NOAA Technical Memorandum NMFS-XXX-##

GAP Production Data Documentation

Emily Markowitz, Zack Oyafuso, Sarah Friedman

U.S. DEPARTMENT OF COMMERCE

National Oceanic and Atmospheric Administration
National Marine Fisheries Service
Northwest Fisheries Science Center

GAP Production Data Documentation

Emily Markowitz[1,*], Zack Oyafuso[2,*] and Sarah Friedman[2,*]

1. NOAA Fisheres Alaska Fisheries Science Center, Groundfish Assessment Program, Bering Sea Survey Team
2. NOAA Fisheries Alaska Fisheries Science Center, Groundfish Assessment Program, Gulf of Alaska and Aleutian Island Survey Team

* *Correspondence:* Emily Markowitz emily.markowitz@noaa.gov * *Correspondence:* Zack Oyafuso zack.oyafuso@noaa.gov * *Correspondence:* Sarah Friedman sarah.friedman@noaa.gov

Table of contents

List of Figures

List of Tables

1. Welcome

> Please consider this resource to be a **Living Document**. The code in this repository is regularly being updated and improved. Please refer to releases for finalized products and project milestones.

1.1. Background of the gap_products repo

This work is the result of the massive efforts of three concurrent GAP working groups:

1) Index Computation Working Group: consolidation of the methods used to produced design-based estimates of abundance and size/age composition between the Bering Sea and AIGOA survey regions.

2) Data Processes Working Group: reorganization of the Oracle data infrastructure that houses the standard data products produced by GAP.

3) Gulf of Alaska Survey Restratification Working Group: implementation of a new stratified random survey design in the Gulf of Alaska bottom trawl survey.

We began this effort in collaboration with the Status of Stocks team (SSMA) to present both the orientation and opportunity to interact with Gulf of Alaska data from the restratified survey design that we will be implementing in the 2025 field season. As that part of the project evolved, the Data Processes Working Group identified the opportunity and need for gaining efficiencies by redesigning and consolidating the Oracle objects (tables and materialized views) that have historically served these data. The Index Computation Working Group also identified an opportunity to gain efficiencies by consolidating the various scripts that were developed independently by both survey region groups into a workflow that was more accessible and documented.

The Index Computation Working Group developed the gapindex R package, a code repository that consolidates the code that calculates the various standard GAP products (e.g., CPUE, total biomass, size/age composition) for both the Bering Sea and AIGOA survey regions. The Data Processes Working Group was responsible for compiling the data structures needed to support data product tables that were consistent across all of the AFSC GAP survey regions as well as the creation of the GAP_PRODUCTS oracle schema that will house these consolidated products in the future.

This gap_products GitHub repository houses the code that will conduct the "standard production run" that produces the new data tables via the gapindex R package and upload those tables to the GAP_PRODUCTS Oracle schema.

1.2. Major Advantages

- Consolidated production tables include all standard data products for all surveys. Data will be provided in the same format, with the same units, and created using the same mathematical methodology. This should limit data pulls, reduce complexity for data access, and reduce complicated secondary data wrangling.
- Consistent naming conventions for schemata, tables, and column metadata. Columns across all tables will use the same naming conventions, units, and data types. Restricting standard data product table content to absolutely necessary columns.
- Removal of redundant data columns that can be acquired by joining to reference tables is key for providing consistent and up-to-date data while limiting data table sizes.
- Consolidation and repurposing of Oracle schemata. This will help the GAP team limit unnecessary access to unprocessed or problematic data by outside users.
- Vetted data methods. All code and data inclusion decisions and wrangling are documented in the {gapindex} R package. Streamlined and rapid data production. Improved and consolidated data creation and documentation provide data creators and users with greater confidence in the data products and enhanced ability to share the data.

1.3. GOA 2025 Restratification – Mock Data for Testing

Additionally, the inclusion of mock data for the under the new 2025 GOA stratified random survey (labeled in the GAP_PRODUCTS tables as YEAR 2025) will provide stock authors with the opportunity to interact with data from the new survey design to be implemented in 2025. In 2023, we will be populating both the historical tables as we have been AND the new GAP_PRODUCTS tables with this summer's survey data. The plan will be, once all are satisfied with the new GAP_PRODUCTS schema and tables, to sunset the historic product tables in 2024 and proceed with only GAP_PRODUCTS for the 2024 post-survey stock assessment season.

1.4. Progress Timeline

- October 2022: The data processes and index computation working group convened to address the development of standard survey data products (e.g., biomass/abundance, size composition, age composition, CPUE).
  - Index Computation Working Group: consolidation of index computation methods between the Bering Sea and AI-GOA regions.
  - Data Processes Working Group: consolidation, clean up, and reorganization of survey oracle schemata, tables, and other data for all surveys.
- December 2022: GAP and SSMA discuss integration of the restratification of the GOA survey design into standard data products.
  - Stock assessors requested a ”dry run” test to work with new mock restratified GOA survey data before implementation of the new survey design.
  - This prompted the postponement of the restratified GOA design to 2025.
- February 2023: Decision was made to include the mock restratified GOA data with the development of the new consolidated standard data products..
- May 2023: Release of new, draft, standard data product tables, including restratified GOA data. Stock assessment authors will have the opportunity to explore differences between datasets, test workflows, and provide comments and issues during summer 2023.
- September 2023: Release of provisional standard data product tables and historical tables containing 2023 survey data in the GAP_PRODUCTS schema.
- December 2023 - March 2024: Meeting between GAP and stock assessment groups in early December 2023 to update progress on the GAP_PRODUCTS testing phase. **Deadline for Comments and Feedback on GAP_PRODUCTS data structures is March 8, 2024.**

September 2024: GAP will only release data products according to the new standard. Current, historical data product tables will be archived in a new schema called ”**GAP_ARCHIVE**”.

1.5. What is the research objective?

The objectives of these surveys are to:

- monitor trends in the marine ecosystem of the Bering Sea, Aleutian Islands, and Gulf of Alaska
- produce fishery-independent biomass and abundance estimates for commercially important fish and crab species
- collect biological and environmental data for use in ecosystem-based fishery management.

Learn more about the program

Figure 1.1.: Sorting and weighing fish on deck on the 2022 Bering Sea groundfish survey aboard the F/V Alaska Knight. Credit: Emily Markowitz/NOAA Fisheries.

Part I.

Introduction

Our Objective

As part of our commitment to open science and transparency, we provide this interactive metadata guide to compliment our public-domain data. Please refer to our Draft Data Changes Brief. Once finalized, this language will be included here.

User Resources

- Groundfish Assessment Program Bottom Trawl Surveys
- AFSC's Resource Assessment and Conservation Engineering Division
- Survey code books
- Publications and Data Reports
- Research Surveys conducted at AFSC

Cite this data

Use the below bibtext citations, as cited in our group's citation repository for citing the data created and maintained in this repo. Add "note = {Accessed: mm/dd/yyyy}" to append the day this data was accessed. Included here are AFSC RACE Groundfish and Shellfish Assessment Program's:

- Design-Based Production Data (internal) (NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program, 2023).

- AFSC RACE Groundfish Data for AKFIN (Alaska Fisheries Information Network (AKFIN), 2023).
- Public Data hosted on the Fisheries One Stop Shop (FOSS) Data Platform (NOAA Fisheries Alaska Fisheries Science Center, 2023).

```
@misc{GAPProducts,
  author = {{NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program}}
  year = {2023},
  title = {AFSC Goundfish Assessment Program Design-Based Production Data},
  howpublished = {https://www.fisheries.noaa.gov/alaska/science-data/groundfish-assessment-
  publisher = {{U.S. Dep. Commer.}},
  copyright = {Public Domain}
}
```

```
@misc{FOSSAFSCData,
  author = {{NOAA Fisheries Alaska Fisheries Science Center}},
  year = {2023},
  title = {Fisheries One Stop Shop Public Data: RACE Division Bottom Trawl Survey Data Quer
  howpublished = {https://www.fisheries.noaa.gov/foss},
  publisher = {{U.S. Dep. Commer.}},
  copyright = {Public Domain}
}

@misc{GAPakfin,
  author = {{Alaska Fisheries Information Network (AKFIN)}},
  institution = {{NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Prog
  year = {2023},
  title = {AFSC Goundfish Assessment Program Design-Based Production Data},
  howpublished = {https://www.psmfc.org/program/alaska-fisheries-information-network-akfin}
  publisher = {{U.S. Dep. Commer.}},
  copyright = {Public Domain}
}
```

Or cite our latest data reports for survey-specific data and other findings:

Alaska Fisheries Information Network (AKFIN). (2023). *AFSC goundfish assessment program design-based production data*. NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program; https://www.psmfc.org/program/alaska-fisheries-information-network-akfin; U.S. Dep. Commer.

Hoff, G. R. (2016). *Results of the 2016 eastern Bering Sea upper continental slope survey of groundfishes and invertebrate resources* (NOAA Tech. Memo. NOAA-AFSC-339). U.S. Dep. Commer. https://doi.org/10.7289/V5/TM-AFSC-339

Markowitz, E. H., Dawson, E. J., Anderson, A. B., Rohan, S. K., Charriere, N. E., Prohaska, B. K., and Stevenson, D. E. (2023). *Results of the 2022 eastern and northern Bering Sea continental shelf bottom trawl survey of groundfish and invertebrate fauna* (NOAA Tech. Memo. NMFS-AFSC-469; p. 213). U.S. Dep. Commer.

NOAA Fisheries Alaska Fisheries Science Center. (2023). *Fisheries one stop shop public data: RACE division bottom trawl survey data query*. https://www.fisheries.noaa.gov/foss; U.S. Dep. Commer.

NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program. (2023). *AFSC goundfish assessment program design-based production data*. https://www.fisheries.noaa.gov/alaska/science-data/groundfish-assessment-program-bottom-trawl-surveys; U.S. Dep. Commer.

Von Szalay, P. G., and Raring, N. W. (2018). *Data report: 2017 Gulf of Alaska bottom trawl survey* (NOAA Tech. Memo. NMFS-AFSC-374). U.S. Dep. Commer. https:

//doi.org/10.7289/V5/TM-AFSC-374
Von Szalay, P. G., and Raring, N. W. (2020). *Data report: 2018 Aleutian Islands bottom trawl survey* (NOAA Tech. Memo. NMFS-AFSC-409). U.S. Dep. Commer. https://doi.org/10.25923/qe5v-fz70

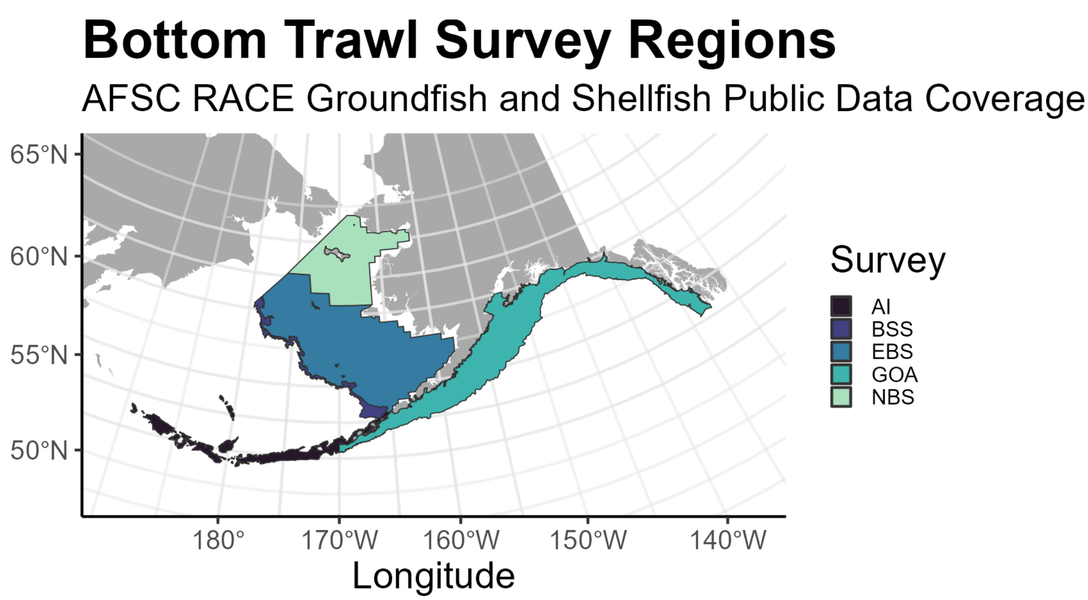
Access Constraints

There are no legal restrictions on access to the data. They reside in public domain and can be freely distributed.

User Constraints: Users must read and fully comprehend the metadata prior to use. Data should not be used beyond the limits of the source scale. Acknowledgement of AFSC Groundfish Assessment Program, as the source from which these data were obtained, in any publications and/or other representations of these data, is suggested.

2. Survey Background

2.1. Bottom trawl surveys and regions



- **Aleutian Islands (AI)** (Von Szalay and Raring, 2020)
  - Triennial (1990s)/Biennial since 2000 in even years
  - Modified Index-Stratified Random of Successful Stations Survey Design
- **Eastern Bering Sea Slope (BSS)** (Hoff, 2016)
  - Intermittent (funding dependent)
  - Modified Index-Stratified Random of Successful Stations Survey Design
- **Eastern Bering Sea Shelf (EBS)** (Markowitz et al., 2023)
  - Annual
  - Fixed stations at center of 20 x 20 nm grid
- **Gulf of Alaska (GOA)** (Von Szalay and Raring, 2018)

- Triennial (1990s)/Biennial since 2001 in odd years
- Stratified Random Survey Design

- **Northern Bering Sea (NBS)** (Markowitz et al., 2023)
  - Biennial/Annual
  - Fixed stations at center of 20 x 20 nm grid

Table 2.1.: Survey summary stats

Survey	Survey Definition ID	Years	Depth (m)	Area (km2)	# Statistical Areas	# Possible Stations
Aleutian Islands Bottom Trawl Survey	52	2022 - 1980 (16)	1 - 500	64,415.(	80	1,312
Eastern Bering Sea Slope Bottom Trawl Survey	78	2016 - 2002 (6)	201 - 800	21,134.2	4	
Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	98	2023 - 1982 (41)	1 - 200	492,989.	29	515
Gulf of Alaska Bottom Trawl Survey	47	2023 - 1984 (18)	1 - 1,000	314,087.4	39	6,939
Northern Bering Sea Crab/Groundfish Survey - Eastern Bering Sea Shelf Survey Extension	143	2022 - 2010 (5)	1 - 100	198,866.	4	144

2.2. Survey History

2.2.1. Aleutian Islands Survey

2.2.2. Bering Sea Survey

2.2.3. Bering Sea Slope Survey

2.2.4. Gulf of Alaska Survey

3. Workflow

The code/run.R script houses the sequence of programs that calculate the standard data products resulting from the NOAA AFSC GAP bottom trawl surveys. Standard data products are the CPUE, BIOMASS, SIZECOMP, and AGECOMP tables in the GAP_-PRODUCTS Oracle schema. The tables are slated to be updated twice a year, once after the survey season following finalization of that summer's bottom trawl survey data to incorporate the new catch, size, and effort data and once prior to an upcoming survey to incorporate new age data that were processed after the prior summer's survey season ended. This second pre-survey production run will also incorporate changes in the data due to the specimen voucher process as well as other post-hoc changes in the survey data.

Below is a summary of the workflow:

1. Import versions of the tables in GAP_PRODUCTS locally within the gap_products repository to compare with the updated production tables. Any changes to a production table will be compared and checked to make sure those changes are intentional and documented.
2. Use the gapindex R package to calculate the four major standard data products: CPUE, BIOMASS, SIZECOMP, AGECOMP. These tables are compared and checked to their respective locally saved copies and any changes to the tables are vetted and documented. These tables are then uploaded to the GAP_PRODUCTS Oracle schema.
3. Calculate the various materialized views for AKFIN and FOSS purposes. Since these are derivative of the tables in GAP_PRODUCTS as well as other base tables in RACEBASE and RACE_DATA, it is not necessary to check these views in addition to the data checks done in the previous steps.

3.1. Data levels

GAP produces numerous data products* that are subjected to different levels of processing, ranging from raw to highly-derived. The suitability of these data products for analysis varies and there is ambiguity about which data products can be used for which purpose. This ambiguity can create challenges in communicating about data products and potentially lead to misunderstanding and misuse of data. One approach to communicating about the level of processing applied to data products and their suitability for analysis is to describe data products using a Data Processing Level system. Data Processing Level systems are widely used in earth system sciences to characterize the extent of processing that has been applied to data products. For example, the NOAA National Centers for Environmental Information (NCEI) Satellite Program uses a Data Processing Level system to describe data on a scale of 0-4, where Level 0 is raw data and Level 4 is model output or results from analysis. Example of how NASA remote sensing data products are shared through a public data portal with levels of data processing and documentation.

For more information, see Sean Rohan's October 2022 SCRUGS presentation on the topic.

