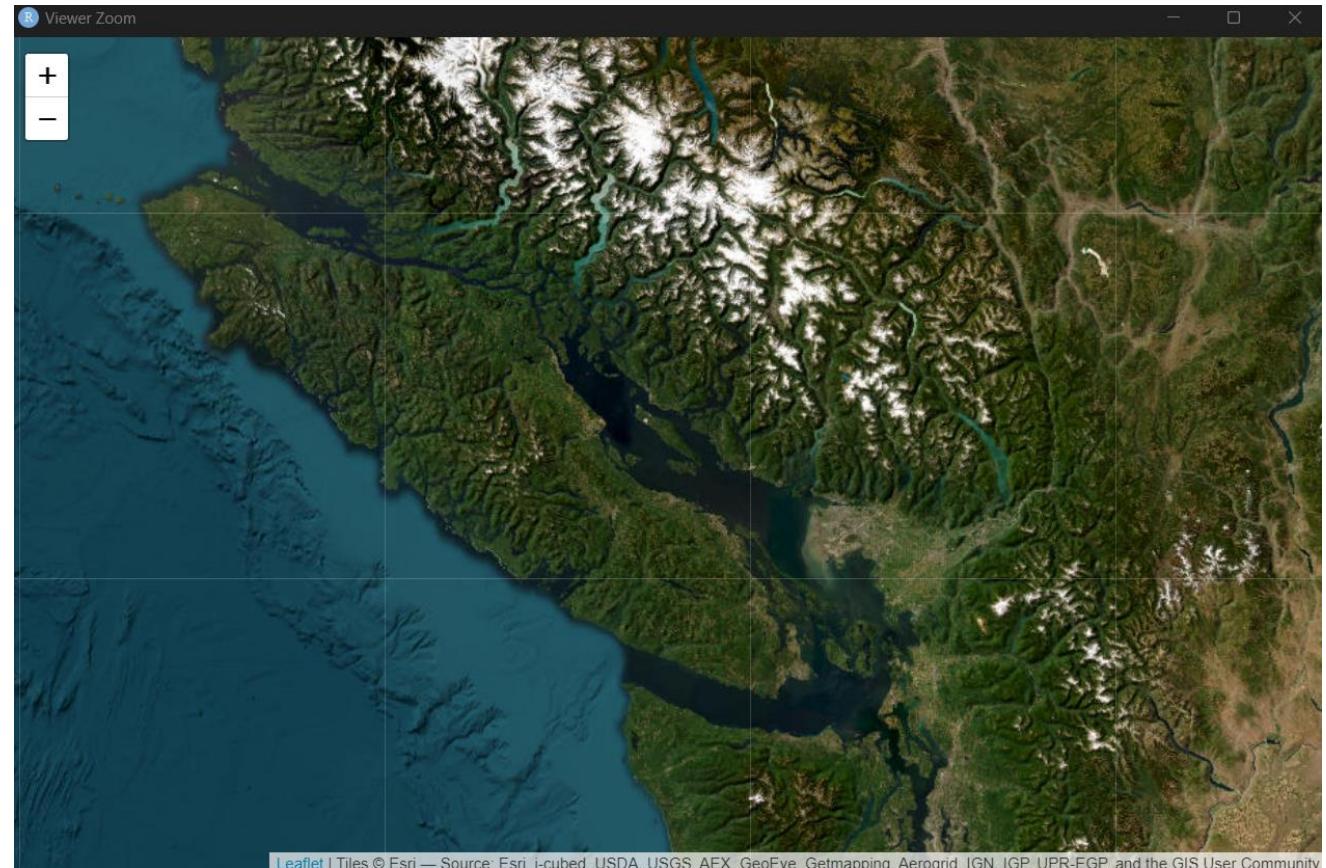
The **leaflet** package allows us to make these maps in R



