



NOAA Technical Memorandum NMFS-XXX-##

GAP Production Data Documentation

Bering Sea Survey Team, Gulf of Alaska and Aleutian Island Survey Team

U.S. DEPARTMENT OF COMMERCE

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



**NOAA
FISHERIES**

GAP Production Data Documentation

Bering Sea Survey Team^{1,*} and Gulf of Alaska and Aleutian Island Survey Team^{1,*}

1. NOAA Fisheries Alaska Fisheries Science Center, Groundfish Assessment Program

* Correspondence: Bering Sea Survey Team afsc.gap.metadata@noaa.gov

* Correspondence: Gulf of Alaska and Aleutian Island Survey Team afsc.gap.metadata@noaa.gov

Table of contents

I. Welcome	1
Our Objective	2
User Resources	2
Background of the gap_products repo	2
Major Advantages	4
1. Survey background	6
1.1. What is the research objective?	6
1.2. Who is conducting the research?	6
1.3. Bottom trawl surveys and regions	7
2. Workflow	12
2.1. Data workflow from boat to production	12
2.2. Organization	12
2.3. Data levels	13
3. News	15
3.1. Future plans	15
3.2. Previous updates	15
II. GAP Production Data	17
Data Description	18
Cite this data	18
4. Data description	19
4.1. Data tables	19

Table of contents

III. AKFIN	35
The Alaska Fisheries Information Network	36
Cite this data	36
5. Data description	37
5.1. Data description	37
5.2. Data tables	37
6. Access data	71
6.1. Access data via Oracle (AFSC only)	71
6.2. Data SQL Query Examples:	71
7. Access API data using R	88
7.1. Ex. 1: Load lingcod data	88
IV. Public Data (FOSS)	90
V. Collaborators and data users	92
Cite this data	93
8. Data description	94
8.1. Data tables	95
9. Using the FOSS platform	113
9.1. Select and filter	113
9.2. Select data format	115
9.3. Run report	117
10. Access via API and R	118
10.1.Ex. 1: Load the first 25 rows (default) of data	118
10.2.Ex. 2: Load the first 10000 rows of data	119
10.3.Ex. 3: Filter by Year	119
10.4.Ex. 4: Filter by species name	120
10.5.Ex. 5: Combination of year and name filters	121
10.6.Ex. 6: Combination of year, srvy, stratum	122
10.7.Ex. 7: Visualize CPUE data in distribution map	123
11. Access via API and Python	126
12. Access via Oracle and R (AFSC only)	132

Table of contents

VI. Data Products	135
13. Open source code	137
13.1.R Packages	137
14. Reports	138
14.1.Annual Publications	138
14.2.NOAA Fisheries Distribution Mapping and Analysis Portal (DisMAP) . .	139
15. Outreach	140
15.1.Real-time temperatures and survey progress	140
VII. Contact us	141
Suggestions and comments	142
16. Production run notes	143
17. R Version Metadata	144
18. Data constraints	146
18.1.Cite this data	146
19. Access Constraints	147
20. Acknowledgments	148
21. Community Acknowledgments	149
22. Technical Acknowledgments	150
22.1.Partners	150
22.2.Collaborators	150
23. References	151

List of Figures

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.	3
1.1. Strata used in the Eastern Bering Sea Survey.	10
1.2. Strata used in the Bering Sea Slope Survey.	11
6.1. Ex. 1: GOA Pacific Ocean perch biomass and abundance.	75
6.2. Ex. 2: AI Rock sole size compositions and ridge plot.	77
6.3. Ex. 3: EBS Walleye Pollock Age Compositions and Age Pyramid.	79
6.4. Ex. 4: NBS Pacific cod biomass and abundance.	82
6.5. Ex. 5: GOA Pacific Ocean perch biomass and line plot.	85
6.6. Ex. 6: EBS Pacific Ocean perch CPUE and akgfmaps map.	87
9.1. AFSC Groundfish and Crab Assessment Program Bottom Trawl Survey data interface on the Fisheries One Stop Shop platform.	114
9.2. Diagram of selection and search tools available on the FOSS platofrom.	115
9.3. Diagram of selection and search tools available on the FOSS platofrom.	116
9.4. Diagram of the pre-set data format options.	116
9.5. Example data returned from running the report.	117
10.1.Ex. 7: Visualize CPUE data in distribution map.	125

List of Tables

1.1. Survey summary stats	7
6.1. Ex. 1: GOA Pacific Ocean perch biomass and abundance.	73
6.2. Ex. 2: AI Rock sole size compositions and ridge plot.	76
6.3. Ex. 3: EBS Walleye Pollock Age Compositions and Age Pyramid.	78
6.4. Ex. 4: NBS Pacific cod biomass and abundance.	81
6.5. Ex. 5: GOA Pacific Ocean perch biomass and line plot.	83
6.6. Ex. 6: EBS Pacific Ocean perch CPUE and akgfmaps map.	86
10.1.Ex. 1: Load the first 25 rows (default) of data.	119
10.2.Ex. 3: Filter by Year.	120
10.3.Ex. 4: Filter by species name.	121
10.4.Ex. 5: Combination of year and name filters.	122
10.5.Ex. 6: Combination of year, srvy, stratum.	122
10.6.Ex. 7: Visualize CPUE data in distribution map.	123

Part I.

Welcome

Our Objective

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.

At this time, these master production and AKFIN tables are **provisional and we are welcoming feedback before the 2024 survey season**. We look forward to hearing from you. Do not hesitate to reach out (to us at either afsc.gap.metadata@noaa.gov or GitHub issues), especially if you find discrepancies in the data or want to suggest improvements to infrastructure. Thank you in advance for your collaboration and partnership with us as we develop our future data universe.

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

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.

Background of the gap_products repo



Figure 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.

Major Advantages

- (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 Restratiification 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 restratiified 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.

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.

Major Advantages

- 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. Survey background

1.1. 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

1.2. Who is conducting the research?

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.

1. Survey background

1.3. Bottom trawl surveys and regions

Bottom Trawl Survey Regions

AFSC RACE Groundfish and Shellfish Public Data Coverage



Each survey conducted by the Groundfish Assessment Program are multispecies bottom trawl surveys. We collect environmental and biological data to assess how climate variability and loss of sea ice are affecting bottom-dwelling marine life on the Bering Sea shelf. We monitor trends in the distribution (location and movement patterns) and abundance of groundfish and crab species as well as oceanographic data (e.g., water temperature, depth). We collect biological information such as organism weight, length, stomachs to learn about diets, and otoliths to determine fish ages. We use this information in annual stock assessments and to assess the state of the ecosystem. This research is conducted on fishing industry contract vessels.

Table 1.1.: Survey summary stats

Survey	Survey Definition ID	Years	Depth (m)	Area (km ²)	# Statistical Areas	# Possible Stations
Aleutian Islands Bottom Trawl Survey	52	2022 - 1980 (16)	1 - 500	64,415.0	80	1,312
Eastern Bering Sea Slope Bottom Trawl Survey	78	2016 - 2002 (6)	201 - 800	21,134.2	4	

1. Survey background

Survey	Survey Definition ID	Years	Depth (m)	Area (km2)	# Statistical Areas	# Possible Stations
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	2023 - 2010 (6)	1 - 100	198,866	4	144

1.3.1. Aleutian Islands

(Von Szalay and Raring, 2020)

- Upper Continental Slope of the Aleutian Islands from Unimak Pass to Stalemate Bank
- Triennial (1990s)/Biennial since 2000 in even years, since 1992
- Modified Index-Stratified Random of Successful Stations Survey Design
- Important commercial fish species include Atka mackerel, Pacific ocean perch, walleye pollock, Pacific cod, sablefish, and other rockfish species.

1.3.2. Gulf of Alaska

(Von Szalay and Raring, 2018)

- Continental Shelf and Upper Slope of the Gulf of Alaska extending from the Islands of Four Mountains 2,300 km east to Dixon Entrance
- Triennial (1990s)/Biennial since 2001 in odd years, since 1991
- Stratified Random Survey Design
- Important commercial species in the Gulf of Alaska include Pacific ocean perch, walleye pollock, Pacific cod, flatfish, and other rockfish species.

1. Survey background

1.3.3. Eastern Bering Sea Shelf

(Markowitz et al., 2023)

- The continental shelf of the eastern Bering Sea from the Aleutian Islands to the Bering Strait
- Conducted annually since 1982.
- Uses a stratified systematic sampling survey design with fixed stations at center of 20 x 20 nm grid.
- Similar in design to the northern Bering Sea shelf bottom trawl survey.
- Focus species for the Bering Sea include walleye pollock, Pacific cod, Greenland turbot, yellowfin sole, northern rock sole, red king crab, and snow and Tanner crabs.