- **Level 0**: Raw and unprocessed data. Ex: Data on the G drive, some tables in RACE_DATA
- **Level 1A**: Data products with QA/QC applied that may or may not be expanded to analysis units, but either not georeferenced or does not include full metadata. Ex: Some tables in RACE_DATA and RACEBASE
- **Level 2**: Analysis-ready data products that are derived for a standardized extent and account for zeros and missing/bad data. Ex: CPUE tables, some data products in public-facing archives and repositories
- **Level 3**: Data products that are synthesized across a standardized extent, often inputs in a higher-level analytical product. Ex: Abundance indices, some data products in public-facing archives and repositories
- **Level 4**: Analytically generated data products that are derived from lower-level data, often to inform management. Ex: Biological reference points from stock assessments, Essential Fish Habitat layers, indicators in Ecosystem Status Reports and Ecosystem and Socioeconomic Profiles

4. News

4.1. September 2023

- Provisional data product tables – CPUE, BIOMASS, SIZECOMP, and AGECOMP – as well as provisional support tables – AREA, STRATUM_GROUPS, METADATA_COLUMN, SPECIES_YEAR, SURVEY_DESIGN – are available in the GAP_PRODUCTS Oracle schema with updated 2023 GOA and EBS survey data.
- Provisional AKFIN and FOSS tables are also available in the GAP_PRODUCTS Oracle schema. These include: AKFIN_AGECOMP, AKFIN_AREA, AKFIN_BIOMASS, AKFIN_CATCH, AKFIN_CPUE, AKFIN_CRUISE, AKFIN_HAUL, AKFIN_LENGTH, AKFIN_METADATA_COLUMN, AKFIN_SIZECOMP, AKFIN_SPECIMEN, AKFIN_SURVEY_DESIGN, AKFIN_STRATUM_GROUPS, FOSS_CATCH, FOSS_CPUE_PRESONLY, FOSS_HAUL, and FOSS_TAXON_GROUP.

Part II.

GAP Production Data

Data Description

The Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC) conducts fisheries-independent bottom trawl surveys to monitor the condition of the demersal fish and crab stocks of Alaska. These data are developed to describe the temporal distribution and abundance of commercially and ecologically important groundfish species, examine the changes in the species composition of the fauna over time and space, and describe the physical environment of the groundfish habitat.

Users must read and fully comprehend the metadata prior to use. Data should not be used beyond the limits of the source scale. Acknowledgement of NOAA, as the source from which these data were obtained, in any publications and/or other representations of these data, is suggested. These data are compiled and approved annually after each summer survey season. The data from previous years are unlikely to change substantially once published. Some survey data are excluded, such as non-standard stations, surveys completed in earlier years using different/non-standard gear, and special tows and non-standard data collections.

Cite this data

Use the below bibtext citations, as cited in our group's citation repository for citing the data created and maintained in this repo (NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program, 2023). Add "note = {Accessed: mm/dd/yyyy}" to append the day this data was accessed.

```
@misc{GAPProducts,
  author = {{NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program}}
  year = {2023},
  title = {AFSC Goundfish Assessment Program Design-Based Production Data},
  howpublished = {https://www.fisheries.noaa.gov/alaska/science-data/groundfish-assessment-
  publisher = {{U.S. Dep. Commer.}},
  copyright = {Public Domain}
}
```

Data Creation

These data are created using the gapindex R package.

5. Universal Column Metadata

These tables provide the column metadata for all of the tables and views in GAP_-PRODUCTS. These tables are created by the Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC). The GitHub repository for the scripts that created this code can be found at https://github.com/afsc-gap-products/gap_products. These data were last updated August 08, 2023. There are no legal restrictions on access to the data. For more information about codes used in the tables, please refer to the survey code books (https://www.fisheries.noaa.gov/resource/document/groundfish-survey-species-code-manual-and-data-codes-manual).

Table 5.1.: Universal stock metadata that users can use to document their table columns.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip- tion
ABUNDAN HAUL	Design-based index approved haul	logical		Logical, describing if this haul was conducted in a standard manner and thus used for design-based index estimates (TRUE) or not (FALSE).
ACTIVE	Vessel Active/Inactive	logical		Logical, describing if a vessel is active (TRUE) or not (FALSE).
AGE	Age bin of taxon	year	NUMBER(3	Age bin of a taxon in years estimated by the age comp estimate.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
AGENCY_ACRONYM	Acroynm of listed Agency	text abbreviated	VARCHAR2(255 BYTE)	Abbreviated agencies that are affiliated with the Alaska bottom trawl survey. The column 'agency_acronym' is associated with the 'agency_short' and 'agency_long' columns.
AGENCY_JOIN	Agency's ID code	ID code	NUMBER(3	Affiliated agency ID code.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
AGENCY_LONG	Agency's Offical Name	text	VARCHAR2(255 BYTE)	Full official name of affiliated agencies to the Alaska bottom trawl survey. The column 'agency_long' is associated with the 'agency_acronym' and 'agency_short' columns.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
AGENCY_SHORT	Agency's Shorthand Name	text	VARCHAR2 BYTE)	A sort version of the full official name of affiliated agencies to the Alaska bottom trawl survey. The column 'agency_short' is associated with the 'agency_acronym' and 'agency_long' columns.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
AREA_ID	Area ID code	ID code	NUMBER(38,0)	Area ID code for each statistical area used to produce production estimates (e.g., biomass, population, age comps, length comps). Each area ID is unique within each survey.
AREA_-KM2	Area (km<sup>2	kilometers squared	NUMBER(3	Area in thousands of square kilometers.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
AREA_-NAME	Area ID Name	text	VARCHAR2(4000 BYTE)	Descriptive name of each AREA_ID. These names often identify the region, depth ranges, or other regional informa-tion for the area ID.
AREA_-SWEPT_-KM2	Area Swept (km)	kilometers	NUMBER(3	The area the net covered while the net was fishing (kilome-ters squared), defined as the distance fished times the net width.

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
AREA_TYPE	Area ID Type Description	category	VARCHAR2(255 BYTE)	The type of stratum that AREA_ID represents. Types include: STRATUM, REGION, DEPTH, SUBAREA, INPFC BY DEPTH, INPFC, SUBAREA BY DEPTH, REGULATORY AREA, NMFS STATISTICAL AREA.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
BIOMASS_CI_LOWER	Estimated Biomass Lower Confidence Interval	numeric	NUMBER(3	The estimated biomass lower confidence interval caught in the survey for a species, group, or total for a given survey.
BIOMASS_CI_UPPER	Estimated Biomass Upper Confidence Interval	numeric	NUMBER(38,6)	The estimated biomass upper confidence interval caught in the survey for a species, group, or total for a given survey.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
BIOMASS_DF	Estimated Biomass Degrees of Freedom	numeric	NUMBER(3	The estimated biomass degrees of freedom caught in the survey for a species, group, or total for a given survey.
BIOMASS_MT	Estimated Biomass	numeric	NUMBER(38,6)	The estimated biomass caught in the survey for a species, group, or total for a given survey.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
BIOMASS_VAR	Estimated Biomass Variance	numeric	NUMBER(3	The estimated biomass variance caught in the survey for a species, group, or total for a given survey.
BOTTOM_TEMPERATURE_C	Bottom Temperature (Degrees Celsius)	degrees Celsius	NUMBER(38,1)	Bottom temperature (tenths of a degree Celsius); NA indicates removed or missing values.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
BOTTOM_TYPE_CODE	Seafloor bottom type code	ID code	NUMBER(3	Bottom type on sea floor at haul location. For a complete list of bottom type ID codes, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).
CLASSIFICATION	Taxonomic classification rank group	category	VARCHAR2(255 BYTE)	Phylogenetic classification group rank for a given species.
CLASS_TAXON	Class phylogenetic rank	category	VARCHAR2 BYTE)	Phylogenetic latin rank of class_taxon of a given species.

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
COMMENTS	Comments	text	VARCHAR2(4000 BYTE)	Comments regarding row observation.
COMMON_NAME	Taxon Common Name	text	VARCHAR2(BYTE)	The common name of the marine organism associated with the 'scientific_name' and 'species_code' columns. For a complete species list, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
COUNT	Taxon Count	count, whole number resolution	NUMBER(38,0)	Total number of individuals caught in haul by taxon, represented in whole numbers used in calculation.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
COUNTRY_ID	Vessel Name	text	VARCHAR2 BYTE)	Name of the vessel used to collect data for that haul. The column 'vessel_-name' is associated with the 'vessel_id' column. Note that it is possible for a vessel to have a new name but the same vessel id number. For a complete list of vessel ID codes, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip- tion
CPUE_-KGHA	Weight CPUE (kg/ha)	kilograms per hectare	NUMBER(38,6)	Catch weight (kilo- grams) divided by area (hectares) swept by the net.
CPUE_-KGKM2	Weight CPUE (kg/km<su	kilograms per kilometers squared	NUMBER(3	Catch weight (kilo- grams) divided by area (squared kilome- ters) swept by the net.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
CPUE_KGKM2_MEAN	Mean Weight CPUE	kilograms per kilometers squared	NUMBER(38,6)	The mean of catch weight (kilograms) divided by area (squared kilometers) swept by the net used in design-based indicie calculation.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
CPUE_KGKM2_VAR	Variance of the Mean Weight CPUE	kilograms per kilometers squared	NUMBER(3	The variance of mean of catch weight (kilograms) divided by area (squared kilometers) swept by the net used in design-based indicie calculation.
CPUE_NOHA	Number CPUE (no/ha)	count per hectare	NUMBER(38,6)	Catch number (in number of organisms) per area (hectares) swept by the net.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
CPUE_NOKM2	Number CPUE (no/km<su	count per kilometers squared	NUMBER(3	Catch number (in number of organisms) per area (squared kilometers) swept by the net.
CPUE_NOKM2_MEAN	Mean Numberic CPUE	count per kilometers squared	NUMBER(38,6)	The mean of catch count (number) divided by area (squared kilometers) swept by the net used in design-based indicie calculation.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
CPUE_NOKM2_VAR	Variance of the Mean Numeric CPUE	count per kilometers squared	NUMBER(3	The variance of mMean of catch count (number) divided by area (squared kilometers) swept by the net used in design-based indicie calculation.
CRS	Coordinate Reference System	ID code	VARCHAR2(5 BYTE)	Coordinate reference system that areas (like AREA_KM2) are calculated in, as defined by https://spatialreference.org/ (e.g., ”+proj=longlat”, ”EPSG:3338”).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
CRUISE	Cruise ID	ID code	NUMBER(3	This is a six-digit number identifying the cruise number of the form: YYYY99 (where YYYY = year of the cruise; 99 = 2-digit number and is sequential; 01 denotes the first cruise that vessel made in this year, 02 is the second, etc.).

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
CRUISEJOIN	Cruise ID	ID code	NUMBER(38,0)	This is a unique numeric identifier assigned to each survey, vessel, and year combination.
DATABASE	Genus phylogenetic rank	category	VARCHAR2(255 BYTE)	Taxonomic database source (e.g., "ITIS", "WORMS").
DATABASE_ID	Subfamily phylogenetic rank	ID code	VARCHAR2(255 BYTE)	Species ID code of a species in the taxonomic "DATABASE" source.
DATE	Date	YYYY-MM-DD	DATE	The date (YYYY-MM-DD) of the event (e.g., cruise).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
DATE_-END	End Date	YYYY-MM-DD	DATE	The date (YYYY-MM-DD) of the end of the event (e.g., cruise).
DATE_-START	Start Date	YYYY-MM-DD	DATE	The date (YYYY-MM-DD) of the beginning of the event (e.g., cruise).
DATE_-TIME	Date and Time	MM/DD/YYYY HH::MM	DATE	The date (MM/DD/YYYY) and time (HH:MM) of the haul.
DATE_-TIME_END	End Date and Time	MM/DD/YY HH::MM	DATE	The date (MM/DD/YYYY) and time (HH:MM) of the end of the haul.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
DATE_-TIME_-START	Start Date and Time	MM/DD/YYYY HH::MM	DATE	The date (MM/DD/YYYY) and time (HH:MM) of the beginning of the haul.
DEPTH_M	Depth (m)	degrees Celsius	NUMBER(3	Bottom depth (tenths of a meter).
DEPTH_-MAX_M	Area ID Maximum Depth (m)	meters	NUMBER(38,3)	Maximum depth (meters) of the area covered by AREA_ID.
DEPTH_-MIN_M	Area ID Minimum Depth (m)	meters	NUMBER(3	Minimum depth (meters) of the area covered by AREA_ID.
DESCRIPTION	Description	text	VARCHAR2(4000 BYTE)	Description of row observation.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
DESIGN_-YEAR	Design year	year	NUMBER(1	The year the survey area stratum (e.g., statistical stratum, summary area, region) was impli-mented in.
DISTANCE_-FISHED_-KM	Distance Fished (km)	degrees Celsius	NUMBER(38,3)	Distance the net fished (thou-sandths of kilome-ters).
DUMMY	dummy	dummy	VARCHAR2 BYTE)	dummy
DURATION_-HR	Tow Duration (decimal hr)	hours	NUMBER(38,1)	This is the elapsed time between start and end of a haul (decimal hours).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
FAMILY	Suborder phylogenetic rank	category	VARCHAR2 BYTE)	Phylogenetic latin rank of family of a given species.
GEAR_DEPTH_M	Gear depth	meters	NUMBER(38,1)	Depth gear was deployed at (tenths of a meter). Gear depth plus net height equals bottom depth.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
GEAR_ID	Gear ID code	ID code	NUMBER(3	Type of trawl or gear deployed. For a complete list of vessel gear type ID codes, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).
GENUS_-TAXON	Genus phyloge-netic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank of genus_-taxon of a given species.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
HAUL	Haul Number	ID code	NUMBER(3	This number uniquely identifies a sampling event (haul) within a cruise. It is a sequential number, in chronological order of occurrence.
HAULJOIN	Haul ID	ID code	NUMBER(38,0)	This is a unique numeric identifier assigned to each (vessel, cruise, and haul) combination.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
HAUL_TYPE	Haul Sampling Type	ID code	NUMBER(3	Type of haul sampling method. For a complete list of haul type ID codes, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).
ID_RANK	Lowest taxonomic rank	text	VARCHAR2(255 BYTE)	Lowest taxonomic rank of a given species entry.

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
ITIS	ITIS Taxonomic Serial Number	ID code	NUMBER(3	Species code as identified in the Integrated Taxonomic Information System (https://itis.gov/).
KINGDOM_TAXON	kingdom_taxon phylogenetic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank of kingdom_taxon of a given species.
LATITUDE_DD	Latitude (decimal degrees)	decimal degrees	NUMBER(3	Latitude (one hundred thousandth of a decimal degree).
LATITUDE_DD_END	End Latitude (decimal degrees)	decimal degrees	NUMBER(38,6)	Latitude (one hundred thousandth of a decimal degree) of the end of the haul.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
LATITUDE_DD_START	Start Latitude (decimal degrees)	decimal degrees	NUMBER(3	Latitude (one hundred thousandth of a decimal degree) of the start of the haul.
LENGTH_MM	Length of a specimen	millimeters	NUMBER(10,0)	Length of a specimen in millimeters.
LENGTH_MM_MEAN	Mean length at age weighted by numbers at length	numeric	NUMBER(3	Mean length estimated in age comp estimate.
LENGTH_MM_SD	standard deviation of length at age weighted by numbers at length	numeric	NUMBER(38,3)	Variance of mean length estimated in age comp estimate.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
LONGITUD DD	Longitude (decimal degrees)	decimal degrees	NUMBER(3	Longitude (one hundred thousandth of a decimal degree).
LONGITUDE_DD_END	End Longitude (decimal degrees)	decimal degrees	NUMBER(38,6)	Longitude (one hundred thousandth of a decimal degree) of the end of the haul.
LONGITUD DD_START	Start Longitude (decimal degrees)	decimal degrees	NUMBER(3	Longitude (one hundred thousandth of a decimal degree) of the start of the haul.
METADATA_COLNAME	Column name	text	VARCHAR2(255 BYTE)	Name of the column in a table.
METADATA COLNAME_DESC	column description	text	VARCHAR2 BYTE)	Descritpion of the column.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
METADATA_-COL-NAME_-LONG	Column name spelled out	text	VARCHAR2(255 BYTE)	Long name for the column.
METADATA SENTENCE	Sentence	text	VARCHAR2 BYTE)	Table metadata sentence.
METADATA_-SEN-TENCE_-NAME	Metadata sentence name	text	VARCHAR2(255 BYTE)	Name of the metadata sentence.
METADATA SEN-TENCE_-TYPE	Sentence type	text	VARCHAR2 BYTE)	Type of sentence to have in table metadata.
METADATA_-UNITS	Units	category	VARCHAR2(255 BYTE)	Units of the column.
NET_-HEIGHT_-M	Net Height (m)	meters	NUMBER(3	Measured or estimated distance (meters) between footrope and headrope of the trawl.

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
NET_MEASURED	Net measured during haul	logical		Logical, describing if the net was measured (TRUE) or not (FALSE) by wheelhouse and marport programs during the haul.
NET_WIDTH_M	Net Width (m)	meters	NUMBER(3	Measured or estimated distance (meters) between wingtips of the trawl.
N_COUNT	Hauls with taxon counts	numeric	NUMBER(38,0)	Total number of hauls with positive taxon counts used in calculation.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
N_HAUL	Valid hauls	numeric	NUMBER(3	Total number of valid hauls used in calculation.
N_LENGTH	Hauls with taxon lengths	numeric	NUMBER(38,0)	Total number of hauls with taxon length data used in calculation.
N_WEIGHT	Hauls with catch	numeric	NUMBER(3	Total number of hauls with positive catch/weighed taxon data used in calculation.
ORDER_TAXON	Subclass phylogenetic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank (order_taxon of a given species.