Interactive maps with leaflet

leaflet is a Javascript library to make interactive maps

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

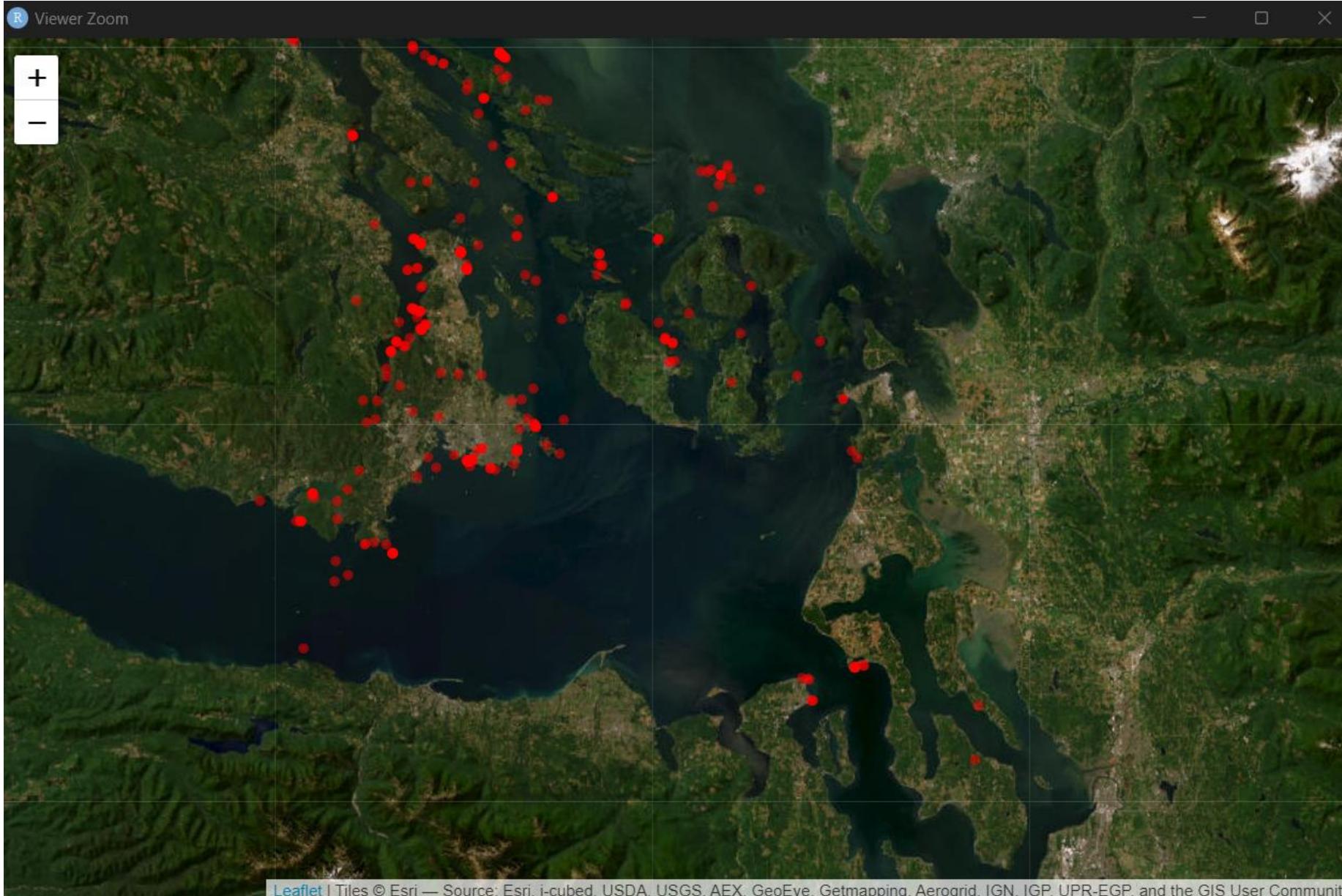


<https://rstudio.github.io/leaflet/>

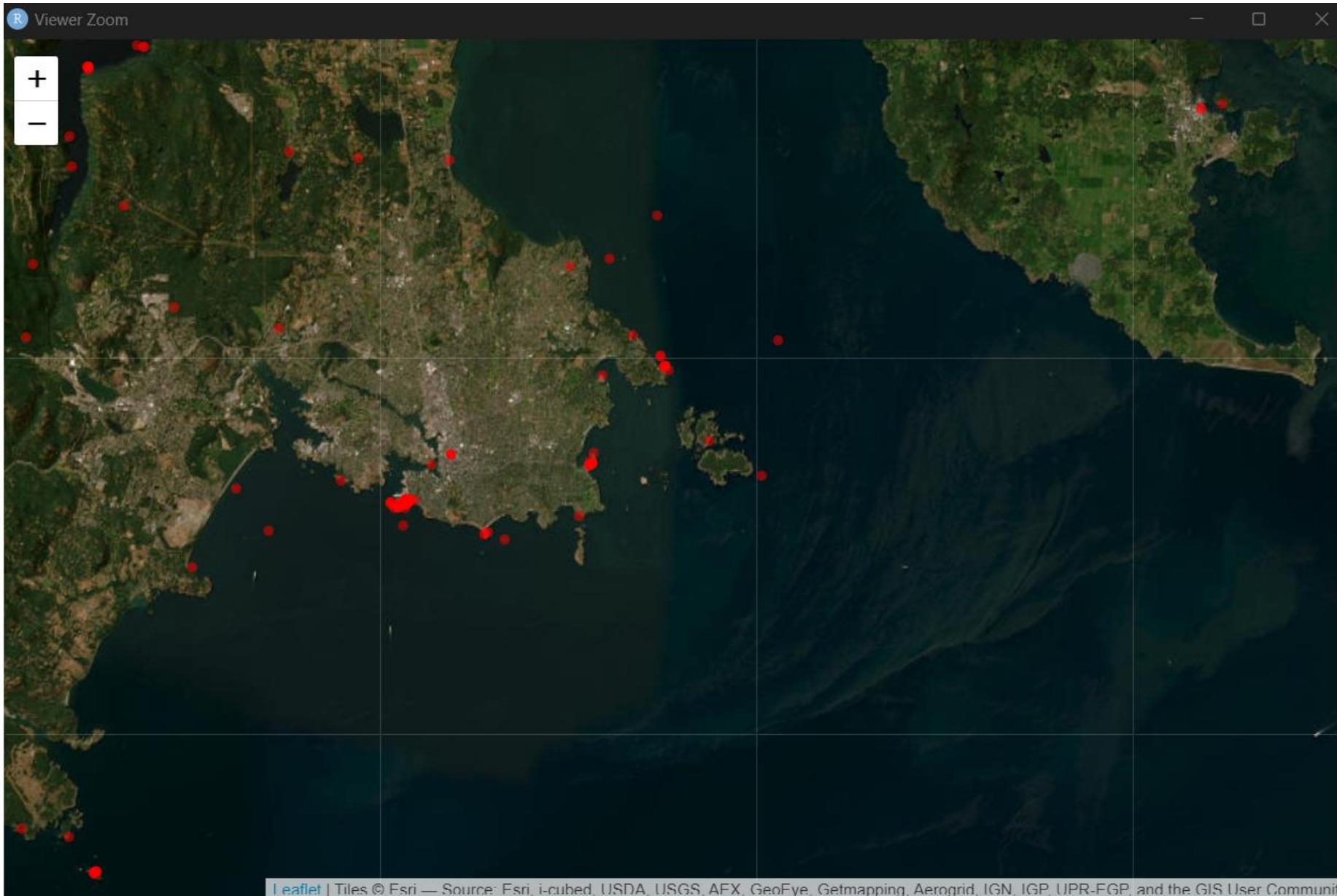
Note: current incompatibility with R 4.4:

<https://github.com/rstudio/rstudio/issues/14603>

Interactive maps with leaflet

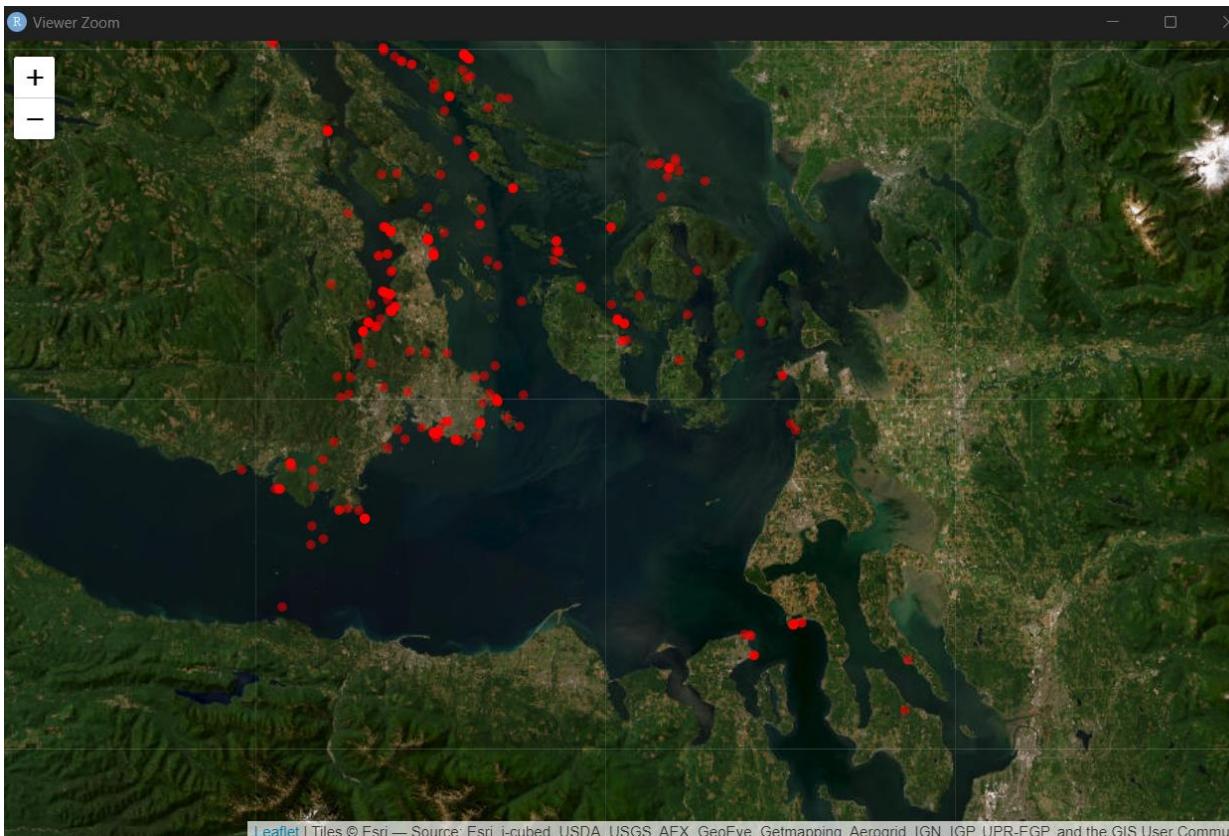


Interactive maps with leaflet



Interactive maps with leaflet

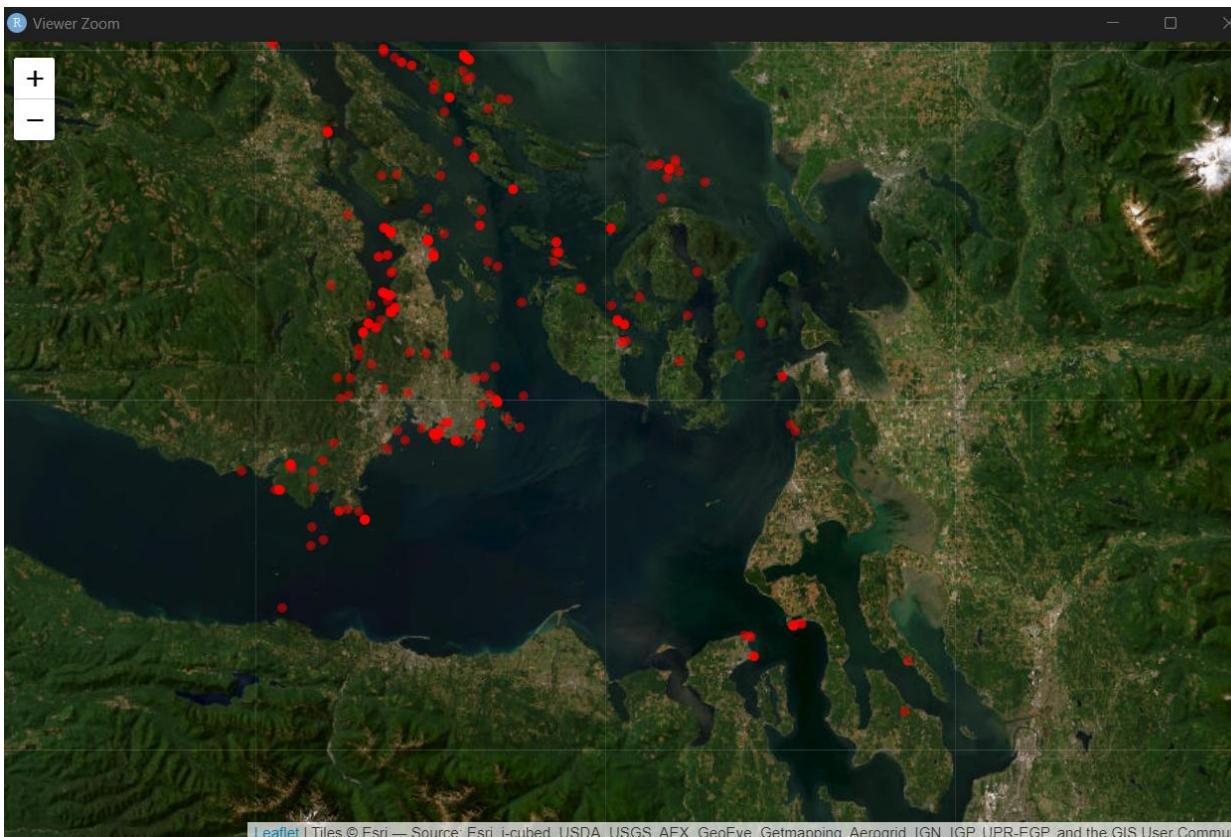
```
leaflet() |>  
  setView(lng = -123, lat = 48.4, zoom = 9) |>  
  addCircles(data = rockfish,  
             lng = rockfish$longitude,  
             lat = rockfish$latitude) |>  
  addProviderTiles(providers$Esri.WorldImagery)
```



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)
```



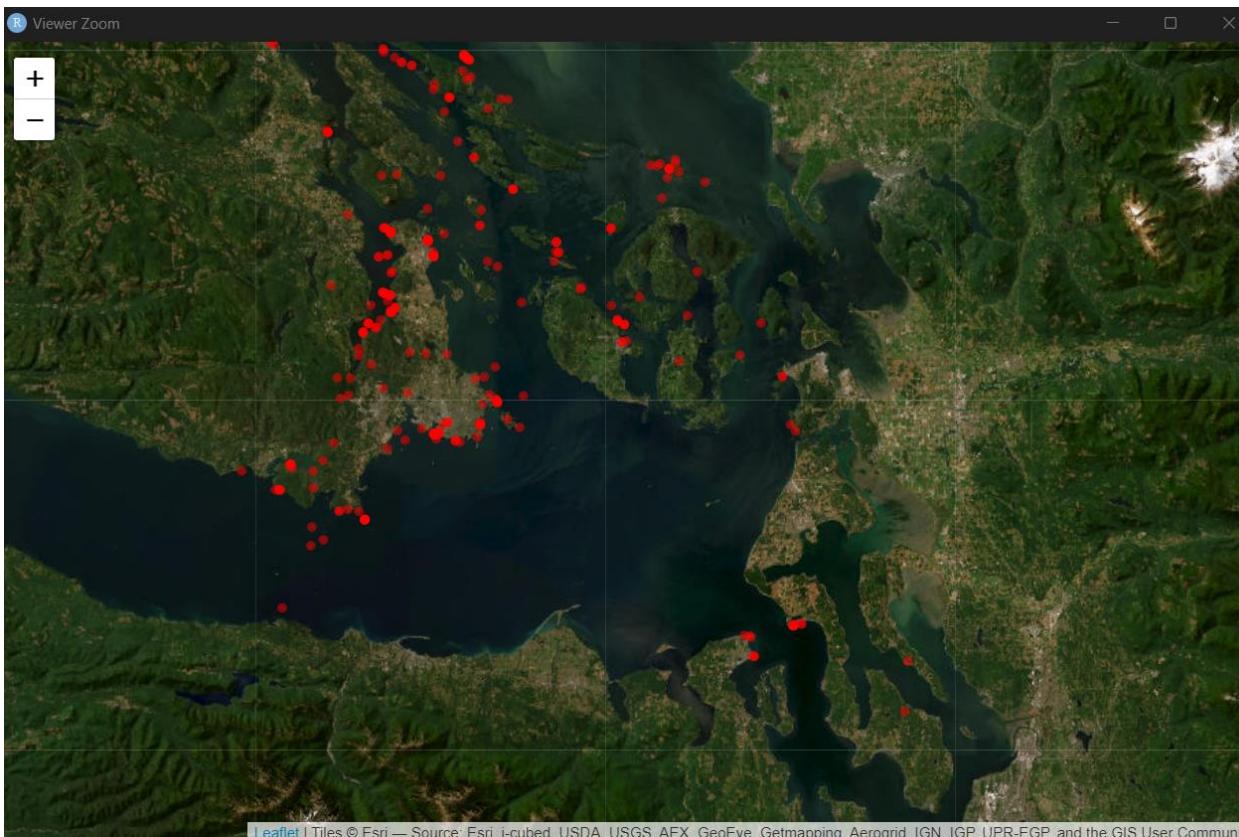
Interactive maps with leaflet

```
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”