1.3.4. Northern Bering Sea

(Markowitz et al., 2023)

- The continental shelf of the northern Bering Sea, including the area north of St. Lawrence Island and Norton Sound
- Biennial/Annual; conducted intermittently since 2010
- Uses a stratified systematic sampling survey design with fixed stations at center of 20 x 20 nm grid.
- Similar in design to the eastern Bering Sea shelf bottom trawl survey.

1.3.5. Eastern Bering Sea Upper Continental Slope

(Hoff, 2016)

- The eastern Bering Sea upper continental slope survey area extends from Unalaska and Akutan Islands to the U.S.-Russian Maritime Boundary at 61° N near the International Date Line (166° E to 180° W) at depths from 200 to 1,200 m
- Conducted intermittently since 2002 (funding dependent)
- Modified Index-Stratified Random of Successful Stations Survey Design
- Focus species for the Bering Sea slope include giant grenadier, Pacific ocean perch, popeye grenadier, walleye pollock, and arrowtooth flounder.

1. Survey background



Figure 1.1.: Strata used in the Eastern Bering Sea Survey.

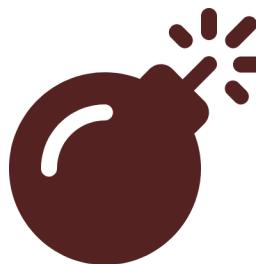
1. Survey background



Figure 1.2.: Strata used in the Bering Sea Slope Survey.

2. Workflow

2.1. Data workflow from boat to production



Syntax error in graph
mermaid version 9.1.1

2.2. Organization

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. Workflow

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.

2.3. 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

2. Workflow

- **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

3. News

3.1. Future plans

3.1.1. GOA 2025 Restratiification – Mock Data for Testing

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.

- 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**".

3.2. Previous updates

- 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.
 - 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.

3. News

- Provisional AKFIN and FOSS tables are also available in the GAP_PROD-UCTS 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.
- 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.
- February 2023: Decision was made to include the mock restratified GOA data with the development of the new consolidated standard data products.
- 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.
- 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.

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. These data are created using the gapindex R package v2.1.0.

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{GAPPproducts,
  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}
}
```

4. Data description

4.1. Data tables

4.1.1. AGECOMP

Region-level age compositions by sex/length bin.

Number of rows: 533,785

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)

4. Data description

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 (millimeters)

LENGTH_MM_SD

standard deviation of length at age weighted by numbers at length

numeric

NUMBER(38,3)

Variance of mean length.

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)

4. Data description

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

Survey Year

year

NUMBER(10,0)

NA

4.1.2. AREA

[There is currently no description for this table.]

Number of rows: 473

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)

4. Data description

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 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.

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").

DEPTH_MAX_M

Area ID Maximum Depth (m)

meters

NUMBER(38,3)

Maximum depth (meters).

DEPTH_MIN_M

Area ID Minimum Depth (m)

meters

NUMBER(38,3)

4. Data description

Minimum depth (meters).

DESCRIPTION

Description

text

VARCHAR2(4000 BYTE)

Description of row observation.

DESIGN_YEAR

Design year

year

NUMBER(10,0)

Year ID associated with a given value AREA_ID. This field describes the changes in the survey design over time.

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

4. Data description

4.1.3. BIOMASS

Stratum/subarea/region-level mean CPUE (weight and numbers), total biomass, and total abundance with associated variances.

Number of rows: 4,574,245

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 total biomass.

BIOMASS_VAR

Estimated Biomass Variance

numeric

NUMBER(38,6)

The estimated variance associated with the total biomass.

CPUE_KGKM2_MEAN

4. Data description

Mean Weight CPUE

kilograms per kilometers squared

NUMBER(38,6)

The mean catch weight (kilograms) per unit effort (area swept by the net, units squared kilometers).

CPUE_KGKM2_VAR

Variance of the Mean Weight CPUE

kilograms per kilometers squared

NUMBER(38,6)

The variance of mean catch weight (kilograms) per unit effort (area swept by the net, units squared kilometers).

CPUE_NOKM2_MEAN

Mean Numeric CPUE

count per kilometers squared

NUMBER(38,6)

The mean of numerical catch per unit effort (area swept by the net, units square kilometers).

CPUE_NOKM2_VAR

Variance of the Mean Numeric CPUE

count per kilometers squared

NUMBER(38,6)

The variance of mean numerical catch per unit effort (area swept by the net, units square kilometers).

N_COUNT

Hauls with taxon counts

numeric

NUMBER(38,0)

Total number of hauls with positive count data.

N_HAUL

4. Data description

Valid hauls

numeric

NUMBER(38,0)

Total number of hauls.

N_LENGTH

Hauls with taxon lengths

numeric

NUMBER(38,0)

Total number of hauls with length data.

N_WEIGHT

Hauls with catch

numeric

NUMBER(38,0)

Total number of hauls with positive catch biomass.

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

4. Data description

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

Survey Year

year

NUMBER(10,0)

NA

4.1.4. CPUE

Haul-level zero-filled weight and numerical catch-per-unit-effort.

Number of rows: 37,905,115

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AREA_SWEPT_KM2

Area Swept (km)

4. Data description

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 whole number of individuals caught in haul.

CPUE_KGKM2

Weight CPUE (kg/km²)

kilograms per kilometers squared

NUMBER(38,6)

Catch weight (kilograms) per unit effort (area swept by the net, units square kilometers).

CPUE_NOKM2

Number CPUE (no/km²)

count per kilometers squared

NUMBER(38,6)

Numerical catch per unit effort (area swept by the net, units square kilometers).

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

4. Data description

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.

4.1.5. SURVEY DESIGN

[There is currently no description for this table.]

Number of rows: 126

Number of columns: 4

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

DESIGN_YEAR

Design year

year

NUMBER(10,0)

Year ID associated with a given value AREA_ID. This field describes the changes in the survey design over time.

SURVEY

Survey Name

text

4. Data description

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.

YEAR

Survey Year

year

NUMBER(10,0)

NA

4.1.6. METADATA_TABLE

[There is currently no description for this table.]

Number of rows: 8

Number of columns: 3

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

METADATA_SENTENCE

Sentence

text

4. Data description

VARCHAR2(255 BYTE)

Table metadata sentence.

METADATA_SENTENCE_NAME

Metadata sentence name

text

VARCHAR2(255 BYTE)

Name of table metadata sentence.

METADATA_SENTENCE_TYPE

Sentence type

text

VARCHAR2(255 BYTE)

Type of sentence to have in table metadata.

4.1.7. STRATUM_GROUPS

[There is currently no description for this table.]

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)

4. Data description

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)

Year ID associated with a given value AREA_ID. This field describes the changes in the survey design over time.

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 region. 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.

4. Data description

4.1.8. SIZECOMP

[There is currently no description for this table.]

Number of rows: 3,134,121

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 bin 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

4. Data description

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

Survey Year

year

NUMBER(10,0)

NA

Part III.

AKFIN

The Alaska Fisheries Information Network

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 Program}},
  year = {2023},
  title = {AFSC Goundfish Assessment Program Design-Based Production Data},
  howpublished = {\url{https://www.psmfc.org/program/alaska-fisheries-information-network-akfin}},
  publisher = {{U.S. Dep. Commer.}},
  copyright = {Public Domain}
}
```

5. Data description

5.1. Data description

[OUTDATED] AKFIN Answers <https://akfin.psmfc.org/akfin-answers/> is an Oracle BI tool used for distributing data to stock assessors and other users. Usernames and passwords are distinct from direct akfin database credentials (though they may be identical). RACE data on the AKFIN Answers stock assessment dashboard is located on the “RACE Survey” tab for groundfish and the “Crab” tab for crab surveys. More detailed descriptions of each report are included within that report.

5.2. Data tables

5.2.1. AKFIN_AGECOMP

[There is currently no description for this table.]

Number of rows: 533,785

Number of columns: 9

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AGE

Age bin of taxon

year

5. Data description

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 (millimeters)

LENGTH_MM_SD

standard deviation of length at age weighted by numbers at length

numeric

NUMBER(38,3)

Variance of mean length.

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

5. Data description

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

Survey Year

year

NUMBER(10,0)

NA

5.2.2. AKFIN_AREA

[There is currently no description for this table.]

Number of rows: 473

Number of columns: 10

Column name from data

Descriptive column Name

Units

5. Data description

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 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.

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").

DEPTH_MAX_M

Area ID Maximum Depth (m)

meters

5. Data description

NUMBER(38,3)

Maximum depth (meters).

DEPTH_MIN_M

Area ID Minimum Depth (m)

meters

NUMBER(38,3)

Minimum depth (meters).

DESCRIPTION

Description

text

VARCHAR2(4000 BYTE)

Description of row observation.

DESIGN_YEAR

Design year

year

NUMBER(10,0)

Year ID associated with a given value AREA_ID. This field describes the changes in the survey design over time.