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
PERFORM	Haul Performance Code	category	NUMBER(3	This denotes what, if any, issues arose during the haul. For more information, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).
PHYLUM_TAXON	phylum_taxon phylogenetic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank (phylum_taxon of a given species.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
POPULATI CI_LOWER	Estimated Population Lower Confidence Interval	numeric	NUMBER(3	The estimated population lower confidence interval caught in the survey for a species, group, or total for a given survey.
POPULATION_-CI_UPPER	Estimated Population Upper Confidence Interval	numeric	NUMBER(38,6)	The estimated population upper confidence interval caught in the survey for a species, group, or total for a given survey.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip- tion
POPULATIO COUNT	Estimated Popula- tion	numeric	NUMBER(3	The estimated popula- tion caught in the survey for a species, group, or total for a given survey.
POPULATION_ DF	Estimated Popula- tion - Degrees of Freedom	numeric	NUMBER(38,6)	The estimated popula- tion degrees of freedom caught in the survey for a species, group, or total for a given survey.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
POPULATI(VAR	Estimated Population Variance	numeric	NUMBER(3	The estimated population variance caught in the survey for a species, group, or total for a given survey.
PRINCIPAL_INVESTIGATOR	Principle Investigator	text	VARCHAR2(255 BYTE)	First and last name of principle investigator for a project.
PROJECT_TITLE	Title of Special Project	text	VARCHAR2 BYTE)	Special project title.
PROJECT_TITLE_SHORT	Short Title of Special Project	text	VARCHAR2(255 BYTE)	Special project short title (short version of PROJECT_TITLE).

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
REASON	Naming status for species	text	VARCHAR2(255 BYTE)	Description of species' naming status (e.g., "synonym", "preoccupied","misspelling")
SCIENTIFIC_NAME	Taxon Scientific Name	text	VARCHAR2(255 BYTE)	The scientific name of the organism associated with the 'common_name' and 'species_code' columns. For a complete taxon list, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
SEX	Sex of a specimen	ID code	NUMBER(3	Sex of a specimen where ”1” = ”Male”, ”2” = ”Female”, ”3” = Unsexed.
SPECIES_CODE	Taxon Code	ID code	NUMBER(38,0)	The species code of the organism associated with the ’common_name’ and ’scientific_name’ columns. For a complete species list, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
SPECIES_-NAME_-ACCEPTED	Scientific name used in taxonomic database	text	VARCHAR2 BYTE)	Scientific name of species used in taxonomic ”DATABASE” column.
SPECIES_-NAME_-SURVEY	Scientific name used in survey data	text	VARCHAR2(5 BYTE)	Scientific name of species histori-cally or currently used in the survey.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
SRVY	Survey	text ab-breviated	VARCHAR2 BYTE)	Abbreviated survey names. The column ’srvy’ is associated with the ’survey’ and ’survey_id’ columns. Northern Bering Sea (NBS), South-eastern Bering Sea (EBS), Bering Sea Slope (BSS), Gulf of Alaska (GOA), Aleutian Islands (AI).

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
STATION	Station ID	ID code	VARCHAR2(255 BYTE)	Alphanumeric designation for the station established in the design of a survey.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
STRATUM	Stratum ID	ID code	NUMBER(1	RACE database statistical area for analyzing data. Strata were designed using bathymetry and other geo-graphic and habitat-related elements. The strata are unique to each survey series. Stratum of value 0 indicates experi-mental tows.

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
SUBCLASS_-TAXON	Subclass phylogenetic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank of subclass_taxon of a given species.
SUBFAMILY_-TAXON	Subfamily phylogenetic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank of subfamily_taxon of a given species.
SUBMISSION_-DATE	Date	YYYY-MM-DD	TIMESTAMP	Date special projects were due to be submitted for the upcoming survey season.
SUBORDER_-TAXON	Suborder phylogenetic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank of suborder_taxon of a given species.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
SUBPHYLUM_-TAXON	Subphylum phyloge-netic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank of subphy-lum_taxon of a given species.
SUPERCLASS_-TAXON	Superclass phyloge-netic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank of super-class_-taxon of a given species.
SUPERFAMILY_-TAXON	Superfamily phyloge-netic rank	category	VARCHAR2(255 BYTE)	Phylogenetic latin rank of superfam-ily_taxon of a given species.
SUPERORD	Superorde phyloge-netic rank	category	VARCHAR2 BYTE)	Phylogenetic latin rank of super-order of a given species.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
SURFACE_TEMPERATURE_C	Surface Temperature (Degrees Celsius)	degrees Celsius	NUMBER(38,1)	Surface temperature (tenths of a degree Celsius); NA indicates removed or missing values.
SURVEY	Survey Name	text	VARCHAR2 BYTE)	Name and description of survey. The column 'survey' is associated with the 'srvy' and 'survey_id' columns.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
SURVEY_-DEFINI-TION_ID	Survey ID	ID code	NUMBER(38,0)	This number uniquely identifies a survey. Name and descrip-tion of survey. The column ’survey_id’ is associated with the ’srvy’ and ’survey’ columns. For a complete list of surveys, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
SURVEY_-ID	Survey ID	ID code	NUMBER(3,	This number uniquely identifies a survey. Name and descrip-tion of survey. The column 'survey_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
				Confidence in the ability of the survey team to correctly identify the taxon to the specified level, based solely on identifica-tion skill (e.g., not likelihood of a taxon being caught at that station on a location-by-location basis). Quality codes follow: **'High'**: High con-fidence and con-sistency. Taxonomy is stable and reliable at this level, and field identifica-tion character-istics are well known and

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
				Confidence in the ability of the survey team to correctly identify the taxon to the specified level, based solely on identification skill (e.g., not likelihood of a taxon being caught at that station on a location-by-location basis). Quality codes follow: **'High'**: High confidence and consistency. Taxonomy is stable and reliable at this level, and field identification characteristics are well known and

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
TRAWLABLE	Trawlable stations	logical		Logical, describing if stations are trawlable (TRUE) or not (FALSE).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
VESSEL_-CALLSIGN	Vessel Call Sign	ID code	NUMBER(3	A call sign is a designated sequence of letters and numbers that are assigned when a vessel, whether it be a sailing yacht, motor yacht, rib or commercial vessel, receives it's Ship Radio Licence. The vessel also receives it's MMSI number, so that each vessel is uniquely identified.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
VESSEL_-COAST_-GUARD_-NUMBER	Vessel Coast Guard Number	ID code	NUMBER(38,0)	Official Identification number as defined by www.dco.uscg.mil. The Official Number (O/N) is the 6 or 7 digit number awarded to the vessel at the time it is first documented with the US Coast Guard. This number remains with the vessel indefinitely and should be marked in accordance with 46 CFR 67.121.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
VESSEL_ID	Vessel ID	ID code	NUMBER(3	ID number of the vessel used to collect data for that haul. The column 'vessel_id' is associated with the 'vessel_-name' column. Note that it is possible for a vessel to have a new name but the same vessel id number. For a complete list of vessel ID codes, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
VESSEL_IMO	Vessel International Maritime Organization Number	ID code	NUMBER(38,0)	The International Maritime Organization (IMO) number consists of the letters ”IMO” followed by a unique, seven-digit number: the pattern is ”NNNNNNN”, where N is a single-digit number, e.g., ”1234567”
VESSEL_LENGTH_M	Vessel Length	meters	NUMBER(3	The length of vessel in meters.

Column name from data	Descriptive column Name	Units	Oracle data type	Column description
VESSEL_-MMSI	Vessel Maritime Mobile Service Identities	ID code	NUMBER(38,0)	Maritime Mobile Service Identities (MMSIs) are nine-digit numbers used by maritime digital selective calling (DSC), automatic identification systems (AIS) and certain other equipment to uniquely identify a ship or a coast radio station.

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
VESSEL_-NAME	Vessel Name	text	VARCHAR2 BYTE)	Name of the vessel used to collect data for that haul. The column 'vessel_-name' is associated with the 'vessel_id' column. Note that it is possible for a vessel to have a new name but the same vessel id number. For a complete list of vessel ID codes, review the [code books](https://www.fisheries.noaa.gov/resou survey-species-code-manual-and-data-codes-manual).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column description
VESSEL_OWNER	Vessel Owner	text	VARCHAR2(255 BYTE)	Name of vessel owner or company.
VESSEL_TONNAGE	Vessel Tonnage	metric tons	NUMBER(3	The tonnage of vessel in metric tons.
WEIGHT_KG	Taxon Weight (kg)	kilograms	NUMBER(38,3)	Weight (thousandths of a kilogram) of individuals in a haul by taxon.
WIRE_LENGTH_M	Trawl wire length	meters	NUMBER(3	Length of wire deployed during a given haul in meters.
WORMS	World Register of Marine Species Taxonomic Serial Number	ID code	NUMBER(38,0)	Species code as identified in the World Register of Marine Species (WoRMS) (https://www.marinespecies.org/).

Column name from data	Descriptiv column Name	Units	Oracle data type	Column descrip-tion
YEAR	Year	year	NUMBER(1	Year the survey was conducted in.

6. Data usage examples

Our production data is created using the {gapindex} R package. [Insert info and examples from {gapindex}]

Part III.

AKFIN

These data are used directly by stock assessors and are provided to The [Alaska Fisheries Information Network (AKFIN)].

The Alaska Fisheries Information Network

The Alaska Fisheries Information Network (AKFIN) is a regional program that consolidates and supports the collection, processing, analysis, and reporting of fisheries statistics for North Pacific and Alaskan fisheries. AKFIN integrates this information into a single data management system using consistent methods and standardized formats. The Network then reports this information on its website, in various publications, and to researchers. The resulting data enables fishery managers, scientists, and associated agencies to supervise fisheries resources more effectively and efficiently.

If you are an AFSC employee with access to data through our internal database Oracle server, use this guide to access our data. If not, reach out to AKFIN for a user account.

Cite this data

Use the below bibtext citations, as cited in our group's citation repository for citing the data created and maintained in this repo (Alaska Fisheries Information Network (AKFIN), 2023). Add "note = {Accessed: mm/dd/yyyy}" to append the day this data was accessed.

```
@misc{GAPakfin,
  author = {{Alaska Fisheries Information Network (AKFIN)}},
  institution = {{NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Prog
  year = {2023},
  title = {AFSC Goundfish Assessment Program Design-Based Production Data},
  howpublished = {https://www.psmfc.org/program/alaska-fisheries-information-network-akfin}
  publisher = {{U.S. Dep. Commer.}},
  copyright = {Public Domain}
}
```

7. Data description

7.1. Data Description

In development

7.2. Data Tables

7.2.1. AKFIN_AGECOMP

Number of rows: 544301

Number of columns: 9

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AGE

Age bin of taxon

year

NUMBER(38,0)

Age bin of a taxon in years estimated by the age comp estimate.

AREA_ID

Area ID code

ID code

NUMBER(38,0)

Area ID code for each statistical area used to produce production estimates (e.g., biomass, population, age comps, length comps). Each area ID is unique within each survey.

LENGTH_MM_MEAN

Mean length at age weighted by numbers at length

numeric

NUMBER(38,3)

Mean length estimated in age comp estimate.

LENGTH_MM_SD

standard deviation of length at age weighted by numbers at length

numeric

NUMBER(38,3)

Variance of mean length estimated in age comp estimate.

POPULATION_COUNT

Estimated Population

numeric

NUMBER(38,6)

The estimated population caught in the survey for a species, group, or total for a given survey.

SEX

Sex of a specimen

ID code

NUMBER(38,0)

Sex of a specimen where “1” = “Male”, “2” = “Female”, “3” = Unsexed.

SPECIES_CODE

Taxon Code

ID code

NUMBER(38,0)

The species code of the organism associated with the 'common_name' and 'scientific_-name' columns. For a complete species list, review the code books.

SURVEY_DEFINITION_ID

Survey ID

ID code

NUMBER(38,0)

This number uniquely identifies a survey. Name and description of survey. The column 'survey_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

YEAR

Year

year

NUMBER(10,0)

Year the survey was conducted in.

7.2.2. AKFIN_AREA

Number of rows: 443

Number of columns: 10

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID code

ID code

NUMBER(38,0)

Area ID code for each statistical area used to produce production estimates (e.g., biomass, population, age comps, length comps). Each area ID is unique within each survey.

AREA_KM2

Area (km²)

kilometers squared

NUMBER(38,3)

Area in thousands of square kilometers.

AREA_NAME

Area ID Name

text

VARCHAR2(4000 BYTE)

Descriptive name of each AREA_ID. These names often identify the region, depth ranges, or other regional information for the area ID.

DEPTH_MAX_M

Area ID Maximum Depth (m)

meters

NUMBER(38,3)

Maximum depth (meters) of the area covered by AREA_ID.

DEPTH_MIN_M

Area ID Minimum Depth (m)

meters

NUMBER(38,3)

Minimum depth (meters) of the area covered by AREA_ID.

DESCRIPTION

Description

text

VARCHAR2(4000 BYTE)

Description of row observation.

DESIGN_YEAR

Design year

year

NUMBER(10,0)

The year the survey area stratum (e.g., statistical stratum, summary area, region) was implimented in.

SURVEY_DEFINITION_ID

Survey ID

ID code

NUMBER(38,0)

This number uniquely identifies a survey. Name and description of survey. The column 'survey_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

TYPE

NA

NA

NA

NA

crs

NA

NA

NA

NA

7.2.3. AKFIN_BIOMASS

Number of rows: 4582456

Number of columns: 16

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID code

ID code

NUMBER(38,0)

Area ID code for each statistical area used to produce production estimates (e.g., biomass, population, age comps, length comps). Each area ID is unique within each survey.

BIOMASS_MT

Estimated Biomass

numeric

NUMBER(38,6)

The estimated biomass caught in the survey for a species, group, or total for a given survey.

BIOMASS_VAR

Estimated Biomass Variance

numeric

NUMBER(38,6)

The estimated biomass variance caught in the survey for a species, group, or total for a given survey.

CPUE_KGKM2_MEAN

Mean Weight CPUE

kilograms per kilometers squared

NUMBER(38,6)

The mean of catch weight (kilograms) divided by area (squared kilometers) swept by the net used in design-based indicie calculation.

CPUE_KGKM2_VAR

Variance of the Mean Weight CPUE

kilograms per kilometers squared

NUMBER(38,6)

The variance of mean of catch weight (kilograms) divided by area (squared kilometers) swept by the net used in design-based indicie calculation.

CPUE_NOKM2_MEAN

Mean Numberic CPUE

count per kilometers squared

NUMBER(38,6)

The mean of catch count (number) divided by area (squared kilometers) swept by the net used in design-based indicie calculation.

CPUE_NOKM2_VAR

Variance of the Mean Numeric CPUE

count per kilometers squared

NUMBER(38,6)

The variance of mMean of catch count (number) divided by area (squared kilometers) swept by the net used in design-based indicie calculation.

N_COUNT

Hauls with taxon counts

numeric

NUMBER(38,0)

Total number of hauls with positive taxon counts used in calculation.

N_HAUL

Valid hauls

numeric

NUMBER(38,0)

Total number of valid hauls used in calculation.

N_LENGTH

Hauls with taxon lengths

numeric

NUMBER(38,0)

Total number of hauls with taxon length data used in calculation.

N_WEIGHT

Hauls with catch

numeric

NUMBER(38,0)

Total number of hauls with positive catch/weighed taxon data used in calculation.

POPULATION_COUNT

Estimated Population

numeric

NUMBER(38,6)

The estimated population caught in the survey for a species, group, or total for a given survey.

POPULATION_VAR

Estimated Population Variance

numeric

NUMBER(38,6)

The estimated population variance caught in the survey for a species, group, or total for a given survey.

SPECIES_CODE

Taxon Code

ID code

NUMBER(38,0)

The species code of the organism associated with the 'common_name' and 'scientific_-name' columns. For a complete species list, review the code books.

SURVEY_DEFINITION_ID

Survey ID

ID code

NUMBER(38,0)

This number uniquely identifies a survey. Name and description of survey. The column 'survey_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

YEAR

Year

year

NUMBER(10,0)

Year the survey was conducted in.

7.2.4. AKFIN_CATCH

Number of rows: 985442

Number of columns: 6

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

CATCHJOIN

NA

NA

NA

NA

COUNT

Taxon Count

count, whole number resolution

NUMBER(38,0)

Total number of individuals caught in haul by taxon, represented in whole numbers used in calculation.

CRUISEJOIN

Cruise ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each survey, vessel, and year combination.

HAULJOIN

Haul ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each (vessel, cruise, and haul) combination.

SPECIES_CODE

Taxon Code

ID code

NUMBER(38,0)

The species code of the organism associated with the 'common_name' and 'scientific_name' columns. For a complete species list, review the code books.

WEIGHT_KG

Taxon Weight (kg)

kilograms

NUMBER(38,3)

Weight (thousandths of a kilogram) of individuals in a haul by taxon.

7.2.5. AKFIN_CPUE

Number of rows: 37655036

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_SWEPT_KM2

Area Swept (km)

kilometers

NUMBER(38,6)

The area the net covered while the net was fishing (kilometers squared), defined as the distance fished times the net width.

