GBRCD Example R Code for CSV version

Ariella Arzey

2024-02-05

R Markdown document with example code for handling GBRCD CSV version

This document contains examples for how users can process, subset and visualise the GBRCD. The example demonstrates filtering the data by proxy, resolution, age and location. The example demonstrates how to produce two types of figures; line plots for visualising the record data and maps for visualising locations.

This document also provides example code for how to produce each of the area charts included in the GBRCD publication.

```r
## Load required packages ##

# Tidyverse includes dplyr (for data frame manipulation),
# lubridate (for handling dates) & ggplot2 (for plotting)
library(tidyverse)
# sf for mapping objects
library(sf)
# ozmaps for Queensland coastline for mapping
library(ozmaps)

# Get Queensland outline from ozmap
qldmap <- ozmaps::ozmap_states %>% filter(NAME == "Queensland")
```

Load metadata and database files

v1.0 is the current version as of February 2024

```r
# Read GBRCD metadata file
metaD <- read.csv("GBRCD_metadata_v1.0.csv")

# Load GBRCD datasets

# Get list of GBRCD file names
data_files_list <- list.files(path = "./GBRCD_files_v1.0/", pattern = ".csv")
# Load list of GBRCD files
data_files <- lapply(data_files_list,
                     function(xx) read.csv(paste0(".\\GBRCD_files_v1.0/", xx)))
# Assign name to all loaded data files
names(data_files) <- gsub("\\.csv$", "", data_files_list)
```

Show metadata

Check top and bottom 5 rows of the GBRCD_metadata. For better visualisation, only the first 4 columns are viewed here.

```
# Use to view first 5 rows of metadata
head(metaD[,1:4], 5)
```

```
##   cdata_datasetID cdata_coreName cdata_altCoreName cdata_collectTime
## 1     AL03DAV01_1       Davies 2     Davies 2 side           1993-10
## 2     AL03DAV01_2       Davies 2     Davies 2 side           1993-10
## 3     AL03PAN01_1       PAN 98-2       PAN 98-2 B3           1998-10
## 4     AL03PAN01_2       PAN 98-2       PAN 98-2 B3           1998-10
## 5     AL03PAN01_3       PAN 98-2       PAN 98-2 B1           1998-10
```

```
# Use to view last 5 rows of metadata
tail(metaD[,1:4], 5)
```

```
##     cdata_datasetID cdata_coreName cdata_altCoreName cdata_collectTime
## 204      WU21MAS01a         MAS02A              <NA>           2017-08
## 205      WU21MAS01b         MAS01E              <NA>           2017-08
## 206       WU21SHW01         SHW82C              <NA>              <NA>
## 207       WU21SMI01         SMI81A              <NA>           2018-02
## 208       XI20ARL01          10AR2              <NA>           2010-04
```

Join the data with the metadata Filter datasets by proxies, resolution, etc.

Attach ID and relevant metadata fields to datasets (left_join) and filter by properties

```
# Add dataset ID column to files
data_ID <- Map(cbind, data_files, cdata_datasetID=names(data_files))


# Join selection of metadata to datasets, add column for region, and translate dates
data_metaD <-
  lapply(data_ID, function (df) df %>%
    # Join (by ID) data files with selected metadata columns e.g.
    # latitude, longitude, site, anomaly flag, etc.
           left_join(.,
            metaD %>% dplyr::select(cdata_datasetID, geo_latitude, geo_longitude,
                        geo_siteName, meths_isAnomaly, meths_primaryVariablesList,
                        meths_hasResolutionNominal, meths_resolutionMedian),
                        by = "cdata_datasetID") %>%
# Add column grouping latitudes into user defined regions
           mutate(Region = ifelse(geo_latitude > -17, "North",
                                  ifelse( geo_latitude < -20, "South",
                                          "Central"))) %>%
# Add columns to convert decimal dates to dates, month and year
  # Note: lubridate date_decimal assumes astronomical numbering so conversion
  # unnecessarily subtracts an additional year from BCE dates.
  # Exceptions to this are whole year integers that lubridate translates to 01/01/[BCE YEAR]
           mutate(MONTH = month(date_decimal(Age)), Year = year(date_decimal(Age))) %>%
           mutate(Year = ifelse(grepl("^LE05", cdata_datasetID) & Year < 0, Year,
                                 ifelse(grepl("^LO14", cdata_datasetID) & Year < 0, Year,
                                       ifelse(grepl("^LE16", cdata_datasetID) & Year < 0, Year,
                                 ifelse(Year < 0, Year + 1, Year))))))
```

Show merged data for a single record

```
# Use to view first 5 rows of named data frame
head(data_metaD[['AL03DAV01_1']][,1:6], 5)
```

```
##        Age    SrCa Distance cdata_datasetID geo_latitude geo_longitude
## 1 1989.497 9.09220    53.75     AL03DAV01_1        -18.8         147.7
## 2 1989.525 9.10272    53.50     AL03DAV01_1        -18.8         147.7
## 3 1989.549 9.12891    53.25     AL03DAV01_1        -18.8         147.7
## 4 1989.577 9.11528    53.00     AL03DAV01_1        -18.8         147.7
## 5 1989.604 9.11299    52.75     AL03DAV01_1        -18.8         147.7
```

```
# Use to view last 5 rows of named data frame
tail(data_metaD[['AL03DAV01_1']][,1:6], 5)
```

```
##          Age    SrCa Distance cdata_datasetID geo_latitude geo_longitude
## 211 1993.700 9.00660     1.25     AL03DAV01_1        -18.8         147.7
## 212 1993.722 9.00262     1.00     AL03DAV01_1        -18.8         147.7
## 213 1993.747 8.97722     0.75     AL03DAV01_1        -18.8         147.7
## 214 1993.768 8.99205     0.50     AL03DAV01_1        -18.8         147.7
## 215 1993.793 9.01461     0.25     AL03DAV01_1        -18.8         147.7
```

Filter datasets by proxies, resolution, etc.

An example for how to subset/filter by properties.