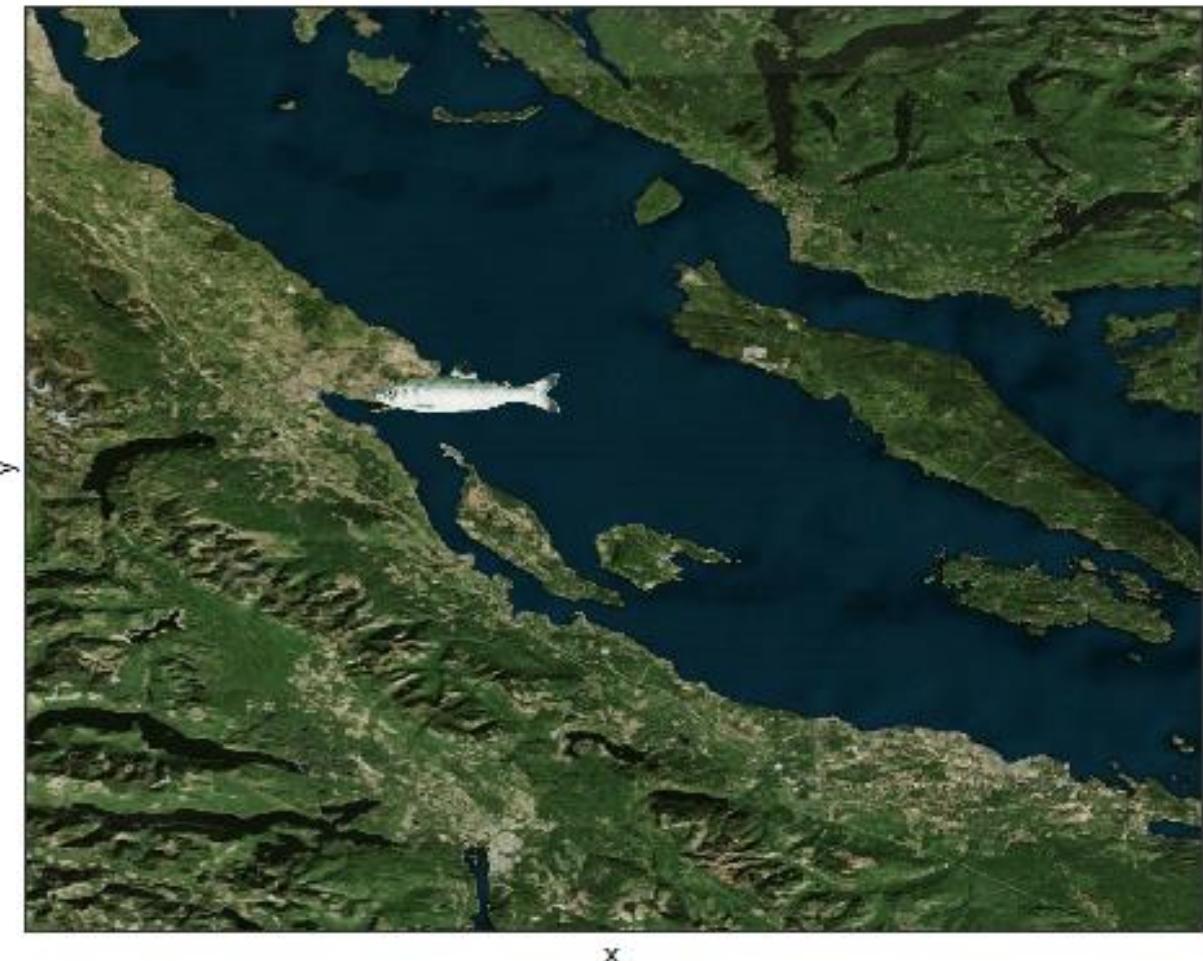
Animated maps

scripts/08-animated-maps.R

ganimate package

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

DateTime: 2022-10-15

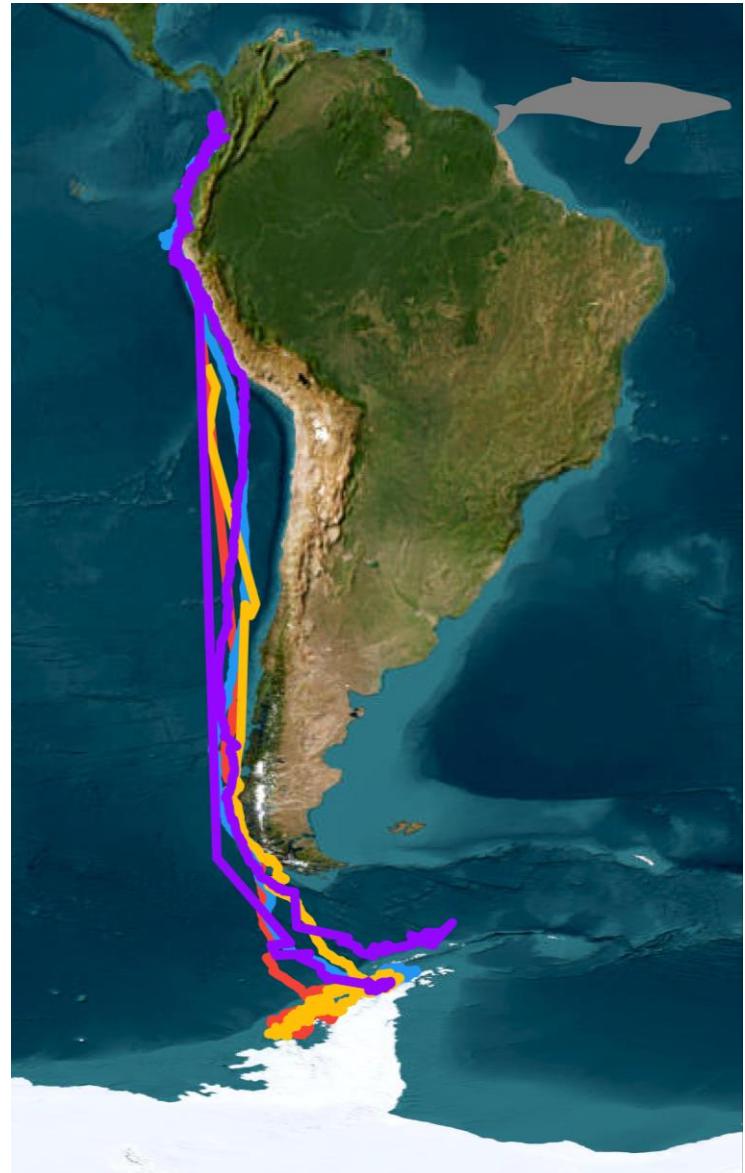


Animated maps

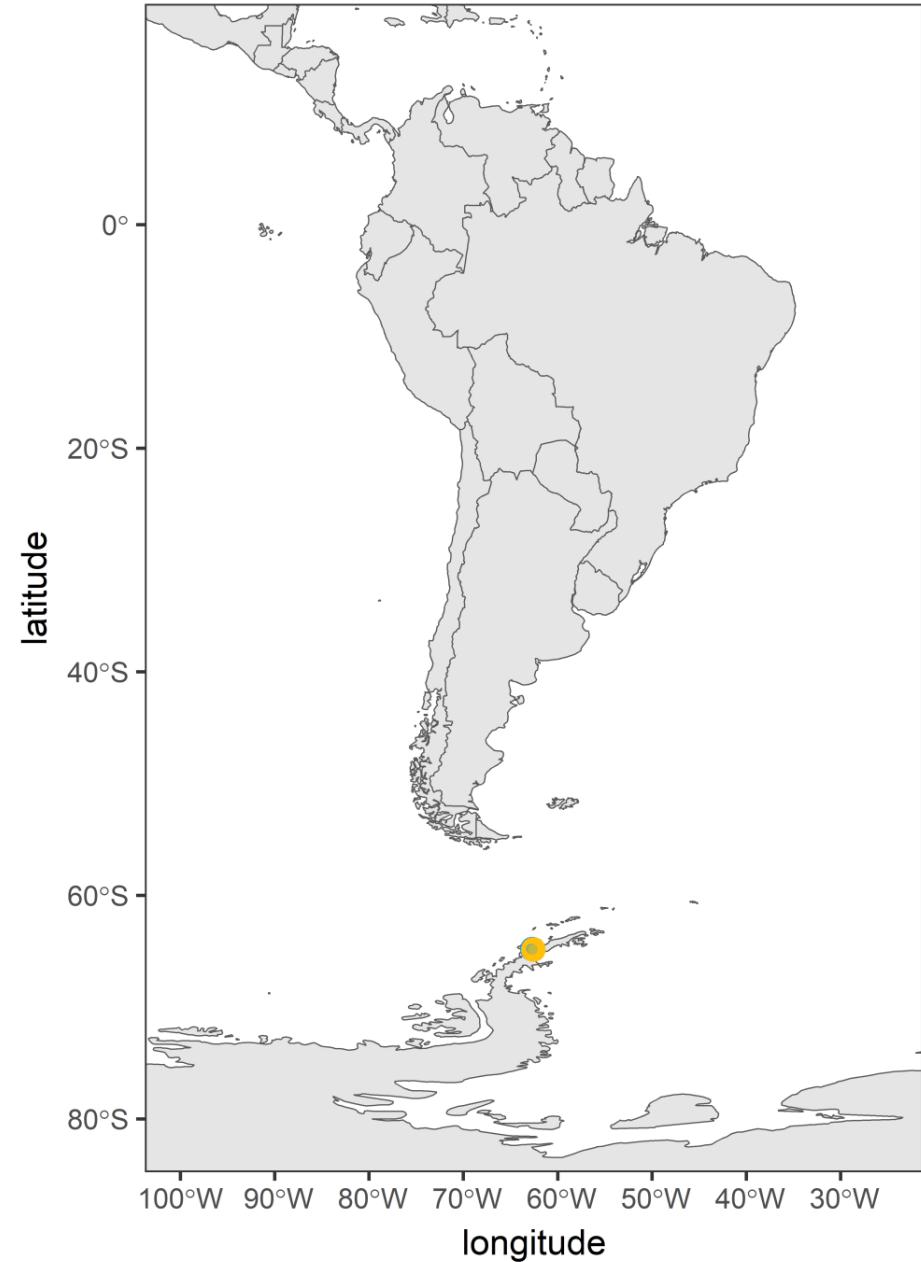
Date: 2023-01-15



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^{1*} , Ladd Irvine¹ and Ari S. Friedlaender^{1,2}

RESEARCH

Open Access



Check for updates

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

Michelle Modest^{1*} , Ladd Irvine², Virginia Andrews-Goff³, William Gough⁴, David Johnston⁵, Douglas Nowacek⁵, Logan Pallin¹, Andrew Read⁵, Reny Tyson Moore⁶ and Ari Friedlaender¹