COUNT

Taxon Count

count, whole number resolution

NUMBER(38,0)

Total number of individuals caught in haul by taxon, represented in whole numbers used in calculation.

CPUE_KGKM2

Weight CPUE (kg/km²)

kilograms per kilometers squared

NUMBER(38,6)

Catch weight (kilograms) divided by area (squared kilometers) swept by the net.

CPUE_NOKM2

Number CPUE (no/km²)

count per kilometers squared

NUMBER(38,6)

Catch number (in number of organisms) per area (squared kilometers) swept by the net.

HAULJOIN

Haul ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each (vessel, cruise, and haul) combination.

SPECIES_CODE

Taxon Code

ID code

NUMBER(38,0)

The species code of the organism associated with the 'common_name' and 'scientific_name' columns. For a complete species list, review the code books.

WEIGHT_KG

Taxon Weight (kg)

kilograms

NUMBER(38,3)

Weight (thousandths of a kilogram) of individuals in a haul by taxon.

7.2.6. AKFIN_CRUISE

Number of rows: 185

Number of columns: 10

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

CRUISE

Cruise ID

ID code

NUMBER(38,0)

This is a six-digit number identifying the cruise number of the form: YYYY99 (where YYYY = year of the cruise; 99 = 2-digit number and is sequential; 01 denotes the first cruise that vessel made in this year, 02 is the second, etc.).

CRUISEJOIN

Cruise ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each survey, vessel, and year combination.

DATE_END

End Date

YYYY-MM-DD

DATE

The date (YYYY-MM-DD) of the end of the event (e.g., cruise).

DATE_START

Start Date

YYYY-MM-DD

DATE

The date (YYYY-MM-DD) of the beginning of the event (e.g., cruise).

SPONSOR_ACRONYM

NA

NA

NA

NA

SURVEY_DEFINITION_ID

Survey ID

ID code

NUMBER(38,0)

This number uniquely identifies a survey. Name and description of survey. The column 'survey_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

SURVEY_NAME

NA

NA

NA

NA

VESSEL_ID

Vessel ID

ID code

NUMBER(38,0)

ID number of the vessel used to collect data for that haul. The column 'vessel_id' is associated with the 'vessel_name' column. Note that it is possible for a vessel to have a new name but the same vessel id number. For a complete list of vessel ID codes, review the code books.

VESSEL_NAME

Vessel Name

text

VARCHAR2(255 BYTE)

Name of the vessel used to collect data for that haul. The column 'vessel_name' is associated with the 'vessel_id' column. Note that it is possible for a vessel to have a new name but the same vessel id number. For a complete list of vessel ID codes, review the code books.

YEAR

Year

year

NUMBER(10,0)

Year the survey was conducted in.

7.2.7. AKFIN_LENGTH

Number of rows: 2574444

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

FREQUENCY

NA

NA

NA

NA

HAULJOIN

Haul ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each (vessel, cruise, and haul) combination.

LENGTH_MM

Length of a specimen

millimeters

NUMBER(10,0)

Length of a specimen in millimeters.

LENGTH_TYPE

NA

NA

NA

NA

SAMPLE_TYPE

NA

NA

NA

NA

SEX

Sex of a specimen

ID code

NUMBER(38,0)

Sex of a specimen where “1” = “Male”, “2” = “Female”, “3” = Unsexed.

SPECIES_CODE

Taxon Code

ID code

NUMBER(38,0)

The species code of the organism associated with the 'common_name' and 'scientific_-name' columns. For a complete species list, review the code books.

7.2.8. AKFIN_METADATA_COLUMN

Number of rows: 134

Number of columns: 5

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

METADATA_COLNAME

Column name

text

VARCHAR2(255 BYTE)

Name of the column in a table.

METADATA_COLNAME_DESC

column description

text

VARCHAR2(4000 BYTE)

Descritpion of the column.

METADATA_COLNAME_LONG

Column name spelled out

text

VARCHAR2(255 BYTE)

Long name for the column.

METADATA_DATATYPE

NA

NA

NA

NA

METADATA_UNITS

Units

category

VARCHAR2(255 BYTE)

Units of the column.

7.2.9. AKFIN_SIZECOMP

Number of rows: 3113209

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID code

ID code

NUMBER(38,0)

Area ID code for each statistical area used to produce production estimates (e.g., biomass, population, age comps, length comps). Each area ID is unique within each survey.

LENGTH_MM

Length of a specimen

millimeters

NUMBER(10,0)

Length of a specimen in millimeters.

POPULATION_COUNT

Estimated Population

numeric

NUMBER(38,6)

The estimated population caught in the survey for a species, group, or total for a given survey.

SEX

Sex of a specimen

ID code

NUMBER(38,0)

Sex of a specimen where “1” = “Male”, “2” = “Female”, “3” = Unsexed.

SPECIES_CODE

Taxon Code

ID code

NUMBER(38,0)

The species code of the organism associated with the ‘common_name’ and ‘scientific_name’ columns. For a complete species list, review the code books.

SURVEY_DEFINITION_ID

Survey ID

ID code

NUMBER(38,0)

This number uniquely identifies a survey. Name and description of survey. The column ‘survey_id’ is associated with the ‘srvy’ and ‘survey’ columns. For a complete list of surveys, review the code books.

YEAR

Year

year

NUMBER(10,0)

Year the survey was conducted in.

7.2.10. AKFIN_SPECIMEN

Number of rows: 359317

Number of columns: 17

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AGE_DETERMINATION_METHOD

NA

NA

NA

NA

AGE_YEARS

NA

NA

NA

NA

CRUISEJOIN

Cruise ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each survey, vessel, and year combination.

GONAD_G

NA

NA

NA

NA

HAULJOIN

Haul ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each (vessel, cruise, and haul) combination.

LENGTH_MM

Length of a specimen

millimeters

NUMBER(10,0)

Length of a specimen in millimeters.

MATURITY

NA

NA

NA

NA

MATURITY_TABLE

NA

NA

NA

NA

REGION

NA

NA

NA

NA

SEX

Sex of a specimen

ID code

NUMBER(38,0)

Sex of a specimen where “1” = “Male”, “2” = “Female”, “3” = Unsexed.

SPECIES_CODE

Taxon Code

ID code

NUMBER(38,0)

The species code of the organism associated with the ‘common_name’ and ‘scientific_-name’ columns. For a complete species list, review the code books.

SPECIMEN_ID

NA

NA

NA

NA

SPECIMEN_SAMPLE_TYPE

NA

NA

NA

NA

SPECIMEN_SUBSAMPLE_METHOD

NA

NA

NA

NA

STRATUM

Stratum ID

ID code

NUMBER(10,0)

RACE database statistical area for analyzing data. Strata were designed using bathymetry and other geographic and habitat-related elements. The strata are unique to each survey series. Stratum of value 0 indicates experimental tows.

VESSEL_ID

Vessel ID

ID code

NUMBER(38,0)

ID number of the vessel used to collect data for that haul. The column 'vessel_id' is associated with the 'vessel_name' column. Note that it is possible for a vessel to have a new name but the same vessel id number. For a complete list of vessel ID codes, review the code books.

WEIGHT_G

NA

NA

NA

NA

7.2.11. AKFIN_STRATUM_GROUPS

Number of rows: 774

Number of columns: 4

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_ID

Area ID code

ID code

NUMBER(38,0)

Area ID code for each statistical area used to produce production estimates (e.g., biomass, population, age comps, length comps). Each area ID is unique within each survey.

DESIGN_YEAR

Design year

year

NUMBER(10,0)

The year the survey area stratum (e.g., statistical stratum, summary area, region) was implimented in.

STRATUM

Stratum ID

ID code

NUMBER(10,0)

RACE database statistical area for analyzing data. Strata were designed using bathymetry and other geographic and habitat-related elements. The strata are unique to each survey series. Stratum of value 0 indicates experimental tows.

SURVEY_DEFINITION_ID

Survey ID

ID code

NUMBER(38,0)

This number uniquely identifies a survey. Name and description of survey. The column 'survey_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

7.2.12. AKFIN_SURVEY_DESIGN

Number of rows: 126

Number of columns: 3

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

DESIGN_YEAR

Design year

year

NUMBER(10,0)

The year the survey area stratum (e.g., statistical stratum, summary area, region) was implimented in.

SURVEY_DEFINITION_ID

Survey ID

ID code

NUMBER(38,0)

This number uniquely identifies a survey. Name and description of survey. The column 'survey_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

YEAR

Year

year

NUMBER(10,0)

Year the survey was conducted in.

8. Accessing Data

8.1. Access data via Oracle (AFSC only)

AFSC `Oracle` users can access the database via `SQL developer` to view and pull the production data directly from the `GAP_PRODUCTS` `Oracle` schema. The user can also use `SQL developer` to view and pull the GAP Products data directly from the `GAP_PRODUCTS` `Oracle` schema.

8.1.1. Connect to Oracle from R

Many users will want to access the data from `Oracle` using `R`. The user will need to install the `RODBC` `R` package and ask OFIS (IT) connect `R` to `Oracle`. Then, use the following code in `R` to establish a connection from `R` to `Oracle`:

Here, the user can establish the oracle connection by entering their username and password in the `channel <- gapindex::oracle_connect()` function. Never save usernames or passwords in scripts that may be intentionally or unintentionally shared with others. If no username and password is entered in the function, pop-ups will appear on the screen asking for the username and password.

8.2. Data SQL Query Examples:

8.2.1. Ex. 0: Select all data from a table

You can download all of the tables locally using a variation of the code below. Once connected, pull and save the tables of interest into the `R` environment.

```
locations <- c(
  "GAP_PRODUCTS.AKFIN_AGECOMP",
  "GAP_PRODUCTS.AKFIN_AREA",
  "GAP_PRODUCTS.AKFIN_BIOMASS",
  "GAP_PRODUCTS.AKFIN_CATCH",
  "GAP_PRODUCTS.AKFIN_CPUE",
  "GAP_PRODUCTS.AKFIN_CRUISE",
  "GAP_PRODUCTS.AKFIN_HAUL",
  "GAP_PRODUCTS.AKFIN_LENGTH",
  "GAP_PRODUCTS.AKFIN_METADATA_COLUMN",
  "GAP_PRODUCTS.AKFIN_SIZECOMP",
  "GAP_PRODUCTS.AKFIN_SPECIMEN",
  "GAP_PRODUCTS.AKFIN_STRATUM_GROUPS",
  "GAP_PRODUCTS.AKFIN_SURVEY_DESIGN",
  "GAP_PRODUCTS.AKFIN_TAXONOMICS_WORMS"
)

for (i in 1:length(locations)) {
  print(locations[i])
  a <- RODBC::sqlQuery(channel, paste0("SELECT * FROM ", locations[i]))
  write.csv(x = a, file = here::here("data", paste0(locations[i], ".csv")))
}
```

8.2.2. Ex. 1: GOA Pacific Ocean perch biomass and abundance

Biomass and abundance for Pacific Ocean perch from 1990 – 2023 for the western/central/eastern GOA management areas as well as for the entire region.

```
dat <- RODBC::sqlQuery(channel = channel,
                       query =
"WITH FILTERED_STRATA AS (
SELECT AREA_ID, DESCRIPTION FROM GAP_PRODUCTS.AKFIN_AREA
WHERE TYPE in ('REGULATORY_AREA', 'REGION')
AND SURVEY_DEFINITION_ID = 47)
SELECT
BIOMASS_MT,
POPULATION_COUNT,
YEAR,
DESCRIPTION
```

```
FROM GAP_PRODUCTS.AKFIN_BIOMASS BIOMASS
JOIN FILTERED_STRATA STRATA
ON STRATA.AREA_ID = BIOMASS.AREA_ID
WHERE BIOMASS.SURVEY_DEFINITION_ID IN 47
AND BIOMASS.SPECIES_CODE = 30060")
```

```
dat0 <- dat %>%
  janitor::clean_names() %>%
  dplyr::select(biomass_mt, population_count, year, area = description) %>%
  pivot_longer(cols = c("biomass_mt", "population_count"),
               names_to = "var",
               values_to = "val") %>%
  dplyr::mutate(
    val = ifelse(var == "biomass_mt", val/1e6, val/1e9),
    var = ifelse(var == "biomass_mt", "Biomass (Mmt)", "Population (B)"),
    area = gsub(x = area, pattern = " - ", replacement = "\n"),
    area = gsub(x = area, pattern = ": ", replacement = "\n"),
    type = sapply(X = strsplit(x = area, split = "\n", fixed = TRUE), `[[`, 2))  %>%
  dplyr::arrange(type) %>%
  dplyr::mutate(
    area = factor(area, levels = unique(area), labels = unique(area), ordered = TRUE))

flextable::flextable(head(dat)) %>%
  theme_zebra() %>%
  flextable::colformat_num(x = ., j = "YEAR", big.mark = "")
```

Table 8.1.: Ex. 1: GOA Pacific Ocean perch biomass and abundance.

BIOMASS_MT	POPULATI COUNT	YEAR	DESCRIPTION
483,622.68	33,902,16	1993	GOA Region: All Strata
771,412.8	1,252,616,603	1996	GOA Region: All Strata

Table 8.1.: Ex. 1: GOA Pacific Ocean perch biomass and abundance.

BIOMASS_MT	POPULATION_COUNT	YEAR	DESCRIPTION
727,063.5	1,212,034,9	1999	GOA Region: All Strata
673,155.1	1,189,370,120	2001	GOA Region: All Strata
457,421.678	1,034,22	2003	GOA Region: All Strata
764,901.4	1,343,536,275	2005	GOA Region: All Strata

```
# install.packages("scales")
library(scales)
figure <- ggplot2::ggplot(
  dat = dat0,
  mapping = aes(x = year, y = val, color = type)) +
  ggplot2::geom_point(size = 3) +
  ggplot2::facet_grid(cols = vars(area), rows = vars(var), scales = "free_y") +
  ggplot2::scale_x_continuous(name = "Year", n.breaks = 3) +
  ggplot2::scale_y_continuous(name = "Estimate", labels = comma) +
  ggplot2::labs(title = 'GOA Pacific Ocean perch biomass and abundance 1990 - 2023')  +
  ggplot2::guides(color=guide_legend(title = "Region Type"))+
  ggplot2::scale_color_grey() +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.direction = "horizontal",
                 legend.position = "bottom")

figure
```

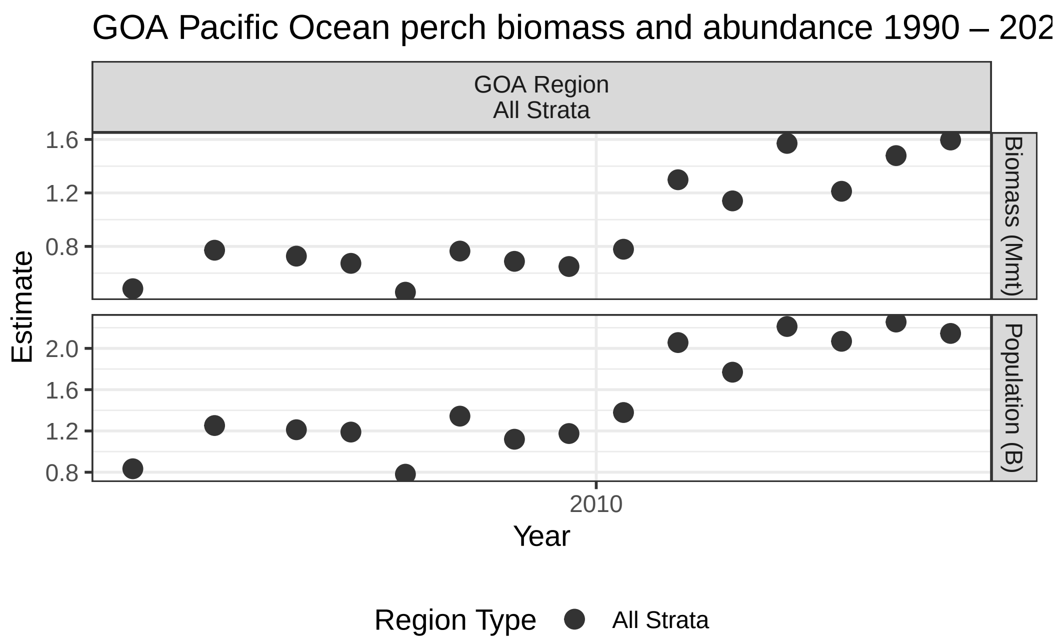


Figure 8.1.: Ex. 1: GOA Pacific Ocean perch biomass and abundance.

8.2.3. Ex. 2: AI Rock sole size compositions and ridge plot

Northern and Southern rock sole size composition data from 1991 – 2022 for the Aleutian Islands, with Ridge plot from `ggridges`.

```
dat <- RODBC::sqlQuery(channel = channel,
                       query =
"WITH FILTERED_STRATA AS (
SELECT
AREA_ID,
DESCRIPTION
FROM GAP_PRODUCTS.AKFIN_AREA
WHERE TYPE = 'REGION'
AND SURVEY_DEFINITION_ID = 52)
SELECT
LENGTH_MM,
YEAR
FROM GAP_PRODUCTS.AKFIN_SIZECOMP SIZECOMP
```

```
JOIN FILTERED_STRATA STRATA
ON STRATA.AREA_ID = SIZECOMP.AREA_ID
WHERE SIZECOMP.SURVEY_DEFINITION_ID IN 52
AND SIZECOMP.SPECIES_CODE IN (10261, 10262)")
```

```
dat0 <- dat %>%
  janitor::clean_names() %>%
  dplyr::mutate(length_cm = length_mm/10)
flextable::flextable(head(dat)) %>%
  theme_zebra() %>%
    flextable::colformat_num(x = ., j = "YEAR", big.mark = "")
```

Table 8.2.: Ex. 2: AI Rock sole size compositions and ridge plot.