Ba/Ca is used as an example variable to filter the GBRCD

Suggested fields for filtering:

- Record coverage (note this is number of years of data and accounts for gaps):
  - cdata_dataCoverageGroup (1 = >100 years, 2 = 10-100 years & 3 = <10 years of data)
- Proxy Type:
  - meths_primaryVariablesList (e.g. BaCa, d11B, d18O, SrCa)
  - meths_additionalVariablesList (e.g. d18Osw, d11B_ph)
- Temporal Coverage (note this is total temporal span of records):
  - cdata_minYear (record start year)
  - cdata_maxYear (record end year)
- Record Resolution:
  - meths_hasResolutionNominal (nominal resolution)

  - meths_resolutionMax (maximum resolution; data points per year)
  - meths_resolutionMin (minimum resolution; data points per year)
  - meths_resolutionMean (mean resolution; data points per year)
  - meths_ResolutionMedian (median resolution; data points per year)

- Location:

  - geo_latitude (record latitude; degrees N (all GBR latitudes are negative))
  - geo_longitude (record longitude; degrees E (all GBR longitudes are positive))
  - geo_siteName (name of the site/reef)

- Species: cdata_archiveSpecies

- Record Method:

  - meths_[x]Method (method used for trace element measurement; [x] should be replaced by choice of data i.e TE = trace element, isotope, lumin = luminescence)

- SST Calibration:

  - calib_isSSTCalibration (record is SST calibration dataset (SrCa, UCa, d18O); T/F)
  - calib_useSSTCalibration (record uses SST calibration (SrCa, UCa, d18O); T/F)

** All metadata fields may be be used for filtering, but the above list above includes the suggested starting point for investigating the data.

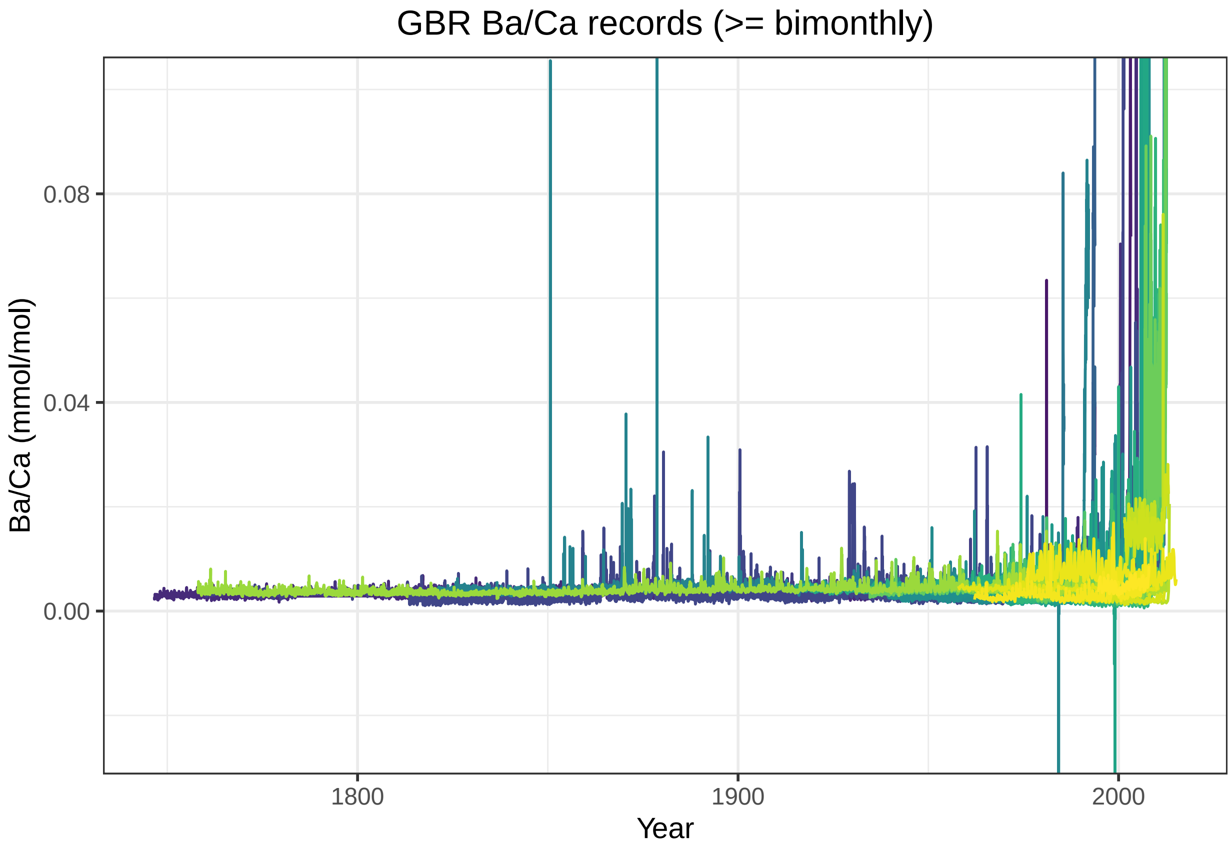
```
# Filter selection of datasets by metadata values -
# e.g. only datasets that include Ba/Ca (BaCa)
# since the 1500s and are bimonthly or higher resolution and excluding anomaly data
data_metaDFILT <- lapply(data_metaD, function (df) df %>%
  # Filter for resolution by median number of values per year
                     filter(meths_resolutionMedian >=6) %>%
  # Filter out records that are anomaly data
                     filter(meths_isAnomaly != TRUE) %>%
  # Filter for ages after 1700
                     filter(Age > 1700) %>%
  # Set up filter column for variable of choice (e.g. BaCa) and drop NA values
                     mutate(BACA_yes = ifelse("BaCa" %in% colnames(df), "Yes", NA)) %>%
                     drop_na(BACA_yes) %>%
                     select(-BACA_yes))

# Add below lines to filter a selection of duplicate records
# (e.g. shorter or lower resolution records removed)
        # filter(!cdata_datasetID %in% c("FA03MYR01","TH22DAV01","TH22DAV02")))

# Keep only data frames with Ba/Ca
baca_data <- keep(data_metaDFILT, ~ nrow(.x) > 0)

# Create a single data frame from list of data frames
baca_data_DF <- bind_rows(baca_data)
```