NOAA Fisheries

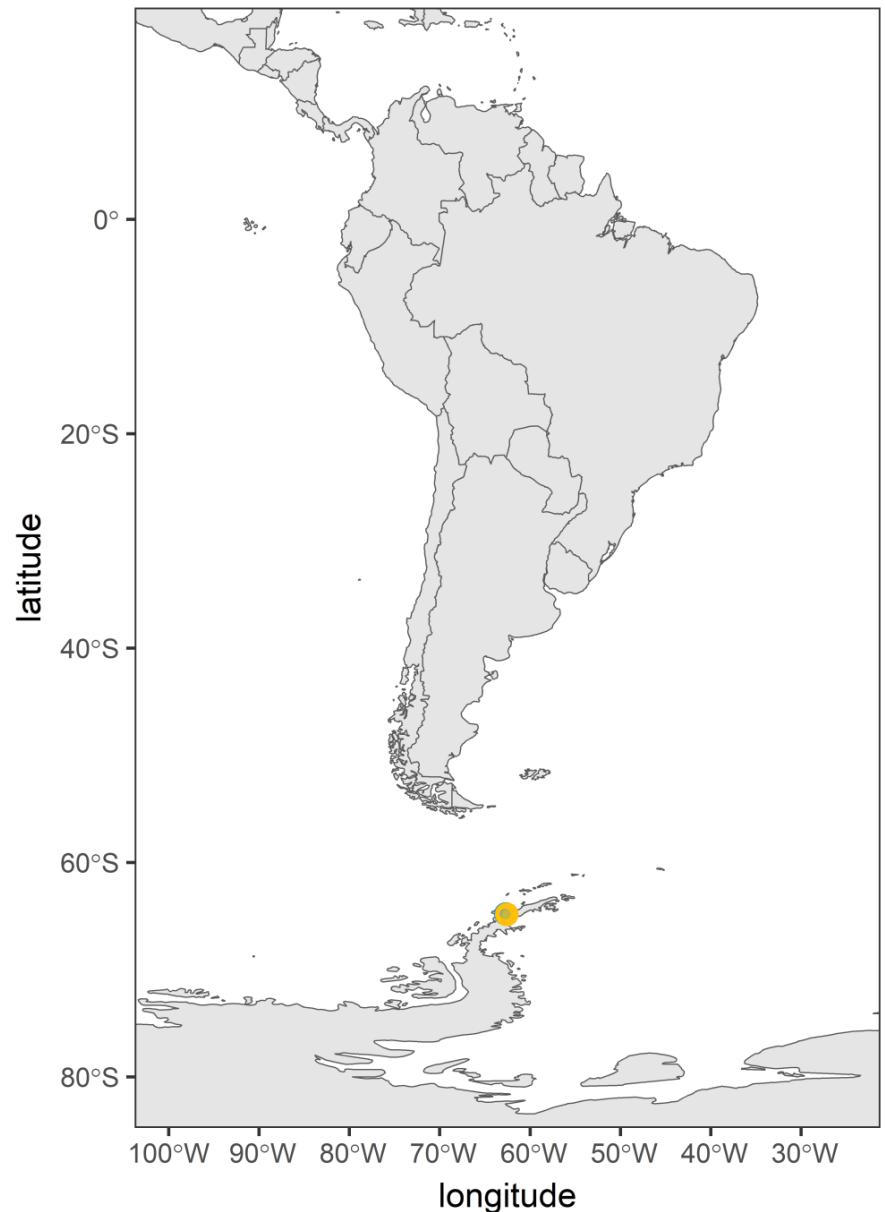


National Geographic

Animated maps

Date: 2013-02-05

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



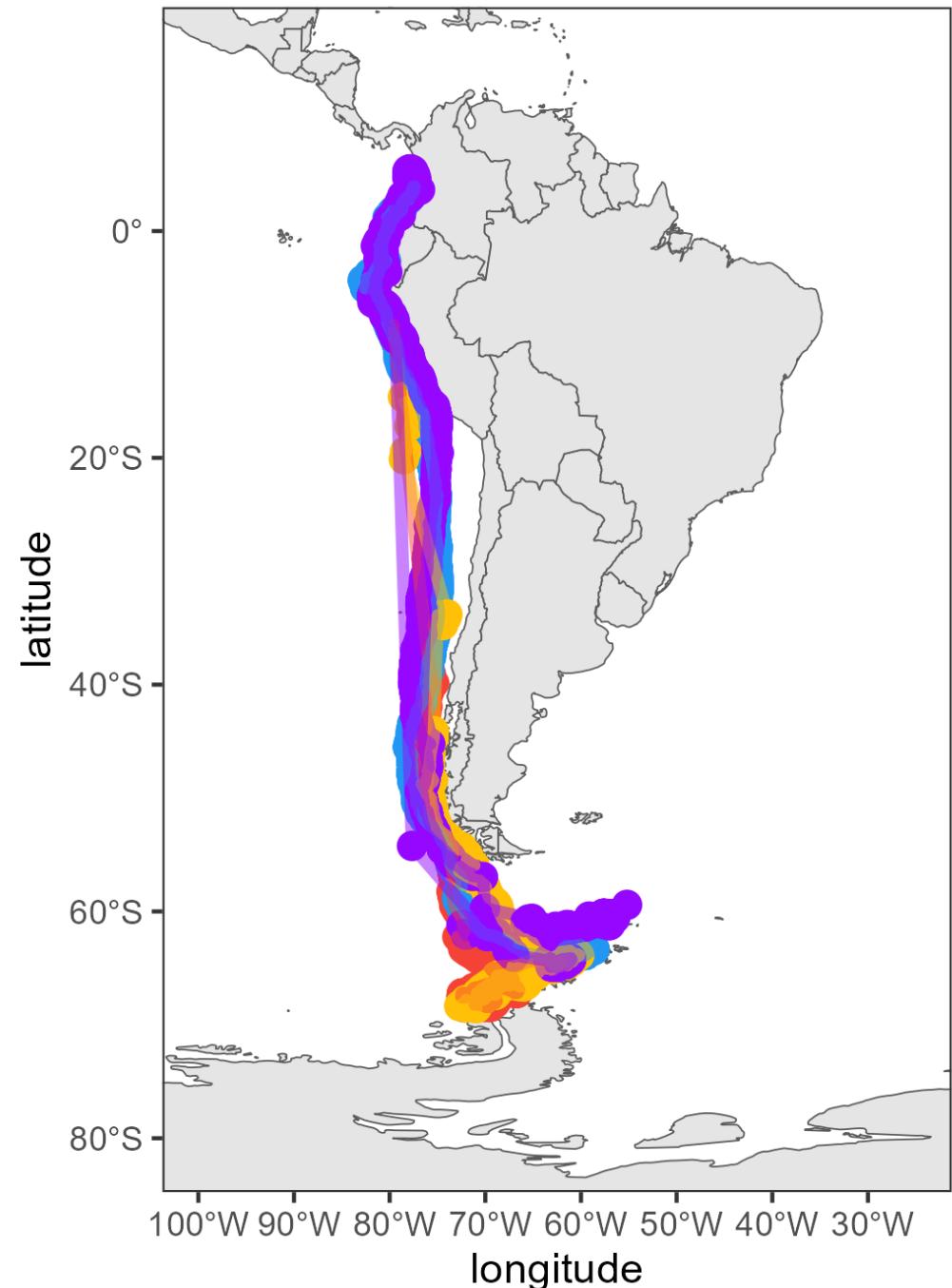
Animated maps