LENGTH_-MM	YEAR
110	1997
130	1997
140	1997
150	1997
160	1997
170	1997

```
# install.packages("ggridges")
library(ggridges)
figure <-
  ggplot2::ggplot(
    data = dat0,
    mapping = aes(x = length_cm, y = as.factor(year), fill = stat(x))) +
  ggridges::theme_ridges(center_axis_labels = TRUE) +
  ggridges::geom_density_ridges_gradient(scale = 4, show.legend = FALSE) +
  ggplot2::scale_y_discrete(name = "Year", expand = c(0.01, 0)) +
  ggplot2::scale_x_continuous(name = "Length (cm)", expand = c(0.01, 0)) +
  # ggplot2::scale_fill_grey() +
  ggplot2::labs(title = 'AI Rock sole Size Compositions 1991 - 2022')

figure
```

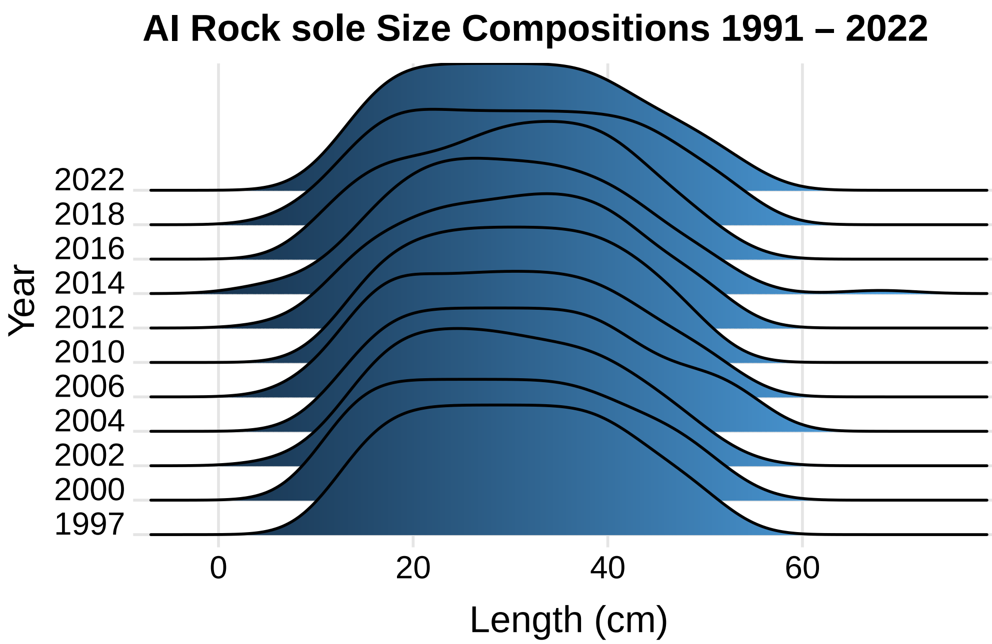


Figure 8.2.: Ex. 2: AI Rock sole size compositions and ridge plot.

8.2.4. Ex. 3: EBS Walleye Pollock Age Compositions and Age Pyramid

Walleye pollock age composition for the EBS Standard Area from 1982 – 2022 and the EBS + NW Area from 1987 – 2022, with age pyramid plot.

```
dat <- RODBC::sqlQuery(channel = channel,
                       query =
"WITH FILTERED_STRATA AS (
SELECT
AREA_ID,
DESCRIPTION
FROM GAP_PRODUCTS.AKFIN_AREA
WHERE TYPE = 'REGION' AND
SURVEY_DEFINITION_ID = 98)
SELECT
AGECOMP.AGE,
```

```
AGECOMP.POPULATION_COUNT,
AGECOMP.SEX
FROM GAP_PRODUCTS.AKFIN_AGECOMP AGECOMP
JOIN FILTERED_STRATA STRATA
ON STRATA.AREA_ID = AGECOMP.AREA_ID
WHERE SURVEY_DEFINITION_ID = 98
AND SPECIES_CODE = 21740
AND AGE >= 0")
```

```
dat0 <- dat %>%
  janitor::clean_names() %>%
  dplyr::filter(sex %in% c(1,2)) %>%
  dplyr::mutate(
    sex = ifelse(sex == 1, "M", "F"),
    population_count = # change male population to negative
      ifelse(sex=="M", population_count*(-1), population_count*1)/1e9)

flextable::flextable(head(dat)) %>% theme_zebra()
```

Table 8.3.: Ex. 3: EBS Walleye Pollock Age Compositions and Age Pyramid.

AGE	POPULATI COUNT	SEX
	133,930,956	1
	2314,043,443	1
	3103,452,65	1
	447,525,134	1
	5203,340,10	1
	6246,665,076	1

```
figure <- ggplot2::ggplot(
  data = dat0,
  mapping =
               aes(x = age,
                   y = population_count,
                   fill = sex)) +
```

```
  ggplot2::scale_fill_grey() +
  ggplot2::geom_bar(stat = "identity") +
  ggplot2::coord_flip() +
  ggplot2::scale_x_continuous(name = "Age") +
  ggplot2::scale_y_continuous(name = "Population (billions)", labels = abs) +
  ggplot2::ggtitle(label = "EBS Walleye Pollock Age Compositions 1982 - 2022")  +
  ggplot2::guides(fill = guide_legend(title = "Sex"))+
  ggplot2::theme_bw()

figure
```

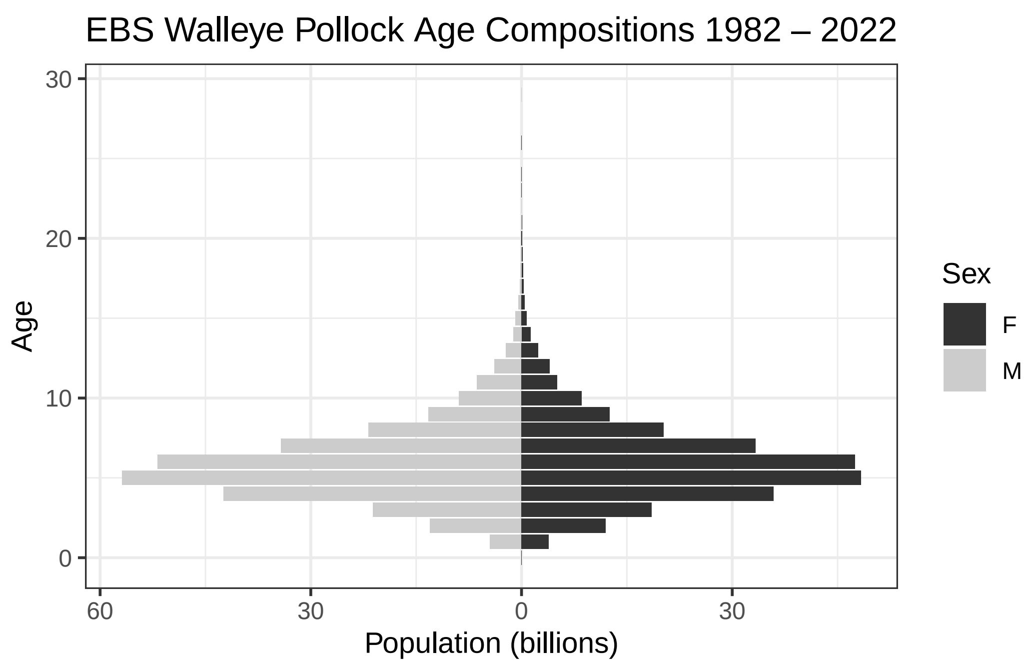


Figure 8.3.: Ex. 3: EBS Walleye Pollock Age Compositions and Age Pyramid.

8.2.5. Ex. 4: NBS Pacific cod biomass and abundance

Pacific cod biomass and abundance data for the NBS by stratum.

```
dat <- RODBC::sqlQuery(channel = channel,
                       query =
```

```
"WITH FILTERED_STRATA AS (
SELECT
AREA_ID,
AREA_NAME,
DESCRIPTION
FROM GAP_PRODUCTS.AKFIN_AREA
WHERE TYPE in ('STRATUM') AND
SURVEY_DEFINITION_ID = 143)
SELECT
BIOMASS.BIOMASS_MT,
BIOMASS.POPULATION_COUNT,
BIOMASS.YEAR,
STRATA.AREA_NAME
FROM GAP_PRODUCTS.AKFIN_BIOMASS BIOMASS
JOIN FILTERED_STRATA STRATA
ON STRATA.AREA_ID = BIOMASS.AREA_ID
WHERE BIOMASS.SURVEY_DEFINITION_ID IN 143
AND BIOMASS.SPECIES_CODE = 21720")
```

```
dat0 <- dat %>%
  janitor::clean_names() %>%
  dplyr::select(biomass_mt, population_count, year, area = area_name) %>%
  pivot_longer(cols = c("biomass_mt", "population_count"),
               names_to = "var",
               values_to = "val") %>%
  dplyr::mutate(
    val = ifelse(var == "biomass_mt", val/1e6, val/1e9),
    var = ifelse(var == "biomass_mt", "Biomass (Mmt)", "Population (B)"),
    area = factor(area, levels = unique(area), labels = unique(area), ordered = TRUE))
flextable::flextable(head(dat)) %>%
  theme_zebra() %>%
  flextable::colformat_num(x = ., j = "YEAR", big.mark = "")
```

Table 8.4.: Ex. 4: NBS Pacific cod biomass and abundance.

BIOMASS_MT	POPULATION_COUNT	YEAR	AREA_NAME
7,462.559	4,724,153	2010	Inner Domain
95,849.983	68,767,498	2021	Inner Domain
107,096.73	102,734,14	2019	Inner Domain
132,490.156	26,187,245	2017	Inner Domain
96,500.697	60,433,135	2022	Inner Domain
147,971.456	45,078,489	2017	Inner Domain

```r
figure <- ggplot2::ggplot(
  dat = dat0,
  mapping = aes(y = val, x = year, fill = area))  +
  ggplot2::geom_bar(position="stack", stat="identity") +
  ggplot2::facet_grid(rows = vars(var), scales = "free_y") +
  ggplot2::scale_y_continuous(name = "Estimate", labels = comma) +
  ggplot2::scale_x_continuous(name = "Year", breaks = unique(dat0$year)) +
  ggplot2::labs(title = 'NBS Pacific cod biomass and abundance by stratum')  +
  ggplot2::guides(fill=guide_legend(title = "Region Type"))+
  ggplot2::scale_fill_grey() +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.direction = "horizontal",
                 legend.position = "bottom")

figure
```

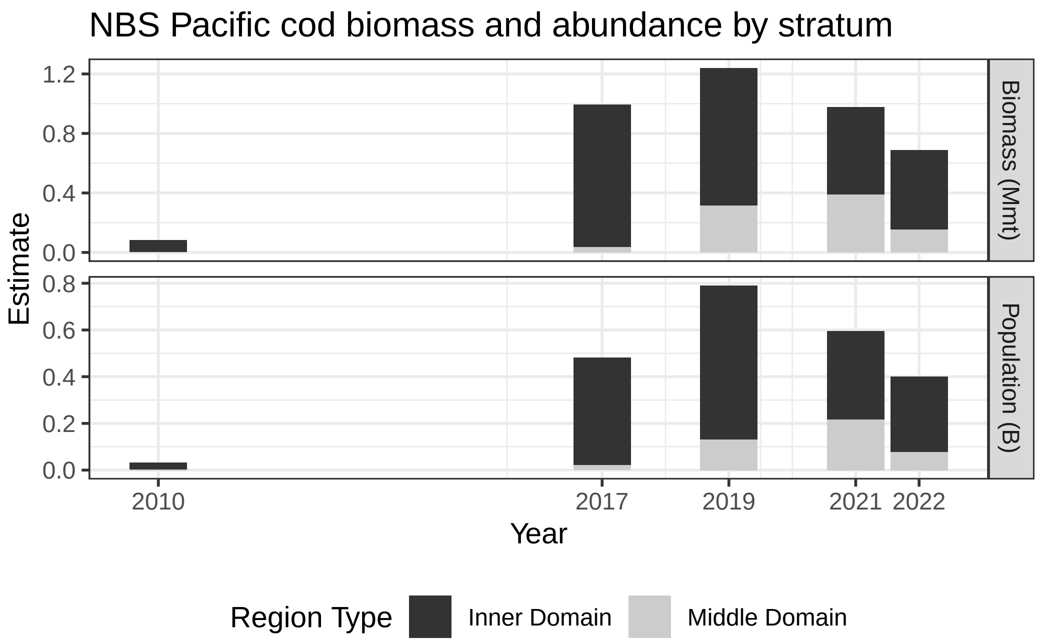


Figure 8.4.: Ex. 4: NBS Pacific cod biomass and abundance.

8.2.6. Ex. 5: GOA Pacific Ocean perch biomass and line plot

Pacific Ocean perch biomass totals for GOA between 1984-2021 from `GAP_PRODUCTS.AKFIN_BIOMASS`

```
dat <- RODBC::sqlQuery(channel = channel,
                       query =
"SELECT
SURVEY_DEFINITION_ID,
BIOMASS_MT,
YEAR
FROM GAP_PRODUCTS.AKFIN_BIOMASS
WHERE SPECIES_CODE = 30060
AND SURVEY_DEFINITION_ID = 47
AND AREA_ID = 99903
AND YEAR BETWEEN 1984 AND 2021;") %>%
  janitor::clean_names() %>%
  dplyr::mutate(biomass_mt = biomass_mt/1000)
```

```
a_mean <- dat %>%
  dplyr::group_by(survey_definition_id) %>%
  dplyr::summarise(biomass_mt = mean(biomass_mt, na.rm = TRUE),
                   minyr = min(year, na.rm = TRUE),
                   maxyr = max(year, na.rm = TRUE))
flextable::flextable(head(dat)) %>%
  theme_zebra() %>%
  flextable::colformat_num(x = ., j = "year", big.mark = "")
```

Table 8.5.: Ex. 5: GOA Pacific Ocean perch biomass and line plot.

survey_-definition_id	biomass_-mt	year
47	483.6226	1993
47	771.4128	1996
47	727.0635	1999
47	673.1551	2001
47	457.4216	2003
47	764.9014	2005

```
figure <-
  ggplot(data = dat,
         mapping = aes(x = year,
                       y = biomass_mt)) +
  ggplot2::geom_point(size = 2.5, color = "grey40") +
  ggplot2::scale_x_continuous(
    name = "Year",
    labels = scales::label_number(
      accuracy = 1,
      big.mark = ""))   +
  ggplot2::scale_y_continuous(
    name = "Biomass (Kmt)",
    labels = comma) +
  ggplot2::geom_segment(
    data = a_mean,
```

```
    mapping = aes(x = minyr,
                  xend = maxyr,
                  y = biomass_mt,
                  yend = biomass_mt),
    linetype = "dashed",
    linewidth = 2) +
  ggplot2::ggtitle(
    label = "GOA Pacific Ocean Perch Biomass 1984-2021",
    subtitle = paste0("Mean = ",
                      formatC(x = a_mean$biomass_mt,
                              digits = 2,
                              big.mark = ",",
                              format = "f"),
                      " Kmt")) +
  ggplot2::theme_bw()

figure
```

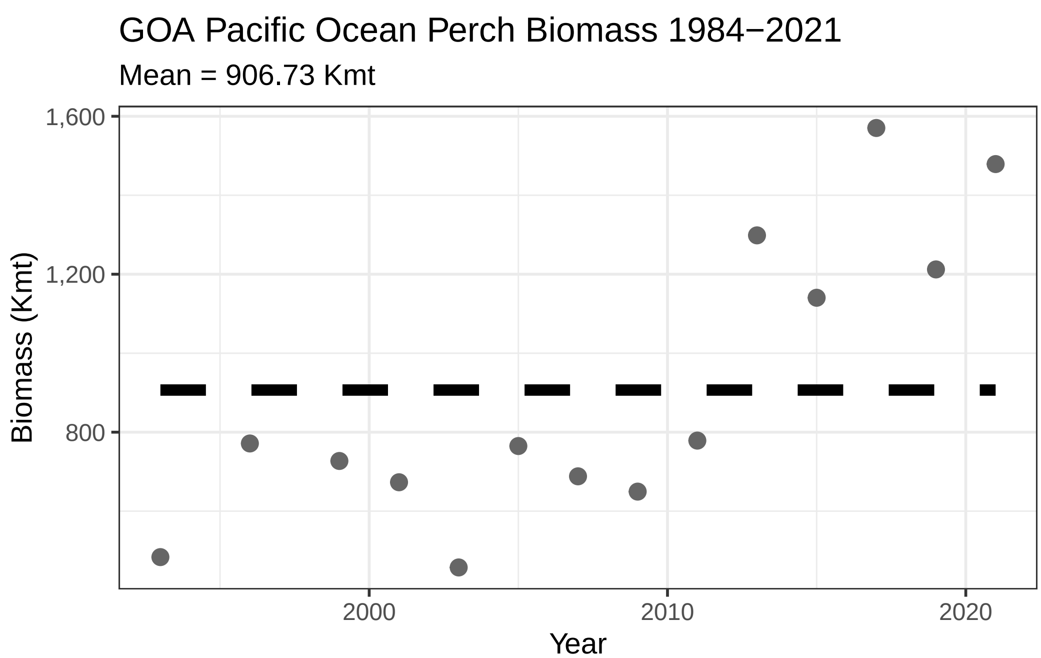


Figure 8.5.: Ex. 5: GOA Pacific Ocean perch biomass and line plot.