Plot GBRCD Ba/Ca

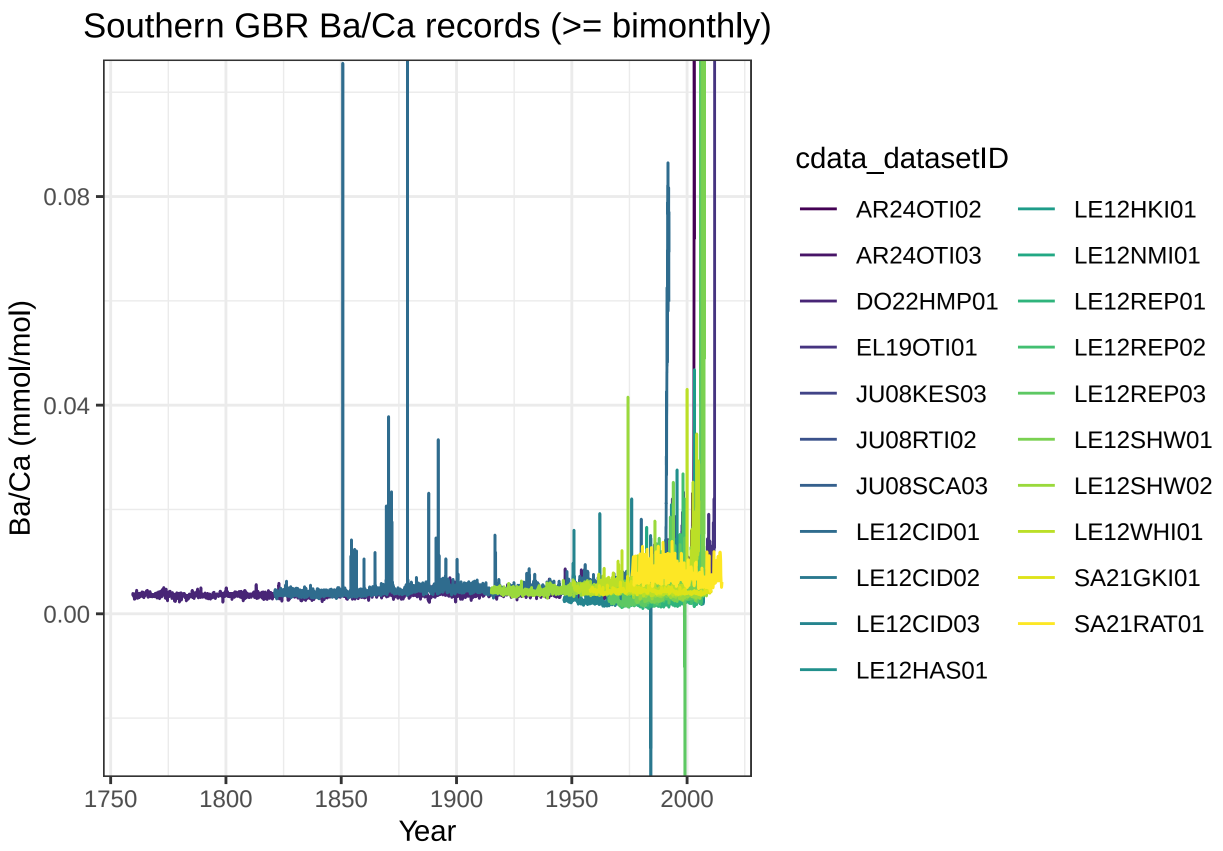
Plotting all ‘modern’ bimonthly or higher resolution Ba/Ca data (entire GBR, southern GBR and southern GBR south of -22.5)

For ease of viewing, Ba/Ca plots y-axis limits are restricted.