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

5. Data description

NA

NA

5.2.3. AKFIN_BIOMASS

[There is currently no description for this table.]

Number of rows: 4,574,245

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 total biomass.

BIOMASS_VAR

Estimated Biomass Variance

numeric

NUMBER(38,6)

5. Data description

The estimated variance associated with the total biomass.

CPUE_KGKM2_MEAN

Mean Weight CPUE

kilograms per kilometers squared

NUMBER(38,6)

The mean catch weight (kilograms) per unit effort (area swept by the net, units squared kilometers).

CPUE_KGKM2_VAR

Variance of the Mean Weight CPUE

kilograms per kilometers squared

NUMBER(38,6)

The variance of mean catch weight (kilograms) per unit effort (area swept by the net, units squared kilometers).

CPUE_NOKM2_MEAN

Mean Numeric CPUE

count per kilometers squared

NUMBER(38,6)

The mean of numerical catch per unit effort (area swept by the net, units square kilometers).

CPUE_NOKM2_VAR

Variance of the Mean Numeric CPUE

count per kilometers squared

NUMBER(38,6)

The variance of mean numerical catch per unit effort (area swept by the net, units square kilometers).

N_COUNT

Hauls with taxon counts

numeric

NUMBER(38,0)

5. Data description

Total number of hauls with positive count data.

N_HAUL

Valid hauls

numeric

NUMBER(38,0)

Total number of hauls.

N_LENGTH

Hauls with taxon lengths

numeric

NUMBER(38,0)

Total number of hauls with length data.

N_WEIGHT

Hauls with catch

numeric

NUMBER(38,0)

Total number of hauls with positive catch biomass.

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.

5. Data description

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

Survey Year

year

NUMBER(10,0)

NA

5.2.4. AKFIN_CATCH

[There is currently no description for this table.]

Number of rows: 989,351

Number of columns: 6

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

5. Data description

CATCHJOIN

NA

NA

NA

NA

COUNT

Taxon Count

count, whole number resolution

NUMBER(38,0)

Total whole number of individuals caught in haul.

CRUISEJOIN

Cruise ID

ID code

NUMBER(38,0)

Unique interger ID 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

5. Data description

Taxon Weight (kg)

kilograms

NUMBER(38,3)

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

5.2.5. AKFIN_CPUE

[There is currently no description for this table.]

Number of rows: 37,905,115

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 whole number of individuals caught in haul.

CPUE_KGKM2

Weight CPUE (kg/km²)

5. Data description

kilograms per kilometers squared

NUMBER(38,6)

Catch weight (kilograms) per unit effort (area swept by the net, units square kilometers).

CPUE_NOKM2

Number CPUE (no/km²)

count per kilometers squared

NUMBER(38,6)

Numerical catch per unit effort (area swept by the net, units square kilometers).

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.

5. Data description

5.2.6. AKFIN_CRUISE

[There is currently no description for this table.]

Number of rows: 187

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 integer 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)

Unique interger ID 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

5. Data description

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.

5. Data description

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

Survey Year

year

NUMBER(10,0)

NA

5.2.7. AKFIN_HAUL

[There is currently no description for this table.]

Number of rows: 36,114

Number of columns: 25

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

ACCESSORIES

NA

NA

NA

NA

5. Data description

BOTTOM_TYPE

NA

NA

NA

NA

CRUISEJOIN

Cruise ID

ID code

NUMBER(38,0)

Unique integer ID assigned to each survey, vessel, and year combination.

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_GEAR_M

NA

NA

NA

NA

DEPTH_M

Depth (m)

degrees Celsius

NUMBER(38,1)

Bottom depth (meters).

DISTANCE_FISHED_KM

Distance Fished (km)

5. Data description

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).

GEAR

NA

NA

NA

NA

GEAR_TEMPERATURE_C

NA

NA

NA

NA

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)

5. Data description

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

HAUL_TYPE

Haul Sampling Type

ID code

NUMBER(38,0)

Type of haul sampling method. For a complete list of haul type ID codes, review the code books.

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.

5. Data description

NET_HEIGHT_M

Net Height (m)

meters

NUMBER(38,1)

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

NET_MEASURED

Net measured during haul

logical

NA

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(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.

STATION

Station ID

ID code

VARCHAR2(255 BYTE)

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

5. Data description

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 region. 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.

WIRE_LENGTH_M

Trawl wire length

meters

NUMBER(38,0)

Length of wire deployed during a given haul in meters.

5.2.8. AKFIN_LENGTH

[There is currently no description for this table.]

Number of rows: 2,587,694

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

5. Data 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 bin in millimeters.

LENGTH_TYPE

NA

NA

NA

NA

SAMPLE_TYPE

NA

NA

NA

NA

SEX

Sex of a specimen

5. Data description

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.

5.2.9. AKFIN_METADATA_COLUMN

[There is currently no description for this table.]

Number of rows: 128

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

5. Data description

VARCHAR2(4000 BYTE)

Description 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.

5.2.10. AKFIN_SIZECOMP

[There is currently no description for this table.]

Number of rows: 3,134,121

Number of columns: 7

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

5. Data 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 bin 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.

5. Data description

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

Survey Year

year

NUMBER(10,0)

NA

5.2.11. AKFIN_SPECIMEN

[There is currently no description for this table.]

Number of rows: 360,602

Number of columns: 13

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

AGE_DETERMINATION_METHOD

NA

NA

NA

NA

AGE_YEARS

5. Data description

NA

NA

NA

NA

CRUISEJOIN

Cruise ID

ID code

NUMBER(38,0)

Unique interger ID 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 bin in millimeters.

MATURITY

NA

NA

5. Data description

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

5. Data description

WEIGHT_G

NA

NA

NA

NA

5.2.12. AKFIN_STRATUM_GROUPS

[There is currently no description for this table.]

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)

Year ID associated with a given value AREA_ID. This field describes the changes in the survey design over time.

5. Data description

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 region. 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.

5.2.13. AKFIN_SURVEY DESIGN

[There is currently no description for this table.]

Number of rows: 126

Number of columns: 4

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

DESIGN_YEAR

Design year

year

NUMBER(10,0)

5. Data description

Year ID associated with a given value AREA_ID. This field describes the changes in the survey design over time.

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.

YEAR

Survey Year

year

NUMBER(10,0)

NA

5.2.14. AKFIN_TAXONOMIC_CLASSIFICATION

[There is currently no description for this table.]

Number of rows: 2,757

Number of columns: 19

Column name from data

Descriptive column Name

Units

5. Data description

Oracle data type

Column description

CLASS_TAXON

Class phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of class_taxon of a given species.

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.

DATABASE

Database source

category

VARCHAR2(255 BYTE)

Taxonomic database source, either ITIS or WoRMS.

DATABASE_ID

Species ID in Database

ID code

VARCHAR2(255 BYTE)

Species ID code of a species in the taxonomic "DATABASE" source.

FAMILY_TAXON

Family phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of family_taxon of a given species.

5. Data description

GENUS_TAXON

Genus phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of genus_taxon of a given species.

ID_RANK

Lowest taxonomic rank

text

VARCHAR2(255 BYTE)

Lowest taxonomic rank of a given species entry.

KINGDOM_TAXON

Kingdom phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of kingdom_taxon of a given species.

ORDER_TAXON

Order phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of order_taxon of a given species.

PHYLUM_TAXON

Phylum phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of phylum_taxon of a given species.

SPECIES_CODE

Taxon Code

5. Data description

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.

SPECIES_NAME

NA

NA

NA

NA

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.

SUBORDER_TAXON

Suborder phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of suborder_taxon of a given species.

SUBPHYLUM_TAXON

Subphylum phylogenetic rank

category

VARCHAR2(255 BYTE)

5. Data description

Phylogenetic latin rank of subphylum_taxon of a given species.

SUPERCLASS_TAXON

Superclass phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of superclass_taxon of a given species.

SUPERFAMILY_TAXON

Superfamily phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of superfamily_taxon of a given species.

SUPERORDER_TAXON

Superorder phylogenetic rank

category

VARCHAR2(255 BYTE)

Phylogenetic latin rank of superorder_taxon of a given species.

6. Access data

6.1. 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.

6.1.1. 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.

6.2. Data SQL Query Examples:

6.2.1. 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.

6. Access data

```
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_TAXONOMY"
)

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")))
}
```

6.2.2. 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
```

6. Access data

```
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 6.1.: Ex. 1: GOA Pacific Ocean perch biomass and abundance.