8.2.7. Ex. 6: EBS Pacific Ocean perch CPUE and akgfmaps map

Pacific Ocean perch catch-per-unit-effort estimates for EBS in 2021 from GAP_-PRODUCTS.AKFIN_CPUE and map constructed using akgfmaps. Here, we'll use AKFIN HAUL and CRUISES data also included in this repo, for convenience, though they are very similar to their RACEBASE analogs.

```
dat <- RODBC::sqlQuery(channel = channel,
                       query =
"SELECT
(cp.CPUE_KGKM2/100) CPUE_KGHA, -- akgfmaps is expecting hectares
hh.LATITUDE_DD_START LATITUDE,
hh.LONGITUDE_DD_START LONGITUDE

FROM GAP_PRODUCTS.AKFIN_CPUE cp

-- Use HAUL data to obtain LATITUDE & LONGITUDE and connect to cruisejoin
LEFT JOIN GAP_PRODUCTS.AKFIN_HAUL hh
ON cp.HAULJOIN = hh.HAULJOIN

-- Use CRUISES data to obtain YEAR and SURVEY_DEFINITION_ID
LEFT JOIN GAP_PRODUCTS.AKFIN_CRUISE cc
ON hh.CRUISEJOIN = cc.CRUISEJOIN

WHERE cp.SPECIES_CODE = 30060
AND cc.SURVEY_DEFINITION_ID = 98
AND cc.YEAR = 2021;")
```

```
flextable::flextable(head(dat)) %>% theme_zebra()
```

Table 8.6.: Ex. 6: EBS Pacific Ocean perch CPUE and akgfmaps map.

CPUE_-KGHA	LATITUDE	LONGITUDE
0.0000000	58.66802	-176.1673
0.00000000	60.69381	-175.4619
0.0000000	58.97738	-173.0898

Table 8.6.: Ex. 6: EBS Pacific Ocean perch CPUE and `akgfmaps` map.

CPUE_-KGHA	LATITUDE	LONGITUDE
0.00000000	61.68338	-173.6652
0.0000000	60.65295	-176.2033
0.03091028	59.97384	-176.7033

```
# devtools::install_github("afsc-gap-products/akgfmaps", build_vignettes = TRUE)
library(akgfmaps)

figure <- akgfmaps::make_idw_map(
  x = dat, # Pass data as a data frame
  region = "bs.south", # Predefined EBS area
  set.breaks = "jenks", # Gets Jenks breaks from classint::classIntervals()
  in.crs = "+proj=longlat", # Set input coordinate reference system
  out.crs = "EPSG:3338", # Set output coordinate reference system
  grid.cell = c(20000, 20000), # 20x20km grid
  key.title = "Pacific Ocean perch") # Include in the legend title
```

```
[inverse distance weighted interpolation]
[inverse distance weighted interpolation]
```

```
figure$plot +
  ggplot2::guides(fill=guide_legend(title = "Pacific Ocean perch\nCPUE (kg/km2)"))  |>
  change_fill_color(new.scheme = "grey", show.plot = FALSE)
```

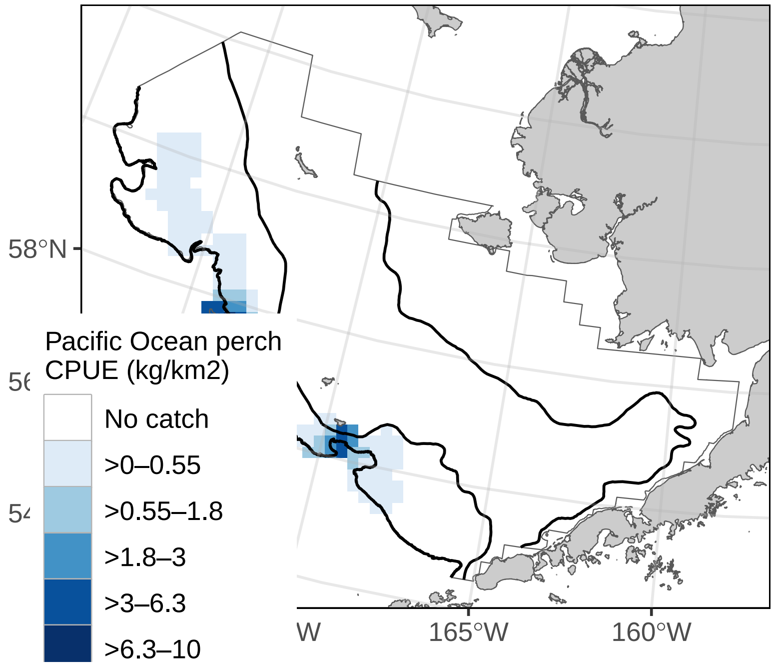


Figure 8.6.: Ex. 6: EBS Pacific Ocean perch CPUE and `akgfmaps` map.

Part IV.

Public Data (FOSS)

This data contains all of the catch, environmental, and haul data from the fisheries-independent Groundfish and Shellfish Assessment Program surveys in the Bering Sea, Aleutian Islands, and Gulf of Alaska. This data is sought after by the general public, private entities, and NOAA partners alike, including tribal organizations, K-12 classrooms, academic institutions, for-profit groups, and non-profit groups. This data is compiled and approved once a year after each summer survey season and is available for open access.

Part V.

Collaborators and data users

Below are a few packages and products currently using this data. If you have developed a product, performed an analysis, or exhibited this data in any way, reach out so we can showcase your hard work.

- **NOAA Fisheries Distribution Mapping and Analysis Portal**; *NOAA Fisheries Office of Science and Technology*
- **Pull data with python and explore the in-browser visualization tool. Reference their example Python notebook**; *The Eric and Wendy Schmidt Center for Data Science and the Environment at UC Berkeley, including sam.pottinger@berkeley.edu, ccmartinez@berkeley.edu, gzarpellon@berkeley.edu, and kkoy@berkeley.edu.*

Cite this data

Use the below bibtext citations, as cited in our group's citation repository for citing the data created and maintained in this repo (NOAA Fisheries Alaska Fisheries Science Center, 2023). Add "note = {Accessed: mm/dd/yyyy}" to append the day this data was accessed.

```
@misc{FOSSAFSCData,
  author = {{NOAA Fisheries Alaska Fisheries Science Center}},
  year = {2023},
  title = {Fisheries One Stop Shop Public Data: RACE Division Bottom Trawl Survey Data Quer
  howpublished = {https://www.fisheries.noaa.gov/foss},
  publisher = {{U.S. Dep. Commer.}},
  copyright = {Public Domain}
}
```

9. Data description

The Resource Assessment and Conservation Engineering Division (RACE) Groundfish Assessment Program (GAP) of the Alaska Fisheries Science Center (AFSC) conducts fisheries-independent bottom trawl surveys to monitor the condition of the demersal fish and crab stocks of Alaska. These data are developed to describe the temporal distribution and abundance of commercially and ecologically important groundfish species, examine the changes in the species composition of the fauna over time and space, and describe the physical environment of the groundfish habitat.

There are no legal restrictions on access to the data. They reside in the public domain and can be freely distributed. Users must read and fully comprehend the metadata prior to use. Data should not be used beyond the limits of the source scale. Acknowledgement of NOAA, as the source from which these data were obtained, in any publications and/or other representations of these data, is suggested. These data are compiled and approved annually after each summer survey season. The data from previous years are unlikely to change substantially once published.

These data are zero-filled (presence and absence) observations from surveys conducted on fishing vessels. These surveys monitor trends in distribution and abundance of groundfish, crab, and bottom-dwelling species in Alaska's marine ecosystems. These data include estimates of catch-per-unit-effort (CPUE) for all identified species for index stations. Some survey data are excluded, such as non-standard stations, surveys completed in earlier years using different/non-standard gear, and special tows and non-standard data collections.

Though not included in the public data, these surveys also collect oceanographic and environmental data, and biological data such as length, weight, stomach contents (to learn more about diet), otoliths (fish ear bones to learn about age), and tissue samples for genetic analysis, all of which can be shared upon special request. Also not included in the public data are estimated biomass (average total weight of all fish and crabs sampled) of crabs and groundfish that support the creation of annual stock assessments.

9.1. Data tables

9.1.1. FOSS_CATCH

Number of rows: 42281918

Number of columns: 12

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

COMMON_NAME

Taxon Common Name

text

VARCHAR2(255 BYTE)

The common name of the marine organism associated with the 'scientific_name' and 'species_code' columns. For a complete species list, review the code books.

COUNT

Taxon Count

count, whole number resolution

NUMBER(38,0)

Total number of individuals caught in haul by taxon, represented in whole numbers used in calculation.

CPUE_KGKM2

Weight CPUE (kg/km²)

kilograms per kilometers squared

NUMBER(38,6)

Catch weight (kilograms) divided by area (squared kilometers) swept by the net.

CPUE_NOKM2

Number CPUE (no/km²)

count per kilometers squared

NUMBER(38,6)

Catch number (in number of organisms) per area (squared kilometers) swept by the net.

HAULJOIN

Haul ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each (vessel, cruise, and haul) combination.

ID_RANK

Lowest taxonomic rank

text

VARCHAR2(255 BYTE)

Lowest taxonomic rank of a given species entry.

ITIS

ITIS Taxonomic Serial Number

ID code

NUMBER(38,0)

Species code as identified in the Integrated Taxonomic Information System (https://itis.gov/).

SCIENTIFIC_NAME

Taxon Scientific Name

text

VARCHAR2(255 BYTE)

The scientific name of the organism associated with the ‘common_name’ and ‘species_code’ columns. For a complete taxon list, review the code books.

SPECIES_CODE

Taxon Code

ID code

NUMBER(38,0)

The species code of the organism associated with the 'common_name' and 'scientific_-name' columns. For a complete species list, review the code books.

TAXON_CONFIDENCE

Taxon Confidence Rating

category

VARCHAR2(255 BYTE)

Confidence in the ability of the survey team to correctly identify the taxon to the specified level, based solely on identification skill (e.g., not likelihood of a taxon being caught at that station on a location-by-location basis). Quality codes follow: **'High'**: High confidence and consistency. Taxonomy is stable and reliable at this level, and field identification characteristics are well known and reliable. **'Moderate'**: Moderate confidence. Taxonomy may be questionable at this level, or field identification characteristics may be variable and difficult to assess consistently. **'Low'**: Low confidence. Taxonomy is incompletely known, or reliable field identification characteristics are unknown. Documentation: Species identification confidence in the eastern Bering Sea shelf survey (1982-2008), Species identification confidence in the eastern Bering Sea slope survey (1976-2010), and Species identification confidence in the Gulf of Alaska and Aleutian Islands surveys (1980-2011).

WEIGHT_KG

Taxon Weight (kg)

kilograms

NUMBER(38,3)

Weight (thousandths of a kilogram) of individuals in a haul by taxon.

WORMS

World Register of Marine Species Taxonomic Serial Number

ID code

NUMBER(38,0)

Species code as identified in the World Register of Marine Species (WoRMS) (https://www.marinespecies.org/).

9.1.2. FOSS_HAUL

Number of rows: 32510

Number of columns: 27

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_SWEPT_KM2

Area Swept (km)

kilometers

NUMBER(38,6)

The area the net covered while the net was fishing (kilometers squared), defined as the distance fished times the net width.

BOTTOM_TEMPERATURE_C

Bottom Temperature (Degrees Celsius)

degrees Celsius

NUMBER(38,1)

Bottom temperature (tenths of a degree Celsius); NA indicates removed or missing values.

CRUISE

Cruise ID

ID code

NUMBER(38,0)

This is a six-digit number identifying the cruise number of the form: YYYY99 (where YYYY = year of the cruise; 99 = 2-digit number and is sequential; 01 denotes the first cruise that vessel made in this year, 02 is the second, etc.).

CRUISEJOIN

Cruise ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each survey, vessel, and year combination.

DATE_TIME

Date and Time

MM/DD/YYYY HH::MM

DATE

The date (MM/DD/YYYY) and time (HH:MM) of the haul.

DEPTH_M

Depth (m)

degrees Celsius

NUMBER(38,1)

Bottom depth (tenths of a meter).

DISTANCE_FISHED_KM

Distance Fished (km)

degrees Celsius

NUMBER(38,3)

Distance the net fished (thousandths of kilometers).

DURATION_HR

Tow Duration (decimal hr)

hours

NUMBER(38,1)

This is the elapsed time between start and end of a haul (decimal hours).

HAUL

Haul Number

ID code

NUMBER(38,0)

This number uniquely identifies a sampling event (haul) within a cruise. It is a sequential number, in chronological order of occurrence.

HAULJOIN

Haul ID

ID code

NUMBER(38,0)

This is a unique numeric identifier assigned to each (vessel, cruise, and haul) combination.

LATITUDE_DD_END

End Latitude (decimal degrees)

decimal degrees

NUMBER(38,6)

Latitude (one hundred thousandth of a decimal degree) of the end of the haul.

LATITUDE_DD_START

Start Latitude (decimal degrees)

decimal degrees

NUMBER(38,6)

Latitude (one hundred thousandth of a decimal degree) of the start of the haul.

LONGITUDE_DD_END

End Longitude (decimal degrees)

decimal degrees

NUMBER(38,6)

Longitude (one hundred thousandth of a decimal degree) of the end of the haul.

LONGITUDE_DD_START

Start Longitude (decimal degrees)

decimal degrees

NUMBER(38,6)

Longitude (one hundred thousandth of a decimal degree) of the start of the haul.

NET_HEIGHT_M

Net Height (m)

meters

NUMBER(38,1)

Measured or estimated distance (meters) between footrope and headrope of the trawl.

NET_WIDTH_M

Net Width (m)

meters

NUMBER(38,1)

Measured or estimated distance (meters) between wingtips of the trawl.

PERFORMANCE

Haul Performance Code

category

NUMBER(38,0)

This denotes what, if any, issues arose during the haul. For more information, review the code books.

SRVY

Survey

text abbreviated

VARCHAR2(255 BYTE)

Abbreviated survey names. The column 'srvy' is associated with the 'survey' and 'survey_id' columns. Northern Bering Sea (NBS), Southeastern Bering Sea (EBS), Bering Sea Slope (BSS), Gulf of Alaska (GOA), Aleutian Islands (AI).

STATION

Station ID

ID code

VARCHAR2(255 BYTE)

Alpha-numeric designation for the station established in the design of a survey.

STRATUM

Stratum ID

ID code

NUMBER(10,0)

RACE database statistical area for analyzing data. Strata were designed using bathymetry and other geographic and habitat-related elements. The strata are unique to each survey series. Stratum of value 0 indicates experimental tows.

SURFACE_TEMPERATURE_C

Surface Temperature (Degrees Celsius)

degrees Celsius

NUMBER(38,1)

Surface temperature (tenths of a degree Celsius); NA indicates removed or missing values.

SURVEY

Survey Name

text

VARCHAR2(255 BYTE)

Name and description of survey. The column 'survey' is associated with the 'srvy' and 'survey_id' columns.

SURVEY_DEFINITION_ID

Survey ID

ID code

NUMBER(38,0)

This number uniquely identifies a survey. Name and description of survey. The column 'survey_id' is associated with the 'srvy' and 'survey' columns. For a complete list of surveys, review the code books.

SURVEY_NAME

NA

NA

NA

NA

VESSEL_ID

Vessel ID

ID code

NUMBER(38,0)

ID number of the vessel used to collect data for that haul. The column 'vessel_id' is associated with the 'vessel_name' column. Note that it is possible for a vessel to have a new name but the same vessel id number. For a complete list of vessel ID codes, review the code books.

VESSEL_NAME

Vessel Name

text

VARCHAR2(255 BYTE)

Name of the vessel used to collect data for that haul. The column 'vessel_name' is associated with the 'vessel_id' column. Note that it is possible for a vessel to have a new name but the same vessel id number. For a complete list of vessel ID codes, review the code books.

YEAR

Year

year

NUMBER(10,0)

Year the survey was conducted in.

10. Using the API

10.1. Select and filter

Select, filter, and package this and other NOAA Fisheries data from the Fisheries One Stop Shop (FOSS) platform. A user guide for the FOSS platform can be found here. To begin a report, select options from the boxes what you need data for.

For a given box, select one or a few options from the “options box” (list on the left) to query by highlighting them. To select multiple options, hold down the CTRL key while clicking on the options of interest, or click and drag down the list. Once the options you wish to be included in your query are highlighted, click the right-pointing arrow (>) to move them into the “selection box” (list on the right). If you accidentally select an option that you do not want to query, simply select the unwanted option from the selection box and click the left-pointing arrow (<).