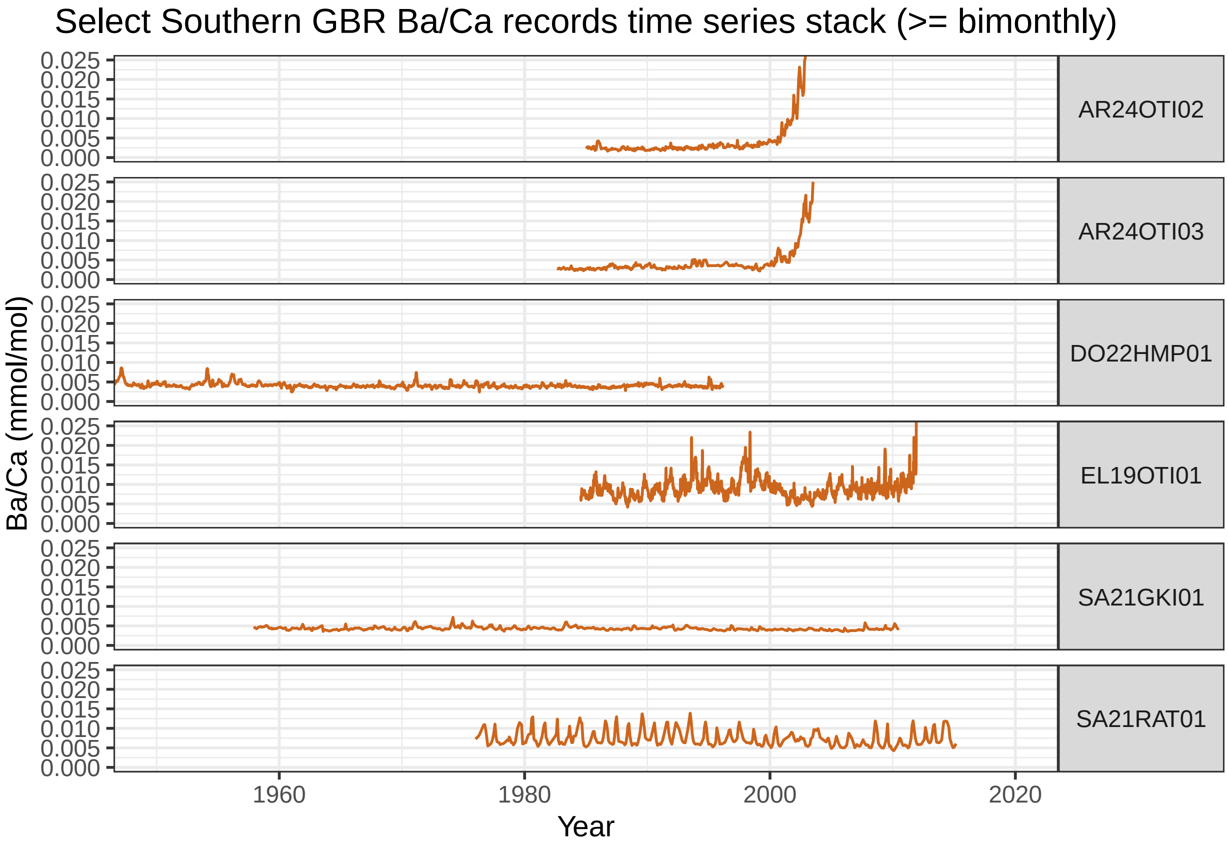
```
# Plot all Ba/Ca records for entire GBR
ggplot(baca_data_DF %>%
  # Drop rows with no BaCa value
         drop_na(BaCa),
         aes(x = Age, y = BaCa, colour = cdata_datasetID))+
  geom_line()+
  # Restrict y axis for BaCa from -0.025 to 0.1
  coord_cartesian(ylim=c(-0.025, 0.1))+
  # Set colour theme used for records
  scale_colour_viridis_d()+
  # Set plot, y axis and x axis titles
  ggtitle("GBR Ba/Ca records (>= bimonthly)")+
  ylab("Ba/Ca (mmol/mol)")+
  xlab("Year")+
  theme_bw()+
  # Suppress legend due to number of records and centre align title
  theme(legend.position="none", plot.title = element_text(hjust = 0.5))
```



```
# Plot Ba/Ca records for only the southern GBR
ggplot(baca_data_DF %>%
  # Drop rows with no BaCa value
         drop_na(BaCa) %>% filter(Region == "South"),
       aes(x = Age, y = BaCa, colour = cdata_datasetID))+
  geom_line()+
  # Restrict y axis for BaCa from -0.025 to 0.1
  coord_cartesian(ylim=c(-0.025, 0.1))+
  # Set colour theme used for records
  scale_colour_viridis_d()+
  # Set plot, y axis and x axis titles
  ggtitle("Southern GBR Ba/Ca records (>= bimonthly)")+
  ylab("Ba/Ca (mmol/mol)")+
  xlab("Year")+
  theme_bw()+
  # Centre align title
  theme(plot.title = element_text(hjust = 0.5))
```



```
# Plot only southern GBR and records latitudes south of -22.5
ggplot(baca_data_DF %>%
  # Drop rows with no BaCa value
         drop_na(BaCa) %>%
  # Filter by region
  filter(Region == "South") %>%
  # Filter by latitude
         filter(geo_latitude < -22.5),
       aes(x = Age, y = BaCa))+
  # Plot all BaCa records as the same colour
  geom_line(colour = "#CD661D")+
  # Restrict y axis for BaCa from 0 to 0.025 and x axis ages to after 1950
  coord_cartesian(ylim=c(0, 0.025), xlim=c(1950, 2020))+
  # Set plot, y axis and x axis titles
  ggtitle("Select Southern GBR Ba/Ca records time series stack (>= bimonthly)")+
  ylab("Ba/Ca (mmol/mol)")+
  xlab("Year")+
  # Create record time series stack by ID (alphabetical order)
  facet_grid("cdata_datasetID")+
  theme_bw()+
  # Centre align title  & set horizontal direction for facet label
  theme(plot.title = element_text(hjust = 0.5), strip.text.y = element_text(angle = 0))
```



Create GBRCD outputs for mapping variables

Create maps of variables of interest e.g. Ba/Ca

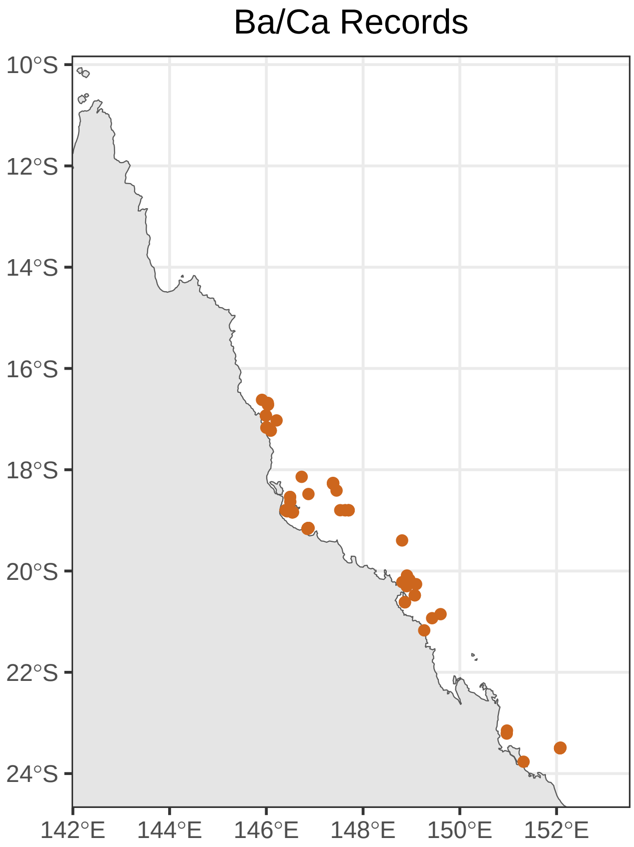
```
# Filter metadata for BaCa record
metaD_BaCa <- metaD %>% filter(grepl("BaCa", meths_primaryVariablesList))


# Filter metadata for BaCa record and group by location (geo_siteName)
metaD_BaCaGRP <- metaD_BaCa %>% group_by(geo_siteName) %>%
  summarise(lat = mean(geo_latitude), long = mean(geo_longitude), counts = n())
```

```
# Convert all BaCa and grouped BaCa metadata into map object
metaBaCa_sf <- metaD_BaCa %>% st_as_sf(coords = c("geo_longitude", "geo_latitude"),
                              crs=4283, #EPSG: 4283 - GDA94
                              remove=FALSE)

metaBaCa_sfGRP <- metaD_BaCaGRP %>% st_as_sf(coords = c("long", "lat"),
                                       crs=4283, #EPSG: 4283 - GDA94
                                       remove=FALSE)
```