BIOMASS_POPULATI MT COUNT	YEARDESCRIPTION
483,622.6833,902,16	GOA 1993Region: All Strata
483,622.6833,902,161	GOA 1993Region: All Strata

6. Access data

BIOMASS_POPULATI MT	COUNT	YEAR	DESCRIPTION
771,412.81,252,616,6		1996	GOA Region: All Strata
771,412.81,252,616,603		1996	GOA Region: All Strata
727,063.51,212,034,9		1999	GOA Region: All Strata
727,063.51,212,034,913		1999	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
```

6. Access data

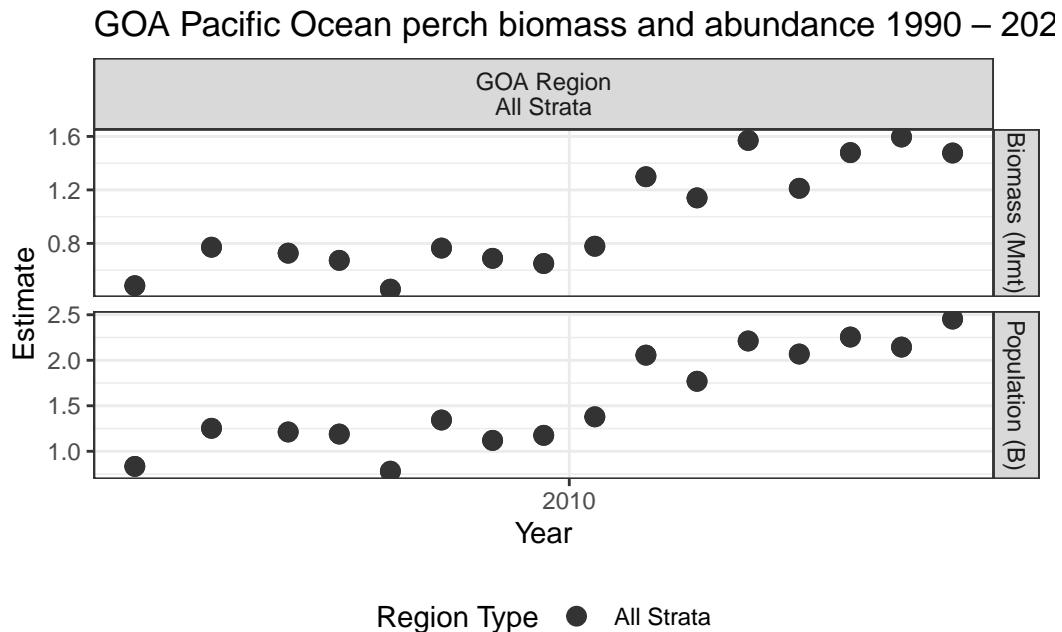


Figure 6.1.: Ex. 1: GOA Pacific Ocean perch biomass and abundance.

6.2.3. 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
```

6. Access data

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

LENGTH_MM	YEAR
180	2014
190	2014
200	2014
210	2014
220	2014
230	2014

```
# 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
```

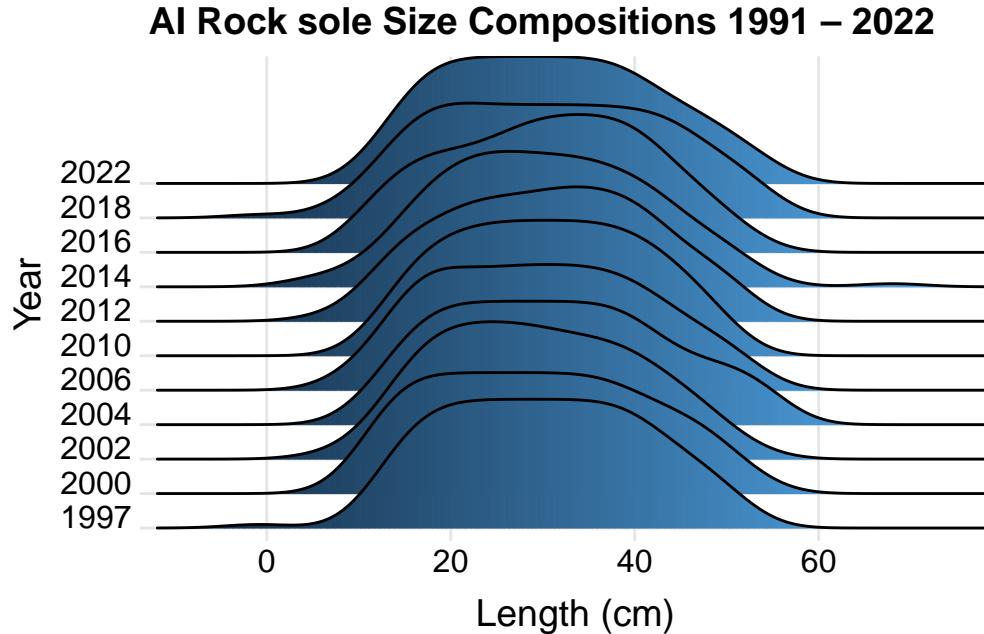


Figure 6.2.: Ex. 2: AI Rock sole size compositions and ridge plot.

6.2.4. 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,
```

6. Access data

```

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

AGE	POPULATI COUNT	SEX
9	39,371	3
10	32,156	3
11	15,200	3
12	9,976	3
13	1,957	3
	1131,950,343	1

```

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

```

6. Access data

```
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::ggttitle(label = "EBS Walleye Pollock Age Compositions 1982 – 2022") +  
  ggplot2::guides(fill = guide_legend(title = "Sex"))+  
  ggplot2::theme_bw()  
  
figure
```

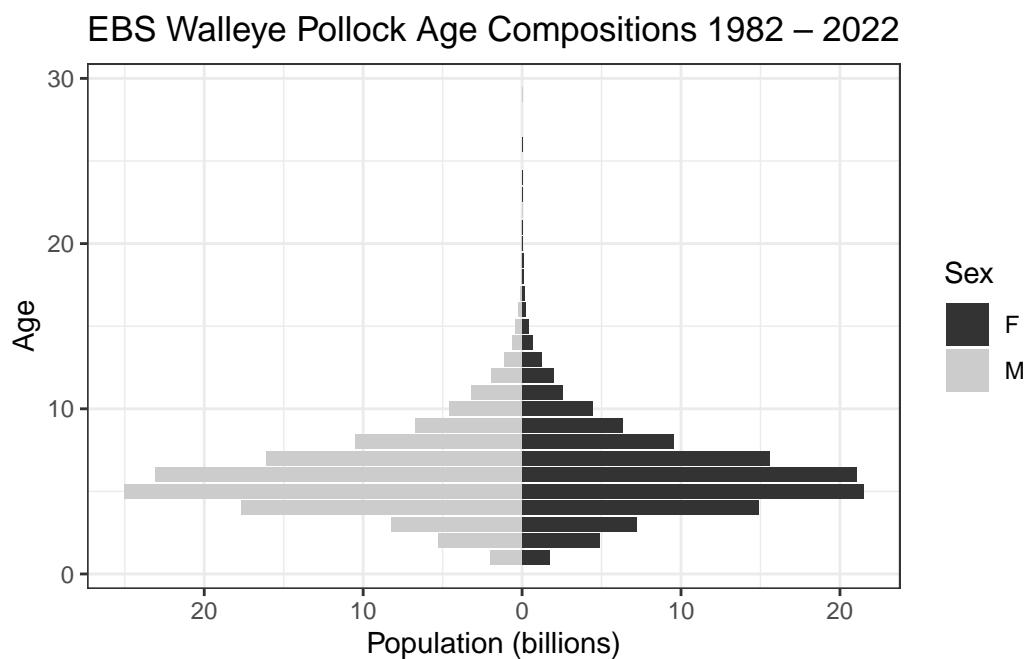


Figure 6.3.: Ex. 3: EBS Walleye Pollock Age Compositions and Age Pyramid.