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

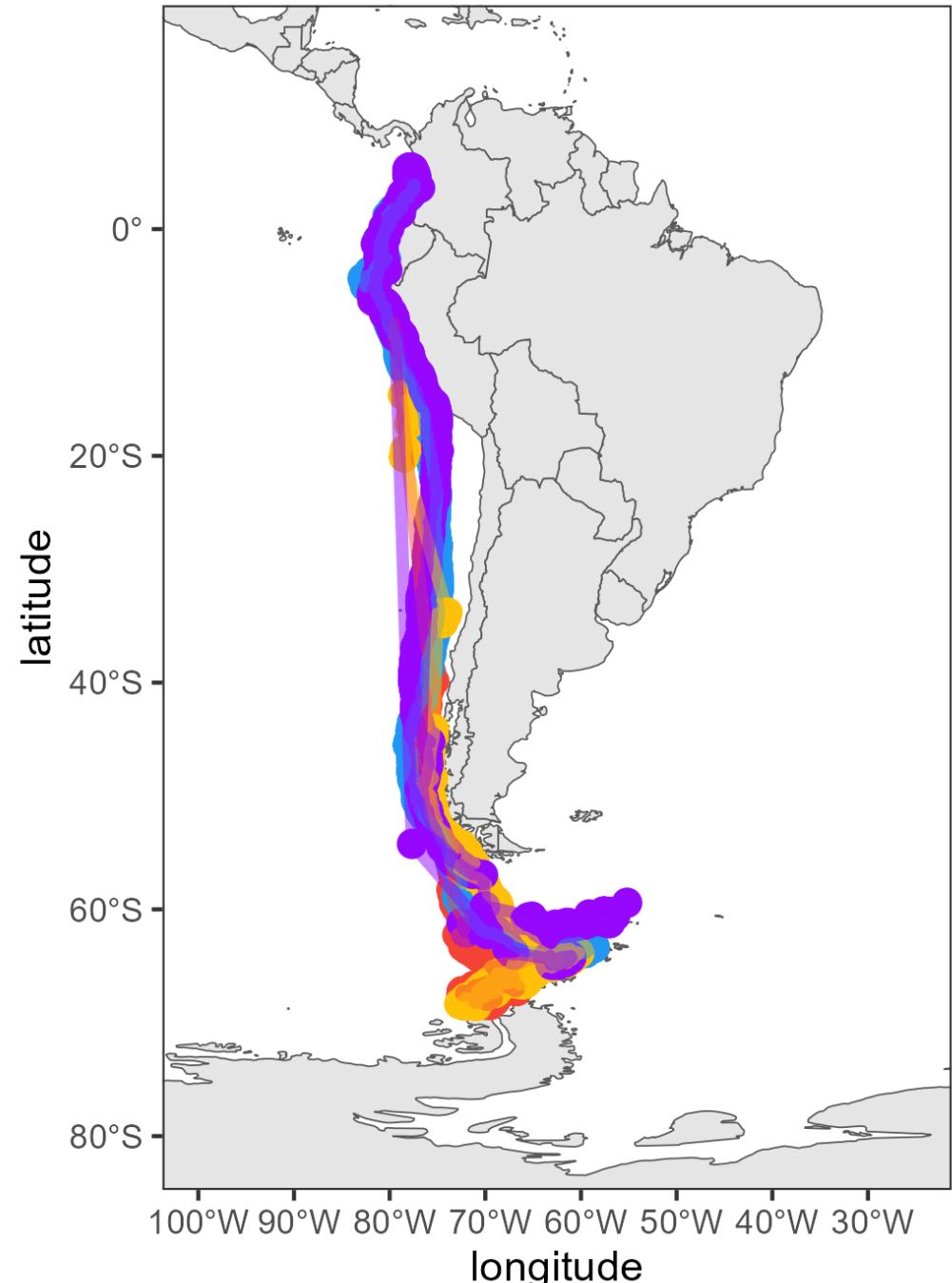


Animated maps

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

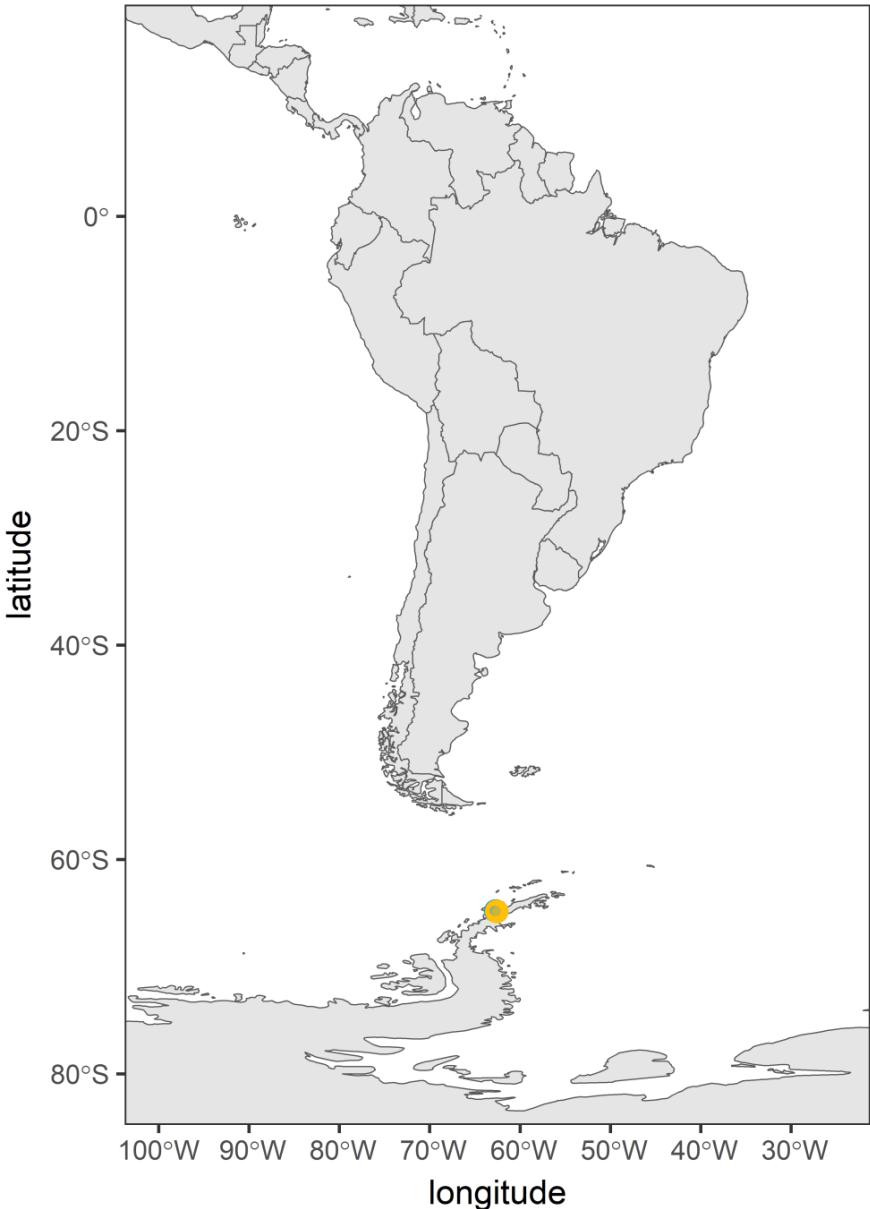
1. Make static map with ggplot()