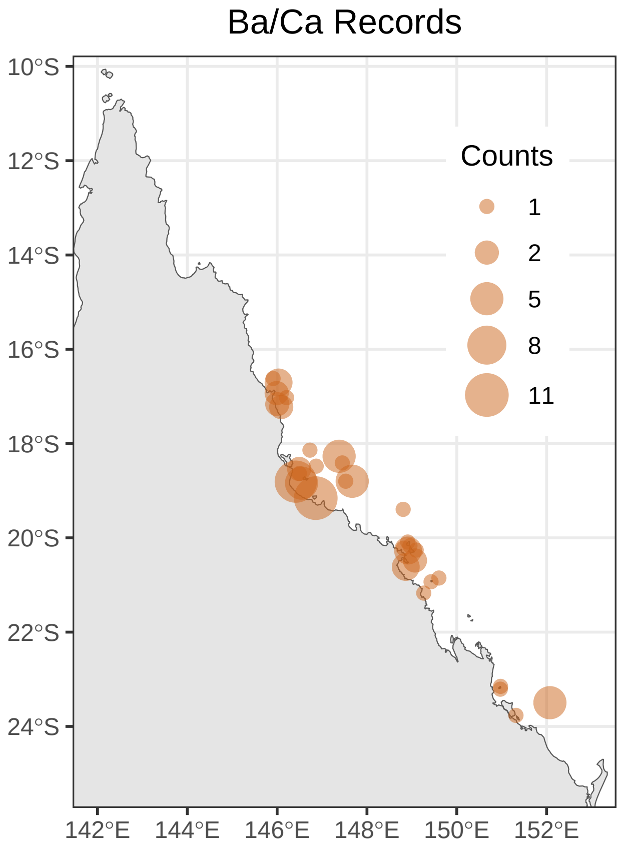
GBR Database Maps of Ba/Ca

Create map of Ba/Ca data, plotting locations for every record and plotting point sizes relative to number of records per site.

```
# Plot all BaCa points
ggplot() +
  # Plot QLD outline
  geom_sf(data = qldmap) +
  # Plot record locations
  geom_sf(data = metaBaCa_sf, colour = "#CD661D")+
  coord_sf()+
  # Set limits on x axis (longitude) and y axis (latitude)
  ylim(-24, -10.5)+
  xlim(142.5, 153)+
  ggtitle("Ba/Ca Records")+
  # Set up legend of mapped records
  labs(colour = "Legend")+
  guides(colour = "none")+
  theme_bw()+
  # Set legend position and plot title alignment
  theme(legend.position = c(0.8, 0.9), plot.title = element_text(hjust = 0.5))
```



```
# Plot all BaCa records grouped by location (geo_siteName) (size = number per location)
ggplot() +
  # Plot QLD outline
  geom_sf(data = qldmap) +
  # Plot grouped record locations
  geom_sf(data = metaBaCa_sfGRP, aes(size = counts), colour = "#CD661D", alpha = 0.5)+
  coord_sf()+
  # Set limits on x axis (longitude) and y axis (latitude)
  ylim(-25, -10.5)+
  xlim(142, 153)+
  ggtitle("Ba/Ca Records")+
  # Set up legend and set the scale/size of mapped records
  labs(colour = "Legend")+
  scale_size(name = "Counts", range = c(2, 7), breaks = c(1, 2, 5, 8, 11))+
  guides(colour = "none")+
  theme_bw()+
  # Set legend position and plot title alignment
  theme(legend.position = c(0.8, 0.7), plot.title = element_text(hjust = 0.5))
```



GBRCD - Configuring the data for area charts

Preparation for creating area charts using ggplot.

Bind all data and join with metadata, then summarise by relevant variables e.g. record coverage (cdata_dataCoverageGroup) or nominal resolution (meths_hasResolutionNominal).