6.2.5. 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 =
```

6. Access data

```
"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 = "")
```

6. Access data

Table 6.4.: Ex. 4: NBS Pacific cod biomass and abundance.

BIOMASS_POPULATI MT COUNT	YEAR	AREA_- NAME
95,849.98368,767,498	2021	Inner Domain
107,096.73002,734,142	2019	Inner Domain
76,708.43339,605,860	2023	Inner Domain
132,490.1506,187,245	2017	Inner Domain
96,500.69760,433,135	2022	Inner Domain
7,462.559 4,724,153	2010	Inner Domain

```
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
```

6. Access data

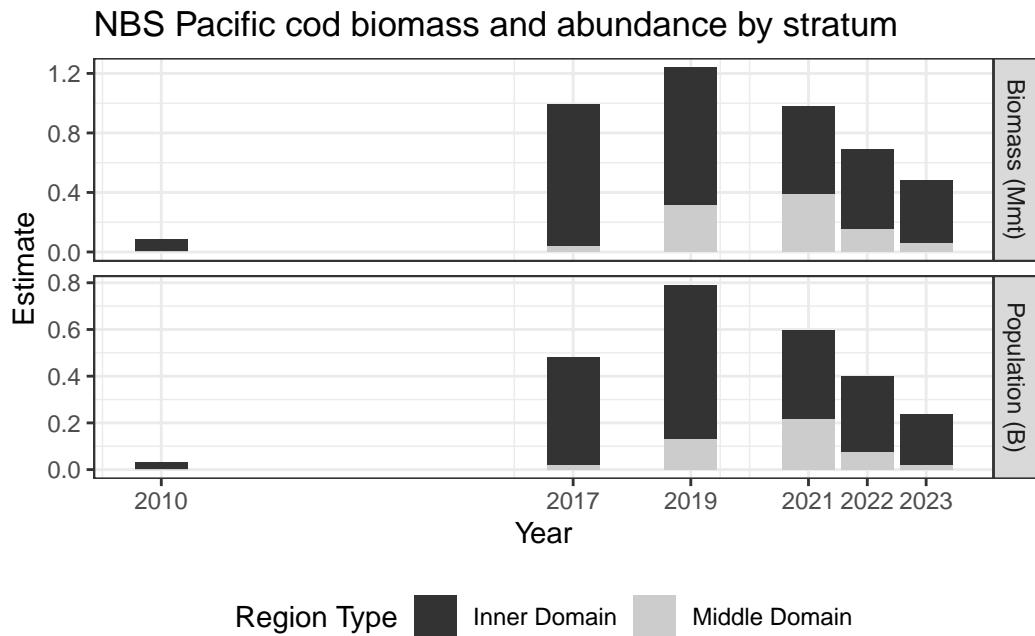


Figure 6.4.: Ex. 4: NBS Pacific cod biomass and abundance.

6.2.6. 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,
BIOMASS_VAR,
YEAR
FROM GAP_PRODUCTS.AKFIN_BIOMASS
WHERE SPECIES_CODE = 30060
AND SURVEY_DEFINITION_ID = 47
AND AREA_ID = 99903
AND YEAR BETWEEN 1984 AND 2023;") %>%
  janitor::clean_names() %>%
```

6. Access data

```
dplyr::mutate(biomass_kmt = biomass_mt/1000,
  # **approximate** 95% confidence interval
  biomass_kci_up = (biomass_mt + (2*sqrt(biomass_var)))/1000,
  biomass_kci_dw = (biomass_mt - (2*sqrt(biomass_var)))/1000)

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

Table 6.5.: Ex. 5: GOA Pacific Ocean perch biomass and line plot.

survey_definition_id	biomass_mt	biomass_var	year	biomass_kmt	biomass_kci_up	biomass_kci_dw
47	483,622.611,803,384		1993	483.6226	700.9093266.33581	
47	771,412.841,434,152,202		1996	771.41281,178.5204364.30515		
47	727,063.5150,983,54		1999	727.06351,504.1955	-50.06854	
47	673,155.149,285,342,922		2001	673.15511,117.1611229.14901		
47	457,421.65,186,126,!		2003	457.4216	601.4511313.39204	
47	764,901.421,499,807,010		2005	764.90141,058.1577471.64517		

```
a_mean <- dat %>%
  dplyr::group_by(survey_definition_id) %>%
  dplyr::summarise(biomass_kmt = mean(biomass_kmt, na.rm = TRUE),
    minyr = min(year, na.rm = TRUE),
    maxyr = max(year, na.rm = TRUE))

figure <-
  ggplot(data = dat,
    mapping = aes(x = year,
      y = biomass_kmt)) +
  ggplot2::geom_point(size = 2.5, color = "grey40") +
  ggplot2::scale_x_continuous(
    name = "Year",
    labels = scales::label_number(
      accuracy = 1,
```

6. Access data

```
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_kmt,
                 yend = biomass_kmt),
  linetype = "dashed",
  linewidth = 2) +
ggplot2::geom_errorbar(
  mapping = aes(ymin = biomass_kci_dw, ymax = biomass_kci_up),
  position = position_dodge(.9),
  alpha = 0.5, width=.2) +
ggplot2::ggttitle(
  label = "GOA Pacific Ocean Perch Biomass 1984-2021",
  subtitle = paste0("Mean = ",
                    formatC(x = a_mean$biomass_kmt,
                           digits = 2,
                           big.mark = ",",
                           format = "f"),
                    " Kmt")) +
ggplot2::theme_bw()

figure
```

6. Access data

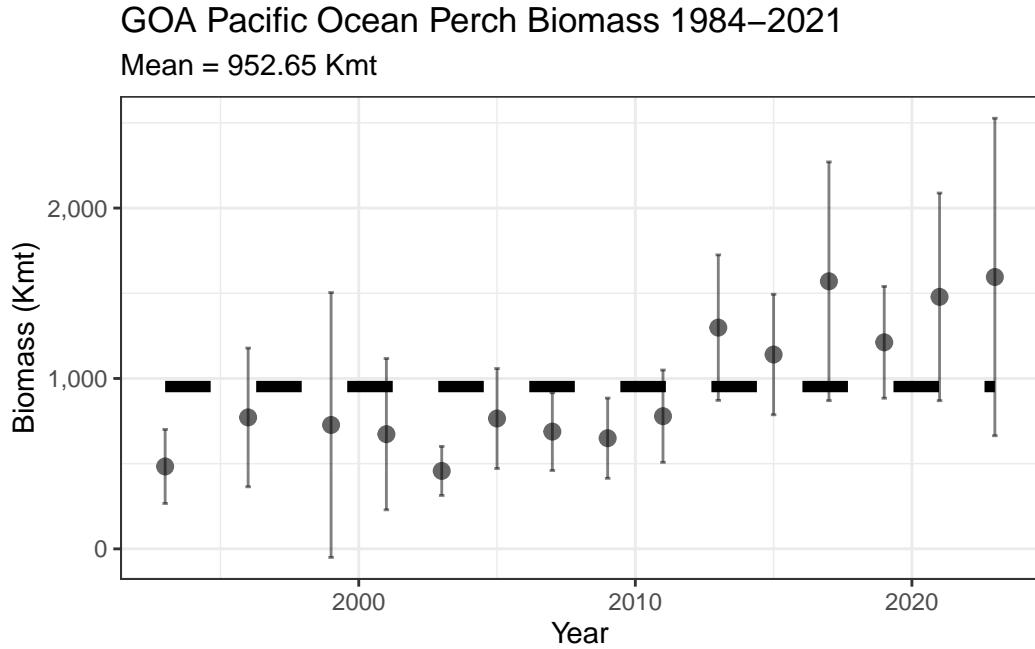


Figure 6.5.: Ex. 5: GOA Pacific Ocean perch biomass and line plot.

6.2.7. 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
```

6. Access data

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

CPUE - KGHA	LATITUDE	LONGITUDE
0.0000000	58.75863	-174.9285
0.2813533	57.32545	-173.3217
0.0000000	57.64161	-172.7963
0.0000000	59.67831	-172.5754
0.0000000	60.96936	-174.8760
0.0000000	58.64012	-173.5922

```
# 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]
```

6. Access data

[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)
```

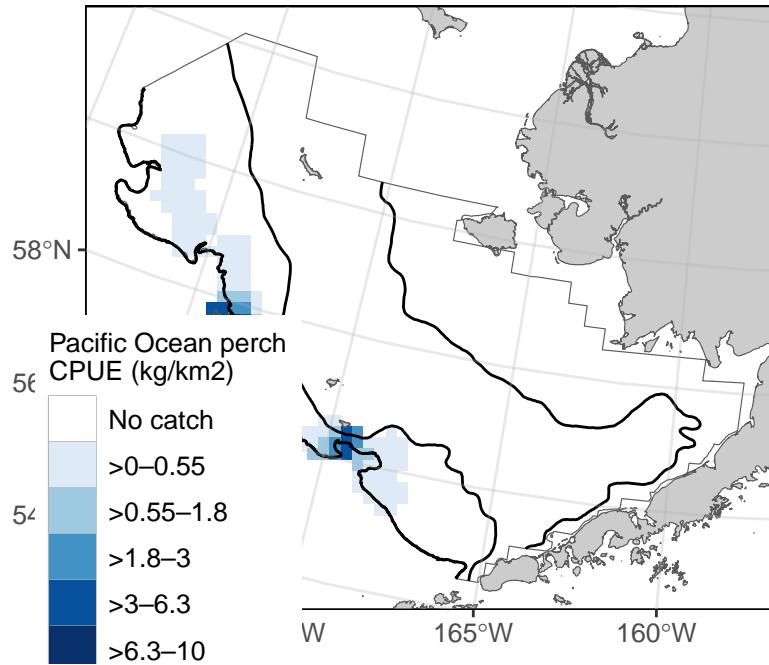


Figure 6.6.: Ex. 6: EBS Pacific Ocean perch CPUE and akgridmaps map.

7. Access API data using R

Use the below function to pull AKFIN data through AKFIN's API.