If you wish to select all options from the options box and send them to the selection box, simply click the double right-pointing arrow (>>). If you want to unselect all options from the selection box, use the double left-pointing arrow (<<) or the reset icon.

To find a specific species or group more quickly you can use the `Search Species` option to quickly narrow the options. Search for parts of species common names in the `Search Species` box by entering a term and clicking the `search` button. The platform will return a shorter list in the `Speices` options box of only species that contain a match to that search term.

Use the `Reset All Parameters` button to reset all parameters for entire form.

Filter options:

- `Survey`: Each survey has different in design, time series, and history. More information on each survey and their designs can be found in our annual data reports.
- `Year`: Surveys are not conducted in all years, so only data from the years for which the survey was conducted will be returned.

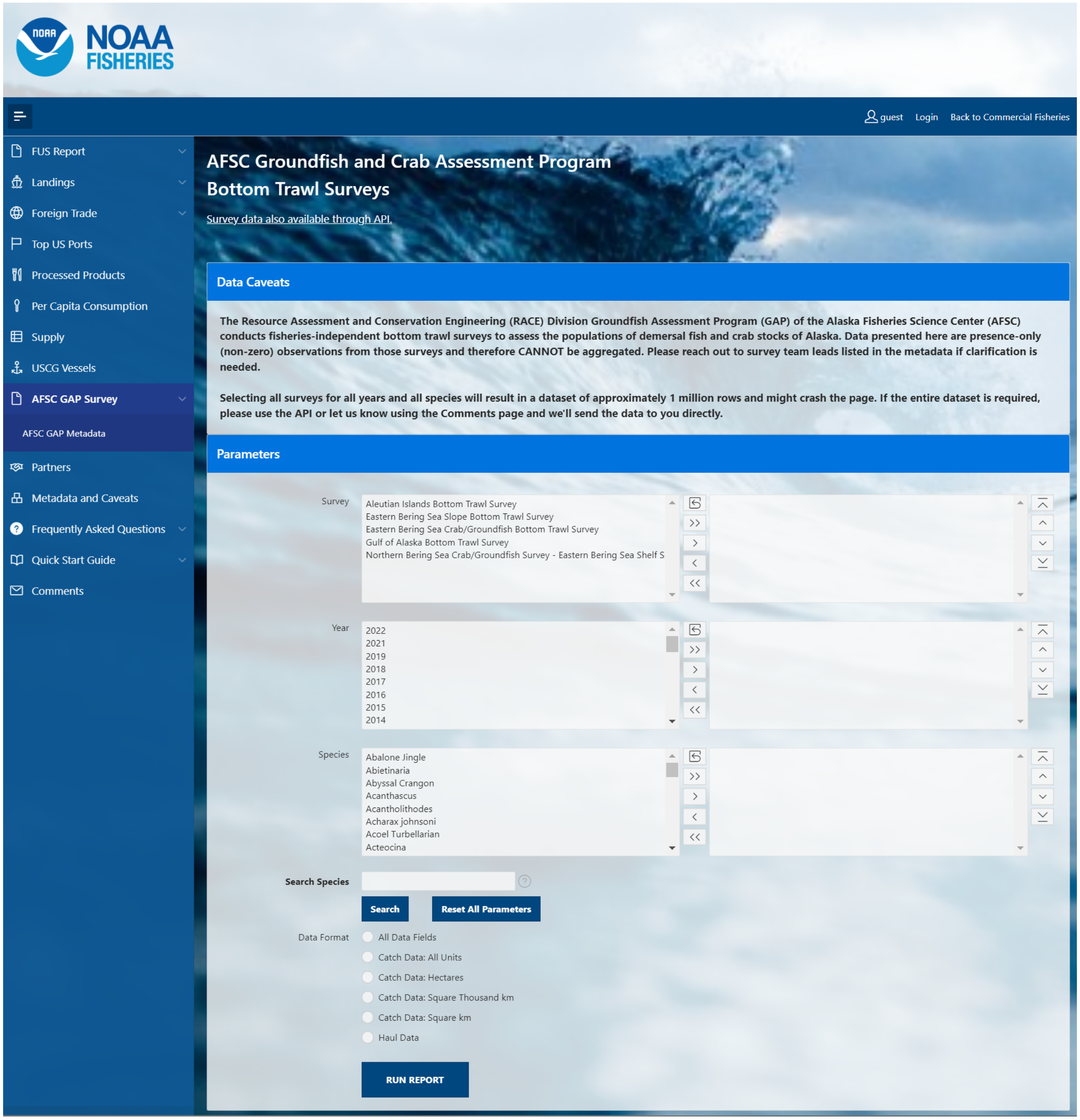


Figure 10.1.: AFSC Groundfish and Crab Assessment Program Bottom Trawl Survey data interface on the Fisheries One Stop Shop platform.

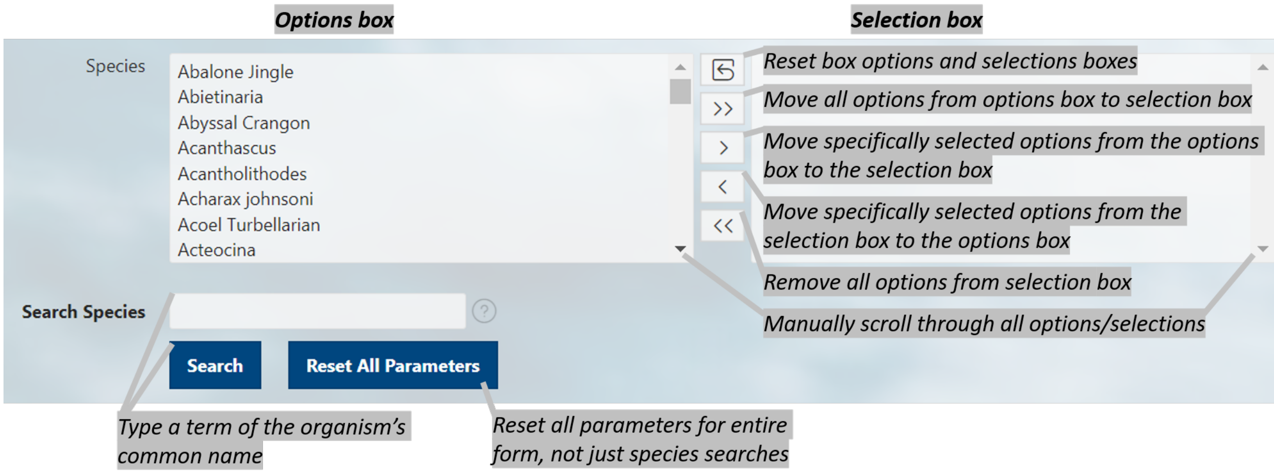


Figure 10.2.: Diagram of selection and search tools available on the FOSS platofrom.

- `Species`: Common name of all species ever encountered in the survey. Find more information about these species in our survey code books.

  In this example, we'll select for 2022 eastern Bering Sea Pacific cod data. Here, we used the `Search Species` box to search for species with the term "cod" in their common names and selected "Pacific cod" from that shortened list.

10.2. Select data format

Select from the below radio list of pre-designed output tables. Once you run the report, the user can further specify filter data and select columns of interest. The tables below will only include data from the selections made in the previous step.

- `All Data Fields: Presence and Absence (zero-filled)`: The most complete version of the data, including species, catch, haul, and environmental data. This data will include catch data for where species were caught and zeros for where the species were not caught. This is important for calculating catch-per-unit-effort data, preparing distribution plots (e.g., using the akgfmaps R package), and many statistical analyses.
- `All Data Fields: Presence-only (non-zero)`: The second most complete version of the data, including species, catch, haul, and environmental data. However, this data only includes catch data for where species were caught and does

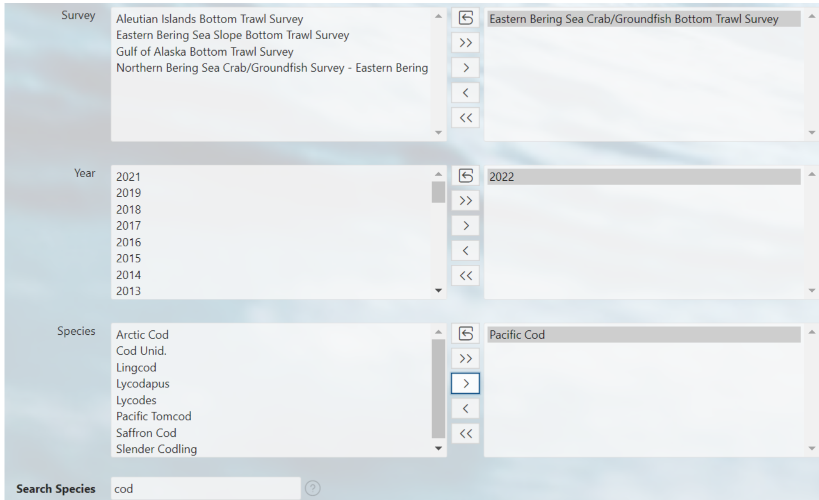


Figure 10.3.: Diagram of selection and search tools available on the FOSS platofrom.

not include zeros for where the species were not caught. This will return smaller, more focused data and can be useful for quickly assessing how many species were caught or how many stations species were caught at.

- `Catch data: Presence and Absence (zero-filled)`: This data set is similar to `All Data Fields: Presence and Absence (zero-filled)`, but only includes catch and species data columns.
- `Catch data: Presence-only (non-zero)`: This data set is similar to `All Data Fields: Presence-only (non-zero)`, but only includes catch and species data columns.
- `Haul Data`: This data set only includes haul and environmental data collected from the survey. This data will only include one observation per haul event/station.

In this example, we'll select `All Data Fields: Presence and Absence (zero-filled)`.

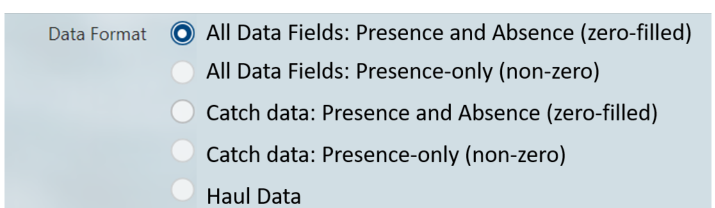


Figure 10.4.: Diagram of the pre-set data format options.

10.3. Run report

Click the `RUN REPORT` button. Below the select and filter area, the results of your query will appear below the page in the format you selected. To change the format, make a different selection and run the report again. Further modifications to your results can be made by clicking on the `Actions` button above your data. Here you can `download` your data, `select columns` included in your results, and apply a variety of `filters` and mathematical tools.

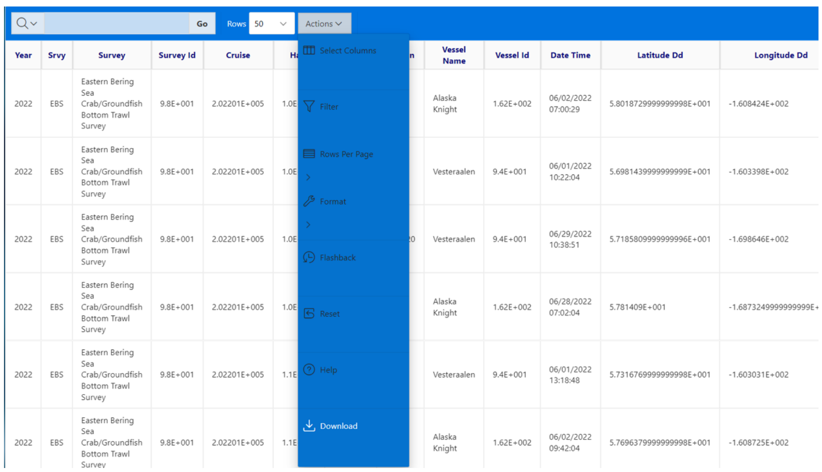


Figure 10.5.: Example data returned from running the report.

11. Access API data using R

An application programming interface (API) is a way for two or more computer programs to communicate with each other.

More information about how to amend API links can be found here. Useful introductions to using APIs in `R` can be found here.

11.1. Ex. 1: Load the first 25 rows (default) of data

```
# install.packages(c("httr", "jsonlite"))
library(httr)
library(jsonlite)
library(dplyr)
# link to the API
api_link <- "https://apps-st.fisheries.noaa.gov/ods/foss/afsc_groundfish_survey/"

res <- httr::GET(url = api_link)
# res # Test connection
data <- jsonlite::fromJSON(base::rawToChar(res$content))
# names(data)
flextable::flextable(head(data$items, 3))
```

yearsrvy	survey	survey_id	cruise	haul	stratum	station	vessel_-name	vessel_
2,002AI	Aleutian Islands Bottom Trawl Survey	5.2E+001	2.00201E+005	6.0E+000	7.22E+002	307-63	Vesteraalen	9.4E+0
2,002AI	Aleutian Islands Bottom Trawl Survey	5.2E+001	2.00201E+005	6.0E+000	7.22E+002	307-63	Vesteraalen	9.4E+0
2,002AI	Aleutian Islands Bottom Trawl Survey	5.2E+001	2.00201E+005	6.0E+000	7.22E+002	307-63	Vesteraalen	9.4E+0

11.2. Ex. 2: Load the first 10000 rows of data

```
# Not run because too big:
res <- httr::GET(url = paste0(api_link, "?offset=0&limit=10000"))
data <- jsonlite::fromJSON(base::rawToChar(res$content))
print(paste0("rows: ", dim(data$items)[1], "; cols: ", dim(data$items)[2]))
```

```
[1] "rows: 10000; cols: 36"
```

11.3. Ex. 3: Filter by Year

Show all the data greater than the year 2020.

```
res <- httr::GET(url = paste0(api_link, '?q={"year":{"$gt":2020}}'))
data <- jsonlite::fromJSON(base::rawToChar(res$content))
flextable::flextable(
  data$items[1:3, c("year", "srvy", "stratum", "species_code", "cpue_kgkm2")]) %>%
  flextable::theme_zebra()
```

Table 11.2.: Ex. 3: Filter by Year.

yearsrvy	stratum	species_code	cpue_kgkm2
2,022AI	7.22E+002	1.0261E+0	6.7332582200000002E+002
2,022AI	7.93E+002	8.054E+004	3.6112E-001
2,022AI	7.22E+002	2.1347E+0	7.5809130500000003E+002

11.4. Ex. 4: Filter by species name

Show all the data where the product name contains pollock Please note that here the word pollock is case sensitive.

The notation for finding a string is to use % around it. Since % is a reserved character in a URL, you have to replace `%` with `%25`.

```
res <- httr::GET(
  url = paste0(api_link, '?q={"common_name":{"$like":"%25pollock%25"}}'))
data <- jsonlite::fromJSON(base::rawToChar(res$content))
flextable::flextable(
  data$items[1:3, c("year", "srvy", "stratum", "species_code", "cpue_kgkm2")]) %>%
  flextable::theme_zebra()
```

Table 11.3.: Ex. 4: Filter by species name.

yearsrvy	stratum	species_-code	cpue_-kgkm2
2,002AI	7.21E+002	2.174E+004	6.3989099999999999E-001
2,002AI	7.22E+002	2.174E+004	7.7532226400000002E+002
2,002AI	7.22E+002	2.174E+004	1.0685806397E+004

11.5. Ex. 5: Combination of year and name filters

Show all the data where years > 2020 and the product name contains pollock

```
res <- httr::GET(
  url = paste0(api_link,
               '?q={"year":{"$gt":2020},"common_name":{"$like":"%25pollock%25"}}'))
data <- jsonlite::fromJSON(base::rawToChar(res$content))
flextable::flextable(
  data$items[1:3, c("year", "srvy", "stratum", "species_code", "cpue_kgkm2")]) %>%
  flextable::theme_zebra()
```

Table 11.4.: Ex. 5: Combination of year and name filters.

yearsrvy	stratum	species_-code	cpue_-kgkm2
2,022AI	7.22E+002	2.174E+004	2.2754334435000001E+004
2,022AI	7.93E+002	2.174E+004	7.8536315350000004E+003
2,022AI	7.21E+002	2.174E+004	7.2350103259999996E+003

11.6. Ex. 6: Combination of year, srvy, stratum

Show all the data where year = 1989, srvy = “EBS”, and stratum is not equal to 81

```
res <- httr::GET(
  url = paste0(api_link, '?q={"year":1989,"srvy":"EBS","stratum":{"$ne":"81"}}'))
data <- jsonlite::fromJSON(base::rawToChar(res$content))
flextable::flextable(
  data$items[1:3, c("year", "srvy", "stratum", "species_code", "cpue_kgkm2")]) %>%
  flextable::theme_zebra()
```

Table 11.5.: Ex. 6: Combination of year, srvy, stratum.

yearsrvy	stratum	species_code	cpue_kgkm2
1,989EBS	1.0E+001	4.05E+004	9.6200360000000007E+000
1,989EBS	1.0E+001	6.8578E+00	9.6200360000000007E+000
1,989EBS	1.0E+001	2.1313E+0	1.8179039E+001