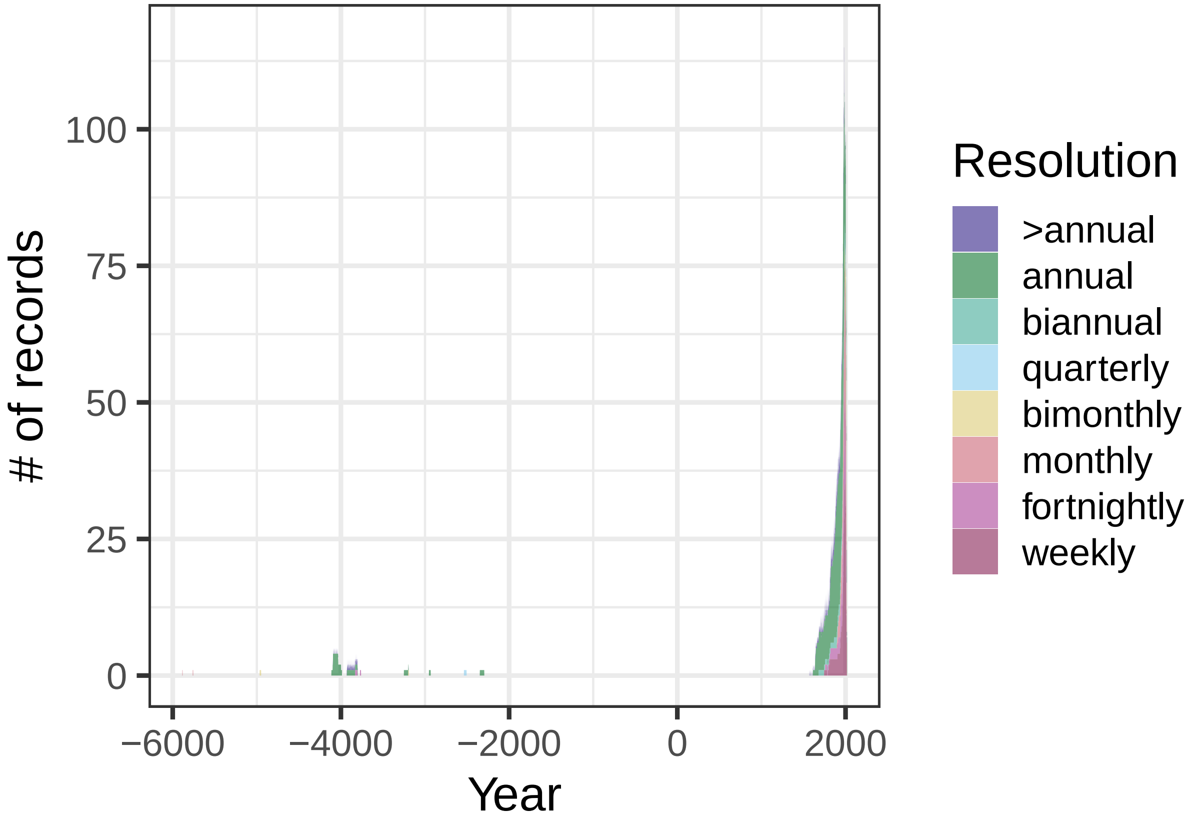
To ensure that gaps are appropriate represented in the area charts and filler data frame is created for every year

```
# Join all data frames by ID
data_all_bind <- bind_rows(data_ID, .id = "cdata_datasetID")

# Sort column order so ID is first
data_all_bind <- data_all_bind[, c(4, 1,3, 2, 5:ncol(data_all_bind))]

# Join all metadata to record data
data_all_bind_join <- data_all_bind %>%
  left_join(metaD, by = c("cdata_datasetID" = "cdata_datasetID"))

# Summarise GBRCD data by coverage group (cdata_dataCoverageGroup) -
# change record coverage values to be more descriptive
# i.e. 1 changed to Group 1 >100 years
gbrcd_cover_stats <- data_all_bind_join %>%
  # Group by variables to summarise on
  group_by(cdata_datasetID, Year = year(date_decimal(Age))) %>%
  # Summarise records - return only 1 value per year
  summarise(records = first(cdata_datasetID),
            lengthgrp = first(cdata_dataCoverageGroup), count = 1) %>%
  # Make coverage values descriptive
  mutate(lengthgrp = ifelse(lengthgrp== 1, "Group1 >100 years",
                            ifelse( lengthgrp == 2, "Group2 10-100 years",
                                    "Group3 <10 years"))) %>%
  ungroup() %>%
  # Summarise on groups for sum of records per coverage group per year
  group_by(Year, lengthgrp) %>%  summarise(records = sum(count)) #%>%

# Summarise GBRCD by nominal resolution (meths_hasResolutionNominal) -
# remove "_uneven" so x and x_uneven are now 1 group
gbrcd_stats_nominal <- data_all_bind_join %>%
  # Group by variables to summarise on
  group_by(cdata_datasetID, Year = year(date_decimal(Age))) %>%
  # Remove '_uneven' from nominal resolution description
  mutate(nominalRes = stringr::str_remove(meths_hasResolutionNominal, "_uneven")) %>%
  # Summarise records - return only 1 value per year
  summarise(records = first(cdata_datasetID), nomRes = first(nominalRes), count = 1) %>%
  ungroup() %>%
  # Summarise on groups for sum of records per resolution group per year
  group_by(Year, nomRes) %>%  summarise(records = sum(count))




#Levels of the resolution and coverage groups
nomlvls <- c(">annual", "annual", "biannual", "quarterly", "bimonthly",
             "monthly", "fortnightly", "weekly")
covlvls <- c("Group1 >100 years", "Group2 10-100 years", "Group3 <10 years")

# Create data frame of every year from -5890 CE to 2017 CE  for
# every group for nominal resolution (nomRes)
stats_seqNOM <- data.frame(Year = rep(seq(-5890, 2017, 1), 8),
                     nomRes = gl(length(nomlvls), k = 7908, labels=nomlvls, ordered=TRUE))


# Create data frame of every year from -5890 CE to 2017 CE  for
# every group for record coverage (lengthgrp)
stats_seqCOV <- data.frame(Year = rep(seq(-5890, 2017, 1), 3),
                     lengthgrp = gl(length(covlvls), k = 7908, labels=covlvls, ordered=TRUE))

# Join data frame of years with GBRCD based on nominal resolution and year value
gbrcd_statsNOM_SEQ <- stats_seqNOM %>%
  left_join(gbrcd_stats_nominal, by = c("Year", "nomRes")) %>% arrange(Year, nomRes)

# Change NA values to 0
gbrcd_statsNOM_SEQ <- gbrcd_statsNOM_SEQ %>%
  mutate(recordsCount = ifelse(is.na(records), 0, records))

# Add factor levels to nomRes to plot from lowest resolution to highest
gbrcd_statsNOM_SEQ$nomRes <- factor(gbrcd_statsNOM_SEQ$nomRes,
                                    levels = c(">annual", "annual", "biannual",
                                               "quarterly", "bimonthly", "monthly",
                                               "fortnightly", "weekly"))

# Join data frame of years with GBRCD based on record coverage and year value
gbrcd_statsCOV_SEQ <- stats_seqCOV %>%
  left_join(gbrcd_cover_stats, by = c("Year", "lengthgrp")) %>%
  arrange(Year, lengthgrp)

# Change NA values to 0
gbrcd_statsCOV_SEQ <- gbrcd_statsCOV_SEQ %>%
  mutate(recordsCount = ifelse(is.na(records), 0, records))
```