```
# load libraries
library(dplyr)
library(magrittr)
library(httr)
library(flextable)

# tell R to not use scientific notation
options(scipen=999)

# function for pulling data from the api using the httr package
get_gap_biomass<-function(area_id, species_code) {
  # paste(... collapse=",") puts commas between vector elements
  area_id <- paste(area_id, collapse = ",")
  species_code <- paste(species_code, collapse = ",")
  # httr code, parameters are after the '?'
  httr::content(
    httr::GET(paste0("https://apex.psmfc.org/akfin/data_marts/akmp/gap_biomass?area_id=",
                     area_id,
                     "&species_code=",
                     species_code)),
    type = "application/json") %>%
    # convert to data frame
    bind_rows()
}
```

7.1. Ex. 1: Load lingcod data

7. Access API data using R

```
lingcod_biomass <- get_gap_biomass(area_id=c(40, 41), species_code=21910)
flextable::flextable(head(lingcod_biomass)) %>%
  flextable::theme_zebra()
```

Part IV.

Public Data (FOSS)

The final, validated survey data are publicly accessible soon after surveys are completed on the Fisheries One Stop Shop (FOSS) platform. This data includes catch, haul, and environmental data collected at each station. On the FOSS data platform, users can interactively select, view, and download data. Descriptive documentation and user-examples are available on the metadata page.

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

Cite this data

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 Query},
  howpublished = {https://www.fisheries.noaa.gov/foss},
  publisher = {{U.S. Dep. Commer.}},
  copyright = {Public Domain}
}
```

8. 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.

8. Data description

8.1. Data tables

8.1.1. FOSS_CATCH

[There is currently no description for this table.]

Number of rows: 37,591,876

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 whole number of individuals caught in haul.

CPUE_KGKM2

Weight CPUE (kg/km²)

kilograms per kilometers squared

NUMBER(38,6)

Catch weight (kilograms) per unit effort (area swept by the net, units square kilometers).

8. Data description

CPUE_NOKM2

Number CPUE (no/km²)

count per kilometers squared

NUMBER(38,6)

Numerical catch per unit effort (area swept by the net, units square kilometers).

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://it.is.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.

8. Data description

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)

8. Data description

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

8.1.2. FOSS_CPUE_PRESONLY

[There is currently no description for this table.]

Number of rows: 37,591,876

Number of columns: 37

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.

COMMON_NAME

Taxon Common Name

text

VARCHAR2(255 BYTE)

8. Data description

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 whole number of individuals caught in haul.

CPUE_KGKM2

Weight CPUE (kg/km²)

kilograms per kilometers squared

NUMBER(38,6)

Catch weight (kilograms) per unit effort (area swept by the net, units square kilometers).

CPUE_NOKM2

Number CPUE (no/km²)

count per kilometers squared

NUMBER(38,6)

Numerical catch per unit effort (area swept by the net, units square kilometers).

CRUISE

Cruise ID

ID code

NUMBER(38,0)

This is a six-digit integer 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)

8. Data description

Unique integer ID 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 (meters).

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

8. Data description

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://it.is.gov/>).

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)

8. Data description

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.

SCIENTIFIC_NAME

Taxon Scientific Name

text

8. Data description

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.

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 region. Stratum of value 0 indicates experimental tows.

SURFACE_TEMPERATURE_C

8. Data description

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.

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

8. Data description

slope survey (1976-2010), and Species identification confidence in the Gulf of Alaska and Aleutian Islands surveys (1980-2011).

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.

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/>).

YEAR

8. Data description

Survey Year

year

NUMBER(10,0)

NA

8.1.3. FOSS_HAUL

[There is currently no description for this table.]

Number of rows: 32,626

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

8. Data description

ID code

NUMBER(38,0)

This is a six-digit integer 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)

Unique interger ID 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 (meters).

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

8. Data description

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)

8. Data description

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)

8. Data description

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 region. 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

8. Data description

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

Survey Year

year

NUMBER(10,0)

8. Data description

NA

8.1.4. FOSS_TAXON_GROUP

[There is currently no description for this table.]

Number of rows: 37,606

Number of columns: 3

Column name from data

Descriptive column Name

Units

Oracle data type

Column description

CLASSIFICATION

Taxonomic classification rank group

category

VARCHAR2(255 BYTE)

Phylogenetic classification group rank for a given species.

RANK_ID

NA

NA

NA

NA

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.

9. Using the FOSS platform

9.1. 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 Species 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.

9. Using the FOSS platform

The screenshot shows the AFSC Groundfish and Crab Assessment Program Bottom Trawl Surveys data interface. At the top, the NOAA Fisheries logo is visible. The main title "AFSC Groundfish and Crab Assessment Program" and "Bottom Trawl Surveys" is displayed above a banner image of a fish swimming in water. Below the title, a note says "Survey data also available through API". On the left, a sidebar menu includes links to FUS Report, Landings, Foreign Trade, Top US Ports, Processed Products, Per Capita Consumption, Supply, USCG Vessels, AFSC GAP Survey (selected), AFSC GAP Metadata, Partners, Metadata and Caveats, Frequently Asked Questions, Quick Start Guide, and Comments. The main content area has a "Data Caveats" section with a note about survey data being presence-only. A "Parameters" section contains dropdown menus for Survey (Aleutian Islands Bottom Trawl Survey, Eastern Bering Sea Slope Bottom Trawl Survey, Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey, Gulf of Alaska Bottom Trawl Survey, Northern Bering Sea Crab/Groundfish Survey - Eastern Bering Sea Shelf S), Year (2022, 2021, 2019, 2018, 2017, 2016, 2015, 2014), and Species (Abalone Jingle, Abietinaria, Abyssal Crangon, Acanthascus, Acantholithodes, Achanax johnsoni, Acoel Turbellarian, Acteocina). There is also a "Search Species" input field, "Search" and "Reset All Parameters" buttons, and a "Data Format" section with radio buttons for All Data Fields, Catch Data: All Units, Catch Data: Hectares, Catch Data: Square Thousand km, Catch Data: Square km, and Haul Data. A large blue "RUN REPORT" button is at the bottom.

Figure 9.1.: AFSC Groundfish and Crab Assessment Program Bottom Trawl Survey data interface on the Fisheries One Stop Shop platform.

9. Using the FOSS platform

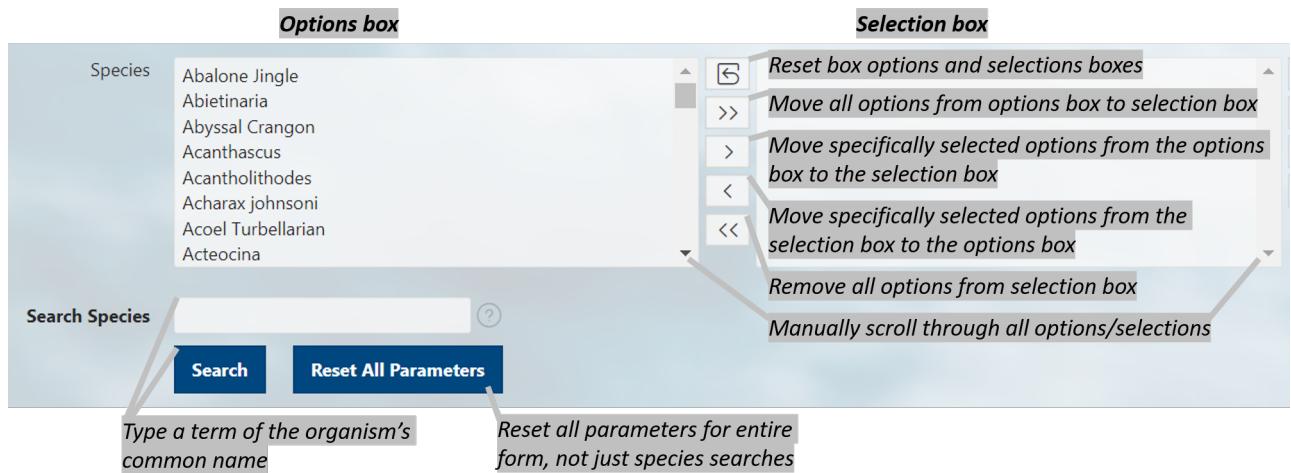


Figure 9.2.: Diagram of selection and search tools available on the FOSS platform.

- **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.