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)
```

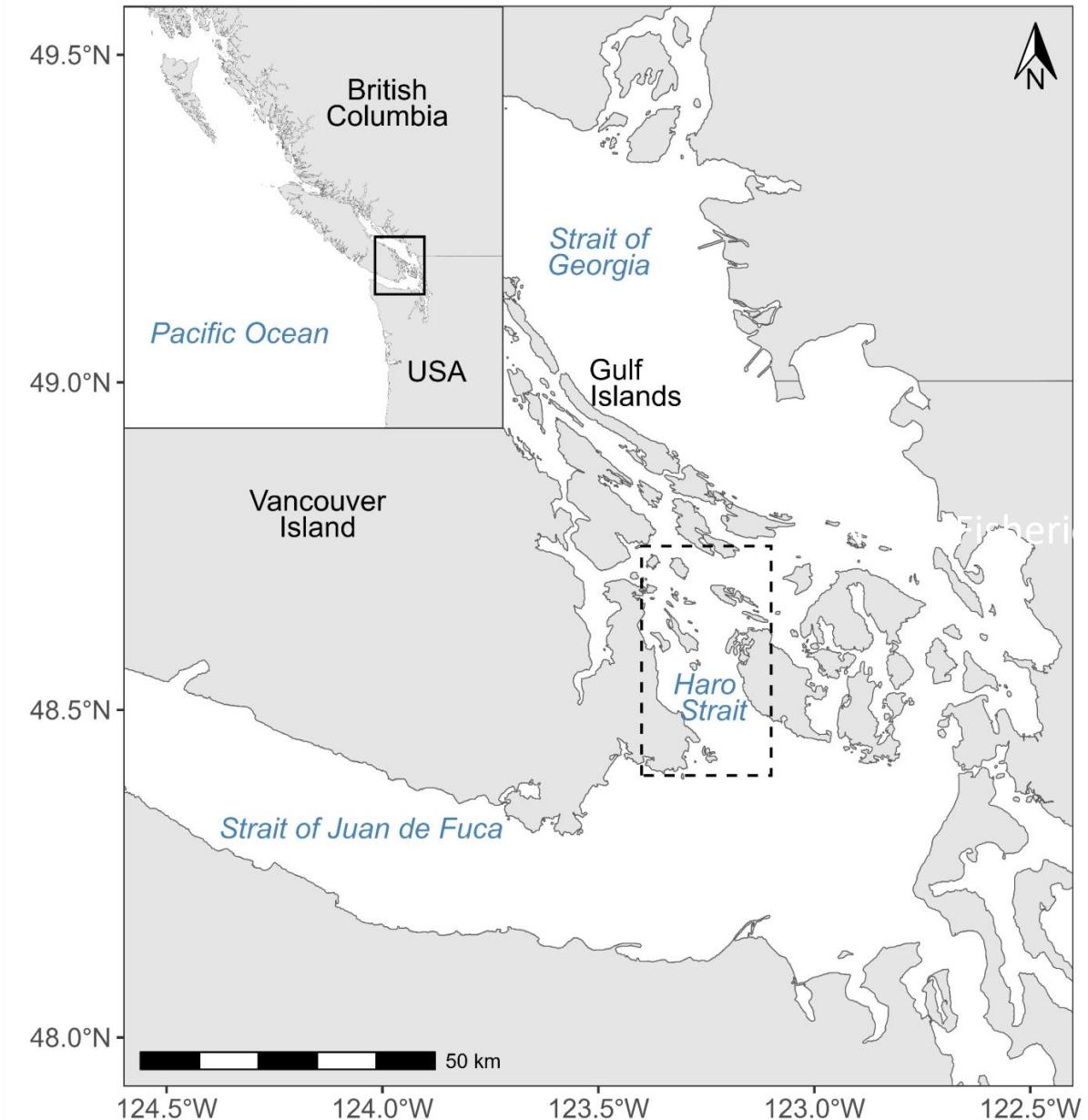
Single point for each whale, with path
following behind the point

Date: 2013-02-05



Key Points

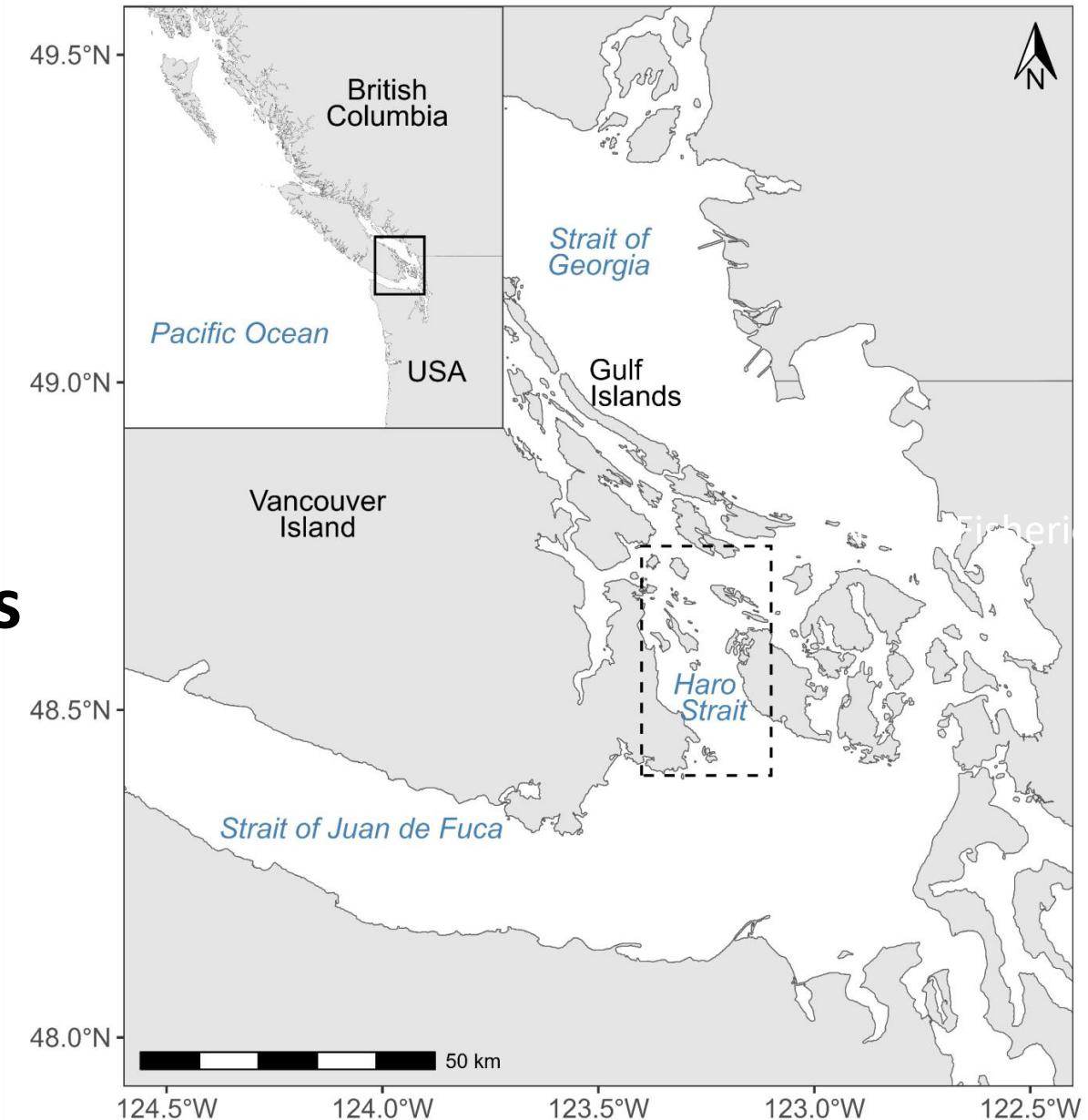
1. Study area maps:
 - a. high resolution
 - b. show what's important
 - c. clean



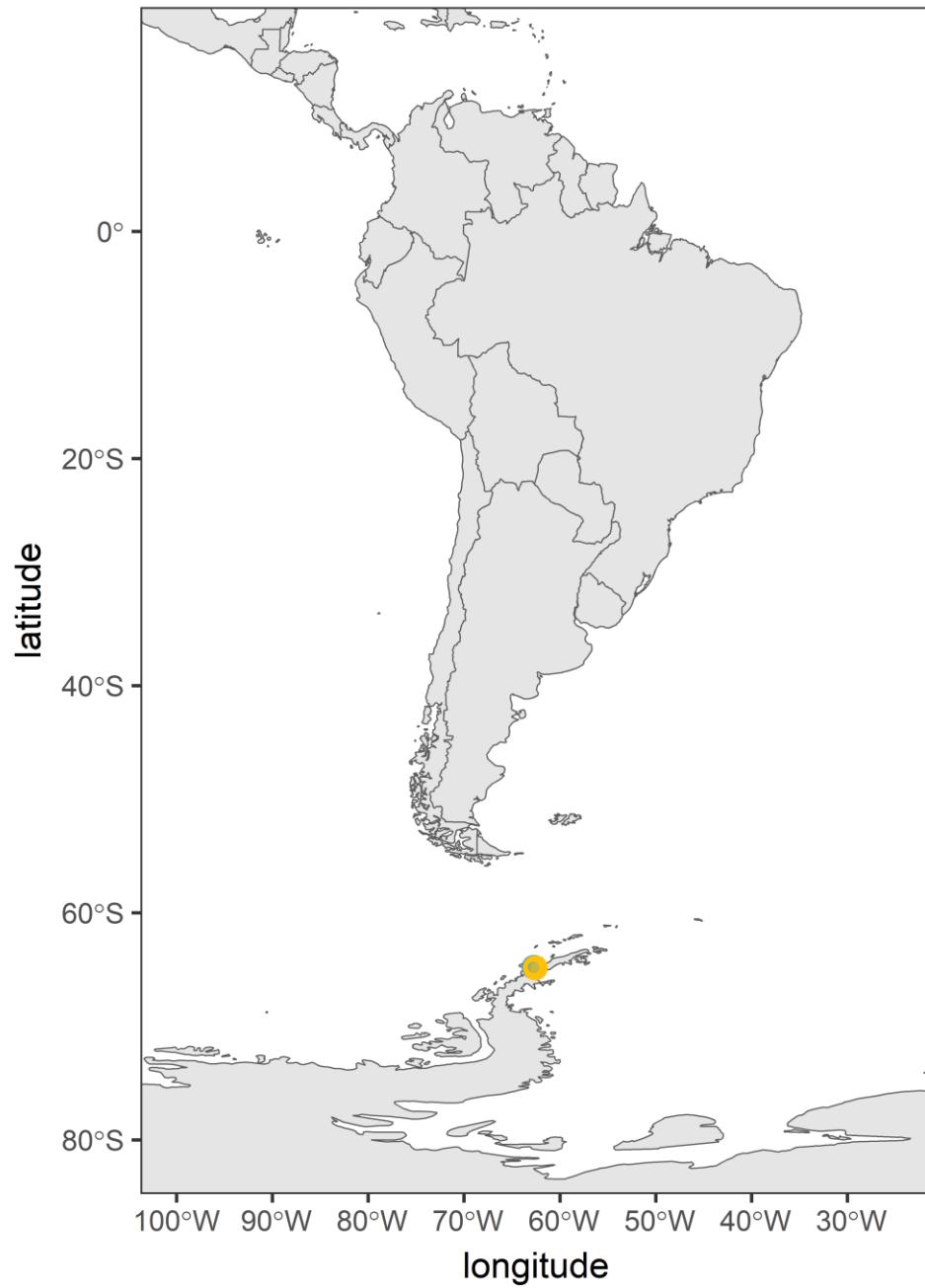
Key Points

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

CSEE Organizing Committee

Canadian Institute of Ecology and Evolution

Francis Juanes, Will Duguid, and Juanes/Baum labmates



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!

Enjoy the rest of CSEE!



Let's make our own maps!

Questions/comments:

wgreentree@outlook.com