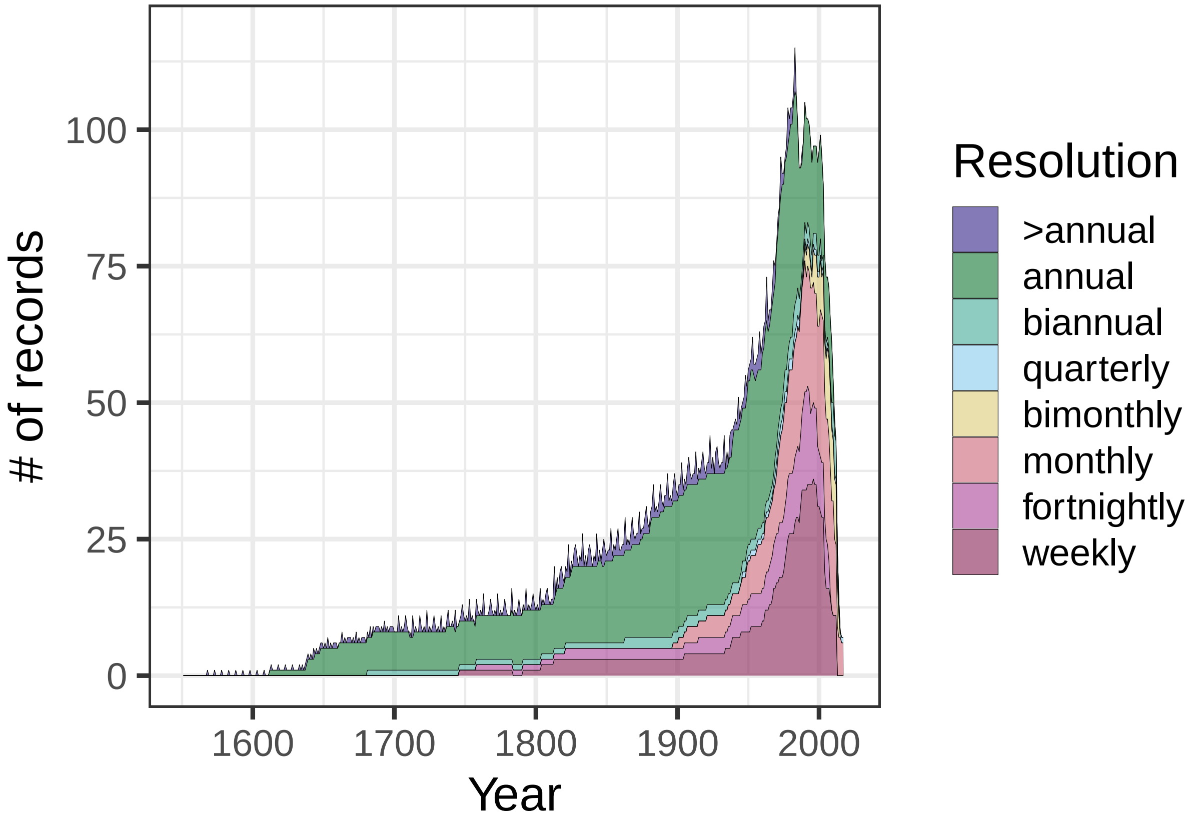
GBRCD - Area Charts for Nominal Resolution

Area chart of all records based on record coverage for the entire GBRCD -5884 to 2017 CE and for records from 1550 to 2017 CE

```
# Selection of colours for area charts
mycolours <- c("#332288", "#117733", "#44AA99", "#88CCEE", "#DDCC77",
                "#CC6677", "#AA4499", "#882255")

# Plot nominal resolution for entire GBRCD period
ggplot(gbrcd_statsNOM_SEQ,
       aes(x=as.numeric(Year), y = recordsCount, fill = as.factor(nomRes)))+
  geom_area(alpha=0.6 , linewidth=.1, stat = "align")+
  # Set limits and breaks for the y axis
  scale_y_continuous(breaks=seq(0, 117, 25), limits=c(0, 117))+
  # Set up the legend values
  scale_fill_manual(name = "Resolution", values = mycolours)+
  # Assign labels to x and y axis
  xlab("Year")+
  ylab("# of records")+
  theme_bw(base_size = 18)
```



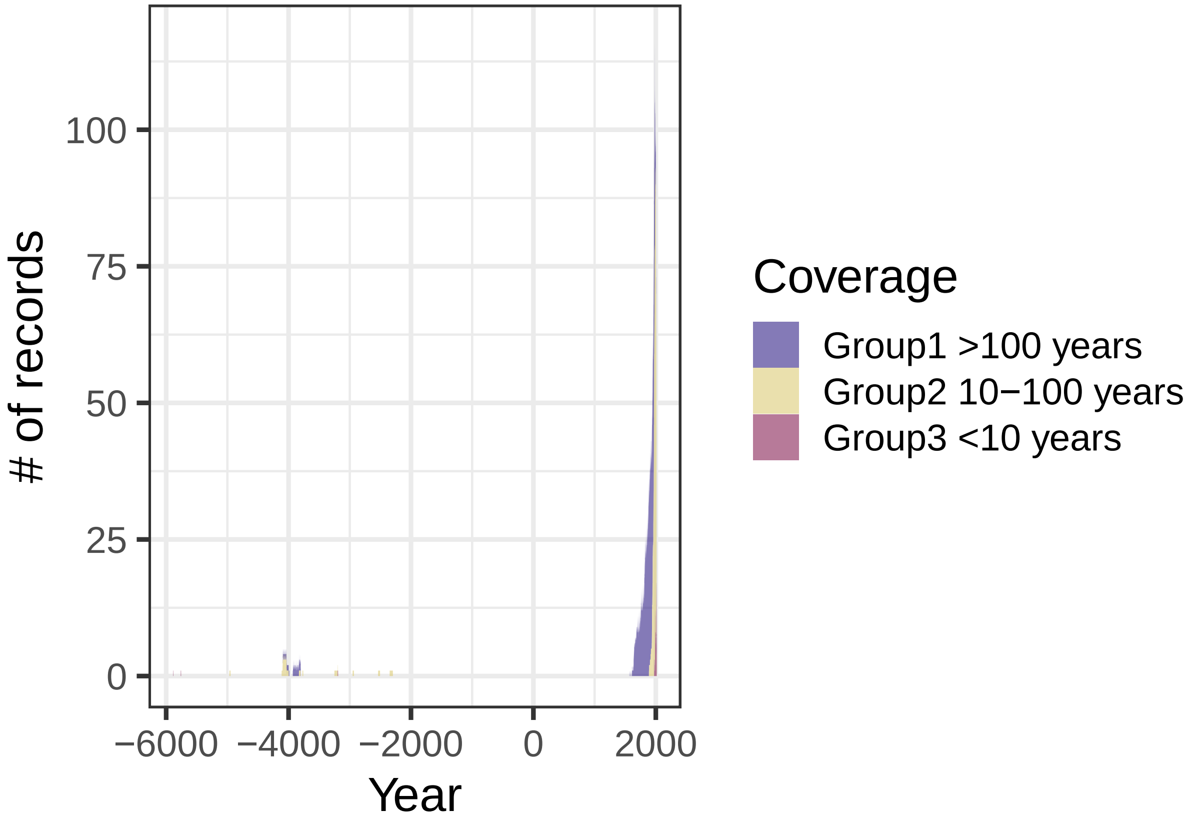
```r
# Plot nominal resolution post-1550
ggplot(gbrcd_statsNOM_SEQ %>% filter(Year > 1550),
       aes(x=as.numeric(Year), y = recordsCount, fill = as.factor(nomRes)))+
  geom_area(alpha=0.6 , linewidth=.1, colour="black", stat = "align")+
  # Set limits and breaks for the x and y axis
  scale_y_continuous(breaks=seq(0, 117, 25), limits=c(0, 117))+
  scale_x_continuous(breaks=seq(1500, 2020, 100), limits=c(1550, 2020))+
  # Set up the legend values
  scale_fill_manual(name = "Resolution", values = mycolours)+
  # Assign labels to x and y axis
  xlab("Year")+
  ylab("# of records")+
  theme_bw(base_size = 18)
```