9.2. 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

9. Using the FOSS platform

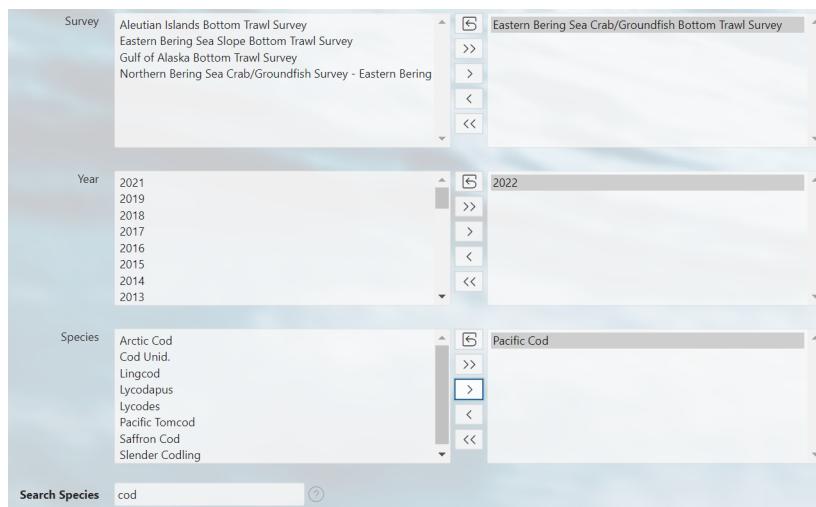


Figure 9.3.: Diagram of selection and search tools available on the FOSS platoform.

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).

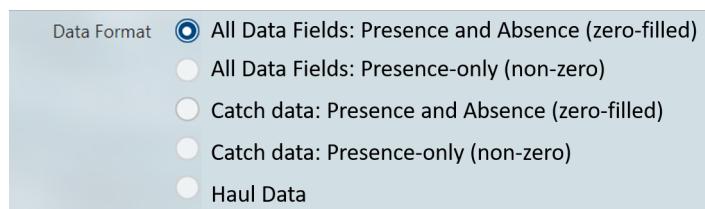
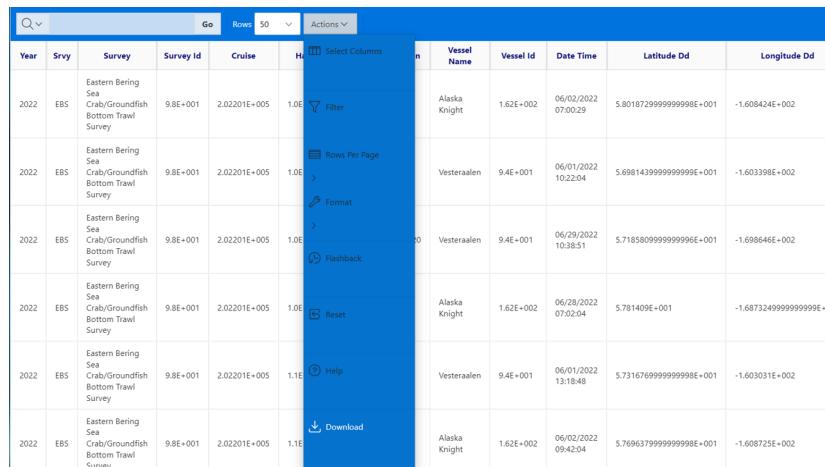


Figure 9.4.: Diagram of the pre-set data format options.

9. Using the FOSS platform

9.3. 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.



The screenshot shows a report interface with a table of survey data. A context menu is open over the last row of the table, listing options: Select Columns, Filter, Rows Per Page, Format, Flashback, Reset, Help, and Download. The table has columns for Year, Srvy, Survey, Survey Id, Cruise, and H. The data rows represent surveys from 2022, specifically EBS surveys for the Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey. The last row is highlighted with a blue background.

Year	Srvy	Survey	Survey Id	Cruise	H	
2022	EBS	Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	9.8E+001	2.02201E+005	1.0E	Filter
2022	EBS	Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	9.8E+001	2.02201E+005	1.0E	Rows Per Page
2022	EBS	Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	9.8E+001	2.02201E+005	1.0E	Format
2022	EBS	Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	9.8E+001	2.02201E+005	1.0E	Flashback
2022	EBS	Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	9.8E+001	2.02201E+005	1.1E	Reset
2022	EBS	Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	9.8E+001	2.02201E+005	1.1E	Help
2022	EBS	Eastern Bering Sea Crab/Groundfish Bottom Trawl Survey	9.8E+001	2.02201E+005	1.1E	Download

Figure 9.5.: Example data returned from running the report.

10. Access via API and 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](#).

10.1. 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$content))
# names(data)
tibble::as_tibble(data$items) %>%
  dplyr::mutate_if(is.character, type.convert, as.is = TRUE) %>%
  dplyr::mutate(across(where(is.numeric), round, 3)) %>%
  head(3) %>%
flextable::flextable() %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "cruise", "species_code", "tsn", "ak_surv
```

yearsrvy	survey	survey_id	cruise	haul	stratumstation	vessel_name	vessel
2002AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722307-63	Vesteraale	
2002AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722307-63	Vesteraalen	
2002AI	Aleutian Islands Bottom Trawl Survey	52	200201	6	722307-63	Vesteraale	

10.2. 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"

10.3. Ex. 3: Filter by Year

Show all the data greater than the year 2020.

10. Access via API and R

```
res <- httr::GET(url = paste0(api_link, '?q={"year":{$gt":2020}}'))  
data <- jsonlite::fromJSON(base::rawToChar(res$content))  
  
as_tibble(data$items) %>%  
  mutate_if(is.character, type.convert, as.is = TRUE) %>%  
  head(3) %>%  
  dplyr::mutate(across(where(is.numeric), round, 3)) %>%  
  dplyr::select(year, srvy, stratum, species_code, cpue_kgkm2) %>%  
  flextable::flextable() %>%  
  flextable::theme_zebra() %>%  
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 10.2.: Ex. 3: Filter by Year.

year	srvy	stratum	species_code	cpue_kgkm2
2022AI		722	10261	673.326
2022AI		793	80540	0.361
2022AI		722	21347	758.091

10.4. 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))  
  
as_tibble(data$items) %>%  
  mutate_if(is.character, type.convert, as.is = TRUE) %>%  
  head(3) %>%  
  dplyr::mutate(across(where(is.numeric), round, 3)) %>%
```

```
dplyr::select(year, srvy, stratum, species_code, cpue_kgkm2) %>%
flextable::flextable() %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 10.3.: Ex. 4: Filter by species name.

yearsrvy	stratum	species_- code	cpue_- kgkm2
2002AI	721	21740	0.640
2002AI	722	21740	775.322
2002AI	722	21740	10,685.806

10.5. 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": "%pollock%"} }')))
data <- jsonlite::fromJSON(base::rawToChar(res$content))

as_tibble(data$items) %>%
  mutate_if(is.character, type.convert, as.is = TRUE) %>%
  head(3) %>%
  dplyr::mutate(across(where(is.numeric), round, 3)) %>%
  dplyr::select(year, srvy, stratum, species_code, cpue_kgkm2) %>%
flextable::flextable() %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 10.4.: Ex. 5: Combination of year and name filters.

yearsrvy	stratum	species_- code	cpue_- kgkm2
2022AI	722	2174022	754.334
2022AI	793	21740	7,853.632
2022AI	721	21740	7,235.010

10.6. 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))

as_tibble(data$items) %>%
  mutate_if(is.character, type.convert, as.is = TRUE) %>%
  head(3) %>%
  dplyr::mutate(across(where(is.numeric), round, 3)) %>%
  dplyr::select(year, srvy, stratum, species_code, cpue_kgkm2) %>%
  flextable::flextable() %>%
  flextable::theme_zebra() %>%
  flextable::colformat_num(x = ., j = c("year", "species_code"), big.mark = "")
```

Table 10.5.: Ex. 6: Combination of year, srvy, stratum.

yearsrvy	stratum	species_- code	cpue_- kgkm2
1989EBS	10	40500	9.620
1989EBS	10	68578	9.620
1989EBS	10	21313	18.179

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

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

```
# 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::mutate(across(where(is.numeric), round, 3)) %>%
  dplyr::distinct()

data <- dplyr::left_join(data_haul, data_catch) %>%
  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 10.6.: 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	-173.07761	3,256.716473
81		61.34965	-172.22516	0.958261

10. Access via API and R

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

figure <- akgfmmaps::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]

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

10. Access via API and R

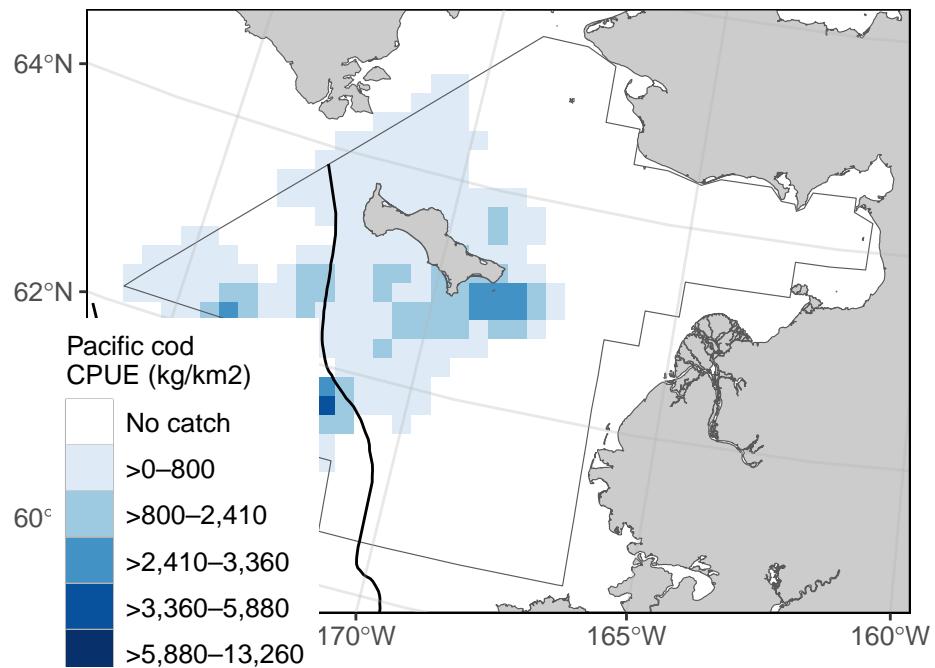


Figure 10.1.: Ex. 7: Visualize CPUE data in distribution map.