11.7. Ex. 7: Visualize CPUE data in distribution map

Pacific cod catch-per-unit-effort estimates for NBS in 2021 and map constructed using `akgfmaps`.

```
# res <- httr::GET(
#   url = paste0(api_link, "?offset=0&limit=10000"),
#   query = list(year = 2021, srvy = "EBS", species_code = 30060))
res <- httr::GET(
  url = paste0(api_link, '?q={"year":2021,"srvy":"NBS","species_code":21720}'))
data_catch <- jsonlite::fromJSON(base::rawToChar(res$content))$items %>%
  dplyr::select(stratum, station, cpue_kgkm2)

# zero-fill data (imperfectly, but effective for this example)
res <- httr::GET(
  url = paste0(api_link, '?q={"year":2021,"srvy":"NBS"}offset=0&limit=10000'))
data_haul <- jsonlite::fromJSON(base::rawToChar(res$content))$items %>%
  dplyr::select(stratum, station, latitude_dd, longitude_dd) %>%
  dplyr::distinct()

data <- dplyr::left_join(data_haul, data_catch) %>%
```

```r
  dplyr::mutate(cpue_kgkm2 = ifelse(is.na(cpue_kgkm2), 0, cpue_kgkm2),
                dplyr::across(dplyr::everything(), as.numeric))

flextable::flextable(data[1:3,]) %>%
  flextable::theme_zebra()
```

Table 11.6.: Ex. 7: Visualize CPUE data in distribution map.

stratum	station	latitude_dd	longitude_dd	cpue_kgkm2
71		63.70028	-171.0225	1.183039
81		61.68600	-173.0776	13,256.716473
81		61.34965	-172.2251	600.958261

```r
# devtools::install_github("afsc-gap-products/akgfmaps", build_vignettes = TRUE)
library(akgfmaps)

figure <- akgfmaps::make_idw_map(
  CPUE_KGHA = data$cpue_kgkm2, # calculates the same, regardless of units.
  LATITUDE = data$latitude_dd,
  LONGITUDE = data$longitude_dd,
  region = "bs.north", # Predefined EBS area
  set.breaks = "jenks", # Gets Jenks breaks from classint::classIntervals()
  in.crs = "+proj=longlat", # Set input coordinate reference system
  out.crs = "EPSG:3338", # Set output coordinate reference system
  grid.cell = c(20000, 20000), # 20x20km grid
  key.title = "Pacific Ocean perch") # Include in the legend title
```

```
[inverse distance weighted interpolation]
[inverse distance weighted interpolation]
```

```r
figure$plot +
  ggplot2::guides(fill=guide_legend(title = "Pacific cod\nCPUE (kg/km2)"))
```

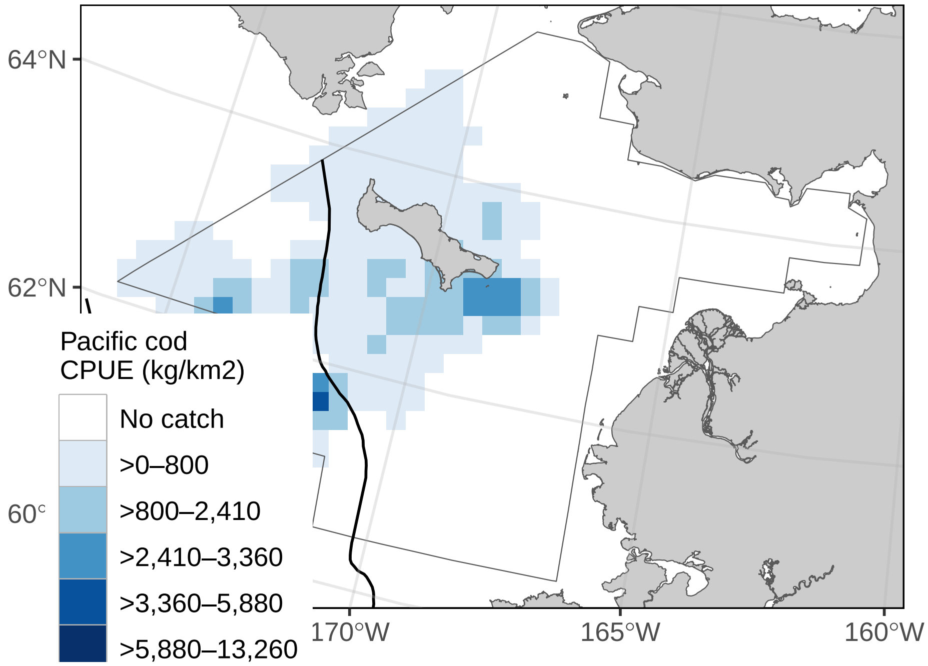


Figure 11.1.: Ex. 7: Visualize CPUE data in distribution map.

12. Access API data using Python

12.0.1. {afscgap} Library Installation

> author: Sam Pottinger (sam.pottinger@berkeley.edu; GitHub::sampottinger)
> date: May 13, 2023

The third-party `afscgap` Python package interfaces with FOSS to access AFSC GAP data. It can be installed via pip:

```
#The reticulate package provides a comprehensive set of tools for interoperability between
library(reticulate)
```

```
pip install afscgap
pip install git+https://github.com/SchmidtDSE/afscgap.git@main
```

For more information on installation and deployment, see the library documentation.

12.0.2. Basic query

This first example queries for Pacific glass shrimp (*Pasiphaea pacifica*) in the Gulf of Alaska in 2021. The library will automatically generate HTTP queries, converting from Python types to ORDS query syntax.

```
import afscgap

query = afscgap.Query()
query.filter_year(eq=2021)
query.filter_srvy(eq='GOA')
query.filter_scientific_name(eq='Pasiphaea pacifica')

results = query.execute()
```

The `results` variable in this example is an iterator that will automatically perform pagination behind the scenes.

12.0.3. Iterating with a for loop

The easiest way to interact with results is a simple for loop. This next example determines the frequency of different catch per unit effort where Pacific glass shrimp were reported:

```
import afscgap

# Mapping from CPUE to count
count_by_cpue = {}

# Build query
query = afscgap.Query()
query.filter_year(eq=2021)
query.filter_srvy(eq='GOA')
query.filter_scientific_name(eq='Pasiphaea pacifica')
results = query.execute()

# Iterate through results and count
for record in results:
  cpue = record.get_cpue_weight(units='kg/ha')
  cpue_rounded = round(cpue)
  count = count_by_cpue.get(cpue_rounded, 0) + 1
  count_by_cpue[cpue_rounded] = count

# Print the result
print(count_by_cpue)
```

Note that, in this example, only records with Pacific glass shrimp are included ("presence-only" data). See zero catch inference below. In other words, it reports on CPUE only for hauls in which Pacific glass shrimp were recorded, excluding some hauls like those in which Pacific glass shrimp were not found at all.

12.0.4. Iterating with functional programming

A for loop is not the only option for iterating through results. List comprehensions and other functional programming methods can be used as well.

```
import statistics

import afscgap

# Build query
query = afscgap.Query()
query.filter_year(eq=2021)
query.filter_srvy(eq='GOA')
query.filter_scientific_name(eq='Pasiphaea pacifica')
results = query.execute()

# Get temperatures in Celsius
temperatures = [record.get_bottom_temperature(units='c') for record in results]

# Take the median
print(statistics.median(temperatures))
```

This example reports the median temperature in Celcius for when Pacific glass shrimp was reported.

12.0.5. Load into Pandas

The results from the `afscgap` package are serializable and can be loaded into other tools like Pandas. This example loads Pacific glass shrimp from 2021 Gulf of Alaska into a data frame.

```
import pandas

import afscgap

query = afscgap.Query()
query.filter_year(eq=2021)
query.filter_srvy(eq='GOA')
query.filter_scientific_name(eq='Pasiphaea pacifica')
```

```
results = query.execute()

pandas.DataFrame(results.to_dicts())
```

Specifically, `to_dicts` provides an iterator over a dictionary form of the data that can be read into tools like Pandas.

12.0.6. Advanced filtering

Queries so far have focused on filters requiring equality but range queries can be built as well.

```
import afscgap

# Build query
query = afscgap.Query()
query.filter_year(min_val=2015, max_val=2019)   # Note min/max_val
query.filter_srvy(eq='GOA')
query.filter_scientific_name(eq='Pasiphaea pacifica')
results = query.execute()

# Sum weight
weights = map(lambda x: x.get_weight(units='kg'), results)
total_weight = sum(weights)
print(total_weight)
```

This example queries for Pacific glass shrimp data between 2015 and 2019, summing the total weight caught. Note that most users will likely take advantage of built-in Python to ORDS query generation which dictates how the library communicates with the API service. However, users can provide raw ORDS queries as well using manual filtering.

12.0.7. Zero-catch inference

Until this point, these examples use presence-only data. However, the `afscgap` package can infer negative or "zero catch" records as well.

```
import afscgap

# Mapping from CPUE to count
count_by_cpue = {}

# Build query
query = afscgap.Query()
query.filter_year(eq=2021)
query.filter_srvy(eq='GOA')
query.filter_scientific_name(eq='Pasiphaea pacifica')
query.set_presence_only(False)  # Added to earlier example
results = query.execute()

# Iterate through results and count
for record in results:
  cpue = record.get_cpue_weight(units='kg/ha')
  cpue_rounded = round(cpue)
  count = count_by_cpue.get(cpue_rounded, 0) + 1
  count_by_cpue[cpue_rounded] = count

# Print the result
print(count_by_cpue)
```

This example revisits the earlier snippet for CPUE counts but `set_presence_only(False)` directs the library to look at additional data on hauls, determining which hauls did not have Pacific glass shrimp. This lets the library return records for hauls in which Pacific glass shrimp were not found. This can be seen in differences in counts reported:

Rounded CPUE	Count with set_presence_only(True)	Count with set_presence_only(False)
0 kg/ha	44	521
1 kg/ha	7	7
2 kg/ha	1	1

Put simply, while the earlier example showed CPUE counts for hauls in which Pacific glass shrimp were seen, this revised example reports for all hauls in the Gulf of Alaska in 2021.

12.0.8. More information

Please see the API documentation for the Python library for additional details.

13. Access data using R (AFSC only)

If the user has access to the AFSC `Oracle` database, the user can use `SQL developer` to view and pull the FOSS public data directly from the `RACEBASE_FOSS` `Oracle` schema.

13.0.1. Connect to Oracle from R

Many users will want to access the data from `Oracle` using `R`. The user will need to install the `RODBC` `R` package and ask OFIS (IT) connect `R` to `Oracle`. Then, use the following code in `R` to establish a connection from `R` to `Oracle`:

Here, the user can write in their username and password directly into the `RODBC` connect function. Never save usernames or passwords in scripts that may be intentionally or unintentionally shared with others. If no username and password is entered in the function, pop-ups will appear on the screen asking for the username and password.

```
#' Define RODBC connection to ORACLE
#'
#' @param schema default = 'AFSC'.
#'
#' @return oracle channel connection
#' @export
#'
#' @examples
#' # Not run
#' # channel <- oracle_connect()
oracle_connect <- function(
    schema='AFSC',
    username = NULL,
    passowrd = NULL){(echo=FALSE)

  library("RODBC")
  library("getPass")
  if (is.null(username)) {
```

```
    username <- getPass(msg = "Enter your ORACLE Username: ")
  }
  if (is.null(password)) {
    password <- getPass(msg = "Enter your ORACLE Password: ")
  }
  channel  <- RODBC::odbcConnect(
    paste(schema),
    paste(username),
    paste(password),
    believeNRows=FALSE)
  return(channel)
}

channel <- oracle_connect()
```

13.0.2. Ex. 1: Join data

To join these tables in Oracle, you may use a variant of the following code:

13.0.3. Ex. 2: Subset data

Once connected, pull and save (if needed) the tables into the `R` environment.

To pull a small subset of the data (especially since files like `RACEBASE_FOSS.FOSS_CPUE_ZEROFILLED` are so big), use a variation of the following code. Here, we are pulling EBS Pacific cod from 2010 - 2021:

```
# Pull data
a <- RODBC::sqlQuery(
channel = channel,
query =
"SELECT * FROM GAP_PRODUCTS.FOSS_CATCH cc
JOIN GAP_PRODUCTS.FOSS_HAUL hh
ON cc.HAULJOIN = hh.HAULJOIN
WHERE SRVY = 'EBS'
AND COMMON_NAME = 'Pacific cod'
AND YEAR >= 2010
AND YEAR < 2021")
```

```
# Save table to local directory
write.csv(x = a, file = "RACEBASE_FOSS-FOSS_CPUE_ZEROFILLED-ebs_pcod_2010-2020.csv")
```

Part VI.

Notes

Thank you for using our data guide!

14. Production Run Notes

15. R Version Metadata

```
R version 4.3.0 (2023-04-21 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

Matrix products: default


locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.3.0    fastmap_1.1.1     cli_3.6.1         tools_4.3.0
 [5] htmltools_0.5.5   rstudioapi_0.15.0 yaml_2.3.7        rmarkdown_2.23
 [9] knitr_1.43        jsonlite_1.8.7    xfun_0.39         digest_0.6.33
[13] rlang_1.1.1       evaluate_0.21
```

15.0.1. NOAA README

This repository is a scientific product and is not official communication of the National Oceanic and Atmospheric Administration, or the United States Department of Commerce. All NOAA GitHub project code is provided on an ‘as is’ basis and the user assumes responsibility for its use. Any claims against the Department of Commerce or

15.0.2. NOAA License

16. Acknowledgments

17. Community Acknowledgments

We would like to thank the many communities of Alaska and their members who have helped contribute to this body of work. The knowledge, experiences, and insights have been instrumental in expanding the scope of our science and knowledge to encompass the many issues that face this important ecosystem. We appreciate feedback from those residing in the region that are willing to share their insights and participation in an open dialog about how we can improve our collective knowledge of the ecosystem and the region.

18. Technical Acknowledgments

This quarto book is based off the NOAA-quarto-book GitHub repo designed by Eli Holmes.

This repo and GitHub Action was based on the tutorial by Openscapes quarto-website-tutorial by Julia Lowndes and Stefanie Butland.

18.1. Partners

Scientists from the Alaska Fisheries Science Center conduct these bottom trawl surveys with participation from the Alaska Department of Fish & Game (ADF&G), the International Pacific Halibut Commission (IPHC), and universities. This research is conducted on chartered fishing vessels.

19. References

Alaska Fisheries Information Network (AKFIN). (2023). *AFSC goundfish assessment program design-based production data*. NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program; https://www.psmfc.org/program/alaska-fisheries-information-network-akfin; U.S. Dep. Commer.

Hoff, G. R. (2016). *Results of the 2016 eastern Bering Sea upper continental slope survey of groundfishes and invertebrate resources* (NOAA Tech. Memo. NOAA-AFSC-339). U.S. Dep. Commer. https://doi.org/10.7289/V5/TM-AFSC-339

Markowitz, E. H., Dawson, E. J., Anderson, A. B., Rohan, S. K., Charriere, N. E., Prohaska, B. K., and Stevenson, D. E. (2023). *Results of the 2022 eastern and northern Bering Sea continental shelf bottom trawl survey of groundfish and invertebrate fauna* (NOAA Tech. Memo. NMFS-AFSC-469; p. 213). U.S. Dep. Commer.

NOAA Fisheries Alaska Fisheries Science Center. (2023). *Fisheries one stop shop public data: RACE division bottom trawl survey data query*. https://www.fisheries.noaa.gov/foss; U.S. Dep. Commer.

NOAA Fisheries Alaska Fisheries Science Center, Goundfish Assessment Program. (2023). *AFSC goundfish assessment program design-based production data*. https://www.fisheries.noaa.gov/alaska/science-data/groundfish-assessment-program-bottom-trawl-surveys; U.S. Dep. Commer.

Von Szalay, P. G., and Raring, N. W. (2018). *Data report: 2017 Gulf of Alaska bottom trawl survey* (NOAA Tech. Memo. NMFS-AFSC-374). U.S. Dep. Commer. https://doi.org/10.7289/V5/TM-AFSC-374

Von Szalay, P. G., and Raring, N. W. (2020). *Data report: 2018 Aleutian Islands bottom trawl survey* (NOAA Tech. Memo. NMFS-AFSC-409). U.S. Dep. Commer. https://doi.org/10.25923/qe5v-fz70

20. Contact us

General questions and more specific data requests can be sent to afsc.gap.metadata@noaa.gov or submitted as an issue on our GitHub Organization. The version of this data used for stock assessments can be found through the Alaska Fisheries Information Network (AKFIN). For questions about the eastern Bering Sea surveys, contact Duane Stevenson (Duane.Stevenson@noaa.gov). For questions about the Gulf of Alaska or Aleutian Islands surveys, contact Ned Laman (Ned.Laman@noaa.gov). For questions specifically about crab data in any region, contact Mike Litzow (Mike.Litzow@noaa.gov), the Shellfish Assessment Program lead.

For questions, comments, and concerns specifically about the Fisheries One Stop Shop (FOSS) platform, please contact us using the Comments page on the FOSS webpage.

Alaska Fisheries Science Center (AFSC)
National Oceanic and Atmospheric Administration (NOAA)
Resource Assessment and Conservation Engineering Division (RACE)
Groundfish Assessment Program (GAP)
7600 Sand Point Way, N.E. bldg. 4
Seattle, WA 98115 USA

20.1. Suggestions and comments

If the data or metadata can be improved, please create a pull request, submit an issue to the GitHub organization or submit an issue to the code's repository.