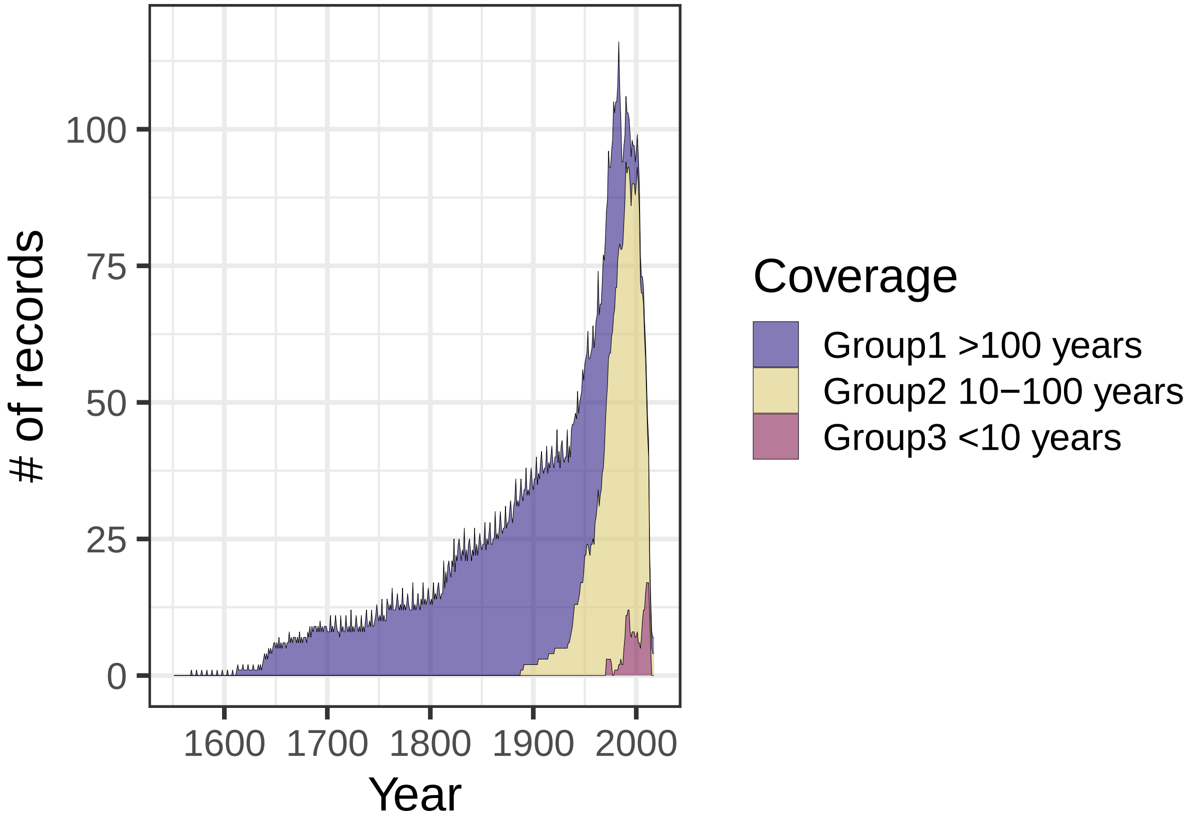
GBRCD - Area Charts for Group Coverage

Area chart of all records based on record coverage for the entire GBRCD -5884 to 2017 CE and for records from 1550 to 2017 CE

```
# Plot record coverage for entire GBRCD period
ggplot(gbrcd_statsCOV_SEQ , aes(x=Year, y = recordsCount, fill = as.factor(lengthgrp)))+
  geom_area(alpha=0.6 , linewidth=.01, stat = "align")+
  # Set limits and breaks for the y axis
  scale_y_continuous(breaks=seq(0, 117, 25), limits=c(0, 117))+
  # Set up the legend values
  scale_fill_manual(name = "Coverage",
                    values = c("Group1 >100 years" = mycolours[1],
                               "Group2 10-100 years" = mycolours[5],
                               "Group3 <10 years" = mycolours[8]),
                    aesthetics = c("color", "fill")) +
  # Assign labels to x and y axis
  xlab("Year")+
  ylab("# of records")+
  theme_bw(base_size = 18)
```



```
# Plot record coverage post-1550
ggplot(gbrcd_statsCOV_SEQ %>% filter(Year > 1550),
       aes(x=Year, y = recordsCount, fill = as.factor(lengthgrp)))+
  geom_area(alpha=0.6 , linewidth=.1, colour="black", stat = "align")+
  # Set limits and breaks for the x and y axis
  scale_x_continuous(breaks=seq(1500, 2020, 100), limits=c(1550, 2020))+
  scale_y_continuous(breaks=seq(0, 117, 25), limits=c(0, 117))+
  # Set up the legend values
  scale_fill_manual(name = "Coverage",
                    values = c("Group1 >100 years" = mycolours[1],
                               "Group2 10-100 years" = mycolours[5],
                               "Group3 <10 years" = mycolours[8]),
                    aesthetics = c("color", "fill")) +
  # Assign labels to x and y axis
  xlab("Year")+
  ylab("# of records")+
  theme_bw(base_size = 18)
```



# of records
0
25
50
75
100
1600
1700
1800
1900
2000
Year
Coverage
Group1 >100 years
Group2 10−100 years
Group3 <10 years