11. Access via API and Python

11.0.1. {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.

11.0.2. 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()
```

11. Access via API and Python

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

11.0.3. 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.

11. Access via API and Python

11.0.4. 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.

11.0.5. 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')
```

11. Access via API and Python

```
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.

11.0.6. 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.

11.0.7. 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.

11. Access via API and Python

```
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 <code>set_presence_only(True)</code>	Count with <code>set_presence_only(False)</code>
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.

11. Access via API and Python

11.0.8. More information

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

12. Access via Oracle and 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 GAP_PRODUCTS Oracle schema.

12.0.1. 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)) {
```

12. Access via Oracle and R (AFSC only)

```
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()
```

12.0.2. Ex. 1: Join data

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

```
SELECT * FROM GAP_PRODUCTS.JOIN_FOSS_CPUE_HAUL
FULL JOIN GAP_PRODUCTS.JOIN_FOSS_CPUE_CATCH
ON GAP_PRODUCTS.JOIN_FOSS_CPUE_HAUL.HAULJOIN = GAP_PRODUCTS.JOIN_FOSS_CPUE_CATCH.HAULJOIN;
```

12.0.3. 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 GAP_PRODUCTS.FOSS_CPUE_ZEROFILED 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")
```

12. Access via Oracle and R (AFSC only)

```
WHERE SRVY = 'EBS'  
AND COMMON_NAME = 'Pacific cod'  
AND YEAR >= 2010  
AND YEAR < 2021")  
# Save table to local directory  
write.csv(x = a, file = "ebs_pcod_2010-2020.csv")
```

Part VI.

Data Products

To accompany these data, we also produce data products to make using our data more accessible and straightforward.

13. Open source code

13.1. R Packages

13.1.1. akgfmaps R package

Bottom trawl survey maps layers and plotting examples. **POC:** Sean Rohan

13.1.2. coldpool R package

Cold pool area and temperature data products for the Bering Sea. **POC:** Sean Rohan

13.1.3. akfishcondition R package

Groundfish morphometric condition indicators for fish in the Bering Sea, Aleutian Islands, and Gulf of Alaska. **POC:** Sean Rohan

13.1.4. gapindex R package

Calculation of Design-Based Indices of Abundance and Composition for AFSC GAP Bottom Trawl Surveys. **POC:** Zack Oyafuso and Margaret Siple

14. Reports

14.1. Annual Publications

14.1.1. Annual Data Report Tech Memos

The Alaska Fisheries Science Center uses the NOAA Technical Memorandum series to issue informal scientific and technical publications when complete formal review and editorial processing are not appropriate or feasible. Documents within this series reflect sound professional work and may be referenced in the formal scientific and technical literature. These are available online and the latest publications for each survey are listed below.

POCs: Bering Sea: Emily Markowitz, Liz Dawson, Chris Anderson Gulf of Alaska and Aleutian Islands: margaret Siple, Alex Dowlin

14.1.2. North Pacific Groundfish Plan Team

Each year, the survey teams present their findings to the North Pacific Groundfish Plan Team. Find those presentations, recordings, and attachments on the North Pacific Fishery Management Council website. ### Research Briefs

Each year, the Groundfish Assessment Program produces research briefs, or research plans, of their upcoming surveys and research.

14.1.3. Code Books

The Species and Gear Code book is a listing of codes used for fish and invertebrates identified in RACE Division surveys.

POCs: Nancy Roberson

14. Reports

14.1.4. Survey Protocols

Groundfish bottom trawl survey protocols are documented in NOAA protocols for groundfish bottom trawl surveys of the nation's fishery resources.

POCs: Nancy Roberson

14.2. NOAA Fisheries Distribution Mapping and Analysis Portal (DisMAP)

Data from this and other fisheries-independent surveys are used in the NOAA Fisheries Distribution Mapping and Analysis Portal (DisMAP), which provides easy access to information to track and understand distributions of marine species in U.S. Marine Ecosystems.

POCs: NOAA Fisheries Office of Science and Technology

15. Outreach

15.1. Real-time temperatures and survey progress

Maps of near real-time temperatures and survey progress from each of our bottom trawl surveys are posted each business day of the survey. Water temperature affects many species' spawning times, access to food, growth rates, and overall range. Collecting temperature data helps better understand species' habitats and the larger ecosystem.

POCs: Bering Sea: Emily Markowitz, Liz Dawson, Chris Anderson

15.1.1. Community Highlights

Each year, the Groundfish Assessment Program compiles its survey findings for communities around Alaska (Markowitz et al., 2022). *Please note: This document is for informational purposes only and does not necessarily represent the views or official position of the Department of Commerce, the National Oceanic and Atmospheric Administration, or the National Marine Fisheries Service. Not to be cited without permission from the authors.*

POCs: Bering Sea: Emily Markowitz, Liz Dawson, Chris Anderson

Part VII.

Contact us

Suggestions and comments

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 web-page.

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

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.

16. Production run notes

17. 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_0.5.6   rstudioapi_0.15.0  yaml_2.3.7     rmarkdown_2.24
 [9] knitr_1.44        jsonlite_1.8.7    xfun_0.40     digest_0.6.33
[13] rlang_1.1.1       evaluate_0.21
```

17.0.1. 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

17. R Version Metadata

Department of Commerce bureaus stemming from the use of this GitHub project will be governed by all applicable Federal law. Any reference to specific commercial products, processes, or services by service mark, trademark, manufacturer, or otherwise, does not constitute or imply their endorsement, recommendation or favoring by the Department of Commerce. The Department of Commerce seal and logo, or the seal and logo of a DOC bureau, shall not be used in any manner to imply endorsement of any commercial product or activity by DOC or the United States Government.

17.0.2. NOAA License

Software code created by U.S. Government employees is not subject to copyright in the United States (17 U.S.C. §105). The United States/Department of Commerce reserve all rights to seek and obtain copyright protection in countries other than the United States for Software authored in its entirety by the Department of Commerce. To this end, the Department of Commerce hereby grants to Recipient a royalty-free, nonexclusive license to use, copy, and create derivative works of the Software outside of the United States.

18. Data constraints

18.1. 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.
- AFSC RACE Groundfish Data for AKFIN.
- Public Data hosted on the Fisheries One Stop Shop (FOSS) Data Platform.

```
\n@misc{GAPPProducts,\n  author = {{NOAA Fisheries Alaska Fisheries Science Center, Goundfis}}
```

19. 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.

20. Acknowledgments

21. 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.

22. 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.

22.1. 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.

22.2. Collaborators

Our data are used in many annual publications, including but not limited to the list below:

- Alaska Stock Assessments
- North Pacific Groundfish Stock Assessment and Fishery Evaluation Reports
- Groundfish Economic Status Reports for the Gulf of Alaska and Bering Sea and Aleutian Islands
- Alaska Marine Ecosystem Status Report Database
- Southeast Alaska Coastal Monitoring Survey Reports
- Alaska Fisheries Life History Database
- Essential Fish Habitat Research Plan in Alaska

23. References

- Alaska Fisheries Information Network (AKFIN). (2023). *AFSC groundfish 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.
- Markowitz, E. H., Dawson, E. J., Anderson, C., Charriere, N. E., Richar, J. I., Rohan, S. K., Prohaska, B. K., Haehn, R. A., and Stevenson, D. E. (2022). *2022 northern Bering Sea groundfish and crab trawl survey highlights* [Outreach]. University of Alaska Fairbanks Strait Science Seminar; <https://www.youtube.com/watch?v=TGXN2pIDhfc>.
- 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 groundfish 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>